A Composite Syntactic-Semantic Interpretable Text Entailment Approach Exploring Commonsense Knowledge Graphs

Vivian dos Santos Silva

A dissertation submitted to the Faculty of Computer Science and Mathematics in partial fulfillment of the requirements for the degree of Doctor of Natural Sciences

Advisor: Prof. Dr. Siegfried Handschuh

Passau, October 2020

ABSTRACT

Natural Language Processing has an important role in Artificial Intelligence for easing human-machine interaction. Processing human language, though, poses many challenges, among which is the semantics-related phenomenon known as *language variability*, the fact that the same thing can be said in several ways. NLP applications' inputs and outputs can be expressed in different forms, whose equivalence can be verified through *inference*. The *textual entailment* paradigm was established to enable the creation of a unifying framework for applied inference, providing a means of delivering other NLP task from handling inference issues in an ad-hoc manner, using instead the outputs of an inference-dedicated mechanism.

Text entailment, the task of determining whether a piece of text logically follows from another piece of text, involves different scenarios, which can range from a simple syntactic variation to more complex semantic relationships between sentences. However, most approaches try a one-size-fits-all solution that usually favors some scenario to the detriment of another. The commonsense world knowledge necessary to support more complex inferences is also usually employed in a limited way, with most approaches sticking to shallow semantic information, leaving more elaborate semantic relationships aside. Furthermore, most systems still work as a "black box", providing a *yes/no* answer that does not explain the underlying reasoning process.

This thesis aims at addressing these issues by proposing a composite interpretable approach for recognizing text entailment where the entailment pair is analyzed so the most relevant phenomenon is detected and the suitable method can be used to solve it. Syntactic variations are dealt with through the analysis of the sentences' syntactic structures, and semantic relationships are detected with the aid of a knowledge graph built from natural language dictionary definitions. Also, if a semantic matching is involved, the answer is made interpretable through the generation of natural language justifications that explain the semantic relationship between the pieces of text. The result is the *XTE – Explainable Text Entailment* – a system that outperforms well-established tools based on single-technique entailment algorithms, and that also gives an important step towards Explainable AI, allowing the inference model interpretation, making the semantic reasoning process explicit and understandable.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Artificial Intelligence (Russell & Norvig, 2010) is now a pervasive concept, affecting every aspect of modern life. AI technology has been making quick and significant progress in the recent years and can now perform a wide range of tasks that can be said to demand "intelligence", from labeling a picture according to its contents to diagnose diseases based on patients' records (Hosny, Parmar, Quackenbush, Schwartz, & Aerts, 2018; Hirasawa et al., 2018; Poplin et al., 2018). Decision-making systems can now support a wide range of tasks at work, making them easier by analyzing huge amounts of data, identifying patterns, and making predictions (Duan, Edwards, & Dwivedi, 2019). AI also permeates through our personal lives, not only recommending products to buy or movies to watch but even influencing human interactions by suggesting connections in social networks or other applications (Ma, Yang, Lyu, & King, 2008).

One of the reasons why AI became so widespread is the ease with which people can now interact with intelligent applications. Users can now talk to personal assistants on their smartphones or smart speakers or engage in a conversation with a chatbot in the same they would do with another person (Canbek & Mutlu, 2016). Letting users express themselves in their own language makes any technology much more widely accessible, granting Natural Language Processing (NLP) a fundamental role in AI popularization. Be it for recognizing and executing spoken commands, interpreting a question posed in natural language in a search engine and retrieving the relevant answer, or finding the best translation for a piece of text, NLP makes human-machine communication possible and smooths users' interactions with AI technology (Cambria & White, 2014).

Processing human language, though, comes with plenty of challenges. There is a large gap between the way humans deal with language and what computers can do. When people communicate, there's an implicit assumption about their similar "mental structures" and accumulated world knowledge, which provides the context that enables the generation and understanding of highly condensed messages (Nilsson, 2014). Cambria and White (2014) observe that the automatic analysis of natural language text requires a deep understanding of such language by machines, which requires high-level symbolic capabilities, such as the acquisition and access of lexical, semantic, and episodic memories, or the representation of abstract concepts, among others. These symbolic capabilities would enable a machine to emulate humans' mental structures to go one step further beyond what it "sees", that is, what is explicitly expressed in the text.

Cambria and White (2014) also argue that this further step is necessary to go from mere *processing*, which they regard as an interpretation of text at the lexical-syntactic level, to *understanding*, that is, emulating the way the human brain processes natural language, in which every word "activates a cascade of semantically related concepts, relevant episodes, and sensory experiences". In other words, it is not enough to get what a text *says*, it is necessary to grasp what it *means* beyond what it is said. Semantics is, then, a core component of NLP, which must be taken into account by any computational model dealing with natural language.

1.1 Language Variability in NLP

The importance of considering semantic features in NLP tasks becomes evident when we deal with a recurring phenomenon: natural language variability. *Variability* refers to the fact that the same thing can be said in several different ways. Robust language processing applications must be able to deal with the different forms in which their inputs and requested outputs might be expressed (Dagan & Glickman, 2004).

As an example, consider a Question Answering system which receives as input the question "Who painted the *Mona Lisa*?". The standard output for this input would be "Leonardo da Vinci painted the *Mona Lisa*". Likewise, the standard input for the output "Leonardo da Vinci painted the *Mona Lisa?*" would be "Who painted the *Mona Lisa*?", making up the ideal 1:1 input-output relationship. Figure 1.1 shows how this ideal relationship can be perturbed.

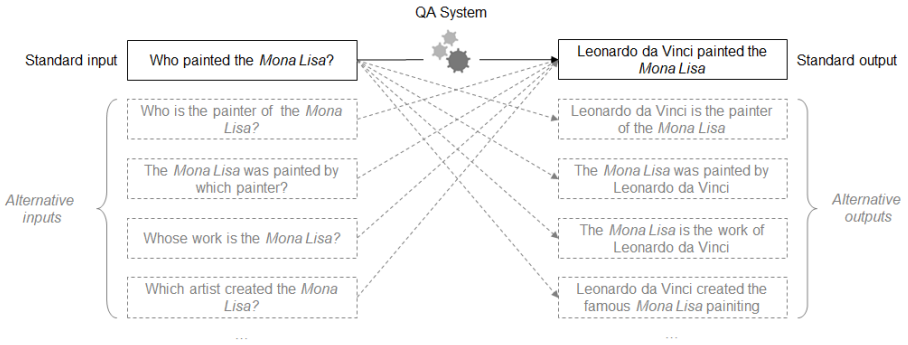

Figure 1.1: Example of a n:n relationship between inputs and outputs in an NLP task.

An efficient Question Answering (or any other NLP) system must account for all possible inputs leading to the same output and all the possible outputs that can be generated for the same input. Nevertheless, knowing and listing all those possibilities beforehand is not feasible. Therefore, it is necessary to determine if an alternative input or output is equivalent to the standard input or output, respectively.

The equivalence between inputs or outputs can be translated into an *inference* relationship. Inference is generally defined as the process by which new facts are concluded from given information in the form of facts or premises (Dagan, Roth, Sammons, & Zanzotto, 2013). Therefore, given the fact that "a human is a mammal", it is possible to infer that "a human is an animal", but it is not possible to conclude that "a human is rational". That means that the truth of the consequent is not enough for reaching a conclusion, because inference is a binary relationship; although we know that the second statement is true, its truth does not follow from the truth of the given fact.

In NLP, inference can be defined as the process of concluding the truth of a textual statement based on (the truth of) another given piece of text (Dagan et al., 2013). As mentioned before, since it deals with natural language, this process must involve not only the explicit information but all the context that can be derived from it trough commonsense knowledge. As Nilsson (2014) summarizes, an intelligent NLP system "capable of understanding a message in natural language would [...] require (no less than would a human) both the contextual knowledge and the processes for making the inferences (from his contextual knowledge and from the message) assumed by the message generator".

1.2 The Textual Entailment Paradigm

Given that language variability is a common and recurring phenomenon, textual inference is a need across numerous NLP applications. Equally numerous are the ways it had been dealt with to meet the requirements of each of these applications. Question Answering, Text Summarization, Machine Translation, Information Retrieval, and Information Extraction systems, among others, would typically have their own inference mechanism to support their main task. Application-independent resources and tools that can assist in the inference task usually cover only specific sub-tasks, such as Named Entity Recognition, Semantic Role Labeling, or Word Sense Disambiguation. The inference itself, however, remained an ad-hoc feature. Built within each specific NLP applications, the inference mechanism used to cover only the application's needs and could not be easily reused by other applications, since researchers in one area might not be aware of relevant methods developed in the context of another application (Dagan et al., 2013).

The *textual entailment* paradigm was introduced to enable the creation of a unified framework for dealing with inference issues. Based on the principle that inference is a task on its own right, text entailment[1] aims at tackling it in an application-independent manner, allowing researchers to focus on core inference issues while still making the results applicable to a number of different NLP applications.

Text entailment is based on a common human understanding of language, and is formally defined as follows (Dagan et al., 2013):

Definition 1.1. *Textual entailment* is defined as a **directional** relationship between pairs of text expressions, denoted by T – the entailing "Text" – and H – the entailed "Hypothesis". We say that T entails H if humans reading T would typically infer that H is most likely true.

The directional nature of text entailment distinguishes it from simple paraphrase, defined as "phrases, sentences, or longer texts that convey the same, or almost the same information" (Androutsopoulos & Malakasiotis, 2010). That implies it is a bidirectional relationship, so, for example, if we have the following two expressions a and b:

a. Shakespeare wrote *Hamlet*.

[1] In this thesis, the expressions "textual entailment" and "text entailment" are used interchangeably.

b. *Hamlet* was written by Shakespeare.

and we say that a is a paraphrase of b, then necessarily b is a paraphrase of a.

Considering the text entailment directionality constraint, three main different scenarios can be observed (Dagan et al., 2013):

(1) T and H are equivalent statements but expressed in different ways. Ex.:

T: The badger is burrowing a hole.
H: A hole is being burrowed by the badger.

(2) H generalizes information from T. Ex.:

T: A dog is riding a skateboard.
H: An animal is riding a skateboard.

(3) H present new information derived from T. Ex.:

T: Iran is a signatory to the Chemical Weapons Convention.
H: The Chemical Weapons Convention is an agreement.

We notice that (1) is also a paraphrase, therefore, a paraphrase is a specific kind of text entailment. What makes entailment a more complex task, capable of encapsulating the inference aspects, are cases like (2) and (3), where, as already pointed earlier, it is necessary to go beyond what we "see" in T and H and consider, besides the message, the background knowledge which contextualizes it and makes the entailment true.

1.3 Problem

Casting inference problems in NLP as textual entailments allows us to abstract from the application which will use it as an input and focus only on the entailment task requirements. That by no means makes it an easier task. In fact, the amount and variety of issues involved in text entailment recognition poses a number of challenges which, despite many advancements in the area, remains to be better addressed.

The first and most prominent challenge is the high variety of phenomena that may be involved, which may range from linguistic constructs, such as changes from active to passive voice, co-reference, or subset (when H is a subset of T), to simple semantic relations, like synonymy or hypernymy, or more complex relationships such as location, cause-effect, or parthood, to name a few. This

is exemplified by the three entailment scenarios listed in Section 1.2: while (1) can usually be resolved syntactically, given that only the sentence structure is altered, and (2) requires only shallow semantic information, such as synonyms and hypernyms, (3) requires knowledge that goes beyond what is expressed in T and H, demanding the use of external commonsense world knowledge to solve the entailment.

Some text entailment approaches focus on exploring the syntactic structures of T and H, trying to transform the syntactic representation of T into that of H to determine whether they are equivalent and confirm the entailment. This kind of approach can fall short of identifying more complex semantic variations, like that observed in (3). On the other hand, techniques concentrating purely on finding semantic relations between T and H will struggle to deal with pairs like the one shown in (1) where only a syntactic variation holds. Addressing many different phenomena while still attending to the specificities of each of them is one of text entailment main pain points.

Another challenge is the knowledge acquisition for solving entailments like (2) and (3). Some text entailment approaches, especially those relying on more complex semantic interpretation, use knowledge bases and linguistic resources to track down semantic relationships between text and hypothesis. WordNet (Fellbaum, 1998) is notably the most commonly used resource, which provides links between terms such as synonym, hypernym or derivational form, among others. Other common lexical resources include VerbNet (Kipper, Korhonen, Ryant, & Palmer, 2006), FrameNet (Baker, Fillmore, & Lowe, 1998), and VerbOcean (Chklovski & Pantel, 2004), which gather classes, frames and semantic relationships such as similarity, antonymy, or enablement, for verbs. Nevertheless, the knowledge that those approaches usually extract from these resources is limited because, besides covering only shallow semantic relationships, it is restricted to the information that is available in a structured format, in the form of explicit links between terms.

While shallow semantic information can suffice in some entailment scenarios, like the one in example (2), more complex semantic relationships, like the one between "signatory" and "document" observed in example (3), will require deeper commonsense world knowledge usually not contained in lexical resource's links. Even inference rules databases like DIRT (D. Lin & Pantel, 2001) will mostly cover equivalence, paraphrase-style relationships, but hardly something further than that. The necessary world knowledge is, though, largely available in the Web, but in the form of natural language text. Unstructured text is

undoubtedly a rich source of knowledge, from which a wide range of semantic relationships can be identified, but that requires extraction methods and representation models that enable easy information structuring and querying, being seldom explored by text entailment approaches.

A further downside of most existing text entailment approaches is their lack of interpretability. Rendering a system interpretable, that is, making it able to explain how it reaches its decisions, is becoming a key requirement due to the recent surge of Explainable AI (Gunning, 2017). Most entailment system will only output a *yes/no* answer and, sometimes, a confidence score, but no further explanation on how this output was computed. This lack of explanation becomes more critical when a more complex semantic relationship is involved in the entailment; if only a syntactic variation is present (for example, an active-passive voice change like the one in example (1) above) it is clear why the entailment holds, because T and H present the same information, but if they present different information and there is a lot going on regarding the use of knowledge and reasoning, it is easier for the final user to trust the answer if they understand how it was reached, which pieces of knowledge were used, and how this knowledge links things together. Automated complex inference is a challenging task, and making a text entailment system interpretable enables us to check whether it is accomplishing this task in a consistent and reliable way.

1.4 Motivation

Besides many advancements in recent years, due to the textual inference task's inherent complexity, there is still a number of open challenges in the text entailment field. As pointed in Section 1.3, there is the variety of phenomena involved in text entailment, the need for world knowledge encoding more complex semantic relationships (which may demand knowledge extraction from natural language text), and the emerging demand for interpretability features (especially when those more complex semantic relationships and, consequently, world knowledge and reasoning are involved). These are though points with room left for further development and refinement. But why is it worth the effort? What is the concrete contribution of advancing text entailment capabilities?

As mentioned before, text entailment can support many other NLP applications that need to deal with language variability. To show how text entailment recognition impacts other areas, and how its advancement benefits the NLP

field as a whole, we list some of its applications in the context of different NLP tasks.

Question Answering

In a Question Answering system, a set of documents are retrieved from which the answer for a natural language question posed to the system is to be extracted. A common strategy is to rephrase the question as an affirmative hypothesis template, with a variable representing the expected answer (Dagan et al., 2013). Consider as an example the question "Who painted '*The Scream*'?". The hypothesis template would then be "X painted '*The Scream*'.", being X the answer to be found. Suppose the retrieved documents are the ones whose relevant excerpts are shown in Figure 1.2. Abstracting the QA system inner workings, we would have that, from the retrieved documents, X = "Edvard Munch" and, then, "Edvard Munch painted '*The Scream*'." is the candidate answer for the above-mentioned question.

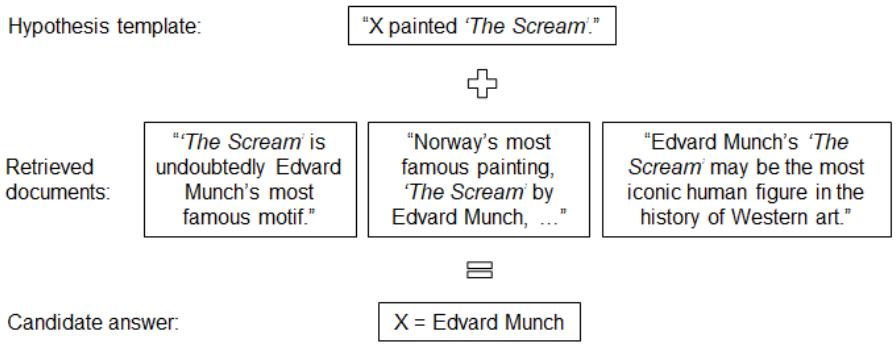

Figure 1.2: A Question Answering example.

The need for inference here translates into the need for verifying if the candidate answer is valid, that is, if it is confirmed by some of the text passages. Casting the problem as a text entailment task, we have:

T: '*The Scream*' is undoubtedly Edvard Munch's most famous motif.
H: Edvard Munch painted '*The Scream*'.

The first document in Figure 1.2 was used as the text T in the entailment pair, but any of the other documents would fit as well. The candidate answer can

be confirmed, then, if the entailment engine confirms that H is in fact entailed by T. Note that, in this example, some commonsense knowledge is required in order to establish the relationship between T and H, like the facts that "a motif is something that was painted" and "if X is Y's motif then Y painted X", demanding more than an analysis of the syntactic features of T and H.

Text Summarization Evaluation

The main purpose of a Text Summarization system is to extract the main idea and topics in a document for representing it in a shortened way (Tas & Kiyani, 2007). Consider the example shown in Figure 1.3, which shows a document and its candidate summary.

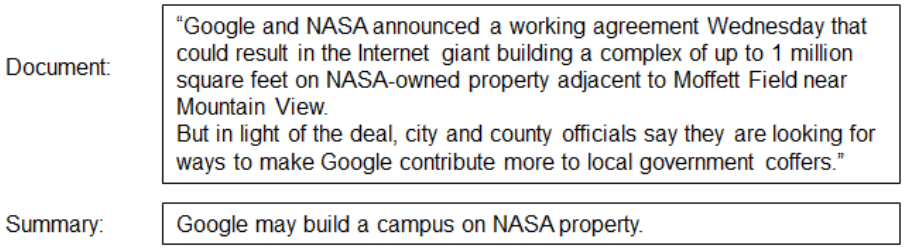

Figure 1.3: A Text Summarization example.

Evaluating if the summary is correct, that is, if it actually grasps the most relevant points and summarizes the main idea in the document, is somewhat subjective and more of a qualitative assessment (Dagan et al., 2013). Nevertheless, a more objective evaluation regarding the *consistency* of the summary is also necessary and can be cast as an inference problem as well. Verifying if the summary is consistent with the original document is equivalent to check whether all the summary's sentences can be inferred by some of the document's sentences.

Again abstracting some side tasks, such as sentence splitting and co-reference resolution, the summary consistency evaluation can be cast as a text entailment problem as follows:

T: Google building a complex of up to 1 million square feet on NASA-owned property.
H: Google may build a campus on NASA property.

Besides the objective consistency evaluation, text entailment can also help in the correctness assessment, as in the work presented by Harabagiu et al. (2007), where they generate a set of candidate summaries and choose the best one by using an entailment engine to evaluate how well each summary matches previously extracted document chunks, and adding this assessment to the scores used to rank the summaries.

Text Summarization Update

Another task related to Text Summarization is the evaluation of the consistency of the summary after its update following the introduction of new documents in the collection. The new documents may contain relevant information, which must be added to the summary, but might as well contain redundant content also present in other documents already in the collection, which should not be included in the summary if it is already there. Consider again the summary in the example in Figure 1.3, and the new document on the same subject added to the collection shown in Figure 1.4. Assuming that the summary must be updated, some candidate summary sentences are extracted from the new document to be added to it.

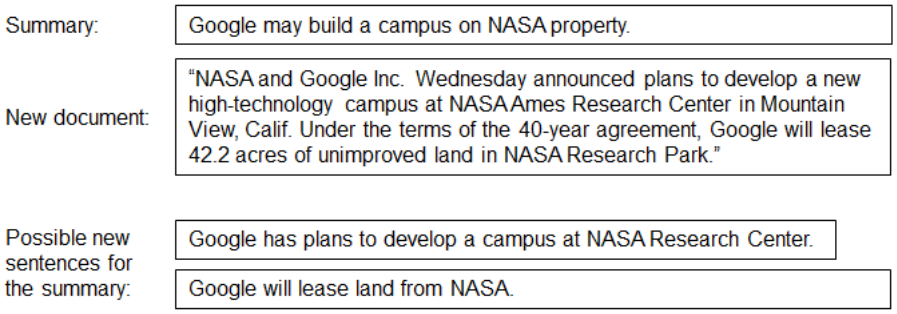

Figure 1.4: A Text Summarization update example.

Inference is necessary here to determine whether the candidate sentence contains new information, that is, if it is not yet included in the summary, which is equivalent to identify whether it can be inferred from the summary. As a text entailment problem, for each of the two candidate sentences in Figure 1.4 we would have:

T: Google may build a campus on NASA property.
H: Google has plans to develop a campus at NASA Research Center.

T: Google may build a campus on NASA property.
H: Google will lease land from NASA.

This case is different from the other examples seem so far, where a positive output from the entailment engine means a positive answer also for the final application. Here, what we are looking for is a negative answer, that is, for the candidate sentence to contain novel information, it should *not* be entailed by the summary. Therefore, the first entailment pair, which would result in an *entailment* answer, leads to the candidate sentence rejection, while the second one, for which a *non-entailment* output would be given, since H cannot be inferred from T, is the right answer for the Text Summarization application.

Automatic Answer Assessment

Automatic Answer Assessment is the task of validating the answers for open questions, that is, questions in an exam for which students need to provide a fully written answer, as opposed to multiple-choice questions. In order to assess the correctness of a student's answer, it must be compared against a predefined reference answer. Consider the example in Figure 1.5, which shows an exam question, its reference answer and a student answer.

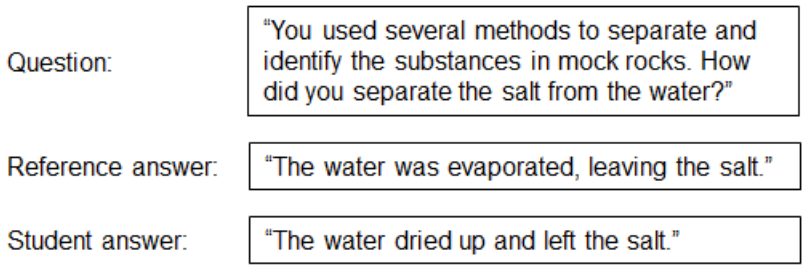

Figure 1.5: An Automatic Answer Assessment example.

Determining whether the student's answer is correct, that is, if it is equivalent to the reference answer, is an inference problem, and can be cast as a text entailment as follows:

T: The water was evaporated, leaving the salt.
H: The water dried up and left the salt.

A student answer is considered correct, then, if it is fully entailed by the reference answer. Text entailment can also be used for Automatic Answer Validation in QA systems, where the focus is to validate the system's outputs, comparing them against the gold standards for evaluating the system accuracy.

Machine Translation Evaluation

A task similar to Automatic Answer Assessment is Machine Translation Evaluation, where the correctness of a translation produced automatically by a machine translation system for a piece of text is verified through its comparison to the reference translation produced by a professional translator. Figure 1.6 shows an example of a piece of text in Spanish, its reference translation, and the translation provided by a translation system.

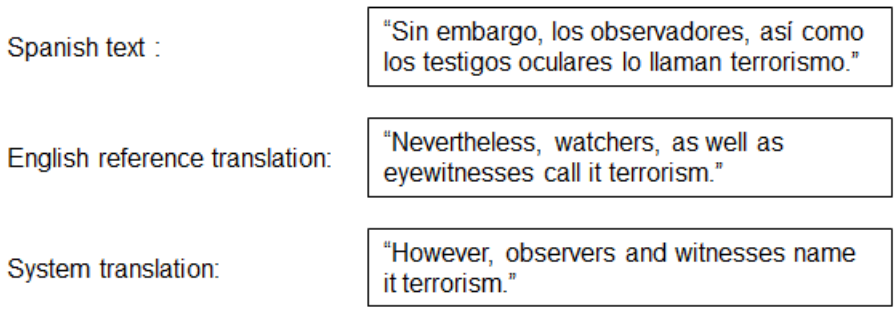

Figure 1.6: A Machine Translation example.

Again, determining whether the system translation is correct is equivalent to answering whether it is entailed by the reference translation:

T: Nevertheless, watchers, as well as eyewitnesses call it terrorism.
H: However, observers and witnesses name it terrorism.

Given that grammatical rules and structures can vary largely from one language to another, automated translations can also diverge syntactically from the reference translation. Hence, the importance of taking into account the

sentences meaning, in addition to the commonly used lexical-syntactic features, such as n-grams (Papineni, Roukos, Ward, & Zhu, 2002), word alignment (Denkowski & Lavie, 2014), and longest common subsequence (C.-Y. Lin & Och, 2004).

The NLP tasks just described are only some of the possible applications of text entailment that illustrates its impact on real-world applications. Inference is a concrete need across many NLP tasks, especially the ones dealing with Web content, whose scale and heterogeneity magnify the language variability issues, so more robust and flexible text entailment approaches are still a welcome addition to the NLP landscape.

Improving an entailment system ability to recognize more complex semantic relationships is a particularly important requirement, which can also boost the capabilities of the other systems using it. As a consequence of enhanced semantic exploration, which involves external knowledge and reasoning, explainability also becomes a substantial demand. The interpretability of a text entailment system can reflect on the interpretability of the target application, helping it to explain its own output, like a QA system justifying the answer for a question through the documents it retrieves but also through the relationships, established by the entailment engine, between these documents and the question, for example. The positive side effects for a variety of applications of advancing these points is, then, the motivation of this work.

1.5 Research Questions

Given the problems described in Section 1.3, and the motivations for addressing them listed in Section 1.4, the goal of this work was to develop an improved text entailment approach capable of tackling some of the current issues in the field. The research was planned and conducted with the aim of answering the following questions:

Q1: *How to deal adequately with the different phenomena that can be present in entailments, which can go from a simple syntactic variation to a complex semantic relationship?*

Q2: *How to improve the recognition of entailments involving semantic relationships, which require commonsense world knowledge to be solved?*

Q3: *Where to find the knowledge necessary to solve entailments involving semantic relationships?*

Q4: *How to maximize the interpretability of the entailment system?*

1.6 Hypotheses

For answering the research questions listed in Section 1.5 and guiding the development of the proposed text entailment approach, this research sought to evaluate the following hypotheses:

H1: *The use of different methods for addressing different (syntactic or semantic) entailment phenomena increases the accuracy of the overall entailment approach.*

The first hypothesis has a direct relation with the research question Q1 and assumes that a single method cannot be suitable for both syntactic and semantic phenomena. An approach made up by a composition of methods, each of them tending the specificities of each phenomenon, will, then, reach better overall accuracy.

H2: *Solving semantic entailments by searching for the key semantic relationship between T and H in a knowledge graph (a knowledge base structured as a set of concepts linked by semantic relationships) increases the accuracy of the system, especially for world knowledge-demanding datasets.*

The second hypothesis relates to the research question Q2 and complements the hypothesis Q1, assuming that the best way to deal with semantic phenomena is to go beyond syntactic features and shallow semantic information and inject commonsense knowledge from a knowledge graph in the reasoning process for establishing the relationship between T and H.

H3: *Natural language dictionary definitions, extracted from lexical resources, can provide the commonsense knowledge necessary to solve semantic entailments.*

The third hypothesis is related to both the research question Q3 and the hypothesis H2, assuming that natural language lexical definitions contain the knowledge necessary for establishing the semantic relationship between T and H, and can be represented as a knowledge graph for being explored as a knowledge source in the entailment recognition process.

H4: *By traversing a definition knowledge graph to find the key semantic relationship between T and H, it is possible to generate a natural language justification from the retrieved path, making the system decision interpretable.*

The fourth and last hypothesis addresses the research question Q4 and follows from the hypotheses H2 and H3, assuming that a knowledge graph built from lexical definitions, besides allowing the recognition of entailments involving more complex semantic relationships, also provides the evidence for explaining the reasoning process followed by the system. This evidence can be formatted into a natural language justification, making the entailment decision interpretable.

1.7 Methodology

The research methodology followed in the development of this work intended to validate the hypotheses listed in Section 1.6, and comprised the following steps:

1. Literature review encompassing the relevant areas related to this work, covering both text entailment recognition systems and semantic interpretability in AI models.

2. Categorization of existing approaches and identification of gaps in the area.

3. Systematic study of world commonsense knowledge-demanding text entailment datasets for categorizing the knowledge needs.

4. Analysis of the linguistic and semantic features of natural language lexical definitions, and development of a conceptual model for representing them.

5. Creation and implementation of a methodology for automatically converting natural language definitions into knowledge graphs.

6. Conceptualization, design, and implementation of the composite interpretable text entailment approach.

   (a) Implementation of a distributional semantics-based graph navigation model for solving semantic entailments and generating natural language justifications.

   (b) Implementation of a routing mechanism for identifying the predominant phenomenon in an entailment pair and, therefore, the suitable method to be used.

   (c) Adaptation of a tree edit distance model for solving syntactic entailments.

   (d) Implementation of a complementary module for extracting additional context information from the sentences in the entailment pair.

   (e) Integration of all the modules into the final system.

7. Design and execution of experiments for the proposed approach evaluation.

   (a) Construction of various definition knowledge graphs from different lexical resources.

   (b) Quantitative evaluation through a comparison with existing baselines, using precision, recall, and F-score as measures.

   (c) Qualitative evaluation of the justifications generated by the system.

   (d) Quantitative and qualitative evaluation of the different definition knowledge graphs.

8. Error analysis, including error classification and quantification.

1.8 XTE: A Composite Interpretable Text Entailment System

The outcome of the research methodology described in Section 1.7 is a composite interpretable text entailment approach materialized in the entailment system called *XTE – Explainable Text Entailment* – a system that uses different methods to tackle different entailment scenarios, integrated as components into a composite approach that performs a *routing*, that is, it analyzes the entailment pair, identifies the most relevant phenomenon present, and sends it to the most suitable component to solve it.

In the context of this thesis, entailment phenomena are split into two broad categories: *syntactic* and *semantic*. For solving syntactic entailments, the approach adopts a tree edit distance algorithm, which operates over a dependency

tree representation of both T and H. For semantic entailments, it looks for the semantic relationships holding between T and H, employing a distributional (word embedding-based) navigation algorithm that explores a graph knowledge base composed of natural language dictionary definitions. By finding a path is this graph linking T and H, the system can provide human-readable justifications that shows explicitly what the semantic relationship holding between them is, which is an important feature that renders the system *interpretable*.

The development of the entailment approach and its implementation into the XTE system were based on the following core **assumptions**:

1. The syntactic and semantic components are self-contained and operate independently, being each able to deliver a final decision.

2. For semantic entailments, the focus is on the *key* semantic relationship, that is, the relationship that maximizes the semantic relatedness between T and H.

3. By using lexical definitions for injecting commonsense world knowledge into the inference process, the approach primarily relies on *intensional*, that is, meaning-based, background knowledge.

4. The interpretability dimension of the resulting system translates as *post-hoc explanations*, meaning that the natural language justifications suffice for explaining the system's decisions, without the need of exposing its internal operations.

The core **contributions** of this approach and of this thesis as a whole are:

- A conceptual model and a methodology for automatically building knowledge graphs from dictionaries.

- A set of definition graph knowledge bases.

- A more flexible way to deal with different entailment scenarios, employing the most suitable method for each entailment phenomenon.

- An interpretable definition-based commonsense reasoning model which, through the generation of natural language explanations, allows the final users to understand and assess the inference process leading to a decision.

- A quantitative and qualitative analysis of different knowledge bases generated from various lexical resources, showing how they compare especially from the interpretability point of view.

The contributions of this work are not restricted to the text entailment field: besides the main contribution for a number of applications that rely on efficient inference models, the knowledge resources developed in the context of this research, including the graph construction methodology and the knowledge bases themselves, can also be reused in any task that demands commonsense knowledge, benefiting the NLP research area in a wider way.

1.9 Thesis Outline

This thesis is organized as follows:

Chapter II – *Literature Review* – presents the literature survey on both the text entailment recognition field and the semantic interpretability area. The text entailment review covers the main characteristics of the entailment problem, including the phenomena and the knowledge requirements involved in entailment recognition. The most prominent base methods, approaches, and evaluation initiatives are analyzed, and an overview of the Natural Language Inference subtask is given. The semantic interpretability survey was carried out in an application-independent manner, covering different AI models to show how each research area define interpretability, provide transparency and/or explanations, and evaluate a model's degree of interpretability. A further analysis draws the relationship between text entailment systems and the Explainable AI requirements, and a gap analysis points at some aspects of text entailment approaches that still need improvements, and that will be addressed in this thesis.

Chapter III – *From Lexicons to Knowledge Graphs* – describes the graph knowledge bases creation process, including the modeling, knowledge extraction, and representation procedures. The most popular existing commonsense knowledge graphs are reviewed and, then, the conceptual model designed for representing lexical definitions in a structured form is described. A rule-based filtering procedure for cleaning a knowledge source, removing ill-formed definitions is detailed, and the complete methodology for automatically converting a dictionary into a knowledge graph is presented.

Chapter IV – *Composite Syntactic-Semantic Interpretable Text Entailment* – presents the proposed composite interpretable text entailment approach. The

first part of the development includes a brief introduction to Distributional Semantics, followed by the description of the Distributional Graph Navigation model, used to solve and justify semantic entailments. The second part details the complete entailment system, including the definition of the routing mechanism which analyzes the entailment pair to choose the model to be used, the adaptation of the Tree Edit Distance model for solving syntactic entailments, the improvements introduced in the semantic entailment module for providing better inputs for the Distributional Graph Navigation algorithm, and the development of the Context Analysis module.

Chapter V – *Evaluation* – describes the experiments carried out for evaluating the proposed text entailment approach. The experimental setup is described, including the definition of the system's main parameters, the description of the datasets tested, the knowledge bases employed and the baselines used for comparison. The quantitative results are presented and discussed, and a qualitative analysis of the justifications produced by the system is also presented. A discussion on the impact of the different knowledge bases in the results, based on their characteristics, is provided through a quantitative and qualitative comparative analysis.

Chapter VI – *Error Analysis* – presents a detailed systematic analysis of the cases for which the proposed approach produced wrong entailment decisions. Errors are identified, classified and quantified, and an analysis on the approach limitations and the possible enhancements to overcome them is presented.

Chapter VII – *Conclusion* – summarizes the thesis, listing the main developments, findings, and contributions. Research hypotheses are recalled and compared against the results obtained to confirm their validity, and future work opportunities are identified.

1.10 Related Publications

The following publications were produced throughout the development of this work, and are related to the aforementioned chapters as indicated:

- Vivian S. Silva, Siegfried Handschuh, André Freitas. Categorization of Semantic Roles for Dictionary Definitions. Cognitive Aspects of the Lexicon (CogALex-V), Workshop at the 26th International Conference on Computational Linguistics (COLING), Osaka, Japan, 2016.

  Includes content described in Chapter III.

- Vivian S. Silva, André Freitas, Siegfried Handschuh. Building a Knowledge Graph from Natural Language Definitions for Interpretable Text Entailment Recognition. 11th Language Resources and Evaluation Conference (LREC), Miyazaki, Japan, 2018.

  Includes content described in Chapter III.

- Vivian S. Silva, André Freitas, Siegfried Handschuh. Recognizing and Justifying Text Entailment through Distributional Navigation on Definition Graphs. Thirty-Second AAAI Conference on Artificial Intelligence (AAAI-18), New Orleans, USA, 2018.

  Includes content described in Chapters IV and V.

- Vivian S. Silva, André Freitas, Siegfried Handschuh. Exploring Knowledge Graphs in an Interpretable Composite Approach for Text Entailment. Thirty-Third AAAI Conference on Artificial Intelligence (AAAI-19), Honolulu, USA, 2019.

  Includes content described in Chapters IV and V.

- Vivian S. Silva, André Freitas, Siegfried Handschuh. On the Semantic Interpretability of Artificial Intelligence Models. arXiv preprint arXiv:1907.04105. 2019.

  Includes content described in Chapter II.

- Vivian S. Silva, André Freitas, Siegfried Handschuh. XTE: Explainable Text Entailment. arXiv preprint arXiv:2009.12431. 2020.

  Includes content described in Chapters IV and V.

Chapter 2

Literature Review

In this chapter, the literature regarding the main areas relevant to this work is reviewed. As described in Chapter 1, this work proposes an approach for recognizing textual entailment (TE) in an interpretable way. Emphasizing the importance of the semantic aspects of entailments, the proposed approach seeks to better address different (syntactic or semantic) phenomena for better entailment recognition while also explaining the reasoning mechanism behind each decision, allowing the user to interpret and understand the system output.

Therefore, this literature review is divided into two main parts, covering the text entailment recognition field, and the semantic interpretability research area. While the text entailment review focuses on the aspects, approaches, and evaluation initiatives for this specific task, the interpretability study is carried out in a cross-application manner, in order to identify the explainability features being developed and adopted by AI models in general to perform a wide range of tasks. Nevertheless, once the big picture of AI interpretability is drawn, its relationship with text entailment task in particular is also analyzed, so the main approaches covered in the first part of the review could be examined also through their interpretability dimension.

The text entailment part starts with a description of the entailment recognition task, the main phenomena it may involve, and the knowledge requirements triggered by such phenomena. A high-level classification and an overview of the main sources available for knowledge acquisition is presented as well. Next, the main base methods used in text entailment recognition are described. Base methods are methods that tackle a single syntactic or semantic task, and that can be combined in different ways by the same approach. The different ap-

proaches building upon these methods are then analyzed. Evaluation initiatives are listed, and the Natural Language Inference, an emerging subtask of textual entailment recognition, is also reviewed.

The semantic interpretability part starts with an overview of the importance of interpreting and understanding the model rationale, followed by an examination of the concept of interpretability across different areas. Next, interpretable AI models, as well as evaluation initiatives, are analyzed and categorized, and an investigation of the impacts of the types of interpretability features and the way they are implemented on the final user wraps up the main findings. A further analysis of text entailment system from the point of view of their interpretability and a gap analysis conclude the review.

2.1 Text Entailment Recognition

The Natural Language Processing field has expanded and advanced to cover a wide range of tasks, providing solutions that allow applications to deal with language in an increasingly better way. Question Answering, Information Extraction, Machine Translation, Relation Extraction, Multi-Document Summarization, and Information Retrieval are some of the tasks whose inputs and outputs are expressed in natural language, and where the correct interpretation of sentences, not only at the lexical and syntactic level but also from the semantic point of view, is possibly one of the most crucial factors of success for the application (Dagan et al., 2013).

While syntactic processing is a fairly mature research field, semantic interpretation is still an area filled with open-challenges, due to the large number of phenomena which can affect the meaning of natural language inputs, outputs, and the relationship between them (Dagan et al., 2013). One such phenomenon is language variability: the fact that the same meaning can be stated in various different ways. NLP applications, then, need a model for this variability phenomenon in order to recognize that a particular target meaning is equivalent to its possible alternatives, that is, it can be inferred from different text variants (Dagan, Dolan, Magnini, & Roth, 2009).

Textual inference is a task by its own and, given that, as observed by Cabrio and Magnini (2014), "language variability manifests itself at different levels of complexity, and involves almost all linguistic phenomena of natural languages, including lexical, syntactic and semantic variation", it can be better addressed

in an application-independent manner rather than as a side functionality in the context of another task. The *textual entailment* paradigm was introduced with the goal of allowing researchers to focus on core inferences issues, while providing a way for dealing with language variability that could be reused across different NLP applications. Recalling the definition given in Chapter 1, we have that text entailment is a directional relationship between a pair of text expressions, denoted by T – the entailing text, and H – the entailed hypothesis. We say that T entails H if, typically, a human reading T would infer that H is most likely true (Dagan, Glickman, & Magnini, 2006).

The variety of phenomena involved in text entailment is one of its most remarkable and challenging characteristics. Cabrio and Magnini (2014), investigating some popular entailment datasets (see Section 2.1.4), highlight two relevant aspects of textual inference: the *logical* dimension and the *linguistic* dimension. The logical dimension refers to the capacity of the inference to prove the conclusion from its premises, independent of the way they are expressed, and the linguistic dimension deals with the linguistic devices that are used to accomplish the goal of the inference and, therefore, are representation-dependent. The representation is the language in which the premises and conclusion (or T and H) are expressed.

According to the rationale that guided this investigation, when textual inferences are seen as logical arguments, they can be classified into three categories, as described by Cabrio and Magnini (2014):

- *Deductive arguments*, whose conclusion follows necessarily from their basic premises.
- *Inductive arguments*, whose conclusion does not necessarily follow from their basic premises.
- *Abductive arguments*, where the reasoning goes from data description of something to a hypothesis that accounts for the reliable data and seeks to explain relevant evidence.

When analyzed in terms of linguistic and knowledge phenomena, textual inferences are divided into five macro categories, each one including the listed fine-grained phenomena:

- *Lexical*: identity, format[1], acronymy, demonymy, synonymy, semantic opposition, hyperonymy, geographical knowledge.

[1]Numerical format variations, e.g. 15 April → 15/04.

- *Lexical-syntactic*: nominalization, verbalization, causative, paraphrase, transparent heads[2].
- *Syntactic*: negation, modifier, argument realization, apposition, list, coordination, active/passive alternation.
- *Discourse*: co-reference, apposition, zero anaphora, ellipsis, statements[3].
- *Reasoning*: apposition, modifiers, genitive, relative clause, elliptic expressions, meronymy, metonymy, membership/representativeness, reasoning on quantities, temporal and spatial reasoning, all the general inferences using background knowledge.

The above-mentioned investigation showed that almost three fourths of entailment pairs correspond to deductive arguments, and that abductive arguments are very rare, which is in line with the definition of textual entailment, where the dependency of H on T has a heavy weight. Regarding the linguistic dimension, there is no absolute quantification, since categories are not mutually exclusive. They also observe that world knowledge has not been categorized separately because it was considered as "an omni-pervasive phenomenon".

In fact, world commonsense knowledge is not a phenomenon, but rather the underlying basis supporting a wide range of different phenomena. Analyzing the proposed classification, though, some issues can be noticed, like overlaps, as in the *apposition* being classified in both the *discourse* and *reasoning* categories, and misplaced phenomena, such as *causative*, which implies a semantic cause-effect relationship, in the *lexical-syntactic* category, suggesting that an overly fine-grained categorization may not be the ideal classification tool.

From the point of view of textual entailment needs, that is, what is necessary to be identified so the entailment can be solved, a macro-categorization grouping similar classes of phenomena, which will demand similar techniques, would comprise the following categories:

- *Lexical-Syntactic Variation*: phenomena referring to changes in sentence structure and phrasing. Text entailments involving such phenomena demand only an analysis of the T and H sentences and simple word replacements using shallow semantic information. Includes most lexical, lexical-syntactic, syntactic and discourse subcategories.

[2]When the syntactic head of a noun phrase is not its most semantically significant noun.
[3]Reported discourse.

- *Semantic Reasoning*: phenomena that depend on supporting known facts linking the meaning of sentences. Text entailments involving this kind of phenomena requires, besides the sentences content analysis, background knowledge from which the supporting facts can be extracted. Includes most reasoning subcategories, but also categories like geographical knowledge, demonymy, or causative, which usually can't be solved with simple word replacement.

Since phenomena are grouped in terms of their needs regarding the analysis of sentence features and the use of external knowledge sources, this macro-categorization allows for a more intelligible review of different entailment methods and approaches. Therefore, in this review we use this categorization as a framework for studying the text entailment state-of-the-art, analyzing how different approaches address these two dimensions and how they make use of external sources of world knowledge.

The survey starts with an overview of the knowledge requirements in text entailment recognition, and then use these requirements as one of the factors for analyzing different entailment methods and approaches. Base methods and the most prominent approaches using one or more of these methods are described, followed by an account of the RTE Challenges, a text entailment evaluation initiative, and an overview of Natural Language Inference, a prominent text entailment subtask.

2.1.1 Knowledge Requirements in TE Recognition

Like any Natural Language Processing task, text entailment abides by human understanding of language, which requires not only the processing of text itself but also of the underlying commonsense world knowledge shared and assumed by humans when they express themselves and communicate in natural language.

Taking into account the need for background knowledge, Dagan et al. (2013) refine the definition of text entailment to include this requirement:

Definition 2.1. A text T entails a hypothesis H if there *exists* some background knowledge K such that T and K *together* entail H while K alone does not.

This definition stresses the role of T in the entailment reasoning for inferring the truth of H, emphasizing that the assumed background knowledge K may be used to augment the information represented by the text in order to entail the hypothesis but cannot entail the hypothesis on its own.

Knowledge acquisition is a major issue in textual entailment research, and several studies tried to identify the linguistic and shallow semantic knowledge as well as the general commonsense world knowledge required to solve entailments across different datasets (Clark et al., 2007; Cabrio & Magnini, 2014; Sammons, Vydiswaran, & Roth, 2010). The most common types of world knowledge (excluding linguistic knowledge, hyponymy, and synonymy) can be classified in the following categories, according to the classification offered by LoBue and Yates (2011):

Form-based Categories

Cause and Effect: an event described in H is an effect of an event in T. Example:

T: Mary gave birth.
H: Mary has a child.

Entailment: Yes

Precondition: an event described in H is a condition for an event in T. Example:

T: Mary gave birth.
H: Mary was pregnant.

Entailment: Yes

Simultaneous Condition: events described in T and H must happen at the same time. Example:

T: John works for Google.
H: Google pays John's salary.

Entailment: Yes

Argument Type: arguments in a relationship can only be of a specific type. In the example below, the predicate "adopt" has an explicit argument ("child"), and the second argument can only be of type "person" (following commonsense, only a person can adopt a child):

T: The child was adopted yesterday.
H: A person adopted the child yesterday.

Entailment: Yes

Prominent Relationship: relationship in H between entities in T is defined by the entities' types. In the examples below, the possessive marker linking the entities "Leonardo da Vinci" and "*Mona Lisa*" could indicate either authorship or ownership, but, given the background knowledge about these entities, "authorship" is the most prominent relationship between them.

T: Leonardo da Vinci's *Mona Lisa* is famous worldwide.
H: Leonardo da Vinci painted the *Mona Lisa*.

Entailment: Yes

T: Leonardo da Vinci's *Mona Lisa* is famous worldwide.
H: Leonardo da Vinci bought the *Mona Lisa*.

Entailment: No

Definitional: the relationship between entities in T and H is given by their basic attributes, contained in their definitions. Example, supported by the WordNet definition for "sell" – "exchange or deliver for money or its equivalent":

T: John sold a car to Mary.
H: John received money from Mary.

Entailment: Yes

Functional: entities in T are linked by a *functional relationship*, where one of the entities only can assume a single value, formally denoted as $\forall x, y, y' R(x, y) \wedge R(x, y') \Rightarrow y = y'$. In the examples below, the entailment is true for the relationship "father of", which is functional (a person can only have one father), but not for the relationship "friend of":

T: Mary's father lives in Italy. Bill is Mary's father.
H: Bill lives in Italy.

Entailment: Yes

T: Mary's friend lives in Italy. Lucy is Mary's friend.
H: Lucy lives in Italy.

Entailment: No

Mutual Exclusivity: an entity can't participate in events described in T and H at the same time. Examples:

T: John is sitting in a chair.
H: John is not walking.

Entailment: Yes

T: John is sitting in a chair.
H: John is not reading a book.

Entailment: No

Transitivity: Entities participate in a *transitive relationship*, formally $R(a, b) \wedge R(b, c) \Rightarrow R(a, c)$. Example:

T: Mary supports the Party. The Party supports John as candidate.
H: Mary supports John as candidate.

Entailment: Yes

Content-based Categories

Arithmetic: involves arithmetic operations, comparisons, and rounding. Example:

T: The plane was carrying 115 passengers and 6 crew.
H: The plane was carrying 121 people.

Entailment: Yes

Geographical: involves knowledge about the type of geographic entities as well as the relationships between them. Examples:

T: Mary visited Rome.
H: Mary visited the capital of Italy.

Entailment: Yes

T: John was in Passau.
H: John was near Austria.

Entailment: Yes

Public Entities: involves knowledge about highly-recognizable named entities. Example:

T: Barack Obama is writing a book.
H: A former US president is writing a book.

Entailment: Yes

Cultural/Situational: involves knowledge of or shared by a particular culture. In the example below, the entailment is possibly true, as long as the equivalence between T and H is accepted as valid in the context of some culture:

T: The cities are a half-hour drive apart.
H: The cities are close to each other.

Entailment: Yes

Membership: includes knowledge about typical membership relationships between entities and organizations. Example:

T: Mary is a minister.
H: Mary works for the government.

Entailment: Yes

Parthood: entities in T and H are linked through the *part-of* relationship, also known as *meronymy*. Example:

T: The forest was destroyed.
H: Trees were destroyed.

Entailment: Yes

Support/Opposition: the action or relationship toward an entity indicates positive or negative feelings toward it. Example:

T: John and Mary are friends.
H: John likes Mary.

Entailment: Yes

Accountability: involves the relationships between entities and the actions they are responsible for. Example:

T: US bombs were thrown in Iraq in 1998.
H: The US military threw bombs in Iraq in 1998.

Entailment: Yes

Synecdoche: involves the relationships between entities and the organizations/institutions they represent. Example:

T: The US president supported the war.
H: The US supported the war.

Entailment: Yes

Miscellaneous Categories

Probabilistic Dependency: a combination of facts in T contributes to making H true, while these same facts in isolation are not enough to support H. Examples:

T: Temperatures will fall below 0°C.
H: It may snow.
Entailment: No

T: Humidity will be high.
H: It may snow.
Entailment: No

T: Temperatures will fall below 0°C and humidity will be high.
H: It may snow.
Entailment: Yes

LoBue and Yates (2011) also cite *Omniscience* as a miscellaneous category, defining it as the assumption that T includes all the necessary information, so any fact not mentioned in T can be discredited. Given the definition of text entailment, this category can be considered redundant, since the truth of H necessarily follows from the information contained in T (in this case, new facts mentioned in H are neither contained in T nor can be derived from T through any other kind of background knowledge). Other relevant categories not covered in this categorization but cited in the study of Clark et al. (2007) are:

Spatial Co-location: entities physically interacting must be at the same location. Example:

T: John was at the party dancing with Mary.
H: Mary was at the party.
Entailment: Yes

Metonymy: an entity in T is replaced in H by another closely related entity that conveys the same meaning. Example:

T: Germany is a big beer consumer.
H: People in Germany consume beer.
Entailment: Yes

The high granularity of this categorization gives us a broad view of the variety of knowledge types and suggests that concentrating on collecting these kinds of commonsense world knowledge can make a large difference in text entailment recognition (LoBue & Yates, 2011). But, on the other hand, this level of detail makes it harder to point to sources for knowledge acquisition for each of the identified types, since they are very specific and atomic. The analysis of this classification suggests that many categories have common characteristics and the knowledge they cover could, consequently, be acquired from the same type of knowledge source. After narrowing this classification, and, hence, the universe of possible knowledge sources, the following macro-categories can be considered:

1. ***Event Chain/Temporal Relations***: knowledge about the interactions between events. Includes categories such as Cause and Effect, Precondition, Simultaneous Condition, Mutual Exclusivity, and Probabilistic Dependency. Figure 2.1 illustrates some of the links between these categories, showing how several of them usually apply to a single set of related events.

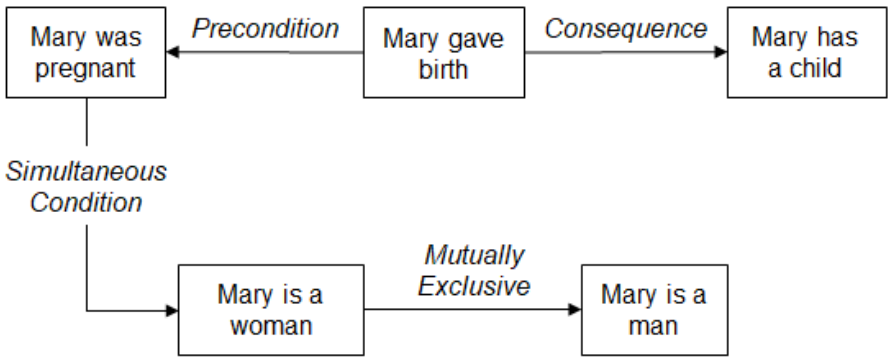

Figure 2.1: An example of the links between temporal relations.

2. ***Geographical/Spatial Relations***: knowledge about the relationships involving geographical entities. Includes relations such as spatial inclusion, location, co-location, nearness, distance, etc. Figure 2.2 shows some spatial relations and the inferences that can be derived from them through transitivity.

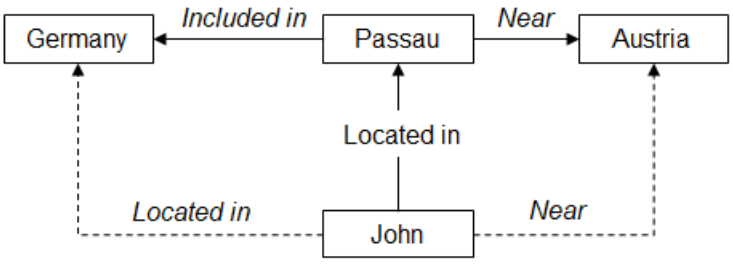

Figure 2.2: An example of the links between spatial relations.

3. ***Entity Types***: knowledge about the type of arguments allowed in a relationship or the types of known named entities. Includes the categories Argument Types, Prominent Relationship, and Public Entities. Figure 2.3 shows the similarities between these categories.

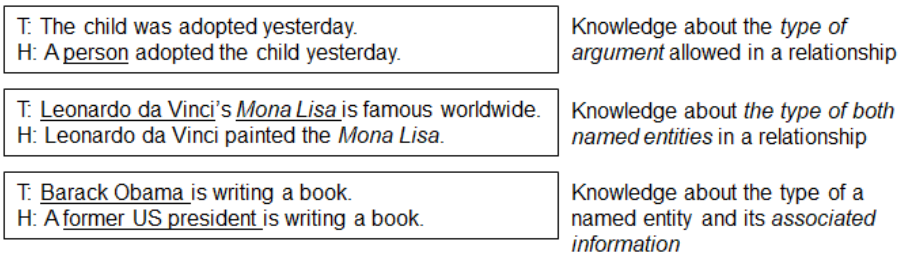

Figure 2.3: Relations involving entity types. From top to bottom: Argument Type, Prominent Relationship, and Public Entities.

4. ***Metonymy***: knowledge about pairs of entities that can be used interchangeably without being synonyms. Includes, besides Metonymy itself, the Accountability and Synecdoche categories. Figure 2.4 shows the common point across these categories.

A possible fifth category is *Definitional Knowledge*, which is any knowledge that can be contained in a definition. This category, though, is not mutually exclusive with the other ones, since a definition can contain a number of different relations between entities. Figure 2.5 shows a few examples of definitions from

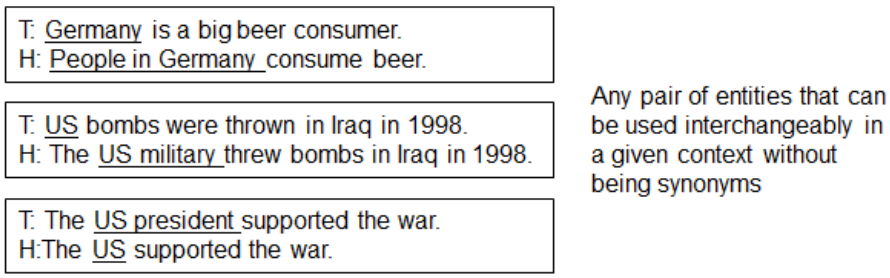

Figure 2.4: Relations involving metonymy. From top to bottom: Metonymy, Accountability, and Synecdoche.

WordNet and the relations that can be extracted from them that are relevant for inference purposes.

Some of the relations in Figure 2.5 are covered by some of the knowledge macro-categories, like *Precondition* in *Temporal Relations*, *Location* in *Spatial Relations*, and *Entity Type/Info* in *Entity Types*, and others, like *Membership* and *Parthood*, are not. In fact, this macro-categorization does not cover other knowledge types, like *functionality* and *transitivity*, which can be better seen as properties of relationships than relations themselves, and which depends on more sophisticated representation mechanisms, like ontologies, to be encoded. *Arithmetic* knowledge, on the other hand, is the kind of information that, if correctly detected, can be dealt with algorithmically, not depending on semantic interpretation. With these exceptions in mind, *Definitional* knowledge could, then, be considered a cross-category, overlapping (but not completely covering) the other macro-categories while also covering further relations. Figure 2.6 drafts the proposed hierarchy of knowledge types required in text entailment recognition. These types refer to the knowledge requirements identified in the investigated datasets, therefore the lists of subcategories are not exhaustive.

2.1.2 Knowledge Acquisition Sources

There are a number of sources from which many of these types of knowledge can be acquired. These knowledge sources have been developed for more general purposes rather than specifically for text entailment, being employed in a wide range of NLP tasks. Traditionally, textual inference have been relying on classical lexical-semantic relations, also called shallow semantic information, such as

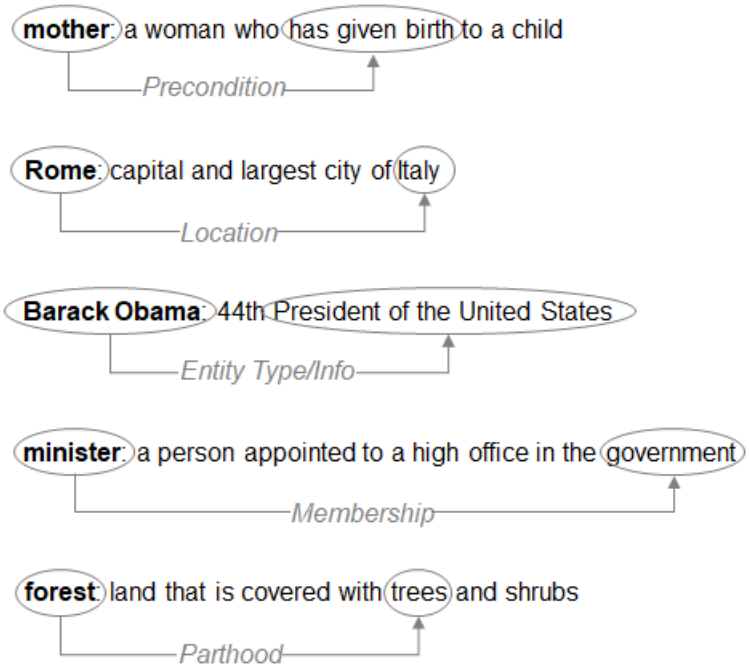

Figure 2.5: Examples of relations covered by definitional knowledge.

synonymy or hypernymy, but the more complex semantic relationships involved in text entailment is gradually led to the integration of more diverse knowledge sources into the reasoning process.

There are resources built manually by professional lexicographers and automatically generated from large corpora. Among the manually built resources, some were constructed with the aim of being machine-understandable, while other ones are primarily for human consumption. WordNet (Fellbaum, 1998) is a computation-oriented manually built lexical resource and the most widely used knowledge source in Natural Language Processing. It contains structured information in the form of links between lexical items, such as synonymy, hypernymy (for nouns) and troponymy (for verbs), and derivationally related form (linking terms which are morphological derivations of one another), which covers the general shallow semantic knowledge needs. It also provides antonymy links, which can, to some extent, help in the identification of mutual exclusivity. WordNet also provides natural language definitions for all its terms, from which

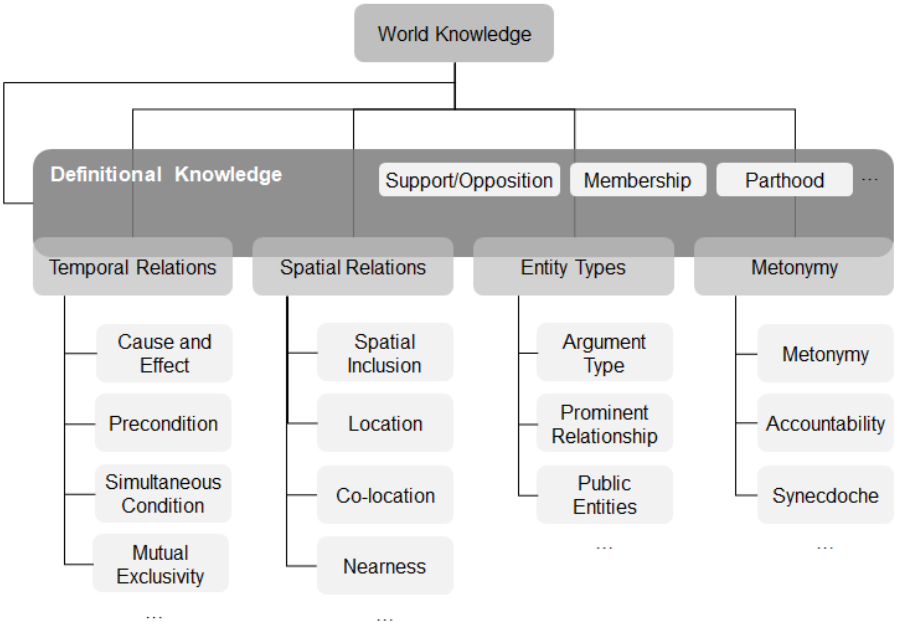

Figure 2.6: The proposed hierarchy of knowledge types required in textual entailment.

definitional knowledge can be extracted, but since it is unstructured information, some previous knowledge extraction process is required.

FrameNet (Baker et al., 1998) is another manually built lexicographic resource which provides frames for verbs. These frames contain information about the argument roles relevant for a specific verb and can provide knowledge about allowed argument types for a given predicate. VerbNet (Kipper et al., 2006) also provides frames but with more refined selectional restrictions on the arguments, which can allow more accurate argument type identification.

Resources created for human consumption can also be a source of knowledge for text entailment, and Wikipedia is the most popular among them, due to its vast size and scope. Its structured information can provide different types of knowledge: *redirect links* can be used for inferring synonymy, *category tags* can provide taxonomic relations, and the *infoboxes*, with values for a number of properties, which vary according to the type of the entity being described, can cover different relations, usually related to entity types and the most important information associated to named entities. As much as WordNet, Wikipedia's

unstructured content is a rich source of definitional knowledge, from which a wide range of relations can be extracted, but, due to the larger amount of information contained in an article, the knowledge extraction methods needed can be even more challenging.

The automatic learning of semantic information from corpora aims at overcoming the inherent incompleteness of manually built resources. VerbOcean (Chklovski & Pantel, 2004) is one such resource, built semi-automatically and oriented to verb description. Instead of specifying frames like FrameNet and VerbNet, it collects semantic relationships between verbs, from which some (limited) knowledge about temporal relations, e.g. *enablement*, *happens-before*, etc., can be extracted.

The Paraphrase Database (PPDB) (Ganitkevitch, Van Durme, & Callison-Burch, 2013) is a resource automatically built from parallel corpora which contains lexical paraphrases and meaning-preserving syntactic transformations. Since paraphrase is a bidirectional relationship, the use of PPDB in text entailment is limited to the detection of equivalence through synonymy. Although these synonymy relations are more elaborated than those that can be found in WordNet, for example, for considering also longer text expressions, other types of knowledge necessary for capturing more complex semantic relationships not necessarily implying equivalence, like cause-effect, parthood, or membership, to name a few, cannot be apprehended from it.

The DIRT (Discovery of Inference Rules from Text) Database (D. Lin & Pantel, 2001) is built in a way similar to PPDB but intended to cover other inference rules besides equivalence. These inference rules hold between binary predicates, in the form $Xpred_1Y \approx Xpred_2Y$, and can cover some temporal and spatial relations. However, since the extraction method was based on similarity between branches of syntactic parse trees, in practice most rules describe paraphrase-style relations, making DIRT one more rich source of shallow semantic knowledge.

The use of external resources and the type of knowledge extracted from them are important features for characterizing a text entailment approach. These points contribute to determining the extent to which it can deal with different entailment scenarios, from simple syntactic variations to shallow semantic relationships to deeper commonsense world knowledge-backed reasoning.

2.1.3 Methods and Approaches

Text entailment recognition was first materialized as a generic model for capturing language variability at a shallow semantic level, meaning that it processed sentences without any explicit interpretation into meaning representations, but rather operating directly over lexical-syntactic units (Dagan & Glickman, 2004). Such approach, as well as other shallow methods, like word overlap and statistical lexical relations, have been gradually giving way to more sophisticated approaches, or *deep methods*, which combine the analysis of the sentence structure with logical features and information from external linguistic resources (Androutsopoulos & Malakasiotis, 2010).

As a common starting point, these approaches translate the text and hypothesis to some kind of structured representation and then try to determine if the representation of the hypothesis is subsumed by that of the text. Therefore, it is possible to distinguish between base methods and approaches. The base methods define the kind of representations to be extracted from T and H, determining the scope of operations that can be performed over these representations. The approaches, then, use one or more base methods to recognize the entailments, handling the representations in different ways.

2.1.3.1 Base Methods

Considering the text entailment field as a whole, base methods can be divided into three categories, which differ mainly in terms of their final goal (Androutsopoulos & Malakasiotis, 2010):

- *Recognition*: the input is a pair of pieces of text, and the output is a judgment, possibly along with a confidence score, on whether or not one piece is entailed by the other one.

- *Generation*: the input is a single piece of text, and the output is a set of text expressions – the hypotheses – that can be entailed by the input text.

- *Extraction*: the input is a text corpus, and the output is a set of pairs of text expressions where one of them is entailed by the other.

While generation and extraction methods can provide background knowledge and rules to be later used in entailment recognition, they are more related to text mining and information extraction than to inference itself. We, then, focus

our analysis on recognition methods, presenting them in what can be thought of as an increasing level of complexity.

Surface String Similarity

Surface String Similarity base methods operate directly over the lexical representations of T and H (Androutsopoulos & Malakasiotis, 2010). This representation is usually a bag-of-words, where lexical units are considered as independent elements, and allows comparisons using similarity measures such as string edit distance, coverage (the number of words H have in common with T), and longest common subsequence (LCS) (Gomaa, Fahmy, et al., 2013).

A combination of several string similarity measures is also possible. In the cases where T is much longer than H, string similarity can be low even if H is fully contained in T, being necessary to introduce complementary approaches, such as comparing H to a sliding window of T's surface string of the same size as H (Androutsopoulos & Malakasiotis, 2010), and use this window to compute the coverage or LCS, for example.

Syntactic Similarity

Syntactic Similarity base methods operate over the syntactic representation of T and H, which can be given by their constituency parse or dependency parse. Syntactic trees can be seen as a graph or treated as a set of independent elements, such as a bag-of-syntactic-dependencies. The comparison, then, can take into account only the number of common edges (considered along with their source and destination nodes) or use more sophisticated tree similarity measures, like tree edit distance (Androutsopoulos & Malakasiotis, 2010).

It is also possible to compare the parse tree of H to subtrees of T. Similarity measures applied to lexical representations, such as coverage and longest common subsequence, also work for syntactic representations. However, when such representations are considered as a unit, graph isomorphism-related measures, like common subtrees or largest common subtree, are the most commonly used (Dagan et al., 2013).

Symbolic Meaning Similarity

Symbolic Meaning Similarity base methods operate over graph representations of T and H, where edges represent semantic relations between words. These relations are usually extracted through semantic role labeling, FrameNet's frames,

or PropBank (Palmer, Gildea, & Kingsbury, 2005) semantic roles. Other resources, such as WordNet, can be also used to identify shallow semantic relations between the words of T and H, such as synonyms, hypernym-hyponyms, nominalizations, and other derivationally related forms, to expand the symbolic meaning representation (Androutsopoulos & Malakasiotis, 2010).

As a graph, the same similarity measures employed for syntactic representations apply. The representations derived from role labeling and frames are somewhat limited as a semantic representation, for being restricted to predicate-argument structures. Nevertheless, they allow the extraction of relations, such as temporal and spatial information regarding an event, not captured by syntactic dependencies.

Logic-based

Logic-based methods operate over logical representations of T and H, using logical theorem provers to recognize the entailment. Due to the high level of complexity involved in defining a standard canonical logical representation, the induced representation "can be thought of as a quasi-formal interpretation, with an associated quasi-formal semantics" (Dagan et al., 2013). These representations are derived from the natural language lexical representations of T and H, usually based on predicate-argument or dependency structures extracted from syntactic parse (Androutsopoulos & Malakasiotis, 2010).

Formal proof-theoretic models encode facts using a formal representation such as Propositional or First-Order Logic and rely on theorem-proving techniques to apply rules of inference to determine the set of facts that can be derived from the already encoded facts (Dagan et al., 2013). General world knowledge can also be used, and linguistic resources like WordNet, VerbNet, and FrameNet can serve as a knowledge base from where logical axioms can be derived (Androutsopoulos & Malakasiotis, 2010).

Rule-based

Rule-based methods use a set of rules to incrementally transform T into H. They can be seen as a relaxation of logic-based methods where, instead of the formal inference rules, a looser concept of rule is employed, corresponding to pairs of related text expressions which are extracted from text corpora through entailment extraction techniques and gathered in resources like PPDB or DIRT (Androutsopoulos & Malakasiotis, 2010).

Also called decoding, this kind of method works by searching a sequence of rules that, applied to T, can turn it into H. Rules can be associated with a confidence score; in this case, the sequence of rules with the maximum score represents the optimal transformation sequence (Androutsopoulos & Malakasiotis, 2010). Although the paraphrase-style rules are more commonly used, handcrafted rules can also be employed, especially for specifying syntactic transformations such as passive/active voice alternations, for example.

Base methods rely on a series of annotations, usually generated during a preprocessing stage, to derive the relevant representation. These annotations include sentence and word segmentation, lemmatization, part-of-speech (POS) tagging, syntactic dependency or constituency parser, named entity recognition, co-reference resolution, and semantic role labeling (Dagan et al., 2013). The amount of annotations differ depending on the method, and the way how the information they provide is used in the entailment recognition also varies across the different approaches, presented next.

2.1.3.2 Entailment Approaches

The base methods presented in Section 2.1.3.1 provide the foundations upon which most text entailment approaches develop a recognition strategy. These approaches can be divided into three main groups, according to their characteristics:

- *alignment* approaches, which are based on comparisons between T and H,
- *transformation* approaches, which, besides comparisons, rely on replacements to turn T into H, and
- *classification* approaches, which combine several representations and similarity measures to learn the patterns distinguishing entailments from non-entailments.

Most approaches use learning techniques, be it for simply deriving weights to be associated with different features or rules, in the case of alignment and transformation techniques, or for delivering the final entailment decision, as it is commonly done by classification systems. The approaches surveyed were chosen according to their performance, being among the best scoring ones in different evaluation tasks, and also due to the representativeness of their strategies within a given category.

Alignment Approaches

Alignment approaches work by searching for similar elements in T and H. This elements can range from a single word to a whole subtree and, therefore, surface string, syntactic and symbolic meaning similarity base methods can be used. The alignment creates *anchors*, which are links between elements in T and H, and each anchor can have a similarity score associated with it. Anchors can be created between identical elements (for which the similarity score will be 1, considering scores ranging between 0 and 1), or between similar ones, when word replacements, such as synonym, nominalizations or verbalizations, are allowed. The set of similarity scores are averaged or added, generating a final score which is compared against a threshold so the entailment decision can be made (Dagan et al., 2013).

The simplest alignment approach is the one operating only at the lexical level, using a bag-of-words representation. Glickman and Dagan (2005) implemented such approach as a probabilistic model that uses document co-occurrence statistics as a measure for aligning lexical units. This approach was, though, only focused on *lexical entailment*, and didn't intend to tackle the textual entailment problem as a whole. MacCartney et al. (2008) take the alignment to the phrase level, defining *phrase* as any contiguous span of tokens, not necessarily corresponding to a syntactic phrase. Replacements are allowed, and their cost is computed through a combination of various string similarity functions, measures of synonymy, hypernymy, antonymy, and semantic relatedness based on WordNet and PropBank relations, and distributional similarity metrics.

Iftene and Balahur-Dobrescu (2007) advance to the syntactic level, generating the dependency tree representation for T and H and looking for a mapping between their entities. They define an entity as a node in the dependency tree associated with information about its edges, which are the dependencies having this node as an argument. For every mapping the *local fitness* is calculated, which is 1 if the entities are identical, and a value between 0 and 1, computed with the aid of relations extracted from WordNet, DIRT, Wikipedia, and VerbOcean, otherwise. The final alignment score is the sum of all local fitness values. Background knowledge is generated ad-hoc, and only for named entities and numbers, covering only the *is-a* and *is-in* relationships. Sammons et al. (2009) add semantic role labeling and split the representation of T and H into multiple layers. Besides the role labeling, they generate a number of

annotations by applying tokenization, POS tagging, named entity recognition, syntactic parsing, and co-reference resolution. Each annotation corresponds to a representation layer, and alignments are performed only between layers of the same type. A set of predefined features are then extracted from the multiple alignments and weighted to generate the final entailment decision.

Hickl and Bensley (2007) presented an approach where they precede the alignment with a sentence expansion step, where T and H are decomposed into a series of *discourse commitments*. These commitments are simpler constructs that are true regardless of the entailment relation between T and H. As an example, they show how the sentence "A Revenue Cutter, the ship was named for Harriet Lane, niece of President James Buchanan, who served as Buchanan's White House hostess." can be decomposed into the commitments "A Revenue Cutter is a ship.", "The ship was named for Harriet Lane", "Harriet Lane was the niece of President James Buchanan.", etc. These commitments are generated with the aid of constituency and dependency parsing, named entity recognition, co-reference resolution, and numeric quantity recognition. The alignment is, then, performed between all the commitments extracted from T and the ones extracted from H. At the alignment step, surface string similarity is adopted, and multiple similarity metrics are used.

Transformation Approaches

Transformation approaches work by successively replacing elements in T to try to transform it into H. If there is a sequence of transformation that can successfully turn T into H, then the entailment is true. Transformations can be performed over the syntactic or symbolic meaning, or logic-based representations of T and H, using formal logic rules or the more general entailment rules extracted from text employed by rule-based base methods (Dagan et al., 2013).

The approach proposed by Dagan and Glickman (2004), who introduced the textual entailment paradigm as a means for providing a unified inference framework, is the first example of a transformation approach. They explicitly abstain from representing the sentence's meaning, performing entailment inferences directly over lexical-syntactic representations obtained from syntactic parsing. The proposed inference mechanism uses a small set of hand-coded rules which specify valid syntactic transformations, and a knowledge base of paraphrase-style entailment patterns extracted from text. As a probabilistic model, it estimates the likelihood that the entailment holds between T and H

using the probabilities returned by each rule in the transformation chain. A similar probabilistic approach was developed by Harmeling (2009), who models the probability of the entailment as the maximally achievable probability of preserving the truth along the sequence of transformations, using hand-coded rules supported mainly by the synonymy, antonymy, and hypernymy relations from WordNet.

Also operating at the lexical-syntactic level, the approach proposed by Bar-Haim et al. (2007) implements a proof system supported by different types of rules, such as lexical, syntactic, polarity, negation, and modality annotation rules. T and H are represented as dependency trees, and at each step of the proof an inference rule generates a derived tree from the tree representation of T. The entailment is considered true if a complete proof is found, that is, if H is generated from T, but also if the total transformation cost is below a predefined threshold, in order to cope with knowledge gaps resulting from insufficient rule coverage. Some rules are hand-coded, while others are generated from WordNet and the DIRT algorithm applied over the Reuters RCV1 corpus[4]. Stern and Dagan (2011) extend this model by adding a set of syntactically-motivated on-the-fly transformations, which include dependency relation change, POS tag change, and multi-word expression cut or expansion, among others, to overcome the rule coverage limitations. The entailment rules used in their approach also comes from a wider range of resources, including VerbOcean, Wikipedia, and DIRT.

Other rule-based approaches introduced more elaborate representations, such as the one proposed by Braz et al. (2005). In a model called *Hierarchical Knowledge Representation*, sentences are represented as concept graphs, where the nodes represent single words or phrases (phrasal nodes in a syntactic parse tree). Each node can have a number of attributes, ranging from word-level syntactic features, such as lemma and POS tag, to semantic role labels assigned to the arguments of a predicate, represented by a verb. Edges between nodes represent relationships of precedence between words in the sentence, and relations between a predicate and its arguments. The same concepts graph representation is used to express rewrite rules, which are inference rules obtained mostly from paraphrase repositories. The inference algorithm then takes as input a pair of sentences, called source (S) and target (T), corresponding to T and H in the entailment task, respectively, and try to determine whether the repre-

[4]https://trec.nist.gov/data/reuters/reuters.html

sentation of S subsumes that of T, incrementally rewriting S with the help of the inference rules. The system uses DIRT and WordNet as external resources. Although it implements a sophisticated combination of modules to perform different types of comparative analysis between S and T, apart from the semantic role labeling, the majority of the features used in the representation remains at the lexical-syntactic level.

With a more rigorous formalism, Raina et al. (2005) proposed an abductive reasoning mechanism using a logical-formula semantic representation of text. This representation is generated from the syntactic constituency parse trees of T and H. Dependencies between nodes are detected through hand-coded rules, then each node is represented as a predicate. Dependencies define predicates' arguments in the resulting logical formula. Over this representation, an abductive theorem prover tries to find the minimum cost set of assumptions necessary to show that H follows from T. These costs are learned from syntactic and semantic features and also from relationships retrieved from WordNet. If the transformation is given by only highly plausible, low-cost assumptions, then the entailment is considered true. The *COGEX* system (Fowler et al., 2005), adopts a similar syntactic parse-derived logical representation, but employs a richer set of axioms in the theorem prover. A small set of hand-coded world knowledge axioms, extracted from training data, and lexical chains extracted from WordNet are added. The prover employs proof by refutation, and the cost for each step in the proof is set manually.

Bos and Markert (2005) presented a logic-based approach that sought to refine both the logical representation of T and H, employing a CCG parser to generate more fine-grained semantic representations, and the use of background knowledge by the theorem prover. As most approaches, it uses lexical relations extracted from WordNet and a few hand-coded general knowledge axioms, but also geographical knowledge axioms extracted from the CIA Factbook[5]. The classic logical theorem prover is coupled with a model builder, and surface string similarity is also used to draw further lexical relations between T and H. The *BLUE* (Boeing Language Understanding Engine) system (Clark & Harrison, 2009) also uses a bag-of-words representation in addition to the logical formula. The surface string similarity, however, is used as a back-off module called only when the logical prover can't reach a decision. It is more limited in terms of world knowledge usage, extracting rules only from WordNet and DIRT and

[5]https://www.cia.gov/library/publications/the-world-factbook/

lacking the ability to deal with more complex semantic variations between T and H, but introduces an important feature: the generation of user-oriented explanations for the entailment decision, derived from the inference proof.

Although transformation approaches are more commonly associated with logic-based and rule-based methods, Tree Edit Distance (TED) is also a widely employed way for checking whether T can be transformed into H through a sequence of simple operations. It refines the string-based edit distance similarity, allowing the manipulation of entailment pairs using their syntactic dependency representations (Dagan et al., 2013), and is a straightforward yet fairly efficient approach for detecting syntactic variants when no complex semantic variation is involved and only shallow semantic knowledge suffices. Kouylekov and Magnini (2005) implement the tree edit distance algorithm proposed by Zhang and Shasha (1989) and apply it to the dependency trees of T and H to find the mapping between them. They define a mapping as a sequence of editing operations, namely node insertion, deletion, and replacement, needed to transform T into H, where each edit operation has a cost associated with it. The operation costs are computed with the aid of a syntactic similarity thesaurus, and the entailment is considered true if the overall cost of the transformation is below a certain threshold.

The *EDITS* (Edit Distance Textual Entailment Suite) package (Mehdad, Negri, Cabrio, Kouylekov, & Magnini, 2009) is a configurable framework which extends the basic Tree Edit Distance model, allowing edit operations at the string, token, and tree-level. It also allows the use of external resources from which shallow semantic information (synonymy, hypernymy, troponymy, and semantic relatedness) can be retrieved for word-level replacements, having been tested with WordNet, VerbOcean and a Latent Semantic Analysis model learned from Wikipedia. Heilman and Smith (2010) proposed another extension to the TED model, expanding the set of edit operations, allowing, besides node insertion, deletion, and replacement, more complex operations such as moving entire subtrees, re-ordering child nodes, and replacing the tree's root node. These new operations enable transformations that represent some common syntactic variations with fewer edits, and, consequently, a smaller cost, but no external resource is used to provide similarity measures or other kinds of relationships.

The *NatLog* system (MacCartney & Manning, 2007) is a hybrid approach that extends the tree edit-based transformations with natural logic operators aimed at dealing with monotonicity, polarity, and general quantifiers. It does not employ a logical representation, operating over the syntactic parse tree

instead. The basic insertion, deletion, and replacement edit operations apply, but after each edit a monotonicity calculus is carried out in order to check whether there was a change in polarity, which can, in turn, change the overall entailment relation between T and H. Although it represents an advancement regarding TED models, it covers only a very limited set of semantic phenomena, namely upward-, downward-, and non-monotonicity, and is not able to deal with more complex inferences requiring world knowledge.

Classification Approaches

Classification approaches work by combining multiple similarity measures, computed over lexical, syntactic and semantic representations, to learn the patterns common to entailment and non-entailment pairs. Any base method can be used, and their outputs are used as the features which will make up the vector $v = \{f_1, f_2, ..., f_m\}$ that represents the [T, H] pair. When a supervised machine learning model is used, pairs are manually classified as entailment or non-entailment, and, after training, the model can then classify unseen entailment pairs by examining their features (Androutsopoulos & Malakasiotis, 2010).

Tatu et al. (2006) use three different measures for classifying entailment pairs: two generated by the COGEX logical prover and one from a lexical alignment module. The two measures generated by COGEX correspond to the proof costs computed for two different logical forms, one generated from a constituency parser and the other one from a dependency parser. World knowledge is provided by hand-coded axioms and by logical rules derived from WordNet. The lexical overlap component computes the edit distance between T and H, allowing synonym replacement. Each component returns a score between 0 and 1 (the higher the score, the better), and a classifier based on the linear combination of the three scores makes the final entailment decision.

Wang and Neumann (2008a) adopt what they call a "divide-and-conquer" strategy, implementing several modules to deal with different features and then selecting the decision of the highest scoring module. Three specialized modules focus on extracting temporal expression, identifying named entities and linking them to events, and analyzing the tree skeleton, which is a representation derived from the syntactic dependency parse tree. Two backup modules, whose results are used in case no expert module reaches a decision, compute the lexical overlap and the dependency edge overlap. Each module returns an *entailment* or *non-entailment* decision and a confidence score. After their predictions are

made, the modules are ordered in a list according to their confidence score and the prediction of the first module on the list that returned a non-null decision is taken as the final result. The extraction of temporal expressions and the identification of the event they refer to can cover some temporal relations. However, apart from WordNet and VerbOcean, which are used to improve the event matching, no other knowledge sources are employed.

More recent approaches are mostly classification-based and usually employ machine learning models such as decision trees, support vector machine (SVM), and neural networks, which are fed with a large set of features. Jimenez et al. (2014) proposed a classifier which uses the notion of *soft cardinality*, a relaxation of the concept of set cardinality that considers, besides identity, the similarity between elements. Each sentence in an entailment pair is seen as a collection of words. The soft cardinality, which rely on several measures commonly used in textual semantic similarity, is used for comparing the collections for T and H and extracting features based on n-gram overlap, explicit semantic analysis (ESA) (Gabrilovich & Markovitch, 2007), part-of-speech grouping, syntactic dependencies, antonymy, hypernymy, and negation. A decision tree model is then trained for predicting the entailment label (either *yes* or *no*). At the feature extraction step, only WordNet is queried for both the usual shallow semantic information and for generating the ESA representation.

Also seeking to reuse the most common features employed in the textual semantic similarity task, Zhao et al. (2014) experiment with different machine learning models for finding the best classifier. Using a large set of features composed of measures derived from sentence length difference, surface string similarity, syntactic similarity, weighted word overlap, n-gram similarity, longest common subsequence, corpus-based co-occurrence, and latent semantic analysis (LSA), they trained SVM, Random Forest, Gradient Boosting, k-nearest neighbors, and Stochastic Gradient Descent models, being the Gradient Boosting the best performing one. WordNet is used for preprocessing and for antonym lookup, but, since the focus is on sentence similarity, no other knowledge sources that could support more complex inferences is considered in this approach.

Zhang et al. (2017) advance to the neural network domain, implementing a supervised Context-Enriched Neural Network (CENN) which uses multiple embedding vectors from different contexts to represent the words in T and H in order to deal with homonymy and polysemy. They then apply different combination methods for optimizing the neural network weights and identifying the preferable context for a given pair of words. This approach is, however, intended

to solve only lexical entailment, that is, entailment at the word level rather than at the sentence level, such as, for example, "dog" entails "animal" and "walk" entails "move". Therefore, despite the more sophisticated architecture, only a few relations, namely synonymy and hypernymy, can be covered by this model. Even if it is seen as a proxy for textual entailment, in the sense that a large number of positive lexical entailments maximizes the likelihood of the overall positive entailment between T and H, it is limited to such shallow semantic relationships, requiring its association with other models if more complex inferences are to be accomplished. Neural networks designed for more general inference are more commonly developed in the context of the Natural Language Inference task, which is reviewed in Section 2.1.5.

Tables 2.1 and 2.2 summarize the most important characteristics of the reviewed approaches. The columns *Lexical-Syntactic* and *Semantic* indicates whether the approach can deal with syntactic variations and more complex semantic relationships, respectively. In the *Lexical-Syntactic* column, *Partial* means the approach can only solve lexical entailment, while in the *Semantic* column, *Partial* means the approach uses only shallow semantic information, such as links from WordNet, semantic role labeling, and word embeddings. Also in this column, *Yes* means it incorporates some kind of world knowledge, even if in a limited way, from other external knowledge sources.

2.1.4 Evaluation Initiatives

The main effort to evaluate text entailment systems is the Recognizing Textual Entailment (RTE) Challenge, which ran from 2004 to 2013 as "an attempt to promote an abstract generic task that captures major semantic inference needs across applications" (Dagan et al., 2009). All RTE Challenges were organized by the European PASCAL and PASCAL-2 Networks of Excellence. RTE-4 through RTE-7 were realized as a track in the Text Analysis Conference (TAC), organized by the U.S. National Institute of Standards and Technology (NIST) (Dagan et al., 2013). RTE-8 was co-located with the Student Response Analysis at SemEval 2013.

In the RTE task, the gold standards are defined by annotators who decide whether a textual entailment relationship holds for a given pair of texts or not, following pre-defined judgment criteria. The criteria are based on the definition of text entailment (see Chapter 1), which is, in turn, grounded on human judg-

Approach	Type	Base Methods	Representation Schema
Glickman & Dagan (2005)	Alignment	String Sim	bag-of-words
MacCartney et al. (2008)	Alignment	String Sim	bag-of-phrases
Iftene & Balahur-Dobrescu (2007)	Alignment	Syntactic Sim, Symbolic Meaning Sim	dependency tree
Sammons et al. (2009)	Alignment	String Sim, Syntactic Sim, Symbolic Meaning Sim	bag-of-words, concept graph, constituency tree,
Hickl & Bensley (2007)	Alignment	String Sim, Symbolic Meaning Sim	bag-of-words, dependency tree
Dagan & Glickman (2004)	Transformation	String Sim, Syntactic Sim, Rule-based	bag-of-dependencies
Harmeling (2009)	Transformation	Syntactic Sim, Rule-based	dependency tree
Bar-Haim et al. (2007)	Transformation	Syntactic Sim, Rule-based	dependency tree
Stern & Dagan (2011)	Transformation	Syntactic Sim, Rule-based	dependency tree
Braz et al. (2005)	Transformation	Symbolic Meaning Sim, Rule-based	concept graph
Raina et al. (2005)	Transformation	Logic-based	logical formula
Fowler et al. (2005)	Transformation	Logic-based	logical formula
Bos & Markert (2005)	Transformation	String Sim, Logic-based	bag-of-words, logical formula
Clark & Harrison (2009)	Transformation	String Sim, Logic-based	bag-of-words, logical formula
Kouylekov & Magnini (2005)	Transformation	Syntactic Sim	dependency tree
Mehdad et al. (2009)	Transformation	String Sim, Syntactic Sim	bag-of-words, bag-of-tokens, dependency tree
Heilman & Smith (2010)	Transformation	Syntactic Sim	dependency tree
MacCartney & Manning (2007)	Transformation	Syntactic Sim, Logic-based	constituency tree
Tatu et al. (2006)	Classification	String Sim, Logic-based	bag-of-words, logical formula
Wang & Neumann (2008a)	Classification	String Sim, Syntactic Sim, Symbolic Meaning Sim	bag-of-words, bag-of-dependencies, dependency tree, concept graph
Jimenez et al. (2014)	Classification	String Sim, Syntactic Sim, Symbolic Meaning Sim	bag-of-words, n-grams, bag-of-dependencies, word vectors
Zhao et al. (2014)	Classification	String Sim, Syntactic Sim, Symbolic Meaning Sim	bag-of-words, n-grams, bag-of-dependencies, word vectors
Zhang et al. (2017)	Classification	String Sim, Syntactic Sim	bag-of-words, word vectors, bag-of-dependencies

Table 2.1: Analyzed textual entailment recognition approaches: type, methods and representation schema.

Approach	Lexical-Syntactic	Semantic	Knowledge Resources
Glickman & Dagan (2005)	Partial	No	N/A
MacCartney et al. (2008)	Yes	No	WordNet, PropBank
Iftene & Balahur-Dobrescu (2007)	Yes	Partial	WordNet, DIRT, Wikipedia, VerbOcean
Sammons et al. (2009)	Yes	Partial	PropBank
Hickl & Bensley (2007)	Yes	Partial	N/A
Dagan & Glickman (2004)	Yes	Partial	N/A
Harmeling (2009)	Yes	Partial	WordNet
Bar-Haim et al. (2007)	Yes	Partial	WordNet
Stern & Dagan (2011)	Yes	Partial	WordNet, VerbOcean, Wikipedia, DIRT
Braz et al. (2005)	Yes	Partial	WordNet, DIRT
Raina et al. (2005)	Yes	Partial	WordNet
Fowler et al. (2005)	Yes	Yes	WordNet
Bos & Markert (2005)	Yes	Yes	WordNet, CIA Factbook
Clark & Harrison (2009)	Yes	Partial	WordNet, DIRT
Kouylekov & Magnini (2005)	Yes	No	Syntactic Similarity Thesaurus
Mehdad et al. (2009)	Yes	Partial	WordNet, VerbOcean, Wikipedia
Heilman & Smith (2010)	Yes	No	N/A
MacCartney & Manning (2007)	Yes	Partial	WordNet
Tatu et al. (2006)	Yes	Yes	WordNet
Wang & Neumann (2008a)	Yes	Yes	WordNet, VerbOcean
Jimenez et al. (2014)	Yes	Partial	WordNet
Zhao et al. (2014)	Yes	Partial	WordNet
Zhang et al. (2017)	Partial	No	N/A

Table 2.2: Analyzed textual entailment recognition approaches: entailment types and knowledge resources used.

ment about the truth of H based on (the truth of) T, also assuming common human understanding of language as well as common background knowledge (Dagan et al., 2009). By common background knowledge, the organizers meant "typical" knowledge of an educated person reading the news (Dagan et al., 2013).

The RTE datasets are composed of entailment pairs extracted from many different application scenarios, such as information extraction (IE), information retrieval (IR), question answering (QA), and summarization (SUM), with the aim of reflecting the way by which each of those applications could, in fact, benefit from an automatic text entailment decision. The data is usually split into development and test sets and the availability of such datasets for training allowed the development of classification approaches (Section 2.1.3.2), where features are extracted from the training examples and then used by machine learning algorithms in order to build a classifier, which is finally applied to the test data to classify each pair either as positive or negative (Bentivogli, Dagan, & Magnini, 2017). RTE-1 through RTE-3 and RTE-8 datasets are freely available, and RTE-4 to RTE-7 datasets are available upon request to NIST[6].

The main task in the RTE Challenges was *classification*, expressed as a two (*yes*/*no*) way decision. Optional tasks, present only in some of the challenge editions, included *ranking* the entailment pairs, according to a confidence score assigned to them, and *justifying* the entailment decision, providing a human readable explanation of the given answer. For classification, the main evaluation measure was *accuracy*, i.e., the percentage of pair correctly judged. Other evaluation measures for this task include *micro-averaged precision*, *recall*, and *F-measure*. For ranking, the *average precision* measure was used, computed as the average of the system's precision values at all points in the ranked list in which recall increases, i.e., at all points in the ranked list for which the gold standard annotation is *yes* (Dagan et al., 2009). Justifications provided to explain entailment decisions were evaluated by human judges, taking into account criteria such as correctness and readability (Voorhees, 2008).

Other evaluation efforts focus on applying textual entailment in the context of a specific application. By reformulating tasks such as Automatic Answer Validation and Question Answering as a textual entailment problem, initiatives like the CLEF Answer Validation Exercise (AVE), the Parser Evaluation task (PETE), the Cross-lingual Textual Entailment (CLET) and EVALITA, for

[6]Links to all datasets at http://tiny.cc/709hcz

Italian language entailments(Dagan et al., 2013), assess the efficacy of a text entailment approach by measuring its impact on the accuracy of a particular application as a whole.

2.1.5 One Step Further: Natural Language Inference

In the last years, with the end of the RTE Challenges, the development of new textual entailment approaches have slowed down. On the other hand, a new subtask derived from it have emerged, leveraged by a new set of datasets and machine learning methods. As opposed to the original text entailment task, which is a binary classification task where the answer is either *yes* or *no* (corresponding to *entailment* or *non-entailment*), the *Natural Language Inference* (NLI) subtask is intended to perform a three-way classification, labeling entailment pairs as *entailment*, *neutral*, or *contradiction*.

NLI is usually associated with deep learning methods, which was enabled by the introduction of large machine learning oriented datasets, such as the *Stanford Natural Language Inference* (SNLI) corpus (Bowman, Angeli, Potts, & Manning, 2015), with approximately 570,000 pairs, and the *Multi-Genre Natural Language Inference* (MultiNLI) corpus (Williams, Nangia, & Bowman, 2018), with around 433,000 examples. Benefiting from the large amount of training data, NLI systems can make use of models and techniques now widely adopted for natural language text processing, among which stand out the *Long Short Term Memory* (LSTM) models (Sundermeyer, Schlüter, & Ney, 2012) and the attention mechanisms integrated into deep neural networks.

In the NLI task, the text (T) sentence is usually called the *premise*, and sentence embedding is a common way of representing the premise and the hypothesis. Bowman et al. (2015), who coordinated the creation of the SNLI corpus, proposed baseline systems where a vector representation of each of the two sentences is produced separately, and the two resulting vectors are then passed to a neural network classifier, which predicts the label for the pair. They implemented a plain Recurrent Neural Network (RNN) and an LSTM RNN, initializing the word embeddings with the 300 dimensions GloVe (Pennington, Socher, & Manning, 2014) vectors. These vectors, which are intended to approximate the sentences' meaning, are, then, the only features used in the models. Rocktäschel et al. (2016) use the same sentence embedding for the premise and the hypothesis, but add an attention-weighted representation of the premise to its embedding by taking into account the alignment between the two sentences.

Wang and Jiang (2016) also use an LSTM architecture supported by an attention-weighted model but propose a different input representation. Instead of generating sentence embeddings for the premise and the hypothesis, they use a match-LSTM to perform word-by-word matching between the sentences, seeking to put more emphasis on the most important word-level matching results. Chen et al. (2017) also adopt a word-level representation, using a Bidirectional LSTM (BiLSTM) model (Graves & Schmidhuber, 2005) that learns to represent a word and its context. BiLSTM is also used to perform the inference composition so the final prediction can be computed. Besides the word embeddings, they also use syntactic parsing information as a feature.

Parikh et al. (2016) stick to a simple LSTM model but propose a decomposable attention model, where the premise and hypothesis are represented as bag-of-words embedding vectors and the alignment between them is computed individually to softly align each word from one sentence to the content of the other one. This is equivalent to decompose the whole inference problem into subproblems that can be solved separately in a parallel manner. They also experiment with intra-sentence attention to encode compositional relationships between words within each sentence in order to capture relevant sequence information.

A number of variants regarding the attention mechanism have been proposed, including *inner-attention* to detect the most important portions of one sentence in the pair regardless of the content of the other one (Y. Liu, Sun, Lin, & Wang, 2016), *self-attention* to model the long-term dependencies in a sentence (Im & Cho, 2017; Gong, Luo, & Zhang, 2017), and *co-attention* to preserve information from all the network layers (S. Kim, Kang, & Kwak, 2019). Despite the differences in the way they attend sentences and align their words, these approaches build upon similar architectures, using LSTM or BiLSTM models with no or almost no feature engineering, and no external resources.

Even though NLI datasets are semantically simpler than text entailment ones, still not all the knowledge needed for the inference is self-contained within the training data. Some approaches try to address this issue by incorporating external knowledge in the inference process. Chen et al. (2018) do this by extracting synonymy, antonymy, hypernymy, and co-hyponymy (relation between sibling words, that is, words having the same hypernym) relations from WordNet and using them as features to enrich the input representation that will feed a co-attention mechanism in a BiLSTM model. Wang et al. (2019) goes further and, besides WordNet, use also ConceptNet and DBPedia as knowledge sources.

In this approach, they represent the premise and the hypothesis as a graph, mapping the words in the sentences to concepts in the knowledge source (when they exist), and then retrieving those concepts' first-level and second-level neighbors. The embeddings of the two resulting graphs are then passed as inputs to the neural network. Relationships retrieved from the knowledge sources will not, therefore, be used to determine the overall semantic relationship between the premise and the hypothesis, but rather be used as additional features through which the attention model computes concepts' degree of similarity to perform the word-by-word alignment.

Although NLI systems show great quantitative improvement when compared with text entailment applications, since all of them use very advanced deep neural network models, the increasing number of different approaches present only incremental improvements among them. Furthermore, even though the advances introduced in the NLI field are arguably invaluable, the high accuracy the approaches achieve may nevertheless be partly influenced by bias in the training datasets. In a study conducted by Gururangan et al., (2018) (in which Samuel R. Bowman, one of the researchers responsible for the creation of both SNLI and MultiNLI, also participated), it is shown that NLI datasets contain a significant number of annotation artifacts that can help a classifier detect the correct class without ever observing the premise. The presence of such artifacts is a result of the crowdsourcing process adopted for the dataset creation, because crowd workers adopt heuristics in order to generate hypotheses quickly and efficiently, producing certain patterns in the data. Through a shallow statistical analysis of the data, focusing on lexical choice and sentence length, they found, for example, that entailed hypotheses tend to contain gender-neutral references to people, purpose clauses are a sign of neutral hypotheses, and negation is correlated with contradiction.

Besides the dataset statistical analysis, they also built a hypothesis-only classifier, showing that a significant portion of SNLI and MultiNLI test sets can be correctly classified without looking at the premise. Then, they re-evaluated high-performing NLI models on the subset of examples on which the hypothesis-only classifier failed (which were considered to be "hard"), showing that the performance of these models on the "hard" subset is dramatically lower than their performance on the rest of the instances. They conclude that supervised models perform well on these datasets without actually modeling natural language inference because they leverage annotation artifacts and these artifacts inflate model performance, so the success of NLI models to date has been overesti-

mated. Poliak et al. (2018) reinforce these conclusions, implementing a similar hypothesis-only classifier but extending the study to other eight datasets besides SNLI and MultiNLI, underlining that such statistical irregularities lead models to skimp over a fundamental principle of textual entailment and, by extension, of NLI: that the truth of the hypothesis necessarily follows from the premise and, then, the premise must be indispensable if actual inference is to be performed.

The problems evidenced by the bias in the datasets show that NLI is still an open challenge, but one more issue can be highlighted: due to their increasingly more complex architectures, NLI models will invariably show poor interpretability, making it even harder for user to know how decisions are reached, and, therefore, if they are reliable or not. Semantic interpretability in AI models in general, and in inference models in particular, is discussed in more detail in the next Section.

2.2 Semantic Interpretability

Artificial Intelligence is becoming a ubiquitous presence in our everyday lives. Supporting and expanding the cognitive abilities of humans or even replacing them, powerful algorithms along with huge amounts of data can now perform a wide variety of tasks, from labeling images to predicting cancer, as well as or even better than a human would do. As AI models grew in processing power and accuracy, they also became more complex, and their predictions, as accurate as they may be, don't bring with them a clear explanation on how they were achieved. Such operation paradigm may bring drawbacks with it because, as it was recently reinforced (Kuang, 2017), AI must "conform to the society we've built – one in which decisions require explanations, whether in a court of law, in the way a business is run or in the advice our doctors give us". This leads us to the need for moving from simply accepting *what a model does*, to interpreting it to understand *how it does* so.

Interpreting the model behavior is not always necessary. Users probably won't want further explanation about the outputs of systems performing voice recognition or image classification, as long as they behave reasonably as expected. But, even in those cases, understanding how the model works can sometimes be of great help. Back in 2015, Google was embarrassed by its Photos service labeling pictures of black people as "gorillas" (Simonite, 2018). Why

was the image recognition software doing so? Certainly not because it is racist and deliberately meant to offend people, but rather because it was making the wrong correlations between the features extracted from the pictures. Google's answer to the problem was to ban the words "gorilla", "chimp", "chimpanzee", and "monkey" from the Photos lexicon altogether, avoiding people (and monkeys) being assigned such labels, and this workaround remains in place, more than four years on. Although there may be many different reasons for Google's approach towards the issue, it becomes clear that the complexity of the machine learning algorithm employed for image recognition may have prevented a proper quick fix, that is, it wasn't possible to quickly interpret the model, identify, and adjust its malfunctioning parts in a timely manner.

However, misclassifications can cause much more than annoyance. The greatest importance of model interpretability rests on decision-making systems, whose outputs can have a material impact on the lives of individuals. Artificial intelligence techniques are now being largely used in tasks such as medical diagnosis, insurance and credit assessment, and criminal recidivism prediction, among others. In those cases, even though a system is known to make accurate predictions, explaining and justifying these predictions may be crucial for users to trust it and make further decisions based on these outputs. Same Google has just released a new AI algorithm capable of predicting heart diseases by analyzing data generated from scans of the back of patients' eyes (Poplin et al., 2018). This algorithm can make the assessment of a patient's cardiovascular risk quicker and easier, but, although all authors say is that it still needs to be thoroughly further tested before being used by doctors, it also needs to be interpretable: it must make clear what information from the scans and what correlations between them are leading to a diagnosis, so doctors can have the necessary evidence to judge whether to follow the system's recommendation or not. Moreover, relying on such technology for prescribing medical treatments won't allow for a quick workaround in case the model start showing undesirable behavior, as in the image classification scenario.

The importance of the so-called Explainable AI lies not only on the need for evidence to support decision making but also on the demand to easily identify biased correlations that could go unnoticed otherwise. Zhao et al. (2017) argues that many prediction models risk reflecting social biases found in data, showing that, using an image dataset containing significant gender bias where the activity "cooking" was over 33% more likely to involve women than men, a trained model further amplifies the disparity to 68% at test time. Transported to the

decision-making scenario, sensitive information such as gender, race, religion or income, for example, can lead to unfair predictions on tasks that involve individual profiling, such as hiring, loan granting or crime prediction, to name a few, where certain groups can be subject to discrimination. Even though the model's predictions may seem to conform to previous decisions, if those decisions were influenced by social biases and this is reflected in the data they generated, an interpretable model can make that clear and allow for more fairness on future verdicts.

Although semantic interpretability is gaining renewed attention due to the increasing use of machine learning, being interpretable is not an issue exclusive to these models, but a requirement for any approach dealing with AI. For example, in an evaluation challenge asking participants to rate the semantic similarity of pieces of texts and explain their decisions (Agirre, Banea, et al., 2015), approaches as different as knowledge base search (Hänig, Remus, & De La Puente, 2015; Hassan, AbdelRahman, & Bahgat, 2015; Banjade et al., 2015), rule-based (Banjade et al., 2015; Karumuri, Vuggumudi, & Chitirala, 2015), referential translation machine (Biçici, 2015), and support vector machine (Agirre, Gonzalez-Agirre, et al., 2015) were employed to implement an interpretable text analysis system. Interpretability is an AI concern rather than purely a machine learning matter. Therefore, in this review we seek to offer a broader view on interpretability, analyzing the efforts of different types of AI models to become more interpretable and how the concept of interpretability is dealt with by each of them.

We start by examining the concept of interpretability itself: how it is regarded across different fields and what shapes it can assume. We then analyze several models that claim to be interpretable as well as the evaluation methods and initiatives intended to measure a model's level of interpretability. Finally, we look at the human-centric aspect of semantic interpretability, classifying models according to how they implement and what they offer as interpretations and pointing to gaps that still need to be addressed to meet explanation requirements from the final user point of view. The list of analyzed models is by no means exhaustive; we sought to pick representative examples of each class, focusing on the ones that emphasize and prioritize interpretability as a driving design choice in the model construction.

2.2.1 Interpretability across Models

Interpretability issues have been gaining the spotlight in recent years due to the fast advancements and widespread utilization of machine learning techniques, especially the ones based on deep neural network models. Such models proved to be powerful predictors, but its complexity usually prevents the user from understanding its internal dynamics. To trust supervised machine learning models, we need them to be not only accurate, but interpretable (Lipton, 2016). However, the need for interpretability is not exclusive to machine learning models. Rule-based models can also grow in size and complexity to a point where users are similarly left unable to comprehend them, also requiring interpretability issues to be taken into account, so that keeping track of those models' decisions becomes feasible. In fact, interpretability has been a key issue in many different areas of AI for many years, notably in the design of fuzzy logic models, giving origin to a number of theoretical and practical studies regarding this topic (Alonso & Magdalena, 2011; Mencar, Castiello, Cannone, & Fanelli, 2011; Alonso, Mencar, Castiello, & Magdalena, 2014).

Being interpretable is sometimes regarded as an inherent attribute of the model. Kotsiantis (2007), for example, states that logic-based algorithms such as Naïve Bayes, decision trees, and rule learners are naturally easy to interpret, while neural networks, SVMs and K-NNs have very poor interpretability. Nevertheless, this is not always accepted as a fact. Lipton (2016) questions this assumption arguing that "neither linear models, rule-based systems, nor decision trees are intrinsically interpretable", adding that "sufficiently high-dimensional models, unwieldy rule lists, and deep decision trees could all be considered less transparent than comparatively compact neural networks". This suggests that interpretability is a property that must be pursued rather than being taken for granted as a result of the model choice.

But what could, in fact, be called *interpretability*? Being tackled in the scope of different approaches, it is natural that the definition of "interpretable model" is not yet something uniformly agreed upon. Lipton (2016) observes that "both the motives for interpretability and the technical descriptions of interpretable models are diverse and occasionally discordant, suggesting that interpretability refers to more than one concept". He argues that the purpose of an interpretation is to convey useful information and that this can be done even without shedding light on a model's inner workings. This leads to the division of interpretability techniques into two broad categories: one referring

to *transparency*, asking *how the model works*, and the other relating to *post-hoc explanations*, inquiring *what else the model can tell* (Lipton, 2016). That means that interpretability can be related to the system output, but also to the system architecture itself.

Most of the models that claim to be interpretable focus on *transparency*, that is, given the input data and the model parameters, it should be possible to step through the calculations that lead to a prediction. Transparency refers not only to the model as a whole (*simulatability*), but also to each of its parts: each input, parameter, and calculation should admit an intuitive explanation (*decomposability*) (Lipton, 2016).

Post-hoc explanations, on the other hand, can be seen as an abstraction layer for the model, as they can summarize and translate the system's procedures into a friendlier format, exempting the user from going through algorithmic details. Natural language explanations, visualizations of learned representations or models, and explanations by example (such as presenting the k-nearest neighbors of a word given its vector representation) are some common approaches for providing post-hoc model interpretation (Lipton, 2016). Explanations are especially necessary when the problem formalization is incomplete, due to, for example, the infeasibility of predicting all possible outputs given all possible inputs, or the abstract nature of some system requirements, such as *fairness* and *trust* (Doshi-Velez & Kim, 2017). In an incompleteness scenario, explanations are a resource to make possibly flawed results (and their causes) clearly visible, allowing users to act on them.

Biran and Cotton (2017), focusing on machine learning, also distinguishes between two research branches, which they call *interpretable models*, equivalent to the transparent models in Lipton's classification; and *prediction interpretation and justification*, that is, the previously seen post-hoc explanations. However, they limit the usage of the term *interpretability* only to the understandability of the model's internal operations, arguing that a system can provide justifications (as the output) without being interpretable, that is, without making clear to the user what its internal procedures are.

The importance of both varieties of interpretability can be noted in the European Union's new General Data Protection Regulation, which took effect as law across the EU in 2018. It covers two points tightly related to interpretability: the *non-discrimination* in automated individual decision-making and the *right to explanation* (Goodman & Flaxman, 2016). The first point refers to algorithmic transparency: systems that support decision-making based on indi-

vidual profiling, such as credit and insurance assessment platforms, should ensure that they do not produce discriminatory results by using variables coding for race or ethnicity, income, or any other sensitive information. That means systems must make clear not only what information they are using but also what correlations algorithms are extracting from data for making predictions. The second point is directly related to the systems' ability to provide (post-hoc) explanations and justify decisions reached after algorithmic assessment in a human-understandable way.

Regardless of the chosen technique to render a model interpretable, the most important aspect to be kept in mind is that interpretability is a human-centered feature. Doshi-Velez and Kim (2017) define interpretability as "the ability to explain or to present in understandable terms to a human", while Alonso et al. (2015) observe that "the importance of the human component implicitly suggests a novel aspect to be taken into account in the quest for interpretability", both emphasizing that the aspects of human cognition should be put at the center of modeling decisions. Offering an interpretation of a model can be seen as a knowledge extraction process, and as such it must take into account the human cognitive factor it inherently involves (Vellido Alcacena, Martin Guerrero, & Lisboa, 2012).

Some studies have tried to draw the users' preferences when it comes to interpretation, especially explanations. Miller et al. (2017) summarize the findings pointing that people usually judge explanations based on *pragmatic influences* of causes, which include usefulness and relevance, among others, rather than the probability that the cited cause is actually true. Also, people prefer explanations that are *simpler* (cite few causes), more *general* (they explain more events), and *coherent* (consistent with prior knowledge), favoring simpler explanations over more likely explanations. They conclude that "giving simpler explanations that increase the likelihood that the observer both *understands* and *accepts* the explanation may be more useful to establish trust, if this is the primary goal of the explanation" (Miller et al., 2017).

As important as defining what interpretability *is* is understanding what *it is for*. Models, and consequently the systems they support, are ultimately designed to address some human user need, and making them interpretable is intended to ensure that the users' requirements are met in the sense that they can use, understand and trust the system in the simplest and most effective way possible.

2.2.2 Interpretability-driven AI Models

In this Section, different types of models which emphasize interpretability are reviewed. The review focus lies on models that were explicitly designed to be interpretable, that is, models for which introducing or increasing the interpretability was a primary requirement driving the design choices. The models are divided into three categories: *data models*, *algorithmic models*, and *hybrid models*, which take into account the content where an interpretation is to be extracted from. These categories are not mutually exclusive: algorithmic models can have minor internal data representations and vice-versa; therefore, for classifying the models, the most predominant characteristic of each of them was considered.

2.2.2.1 Data Models

Interpretable data models are models whose core component is preprocessed data, which goes through some clusterization process for posterior use. Although these models are usually built by some machine learning method, what define such models, rather than the construction procedure, is the shape and content of the final product, which will be used as input for a number of other tasks.

The most outstanding examples of such models are interpretable **Distributional Semantics Models (DSMs)**. These models are grounded in the distributional hypothesis, which states that words that occur in similar contexts tend to have similar meanings (Turney & Pantel, 2010), and allow words to be represented as a vector summarizing their patterns of co-occurrence in large text corpora (more details on DSMs in Chapter 4). The vector representations usually go through a dimensionality reduction process, a mathematical operation that makes the vectors more manageable while still capturing the co-occurrence patterns (Baroni, Murphy, Barbu, & Poesio, 2010), being the most common technique the Singular Value Decompositions (SVD) (Klema & Laub, 1980).

Dimensionality reduction results in vectors whose features correspond to very broad domains of knowledge, such as “food”, “sports” or “education”, for example. A direct consequence of this new representation, as observed by Baroni et al. (2010), is that the underlying abstraction behind most DSMs, the Vector Space Models (VSMs), “might be very good at finding out that two concepts are similar, but they tell us little about the internal structure of concepts and, hence, why or how they are similar” (Baroni et al., 2010). What can be obtained

is an overall similarity score that does not convey any additional information about the relationship between similar words. SVD, in particular, produces matrices where, for most dimensions, it is hard to interpret what a high or low score entails for the semantics of a given word (Fyshe, Wehbe, Talukdar, Murphy, & Mitchell, 2015). This is illustrated in the example posed by Murphy et al. (2012), where they retrieve the latent dimension of an SVD-based model for which the word "pear" has its largest weighting and whose most strongly positively associated tokens are "action", "records", "government", "record", and "search". As can be seen, it is not clear at first sight what the relationships between the words in this dimension are or even to what semantic category they all belong in, making it hard to extract an interpretation from it.

To overcome this problem, some approaches for building more interpretable DSMs have been proposed. An example is *Strudel* (structured dimension extraction and labeling), a corpus-based semantic model that induces semantic information from naturally occurring data using part-of-speech (POS) tagging, lemmatization of the corpus, and a set of extraction templates defined over POS sequences. The model's main goal is to extract dimensions which are "interpretable as properties, automatically annotated with information about the nature of the relation they instantiate" (Baroni et al., 2010). That means it involves some relation extraction functionality, but instead of being predefined, the relations are inferred from the co-occurrence patterns, that is, from the distribution of patterns connecting a concept to its properties. For example, for the concept "book", Strudel associates the following properties, along with the correspondent relations (expressed by either verbs or prepositions): they "are written", "published", and "read", they are "by an author", "from a publisher", "for a reader", and "on a subject", they "have pages" and "chapters", and they "are in libraries". When compared with speaker-generated descriptions, the property-based concept representations produced by Strudel showed to be reasonable both quantitatively and qualitatively.

Murphy et al. (2012) present a technique called *Non-Negative Sparse Embedding* (NNSE) for learning interpretable distributional semantic models. They define interpretability from the point of view of cognitive plausibility, stating that a word representation is interpretable if each of its dimensions is semantically coherent. They measure this coherence through the *word intrusion detection* task, in which, for each dimension, a set is created containing its top five words and an *intruder* word. The sets are then presented to human evaluators who need to identify the intruder. A high precision in this task means

the dimension is interpretable because the human evaluator can easily name the category it is representing and pick out the word that is not a member of this category, i.e., the intruder. One example of such sets is the one composed of the words {"bathroom", "closet", "attic", "balcony", "quickly", "toilet"}; here it is easy to name the category as "house parts" and point "quickly" as the intruder.

CNNSE (Compositional NNSE) is a variation of NNSE intended to allow word and phrase vector to adapt to the notion of composition by learning a DSM that supports semantic composition operations (Fyshe et al., 2015). CNNSE phrasal vector representations have shown to be a better match to actual phrase meanings when judged by human evaluators. *JNNSE* (Joint NNSE) is yet another NNSE variation that combines the representations obtained from large text corpora with brain activation data recorded while people read words (Fyshe, Talukdar, Murphy, & Mitchell, 2014).

The *Explainable Principal Component Analysis* (EPCA) (Brinton, 2017) is yet another technique for generating interpretable vector representations. It builds upon the Principal Component Analysis (PCA) dimensionality reduction approach (Jolliffe, 1986), including a human-in-the-loop stage for refining the data. The EPCA process is performed iteratively: basis vectors generated by the regular PCA are analyzed by a (human) model designer, who excludes any word not related to the general category implied by all the other words in the vector, creating the first explainable basis vector. This vector is then excluded from the input data, over which regular PCA followed by the manual procedure are performed again, generating the second explainable basis vector. The process goes on until all the possible explainable basis vectors have been identified. Although the human curation can doubtless improve the model interpretability, the author presents no study regarding the approach feasibility. Relying on such amount of human interaction could be a prohibitive costly task, given that vector representations are usually built over very large corpora.

DSMs are widely used as input features for machine learning models addressing a large variety of tasks. Using more coherent and interpretable data models can potentially increase the interpretability of the algorithmic models using them as input, providing an additional source of information for the generation of post-hoc explanations. Interpretable algorithmic models are discussed next.

2.2.2.2 Algorithmic Models

Interpretable algorithmic models are models which work by executing a sequence of computations over data, possibly using a set of parameters, to perform a given task. Different from data models, what matters here is not what they *are composed of*, but rather *how they work*.

Fuzzy logic is an example of a field of study where interpretability has long been a central concept. As observed by Alonso et al. (2011), "thanks to their semantic expressivity, close to natural language, fuzzy variables and rules can be used to formalize linguistic propositions which are likely to be easily understood by human beings". But they also point out that, besides being a feature taken for granted even inside the fuzzy community, interpretability is not an intrinsic property of fuzzy models. Although fuzzy logic has a natural inclination to interpretability, whether every element in a fuzzy system can be checked and understood by a human being heavily depends on how the system is designed (Alonso & Magdalena, 2011).

Mencar et al. (2011) define fuzzy model interpretability as a relation between *fuzzy sets* – the basic elements of a fuzzy rule base – and *concepts* – basic units of human knowledge. Fuzzy sets and concepts are linked by the common linguistic terms they refer to. A fuzzy model can be said interpretable when its explicit semantics, that is, the linguistic representation of fuzzy sets, is cointensive with its implicit semantics inferred by a human, i.e., the meanings they infer while reading the linguistic representation of the rules (Mencar et al., 2011).

As the knowledge extracted from data by fuzzy systems must be usually communicated to users, the fuzzy community has been, in recent years, taking into account interpretability issues as a major research concern (Alonso et al., 2014). Balázs et al. (2013) proposed an approach based on *meaning preservation* (MP) – having a common vocabulary with the user, by using linguistic terms in the same sense as the user employs them – and a parameterizable search space narrowing method aimed at adjusting the trade-off between interpretability and accuracy commonly observed in fuzzy systems. Interpretability is not measured, but rather regarded as a binary feature: if the resulting rule base meets the predefined interpretability conditions then it is referred to as a valid interpretable solution.

Mencar et al. (2013) argue that the fulfillment of many interpretability constraints (distinguishability, coverage, special elements, etc. (Mencar & Fanelli, 2008)) is guaranteed if *Strong Fuzzy Partitions* (SFPs) are adopted. A *fuzzy*

partition of the data feature is the collection of the fuzzy sets associated with each linguistic term in the model. The authors propose an approach for defining interpretable SFPs based on *cuts*, points of separation between clusters within the data. Again, interpretability is not measured but considered as a natural outcome of the fuzzy partition-based adopted modeling technique.

Visual representation of rules is another resource employed to allow the interpretation of a fuzzy model. Pancho et al. (2013) describe a technique to present *fuzzy association rules* (FARs) to users in a graph format. Association rules identify and represent dependencies between data items in a dataset. Those dependencies are graphically depicted through fuzzy inference-grams, or *fingrams*, which are networks where nodes represent fuzzy rules and the weighted edges represent interactions between rules. Nodes are always labeled with relevant textual information and the edge weighting naturally leads to the formation of distinguishable groups of rules, each associated with some value for a given variable. Although this graphical representation clearly facilitates the visual analysis and comprehension of fuzzy association rules (Pancho et al., 2013), no evaluation was carried out, and the assessment of its usefulness through user feedback remains to be done.

Going beyond the theoretical aspects explored by most of the research in the area, Riid et al. (2013) proposed a practical application to take advantage of an interpretable fuzzy model. They employ a fuzzy classifier to explain the geographical variation of Estonian folk songs metrical features. *Hierarchical Clustering* (HC), an agglomerative procedure based on the idea that objects tend to be more related to nearby objects than to objects farther away, is used to determine the cluster of geographical regions that show similarities regarding the verse metre. A data-driven fuzzy classifier is then used for analyzing the clusters in order to identify which features of the verse metre are critical in cluster assignment (Riid & Sarv, 2013). The result is a clear separation of the geographical regions into three well delineated groups. However, since no evaluation was performed, it is unclear to what extent the comprehension of how the features lead to the cluster separation is, in fact, interpretable from the user point of view.

Another practical application was presented by Conde-Clemente et al. (2013). A prototype that allows a person with visual disabilities to take their own profile photos was implemented as a fuzzy control system including human-in-the-loop using natural language. The authors argue that, by making use of an approach known as *Highly Interpretable Linguistic Knowledge* (HILK) methodology, they

"are able to represent the extracted knowledge in highly interpretable fuzzy rule-based systems" (Conde-Clemente et al., 2013). HILK is a fuzzy modeling methodology intended to produce classifiers easily comprehensible by humans. As in the previously mentioned fuzzy logic approaches, model interpretability is not measured but assumed to be a natural result of the methodology application, where a set of conditions must be met at design time.

Machine learning has recently become the most active field of research on interpretability. More interpretable models are being pursued not only for more complex techniques such as deep neural networks but also for more traditional models, like the interpretable *decision lists* proposed by Letham et al. (2013). A decision list is a series of *if-then-else* statements, where the *if* statements define a partition of a set of features, the *then* statements define a predicted outcome, and the *else* statements define either a new rule to be applied (if followed by another *if* statement) or a default outcome to be assumed as the prediction, in case none of the previous rules were assessed as true. The proposed approach, called *Bayesian List Machine* (BLM), "produces a posterior distribution over permutations of *if...then...* rules, starting from a large set of possible pre-mined rules" (Letham et al., 2013). The authors follow the assumption that the model is intrinsically interpretable because, given their format, the rules naturally provide a reason for the prediction they lead to. However, no quantitative or qualitative evaluation is presented regarding the model's interpretability assessment.

Lakkaraju et al. (2016) introduced a predictive model based on interpretable *decision sets* (DSs): sets of independent *if-then* rules aimed at being human-interpretable while still showing high accuracy. They define decision sets as "a model class that can both accurately predict class labels and interpretably describe its decision boundaries" (Lakkaraju et al., 2016), and claim they are more interpretable than decision lists because the if-then rules, organized in a non-hierarchical structure, apply independently and can be considered in any order. On the other hand, the rules in decision lists are in the if-then-else format, and each rule depends on all the rules above it, being necessary to interpret the whole hierarchical structure to understand why a given rule is applied. This difference is exemplified in Figure 2.7. Decision sets are concerned only with model transparency, and the authors measure the model interpretability both quantitatively (see Section 2.2.3) and through a qualitative evaluation, which measured to what extent human subjects could interpret the rules, that is, determine which decision each rule was leading to (Lakkaraju et al., 2016).

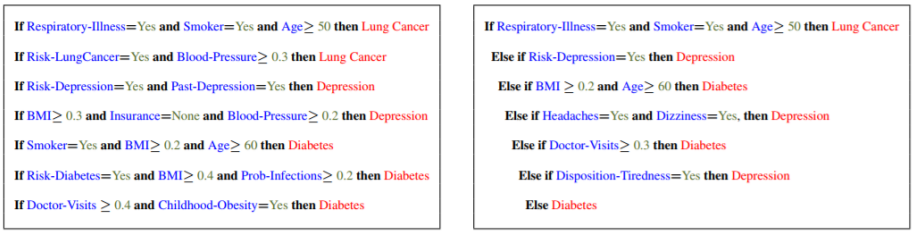

Figure 2.7: A decision set (left) and a decision list (right) learned from a diagnosis dataset, as provided by Lakkaraju et al. (2016). Decision set rules can be interpreted independently, while in decision lists every rule depends on all the rules above it.

Aiming at making statistical modeling accessible to non-experts, Lloyd et al. (2014) introduces the *Automatic Statistician*, a system that analyses data sets and automatically discovers the statistical model describing the data, generating a report with figures and natural language text. The proposed approach, called *Automatic Bayesian Covariance Discovery* (ABCD), focus on Nonparametric Regression Models and uses a greed search procedure to explore the space of regression models, finding, through Bayesian inference, the main components describing the data, such as "periodic function", "linear function", "constant", "smooth function", among others. The discovered model's components are then translated into English phrases, resulting in a report with text, figures, and tables, detailing what has been inferred about the data, besides model checking and criticism (Ghahramani, 2015). The final report allows users to interpret the system decisions without having to go through its internal operations, still, qualitative interpretability evaluation was not presented.

As well as the ABCD approach, the technique proposed by Ribeiro et al. (2016) can be seen as an *interpretability layer* for machine learning models, in the sense that they do not try to adjust the algorithms in order to make them more interpretable, but rather analyze their outputs to build an explanation around it. The *Local Interpretable Model-agnostic Explanations* (LIME) approach can do so for any classifier by learning an interpretable model that approximates a prediction locally (Ribeiro et al., 2016). The explanations consist of a set of artifacts that were relevant for the model's prediction, be it textual (for example, which words were decisive in a text classification task) or visual (for instance, which elements present in a picture influenced an image classification task). These artifacts can assume any representation format and

can be seen as an abstraction for the features used by the model, since these features can be too complex to be presented to the user (like word embeddings, for example). The explanation for a prediction is built by sampling instances in its vicinity and selecting the features that were relevant for those instances classification. This method allows the explanation not only of a single prediction, but also of the whole model, by selecting a set of representative instances, a method called *submodular pick*, and presenting the explanations for them. This allows the users to have an overall insight into how decisions are being made and decide whether they can trust them or not, while the model itself can still be treated as a black box. Quantitative and qualitative evaluations (see Section 2.2.3) show good results for text and image classification tasks. However, what a suitable interpretable representation for an explanation is and how complex is to derive such representation will depend on the model and the task being addressed.

The *Gray Box Decision Characterization* (GBDC) approach (Brinton, 2017) also uses some knowledge about the model's internal procedures to generate post-hoc explanations, without modifying the model itself. That means it could also be used as an interpretability layer for any classifier or regression model. It focuses on characterizing, i.e., explaining a single prediction at a time by performing a sensitivity analysis around its input data vector. The GBDC searches for changes in the basis vector contained in the space region around the input data that lead to changes in the model's output. The explanation for a prediction is then constructed by selecting the features that yield the most significant changes in this specific output. Explanations are provided in natural language but, as the author points out, interpretability evaluation including human subjects is yet to be done.

Datta et al. (2017) go one step further and, besides pointing out which features most likely led to a decision, measure the influence of each of these features on the prediction. By defining a family of *Quantitative Input Influence* (QII) measures, transparency queries can be posed to the model so decisions about individuals and groups can be explained. *Joint influence* of a group of features and *marginal influence* of individual inputs can also be measured when single inputs do not have high influence. Like the GBDC approach, QII forces changes on the inputs to check whether they lead to changes in the output for identifying influential features, but also attributes weights, placing the features in a rating scale that shows clearly how influential each feature was for the decision. The QII measures can also be applied to any classifier and are used

to produce *transparency reports*, which, for a given prediction, shows the QII measures for every input feature in graph format. The transparency reports allow for a clear interpretation of how the decision was reached and make it easier spotting spurious correlations, but still need to be qualitatively evaluated by human users.

Rather than performing post-processing to interpret some model's results, the *Mind the Gap Model* (MGM) opts for embedding interpretability criteria directly into the model (B. Kim, Shah, & Doshi-Velez, 2015). The MGM is a generative approach for feature extraction and selection which aims at identifying not only what features characterize a cluster, but also what features distinguish between clusters. A logic-based feature extraction consolidates dimensions into groups, followed by the identification of important groups based on parameter values, which selects groups having gaps in their parameter values across clusters. The feature groups consist of logical formulas governed by either the *or* or the *and* logical operators, and each group is associated to a cluster through a probability value, which indicates how likely the features in the group are to appear in the cluster. This *feature group* vs. *cluster* matrix is presented to the user as an explanation for the final data clustering. Qualitative evaluation including the participation of human subjects to verify the model interpretability has shown that domain experts could easily understand and quickly write an executive summary of this matrix, as well as finding the differences between clusters.

Figure 2.8 shows some examples of post-hoc explanations produced by the ABCD, LIME, QII, and MGM approaches. ABCD reports rely on natural language descriptions and graphs to describe the discovered data models. LIME shows the features that influenced a prediction in a classification task and QII does the same but placing the features on a numerical scale. MGM presents groups of features along with their likelihoods of belonging in each cluster.

2.2.2.3 Hybrid Models

Interpretable hybrid models are a mix of data and algorithmic models. Besides having a data component, they also include an algorithmic procedure which makes use of this data for a predefined task, meaning that the data component is designed and created specifically to suit the algorithm goals.

Topic models can be seen as hybrid models, as they encompass both a data model – the set of topics, which are collections of words similar to the DSM's

Figure 2.8: Examples of post-hoc explanations: (a) ABCD report (J. R. Lloyd et al., 2014); (b) LIME prediction explanation (Ribeiro et al., 2016); (c) QII transparency report (Datta et al., 2017); and (d) MGM feature matrix (B. Kim et al., 2015).

dimensions – and an algorithmic procedure, used to classify the documents according to the previously discovered topics. Topic modeling algorithms can be applied to large and unstructured collections of documents, discovering their main themes and categorizing their documents based on the discovered themes (Blei, 2012). Formally, probabilistic topic models are resources composed by a set of latent topics for performing the unsupervised analysis of large document corpora, and assume that each document in the collection can be described by a combination of such topics (Chang, Gerrish, Wang, Boyd-Graber, & Blei, 2009).

Although it is usually assumed that the resulting semantic space is always meaningful, its interpretability can't be measured by the commonly used predictive evaluation metrics, that is, quantitative metrics that capture the model's ability to predict the topics for unseen documents. Chang et al. (2009) show that by evaluating topic models generated by three different techniques: *Latent Dirichlet Allocation* (LDA) (Blei, Ng, & Jordan, 2003), *Probabilistic Latent Semantic Indexing* (pLSI) (Hofmann, 1999) and *Correlated Topic Model* (CTM) (Blei & Lafferty, 2005), through the word intrusion and topic intrusion tasks

(see Section 2.2.3). They show that LDA presents the best results for the word intrusion task, and is comparable to pLSI for the topic intrusion one, while CTM showed to be the less interpretable model, despite presenting the highest predictive likelihood among the three. This leads to the conclusion that the highest probability does not entail the best interpretability. As the trend was, in fact, the opposite, this could suggest that as topics become more fine-grained in models with a larger number of topics, they are less useful for humans (Chang et al., 2009).

For learning semantically consistent topics right from the beginning, topic models are starting to be designed with the explicit goal of being interpretable. Among the models that claim to favor interpretability is the *Topical N-Grams* (TNG) (X. Wang, McCallum, & Wei, 2007), a topic model that takes into account the order of words in text to discover topical phrases, in contrast to models such as LDA that generates topics under the bag-of-words assumption, that is, assuming that words are generated independently from each other. The authors argue that whether or not a phrase is a collocation may depend on the topic context and that the TNG is capable of making that distinction. TNG proves to be more interpretable than LDA especially when dealing with generic words, such as "state" or "action", which, alone, may seem misplaced in a topic for being too vague (when interpreted by a human), but are far more meaningful when forming n-grams such as "belief state" or "action selection" in a reinforcement learning-themed topic, for example.

Ramage et al. (2011) also claim that their models, *Partially Labeled Dirichlet Allocation* (PLDA) and *Partially Labeled Dirichlet Process* (PLDP), are more interpretable than unsupervised approaches. PLDA and PLDP combine unsupervised machine learning-based discovery of topics with content annotated with human-provided labels. PLDA "is a generative model for a collection of labeled documents, extending the generative story of LDA to incorporate labels, and of Labeled LDA to incorporate per-label latent topics". PLDP "replaces PLDA's per-label topic mixture [...] with a Dirichlet process mixture model" (Ramage et al., 2011). Both models learn the topic structure within the scope of the observed labels, which impose a kind of semantic constraint on the resulting model. However, although they argue that this constraint improves correlation with similarity judgments, the case studies presented involves no human participation. Tying discovered topics to human interpretable labels can indeed produce a more interpretable topic structure, but how humans actually evalu-

ate both the topic's consistency and the adequacy of the associations between topics and documents provided by these two models remains to be measured.

The work introduced by Song et al. (2011) is a kind of topic modeling focused on short texts, such as Twitter messages. Since such short texts don't provide enough contextual information, the topics can't be derived solely from their content. Instead, the words in the text are mapped to entities, which can be either concepts (e.g. "country"), instances of concepts (e.g. "China", "India") or attributes of concepts/instances (e.g. "language", "population"), in a probabilistic knowledge base called *Probase*. Those entities, along with the relationships between them, will then provide the context for the topic discovery, which the authors call *conceptualization*, i.e., the definition of a set of concepts that best describe the text's content. The conceptualization is carried out through *Bayesian Inference* (BI), benefiting from the large size of the database. The quantitative evaluation, which uses clustering-related measures, such as purity, adjusted random index (ARI) and normalized mutual information (NMI), shows the approach yields good accuracy but, although the resulting topics presented as examples are in fact easily interpretable, no evaluation regarding the model interpretability itself was performed.

Table 2.3 summarizes the main characteristics of the analyzed data, algorithmic and hybrid models.

2.2.3 Evaluation Methods and Initiatives

Despite the crescent effort to build more interpretable models, measuring their level of interpretability is still a challenge, regardless of the type of the model. Some evaluation methods have been proposed, but always in the context of a specific model, whose characteristics determine what will be assessed and which measures are relevant for the evaluation.

An initiative towards the qualitative evaluation of topic models is the set of tasks proposed by Chang et al. (2009), intended to measure the model's interpretability. In the *word intrusion* task, a human evaluator needs to find the word that does not belong in the topic, i.e., the *intruder*, assessing the model's data component. The algorithmic component is evaluated through the *topic intrusion* task, which checks whether the topics assigned to a document by a topic model match the human judgments of the document's content, by presenting a document and a set of topics related to it to a human evaluator and

Approach	Model	Type	Transparency	Explanation	Presentation
Strudel (Baroni et al., 2010)	DSM	data	yes	no	textual
NNSE (Murphy et al., 2012)	DSM	data	yes	no	textual
CNNSE (Fyshe et al., 2015)	DSM	data	yes	no	textual
JNNSE (Fyshe et al., 2014)	DSM	data	yes	no	textual
EPCA (Brinton, 2017)	DSM	data	yes	no	textual
MP (Balázs & Kóczy, 2013)	FL	algorithm	yes	no	textual
SFP (Mencar et al., 2013)	FL	algorithm	yes	no	textual
FAR-Fingrams (Pancho et al., 2013)	FL	algorithm	yes	no	visual
HC (Riid & Sarv, 2013)	FL	algorithm	yes	no	textual/visual
HILK (Conde-Clemente et al., 2013)	FL	algorithm	yes	no	textual
DS (Lakkaraju et al., 2016)	ML	algorithm	yes	no	textual
BLM (Letham et al., 2013)	ML	algorithm	yes	no	textual
ABCD (J. R. Lloyd et al., 2014)	ML	algorithm	no	yes	textual/visual
LIME (Ribeiro et al., 2016)	ML	algorithm	no	yes	textual/visual
GBDC (Brinton, 2017)	ML	algorithm	no	yes	textual
QII (Datta et al., 2017)	ML	algorithm	no	yes	visual
MGM (B. Kim et al., 2015)	ML	algorithm	yes	yes	textual
TNG (X. Wang et al., 2007)	TM	hybrid	yes	no	textual
PLDA (Ramage et al., 2011)	TM	hybrid	yes	no	textual
PLDP (Ramage et al., 2011)	TM	hybrid	yes	no	textual
BI (Song et al., 2011)	PM	hybrid	yes	no	textual

Table 2.3: Analyzed interpretable AI models. DSM = Distributional Semantics Model; FL = Fuzzy Logic Model; ML = Machine Learning Model; TM = Topic Model; PM = Probabilistic Model. The Presentation column indicates how the model transparency and/or post-hoc explanation is presented to the user.

asking them to identify the intruder topic. The underlying rationale is the same for both tasks: if the topic is semantically consistent, that is, if the words that compose it refer to the same semantic category, even if in a broad sense, then humans can easily interpret its meaning and point out what does not belong to it, be it a single word or a whole document. The word intrusion task is also commonly used to assess the semantic coherence of DSM dimensions (Murphy et al., 2012; Fyshe et al., 2015).

In the fuzzy logic context, evaluating model interpretability is not yet a widespread practice. As seen in Section 2.2.2, most fuzzy models claim to be interpretable because they meet a set of constraints at design time, but no quantitative or qualitative evaluation is in fact carried out so the model interpretability can be assessed from a human point of view. Despite that, a large number of objective and subjective indexes for assessing the interpretability of fuzzy systems have been proposed, covering different granularity levels inside the model (Alonso et al., 2015). These indexes are intended to analyze both the structural-based interpretability and semantic-based interpretability of a model, and takes into account a set of constraints and criteria regarding each of the model's abstraction layers, from the lowest to the highest: fuzzy sets, fuzzy partitions, fuzzy rules, and fuzzy rule bases. Structural constraints and criteria refer to the internal organization of the elements that compose a fuzzy model and determine its *readability* level, while semantic constraints and criteria refer to the way the fuzzy system behaves, that is, how it reaches its results, dictating the model's *comprehensibility* level. A detailed list of constraints and criteria, as well as a description of the most outstanding interpretability indexes for evaluating fuzzy models, is provided by (Alonso et al., 2015).

Mencar et al. (2011) go in a different direction and propose an interpretability evaluation method using the notion of *cointension*, defined as "a measure of proximity of the input/output relations of the object of modeling and the model" (Mencar et al., 2011). Exploiting the *logical view*, a set of properties expected to contain the (approximated) implicit semantics and defined as "the propositional structure of the rules in the knowledge base, responding to the laws of formal logics both for the fuzzy rule-based inference and the user thinking" (Mencar et al., 2011), the rule base is transformed into a different one logically equivalent to it. The model interpretability is then measured by the extent to which the retained logical equivalence of the rule bases corresponds to the semantic equivalence. In practice, this is a quantitative assessment of interpretability: the model is interpretable if the logical view of its rules is cor-

rect, what is measured by the comparison between the accuracies of the two rule bases (the lower the difference, the higher the logical view correctness and, consequently, the interpretability).

Regarding machine learning models, Doshi-Velez and Kim (2017) go in a more theoretical direction and propose, rather than a method, a taxonomy for the evaluation of model interpretability. They observe that current evaluation techniques either assesses interpretability in the context of an application or via a quantifiable proxy. In the first case, a system is considered interpretable if it is useful in a practical application (or a simplified version of it); in the second scenario, a model is considered intrinsically interpretable and is just subjected to optimization algorithms. They argue that, although these approaches may seem, at first, reasonable, they lack rigor and, in order to compare methods in an effective way, evaluation criteria must be formalized and based on evidence. Hence, in the proposed taxonomy, the evaluation method, which should be based on sets of task- and method-related latent dimensions, may vary from model to model. It must also take into account the focus of the contribution, which can range from assessing the reliability of a real-world application from the human point of view to better optimizing a given model with regard to interpretability requirements.

In a more practical fashion, Lakkaraju et al. (2016) propose, in addition to asking human subjects to relate a given decision to the rules that generated it, a quantitative evaluation for assessing decision sets' interpretability. They define four metrics: *size* (the number of rules in the set), *length* (the size of each rule), *cover* (the number of points in the data set covered by each rule) and *overlap* (the number of features covered by more than one rule). Under this evaluation framework, the lowest the size, length and overlap of a decision set, the highest its interpretability (cover is used as an intermediary metric to compute overlap).

The evaluation methods introduced by Ribeiro et al. (2016) also target machine learning models, but focus on assessing the interpretability of the post-hoc explanations generated for their predictions. The set of evaluation tasks includes both quantitative and qualitative assessments. The quantitative tasks aim at determining if the explanations are faithful to the model, if a single prediction is trustworthy, and if the model as a whole is trustworthy. As the explanations are basically composed of a set of features, this is done by creating gold standard sets of features and computing the model's recall for each explanation it generates (for measuring faithfulness); by marking a subset of features as "untrustworthy" and measuring the rate of such features in an explanation (for

computing the prediction's trustworthiness); and by adding artificial "noisy" features to the model to evaluate how many of its predictions can be trusted, i.e., how many predictions do not include the untrustworthy noisy features in their explanations (for measuring the model's overall trustworthiness).

The qualitative assessment is carried out by showing human subjects the explanations generated by two classifiers and measuring to what extent these explanations help them to make decisions. First, users need to select the best classifier based on the explanations they provide for a text classification task, where one classifier uses untrustworthy features (words) and the other one (considered the best one) does not. Second, also in the context of a text classification task, (non-expert) users need to improve a classifier, by analyzing the explanations and removing the features (words) they consider untrustworthy for subsequent model training. Using an image classification task as context, the third evaluation task forces a wrong correlation by selecting all images from a class having a given feature that does not generalize in the real world (in this case, the classes used were "wolf" and "husky", and all the wolf pictures contained snow, which would end up being used by the classifier for generalization). The objective is to evaluate if observing the explanation for a prediction, which in this case is a super-pixel of the image, users can have insights on which features are being used by the model, identify if it is making spurious correlations and decide whether it can be trusted or not. The rationale behind the three tasks is the same: if human users make the expected decisions, it can be concluded that the explanations are in fact allowing the correct model interpretation.

A few evaluation challenges have also been realized to stimulate the addition of interpretability features in semantic applications. The third PASCAL Recognizing Textual Entailment Challenge (RTE-3) included an optional task requiring the participant system to justify their answer, that is, besides deciding whether a piece of text (the entailed *hypothesis*) could be entailed from another one (the entailing *text*) or not, they should also provide a post-hoc explanation justifying this decision (Voorhees, 2008). The explanations, which could be a collection of strings of any size, should be given in terms suitable for an end user (i.e., not a system developer), and were evaluated by human judges who assessed their understandability and correctness. Some common concerns and criticisms in the judges' evaluation summaries include verbatim repetition of the text and hypothesis in the justification, use of generic phrases such as "there is a relation" and "there is a match", presentation of system internals such as numerical similarity scores, and use of mathematical notation and linguistic

jargon such as "polarity" and "hyponym". These observations point out that conciseness and specificity are important features from the user point of view, and must have priority when explaining the system behavior.

Also in the Natural Language Processing field, an interpretable semantic text similarity (STS) task was proposed at SemEval (Agirre, Banea, et al., 2015). Participants were asked, in addition to rating the degree of semantic equivalence between two text snippets, to include an explanatory layer responsible for aligning the chunks in the sentence pair and annotating the kind of relation and the similarity score of the chunk pair. Rather than providing a natural language (post-hoc) explanation, systems should only justify the overall similarity score by pointing which parts in both pieces of text contributed to this score. Two scenarios were proposed: in the first one, participants were given gold standards chunks and should only make the correct alignments and provide them with appropriate labels and scores. In the second scenario, they were given raw text as input, and should also segment the input. The relevant relations defined for the task include *EQUI* (semantically equivalent), *OPPO* (oppositional meaning), *SPE1/SPE2* (chunk in sentence 1 is more specific than the one in sentence 2 and vice-versa), *SIM* (similar meaning, other than EQUI, OPPO, and SPE1/SPE2), and *REL* (related meaning, other than EQUI, OPPO, SPE1/SPE2, and SIM). Other relations refer to non-aligned chunks (*NOALI*), or context alignments (*ALIC*), that is, chunks that should be aligned to a chunk that was already aligned previously but can't due to a 1:1 alignment restriction. Since gold standards for chunk alignments, relations and similarity scores were available for all the text pairs, the evaluation was purely quantitative, measuring the F1 score obtained by each participant system. From the interpretability point of view, the output produced by the systems can't be assessed individually, since they followed the task requirements, providing only chunk alignments, with a relation from a predefined set and a similarity score associated with each alignment. Regarding the predefined relationships, SIM and REL relations seem particularly vague: the difference between them is not clear and both can refer to any semantic relation other than equivalence, opposition, and hypernymy, without ever making explicit what this relation is. The task may doubtless have served as a first exercise towards extracting further information for allowing a system interpretation, but turning this information into useful explanation for end users would require subsequent data refinement and formatting.

Table 2.4 summarizes the main characteristics of the aforementioned interpretability evaluation tasks.

Task	Type	Target Model	Element Evaluated	Dimension Evaluated	Human Evaluation
Word intrusion (Chang et al., 2009)	qualitative	TM, DSM	model	consistency	yes
Topic intrusion (Chang et al., 2009)	qualitative	TM	model	consistency	yes
Logical view equivalence (Mencar et al., 2011)	quantitative	FL	model	cointension	no
Rule interpretation (Lakkaraju et al., 2016)	qualitative	ML (DS)	model	understandability	yes
Decision set measuring (Lakkaraju et al., 2016)	quantitative	ML (DS)	model	size, length, cover, overlap	no
Feature recall assessment (Ribeiro et al., 2016)	quantitative	ML	explanation	faithfulness	no
Feature precision assessment (Ribeiro et al., 2016)	quantitative	ML	explanation	trustworthiness	no
Feature noise insertion (Ribeiro et al., 2016)	quantitative	ML	model	trustworthiness	no
Classifier selection (Ribeiro et al., 2016)	qualitative	ML	explanation	understandability	yes
Classifier improvement (Ribeiro et al., 2016)	qualitative	ML	explanation	understandability	yes
Feature identification (Ribeiro et al., 2016)	qualitative	ML	explanation	understandability	yes
RTE justification (Voorhees, 2008)	qualitative	n/a	explanation	understandability, correctness	yes
STS explanation (Agirre, Banea, et al., 2015)	quantitative	n/a	explanation	accuracy	no

Table 2.4: Interpretability evaluation tasks.

2.2.4 A Human-centered Take on Interpretability

Given the heterogeneity of the models tackling interpretability, the product offered as an interpretation and the point in the system's workflow at which this is accomplished also widely vary. Both aspects can affect the way the final user benefits from an interpretable model. It may not be enough for a model being interpretable; model interpretability should be incorporated seamlessly into the user's routine while using the system as a support tool.

To visualize the differences between approaches, consider the generic system architecture depicted in Figure 2.9, which sums up the characteristics of the various models described in Section 2.2.2. In this architecture, an input is sent to an algorithmic component, which will perform a sequence of computations over it and produce an output. Optionally, the algorithm can also employ the features from a data component, which can be an external model developed independently (such as a DSM) or an internal component tailored to the algorithm needs (the set of topics in a topic model, for instance). Besides the output itself, the algorithm may also provide an explanation for it. Alternatively, this explanation can be produced by an interpretability layer, a component that will take the output and induce the model behavior that generated it.

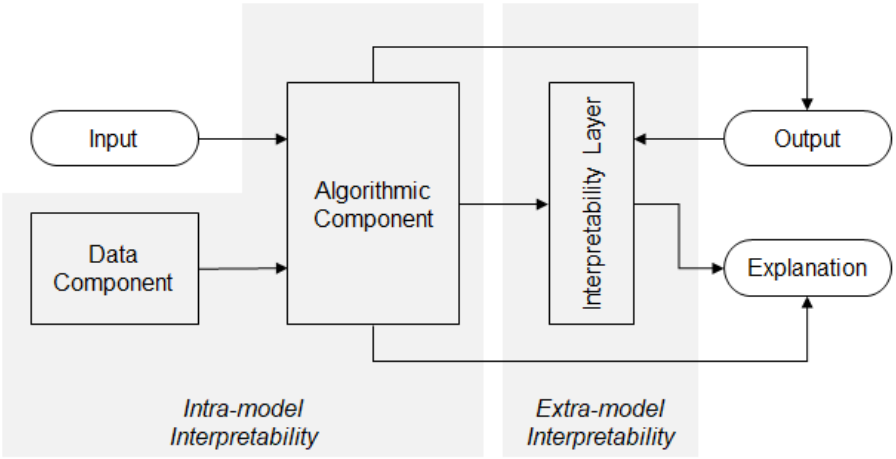

Figure 2.9: A generic AI system architecture and the points where interpretability features can be inserted.

Interpretability features at the data and algorithmic component levels can be considered *intra-model*, that is, the interpretability is embedded in the model

and influences the way it works in order to keep its behavior understandable from the user point of view. On the other hand, an interpretability layer can be seen as an *extra-model* component, since it usually is model-independent and tries to explain the algorithm behavior without modifying it or interfering in the way it works.

Intra-model interpretability naturally favors transparency, but can also work on the generation of post-hoc explanation if the algorithmic component is designed for providing complementary information to the predicted output. Extra-model interpretability is an effective tool for providing comprehensible post-hoc explanations, but at the expense of transparency, since it usually interprets the model locally (around the output being explained); interpreting the whole model is a challenging task, so, globally, it may remain a black box.

In practice, most models that implement intra-model interpretability will be contented with transparency, doing without any kind of post-hoc explanation. Figure 2.10 shows this trend, situating the models analyzed in Section 2.2.2 according to the points where they introduce interpretability features and the kind of interpretation they offer. As can be seen, the majority of the models offers only transparency through intra-model interpretability, while a few provide only post-hoc explanations making use of extra-model interpretability features. The sole exception is MGM (B. Kim et al., 2015), which provides post-hoc explanations while still claiming to be a transparent, interpretable model.

Taking into account the aforementioned human-centric nature of interpretability, this classification allows us to identify two important dimensions: the ease of use of interpretability features and the impact of their introduction from the point of view of the final user. First, it may be easy to understand the internal operations of a transparent model, but not all users will be willing to do so. A machine learning engineer seeking to tune a neural network will be happy to track the flow of computations of the model; a physician looking for the evidence for a diagnosis or a credit analyst who needs to justify a denied loan probably won't. Post-hoc explanations, either in textual or visual formats, tell how the model is working in a user-friendlier way and are better suited for non-developer users.

Second, new, interpretable versions of already existing models are important additions to the artificial intelligence body of knowledge, but, given the current widespread use of such technologies, ditching a deployed, fully functional system altogether for a new explainable one is not always an option. Despite the efforts in the direction of preserving the model performance while making it

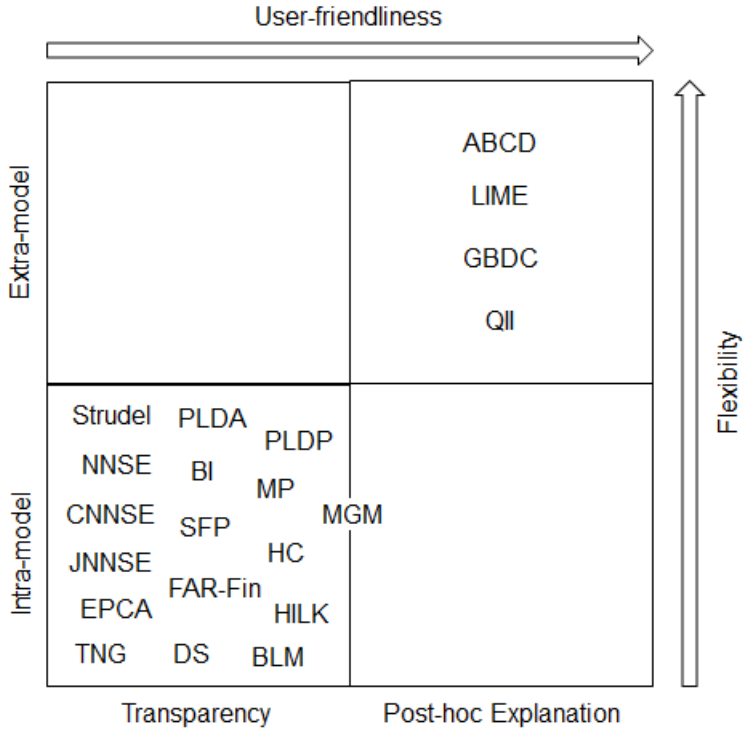

Figure 2.10: The four quadrants of AI models' interpretability.

interpretable, the trade-off between interpretability and accuracy is still an issue (Balázs & Kóczy, 2013). Interpretability layers can offer higher flexibility while evolving a system, since they can be attached to any model, and without affecting the way it is already working, ensuring that results won't change due to newly introduced interpretability functionalities. For large scale and/or critical systems, adding such layers represent a low-risk incremental upgrade, potentially meaning also a low impact on the final user routine.

Increasing model transparency is an important step towards widespread Explainable AI, but it should be only the first one if the user needs are put in the first place. A transparent model already offers a rich material for providing the user with explanations, requiring only a translation step to format the sequence of interpretable operations that led to an output as natural language justifications, image highlighting, graphs, diagrams or any other form of communication

suited to the task at hand, as long as it supports the user's work instead of adding further workload. Any model can benefit from an interpretability layer; even though fuzzy sets or decision lists are more comprehensible than deep neural networks, they still carry logic formalisms, which the user can be spared of. Given the wide variety of available artificial intelligence approaches and tasks to be addressed, generating post-hoc explanations, either at the intra- or extra-model level, is still an open challenge, and a path worth being explored even by the already interpretable (transparent) models.

2.2.5 Interpretability in Text Entailment Systems

As mentioned in Section 2.2.3, one of the PASCAL RTE Challenges proposed a task requiring participant text entailment systems to provide a justification for their answers. The literature review for the field (Section 2.1), though, allows us to conclude that such feature was an ad-hoc functionality for most systems, implemented in order to enable their participation in the task, as explainability, among all the challenges in the textual entailment recognition field, wasn't (and it is still not quite) a priority.

Considering the approaches analyzed in Section 2.1.3.2, it is possible to sketch their degree of interpretability according to the category they fit in. Although, as pointed out in Section 2.2.1, the fact that some models are intrinsically interpretable is debatable, alignment and transformation entailment approaches can be considered fairly transparent, as they don't use a large number of parameters or perform thousands of computations to reach a decision. For alignment systems, it might be easy to follow the rationale by knowing what portions of T and H are being aligned, the confidence scores assigned to each alignment, and how these scores are combined so the final decision is reached. Similarly, in transformation systems, it might also be simple to keep track of the sequence of rules applied in the transformation.

Besides the likely transparency, transformation approaches also have the potential to provide post-hoc explanations. In a positive entailment, when T can successfully be transformed into T, the sequence of logical axioms or rules is the proof that justifies the decision. With some additional effort, especially for logic-based systems, since rule-based approaches employ rules which are already closer to natural language expressions, such proofs could be formatted into human-readable explanations, providing user-oriented justifications. Nevertheless, the

BLUE system (Clark & Harrison, 2009) is the only one to implement such feature.

For classification systems, transparency cannot be considered as a given. With multiple representation schemata, similarity measures, and weighting methods, all being combined by increasingly complex algorithms, it becomes harder to follow every step in the reasoning procedure. Some models only use machine learning for computing weights, but others use it as the classification model itself, in which case the multiple-features vector representations and the complex computations performed over them turn the system into a black box. None of the analyzed classification systems provide post-hoc explanations either.

Table 2.5 summarizes the interpretability characteristics of the entailment approaches analyzed in Section 2.1.3.2. Note that *transparency* is presumed, following the above-mentioned arguments. For completeness, NLI models described in Section 2.1.5 are also included. Nonetheless, their characteristics are rather uniform: as deep neural network models, they are not transparent, and none of them provide post-hoc explanations. Regarding explanations, the newly released e-SNLI dataset (Camburu, Rocktäschel, Lukasiewicz, & Blunsom, 2018), an extension of SNLI with human-annotated natural language explanations for each premise-hypothesis pair, can leverage the developments in this area. So far, a single approach (Thorne, Vlachos, Christodoulopoulos, & Mittal, 2019) has used this dataset, and only for token-level explanation, that is, only the tokens in the premise and in the hypothesis that are relevant for the inference are presented (which is basically the output of the attention mechanism). Fully human-readable explanations made up of full concise and connected natural language sentences are yet to be addressed in NLI.

2.3 Gap Analysis

Text entailment recognition is a very complex task, requiring not only the development and integration of different modules that address different entailment phenomena but also the acquisition of commonsense world knowledge that can support those modules' reasoning process. As observed by Dagan et al. (2013), a complete text entailment model would require solving many types of NLP and Computational Linguistics problems, as well as additional inference problems investigated in Artificial Intelligence.

Approach	Task	Transparency	Explanation
Glickman & Dagan (2005)	TE	Yes	No
MacCartney et al. (2008)	TE	Yes	No
Iftene & Balahur-Dobrescu (2007)	TE	Yes	No
Sammons et al. (2009)	TE	Yes	No
Hickl & Bensley (2007)	TE	Yes	No
Dagan & Glickman (2004)	TE	Yes	No
Harmeling (2009)	TE	Yes	No
Bar-Haim et al. (2007)	TE	Yes	No
Stern & Dagan (2011)	TE	Yes	No
Braz et al. (2005)	TE	Yes	No
Raina et al. (2005)	TE	Yes	No
Fowler et al. (2005)	TE	Yes	No
Bos & Markert (2005)	TE	Yes	No
Clark & Harrison (2009)	TE	Yes	Yes
Kouylekov & Magnini (2005)	TE	Yes	No
Mehdad et al. (2009)	TE	Yes	No
Heilman & Smith (2010)	TE	Yes	No
MacCartney & Manning (2007)	TE	Yes	No
Tatu et al. (2006)	TE	No	No
Wang & Neumann (2008a)	TE	No	No
Jimenez et al. (2014)	TE	No	No
Zhao et al. (2014)	TE	No	No
Zhang et al. (2017)	TE	No	No
Bowman et al. (2015)	NLI	No	No
Rocktäschel et al. (2016)	NLI	No	No
Wang & Jiang (2016)	NLI	No	No
Chen et al. (2017)	NLI	No	No
Parikh et al. (2016)	NLI	No	No
Chen et al. (2018)	NLI	No	No
Wang et al. (2019)	NLI	No	No

Table 2.5: Interpretability features of the analyzed text entailment and NLI approaches.

Nevertheless, text entailment datasets, including those created for the RTE Challenges, are composed of mixed phenomena and were used to evaluate the quality of complete entailment architectures rather than for individual inference modules, that is, modules focused on a specific phenomenon (Dagan et al., 2013). Specialized datasets to evaluate individual phenomena are crucial assets in the development of focused inference components, but this kind of resource is still scarce. Moreover, new datasets now focus on size, since being large enough to

enable machine learning model training is one of the main requirements, and breaking the data by specific subtypes haven't been an option.

An alternative to matching specific modules to the kind of data they are intended to in the absence of specialized datasets is to make the system able to analyze each entailment pair, identify the most relevant phenomenon, and then use the suitable module to solve it. As evidenced by the review presented in Section 2.1, no text entailment approach have tried this strategy so far, with all the systems instead opting for either using a single method or using as many methods as possible, but combining all of them in a single technique that tries to find the better weights for each of them so the most relevant one for a given entailment pair can have more impact in the final decision. In a selective approach, previously identifying the phenomenon and using only the relevant method can not only simplify the reasoning process but also avoid that similarity measures that are not relevant (for example, using syntactic similarity measures when T and H have very different syntactic structures but are still semantically related) have a negative impact on the final decision.

Another gap that could be apprehended from the review refers to the commonsense world knowledge acquisition methods employed by the entailment approaches. Although some systems did consider more complex semantic relationships by, for example, trying to retrieve such knowledge from structured content in Wikipedia, most approaches stop at the shallow semantic level. Entailment rules, which were already somewhat limited, since they mostly expressed paraphrase relations, were gradually being left aside as approaches moved from alignment and transformation techniques to classification strategies. Most NLI models follow this trend, heavily focusing on accuracy and giving almost no attention to external knowledge.

Finding the suitable knowledge source for the task, one that at the same time is reasonably expressive and can be easily integrated into the system, is one of the challenges in knowledge acquisition, especially because of the variety of knowledge types to be covered (Section 2.1.1). In their investigation about the most common knowledge types needed in text entailment recognition, LoBue and Yates (2011) point out that "common knowledge types, like definitional knowledge, arithmetic, and accountability have for the most part been ignored by research on automated knowledge collection". Among these types, definitional knowledge is the one whose extraction could bring more benefits given the varied number of relationships it can cover, as explained in Section 2.1.1. Dagan et al. (2013) reinforce this idea, observing that "The first and most natu-

ral candidate for providing such [entailment] knowledge is dictionary definitions. These definitions have long been identified as a valuable source for semantic relations between words, as they describe words in terms of other words". Despite that, there are no entailment approaches making use of definitional knowledge, or initiatives to extract and represent definitions in a way that suits the text entailment task.

Regarding interpretability features, Table 2.5 shows that the gap in the text entailment field is clear. As argued in Section 2.2.4, providing post-hoc explanations are the most user-friendly way to render a system interpretable, and, in NLP tasks, natural language justifications can be seen as one of the most suitable presentation formats. Generating such justifications is a task for which there is still much room for development.

Therefore, the text entailment approach proposed in this thesis seeks to address these gaps by: (1) implementing different modules to address different phenomena and being able to identify the relevant phenomenon so the entailment pair can be sent to the suitable module; (2) developing a methodology for extracting and representing definitional knowledge, converting dictionaries into knowledge graphs that will be used for detecting semantic relationships between sentences; and (3) making the system interpretable by generating natural language justifications for the entailment decision, using the knowledge from the definition graphs. These contributions represent additions not only for the text entailment field but also for the knowledge representation and semantic interpretability areas as well.

2.4 Summary

In this chapter, the literature referring to the two main research areas addressed in this work – Text Entailment Recognition and Semantic Interpretability – was reviewed. In the text entailment part, the main phenomena involved in entailment recognition were described and categorized. The knowledge requirements for the task were detailed, and a macro-categorization was also proposed, along with an overview of the main knowledge sources from which different types of knowledge can be acquired to be used in the entailment task.

The most used base methods for entailment recognition were described, and different approaches, which combine base methods for matching T and H through alignment, transformation, or classification were reviewed. It was

shown that, although some of them do use external knowledge sources for identifying more complex semantic relations, most approaches focus on exploring the syntactic structure of the sentences and retrieving shallow semantic information, mainly in the form of structured lexical links, semantic role labeling, or paraphrase-style rules. The RTE Challenges, the main evaluation initiative in the area, were described, and a brief review introduced the Natural Language Inference task, a subtask of textual entailment which is gaining increasing importance in the NLP community.

Regarding the semantic interpretability aspect, it was shown that, after successful efforts for making AI models highly accurate, researchers are now being faced with the challenge of making them also interpretable, untangling their complexity to make the rationale behind any prediction clear and intelligible. By understanding the model behavior and being able to explain its decisions, users can not only justify the decisions they make based on it but also identify when it is making spurious correlations or reflecting social biases contained in data to avoid unfair decisions.

Semantic interpretability was reviewed as a cross-field subject, making it possible to go beyond the machine learning perspective and bring to light the efforts of other areas, such as distributional semantics and fuzzy logic, towards increasing model interpretability. By examining how distinct disciplines define and offer interpretability, we outlined the shapes it can assume and, based on them, analyzed several different types of models and interpretability evaluation methods. We further categorized the models according to how they integrate interpretability features into their architectures and assessed how this, along with the type of interpretation offered, impacts the final user routine. Textual entailment approaches were also analyzed with regard to their interpretability features, and the characterization provided in the first part of this literature review was complemented with attributes that reflect each approach's transparency and explainability dimensions.

Finally, a gap analysis highlighted some of the most important points identified in this review that are yet to be improved in the text entailment field, and which are addressed in the approach proposed in this thesis.

Chapter 3

From Lexicons to Knowledge Graphs

In this chapter, the foundations of the knowledge bases (KBs) employed by the text entailment recognition approach proposed in this work are introduced. Such knowledge bases constitute one of the main pillars of the proposed approach, enabling the recognition of text entailments requiring world knowledge and allowing the generation of human-readable explanations, which distinguishes this work as an interpretable entailment system.

The need for external, commonsense knowledge comes from the nature of the text entailment task itself: as detailed in Chapter 1, one of the three possible scenarios is when the hypothesis presents new information derived from the text, requiring knowledge that goes beyond what is expressed in T and H so the entailment can be solved. But the importance of injecting external knowledge in the reasoning process is not restricted to the entailment decisions, that is, the generation of a *yes* or *no* answer. The ability to explain how such decisions are reached, be they entailment decisions or any other intelligent application's prediction, is becoming a key requirement for AI systems (Gunning, 2017). Although a model may produce accurate results, if it lacks transparency, not showing clearly how it is using the data, it can become harder for users to trust its predictions.

Generating natural language justifications is an important feature for increasing a system's interpretability, and the generation of such explanations can be leveraged by the use of external sources of world knowledge. Dictionary-style

definitions are a rich source of such knowledge and, different from formal structured resources like ontologies, they are domain-independent and largely available. Many NLP systems, including text entailment systems (Clark, Fellbaum, & Hobbs, 2008; Herrera, Penas, & Verdejo, 2006), already explore lexicons, notably WordNet (Fellbaum, 1998), but they usually look only at the structured information, that is, links such as synonyms, hypernyms, etc. The natural language definition is left aside, although it contains the largest amount of relevant information about an entity: its type, essential attributes, primary functions, and often many non-essential, but very informative, attributes as well.

The text entailment recognition and justification approach proposed in this work relies on the knowledge provided by lexical dictionary definitions for looking for and explaining the semantic relationships holding between the text and the hypothesis. If such relationships exist and can be found in the knowledge base, they not only confirm the entailment but also explain why the entailment is true. In order to make natural language definitions useful in the entailment reasoning process, dictionaries were structured into a graph knowledge base, here called a *Definition Knowledge Graph* (DKG). A conceptual model for representing the DKG, expressing a lexical definition's main components and the relationships among them, is proposed, and a filtering procedure was developed so invalid definitions, that is, definitions that don't fit the essential structural aspects expressed by this model, could be removed. A graph construction methodology was then developed to populate the conceptual model with the set of filtered definitions, automatically converting a whole (filtered) dictionary into a structured graph knowledge base.

Next, an overview of commonsense knowledge graphs and the applications of this type of resource in NLP tasks is given. Then, the conceptual model for representing dictionary definitions as a knowledge graph is detailed, followed by an account of the definition syntactic filtering procedure, and the description of the graph construction methodology which converts dictionaries into concrete knowledge bases ready to be explored by a world knowledge-driven reasoning model.

3.1 Commonsense Knowledge Graphs

The main goal of Natural Language Processing is to provide computers with the ability to understand and manipulate text expressed in natural language

for performing a given task, by emulating the way human beings understand and use language (Chowdhury, 2003). But humans do not grasp the meaning of sentences based only on what is explicitly written or said. Rather, they use their previously acquired and accumulated knowledge about the world to draw relationships between entities and to infer new facts based on the information at hand. That means that, for properly understanding and making use of natural language, especially in tasks involving inference, computers should be able to link the explicit information contained in text to the implicit facts that can be logically derived from it (Nilsson, 2014; Cambria & White, 2014).

But how can computers acquire the world background knowledge which humans take for granted in their everyday language-mediated interactions? Availability is fortunately not an issue: with billions of websites, the Web gathers a vast amount of knowledge, ranging from very specific domain-related information to the most generic commonsense facts. But in order to make this knowledge useful for NLP applications, it is necessary to extract and make explicit the relationships between objects. Knowledge graphs support the representation of entities and the associations between them in a lightweight manner, allowing the retrieval of simple relations expressed by triples, that is, two nodes linked by an edge, or more complex connections made up by a composition of relations, represented by longer paths between two graph nodes.

The most popular large-scale commonsense knowledge graphs are indeed built out of content extracted from the Web, like DBpedia (Lehmann et al., 2015), YAGO (Suchanek, Kasneci, & Weikum, 2007), KOG (F. Wu & Weld, 2008), and Probase (W. Wu, Li, Wang, & Zhu, 2012). Other important knowledge bases, such as ConceptNet (H. Liu & Singh, 2004) and Freebase (Bollacker, Evans, Paritosh, Sturge, & Taylor, 2008), are populated with content created from scratch in a collaborative manner.

DBpedia extracts structured information from Wikipedia pages, such as infoboxes, category labels, and geographic coordinates. The extraction framework relies mainly on mapping rules to associate elements in an infobox or other features found in a Wikipedia article to terms in the DBpedia ontology. It is a multilingual resource whose largest knowledge base derives from the English version of Wikipedia, with 400 million facts describing 3.7 million entities, which are represented as RDF statements (triples) and resources, respectively. Although Wikipedia does contain articles describing common language concepts, that is, concepts denoted by common nouns, more detailed and informative infoboxes tend to be available only for named entities. In fact, the most popular classes,

which covers the largest number of instances, include "Person", "Place", "Organization", and "Work" ("Musical Work", "Film", "Software", etc.) (Lehmann et al., 2015). DBpedia is used in a large number of NLP applications, especially for named entity annotation and disambiguation (García-Silva, Szomszor, Alani, & Corcho, 2009; Kobilarov et al., 2009; Mendes, Jakob, García-Silva, & Bizer, 2011; Hulpus, Hayes, Karnstedt, & Greene, 2013) and question answering (Unger et al., 2012; Lopez, Fernández, Motta, & Stieler, 2012; Damljanovic, Agatonovic, & Cunningham, 2011; Cabrio et al., 2012), among which the most remarkable is the *IBM Watson* system (Ferrucci et al., 2010). Many datasets, such as DrugBank (Wishart et al., 2017), LinkedGeoData (Stadler, Lehmann, Höffner, & Auer, 2012), the CIA World Factbook (Central Intelligence Agency, 2009), and Book Mashup (Bizer, Cyganiak, & Gauß, 2007) among others, also link to DBpedia for uniquely identifying their entities.

YAGO (*Yet Another Great Ontology*) is another knowledge graph built from Web content which combines information from Wikipedia and WordNet. It relies on the WordNet hierarchy of concepts, given by the *hypernym* and *hyponym* links to draw *Is-A* relationships between individuals. The set of individuals is composed by the union of WordNet nouns and the subjects of Wikipedia articles. Rather than using the infoboxes for extracting attributes as DBpedia does, YAGO uses Wikipedia categories for deriving relationships between entities, using a combination of rule-based and heuristic methods for extracting both entities and relationships from a given category label associated with an article. It is based on an extension of the RDFS data model (Allemang & Hendler, 2011), called *YAGO model*, which can express relations between facts and other relations. A fact is a triple composed of two entities linked by a relation, similar to an RDF statement. YAGO contains around 1 million entities and 5 million facts, and is mainly used for entity linking and disambiguation (Shen, Wang, Luo, & Wang, 2012; Usbeck et al., 2014; Hoffart et al., 2011; Shen, Wang, Luo, & Wang, 2013).

KOG (*Kylin Ontology Generator*, after the autonomous semantic markup system *Kylin* (F. Wu & Weld, 2007)) is yet another Web content-based knowledge graph, which combines the DBpedia and YAGO extraction strategies: it uses information from Wikipedia and WordNet like YAGO, but, for drawing relationships between the entity described by an article and other entities, it extracts such entities from infoboxes, like DBpedia. It then maps the extracted entities to WordNet concepts. KOG aims at refining the infoboxes underlying ontology for supporting advanced queries over data extracted from Wikipedia.

Probase (W. Wu et al., 2012) is a knowledge base that goes beyond the Wikipedia scope, with the ambitious goal of harvesting knowledge from the whole Web, but at the cost of reduced relationship expressiveness. In fact, Probase is only a taxonomy, that is, it covers only *isA* relationships between 2.7 million concepts extracted from 1.68 billion web pages. Its extraction framework combines syntactic and semantic features in an iterative learning process to identify concepts and their sub- or super-concepts in text excerpts, and uses probabilities to model inconsistent, ambiguous and uncertain information. The result is a direct acyclic graph (DAG), where a node can be either a concept, or an instance of a concept, and both types of nodes are distinguished by the fact that instance nodes don't have children (and, therefore, no outgoing edges), that is, they are the leaf nodes in the graph. One of the main applications of Probase is enabling semantic web search through query expansion (Hua, Song, Wang, & Zhou, 2013; Z. Wang, Zhao, Wang, Meng, & Wen, 2015) and better interpretation of HTML content (J. Wang, Wang, Wang, & Zhu, 2012; C. Wang et al., 2015).

ConceptNet (H. Liu & Singh, 2004) puts aside existing web content and relies entirely on data generated collaboratively by lay users. It is part of the Open Mind Common Sense (OMCS) project, which collects general knowledge statements from over 14,000 volunteers and automatically converts them into a semantic network. Users can either enter free-form data or follow simple semi-structured frames with suggested relations, such as "__ can be used to __", where they fill in the blanks with concepts that may later compose a triple, linked by the *usedTo* relationship, for example. Free-form statements are also usually simple, like "An apple is a fruit" or "A city is part of a country", for example, making the rule-based information extraction quite straightforward. The initial OMCS corpus contained over 700,000 statements, which generated around 300,000 nodes and 1.6 million assertions. ConceptNet has a predefined, but open, set of relationships, which includes *IsA*, *PartOf*, *LocationOf*, *UsedFor*, *InstanceOf*, and *CreatedBy*, among others. It is used in different types of tasks, such as query expansion (Bouchoucha, He, & Nie, 2013; Hsu, Tsai, & Chen, 2008; Kotov & Zhai, 2012), sentiment analysis (Cambria, Havasi, & Hussain, 2012; Tsai, Wu, Tsai, & Hsu, 2013; Agarwal, Mittal, Bansal, & Garg, 2015), and question answering (Boteanu & Chernova, 2015; P. Wang, Wu, Shen, Dick, & van den Hengel, 2018).

Freebase (Bollacker et al., 2008) was an attempt to integrate both KB population strategies, combining knowledge extracted from the Web with content

created collaboratively. It also harvested data from Wikipedia, among other sources, but also offered a system for allowing users to collaborate on the data creation, structuring and maintenance in a wiki-style web interface. Freebase main goal was to offer a knowledge base with the structural diversity provided by collaborative wikis while still allowing the scalability and query capabilities of a traditional structured database. As a tuple store, Freebase had more than 2.4 billion facts about around 44 million topics, being used in many question answering systems (Yih, Chang, He, & Gao, 2015; Berant, Chou, Frostig, & Liang, 2013; Dong, Wei, Zhou, & Xu, 2015; Yao & Van Durme, 2014). It was later used as a source for, and replaced by, the Google Knowledge Graph, now perhaps the largest commonsense knowledge graph, but which is not freely available, being used only by Google and its services for enhancing search results, especially about named entities.

What can be noticed when comparing the different knowledge extraction methods employed in the creation of the above-mentioned knowledge graphs is that there is a trade-off between diversity of sources and variety of relationships. While strategies that explore Wikipedia infoboxes and categories, like DBpedia, YAGO, and KOG, can generate graphs with a larger number of different relationships, they can only do so from structured or semi-structured content. Moreover, by targeting Wikipedia content, these graphs predominantly contain knowledge about named entities and little information about the world's most ordinary entities. On the other hand, methods capable of harvesting knowledge from any web page and processing natural language text to extract entities and relationships among them have to do so by committing to a smaller set of predefined relationships, or even ending up with only a taxonomy, like Probase, which contains a larger variety of entities, encompassing both named entities and common world objects, but only a single relationship linking them.

Collaborative knowledge acquisition is a way to overcome expressiveness limitations, since, through the input of thousands of contributors, it is possible to gather a wide range of both entity types and relationships. Nevertheless, resorting to lay users for generating formalized knowledge has as a side effect the decrease in content quality. This becomes clear when we analyze the relationships for common concepts in ConceptNet: users often misunderstand the meaning of relationships and end up creating inconsistent triples. The most remarkable example is the very common confusion between the *IsA* and *InstanceOf* relationships. For example, for the concept "city", the list of other concepts linked to it through the *IsA* relationship mixes actual subclasses such as "capital" and

"town" with instances like "San Francisco", "London", and "Venice". Another example is the relationship *LocationOf*: the list of things located at "Toronto" includes "garbage", "road", and "tower", when such common concepts should be linked to other common concepts (in this case, to "city" so as to express the same idea of location), while links to things located in "Toronto", which is an instance and not a class, should be reserved to other instances, like "Canadian National Tower" or "St. James Anglican Cathedral". Such principles are part of the basic rules of formal knowledge representation and known to any data modeling expert, but usually out of reach for the general population which collaborative efforts rely on, making modeling and representation mistakes a recurrent issue in collaborative environments.

Considering the text entailment task in particular, since it deals with language variability and in most cases involves common language concepts, knowledge graphs focused on named entities can't provide the necessary information for the reasoning process. However, besides word coverage, content quality is also an important attribute for a commonsense knowledge base. The choice for using dictionaries in the scope of this work owes to the fact that they are a rich source of commonsense knowledge, gathering essential information about a wide range of basic language concepts. Nevertheless, no knowledge base dedicated to representing natural language definitions in a structured form with the explicit intent of providing world knowledge for inference tasks had been developed so far. We believe that such resources can provide the knowledge necessary for the text entailment task, helping to meet both reasoning and interpretability requirements.

3.2 DKG Conceptual Model

Commonsense world knowledge is a fundamental resource not only for Text Entailment, but for a number of other Natural Language Processing tasks, such as Question Answering, Information Retrieval, and Machine Translation, to name a few. Nevertheless, existing large-scale knowledge bases, as described in Section 3.1, have coverage and quality limitations, and formal, structured resources that can express more complex relationships between a large number of entities in a reliable way, such as ontologies, are still scarce and usually target a very specific domain. On the other hand, a large number of linguistic resources

gathering dictionary definitions is available not only for particular domains but also addressing wide-coverage commonsense knowledge.

However, understanding the syntactic and semantic "shape" of natural language definitions, i.e., how definitions are usually expressed, is fundamental for the extraction of structured representations and the construction of semantic models from these data sources. In order to make the most of those resources, it is necessary to capture the underlying structure of natural language definitions so they can be represented in a structured way that favors both the information extraction process and the subsequent information retrieval. This allows the effective construction of semantic models from these data sources while keeping the resulting model easily searchable and interpretable. Furthermore, by using these models, the approach for text entailment proposed in this work can increase its own interpretability, benefiting from the structured data for performing traceable reasoning and generating explanations.

Through a systematic analysis of the syntactic and semantic structure of natural language definitions, the predominant definition patterns were identified. Based on these patterns, a set of *semantic roles* for definitions is proposed. Differently from the commonly used event-centered semantic roles, which define the semantic relations holding among a predicate (the main verb in a clause) and its associated participants and properties (Màrquez, Carreras, Litkowski, & Stevenson, 2008), in the context of this work, semantic role means *entity-centered* roles, that is, roles that express the part played by an expression in a definition, showing how it relates to the *definiendum*, that is, the entity being defined.

Although the predicate-oriented semantic role labeling (SRL), which focuses on determining "who" did "what" to "whom", "where", "when", and "how" (Jurafsky & Martin, 2000), is the most widespread SRL task, other sets of semantic labels targeting different relationships have indeed already been proposed, such as the non-event-centered semantic role labeling task focusing on spatial relations between objects (Kordjamshidi, Moens, & van Otterlo, 2010). This task defines roles such as *trajectory*, *landmark*, *region*, *path*, *motion*, *direction* and *frame of reference*, and an approach for annotating sentences containing spatial descriptions, extracting topological, directional and distance relations from their content was developed as well, showing the potential of SRL for other areas beyond predicate analysis.

WordNet (Fellbaum, 1998), one of the most employed linguistic resources in semantic applications, was used as a corpus for the present study. The analysis'

results pointed out the syntactic and semantic regularity of definitions, making explicit an enumerable set of syntactic and semantic patterns which was used to derive the information extraction framework and the underlying DKG semantic model.

3.2.1 Structural Aspects of Definitions

Swartz (1997) describes lexical, or dictionary definitions as reports of common usage (or usages) of a term. He argues that they allow the improvement and refinement of the use of language, because they can be used to increase vocabulary (introducing people to the meaning and use of words new to them), to eliminate certain kinds of ambiguity and to reduce vagueness. A clear and properly structured definition can also provide the necessary identity criteria to correctly allocate an entity in an ontologically well-defined taxonomy (Guarino & Welty, 2002).

Some linguistic resources, such as WordNet, organize concepts in a taxonomy, so the *genus-differentia* definition pattern would be a suitable way to represent the subsumption relationship among them. The genus and differentia concepts date back to Aristotle's writings concerning the theory of definition (Berg, 1982; Granger, 1984; A. C. Lloyd, 1962) and are most commonly used to describe entities in the biology domain, but they are general enough to define concepts in any field of knowledge. An example of a genus-differentia based definition is the Aristotelian definition of a human: "a human is a rational animal". "Animal" is the genus, and "rational" is the differentia distinguishing humans from other animals.

Another important aspect of the theory of definition is the distinction between *essential* and *non-essential* properties. As pointed by Burek (2004), stating that "a human is an animal" informs an essential property for a human (being an animal), but the sentence "human is civilized" does not communicate a fundamental property, but rather something that happens to be true for humans, that is, an incidental property.

Analyzing a subset of the WordNet definitions to investigate their structure, it could be noticed that most of them loosely adhere to the classical theory of definition: with the exception of some samples of what could be called ill-formed definitions, in general, they are composed by a linguistic structure that resembles the genus-differentia pattern, plus optional and variable incidental properties. Such recurring structures were classified and organized into patterns, deriving a

set of semantic roles representing the components of a lexical definition, which are described next.

3.2.2 Semantic Roles for Lexical Definitions

Definitions in WordNet don't follow a strict pattern: they can be constructed in terms of the entity's immediate superclass or rather using a more abstract ancestral class. For this reason, we opted for using the more general term **supertype** instead of the classical *genus*. A supertype is either the immediate entity's superclass, as in "footwear: *clothing* worn on a person's feet", being "footwear" immediately under "clothing" in the taxonomy; or an ancestral, as in "illiterate: a *person* unable to read", where "illiterate" is three levels below "person" in the hierarchy (according to WN hypernym links, "illiterate" is an "uneducated person", an " uneducated person" is an "unskilled person", and an "unskilled person" is a "person").

Two different types of distinguishing features stood out in the analyzed definitions, so the differentia component was split into two roles: **differentia quality** and **differentia event**. A differentia quality is an essential, inherent property that distinguishes the entity from the others under the same supertype, as in "baseball_coach: a coach *of baseball players*". A differentia event is an action, state or process in which the entity participates and that is mandatory to distinguish it from the others under the same supertype. It is also essential and is more common for (but not restricted to) entities denoting roles, as in "roadhog: a driver *who obstructs others*".

As any expression describing events, a differentia event can have several subcomponents, denoting time, location, mode, etc. Although many roles could be derived, we opted to specify only the ones that were more recurrent and seemed to be more relevant for the definitions' classification: **event time** and **event location**. Event time is the time in which a differentia event happens, as in "master_of_ceremonies: a person who acts as host *at formal occasions*"; and event location is the location of a differentia event, as in "frontiersman: a man who lives *on the frontier*".

A **quality modifier** can also be considered a subcomponent of a differentia quality: it is a degree, frequency or manner modifier that constrains a differentia quality, as in "dart: run or move *very* quickly or hastily", where "very" narrows down the differentia quality "quickly" associated to the supertypes "run" and "move".

The **origin location** role can be seen as a particular type of differentia quality that determines the entity's location of origin, but in most of the cases it doesn't seem to be an essential property, that is, the entity only happens to occur or come from a given location, and this fact doesn't account to its essence, as in "Bartramian_sandpiper: large plover-like sandpiper *of North American fields and uplands*", where "large" and "plover-like" are essential properties to distinguish "Bartramian_sandpiper" from other sandpipers, but occurring in "North American fields and uplands" is only an incidental property.

The **purpose** role determines the main goal of the entity's existence or occurrence, as in "redundancy: repetition of messages *to reduce the probability of errors in transmission*". A purpose is different from a differentia event in the sense that it is not essential: in the mentioned example, a repetition of messages that fails to reduce the probability of errors in transmission is still a redundancy, but in "water_faucet: a faucet *for drawing water* from a pipe or cask", "for drawing water" is a differentia event, because a faucet that fails this condition is not a water faucet.

Another event that is also non-essential, but rather brings only additional information to the definition is the **associated fact**, a fact whose occurrence is/was linked to the entity's existence or occurrence, accounting as an incidental attribute, as in "Mohorovicic: Yugoslav geophysicist *for whom the Mohorovicic discontinuity was named*".

Other minor, non-essential roles identified in our analysis are:

Accessory determiner: a determiner expression that doesn't constrain the supertype-differentia scope, as in "camas: *any of several* plants of the genus Camassia", where the expression "any of several" could be removed without any loss in the definition meaning;

Accessory quality: a quality that is not essential to characterize the entity, as in "Allium: *large* genus of perennial and biennial pungent bulbous plants", where "large" is only an incidental property; and

[**Role**] **particle:** a particle, such as a phrasal verb complement, non-contiguous to the other role components, as in "unstaple: take the staples *off*", where the verb "take off" is split in the definition, being "take" the supertype and "off" a supertype particle.

The conceptual model in Figure 3.1 shows the relationship among roles, and between roles and the *definiendum*. Table 3.1 summarizes the identified semantic roles' descriptions. The proposed semantic roles list is by no means definitive or exhaustive, but it covers a reasonable amount of definition properties, rendering enough granularity and expressiveness to the resulting knowledge base.

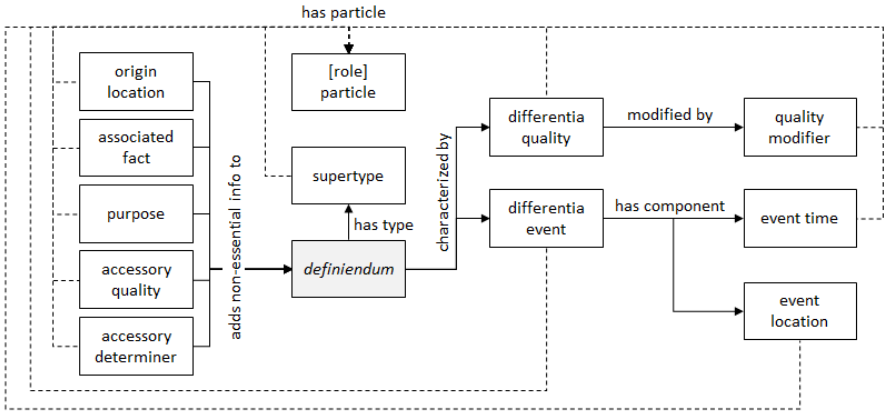

Figure 3.1: Conceptual model for the semantic roles for lexical definitions. Relationships between *[role] particle* and every other role in the model are expressed as dashed lines for readability.

3.2.3 Identifying Semantic Roles in Definitions

Once the relevant semantic roles were identified in the manual analysis, the following question emerged: is it possible to extend this classification to a whole definition database through automated Semantic Role Labeling? Although most SRL systems rely on efficient machine learning techniques (Palmer, Gildea, & Xue, 2010), an initial, preferably large, amount of annotated data is necessary for the training phase.

Since manual annotation is expensive, an alternative is a rule-based mechanism to automatically label the definitions, based on their syntactic structure, followed by manual curation of the generated data. As shown in an experimental study by Punyakanok et al. (2005), syntactic parsing provides fundamental information for event-centered SRL, and, in fact, this is also true for entity-centered SRL.

To draw the relationship between syntactic and semantic structure (as well as defining the set of relevant roles described earlier), a random sample of 100

Role	Description
Supertype	the immediate or ancestral entity's superclass
Differentia quality	a quality that distinguishes the entity from the others under the same supertype
Differentia event	an event (action, state or process) in which the entity participates and that is mandatory to distinguish it from the others under the same supertype
Event location	the location of a differentia event
Event time	the time in which a differentia event happens
Origin location	the entity's location of origin
Quality modifier	degree, frequency or manner modifiers that constrain a differentia quality
Purpose	the main goal of the entity's existence or occurrence
Associated fact	a fact whose occurrence is/was linked to the entity's existence or occurrence
Accessory determiner	a determiner expression that doesn't constrain the supertype-differentia scope
Accessory quality	a quality that is not essential to characterize the entity
[*Role*] particle	a particle, such as a phrasal verb complement, non-contiguous to the other role components

Table 3.1: Semantic roles for dictionary definitions

definitions from the WordNet nouns+verbs database[1] was selected, being 84 nouns and 16 verbs (the verb database size is only approximately 17% of the noun database size).

First, each of the definitions was manually annotated, so each segment in the sentence was assigned the most suitable role. Example sentences and parentheses were not included in the classification. Figure 3.2 shows some examples of annotated definitions. Then, using the Stanford parser (Manning et al., 2014), we generated the syntactic parse trees for all the 100 definitions and compared the semantic patterns with their syntactic counterparts, pairing role labels with phrasal nodes for each relevant segment in the sentences.

Table 3.2 shows the distribution of the semantic patterns for the analyzed sample. As can be seen, *(supertype) (differentia quality)* and *(supertype) (differentia event)* are the most frequent patterns, but many others are composed by a combination of three or more roles, usually the supertype, one or more differentia qualities and/or differentia events, and any of the other roles. Since

[1] Adjectives and adverbs are not organized in a taxonomy in WordNet, so are less likely to follow a supertype-differentia pattern, probably requiring a different classification strategy

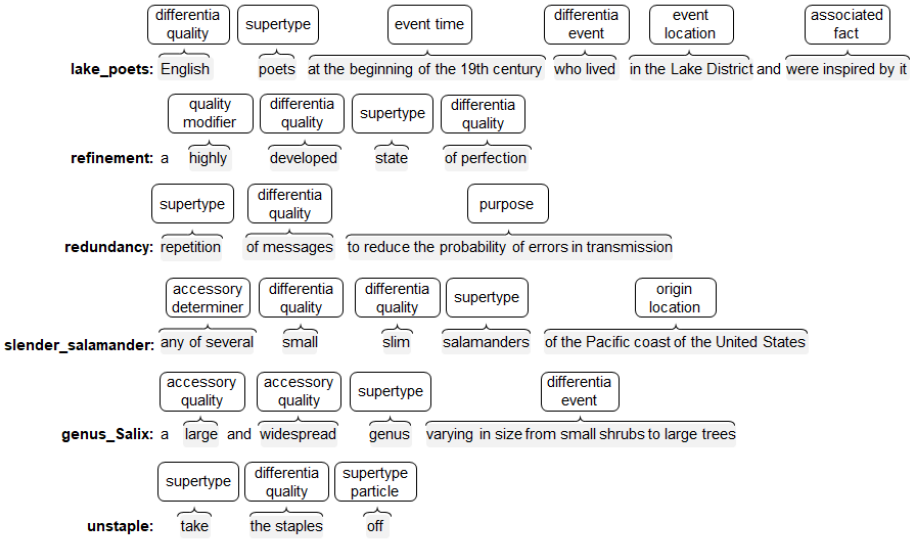

Figure 3.2: Examples of definition role labeling.

most of them occurred only once (29 out of 42 identified patterns), it is easier to analyze the roles as independent components, regardless of the pattern where they appear in. The context can always give some hint about what a role is, but we would expect the role's main characteristics not to change when their "companions" in the sentence varies. The conclusions are as follows[2], and are summarized in Table 3.3:

Supertype: it's mandatory in a well-formed definition, and indeed 99 out of the 100 sentences analyzed have a supertype (the definition for "Tertiary_period" – "from 63 million to 2 million years ago" lacks a supertype and could, then, be considered an ill-formed definition). For verbs, it is the leftmost VB and, in some cases, subsequent VBs preceded by a CC ("or" or "and"). This is the case whenever the parser correctly classifies the definition's head word as a verb (11 out of 16 sentences). For nouns, in most cases (70 out of 83) the supertype is contained in the innermost and leftmost NP containing at least one NN. It is the whole NP (discarding leading DTs) if it exists as an entry in WN, or the largest rightmost sequence that exists in WN otherwise. In the last case, the remaining leftmost words correspond to one or more differentia qualities. If the NP contains CCs, more than one supertype exist, and can be

[2]POS tags and non-terminal symbols are listed in Appendix A

Pattern	**Total**
(supertype) (differentia quality)	27
(supertype) (differentia event)	13
(differentia quality) (supertype)	6
(supertype) (differentia event) (event location)	5
(supertype) (differentia quality) (purpose)	3
(accessory determiner) (supertype) (differentia event)	3
(accessory determiner) (supertype) (differentia quality)	2
(supertype) OR(differentia quality)+	2
(supertype) (origin location)	2
(differentia quality) (supertype) (differentia quality)	2
OR(supertype)+ (differentia event)	2
(differentia quality)+ (supertype)	2
(differentia quality)+ (supertype) (differentia event)	2
Other	29
Total	**100**

Table 3.2: Distribution of semantic patterns for the analyzed definitions. "Other" refers to patterns that occurred only once. (*role*)+ indicated the occurrence of two or more consecutive instances of the role, and OR(*role*)+ indicates the same, but with the conjunction "or" connecting the instances.

identified following the same rules just described. The 13 sentences that don't fit this scenario include some non-frequent grammatical variations, parser errors and the presence of accessory determiners, described later.

Differentia quality: for verbs, this is the most common identifying component in the definition. It occurs in 14 out of the 16 sentences. The other two ones are composed by a single supertype (that would better be seen as a synonym), and by a conjunction of two supertypes. The differentia quality is usually a PP (5 occurrences) or an NP (4 occurrences) coming immediately after the supertype. JJs inside ADJPs (3 occurrences) or RBs inside ADVPs (1 occurrence) are also possible patterns, where the presence of CCs indicates the existence of more than one differentia quality. For nouns, two scenarios stand out: the differentia quality preceding the supertype, where it is composed by the leftmost words in the same NP that contains the supertype but are not part of the supertype itself (the NP's "leftovers"), as described above; and the differentia quality coming after the supertype, predominantly composed by a PP, where the prevailing introductory preposition is "of". These two scenarios cover approximately 90% of all analyzed sentences where one or more differentia qualities occur.

Role	Most common syntactic patterns
Supertype	innermost and leftmost NP containing at least one NN
Differentia quality	leftovers in the innermost and leftmost NP; PP beginning with "of"'
Differentia event	SBAR; VP
Event location	PP inside an SBAR or VP, possibly having a location named entity
Event time	PP inside an SBAR or VP, possibly having a time interval named entity
Origin location	PP not inside an SBAR or VP, possibly having a location named entity
Quality modifier	NN, JJ or RB referring to an element inside a differentia quality
Purpose	VP beginning with TO; PP beginning with "for" with a VP right after
Associated fact	SBAR; PP not beginning with "for" with a VP right after
Accessory determiner	whole expression before supertype; common accessory expression
Accessory quality	JJ, presence of a differentia quality, common accessory word
[Role] particle	PRT

Table 3.3: Most common syntactic patterns for each semantic role.

Differentia event: differentia events occur only for nouns, since verbs can't represent entities that can participate in an event (i.e., they are *endurants* in the ontological view, and only *perdurants* can participate in events). They are predominantly composed by either an SBAR or a VP (under a simple clause or not) coming after the supertype. This is the case in approximately 92% of the analyzed sentences where differentia events occur. In the remaining samples, the differentia event is also composed by a VP, but under a PP and immediately after the introductory preposition.

Event location: event locations only occur in conjunction with a differentia event, so they will usually be composed by a PP appearing inside an SBAR or a VP. Being attached to a differentia event helps to distinguish an event location from other roles also usually composed by a PP, but additional characteristics can also provide some clues, like, for example, the presence of named entities denoting locations, such as "Morocco" and "Lake District", which appear in some of the analyzed definitions.

Event time: the event time role has the same characteristics of event locations: only occurs in conjunction with a differentia event and is usually composed by a PP inside an SBAR or a VP. Again, additional information such as named entities denoting time intervals, for example, "the 19th century" in one of the analyzed definitions, is necessary to tell it apart from other roles.

Origin location: origin locations are similar to event locations, but occurring in the absence of an event, so it is usually a PP that does not appear inside an SBAR or a VP and that frequently contains named entities denoting locations, like "United States", "Balkan Peninsula" and "France" in our sample definitions. A special case is the definition of entities denoting instances, where the origin location usually comes before the supertype and is composed by an NP (also frequently containing some named entity), like the definitions for *Charlotte_Anna_Perkins_Gilman* – "United States feminist" – and *Joseph_Hooker* – "United States general [...]", for example.

Quality modifier: quality modifiers only occur in conjunction with a differentia quality. Though this role wasn't very frequent in our analysis, it is easily identifiable, as long as the differentia quality component has already been detected. A syntactic dependency parsing can show whether some modifier (usually an adjective or adverb) references, instead of the supertype, some of the differentia quality's elements, modifying it.

Purpose: the purpose component is usually composed by a VP beginning with a TO ("to") or a PP beginning with the preposition "for" and having a VP right after it. In a syntactic parse tree, a purpose can easily be mistaken by a differentia event, since the difference between them is semantic (the differentia event is essential to define the entity, and the purpose only provides additional, non-essential information). Since it provides complementary information, it should always occur in conjunction with an identifying role, that is, a differentia quality and/or event. Previously detecting these identifying roles in the definition, although not sufficient, is necessary to correctly assign the purpose role to a definition's segment.

Associated fact: an associated fact has characteristics similar to those of a purpose. It is usually composed by an SBAR or by a PP not beginning with "for" with a VP immediately after it (that is, not having the characteristics of a purpose PP). Again, the difference between an associated fact and a differentia event is semantic, and the same conditions and principles for identifying a purpose component also apply.

Accessory determiner: accessory determiners come before the supertype and are easily recognizable when they don't contain any noun, like "any of several", for example: it will usually be the whole expression before the supertype, which, in this case, is contained in the innermost and leftmost NP having at least one NN. If it contains a noun, like "a type of", "a form of", "any of a class of", etc., the recognition becomes more difficult, and it can be mistaken by the supertype, since it will be the leftmost NP in the sentence. A more extensive analysis in the WN database to collect the most common expressions used as accessory determiners was performed in order to provide further information for the correct role assignment.

Accessory quality: the difference between accessory qualities and differentia qualities is purely semantic. It is usually a single adjective, but the syntactic structure can't help beyond that in the accessory quality identification. Again, the presence of an identifying element in the definition (preferably a differentia quality) associated with knowledge about most common words used as accessory qualities can provide important evidence for the correct role detection.

[**Role**] **particle:** although we believe that particles can occur for any role, in our analysis it was very infrequent, appearing only twice and only for supertypes. It is easily detectable for phrasal verbs, for example, *take off* in "take the staples off", since the particle tends to be classified as PRT in the syntactic tree. For other cases, it would be necessary a larger number of samples such that some pattern could be identified and a suitable extraction rule could be defined.

Figure 3.3, which shows both the (simplified) parse tree and the role labeling for the definition of the concept "lake_poets", illustrates some of the relationships between syntactic structures and the semantic roles shown in Table 3.3. Each relevant segment, that is, a segment that can be considered a self-contained amount of information, is assigned a role label, which is closely related to the segment's syntactic classification.

3.3 Definition Filter

As described in Section 3.2, the proposed conceptual model for representing a set of lexical definitions as a DKG derives from the basic *genus-differentia* pattern. In fact, a well-formed definition should contain at least the definiendum's **type**, informing *what it is*, and its **essential attributes**, stating *how it differs from*

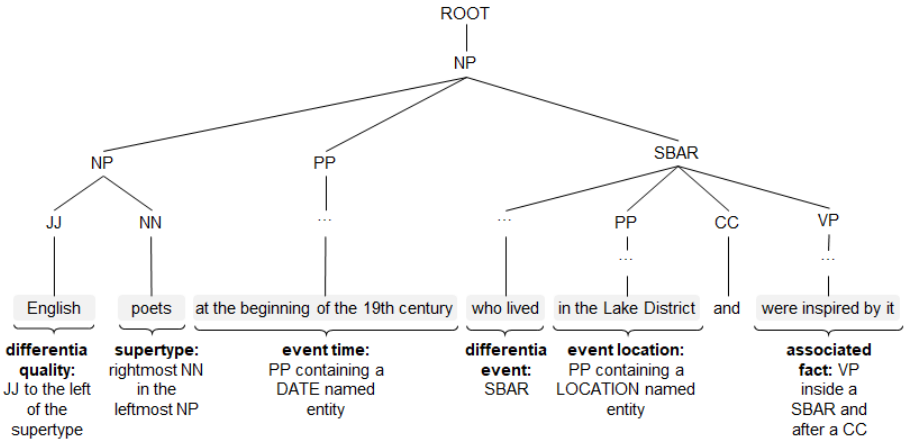

Figure 3.3: The (simplified) parse tree for the definition of the concept "lake_poets" and the correspondences between each relevant phrasal node and the definition's semantic roles.

other entities under the same type. In the proposed conceptual model, this translates to at least a *supertype* and a *differentia* (quality or event) role.

However, we can't always guarantee that definitions in a dictionary will follow these principles. As an example, consider the following definitions extracted from WordNet:

(1) **poorness:** less than adequate
(2) **accession:** agreeing with or consenting to
(3) **codfish_ball:** usually made of flaked salt cod and mashed potatoes
(4) **worrier:** thinks about unfortunate things that might happen
(5) **brown_lemming:** of northwestern Canada and Alaska
(6) **clerid_beetle:** predacious on other insects

What can be noticed in all of these definitions is that none of them explicitly says what the entity being defined is. The definition in (1) could be suitable for an adjective but, since "poorness" is a noun, it would be more appropriate to say that it is "*the quality* of being less than adequate". The same happens in (2): starting a definition with a verb is suitable for defining another verb but not a noun; "accession" should be defined as "*the act* of agreeing with or consenting to". Similarly, (3) and (4) contain a differentia quality and a

differentia event, respectively, but both refer to no subject. It should be explicit that "codfish_ball" is "*a dish* made of flaked salt cod and mashed potatoes" and "worrier" is "*a person* who thinks about unfortunate things that might happen". The definitions in (5) and (6) are even poorer: besides not stating what the definiendum's type is, the only information given regards non-essential attributes, which by no means uniquely characterize the entity being defined.

WordNet is a lexical resource built by expert lexicographers, but even so, it contains a number of ill-formed definitions. This problem becomes more noticeable for collaborative resources, such as the online dictionary Wiktionary, where definitions are entered by a large number of lay users.

The issue introduced by ill-formed definitions is twofold: first, if we use definitions for explaining entailment decisions, poor definitions will generate poor justifications. Second, since, as detailed in Section 3.2.3, definition semantic roles are closely related to specific syntactic patterns, definitions that don't fit the most common patterns are more prone to classification errors because it is harder to automatically induce the right label when the proper syntactic cues are absent.

For preventing ill-formed definitions from being included in the graph knowledge base, a definition filter which implements a set of rules regarding the syntactic structure of a well-formed definition was developed. Although, as mentioned before, a well-formed definition should contain a supertype role and at least a differentia role, we opted for a more flexible interpretation, characterizing as an ill-formed definition one that does not inform the definiendum's type. Therefore, the filter is meant to eliminate definitions for which the supertype role is lacking, such as the ones in examples (1) through (6) above. This choice was made due to the fact that the supertype is the only mandatory role node in the final RDF definition graph, where all the other nodes are structured around it, as described later in the methodology steps description.

The set of rules was obtained through a manual analysis of the syntactic structure of a sample of definitions. Similarly to the conceptual model creation, WordNet was used as the corpus for this analysis. A set of 6,000 definitions were randomly chosen, being 5,148 noun definitions and 852 verb definitions. Using the Stanford parser (Manning et al., 2014), the syntactic parse tree for each definition was generated, and the most common syntactic patterns were identified, grouped and formatted as a rule. The rules not only describe the expected shape of a definition but also account for parser errors, identifying a well-constructed sentence even when the parser fails to capture the right struc-

ture whenever it is possible. The rules differ for noun and verb definitions, and are detailed next.

3.3.1 Noun Definition Patterns

A noun must be defined in terms of another noun, which indicates its *kind*. For example, "a *man* is a *person*", "a *tree* is a *plant*", and "*joy* is an *emotion*" would be proper ways of starting a noun definition. Syntactically, this means that a noun definition must start with a *noun phrase* (NP). In fact, this is the most common syntactic pattern observed in the sample, and can be considered the standard noun definition pattern, from which the first rule derives[3]:

Rule N1: *Definition starts with an NP having at least one NN*

Formally, the leftmost node in the parse tree must be an NP having at least one NN and no other NP under it. The NN is the supertype role candidate, and it may or may not be preceded by other words, usually a determiner and one or more qualifiers. Example in Figure 3.4, where the dashed and shadowed areas indicate the satisfied rule conditions.

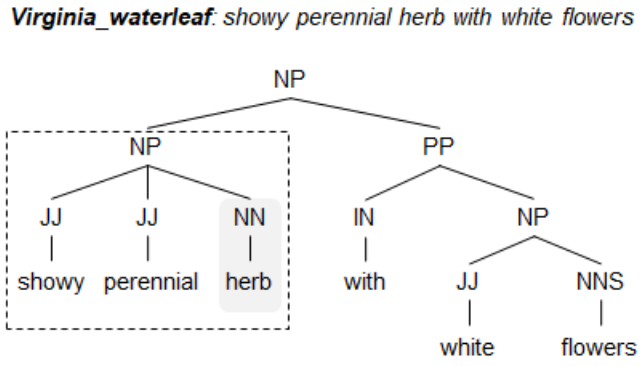

Figure 3.4: Example of a noun definition following the pattern defined by the rule N1.

Rule N1 defines the mandatory structure for a noun definition. Nevertheless, definitions that follow this pattern can be erroneously filtered out due to parser

[3]POS tags and non-terminal symbols are listed in Appendix A

errors. The remaining rules are intended to overcome the most common parser misclassifications observed in the analyzed sample:

Rule N2: *Definition starts with an NP having a word that exists as a noun*

Formally, the leftmost node in the parse tree is an NP having no other NP under it, and whose rightmost child is not an NN, but is a word that exists as a noun in WordNet. That means the rightmost word is a NN misclassified as a JJ, VB, CD, etc. Example in Figure 3.5, where the shadowed branch indicates the parser misclassification; the highlighted area satisfies the rule conditions since "helping" exists as a noun in WordNet.

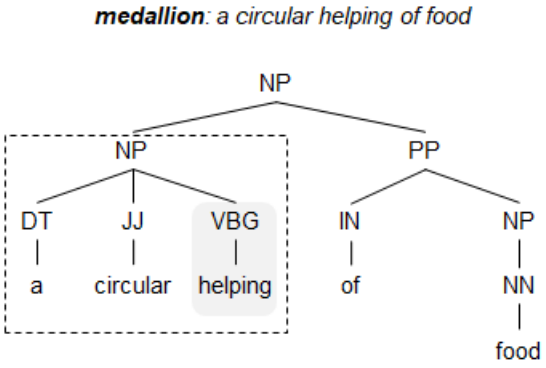

Figure 3.5: Example of a noun definition following the pattern defined by the rule N2.

Rule N3: *Definition starts with an ADJP or ADVP having a word that exists as a noun*

Formally, the leftmost node in the parse tree is an ADJP or ADVP having no NP under it and whose rightmost child is a word that exists as a noun in WordNet. Example in Figure 3.6, where the whole tree satisfies the rule conditions, since there is no NN but "ecclesiastic" exists as a noun in WordNet.

Rule N4: *Definition starts with an NP having an NN followed by a complete simple or subordinate clause*

Formally, the leftmost node in the parse tree is an NP having an NN whose

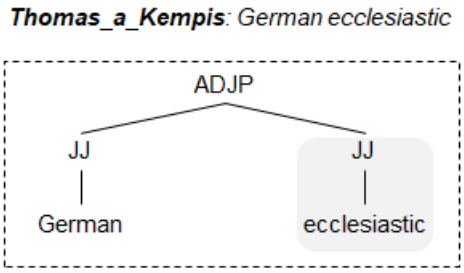

Figure 3.6: Example of a noun definition following the pattern defined by the rule N3.

immediate right sibling is an S or an SBAR. Example in Figure 3.7, where the leftmost NP, which in this case is the root node, has another NP under it (and, therefore, does not satisfy the rule N1), but has an NN having an S immediately to its right.

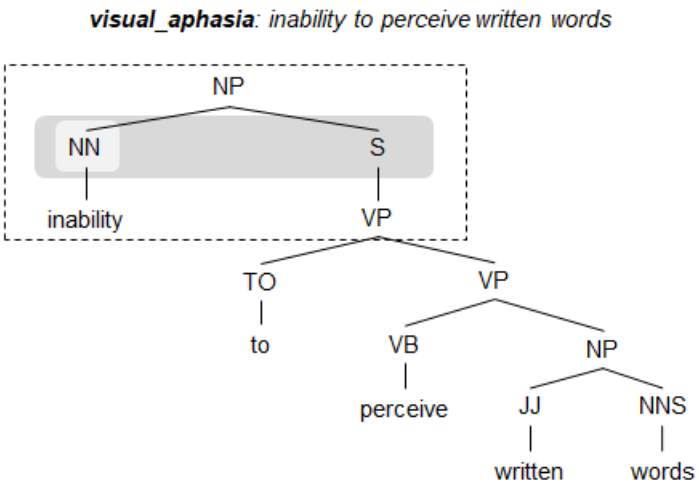

Figure 3.7: Example of a noun definition following the pattern defined by the rule N4.

Rule N5: *Definition starts with an NP having a POS followed by a NN*

Formally, the leftmost node in the parse tree is an NP having a POS that has

a NN as a right sibling. Parse trees whose leftmost NP contains a POS are excluded from rule N1 even if such NP also contains an NN, because this NN is usually not the supertype, therefore it is necessary to ensure there is another NN after the POS. Example in Figure 3.8, where the dashed area indicates the leftmost NP that satisfies the rule. We can notice that it contains an NN ("one") which is not the supertype, and a POS which has an NN to its right. This NN is part of the actual supertype, namely "native language".

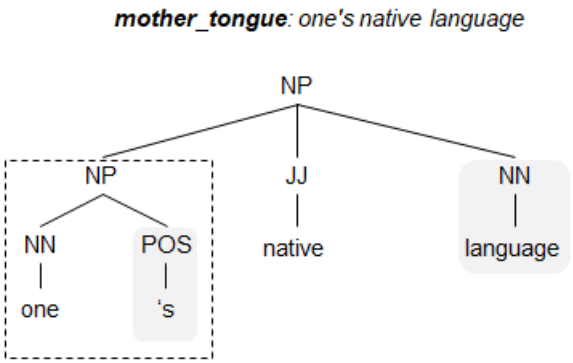

Figure 3.8: Example of a noun definition following the pattern defined by the rule N5.

3.3.2 Verb Definition Patterns

Verb definitions are usually simpler than noun ones: they either start with another verb, or with a verb preceded by an adverb. In the first case, the sentence will start with a VP, and in the second, it may start with an ADVP or an RB, followed by a VP. These scenarios can be considered the standard verb definition patterns, and derive the first two rules:

Rule V1: *Definition starts with a VP*

Formally, the leftmost node in the parse tree must be a VP. Example in Figure 3.9, where the dashed area indicates the satisfied rule condition. It is not necessary to check for a VB in this branch, since a VP will always contain such

node, as shown by the shadowed area in the picture.

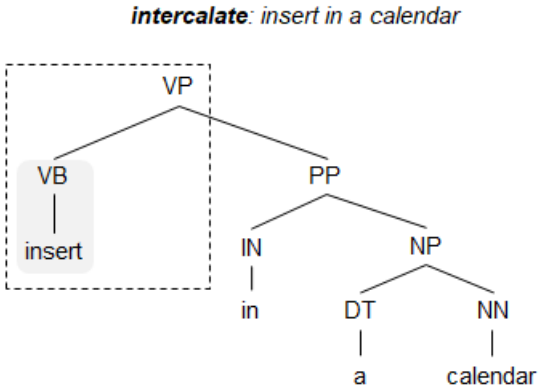

Figure 3.9: Example of a verb definition following the pattern defined by the rule V1.

Rule V2: *Definition starts with an ADVP or RB before a VP*

Formally, the leftmost node in the parse tree is an ADVP, or an RB not under an ADVP, having a VP as the immediate right sibling. This means that the RB (under an ADVP or not) is qualifying the verb and, if removed, the definition fully satisfies the rule N1. Example in Figure 3.10, where the highlighted areas indicate the satisfied rule conditions.

Rules V1 and V2 define the mandatory structures for a verb definition. As for noun definitions, complementary rules were defined to overcome the most common parse errors observed in the analyzed sample:

Rule V3: *Definition starts with an NP whose leading NN exists as a verb*

Formally, the leftmost node in the parse tree is an NP whose leftmost child is an NN that exists as a verb in WordNet. Since in English a large number of words double as a noun and a verb, this is the most common parser misclassification for verb definitions. Example in Figure 3.11, where the highlighted areas indicate the satisfied rule conditions, since “whip” exists as a verb in WordNet.

Rule V4: *Definition starts with an ADJP whose leading JJ exists as a verb*

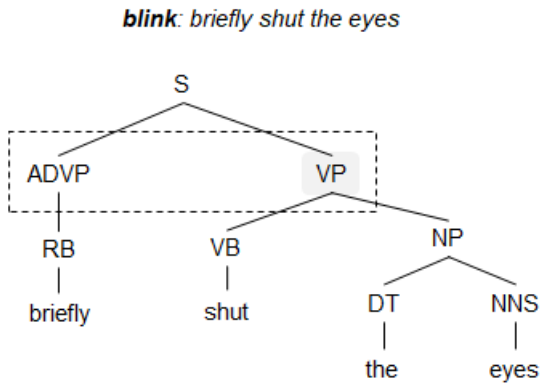

Figure 3.10: Example of a verb definition following the pattern defined by the rule V2.

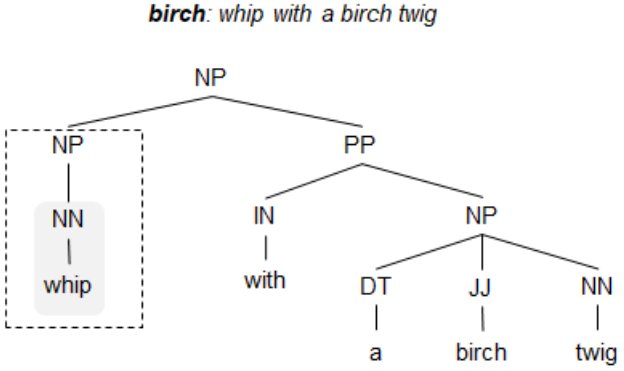

Figure 3.11: Example of a verb definition following the pattern defined by the rule V3.

Formally, the leftmost node in the parse tree is an ADJP whose leftmost child is a JJ that exists as a verb in WordNet. Example in Figure 3.12, where the highlighted areas indicate the satisfied rule conditions, since, although "subject" is tagged as an adjective, it also exists as a verb in WordNet.

Rule V5: *Definition starts with an ADVP whose leading RB exists as a verb*

Formally, the leftmost node in the parse tree is an ADVP whose leftmost child

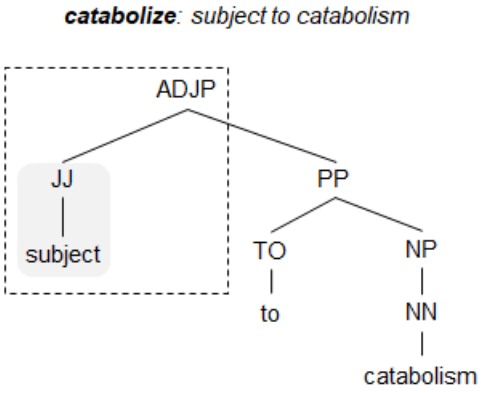

Figure 3.12: Example of a verb definition following the pattern defined by the rule V4.

is an RB that exists as a verb in WordNet. Such ADVP must not be followed by a VP, which means it is not qualifying a verb, otherwise, it would satisfy the rule V2. Example in Figure 3.13, where the highlighted areas indicate the satisfied rule conditions, since "forward", besides being an adverb, also exists as a verb in WordNet.

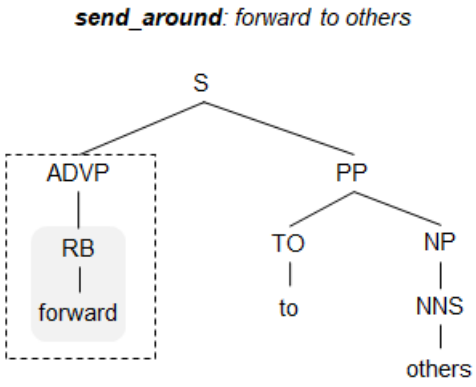

Figure 3.13: Example of a verb definition following the pattern defined by the rule V5.

Table 3.4 lists the formalized set of rules. A definition parse tree D is composed of a set of nodes, indicated by the elements enclosed by "(" and ")". The node after the first $*$ is the leftmost (non-terminal) tree node; $*$ indicates the existence of 0 to n subtrees, x is any terminal symbol, and | stands for "or".

N1	$D \rightarrow (* \ (NP \ * \ (NN)) \ *)$
N2	$D \rightarrow (* \ (NP \ * \ (x)) \ *)$, x is a WordNet noun
N3	$D \rightarrow (* \ (ADJP \mid ADVP \ * \ (x)) \ *)$, x is a WordNet noun
N4	$D \rightarrow (* \ (NP \ * \ (NN) \ (S \mid SBAR) \ *) \ *)$
N5	$D \rightarrow (* \ (NP \ * \ (POS) \ * \ (NN) \ *) \ *)$
V1	$D \rightarrow (* \ (VP \ *) \ *)$
V2	$D \rightarrow (* \ (ADVP \mid RB) \ (VP) \ *)$
V3	$D \rightarrow (* \ (NP \ (NN) \ *) \ *)$, NN is also a WordNet verb
V4	$D \rightarrow (* \ (ADJP \ (JJ) \ *) \ *)$, JJ is also a WordNet verb
V5	$D \rightarrow (* \ (ADVP \ (RB) \ *) \ *)$, RB is also a WordNet verb, no VP after RB

Table 3.4: Definition filter rules.

The definition filter based on this set of rules was implemented with the aid of the TRegEx tool (Levy & Andrew, 2006), which allows the specification of queries with a wide range of conditions over the syntactic parse tree. In order to check whether the set of rules reflects the actual composition of a complete lexicon, the definition filter was applied to the whole noun and verb WordNet databases. The distribution of patterns for both the analyzed sample and the full databases are shown in Table 3.5. As can be noted, rules N1 for nouns and V1 for verbs cover the vast majority of definitions in both the sample and the full WN definition database. While the rule N1 covers 95% of the noun definitions in the sample and 92,6% in the full database, rule V1 covers around 95,5% of verb definitions in both sets. With some slight variation, the pattern distribution in the sample reflects that of the full WordNet. The total of rejected definitions encompasses both noun and verb ones. Although rules N2 through N5 and V2 through V5 covers only a small percentage of definitions, it is important to account for such patterns because, as mentioned before, other linguistic resources, especially the ones built collaboratively like Wiktionary, may have less syntactic regularity, presenting a wider variation of definition styles, which can lead to varying parser interpretations. Therefore, it is desirable to tell parser errors apart from actual definition design errors as much as possible.

The accuracy of the filtering procedure was verified on the analyzed 6,000 sample. The whole set of rejected definitions and a random sample of the same

Rule	WordNet 6,000 Sample	WordNet Full Noun and Verb Databases
N1	95,00%	92,60%
N2	1,50%	2,30%
N3	1,17%	0,15%
N4	0,38%	1,00%
N5	0,11%	0,36%
V1	95,50%	95,40%
V2	0,10%	0,05%
V3	4,20%	4,00%
V4	0,23%	0,50%
V5	0,10%	0,01%
Rejected	2,66%	2,26%

Table 3.5: Distribution of syntactic patterns detected during the definition filtering.

size from the accepted definitions were analyzed, showing that around 70% of rejected definitions were in fact invalid ones, and over 98% of the filtered, i.e., accepted definitions are indeed valid ones. False negatives, that is, rejected definitions that are in fact valid ones, are mainly due to very complex an uncommon syntactic parse trees. Nevertheless, given that the main goal of the filter is to prevent ill-formed definitions from being included in the final graph knowledge base, the high precision (98% of true positives) ensures that the intended cleaning purposes are met. The definition filter was integrated as a preprocessing stage in the knowledge graph construction methodology, which is described in the next Section.

3.4 Graph Construction Methodology

Information Extraction (IE) methods have been largely used in NLP for selecting and structuring relevant data from natural language unstructured text in order to populate some kind of database (Cowie & Wilks, 2000). In the lexical definitions field, IE has also been widely explored with the aim of constructing structured knowledge bases from machine-readable dictionaries (Vossen, 1992; Calzolari, 1991; Vossen, 1991; Vossen & Copestake, 1994).

The use of syntactic information from dictionary definitions is a constant across the different attempts to build structured representations of lexicons. Among early efforts, it is remarkable the creation of the LKB, a Lexical Knowl-

edge Base (Copestake, 1991) based on typed-feature structures that can be seen as a set of attributes for a given concept, such as "origin", "color", "smell", "taste" and "temperature" for the concept "drink", for example. The definitions from a machine-readable dictionary were parsed to extract the definiendum's genus and differentiae, and the values represented by the differentiae filled in the feature structures for that genus. But, since the features, that is, the relevant attributes for a given entity, had to be defined in advance, only a restricted domain could be covered by this approach.

Besides the entity-attributes structure, syntactic parsing was also already used to identify semantic relations such as *is-a*, *part-of*, etc., to convert a dictionary into a directed graph (Dolan, Vanderwende, & Richardson, 1993). Other graph conceptual models were also adopted, such as the one containing only three types of edges, numbered from 0 to 2: the 0-edge represents unary predicates and the 1 and 2-edges connects binary predicates to their arguments (Recski, 2016). In common, these approaches work at the word-level, converting every single word in the definition into a node. This strategy can increase the complexity of the information retrieval over the final knowledge base, given that it may be necessary to concatenate the contents of several nodes to obtain meaningful enough information about an entity. Representations at the multiword expression level, also through the syntactic-semantic analysis of textual definitions, have been proposed (Bovi, Telesca, & Navigli, 2015), but the resulting graphs are used only as an intermediary resource for the final goal of extracting semantic relations between the entities present in the definition.

The methodology for automatic graph construction proposed in this work follows the principle that a knowledge graph derived from a dictionary, where each node is a meaningful phrase, allows the retrieval of intelligible data from a path made up by only a few nodes. This is possible because every node contains a piece of self-contained information about the definiendum. This methodology was developed with the aid of the semantic and syntactic patterns identified in Section 3.2, as well as their association rules. It includes information extraction, semantic role labeling, and RDF conversion procedures, resulting in a framework which receives as input a set of plain text natural language definitions and outputs a structured DKG.

Splitting a natural language definition into comprehensible segments allows the selection of the portions of information regarding an entity's description that are relevant for a certain reasoning task. For example, consider the definition for the concept "lake_poets", which was classified according to the model

described in Section 3.2, illustrated in Figure 3.3. When retrieving data related to this concept, we could be interested only in origin- ("lake poets are English poets"), time- ("lake poets are poets at the beginning of the 19th century") or space- ("lake poets are poets who lived in the Lake District") related information. When each of those roles is represented as a node in the graph we can focus only on the path containing the nodes of interest. Moreover, since the definition is split into segments rather than single words, each node contains a comprehensible amount of information, avoiding the need to visit several nodes to gather intelligible phrases.

The methodology for classifying and structuring natural language definitions to generate a DKG – a knowledge graph using the RDF data model – comprised the following steps:

Synsets Sample Selection

As mentioned in Section 3.2.3, classifying definitions according to the conceptual model described in Section 3.2.2 is a Semantic Role Labeling task, which can be performed automatically but requires an initial set of annotated training data. In order to use a supervised machine learning model to classify the data, 4,000 WordNet synsets[4], along with their definitions, were randomly selected to compose the training set. Out of this 4,000 synsets, 3,443 are noun synsets and 557 are verb synsets (as mentioned before, the verb database size is only around 17% of the noun database size).

Automatic Pre-Annotation

Using the association rules described in Section 3.2.3, an automatic pre-annotation procedure implementing a rule-based heuristic was used to classify the sample set of 4,000 definitions. This procedure uses the Stanford parser (Manning et al., 2014) to generate the syntactic parse tree for each definition and the TRegEx tool (Levy & Andrew, 2006) to identify the relevant phrasal nodes and then assign the semantic roles more often associated to them (see Table 3.3).

Figure 3.14 shows an example of pre-annotation, depicting the parse tree generated for the definition of the concepts "Scotch" – "whiskey distilled in Scotland" – and the semantic roles automatically assigned to each phrasal node. Note that, after being classified as a differentia event, the VP is further analyzed

[4]A *synset* in WordNet is a set of synonyms words or phrases which share the same definition

and a PP containing an event location is identified and assigned its own role label.

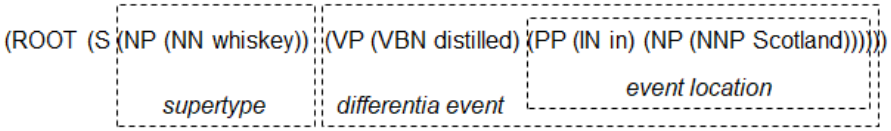

Figure 3.14: Syntactic parse tree for a definition and assigned semantic role labels.

Data Curation

After the automatic pre-annotation, the definitions were submitted to a manual curation procedure for label validation. This task was performed with the aid of the Brat[5] annotation tool. At this stage, all misclassifications were manually fixed and segments missing a role were assigned the appropriate one. Misclassifications and missing roles are due to parser errors or insufficient information (for instance, a PP inside a VP may not contain any named entity, making it hard to correctly distinguish between an event time and an event location). The manual data curation ensured that every segment in each definition, apart of leading determiners and conjunctions between roles (as opposed to conjunctions inside roles), was associated with a semantic role label.

Classifier Training

The curated data was then converted to the IOB (Inside-Outside-Beginning) format and used to train a Recurrent Neural Network (RNN) machine learning model designed for sequence labeling.

The model used was the RNN implementation provided by Mesnil et al. (2015), which reports state-of-the-art results for the slot filling task. Besides focusing on Spoken Language Understanding (SLU), that is, targeted understanding of human speech directed at machines, they define the slot filling task as "a sequence classification problem in which contiguous sequences of words are assigned semantic class labels", which also fits our purposes.

[5] http://brat.nlplab.org/

The annotated dataset was split into training (68%), validation (17%) and test (15%) sets. The best accuracy reached during training was of 73.84% on the validation set and 77.24% on the test set.

Database Classification

The trained classifier was then used to label the set of definitions from different lexical resources. This set was composed of the filtered definitions generated as the output of the filtering procedure described in Section 3.3. Since semantic roles are closely related to syntactic patterns in the definitions, by excluding ill-formed definitions, that is, definitions that don't fit the most common syntactic patterns, it is possible to reduce the probability of classification errors. Lexical resources with different characteristics were chosen, as will be detailed later, and each resource gave origin to a different DKG, which allowed us to test the proposed text entailment approach with different configurations and compare the resources among them.

Data Post-Processing

Since some of the classified definitions lacked the supertype role label, the labeled data had to pass through a post-processing phase. The supertype is a mandatory component in a well-formed definition and, as will be detailed later, the RDF model is structured around it. Following the same syntactic rules adopted for pre-annotation, missing supertypes were identified and the roles around it had their limits adjusted, while the remaining classification was kept unchanged. It is worth reminding that all the definitions submitted to the classifier have been previously filtered, therefore most of the ill-formed ones were removed and the lack of a supertype in the resulting labeling is most likely due to a classification error than a syntactic malformation.

Figure 3.15 shows an example of post-processing, picturing the output of the classification step for the definition of the concept "spur" – "any sharply pointed projection" – and the fixed labeling after the post-processing phase.

RDF Conversion

Finally, the labeled definitions were serialized in RDF format. In the RDF graph, the entity being defined is a node (the *entity node*), and each semantic role in its definition is another role (the *role nodes*). The entity node is linked to

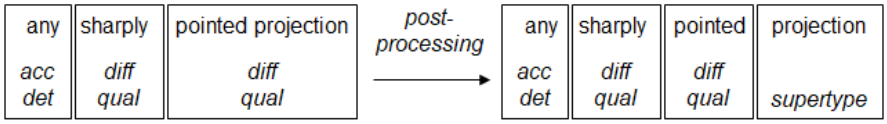

Figure 3.15: Classified definition missing a supertype fixed in the post-processing phase.

its supertype role node, which is, in turn, linked to all the other role nodes. More specifically, a supertype linked to a role is a *reified* node, and this reified node is linked to the entity node. Reification is also used when a role has components, such as event time and/or location for a differentia event and quality modifier for a differentia quality. In this case, the component is linked to its main role, composing a reified node which is linked to the supertype node, creating another reified node which is eventually linked to the entity node. This structure allows the relationships to be fully contextualized.

As an example, consider again the definition for the concept "lake_poets" depicted in Figure 3.3. Figure 3.16 shows the simplified (without reification) RDF representation for this definition. The node defined by the concept "poet" will be linked to several other nodes in the graph, but it is linked to the differentia quality node "English" only in the context of this definition. This is necessary because, like any other entry in the dictionary, the concept "poet" has its own definition, which is represented as another subgraph, making the whole graph interconnected through the words that appear in each definition, but still allowing the information retrieval process to focus unambiguously on the set of nodes pertaining to a specific definition.

Supertype nodes are always represented as resources (ellipses in the graphic representation, according to the RDF notation). The differentia quality and differentia event nodes can be represented as either resources, when they have components (event times and/or locations, or quality modifiers) to be linked to, or literals (rectangles in the graphic notation) otherwise. All the other roles are represented as literals, and properties are named after role names[6].

Figure 3.17 schematizes the graph construction methodology, which can be divided into two parts: the Classifier Construction and the Graph Construction itself. The Classifier Construction includes the Synset Sample Selection, Auto-

[6]RDF model properties and namespaces are listed in Appendix B

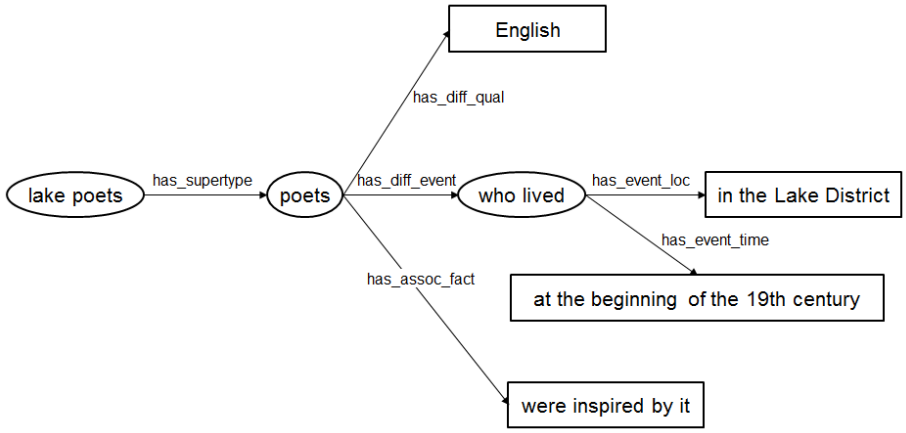

Figure 3.16: RDF representation of a labeled definition.

matic Pre-Annotation, Data Curation, and Classifier Training steps, which were performed a single time, using data from WordNet, with the aim of building a classifier that could label definitions from any dictionary. The products of the Classifier Construction feed the Graph Construction, which encompasses the Database Classification, the Data Post-Processing, and the RDF Conversion steps, performed for every different lexical resource used as a knowledge source, receiving as input the resource's set of filtered natural language definitions and outputting an RDF knowledge graph.

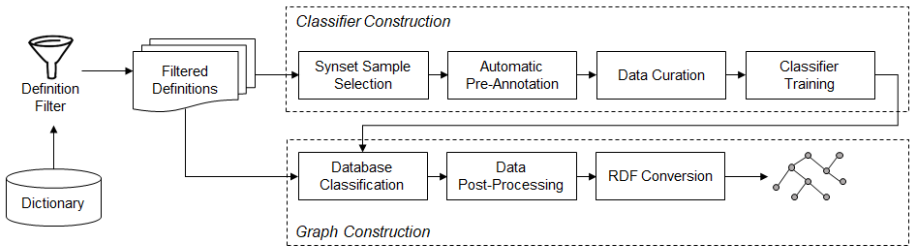

Figure 3.17: Definition knowledge graph construction methodology.

By structuring the commonsense world knowledge contained in dictionary definitions into a model that allows us to focus on specific and meaningful pieces of information regarding an entity, the definition knowledge graphs play

a fundamental part in the proposed text entailment approach, making it possible not only to find semantic relationships between sentences but also explain what these relationships are, contributing to both the core entailment recognition task and the overall system interpretability.

3.5 Summary

In this chapter, the principles supporting the knowledge representation model adopted in the proposed text entailment approach were described in detail. An overview of commonsense knowledge graphs, including a review of the most popular ones, their knowledge extraction methods and main applications, as well as their limitations and disadvantages, was given.

The conceptual model for representing natural language definitions was introduced. A set of semantic roles that reflect the most common structures of dictionary definitions was proposed, and, based on an analysis of a random sample of noun and verb definitions, the main semantic roles and their compositions present on dictionary definitions were identified and named. The semantic patterns were then compared to the definitions' syntactic structure, pointing out the features that would serve as input for the automatic role labeling.

A filtering procedure aimed at cleaning the data by removing invalid definitions was also described. The filter is based on a set of syntactic patterns identified though a manual analysis of a random sample of noun and verb definitions. These patterns derived a set of rules which ensure that definitions included in the knowledge base have at least a supertype role and is, therefore, suitable to be represented in a graph format, according to the proposed conceptual model.

Finally, a methodology for converting whole dictionaries into knowledge graphs was presented. This methodology takes as input the set of filtered definitions and outputs an RDF graph whose nodes and relationships reflect the definition semantic role-based conceptual model. The methodology consists of two main parts: the classifier construction phase, where an automatic data preannotation, followed by manual curation to create the training dataset for the subsequent classifier training resulted in a definition role labeler; and the graph construction phase, where the trained classifier labels the data, which is later post-processed for small adjustments and then converted into an RDF graph.

By designing and implementing the conceptual model, filtering procedure and construction methodology, it became possible to turn any lexicon into a knowledge graph, ready to be explored by the proposed interpretable composite text entailment approach and provide the commonsense world knowledge necessary to recognize and justify entailments involving semantic relationships.

Chapter 4

Composite Syntactic-Semantic Interpretable Text Entailment

In this chapter, the proposed approach for recognizing text entailments based on the entailment pair main phenomenon is presented. As described in Chapter 1, we distinguish between two main phenomena that can be observed in the T-H pair: *syntactic*, when only the structures of T and H differ, but the content is the same; and *semantic*, when T and H state different things which are related through some semantic relationship.

For dealing with different phenomena, different methods are applied. These methods are encapsulated into modules, which are combined to make up a syntactic-semantic *composite* system, capable of analyzing the entailment pair, identifying the relevant phenomenon, and applying the suitable method. This composite syntactic-semantic system is also *interpretable*: for semantic entailment pairs, it finds the semantic relationship between T and H and gives a natural language justification explaining what this relationship is, using the definition knowledge graph described in Chapter 3 for both recognizing and explaining the entailment.

The development of the approach was divided into two parts: first, a method for tackling semantic entailments was designed, and then a complete system was developed so both syntactic and semantic entailment could be dealt with. The option for focusing first on semantic entailment aimed at addressing one of the main gaps in the area: as described in Chapter 2, most entailment approaches will heavily rely on syntactic structures to compare T and H, and the

use of external knowledge seldom goes beyond shallow semantic information and paraphrase-style entailment rules. These strategies account satisfactorily for entailments where the equivalence is given by syntactic similarity but constrain the identification of more complex semantic relationships between T and H, so the first problem to be tackled was: *how can we leverage the use of external world knowledge to find the link between sentences which are syntactically very different, but semantically closely related?*

Once the method for identifying semantically related pairs of words in T and H, finding the relationship between them and justifying the entailment decision was successfully developed, the second problem emerged: since this method only works for semantic entailments and can do very little for entailment scenarios where such semantically related pair of words is not present, *how can we build a complete entailment system capable of dealing with any pair, regardless of the entailment phenomena it involves?*

By developing a complete text entailment system, it was possible to combine an existing and tested well-performing algorithm for solving syntactic entailments with the newly developed semantic-oriented approach, not only improving the latter but also allowing the identification of blind spots in both methods, which were addressed by a complementary support module.

Splitting the approach development into two separate stages allowed for the assessment of two different, but complementary, questions, as it will be detailed later in Chapter 5: first, how a semantic-oriented entailment approach compares to the existing entailment algorithms when dealing with more world knowledge-demanding datasets; and second, how an approach capable of distinguishing between syntactic and semantic entailments and applying the most suitable method for each of them compares to single-method approaches, be they syntactic-only or semantic-only.

Therefore, the chapter starts describing the approach for solving entailments where a semantic relationship exists, including an overview of the concept of distributional semantics and how it is used for exploring a knowledge graph and recognizing semantic entailments in an interpretable way. Next, the complete composite interpretable entailment system is detailed, through the description of the entailment pair routing mechanism, the algorithm adopted for solving syntactic entailments, the improvements to the distributional navigation algorithm, and the Context Analysis support module.

4.1 Towards Semantic Entailment Recognition and Justification

Semantic entailments are those where H presents new information derived from T. Checking whether the information in H logically follows from what is stated in T is equivalent to finding the semantic relationship between T and H, which usually holds between a pair of entities represented by words or phrases, one of them in T and the other one in H. It may be a basic relation, such as synonymy or hypernymy, or a more elaborate association, like the ones denoting location, parthood, cause and effect, or purpose, for example. It could even be a more complex relationship, made up by a composition of two or more atomic relations.

Regardless of the relationship type and level of complexity, the entities linked by it will invariably be semantically similar. By *semantically similar*, we mean that they tend to belong to the same thematic group and to appear together on documents covering such theme. Therefore, the semantic similarity between a given pair of words is a good parameter for guiding the discovery of the type of association between them. Its computation and use in the text entailment context are described next.

4.1.1 Distributional Semantics

The term *distributional*, as pointed by Lenci (2008), can sometimes be used interchangeably with *context-theoretic*, *corpus-based* or *statistical*, and "qualify a rich family of approaches to semantics that share a 'usage-based' perspective on meaning, and assume that the statistical distribution of words in context plays a key role in characterizing their semantic behavior" (Lenci, 2008).

Distributional Semantics is, then, an approach for representing aspects of natural language semantics based on the distributional properties of linguistic elements observed in large corpora. It is grounded in the *distributional hypothesis*, which states that words that occur in similar contexts tend to have similar meanings (Turney & Pantel, 2010). The idea, introduced by Firth (1957), behind the distributional hypothesis that "a word is characterized by the company it keeps" is leveraged by the large amount of text currently available on the Web, which provides the necessary volume of data required for the generation of statistical models.

Distributional Semantic Models (DSMs) allow the approximation of a word meaning representing it as a vector summarizing its pattern of co-occurrence

in large text corpora (Marelli et al., 2014). This representation is supported by Vector Space Models (VSMs), in which objects are represented by points in a space and the closer two points are in this space, the more similar they are. The VSM was first explored in the context of the SMART information retrieval system (Salton, 1971), where the points corresponded to documents in a collection, and a user's query was also represented as a point in the same space as the documents. The distance between each document and the user's query was then computed and the documents were sorted in order of increasing distance from the query to be presented to the user.

In DSMs, words, rather than whole documents, are represented as vectors in the vector space. These vectors are derived from a *word-context* matrix, where the rows correspond to words and the columns correspond to contexts where the words occur. The context can be given by words, phrases, sentences, paragraphs, or other patterns (Turney & Pantel, 2010). Therefore, a cell m_{ij} in a word-context matrix M indicates the number of occurrences of the i-th word in the j-th context.

Figure 4.1 (left) shows an example of a word-context matrix derived by verb-object counts from the British National Corpus[1], where the words are nouns, the contexts are given by verbs, and the frequencies indicate the number of documents in the corpus where both the noun and the verb occur. So, for example, out of all the documents where "eat" occurs, "cat" appears in 6, "pig" in 9, and "cup" in only 1.

	get	see	use	hear	eat	kill
knife	51	20	84	0	3	0
cat	52	58	4	4	6	26
dog	115	83	10	42	33	17
boat	59	39	23	4	0	0
cup	98	14	6	2	1	0
pig	12	17	3	2	9	27
banana	11	2	2	0	18	0

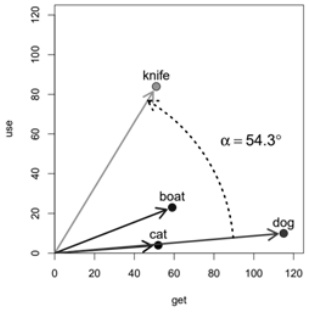

Figure 4.1: A word-context matrix (left) and a representation of the word vectors in a bi-dimensional vector space (right).

[1]Example adapted from Stefan Evert's lecture notes, available at http://tiny.cc/ewefaz

Similar rows in the word-context matrix, which translates to similar distributions, indicate similar word meanings. Figure 4.1 (right) shows the derived word vectors in a bi-dimensional vector space, where the dimensions are given by the contexts "use" and "get". The 2D space is given as an example for simplicity and readability, since word vectors will usually be contained in an n-dimensional vector space, $n \gg 2$. The angle α between vectors can be used as a parameter for computing the distance and, hence, the similarity between them.

A Distributional Semantic Model (DSM) is a scaled and/or transformed word-context matrix where each row represents the distribution of a given word across contexts. Frequencies are normalized, usually in terms of the corpus size, and, since word-context matrices tend to be sparse, the final model is a result of some dimensionality reduction procedure. Figure 4.2 illustrates a possible scaled and normalized version of the matrix in Figure 4.1.

	get	see	use	hear	eat	kill
knife	0.027	-0.024	0.206	-0.022	-0.044	-0.042
cat	0.031	0.143	-0.243	-0.015	-0.009	0.131
dog	-0.026	0.021	-0.212	0.064	0.013	0.014
boat	-0.022	0.009	-0.044	-0.040	-0.074	-0.042
cup	-0.014	-0.173	-0.249	-0.099	-0.119	-0.042
pig	-0.069	0.094	-0.158	0.000	0.094	0.265
banana	0.047	-0.139	-0.104	-0.022	0.267	-0.042

Figure 4.2: A scaled and normalized word-context matrix.

DSMs are composed by dense high-dimensional vectors and benefits from the recent advancements in machine learning techniques for extracting distributional information from very large corpora in an unsupervised manner, without the need of human intervention in the model creation (Turney & Pantel, 2010). DSM-based semantic similarity is, then, computed in terms of word vector similarity. The cosine of the angle between two vectors is one of the most used similarity measures and is given by:

$$similarity(a,b) = cos(\alpha) = \frac{A \cdot B}{||A||\ ||B||} = \frac{\sum_{i=1}^{n} A_i B_i}{\sqrt{\sum_{i=1}^{n} A_i^2}\ \sqrt{\sum_{i=1}^{n} B_i^2}} \tag{4.1}$$

where A and B are the vector representations of words a and b, respectively, and A_i and B_i are the components, or dimensions, of vectors A and B, respectively.

Following Equation 4.1 and using the vectors in Figure 4.1 (all the dimensions considered), we would have, for example, the following similarity values:

$$similarity(cat, knife) = 0.5031$$
$$similarity(cat, boat) = 0.8847$$
$$similarity(cat, dog) = 0.9230$$

We can notice that "cat" and "dog" have the highest similarity value, because, since they belong to the same category ("pets"), they tend to occur in the same contexts. Nevertheless, although DSMs allows us to detect that two words are semantically related, the exact semantic association between them remains unclear, requiring further model interpretation to be made explicit. Even though current distributional models, which usually starts at 300 dimensions, can deliver accurate word vector representations, the semantic similarity measures they provide is only a starting point if the final goal is to reach interpretability by unambiguously identifying the precise semantic relationship between words or sentences, as is the case in the text entailment scenario.

4.1.2 The Distributional Graph Navigation Model

The Distributional Graph Navigation model is based on two main pillars: the use of a knowledge graph automatically extracted from natural language lexical definitions as a commonsense world knowledge base – the DKGs described in Chapter 3 – and a navigation mechanism based on distributional semantics to traverse this graph and find paths between the text and the hypothesis. A path between T and H explains the semantic relationships holding between them, confirming the entailment while also providing evidence that it is true.

As described in Section 4.1.1, DSMs can be used to compute the *semantic similarity/relatedness* measure between words. This computation is used as a

heuristic to navigate in a graph knowledge base in the approach proposed by Freitas et al. (2014), where they define the *Distributional Navigation Algorithm* (DNA), which corresponds to a selective reasoning process in the knowledge graph. Given a pair of terms, namely a *source* and a *target*, and a threshold η, the DNA finds all paths from source to target, with length l, formed by concepts semantically related to the target wrt η (Freitas et al., 2014).

In the text entailment context, and, more specifically, for semantic entailments, the source and target are words from the text and hypothesis, respectively, which we assume have some kind of semantic relationship between them. A path in the definition knowledge graph linking these words, then, explains what this relationship is, confirming the entailment, or rejecting it in case no path is found.

In this work, the DNA was implemented as a *depth-first search* algorithm, exploring first the paths whose next node to be visited has the highest semantic similarity value wrt the target. Given a node in the DKG, starting from the source S, the algorithm retrieves all its neighbors $\{x_1, x_2, ..., x_n\}$ and computes the similarity relatedness $sr(x_i, target)$, keeping only the nodes for which $sr > \eta$ in the set of nodes to be visited next. Each of these nodes generates a new path, and, for each path, the search goes on until the next node to be visited is equal to the target, or until the maximum path length is reached. If no path reaches the target before the maximum number of paths is reached, the search stops. The distributional graph navigation mechanism is schematized in Figure 4.3. The DGN algorithm, which takes as inputs a definition knowledge graph G, a source word S, a target word T, a threshold η, a maximum path length l, and a maximum number of paths m, and outputs the set P of paths from S to T, is listed in Algorithm 1.

Depending on the lexical resource from which the definitions are extracted, entity nodes in a DKG can be identified by a single word or phrase, or by a *synset*, that is, a set of synonym words or phrases. Starting from the source word S, the DGN retrieves all entity nodes identified by S or having S as one of the words in its identifying synset (line 12). Then it retrieves all the neighbors of each entity node, that is, the role nodes that make up its definition (line 19), and keeps only the best ones (line 22).

The next nodes to be visited are given by words present in a role node, which we call the *head words*. The head words are the most relevant words in a role, and are identified following a lexical-syntactic rule-based heuristic:

Algorithm 1 Distributional Graph Navigation Algorithm

```
1: procedure DGN(G, S, T, η, l, m)
2:     P ← ∅
3:     stack ← ∅
4:     newPath ← [S]
5:     Push(stack, newPath)                    ▷ adds the newPath to the stack
6:     while stack ≠ ∅ and P.size < m do
7:         path ← Pop(stack)                   ▷ pulls the path at the top of the stack
8:         nextNode ← path.lastNode
9:         while nextNode ≠ T and path.length < l do
10:            entityNodes ← ∅
11:            for all e_i ∈ G do
12:                if e_i = nextNode then
13:                    Add(entityNodes, e_i)
14:                end if
15:            end for
16:            roleNodes ← ∅
17:            bestRoles ← ∅
18:            for all e_i ∈ entityNodes do
19:                Add(roleNodes, Neighbors(e_i))
20:            end for
21:            for all r_i ∈ roleNodes do
22:                if sr(r_i, T) > η then
23:                    Add(bestRoles, r_i)
24:                end if
25:            end for
26:            bestRoles ← Sort(bestRoles)
27:            nextNodes ← ∅
28:            for all b_i ∈ bestRoles do
29:                Add(nextNodes, HeadWords(b_i))
30:            end for
31:            nextNodes ← Sort(nextNodes)
32:            for x_i ∈ nextNodes, i ← 2, n do
33:                newPath ← path
34:                Add(newPath, x_i)
35:                Push(stack, newPath)
36:            end for
37:            nextNode ← x_1
38:            Add(path, nextNode)
39:            if nextNode = T then
40:                Add(P, path)
41:            end if
42:        end while
43:    end while
44:    return P
45: end procedure
```

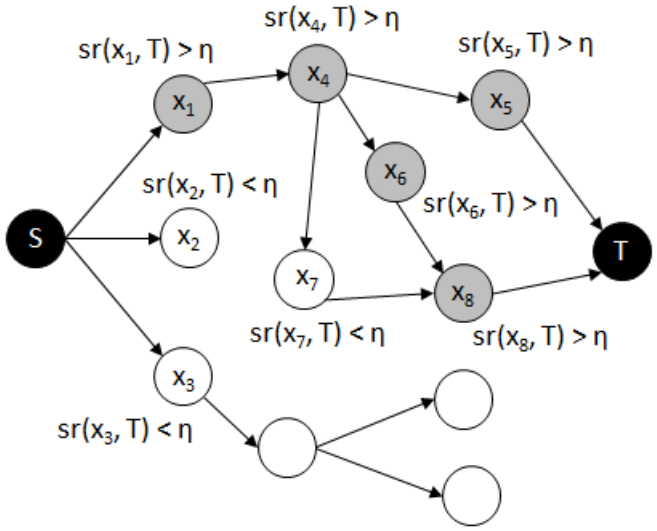

Figure 4.3: The distributional navigation algorithm. Gray nodes, for which $sr(x_i, T) > \eta$, make up valid paths between the source node S and the target node T. The path $\{S, x_1, x_4, x_5, T\}$ is the shortest one

1. For *non-event-centered roles*, such as supertype and differentia quality:
   a. The main noun in the segment

2. For *event-centered roles*, such as differentia event, associated fact or purpose:
   a. The main verb in the segment and its noun objects, if any, or
   b. The main verb and its adjective/adverb modifiers, if no noun object is found

Figure 4.4 shows some examples of head word extraction: in (a) the main noun "writer" is chosen as the head word for the differentia quality role; in (b) the differentia event has "provide" as its main verb, which, in turn, has the noun "information" as its objects, so these are the two role head words; in (c) the main verb "memorize" has no noun objects, so the adverbs "quickly" and "easily" qualifying it are also selected as head words along with the verb. As can be noted across all examples, the supertype role will usually contain a single concept, so its head word will usually be its whole content, be it a single word

or a longer phrase, as long as it exists as an entry in WordNet, the reference dictionary as mentioned in Chapter 3.

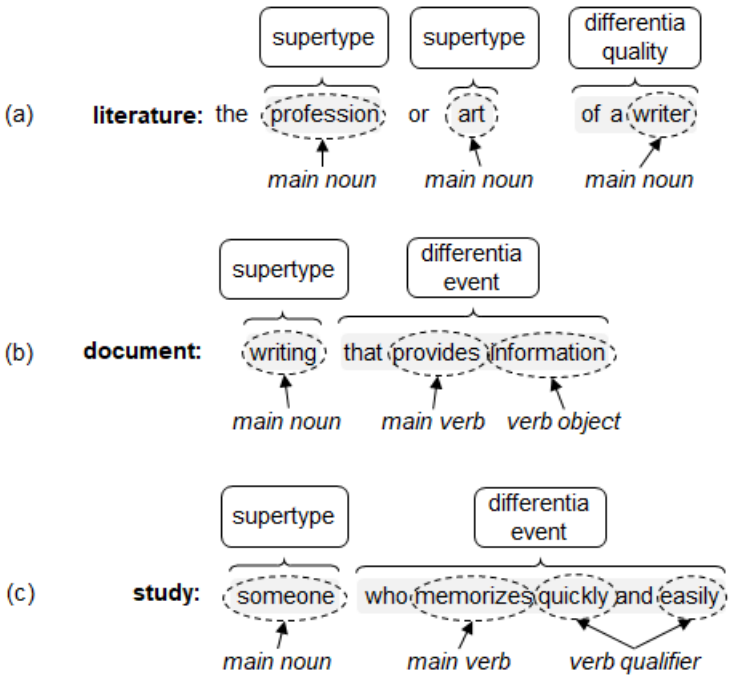

Figure 4.4: Examples of key words extraction for different roles.

The head words are sorted according to their semantic similarity wrt the target T. The highest scoring head word will be the next node to be visited, that is, the next entity node to be searched, and all the other head words are added to a copy of the current path, generating a new path which will be pushed to the stack to be explored later.

Word sense disambiguation comes as a natural consequence of the distributional navigation mechanism while choosing the next nodes to be visited in the graph: by looking for the word/phrases that are more semantically related to the target T, the algorithm naturally selects the correct (or at least the closest) word senses, since unrelated word meanings will have lower similarity scores wrt the target, and the paths containing them will be excluded by the algorithm.

According to Freitas et al. (2014), the worst-case time complexity of the algorithm implemented as a depth-first search "is $O(b^l)$, where b is the branch-

ing factor and l is the depth limit". They show that the algorithm's selectivity ensures that the number of paths does not grow exponentially even when the depth limit increases. In the implementation presented in this work, the maximum number of paths and the maximum path length (depth limit) were obtained empirically in order to optimize the search.

4.1.3 Recognizing Semantic Entailments and Interpreting the Answer

Once both the graph knowledge base and the method to navigate over this graph were developed, they were assembled together into the reasoning mechanism for recognizing and explaining a text entailment. As mentioned before, this reasoning mechanism is aimed at semantic entailments, that is, entailments that require world knowledge, over which some kind of inference is necessary, rather than simple syntactic variations between the text and the hypothesis. For recalling the characteristics of syntactic and semantic entailments, consider the following entailment pair from the Boeing-Princeton-ISI (BPI)[2] dataset:

64.2 T: Skilling was wearing a security tag on his ankle when he stepped into the street to face the press.
64.2 H: Skilling was wearing a security tag.

In this example, the hypothesis is fully contained in the text, and no knowledge external to the entailment pair is necessary, therefore no actual semantic reasoning is required. On the other hand, in the following example, also from the BPI dataset, a simple syntactic analysis would not suffice:

39.3 T: Many cellphones have built-in digital cameras.
39.3 H: Many cellphones can take pictures.

In this case, it is necessary to answer a question: "Given that cellphones have digital cameras, is it true that they can take pictures?". In order to look for the answer to this question, it is necessary to look at the structured definitions in the DKG to check whether the hypothesis is reached from the text in some way. If so, the way this link is established gives a full answer to the original question.

[2] http://www.cs.utexas.edu/users/pclark/bpi-test-suite/

First, it is necessary to identify the relevant elements from T and H for which it is worth to look for a semantic relationship. If the text is too long and includes more than one clause, a sentence simplification is performed to break it into independent simple sentences, and then the sentence that is closest to the hypothesis is chosen among them, using simple Levenshtein edit distance. The edit distance proved to be sufficient at this step, as we just want to identify what text sentence refers to the same topic as the hypothesis, and so share more elements with it. Consider as an example the following BPI entailment pair:

3.6 T: Hanssen, who sold FBI secrets to the Russians, could face the death penalty.
3.6 H: Hanssen received money from the Russians.

After the sentence simplification, the text is split into two sentences:

"Hanssen could face the death penalty"
"Hanssen sold FBI secrets to the Russians"

The second one is the closest to the hypothesis and is selected to compose the new entailment pair.

Next, the *core words* in the text and hypothesis are identified. The core words are similar to the head words for definition's roles (described in Section 4.1.2), but in this case the inputs are full sentences rather than sentence segments, as happens with the roles, so here it is possible to perform a more accurate syntactic analysis. Also following a rule-based heuristic, we have:

1. Get the main noun in the subject
2. Get the main verb in the predicate
3. Get the main verb's noun objects, if any
4. Get the main verb's adjective/adverb modifiers, if no noun object found

Back to the pair 39.3, the core words for the text "Many cellphones have built-in digital cameras" are "cellphones", "have" and "digital cameras"; and for the hypothesis "Many cellphones can take pictures", "cellphones", "take" and "pictures", as shown in Figure 4.5.

We then discard the overlapping words (in Figure 4.5, "cellphones") and words with low *inverse document frequency* (IDF) (Robertson, 2004), which are words that are too frequent and can be reached from almost any node in

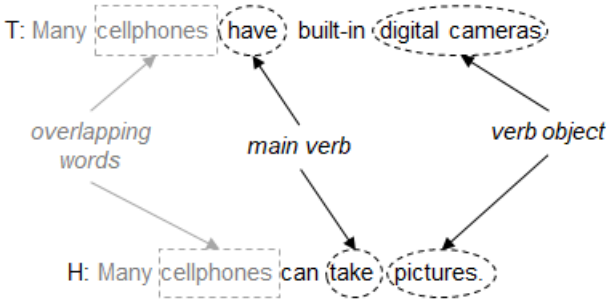

Figure 4.5: Core words extraction for an entailment pair.

the graph, leading to diverting paths, such as the verbs "get", "put", "cause" or "make", to name a few. IDF is calculated using as the corpus the same linguistic resource that gave origin to the knowledge graph being explored by the DGN, where each definition is considered a document. Next, all the remaining words are normalized, resulting in two sets of core words, for the text and the hypothesis, respectively:

$$C_{\mathrm{T}} = \{t_1, t_2, ..., t_{\mathrm{n}}\} \tag{4.2}$$

$$C_{\mathrm{H}} = \{h_1, h_2, ..., h_{\mathrm{m}}\} \tag{4.3}$$

Also using distributional semantics, the set S of semantic similarity measures between all the core words is computed, as a Cartesian product between $\mathrm{C_T}$ and $\mathrm{C_H}$:

$$S = C_{\mathrm{T}} \times C_{\mathrm{H}} \tag{4.4}$$

The results are sorted and the k pairs with the highest similarity values are chosen, being:

$$k = max(n, m) \tag{4.5}$$

where n is the size of C_{T} and m is the size of C_{H}.

Table 4.1 shows the set S for the above-mentioned entailment pair 39.3, with all the possible combinations of words and their respective relatedness scores.

Since $n = 2$ and $m = 2$, $k = max(n, m) = 2$, so the two top scoring pairs are chosen. Although in this example such pairs would be "[digital camera, picture]" and "[have, take]", "have" and "take" would probably have low IDF, being excluded from the set of candidates. IDF depends on the corpus being used but, given that these are very common verbs, we could assume that, regardless of the corpus, they would be discarded, so the most likely final pair for this example is only "[digital camera, picture]".

Pair	**Relatedness Score**
[digital camera, picture]	0.414
[digital camera, take]	0.038
[have, take]	0.263
[have, picture]	0.002

Table 4.1: Relatedness scores for the source-target pairs.

Since each pair is composed of a word (or phrase) from the text and another from the hypothesis, these will be the input for the DGN. For each pair of words found in the previous step, the DGN finds all the paths between them in the definition graph. Finally, among all the paths found, the smallest one is chosen, which is the one that offers the shortest distance between a source and a target and, therefore, shows that their meanings are more closely related. The procedure for recognizing an entailment through the DGN is listed in Algorithm 2 (for readability, further parameters for the DGN procedure were omitted in line 10).

The final path is composed of a sequence of entity nodes and the role nodes that make up those entities' definitions and that are relevant to build a compound relationship between the source and the target. This sequence of nodes is then formatted to provide a human-readable justification explaining the reasoning that led from the text to the hypothesis, giving the necessary evidence that the latter logically follows from the former.

Figure 4.6 shows an example of a path in a DKG between the source "digital camera" and the target "picture", from the entailment pair 39.3. Starting from the source node, the DGN gets all the nodes linked to it, computes their semantic similarity measures wrt the target, chooses the node with the highest value as the next one to be visited, and do this recursively until it reaches the target. Other nodes with high similarity values (higher than the threshold η), such as the differentia quality node "that encodes an image digitally", are also explored

Algorithm 2 Semantic Entailment Recognition through the DGN Algorithm

```
1: procedure ProcessEntailment(T, H)
2:     C_T ← CoreWords(T)
3:     C_H ← CoreWords(H)
4:     n ← Size(C_T)
5:     m ← Size(C_H)
6:     k ← Max(n, m)
7:     S ← C_T × C_H
8:     S' ← TopK(Sort(S))
9:     for all {e_i, e_j} ∈ S' do
10:        Add(allPaths, DGN(e_i, e_j))
11:    end for
12:    if allPaths = ∅ then
13:        entailment ← false
14:    else
15:        entailment ← true
16:        bestPath ← ShortestPath(allPaths)
17:        justification ← WriteJustification(bestPath)
18:    end if
19:    return entailment, justification
20: end procedure
```

later, but the path indicated by the thicker lines in the figure is the shortest, and therefore the best one.

In this path, nodes are linked either by the *has_supertype* property, which defines the *kind* of an entity, or by the *has_diff_qual* property, which introduces a *qualifier* for the entity it describes. In the second case, the *supertype* node is also included in the path because *differentia* role nodes (as well as almost all the other role nodes) don't make much sense without the supertype they refer to. Since the justification takes into account the content of the nodes and the relationships between them, that is, the role names (see Chapter 3), the final, human-readable explanation generated by the algorithm from this sequence of nodes is:

```
A digital camera is a kind of camera
A camera is an equipment for taking photographs
Photograph is synonym of picture
```

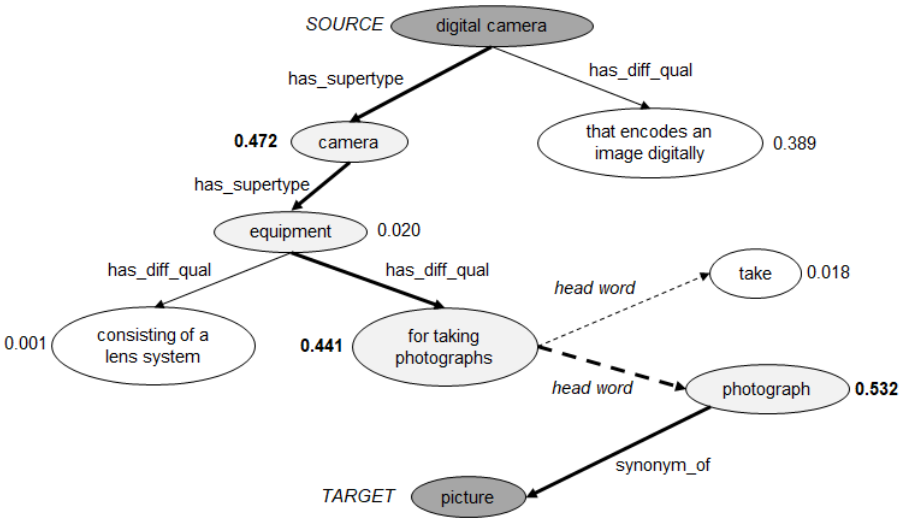

Figure 4.6: A path, indicated by the gray nodes, between source node "digital camera" and target node "picture". Full lines represent actual edges in the graph, while dashed lines represent the algorithm's internal operations, in this case the extraction of head words for multi-word expression nodes. Numbers show the semantic relatedness between each node and the target.

4.2 A Complete Entailment System

The Distributional Graph Navigation model described in Section 4.1 addresses an important gap in textual entailment recognition, going beyond the syntactic and shallow semantic features to recognize and justify entailments that involve more complex semantic relationships. Nevertheless, entailment datasets, as well as real-world data, will include both syntactic and semantic entailment pairs mixed together, and the DGN alone can only do so much. This led to the second phase of the development of the entailment approach proposed in this work, where the DGN model is integrated into a broader system capable of dealing with any entailment pair, be it syntactic or semantic. This system is called *XTE – Explainable Text Entailment* – to emphasize its interpretable nature, since its ability to explain its reasoning process through user-oriented natural language justifications is one of the key characteristics which distinguishes the proposed approach from existing entailment systems. The components of XTE are described next.

4.2.1 Entailment Routing

One of the central points of the proposed composite interpretable text entailment approach is the notion that text entailment can involve syntactic or semantic phenomena, and each of these phenomena categories requires specific approaches to be solved. In the first case, an analysis of the syntactic structure of the sentences may be enough, while in the second it is necessary to identify the semantic relationship holding between the text and the hypothesis. On the other hand, looking for semantic relationships where only a syntactic variation occurs or comparing syntactic structures of very (syntactically) different sentences can be highly counterproductive, hence the importance of choosing the suitable method first and foremost.

To pick the best approach, we need to answer the following question: *Can there be a semantic relationship between T and H?* In Section 4.1, we assumed this relationship existed between a word in T and another word in H. We now formalize this assumption, defining that a semantic relationship must hold between two entities e_1 and e_2, $e_1 \neq e_2$, both referring to a third entity, which we call the *referent*, or r.

The *routing* mechanism that will check these conditions relies on the notion of *overlap* between the text and the hypothesis. The overlap O is computed over the bag-of-words representation of T and H, denoted by:

$$T' = \{t_1, t_2, ..., t_n\} \tag{4.6}$$

where t_i are tokens in T and n is the size of T', and

$$H' = \{h_1, h_2, ..., h_m\} \tag{4.7}$$

where h_i are tokens in H and m is the size of H'. Therefore:

$$O = T' \cap H' = \{w_1, w_2, ..., w_k\} \tag{4.8}$$

where k is the size of O. Formalizing the aforementioned conditions for the existence of a semantic relationship between T and H, we have that:

$$\exists e_1 \in T' \wedge \exists e_2 \in H' \wedge e_1 \neq e_2 \tag{4.9}$$

$$\exists r \in O \tag{4.10}$$

In order to reduce noise, stop words are removed from both T' and H'. After computing O, three scenarios may occur:

(1) *total overlap*, where all the tokens of H' are contained in T' or (less commonly) vice-versa, that is, $k = m$ or $k = n$. In this case, the condition 4.9 is not satisfied;

(2) *partial overlap*, where some but not all of the of tokens of T' are contained in H', so $k < n$ and $k < m$. Both conditions 4.9 and 4.10 are met in this scenario;

(3) *null overlap*, that is, no tokens of T' are contained in H', so $O = \varnothing$ and $k = 0$. Since O is empty, the condition 4.10 can't be satisfied.

Given that we can look for a semantic relationship between T and H solely when both conditions 4.9 and 4.10 are met, the entailment pair will be solved semantically only when a partial overlap occurs. Otherwise, the pair will be solved syntactically because, if there is a total overlap, there are no entities e_1 and e_2 such that $e_1 \neq e_2$ for which a semantic relationship may hold, and, in the case of a null overlap, there is no referent r, so, even if there are some potential candidates e_1 and e_2 that could be semantically related, it is more likely (although not certain) that they are referring to completely different entities.

4.2.2 System Architecture

For solving entailments syntactically, the **Tree Edit Distance** (TED) model is used, and for dealing with entailments involving semantic phenomena the **Distributional Graph Navigation** (DGN) model, described in Section 4.1, is employed. An additional **Context Analysis** module feeds both models with extra information extracted from the entailment pair.

As already discussed in Chapter 2, the Tree Edit Distance model is a straightforward yet efficient text entailment approach which works well when only a syntactic analysis is required for the entailment recognition. As for the DGN model, after some preliminary experiments, which will be detailed later in Chapter 5, some limitations were identified, leading to the development of an improved second version of the algorithm. Both the TED model and the improved DGN model are described in Sections 4.2.3 and 4.2.4, respectively.

The general architecture of the XTE approach is shown in Figure 4.7. The entailment pipeline, which receives as input a T-H pair, starts with a prepro-

cessing stage that generates T' and H'. Next, the router computes O and sends the entailment pair either to the TED or to the DGN model, according to the conditions defined in Section 4.2.1. After the entailment is solved by the suitable model, returning *yes* or *no* as the output, an interpretability module uses the evidence produced by the entailment algorithm to generate a natural language justification explaining the algorithm's decision.

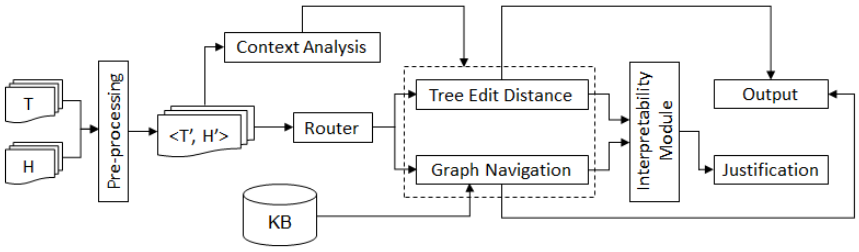

Figure 4.7: General architecture of XTE (Explainable Text Entailment)

4.2.3 The Tree Edit Distance Model

Syntactic entailments are those where T and H express exactly the same information, but in different ways. That means that the information is only *organized* in different ways, and no new knowledge derived from T is expressed in H. Let's consider again the example given in Chapter 1:

T: The badger is burrowing a hole.
H: A hole is being burrowed by the badger.

The bag-of-words (after normalization, stop words already excluded) for both sentences and the resulting overlap set (see Section 4.2.1)is as follows:

$$T' = \{badger, burrow, hole\} \tag{4.11}$$

$$H' = \{hole, burrow, badger\} \tag{4.12}$$

$$O = \{badger, burrow, hole\} \tag{4.13}$$

The total overlap between T and H confirms the absence of semantic relationships (see Condition 4.9 in Section 4.2.1), so their syntactic structures is all that needs to be analyzed in order to detect the entailment. The validity of the entailment in this case can be reduced to a transformation problem, as described in Chapter 2, in which T entails H if the chosen representation of T can be transformed into the representation of H. The Tree Edit Distance (TED) model addresses this transformation problem by comparing the entailment pair's syntactic trees representation (Kouylekov & Magnini, 2005).

The Tree Edit Distance algorithm computes the minimal-cost sequence of operations, namely *insertion*, *deletion* and *replacement* of nodes, necessary to transform one tree into another one. The total cost of the transformation is equal to the sum of the costs of each operation, which in the simplest case is equal to 1. So, for example, for transforming the tree t_1 into the tree t_2 in Figure 4.8, it is necessary to delete the node e, insert the node g and replace the node c by node f, which, considering unitary costs, results in a total cost and, hence, an edit distance of 3. In the text entailment context, the TED task is to compute the cost of transforming the tree representation of T into the tree that represents H. We use the *All Paths Tree Edit Distance* (APTED) (Pawlik & Augsten, 2016) TED implementation, which improves over the classical algorithm of Zhang and Shasha (1989) by being tree-shape independent.

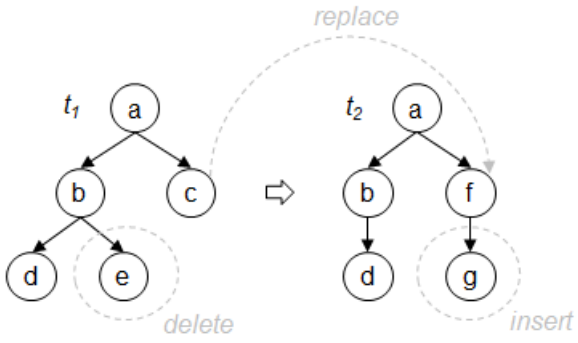

Figure 4.8: The Tree Edit Distance operations.

The edit distance is computed over the syntactic dependency trees of T and H, generated by the Stanford dependency parser (D. Chen & Manning, 2014). This parser generates a dependency graph, but it can be easily converted to an acyclic tree, where nodes with more than one incoming edge are expanded

only at the first time they are referenced, and represented as childless nodes in subsequent references (similar to the pretty-print string representation provided by the parser for the original graph).

Dependencies between terms, which are labeled edges in the original graph, are represented as intermediary nodes between the two nodes they link, that is, the two dependency's arguments. Figure 4.9 shows the graph generated by the dependency parser for the text sentence T in the example above, "The badger is burrowing a hole", and the resulting dependency tree which will be sent as one of the inputs to the TED algorithm.

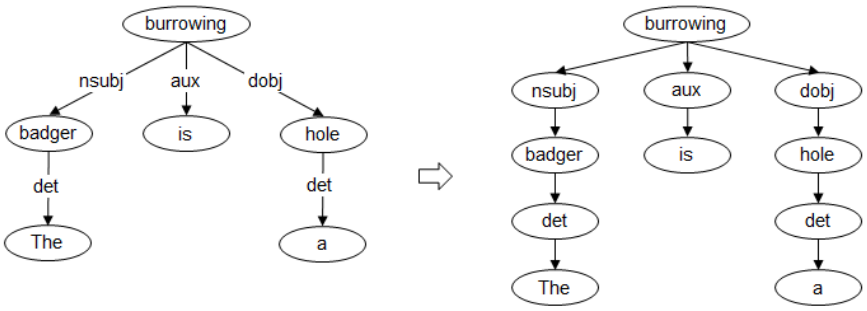

Figure 4.9: Dependency graph (left) and the resulting dependency tree (right) which is sent to the tree edit distance algorithm

Given that dependencies are represented as nodes in the tree, our TED model penalizes node replacement more than insertion and deletion, because replacing a node x between nodes a and b in T by a node y between the same nodes a and b in H means changing the dependency between them, or changing one of the arguments of a dependency, if the replacement comes before or after a sequence of two nodes a and b which are identical in T and H. As an example, consider the following entailment pair from the BPI dataset:

34.100 T: In Pakistan, the Taliban have forbidden women to work.
34.100 H: Women have forbidden the Taliban to work.
34.100 A: NO

The dependency trees for T and H are depicted in Figure 4.10. In both trees, "forbidden" is linked to "Taliban", but having "nsubj" as the intermediary node in T and "dobj" in H , that is, the dependency changed from *subject* in T

to *object* in H. The same happens when nodes "forbidden" and "women" are considered, and such dependency changes (which cause a complete change in meaning) are reflected by node replacements when the edit distance is being computed.

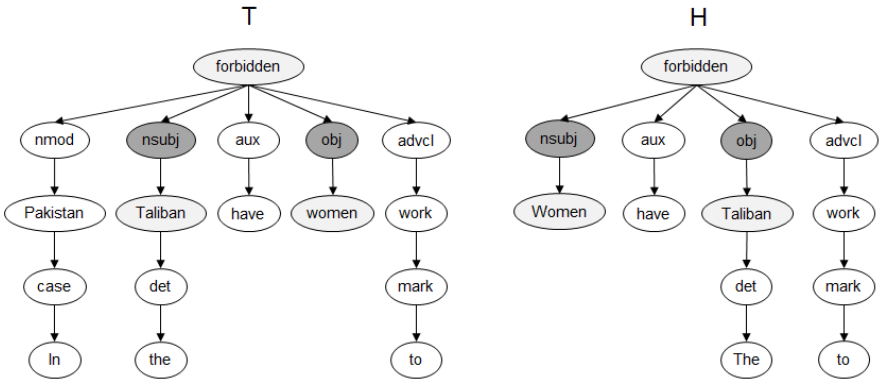

Figure 4.10: An example of node replacement.

The penalization over node replacement is done by a *weighted cost model* with higher weight for replacements than for insertions and deletions, and by the calculation of the relative edit distance *relDist*, which is the edit distance *dist* relative to the difference $diff$ between the sizes of the two trees, given by:

$$relDist = dist/diff \tag{4.14}$$

If the two trees are roughly the same size, but many edit operations are performed, they are probably replacements, which means many dependencies and/or arguments are being changed, so $diff$ is low and $relDist$ increases. On the other hand, if approximately the same number of operations are performed for trees having different sizes (usually, T larger than H), there will be more insertions and/or deletions. In this case, $diff$ is higher and $relDist$ decreases, which favors scenarios where the tree for H is a subtree of the tree for T, and, therefore, insertions/deletions will occur more often and affect the validity of the entailment less than replacements. Consider again the BPI dataset entailment pair 64.2:

64.2 T: Skilling was wearing a security tag on his ankle when he stepped into the street to face the press.
64.2 H: Skilling was wearing a security tag.
64.2 A: YES

Figure 4.11 depicts the trees for T and H. As can be noticed, T's tree is much larger than H's, but the latter is an exact subtree of the former. In order to transform T into H, all the gray nodes in T must be deleted, which results in a high number of operations but, since $diff$ is also high, the $relDist$ between T and H will decrease.

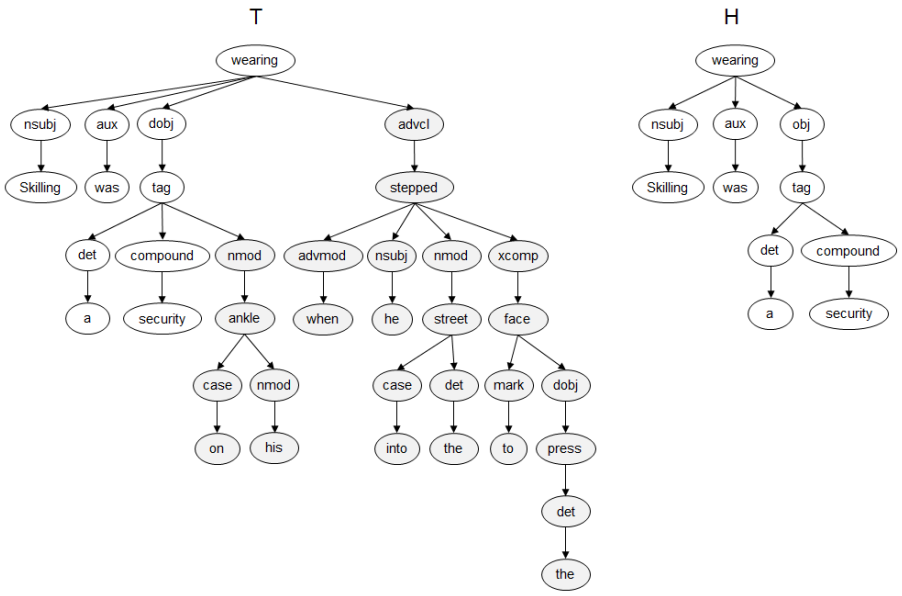

Figure 4.11: An example of node deletion.

The $relDist$ is then compared against a threshold t, and the pair is classified as an entailment if $relDist < t$, and as a non-entailment otherwise.

4.2.4 Improved Distributional Graph Navigation

As mentioned earlier, the goal of the Distributional Graph Navigation algorithm is to find the semantic relationship holding between two entities $e_1 \in T$ and $e_2 \in H$ that confirms and explain the entailment. Therefore, identifying the right pair of entities e_1 and e_2 is crucial for finding the right relationship

between the sentences as a whole. The first implementation of the DGN-based entailment recognition, described in Section 4.1.3, relied on a set of syntactic rules for identifying the *core words* of a sentence, which would later compose the source-target pairs sent as input to the DGN. Although those rules work well for relatively short and structurally simple sentences, as their size and syntactic complexity increases, the rule-based heuristic can fall short of detecting the relevant core words. Consider as an example the following entailment pair from the GHS dataset, which is mostly composed of pairs in which at least T is a very long sentence:

19479 T: Chilling new evidence of the torture and sexual abuse of Iraqi prisoners by American soldiers emerged last night in a secret report accusing the US army leadership of failings at the highest levels.
19479 H: Shock new details of torture by US troops
19479 A: YES

According to the syntactic rule set, the main noun in the subject should be selected as one of the core words, but, in this example, the subject is "Chilling new evidence of the torture and sexual abuse of Iraqi prisoners by American soldiers", a long and complex structure from which it is not straightforward to pick up the main noun. The same happens for the predicate: should only "emerged" be considered the main verb? What would its objects or modifiers be? Sentence simplification also proved to be of limited help in such scenarios, because it also uses information from the sentence's syntactic parse tree and cannot always provide the optimal sentence split, especially when clear structures such as subordinate or coordinate clauses are absent. These issues could, and indeed led to the loss of relevant information during the entailment pair preprocessing.

The conclusions drawn from the analysis of the source-target pair selection algorithm behavior are that, if we want to look for semantic relationships between entities, instead of picking the main syntactic components, we should look for the most semantically related words, using the same semantic relatedness measures already employed by the DGN for the graph search. Therefore, the source-target pairs, that is, the pairs of entities $\{e_i, e_j\}$ which will be sent as input to the DGN are now identified as follows: using the information from the sets T' (Equation 4.6), H' (Equation 4.7) and O (Equation 4.8), we compute the sets T'' and H'', where:

$$T'' = T' - O \tag{4.15}$$

$$H'' = H' - O \tag{4.16}$$

Also using DSMs, we then compute the semantic similarity measures between T'' and H'' as the Cartesian product P:

$$P = T'' \times H'' \tag{4.17}$$

The results are then sorted and the k highest scoring pairs are selected, making up the set P'. Each pair $\{e_i, e_j\} \in P'$ is sent to the DGN algorithm, which returns a set of paths between e_i and e_j. The new version of the procedure for recognizing and justifying an entailment through the DGN is listed in Algorithm 3. The differences regarding Algorithm 2 lie in lines 2 through 8, which deal with the entailment pair preprocessing for source-target pair selection. Lines 9 through 19, where the entailment recognition itself is performed, remain unchanged.

Algorithm 3 Semantic Entailment Recognition through the DGN Algorithm V2

```
1:  procedure ProcessEntailment(T, H)
2:      T' ← Tokenize(T)
3:      H' ← Tokenize(H)
4:      O ← T' ∩ H'
5:      T'' ← T' − O
6:      H'' ← H' − O
7:      P ← T'' × H''
8:      P' ← TopK(Sort(P))
9:      for all {e_i, e_j} ∈ P' do
10:         Add(allPaths, DGN(e_i, e_j))
11:     end for
12:     if allPaths = ∅ then
13:         entailment ← false
14:     else
15:         entailment ← true
16:         bestPath ← ShortestPath(allPaths)
17:         justification ← WriteJustification(bestPath)
18:     end if
19:     return entailment, justification
20: end procedure
```

The same semantic similarity-driven approach was adopted for the identification of *head words* too. As described in Section 4.1.2, the head words are the words present in a role node retrieved from a DKG and that define the next nodes to be visited when the DGN is searching the graph. These words were also selected through a set of syntactic rules in the first implementation and were changed in the second version as well in order to increase the relevance of the DGN inputs also during the graph search.

For getting the head words now, first all stop words and words with low inverse document frequency (IDF) are removed. Again, IDF is calculated using as the corpus the same linguistic resource that gave origin to the knowledge graph being explored by the algorithm. After the irrelevant words are removed, the semantic similarity sr between each remaining word and the target word T is computed, the results are sorted and only the top k words are kept. As before, the highest scoring head word will be the next node to be visited (line 37 in Algorithm 1), and all the other head words are added to a copy of the current path, generating a new path which will be pushed to the stack to be explored later (lines 32 through 36 in Algorithm 1).

Another small improvement, but which has a significant impact on the DGN recall, is the introduction of synonym comparison against the target word. In the DGN first implementation, the search stops successfully when the next node to be visited is equal to the target (line 39 in Algorithm 1). In the second version, the successful stop is also reached when the next node is one of the target's synonyms . This is done with the aid of a synonym table, built from synonym lists gathered across all the tested lexical resources (see Chapter 5) and other online resources[3].

Finally, restricting the cases where the Distributional Graph Navigation is applied, although being a system-wide feature rather than a DGN improvement, is a refinement worth mentioning for having a positive effect on the algorithm's precision. The first implementation only had as a requirement the existence of two entities $e_1 \in T$ and $e_2 \in H$, $e_1 \neq e_2$. By adding the existence of a common referent as a mandatory requirement (Condition 4.10), it is possible to reduce the number of the DGN model's wrong decisions. The following example from the SICK dataset[4] illustrates a likely misclassification scenario:

[3] https://bit.ly/2VPkywz, https://bit.ly/2kimBWS
[4] http://clic.cimec.unitn.it/composes/sick.html

8322 T: A horse is racing
8322 H: Dogs are running on a track
8322 A: NO

The bag-of-words and overlap sets for this examples are:

$$T' = \{horse, race\} \tag{4.18}$$

$$H' = \{dog, run, track\} \tag{4.19}$$

$$O = \varnothing \tag{4.20}$$

If the null overlap set, and, therefore, the absence of a referent is ignored, the DGN could easily find a relationship between e_1 = "race" and e_2 = "run", since these two entities are very semantically related. As it is easily identifiable for a human, the entailment is no true because "race" refers to "horse" and "run" refers to "dogs"; the contents of the overlap set help to provide the system with further hints, especially when additional context information, which will be detailed later in Section 4.2.5, is not available.

The following final example illustrates a scenario where the improved version of the DGN performs better than the first implementation, in this case by eliminating ambiguity in the selection of head words:

47.4 T: Iran is a signatory to the Chemical Weapons Convention.
47.4 H: The Chemical Weapons Convention is an agreement.
47.7 A: YES

In this example (from the BPI dataset), the best source-target pair is e_1 = "signatory" and e_2 = "agreement", and the referent r = "Chemical Weapons Convention", since both e_1 and e_2 refer to this concept. The best path between the source and the target in a DKG, as well as all the semantic similarity measures between each node retrieved by the algorithm and the target, are shown in Figure 4.12.

Note that the differentia quality node labeled "of ownership or obligation" has two nouns and, following a purely syntactic-based extraction as it was done in the first DGN version, it would be more challenging to identify the main noun

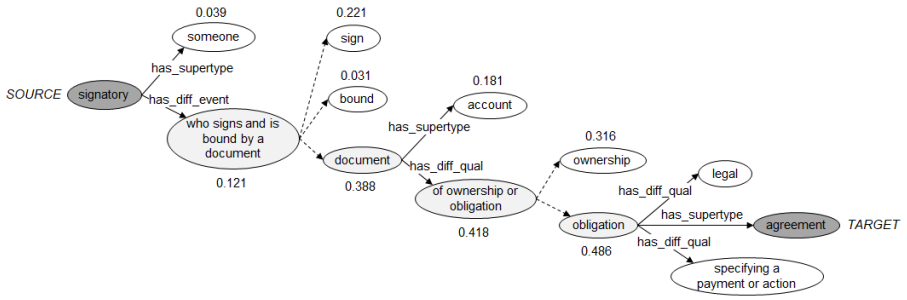

Figure 4.12: A path, indicated by the gray nodes, between source node "signatory" and target node "agreement" in a DKG. Full lines represent actual edges in the graph, while dashed lines represent the algorithm's internal operations, in this case the extraction of head words for multi-word expression nodes. Numbers show the semantic relatedness between each node and the target

(in the sense of the most relevant one). By getting all the words and computing their semantic relatedness wrt the target, the DGN can naturally choose the best option, reducing the risk of information loss. The justification generated from this path is as follows:

```
A signatory is someone who signs and is bound by a document
A document is an account of ownership or obligation
An obligation is a kind of agreement
```

The improvements introduced in the DGN model are a natural consequence of the phased approach development, which allowed an incremental evolution, but are also intended to emphasize the importance of focusing on the *meaning* of words across all the steps of the semantic entailment recognition process. A better comparison between the two versions of the DGN in quantitative terms will be given in Chapter 5.

4.2.5 Context Analysis

The goal of the Context Analysis module is to provide both the Tree Edit Distance and the Distributional Graph Navigation models with information they can't easily grasp, and that, if missed, can lead to erroneous conclusions. In the Tree Edit Distance case, this can happen when minimal, slight modifications completely change the meaning of a sentence. Such modifications may yield

only a small edit distance between T and H, resulting in a wrong *entailment* classification for the pair.

For semantic entailments, the extraction of further contextual information is even more critical, since the Distributional Graph Navigation model, though looking for common referents, primarily considers pairs of terms in isolation and not the sentences as a whole. This means that even when there are two entities $e_1 \in T'$ and $e_2 \in H'$ with a high semantic relatedness score, there may also be, at another point in the sentences, contradictory or inconsistent information which invalidates the entailment, but that the DGN won't catch.

The Context Analysis module receives as input the tokenized output from the preprocessing stage, discards the overlapping entities in the set O, and, using syntactic and semantic features, analyzes the remaining words/phrases in order to look for the following phenomena[5]:

Simple Negation: T is a simple negation of H. Example:

1127 T: A sea turtle is not hunting for fish
1127 H: A sea turtle is hunting for fish
1127 A: NO

Negation adverbs will mostly be considered as stop words, and, therefore, will not be included in the preprocessed T' and H' representations, so entailment pairs where these are the only divergent words will be sent to the Tree Edit Distance model. Detecting the negation allows the model to classify the pair as a non-entailment even if the final edit distance is well below the threshold.

Opposition: T contains a term which is an antonym of a term in H. Example:

3706 T: A woman is taking off eyeshadow
3706 H: A woman is putting make-up on
3706 A: NO

Detecting opposition is an important step when entailments are solved through the Distributional Graph Navigation model. In the above example, the DGN would detect "eyeshadow" in T and "make-up" in H as a candidate source-

[5]All examples from the SICK dataset

target pair and most likely find a path in a DKG linking both entities (since "eyeshadow" is a kind of "make-up"), but the presence of antonym terms "take off" in T and "put" in H prevents the pair from being misclassified as an entailment. Opposition detection is performed with the aid of an antonym table, built from antonym lists extracted from all the tested lexical resources (see Chapter 5) and other online resources[6].

Inverse Specialization: H specializes some information in T. Since text entailment is a directional relationship from T to H, specializations are valid only in this direction, not the other way around. Example:

1382 T: A person is rinsing a steak with water
1382 H: A man is rinsing a large piece of meat
1382 A: NO

In this example, H specializes T since "person" is more general than "man". As much as opposition detection, inverse specialization detection plays an important part in preventing the DGN from misclassifying the entailment pair (in the above example, by finding a relationship between "steak" and "meat"). In the correct direction, that is, from T to H, specializations can be easily detected also by the DGN model, since they are a kind of semantic relationship; for detecting only inverse specializations the hypernym links from WordNet are used.

Unsatisfiable Clauses: H has more information than what can be satisfied by T. Example:

6296 T: A large group of cheerleaders is walking in a parade
6296 H: The cheerleaders are parading and wearing black, pink and white uniforms
6296 A: NO

Coordinated or subordinated clauses in H can be unsatisfiable if T has fewer clauses that H. In the above example, T is composed by a single clause while H has two coordinated clauses and, although the first H's clause can be fully entailed by T, the second one cannot be satisfied. Mismatching number of clauses

[6]https://bit.ly/2Uy3eMf, https://bit.ly/2J6N1wd, https://bit.ly/2HcDK3W, https://bit.ly/2O0GCBs, https://bit.ly/2Tv7UWN, https://bit.ly/2XSm5DN

between T and H is detected through the analysis of the sentences' syntactic parse trees.

Any of the above-described phenomena is considered enough to reject the entailment, so the TED and DGN models always take into account the output of the Context Analysis module and, in case their conclusions diverge, the decision made on the basis of the contextual information prevails.

4.3 Summary

In this chapter, the composite interpretable text entailment approach proposed in this work was described. This approach develops around two main points: (i) that text entailment can involve a wide range of different linguistic and semantic phenomena, and identifying such phenomena and using the most suitable techniques for each of them is key to better accuracy, and (ii) that, by exploring an external knowledge base when dealing with semantic entailments, it is possible to render the system interpretable, generating human-readable justifications which show explicitly what the semantic relationship holding between the text and the hypothesis is.

The development was divided into two phases: first, the Distributional Graph Navigation, an approach for dealing with semantic entailments, was implemented. Focusing on text entailments that require reasoning over world knowledge, this model employs Distributional Semantic Models (DSMs) for computing the semantic relatedness between words, and uses these measures as a parameter for navigating a Definition Knowledge Graph (DKG) and finding the relationship between T and H. The path in the DKG defining this relationship is then formatted into a natural language justification, explaining the entailment decision. Second, a complete entailment system was developed, integrating the DGN with other modules aimed at deciding the best method to be used, addressing syntactic entailments, and extracting additional context information from the entailment pair.

The architecture of the complete entailment system, called XTE – Explainable Text Entailment – was presented and each of its components was described. The conditions that guide the routing mechanism, which analyzes the overlap between T and H and decides whether entailment pairs should be dealt with syntactically or semantically, were introduced and formalized. The Relative

Tree Edit Distance model, which deals with those pairs predominantly showing structural – that is, syntactic – differences, was described, and the improvements introduced in the DGN model, responsible for solving entailments where a semantic relationship exists, were listed and detailed. Such improvements were intended to address some limitations observed by the end of the first phase of the approach development, leveraging the capabilities of the graph navigation mechanism for privileging the meaning of words and the semantic relatedness between them across all the entailment recognition steps, from preprocessing to justification.

Finally, the Context Analysis module, a support component which provides additional information to both the TED and DGN models, was described. This module analyzes syntactic and semantic features from both T and H to detect phenomena not easily caught by the other models, such as simple negation, opposition, inverse specialization, and clause unsatisfiability, helping the system to deliver a better informed entailment decision.

The approach development and the composite interpretable system generated as a result aimed at addressing all the research hypothesis. The focus was on the importance of the injection of commonsense world knowledge for solving more semantically complex entailments, the need for the distinction between syntactic and semantic entailments and the use of different methods to tackle them, and the introduction of an interpretability feature, providing natural language, human-like justifications for the entailment decision. The last point is a major contribution over existing entailment systems: by explaining the system's reasoning steps, it becomes possible to interpret and understand its underlying inference model, taking the entailment decision out of the numerical score black box.

Chapter 5

Evaluation

In this chapter, the experiments carried out for evaluating the proposed approach are presented. In designing the experiments, we sought to give prominence to datasets where world knowledge plays an important part in the entailment recognition, in line with the proposed approach emphasis on semantics and commonsense knowledge. Nevertheless, we also deemed important to include varied types of data, represented not only by the datasets tested but also by the knowledge bases employed, so it could be possible to assess the system behavior in as many different scenarios as possible.

As described in Chapter 4, the development of the approach was divided into two parts: first, an algorithm for solving semantic entailments was designed and implemented, and then a complete entailment system, able to deal with both syntactic and semantic phenomena, was developed. The experiments reflect the flow of activities in the development and aim at both evaluating individually the products of both phases, and showing the evolution and quantitative and qualitative gains from the first solution to the final, complete approach. Therefore, the goals of the experiments are to:

- Evaluate how an entailment model focused on finding semantic relationships between T and H with the aid of a commonsense knowledge base performs when compared to transformation and classification algorithms when dealing with more world knowledge-demanding datasets;

- Evaluate how an entailment system that can recognize the T-H pair's predominant phenomena and employ the most suitable method to solve

each of them compares to approaches that use a single technique, be it syntactic or semantic, for all pairs;

- Evaluate the interpretability dimension of the system, assessing its ability to explain its decisions and the quality of the justifications generated; and

- Evaluate quantitatively and qualitatively how different knowledge bases generated from distinct lexical resources compare, especially from the interpretability point of view, showing their impact in both the entailment recognition and in the generation of justifications.

Throughout this chapter, the first part of the experiments, where the DGN model alone was assessed, is referred to as *stage one*, and the evaluation of XTE, the complete entailment system, is referred to as *stage two*. We start with an account of the experimental setup, detailing the definition of the system's main parameters and describing the resources employed, comprising knowledge bases, datasets, and baselines. Next, the results of both phases are presented and analyzed, including a discussion on the influences of the different characteristics of each knowledge graph on the system accuracy and interpretability.

5.1 Datasets

The following text entailment datasets were tested throughout the execution of the experiments:

RTE3 dataset: the dataset from the third RTE Challenge[1] is one of the most traditional and popular text entailment datasets. It contains 1,600 T-H pairs, split into DEV (800 pairs) and TEST (800 pairs) sets, and is balanced, with half positive and half negative examples.

SICK dataset: SICK[2] (*Sentences Involving Compositional Knowledge*) is a dataset aimed at the evaluation of compositional distributional semantic models, which, besides the semantic relatedness between sentences, also includes annotations about the entailment relation for the sentence pairs (Marelli et al., 2014). It is composed of 9,840 pairs, split into TRAIN (4,439 pairs), TRIAL

[1] https://www.k4all.org/project/third-recognising-textual-entailment-challenge/
[2] http://clic.cimec.unitn.it/composes/sick.html

(495 pairs), and TEST (4,906 pairs). Instead of the binary entailment classification, there are three different relations: *entailment*, *contradiction*, and *neutral*. For coherence with the other datasets, we considered both the *contradiction* and *neutral* labels as non-entailment, leading to 29% positive and 71% negative examples (the original classification is also unbalanced: around 57% of the pairs have the label *neutral*).

BPI dataset: The *Boeing-Princeton-ISI*[3] textual entailment test suite was developed specifically to look at entailment problems requiring world knowledge, being syntactically simpler than RTE datasets but more challenging from the semantic viewpoint. It is composed of 250 pairs, 50% positive and 50% negative.

GHS dataset: The *Guardian Headlines Sample*[4] is a subset of the Guardian Headlines dataset[5], a set of 32,000 entailment pairs automatically extracted from The Guardian newspaper but not validated. The GHS is a random sample of 800 pairs which have been manually curated, leading to a balanced set of 400 positive and 400 negative examples. It also requires a reasonable amount of world knowledge and is the only dataset fully composed of real-world data, without artificially assembled hypotheses: in positive examples, T is the first sentence of a story and H is its headline, and in negative examples T and H are two random sentences from the same story.

The RTE3 and SICK datasets have a stronger emphasis on linguistic phenomena, while BPI and GHS gather more semantic-driven entailments, demanding more world knowledge in the entailment recognition process. Since at stage one the focus of the evaluation was on semantic entailments, only BPI and GHS datasets were tested at this point. For evaluating the complete system, at stage two all four datasets were used in the experiments.

5.2 Knowledge Bases

To evaluate the impact of different lexical resources in the entailment results, especially in the justifications generated, the definitions from four dictionaries

[3] http://www.cs.utexas.edu/users/pclark/bpi-test-suite/
[4] https://goo.gl/4iHdbX
[5] https://goo.gl/XrEwG9

were extracted: **WordNet**, the **Webster's Unabridged Dictionary**[6], **Wiktionary**[7], and the set of definitions extracted from **Wikipedia** pages provided by Faralli and Navigli (2013). Each of the four resulting DKGs differs from the others in some way: The Webster's is an older, conventional dictionary dating from 1913. WordNet and Wiktionary are modern on-line lexicons, but the former is developed by professional lexicographers while the latter is built collaboratively by lay users. Last, Wikipedia is also built collaboratively, but is an encyclopedic, rather than lexical, resource.

The original Webster's dictionary text file was processed so, besides the definitions, the part-of-speech and list of synonyms (when available) for each word could also be extracted[8]. For the Wikipedia dataset, definitions for named entities were excluded with the aid of the Stanford Named Entity Recognizer (NER), so the final content could be closer to a regular dictionary. Due to the natural limitations of the NER, many named entity definitions remained in the final set, but this additional filter helped to set a manageable size for the final graph, without leaving out potentially relevant information.

All the four sets of definitions were filtered, labeled and converted to an RDF graph[9], following the knowledge graph construction methodology described in Chapter 3, yielding four different graph knowledge bases[10]. Table 5.1 shows the dimensions of each of the resulting graphs.

Resource	Noun Definitions	Verb Definitions	Total
WordNet	79,939	13,760	93,699
Webster's	88,620	25,290	113,910
Wiktionary	390,417	73,826	464,243
Wikipedia	859,087	-	859,087

Table 5.1: Final dimensions of the definition knowledge graphs used in the experiments

As much as the text entailment approach development, the creation of the knowledge graphs was also carried on in an incremental manner. WordNet, which is also used as a reference dictionary as reported in Chapter 3, was the first lexicon to be processed, enabling the realization of the first experiments. Hence, at stage one, only the WordNet graph was used as a knowledge source

[6] http://www.gutenberg.org/ebooks/673
[7] https://www.wiktionary.org/
[8] Extraction in JSON format available at https://github.com/ssvivian/WebstersDictionary
[9] Tools for building the graph available at https://github.com/ssvivian/DefRelExtractor
[10] All knowledge graphs in RDF format available at http://tiny.cc/gvtadz

for the DGN model. At stage two, which aimed at not only evaluating the performance of XTE but also comparing the influence of the knowledge bases on the system's results, all four graphs were employed.

5.3 Computing the Thresholds

Two of the most important parameters of the proposed approach are the Tree Edit Distance model's threshold t and the Distributional Graph Navigation algorithm's semantic relatedness threshold η (see Chapter 4). The definition of t was only necessary at stage two, where the TED model was introduced, while η is used at both stages one and two.

The TED threshold t is computed previously through a training procedure which performs a sequential search to look for the distance that better separates positive examples from the negative ones, and is aimed at maximizing the algorithm's accuracy, in our case, the F1-score. For training the model, the training portions of the RTE3 and SICK datasets were combined. This combined dataset, herein called RTE+SICK train dataset, compensates for the lack of training data in the other two datasets while still being representative of their syntactic characteristics: the RTE3 is closer in format to the GHS, both having very long text sentences and usually short hypothesis, while the SICK data is more similar to the BPI entries, with both datasets having short to medium-sized text and hypothesis sentences, and usually not a big difference in size between the two sentences composing an entailment pair. After the training is performed over the RTE+SICK dataset, the learned threshold t is used to compute the syntactic entailments for all the four tested datasets (for the RTE3 and SICK datasets, the evaluation is performed on their test portions).

The DGN threshold η is computed dynamically so the algorithm can always retrieve the highest scoring entries from a list of candidates. While navigating the knowledge graph, the DGN always retrieves a set of nodes and computes the semantic relatedness sr between each node and the target (see Chapter 4), which results in a ranked list of scores. Over this list, we perform a *Semantic Differential Analysis*, adapting the method proposed in (Freitas, Curry, & O'Riain, 2012) to identify score gaps which discriminate between highly semantically related nodes and non-related ones. Given a list of ranked nodes, S_0 is the score for the node with the maximum relatedness value, S_k is the score

for the $k+1$ ranked node and $\delta S_{k,k+1}$ is the semantic differential between two adjacent ranked nodes, that is:

$$\delta S_{k,k+1} = S_k - S_{k+1} \tag{5.1}$$

The gap in the list, occurring between S_n and S_{n+1}, is given by δS_{max}, the maximum semantic differential. S_n and S_{n+1} define the top and bottom relatedness values of δS_{max}, denoted by $S_n^{\top}$ and $S_{n+1}^{\perp}$. Figure 5.1 illustrates the Semantic Differential Model.

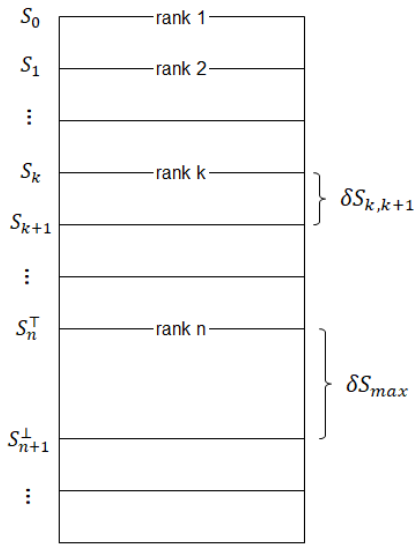

Figure 5.1: The Semantic Differential Model. δS_{max} defines a gap in a ranked list of scores.

To determine η at each step of the graph navigation, δS_{max} is computed over the current ranked list of nodes and the bottom value is selected as the semantic threshold, therefore:

$$\eta = S_{n+1}^{\perp} \tag{5.2}$$

We choose the bottom value, and not the top, which is the one immediately before the gap, in order to keep at least one moderately related node in the list. Since semantic similarity scores depend on the DSM model used to compute them, and, by extension, on the corpus from where the DSM was learned,

average (immediately after the gap) scores can have varying meanings when considered in different contexts. By including the bottom relatedness value in the list, we ensure potential relevant nodes won't be missed. If such nodes prove to be irrelevant, the DGN manages to abort their paths at subsequent steps, eliminating any eventual noise.

5.4 Baselines

To show the improvements provided by both the DGN model alone and, subsequently, the composite XTE system, the results of both experiment stages were compared with two well-established entailment approaches: a transformation-based purely syntactic algorithm and a classification-based syntactic approach that employs linguistic resources from where shallow semantic information is extracted.

The **Edit Distance** (Kouylekov & Magnini, 2005) is the state-of-the-art implementation of the tree edit distance algorithm for recognizing textual entailment, and only considers the syntactic structures of T and H, given by their dependency trees. This approach implements the Zhang and Shasha (1989) tree edit distance algorithm and adopts a cost function based on the *weight*, given by the IDF (inverse document frequency), of the words representing the nodes to be inserted, deleted or replaced.

The **Maximum Entropy Classifier** (R. Wang & Neumann, 2008b) also uses the syntactic dependency trees as features in a classifier which also employs lexical-semantic features from WordNet and VerbOcean (different configurations are available, the *Base+WN+TP+TPPos+TS_EN* configuration, reported to be the one that yields the best results, was used in the experiments). In this approach, structural features are extracted from the tree representations to feed a subsequence-kernel-based classifier. Further features feed individual modules designed to analyze only a specific aspect, such as named entities or temporal expressions. Each module returns its own decision, and the set of decisions are ranked according to their confidence scores so the final decision, which is the highest ranking one, can be chosen.

Both implementations are provided by the text entailment framework *Excitement Open Platform* (EOP) (Magnini et al., 2014). At stage one, where only BPI and GHS datasets were used, both models were trained on the default

RTE3 training set, following the EOP documentation instructions[11]. At stage two, for consistency with the syntactic-driven TED module in XTE, they were also trained on the combined RTE+SICK training dataset (Section 5.3).

Given that one of the main goals of the experiments was to compare an approach supported by a composition of techniques with single-technique algorithms, at stage two the first version of the DGN was also added as a baseline in the XTE evaluation.

5.5 Additional Settings

As detailed in Chapter 4, the first version of the DGN employed syntactic sentence simplification in the entailment pair preprocessing. This was done through the *Sentence Simplification* service (Niklaus, Bermeitinger, Handschuh, & Freitas, 2016) in the information extraction pipeline Graphene[12].

For computing the semantic relatedness between words/phrases, word2vector (Mikolov, Sutskever, Chen, Corrado, & Dean, 2013) was used as the DSM. The measures were obtained through the *Indra*[13] (Sales et al., 2018) service, using the w2v model pre-trained on the Wikipedia 2018 English dump, and the cosine as the score function.

5.6 Results

In all the experiments, we compute the precision, recall, and F1-score (rounded to two decimal places). Results for the Edit Distance and Maximum Entropy Classifier baselines vary from the first to the second stage due to the different training datasets used to generate the model, as described in Section 5.4. The discussions that follow the results are intended to provide a comparison between the proposed approach and the baselines; a detailed analysis of the approach outputs through a review of the system's misclassifications will be presented in Chapter 6.

[11] https://github.com/hltfbk/EOP-1.2.3/wiki/Quick-Start
[12] https://github.com/Lambda-3/Graphene
[13] https://github.com/Lambda-3/Indra

5.6.1 Stage One: Stand-alone DGN

Table 5.2 shows the results of the first round of experiments, where the stand-alone version of the DGN was evaluated and compared against the syntactic-driven baselines.

	BPI			**GHS**		
	Pr	***Re***	***F1***	***Pr***	***Re***	***F1***
EditDistance	0.44	0.65	0.53	0.96	0.30	0.45
MaxEntClassifier	0.46	0.57	0.51	0.50	1.00	**0.66**
DGN	0.65	0.54	**0.59**	0.56	0.50	0.53

Table 5.2: Stage one evaluation results. The upper part shows the baselines, and at the bottom are the stand-alone DGN results.

As can be noticed, the first version of the DGN presents better precision an F1-score for the BPI dataset, which is the most semantically-oriented one, favoring the world knowledge exploration. Nevertheless, it still contains pairs where only a syntactic analysis is necessary, and which the DGN can't deal with, hence the lowest recall.

The GHS is a challenging dataset since it contains longer and more complex sentences, and frequently shows substantial vocabulary variation between text (the first line of a story) and hypothesis (the story's headline), given that journalists tend to avoid repetition of words. The Edit Distance algorithm shows high precision on this dataset, possibly because it has no "tricky" negative examples from the syntactic point of view (i.e., examples intended to exploit the known weaknesses of popular entailment algorithms), as BPI does, but presents very low recall. The Maximum Entropy Classifier shows higher F1-score, but it classifies all but two of the 800 pairs as entailment, hence the 100% recall and 50% precision, since the dataset is balanced. Given that positive pairs are structurally very different from negative ones in this dataset (in positive pairs, H is a short piece of text and T is usually a very long sentence, and in negative pairs both T and H are often long sentences) it is somewhat hard to grasp the Maximum Entropy Classifier's operating principles behind those decisions.

Although these first results are only comparable to those yielded by the baselines, they pointed to the potential of exploring the semantic relationships between T and H, besides providing insights into the DGN strengths and weaknesses so a second improved version could be planned and implemented, before being integrated into the final complete solution.

5.6.2 Stage Two: The Complete XTE System

Table 5.3 shows the results of the second round of experiments, where the composite interpretable XTE system[14] was compared with the single-technique approaches. The different XTE configurations are identified by the DKG used by the Distributional Graph Navigation component, that is, the knowledge bases derived from WordNet (WN), Webster's dictionary (WBT), Wiktionary (WKT), and Wikipedia (WKP).

	RTE3			**SICK**			**BPI**			**GHS**		
	Pr	***Re***	***F1***	***Pr***	***Re***	***F1***	***Pr***	***Re***	***F1***	***Pr***	***Re***	***F1***
ED	0.61	0.51	0.55	0.41	0.76	0.54	0.41	0.45	0.43	0.97	0.15	0.26
MEC	0.56	0.58	0.57	0.65	0.47	0.55	0.29	0.18	0.23	0.99	0.20	0.33
DGN	0.48	0.32	0.38	0.31	0.35	0.33	0.65	0.54	0.59	0.56	0.50	0.53
XTE (WN)	0.57	0.68	0.62	0.50	0.70	0.58	0.56	0.78	**0.65**	0.70	0.53	**0.60**
XTE (WBT)	0.58	0.65	0.61	0.51	0.69	**0.59**	0.53	0.62	0.57	0.69	0.46	0.55
XTE (WKT)	0.59	0.67	**0.63**	0.51	0.70	**0.59**	0.54	0.70	0.61	0.70	0.45	0.55
XTE (WKP)	0.62	0.52	0.56	0.58	0.55	0.57	0.50	0.41	0.45	0.74	0.27	0.40

Table 5.3: Stage two evaluation results. The upper part shows the baselines, and at the bottom are the proposed composite entailment approach results. ED = Edit Distance, MEC = Maximum Entropy Classifier, DGN = Distributional Graph Navigation, XTE = Explainable Text Entailment

The first thing that can be noticed is how the results vary across datasets for those baselines that rely on training data, that is, the Edit Distance and the Maximum Entropy Classifier algorithms. Both approaches present homogeneous results for the RTE3 and SICK datasets, for which training data is available, but their accuracy falls significantly for the BPI and GHS datasets. Their accuracy regarding these two datasets also drops considerably when compared with the results in Table 5.2, where both models were trained on the default training set, indicating a dependency on the kind of data they were designed and developed upon.

Meanwhile, XTE presents consistent results across all datasets, because it does not rely exclusively on the syntactic information learned at the training phase, but rather balances it with the semantic knowledge extracted from the DKGs. Since these knowledge graphs are commonsense and independent resources, they work homogeneously in an unbiased fashion for any unseen data, making the proposed approach less training data-dependent. Moreover, for the BPI and GHS datasets, which require more world knowledge, both baseline algorithms show low recall and are outperformed by XTE, adding to the im-

[14]Source code available at https://github.com/ssvivian/XTE

portance of combining and balancing syntactic and semantic information while solving the entailments.

When compared with the semantic-only stand-alone Distributional Graph Navigation approach, XTE also presents much better results, especially for the RTE3 and SICK datasets, which don't have a heavy focus on world knowledge-based entailments. Besides not dealing well with entailments that don't show any semantic relationship (that is, purely syntactic entailments), as described in Chapter 4, the DGN also uses a syntax-based heuristic to define the source-target input pairs and the head words for multi-word expression graph nodes. This heuristic uses part-of-speech tags to find the main components (subject, verb, objects) in a sentence, which does not work well for long, complex sentences, as is the case in the RTE3 and GHS datasets. By evolving to a semantic similarity-based heuristic, the second version of the DGN, and, consequently, the XTE, can now retrieve better source-target pairs and a higher number of relevant head words, what leads to much better recall and overall F1-score.

5.7 Justification Analysis

The justifications generated for the positive entailments solved by the Distributional Graph Navigation model were systematically analyzed in order to asses their correctness and consistency. This analysis was performed over the outputs of stage two of the experiments, that is, the complete XTE system.

This evaluation was intended to assess the explanations from a *functional* point of view, that is, to determine if they were accomplishing the task of establishing the right relationship between the right terms in T and H. A deeper, psychological evaluation to assess trustworthiness was out of the scope of this work. That means justifications were evaluated to be "right" or "wrong" on a high level, but not "good" or "bad" according to a more subjective user's judgment.

Evaluators were asked to first point the entities establishing a connection between T and H. Then, they should judge if the justification met two requirements: (1) it linked the previously identified entities, and (2) the relationship it describes is the same one intended by the context given by T and H (according to the human judgment).

They then classified justifications into *correct* or *incorrect*. Correct justifications meet both conditions, establishing the right relationship between the

relevant entities $e_1 \in T$ and $e_2 \in H$, presenting the pertinent information about it and making the reasoning clear. Examples[15]:

1.4 T: A council worker cleans up after Tuesday's violence in Budapest.
1.4 H: There was damage in Budapest.
1.4 A: YES

Entailment: yes
Justification:
A **violence** is a state resulting in injuries and **destruction** etc.
A **destruction** is a termination of something by causing so much **damage** to it that it cannot be repaired or no longer exists

116 T: Vasquez Rocks Natural Area Park is a northern Los Angeles County park acquired by LA County government in the 1970s.
116 H: The Vasquez Rocks Natural Area Park is a property of the LA County government.
116 A: YES

Entailment: yes
Justification:
To **acquire** is to come into the **possession**
A **possession** is an act of having and controlling **property**

Incorrect justifications do not meet one or both aforementioned conditions, establishing a relationship between the wrong pair of entities, that is, entities that, although being semantically related, do not establish a logical link between T and H, or being too vague, linking the correct pair of entities $e_1 \in T$ and $e_2 \in H$ but giving only superficial and insufficient information about their semantic relationship. Examples[16]:

149 T: Joining Pinkerton at the Chamber of Commerce, is Lindsey Beverly, who will be the new executive assistant.
149 H: Pinkerton works with Beverly.
149 A: YES

Entailment: yes

[15] Examples from the BPI and RTE3 datasets, respectively.
[16] Examples from the RTE3 dataset.

Justification:
An **assistant** is a person who contributes to the fulfillment of a need or furtherance of an **effort** or purpose
Effort is synonym of **work**

110 T: Leloir was promptly given the Premio de la Sociedad Científica Argentina, one of few to receive such a prize in a country in which he was a foreigner.
110 H: Leloir won the Premio de la Sociedad Científica Argentina.
110 A: YES

Entailment: yes
Justification:
To **receive** is a way of to **take**
To **take** is synonym of to **win**

In the first example, even though the semantic relationship between "assistant" and "work" may seem consistent, the expressions that establish the entailment relation are "join" and "work with". In the second case, although the justification links the correct entities establishing the entailment, that is, "receive" and "win", the explanation, made through the verb "take" with no complements, sounds vague and not informative enough. Definitions for verbs tend to be less detailed than those for nouns, and many times expressed in terms of very broad supertypes (like "take" in the second example), leading justifications generated from paths containing verb entity nodes to be more prone to vagueness.

The distribution of correct and incorrect justifications is given in Table 5.4. The evaluation was performed over the full results obtained with the WordNet graph, which is the knowledge base that yields the overall best results across datasets. A more detailed comparison of all the tested DKGs is given in Section 5.8.

Dataset	**Correct Justifications**	**Incorrect Justifications**
RTE3	50.3%	49.7%
SICK	77.1%	22.9%
BPI	61.3%	38.7%
GHS	43.3%	56.7%

Table 5.4: Distribution of correct and incorrect justifications

As can be seen in Table 5.4, the distribution of correct and incorrect justifications varies depending on the dataset, with SICK and BPI showing the best results. In common, these two datasets have relatively short text and hypothesis sentences, which favors the correct identification of source-target word pairs. On the other hand, the RTE3 and GHS datasets have short hypothesis sentences but very long text sentences. The larger the sentence, the bigger the number of possible source-target pairs, so the likelihood of finding a relationship that doesn't necessarily lead to the entailment increases. The GHS is a particularly challenging dataset because the text T corresponds to the first sentence of a news story, usually expanding the idea highly condensed in the headline, that is, the hypothesis H. The two sentences are very semantically related as a whole, so many concepts in T will have some strong relationship with (sometimes the same) concepts in H, leading to many explanations where, although the relationship by itself may be right, it is not the most suitable answer for the entailment decision, as already shown in the example 149 above, hence the higher number of incorrect justifications.

Although there is still much room for improvement, the proposed approach for generating natural language justifications proved to be a viable solution especially in view of the fact that it employs an unsupervised technique and relies on already existing knowledge sources, yielding reasonable results without the costs and training data-dependency of supervised methods.

5.8 Comparing Definition Knowledge Graphs

Besides the improvements in the quantitative results, the interpretable characteristic of XTE represents a fundamental contribution towards Explainable AI: providing human-like explanations for the entailment decisions whenever a more complex semantic relationship is involved translates into a concrete gain for the final user, who can understand and judge the system's inference process. The justifications, though, depends heavily on the graph knowledge base employed in the entailment recognition. Overall, WordNet, Webster's and Wiktionary graphs deliver close results for the RTE3 and SICK datasets, but WordNet stands out for the more world knowledge-demanding BPI and GHS datasets, as can be seen in Table 5.3.

The impact of each DKG can be better measured by the *recall* obtained when they are queried: the more useful information the graph contains, the

more paths (meaning semantic relationships between source and target words) can be found and, consequently, more entailments can be recognized. Again, WordNet, Webster's and Wiktionary graphs show comparable recalls for the RTE3 and SICK datasets, but WordNet presents a much better recall for BPI and GHS. The Wikipedia graph, on the other hand, shows lower recall for all of the datasets, especially for the more knowledge-oriented BPI and GHS, despite being the larger knowledge base. This happens because Wikipedia, besides not defining verbs, privileges the definitions of people, places, arts and entertainment artifacts (films, books, songs, etc.) and other entities expressed by proper nouns. In fact, Wikipedia lacks definitions for many concepts present in the datasets for which relationships are sought: "violence" and "damage", "signatory" and "agreement", "decontamination" and "contaminants", or "bet" and "gamble", to name a few. This shows that, for the entailment task, the content type is more relevant than the amount of information in the graph. The WordNet graph, for example, corresponds to roughly only 10% of the Wikipedia graph, but contains far more common nouns denoting basic language concepts, better matching the task requirements.

The Wiktionary graph has a good coverage of common nouns, comparable to WordNet, but in some cases the completeness of its definitions may represent an issue: if not enough information regarding essential attributes is contained in the definition, that is, the definition of an entity fails to mention other entities it is essentially related to, paths will start but won't reach the target. Built by expert lexicographers, the WordNet definitions tend to follow some patterns and are more prone to cover essential attributes. On the other hand, in a collaborative environment, despite the larger volume of information that can be generated, high-quality standards cannot always be ensured. This is the reason why, in spite of its much larger dimensions, the Wiktionary graph cannot always surpass the WordNet one, yielding lower recall for the BPI and GHS datasets. Again, the coverage and regularity of the knowledge base contents prove to be more important than its size.

As for the Webster's graph, what was observed as an issue is the oldness of its source: dating from 1913, the Webster's Unabridged Dictionary naturally also covers all the most common language concepts, but, besides sometimes registering obsolete forms, like "camera obscura" for [photographic] "camera", lacks many modern concepts or concepts that were not of widespread use back then, such as "WMD" (Weapons of Mass Destruction), "website", "terrorist act" or "recall" (in the sense of "defective products callback"). Such modern concepts

are frequent in press content, so datasets like the GHS, totally generated from newspaper content, and the BPI, also derived from news content, may pose a challenge to this knowledge graph, which is confirmed by the lower recall (compared to WordNet) returned for these two datasets when the Webster's graph is used.

The justifications generated by each of the graphs are comparable in quality, with WordNet and Wikipedia graphs offering slightly more detailed explanations. An example from the GHS dataset, explained by the WordNet graph:

18623 T: GlaxoSmithKline has been forced to set aside £220m to settle anti-trust cases in the US over its anti-inflammatory drug Relafen.
18623 H: Glaxo hit by £220m US court blow
18623 A: YES

Entailment: yes
Justification:
A **case** is a term for any proceeding in a **court of law** whereby an individual seeks a legal remedy
Court of law is synonym of **court**

From the RTE3 dataset, explained by the Webster's graph:

158 T: Mr. Gotti, who is already serving nine years on extortion charges, was sentenced to an additional 25 years by Judge Richard D. Casey of Federal District Court.
158 H: Gotti was accused of extortion.
158 A: YES

Entailment: yes
Justification:
A **charge** is a kind of **accusation**
An **accusation** is an act of **accusing** or charging with a crime or with a lighter offense

From the SICK dataset, explained by the Wiktionary graph:

1459 T: A man is exercising
1459 H: A man is doing physical activity

1459 A: YES

Entailment: yes
Justification:
To **exercise** is to perform **physical activity** for health or training

From the BPI dataset, explained by the Wikipedia graph:

17.4 T: A Union Pacific freight train hit five people.
17.4 H: The train was moving along a railroad track.
17.4 A: YES

Entailment: yes
Justification:
A **freight train** is a group of freight cars or goods wagons hauled by one or more locomotives on a **railway** [...]
Railway is synonym of **railroad track**

Summing up, the WordNet graph presents the best recall across all datasets, due to its good term coverage and definitions' completeness. The Wiktionary and Webster's graphs also show good recall but their performance can be weakened by the incompleteness resulting from the amateur nature of the definitions creation process, in the Wiktionary case, or by the lack of modern terms which are frequent in contemporary language, in the Webster's instance. The Wikipedia graph, due to its encyclopedic nature, has well constructed and complete definitions, but lacks many basic concepts, yielding the lowest recall regardless of the dataset.

5.9 Summary

In this chapter, the experiments performed for evaluating the proposed text entailment approach were described. Like the approach development, the experiments were also split into two phases so the semantic-oriented Distributional Graph Navigation model and the complete composite interpretable XTE entailment system could be evaluated separately and then compared.

The experiments involved distinct types of datasets and knowledge bases in order to assess the approach behavior in different scenarios. Four text entailment datasets were tested: RTE3 and SICK, which are more focused on

linguistic phenomena, and BPI and GHS, which are more semantically complex and demand a larger amount of world knowledge for the entailment recognition. All four datasets mix syntactic and semantic entailment pairs but in different proportions, and the entailment pairs structural characteristics also vary among them: while SICK and BPI contain pairs composed of short to medium size T and H sentences, RTE3 and GHS have very long T sentences for usually short H statements. As for the knowledge bases, definition graphs were built from four different lexical resources: WordNet, the Webster's Unabridged Dictionary, Wiktionary and Wikipedia, allowing an assessment on how each of them impact the proposed approach results and influenced its interpretability.

From the evaluation of the first version of the DGN, it was observed that it delivered better precision and F1-score for highly world knowledge-demanding datasets when compared with syntactic-oriented approaches, but yielded lower recall for not being able to deal with syntactic entailments. This first round of experiments also allowed the identification of some limitations in the DGN first implementation, which could be fixed before the model was integrated into the complete entailment system. The main conclusion of this experiment stage is that **looking for the semantic relationship between T and H improves the precision**, but a **complimentary syntactic approach is still necessary**.

The second round of experiments evaluated the performance of the complete composite interpretable text entailment system. All the possible combinations of datasets and knowledge graphs were tested, and both the quantitative results and the natural language justifications were analyzed. The main conclusions of the quantitative analysis are:

- The XTE composite approach outperforms entailment algorithms that employ a single technique, be it syntactic or semantic, to tackle all types of entailments. It shows an improvement of, on average, around 6% in F1-score when compared to the best performing baselines.

- It is less dependent on training data, since it combines learned parameters with independent, external commonsense knowledge which works well regardless of the dataset. This leads to less variability among the results obtained for different datasets, including those for which no training data is available.

Regarding the influence of the knowledge bases in the system outputs, given that text entailment deals with language variability, it could be observed that knowledge graphs covering the most basic, everyday language concepts yield the best results, so regular dictionaries, such as WordNet, Webster's and Wiktionary are more useful than encyclopedic KBs like Wikipedia for this task. We also found that definitions created by lexicographers under a controlled environment tend to be more complete and, consequently, provide better recall and somewhat more detailed justifications than those created in collaborative environments by lay users. Furthermore, contemporary resources can show some advantage over older dictionaries for containing modern terms frequently occurring in the present-day language but absent from ancient lexicons like Webster's dictionary. As a summary, the ideal resource for the textual entailment task would have the following characteristics:

- High coverage of common, basic language concepts;
- Completeness of definitions, which should always include, as a minimum, the type and the essential attributes of the definiendum;
- Regularity in the syntactic and semantic structure of definitions, which usually follows from the work of a limited set of professional lexicographers;
- Content constantly updated to include modern terms or new meanings for existing terms.

The experiments proved the hypothesis that a combination of techniques aimed at dealing with different entailment phenomena performs better than single-method approaches, and, by testing and comparing several graph knowledge bases, we also showed that the use of external world knowledge not only improve quantitative results, but is also a valuable feature for increasing the system interpretability.

Chapter 6

Error Analysis

In this chapter, the results of a detailed error analysis are presented. The systematic error analysis was carried out with the aim of identifying the proposed approach limitations, as well as their nature and causes. This study refers to the results obtained at the stage two of the experiments, where the final, complete entailment system was evaluated, as described in Chapter 5.

The chapter starts with the description of the principles that guided the execution of the analysis. A classification of the error types is given next, followed by the distribution of each type across datasets and a discussion about the results pointing out the main error triggers and the directions for possible future developments.

6.1 Analysis Guiding Principles

The error analysis was performed over the full results obtained in the experiments employing the WordNet definition graph, since this is the knowledge base that delivers the overall best results (see Chapter 5). All false negatives and false positives for each of the four tested datasets – RTE3, SICK, BPI, and GHS – were examined to uncover the source of the error, with the aid of further data extracted from the system's internal flow, be it information exchanged between modules, such as the Preprocessing or Context Analysis modules results transmitted to the Distributional Graph Navigation (DGN) module, or intra-module information, like the definition of source-target input word pairs.

While errors in entailments solved by the Tree Edit Distance (TED) module are directly and exclusively linked to the edit distance threshold t, in the DGN module case, many different parameters and factors external to the algorithm itself can have an influence over a wrong entailment decision. The TED model relies on the information extracted from training data to compute average edit distances for both positive and negative entailments, and then, based on these values, assess the validity of the entailment for new data. Therefore, errors occur either when an entailment has a higher-than-average edit distance, generating false negatives, or when a non-entailment has a lower-than-average edit distance, yielding a false positive.

On the other hand, the DGN model results depend on, besides the algorithm itself, the preprocessing stage results, the distributional model used to compute semantic relatedness measures, and the graph knowledge base employed for finding the semantic relationships. The analysis was, then, concentrated on the entailment pairs solved by the DGN module, since in this scenario the errors can be caused by a wider range of factors.

6.2 Error Classification

6.2.1 False Negatives

False negatives occur when there is a semantic relationship between T and H that could not be detected, resulting in an entailment being classified as a non-entailment. The decision can happen before or during the DGN procedure, caused by syntactic or semantic factors.

6.2.1.1 Syntactic Factors

Syntactic errors refer to misclassifications while the structure of the sentences composing the entailment pair is being analyzed. Such errors occur mostly in preprocessing stages and affect the quality of the inputs sent to subsequent steps in the entailment workflow.

Tokenization/POS Tagger/Splitting Error

Tokenization and part-of-speech (POS) tagging are performed by the Stanford parser and are mainly used to generate the bag-of-words used to compute the overlap set that is, in turn, used by the model router to decide which module

the pair is sent to. Splitting is a utility function implemented within the system and built upon the WordNet stemmer, intended to split the sentences into phrases when applicable, detecting multi-word expressions rather than always considering only single words. Splitting is used for building the source-target pairs which is sent as inputs to the DGN.

Errors occur when the splitting procedure does not identify the correct phrases, when the tokenizer does not stem the words correctly, or when words are assigned the wrong part-of-speech tag during the tokenization procedure. Wrong splitting and tokenization make the search start at the wrong node in the graph, not reaching the target. Wrong tokenization can also result in the entailment pair being sent to the wrong module to be solved. The wrong POS tag can prevent the start of the path search, in case the source word is a noun or verb incorrectly classified as an adjective or adverb, or keep a path from reaching the target, if such target word wasn't correctly stemmed and tagged. Examples[1]:

24864 T: Lorraine Heggessey today said she had no regrets over chasing ratings and said accusations that she had "dumbed down" BBC1 had hurt.
24864 H: Heggessey: Dumbing down accusations hurt
24864 A: YES

371 T: A man is resting on a chair and rubbing his eyes
371 H: A man is sitting on a chair and rubbing his eyes
371 A: YES

14.4 T: Britain puts curbs on immigrant labor from Bulgaria and Romania.
14.4 H: Britain restricted workers from Bulgaria.
14.4 A: YES

In the first example, the tokenizer correctly stems "dumbed" in T, returning the verb "dumb" as a token, but fails to do the same for "dumbing" in H, keeping the token as it is and classifying it as a noun. The tokenization error leads to a wrong routing decision since all the tokens in H are also contained in T, therefore this pair should be sent to the Tree Edit Distance model and not to the Distributional Graph Navigation. In the second example, the cor-

[1]Examples from GHS, SICK, and BPI datasets, respectively.

rect source-target pair is "rest" and "sit", for which there is a clear semantic relationship, but the splitting function returns "rest on" instead of "rest" as a token. Since "rest on" has a different meaning, paths starting at this node don't reach the target. In the third example, the entailment is given by the source-target pair "curb" (noun) and "restrict" (verb), but the tokenizer/POS tagger does not recognize "restrict" as a verb or stems it as expected, but rather returns "restricted" as a token, classifying it as an adjective. The path between the noun "curb" and the verb "restrict" indeed exists in the WordNet graph, but since the target word is misclassified and not stemmed, the path between the text and the hypothesis can't be found. To illustrate the impact of wrong syntactic preprocessing, when "restricted" is replaced by "restricts" in the hypothesis in the last example, the entailment decision correctly changes to *yes*, because "restrict" is accurately classified as a verb, and the following justification is generated:

Entailment: yes
Justification:
A curb is an act of restraining power or action or limiting excess
To restrain is synonym of to restrict

6.2.1.2 Semantic Factors

Semantic errors refer to a misinterpretation or lack of information about the correct meaning of terms and, consequently, a failure to establish the correct semantic relationship between them. They can be caused by limitations in the system, but also by external factors such as knowledge base incompleteness or dataset inaccuracy.

Wrong Context Analysis Decision

This error happens when the Context Analysis module finds some information which it considers to be inconsistent, classifying the pair as a non-entailment, but the intended meaning is different from that detected by the module. Examples[2]:

[2]Examples from BPI and SICK datasets, respectively.

4.4 T: Iran purchased plans for building a nuclear reactor from A.Q. Khan.
4.4 H: Khan sold some plans to Iran.
4.4 A: YES

66 T: Two people wearing snowsuits are on the ground making snow angels
66 H: Two people in snowsuits are lying in the snow and making snow angels
66 A: YES

In the first example, the Context Analysis module identifies "purchase" and "sell" as antonym concepts, while the actual intended relationship between these terms is one of dependency and not opposition. In the second example, it concludes that H has an unsatisfiable clause, since it has more clauses that T, although both coordinated clauses in H are fully satisfied by the single clause in T.

Correct Source-Target Pair not Found

This error occurs when the source-target pair that establishes the entailment relationship is not identified as a relevant input for the DGN because, although being semantically related, the relatedness score returned for it by the DSM (in this case, word2vector) is not high enough to place it among the top k best scoring pairs, or because one of the terms is a multi-word expression not caught by the splitting algorithm. A suitable source-target pair, in the form expected by the DGN model, may not even be present in the entailment pair at all. Examples[3]:

27599 T: The Leeds coach Tony Smith has added spice to tonight's meeting of Super League's top two at St Helens with the strongest criticism yet from within the game of the weak team Saints sent to Bradford on Easter Monday.
27599 H: Smith slams Saints over weak team
27599 A: YES

2443 T: Pete Doherty found himself languishing behind bars for a fourth night yesterday after his record label bosses again struggled to raise his £100,000 bail.
2443 H: Fourth night in jail for Pete Doherty

[3]Examples from GHS (first and second) and SICK (third) datasets.

2443 A: YES

7605 T: A brown dog and a gray dog are playing in the snow
7605 H: Two dogs are playing in the snow
7605 A: YES

In the first example, the best pair is given by the words "criticism" in T and "slam" in H, but other highly semantically related pairs (according to the DSM's judgment, and mostly because of another, sports-related meaning of "slam"), such as "game" and "slam", "league" and "slam", or "coach" and "slam", among others, push the correct entry to the end of the list, excluding it from the input set. In the second example, the entailment is established by the pair "behind bars" and "jail", but the splitting algorithm, which considers as a token the longest sequence of words that exists as an entry in WordNet, does not identify "behind bars" as a valid phrase (since it is not in WN), so the right pair cannot be found. In the third example, rather than a semantic relationship between concepts in T and H, what exists is an arithmetic correspondence between the sentences, since "a brown dog and a gray dog" = "two dogs". No suitable pair can be found to be sent as input to the DGN, leaving this entailment scenario out of the proposed approach scope.

Path not Found by the DGN

In this case, although the correct source-target pair was identified, both source and target words were correctly tokenized and POS tagged, and the path between them exists in the definition knowledge graph, the DGN wasn't able to find such path, because the relatedness values returned by the DSM for the relevant role nodes were too low. Low semantic relatedness scores can either exclude a role node from the set of nodes to be visited next, or make such nodes to be located far from the top of the stack of paths to be explored by the DGN. In the last case, the maximum number of paths can be reached before the correct path is explored, and the search ends returning a non-entailment decision. Examples[4]:

16.2 T: Satomi Mitarai bled to death.
16.2 H: Satomi Mitarai died.

[4]Examples from BPI and SICK datasets, respectively.

16.2 A: YES

7916 T: A dog is running towards a ball
7916 H: A dog is running towards a toy
7916 A: YES

In the first example, "death" and "die" are correctly identified as the source-target pair, but "death" has eight different senses in WordNet, hence eight different definitions, but the nodes for the most relevant one – "the event of dying or departure from life" – are not the highest scoring ones, pushing the paths containing them to the bottom of the stack and leaving them unexplored. The same happens in the second example where the semantic relationship holds between "ball" and "toy", but "ball" has twelve different definitions and the relevant one – "a spherical object used as a plaything" ("plaything" is, in turn, synonym of "toy") – does not have its role nodes scoring high enough to be explored in the path search.

Source Word Category not Covered by DKG

This error happens when the part of speech of the source word in the source-target pair that establishes the entailment relationship is not covered by the definition knowledge graphs. Currently, the DKGs include definitions only for nouns and verbs, so if the source word is an adjective or adverb it will not be possible to look for a path in the DKG starting from it. Examples[5]:

6.3 T: Armed men kidnapped an Associated Press photographer on Tuesday as he was walking in Gaza City.
6.3 H: There was a group of men who possessed guns.
6.3 A: YES

616 T: The world's population is set to reach a staggering 10bn by the middle of the next century up from 5.7bn now.
616 H: The current world population is 5.7 billion.
616 A: YES

[5]Examples from BPI and RTE3 datasets, respectively.

In the first example, the entailment is established by the pair "armed" and "gun", but since "armed" is an adjective, the search in the DKG can't proceed for this pair. In the second one, the best pair is given by the words "now" and "current", which also prevents the start of the search, for the source word is an adverb.

This only happens for source words because the search always begins with an entity node. If the target word is an adjective or adverb, it can still be found by the DGN, because, although it does not exist as an entity node, it may be part of a role node, and every head word in a role node is compared against the target regardless of its part-of-speech.

Absent or Insufficient Information in the DKG

This error occurs when the definition for the source word is not present in the resource that gave origin to the DKG, so it will not be found as an entity node, despite being a noun or verb. It also occurs when the source word exists an entity in the graph but the information encoded in its role nodes is insufficient to derive a path that reaches the target. Examples[6]:

7.1 T: Sony is doing a huge retail rollout of their new Playstation.
7.1 H: Sony is rolling out their new Playstation.
7.1 A: YES

35 T: A Revenue Cutter, the ship was named for Harriet Lane, niece of President James Buchanan, who served as Buchanan's White House hostess.
35 H: Harriet Lane was a relative of President James Buchanan.
35 A: YES

In the first example, the pair "rollout" (noun) and "roll out" (verb) establishes the entailment, but the definition for "rollout" is not included in WordNet, so the graph search cannot proceed. It is, nevertheless, present in Wiktionary, so when this graph is used the DGN successfully finds a path between the source and the target and confirms the entailment. In the second example, the source-target pair is given by the words "niece" and "relative" but, although the definition for "niece" exists in WordNet, it does not include the word "relative" or any other word whose definition leads, directly or indirectly, to "relative".

[6]Examples from BPI and RTE3 datasets, respectively.

The DGN generated from the Webster's Dictionary does contain such path, so this entailment pair is correctly classified as an entailment when this graph is employed.

Dataset Classification Error

This error happens when the entailment pair has the wrong classification in the dataset, clearly identifiable by the positive label being inconsistent with the information contained in the pair of sentences. Although counted as an error, since the system's answer differs from the dataset's gold standard, this kind of false negative is actually a correct entailment decision. Examples[7]:

19 T: A person in a black jacket is doing tricks on a motorbike
19 H: A man in a black jacket is doing tricks on a motorbike
19 A: YES

977 T: A man is falling off a horse on a track and is laid in the wild
977 H: A man is getting on a horse on a track laid in the wild
977 A: YES

In the first example, H is more specific than T, and it cannot be concluded that "a person" is "a man". The Context Analysis module correctly identifies this as an inverse specialization, rejecting the entailment, but since the pair is erroneously classified as an entailment in the dataset, it counts as a false negative. In the second example, there is a clear contradiction, since "fall off" is the opposite of "get on" in this context, but the pair is nonetheless also classified as an entailment.

Classification errors only occur in the SICK dataset. In this dataset, an initial set of sentences was expanded for creating similar, contradictory and neutral versions of the original sentences, which were then combined pairwise (Marelli et al., 2014). The entailment relationship holding between each pair was validated through crowdsourcing and the final label was decided by a majority vote schema. The lack of expert manual validation on the final data could possibly explain the classification errors in the dataset.

[7]Examples from SICK dataset

6.2.2 False Positives

False positives occur either when T and H are contradictory sentences, or when there is no relationship between T and H at all, that is, given that T is true it cannot be said whether H is true or false, but the DGN nonetheless finds a semantic relationship classifying the pair as an entailment. As with false negatives, the wrong decision can be due to syntactic or semantic factors, which can arise before or during the DGN procedure.

6.2.2.1 Syntactic factors

Again, syntactic errors refer to the wrong processing of the structure of the sentences composing the entailment pair, occurring mostly during the preprocessing stage and affecting the quality of the inputs sent forward in the entailment workflow.

Tokenization/POS Tagger/Splitting Error

This kind of error happens when sentences are split into the wrong phrases by the splitting procedure, or when words are stemmed incorrectly and/or assigned the wrong part of speech tag during the tokenization procedure. Differently from the false negative scenario, in false positives the most common tokenization-related issue is the same word occurring in T and H being stemmed and tagged differently in each sentence, leading the DGN to consider them as two different, but usually semantically related, concepts. Examples[8]:

7308 T: There is no man wearing a straw hat who is smoking a cigarette
7308 H: A man is wearing a straw hat and smoking a cigarette
7308 A: NO

Entailment: yes
Justification:
To smoke is to inhale smoke from cigarettes, cigars or pipes
A cigarette is a kind of tobacco
Tobacco is the tobacco plant dried and prepared for smoking or ingestion

13.101 T: Eating vegetables may keep the brain young. Eating vegetables may slow the mental decline associated with old age.

[8]Examples from SICK and BPI datasets, respectively.

13.101 H: The brain is eating vegetables.
13.101 A: NO

Entailment: yes
Justification:
Eating is the act of consuming food
To consume is a way of to eat

In the first example, the verb "smoke" is present in both T and H, but it is correctly tokenized and POS tagged only in T. In H, it is tokenized as "smoking" and POS tagged as a noun. Being the same word, "smoke" should be included in the overlap set, enabling the Context Analysis module to detect the negation present in the pair (the Context Analysis module can only detect simple negations, where the only non-overlapping tokens are negation words or expressions). Yet, the wrong tokenization leaves "smoke" and "smoking" out of the overlap set as two different entities and, since they are semantically related, the DGN can find a path between them falsely confirming the entailment. Similarly, in the second example, "eating" is returned as a token and POS tagged as a noun in T, but in H what is returned in the verb "eat". In this case, if "eat" was correctly tokenized and POS tagged, the pair should be sent to the Tree Edit Distance module, since every word in H is also present in T and there is only a (big) difference in the sentences' syntactic structure.

6.2.2.2 Semantic Factors

Semantic errors regard the misinterpretation of the meaning of the sentences as a whole, leading to the establishment of a semantic relationship between them when the entailment cannot be confirmed. They are mainly caused by system limitations, but also by dataset inaccuracy.

Undetected Context Information

This kind of error occurs when, although there is a pair of entities $e_1 \in T$ and $e_2 \in H$ for which a semantic relationship exists, there is also some context information that invalidates the entailment but that involves more complex phenomena out of the scope of the Context Analysis module and, hence, cannot be

detected. Examples[9]:

4.101 T: Iran purchased plans for building a nuclear reactor from A.Q. Khan.
4.101 H: Iran did not buy plans for a nuclear reactor.
4.101 A: NO

Entailment: yes
Justification:
To purchase is synonym of to buy

10.101 T: New mum Madonna finally broke her silence today over the row surrounding her adoption of African baby David Banda.
10.101 H: Madonna was silent today.
10.101 A: NO

Entailment: yes
Model: GraphNavigation
Justification:
A silence is a state of being silent

In the first example, the text is a negation of the hypothesis but there is also a semantic relationship between them established by the concepts "purchase" and "buy". Since the Context Analysis module currently only detects simple negations, the other elements involved lead the DGN to look for a path in the definition graph and reach an incorrect decision. In the second example, although there is a relationship between "silence" and "'silent", the presence of the verb "break" having "silence" as its objects makes T contradicts H, a context change not yet detected by the Context Analysis module either.

Wrong Semantic Relationship Assumption

This error happens when, although there is no entailment relation between the text and the hypothesis, there is a pair of entities $e_1 \in T$ and $e_2 \in H$ which have a high semantic relatedness score and for which the DGN can indeed find a path in the definition graph. If the pair does not include any of the phenomena that can be detected by the Context Analysis module for preventing a misclas-

[9]Examples from BPI dataset.

sification, the DGN erroneously concludes it is an entailment. Examples[10]:

710 T: In the US the Salvation Army offers shelters for the homeless in most areas of the country.
710 H: The US Army provides shelters for the homeless.
710 A: NO

Entailment: yes
Justification:
To offer is synonym of to provide

108 T: A player is throwing the ball
108 H: A player is running with the ball
108 A: NO

Entailment: yes
Justification:
To throw is a way of to move
To move is synonym of to run

30.100 T: There was an explosion on Panam flight 103.
30.100 H: Panam was flying 103 airplanes.
30.100 A: NO

Entailment: yes
Justification:
A flight is a kind of plane
Plane is synonym of airplane

Unlike undetected context information, which involves some kind of information that makes T contradict H, such as negation, opposition, etc., a wrong semantic relationship assumption occurs when it is not possible to conclude that H follows from T but there is no clear contradiction either, so no context information could be detected whatsoever. It may happen when e_1 and e_2, although semantically related, refer to different entities, as in the first example, where "offer" and "provide" have a synonymy relationship, but "offer" has "Salvation Army" as subject and "provide" refers to "US Army" instead. It may also oc-

[10]Examples from RTE3, SICK and BPI datasets, respectively.

cur when the target word is a common and frequent concept which will have a high semantic relatedness score with a wide range of concepts, as in the second example, where the target is "run", a word that can appear in many different contexts. Nevertheless, as can be seen in the last example, where "flight" and "airplane" are indeed related, none of them is too generic and both have the same referent ("Panam"), sometimes, given the DGN model rationale, it is simply too hard to identify a non-entailment if the sentences are highly semantically related and lack contradictory context information.

Dataset Classification Error

As in the false negative case, this kind of misclassification occurs when the entailment pair has the wrong classification in the dataset, having a negative label when the hypothesis can indeed be inferred from the text, characterizing a positive entailment. Examples[11]:

1522 T: A woman is cutting some flowers
1522 H: A woman is cutting some plants
1522 A: NO

Entailment: yes
Justification:
A flower is a kind of plant

1699 T: A kitten is drinking fresh milk
1699 H: The cat is drinking some milk
1699 A: NO

Entailment: yes
Justification:
A kitten is a young cat

In the first example, H is a generalization of T, since "plant" is more general than "flower", as correctly detected and justified by the DGN, hence the entailment is valid but, due to the wrong label, it is counted a false positive. The same happens in the second example, where, despite the non-entailment label, H is also a generalization of T and the DGN accurately finds and explains the

[11] Examples from SICK dataset

relationship between "kitten" and "cat". Again, such classification errors only occur in the SICK dataset.

It is important to emphasize that only the primary error is being considered, that is, the error that happens earlier in the entailment pipeline. For example, if the Context Analysis module makes a wrong decision, it does not mean the pair would be solved successfully otherwise, because the path could not be found in the DKG either, for instance. That means that, although there are still some limitations regarding the algorithms that could be further explored in future work, the system performance also depends on external factors beyond our control, such as the quality and completeness of the linguistic resources used in the construction of the knowledge bases. Table 6.1 summarizes the main characteristics of the errors identified in the analysis.

	Error	**Nature**	**Origin**	**Source**
False Negatives	Tokenization/POS Tagger/Splitting Error	syntactic	internal/ external	util algorithm/ parser
	Wrong Context Analysis Decision	semantic	internal	core algorithm
	Correct Source-Target Pair not Found	semantic	internal/ external	core algorithm/ DSM
	Path not Found by the DGN	semantic	internal/ external	core algorithm/ DSM
	Source Word Category not Covered by DKG	semantic	external	knowledge base
	Absent or Insufficient Information in the DKG	semantic	external	knowledge base
	Dataset Classification Error	semantic	external	dataset
False Positives	Tokenization/POS Tagger/Splitting Error	syntactic	internal/ external	util algorithm/ parser
	Undetected Context Information	semantic	internal	core algorithm
	Wrong Semantic Relationship Assumption	semantic	internal	core algorithm
	Dataset Classification Error	semantic	external	dataset

Table 6.1: Error classification.

As can be noted in Table 6.1, there is a balance between internal and external factors. Internal factors indicate the current limitations of the proposed approach, while external factors refer to third-party resources for which no fine-tuning is possible. Some errors are caused by a mix of both types of factors, like the syntactic errors for both false negatives and false positives, which can be caused by an internally implemented utility algorithm (for multi-word expressions sentence splitting) or by the Stanford parser (for tokenization and POS tagging). The same happens for the errors of types "Correct Source-target Pair not Found" and "Path not Found by the DGN", which are heavily influenced by the semantic similarity scores provided by the DSM (word2vec), but also by the system parameters whose computations are embedded in the core algo-

rithm: the maximum number of source-targets pairs k, in the first case, and the semantic relatedness threshold η in the second.

Purely internal errors are mainly related to the extent to which the system can grasp context information: the Context Analysis module can either misinterpret such information, yielding false negatives (Wrong Context Analysis Decision), or be unable to catch more complex contradictions, resulting in false positives (Undetected Context Information). Wrong Semantic Relationship Assumption errors are also related to context information but they occur when the entailment relationship between T and H is neutral rather than contradictory, making such context much subtler and harder to identify.

Finally, purely external errors are caused by either the contents (or lack thereof) of the graph knowledge base employed by the DGN or the quality of the dataset tested. While incompleteness is an inherent characteristic of many knowledge bases, dataset issues could possibly be associated with their growing size: as most datasets introduced lately are machine learning-oriented, large-scale resources, validation becomes a challenging task, which can compromise the final data quality.

6.3 Error Analysis Results

Table 6.2 presents the results of the error analysis for the four tested datasets. As mentioned before, these numbers refer to the outcome of the experiments employing the WordNet DKG. Figures 6.1 and 6.2 present the same results graphically, divided into false positives and false negatives, respectively.

	Error	**RTE3**	**SICK**	**BPI**	**GHS**
False Negatives	Tokenization/POS Tagger/Splitting Error	9%	12%	20%	2%
	Wrong Context Analysis Decision	37%	19%	36%	69%
	Correct Source-Target Pair not Found	32%	9%	8%	14%
	Path not Found by the DGN	1%	5%	4%	1%
	Source Word Category not Covered by DKG	5%	10%	8%	4%
	Absent or Insufficient Information in the DKG	16%	34%	24%	10%
	Dataset Classification Error	0%	11%	0%	0%
False Positives	Tokenization/POS Tagger/Splitting Error	4%	5%	3%	1%
	Undetected Context Information	10%	15%	14%	0%
	Wrong Semantic Relationship Assumption	86%	77%	83%	99%
	Dataset Classification Error	0%	3%	0%	0%

Table 6.2: Error distribution.

As can be noted in Table 6.2 and Figure 6.1, most false negatives are due to either a wrong Context Analysis decision or DKG incompleteness. Correctly

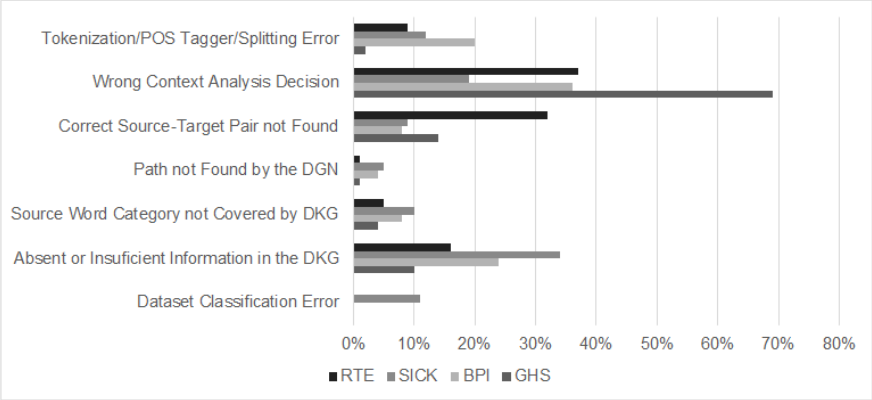

Figure 6.1: Error distribution for false negatives.

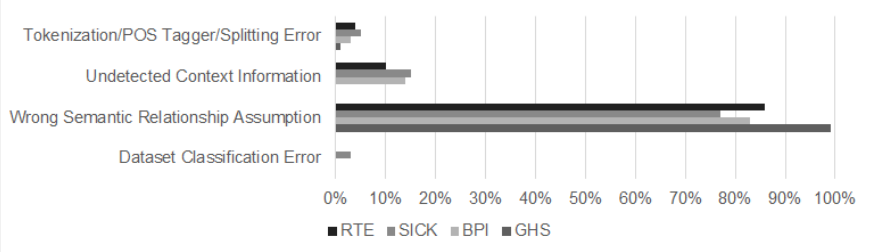

Figure 6.2: Error distribution for false positives.

interpreting context information is especially challenging for the GHS dataset, where T is usually a very long sentence often composed of multiple clauses. Most context misinterpretations in this dataset involve concepts denoting inverse specialization or antonyms, which are indeed present but do not affect the validity of the entailment because, in T, the "offending" terms appears in a clause that does not directly entail H, but rather presents complementary related information.

Another phenomenon worth mentioning involving context information errors is the one observed in the SICK dataset, where many pairs of terms are used as synonyms, but there is also a specialization relationship between them. For example, "path" and "trail", "group" and "team", "pan" and "skillet", "aircraft" and "plane", "man" and "guy", "fire" and "bonfire", among others, are used as synonyms, but formally it is also true that a "trail" is a kind of "path", a

"team" is a kind of "group", and so on. When the first term in the pair appears in T and the second one in H, the Context Analysis module identifies it as an inverse specialization, which cannot be considered altogether wrong, but does not account for spoken language subtleties and variations that not always adhere to the formal language specifications. This phenomenon is only observed in the SICK dataset, possibly because its pairs were created from descriptions of images and videos. The way its sentences are written is closer to the way people speak when making such descriptions, while all the other datasets are derived from press content and therefore present more formal written language. This suggests that detecting entailment in spoken language may require additional resources and a more flexible interpretation of language usage, which may not always obey the formal recorded structures and semantics.

Regarding the errors attributed to the knowledge base used by the DGN, while the source word category-related error rate is constant for all DKGs (all graphs, regardless of the lexicon it is derived from, only contains noun and verb definitions), the absent or insufficient information-related misclassification refers exclusively to the knowledge graph analyzed: entailments that cannot be recognized due to a missing path in the WordNet graph may be successfully identified when another graph is employed. Table 6.3 shows how other DKGs perform when the WordNet graph fails for each tested dataset.

Dataset	**WordNet Misses**	**Hits**		
		Webster's	**Wiktionary**	**Wikipedia**
RTE3	20	8	7	1
SICK	103	47	16	4
BPI	6	1	1	0
GHS	18	3	5	1

Table 6.3: Comparison between WordNet misses and other DKGs hits for the analyzed experiment results.

Table 6.3 shows that other DKGs contain the information absent in WordNet in only a fraction of the cases. For example, for the RTE3 dataset a total of 20 pairs were misclassified due to absent or insufficient information in WordNet. Out of these 20 pairs, 8 could be correctly solved when Webster's graph was used, 7 when Wiktionary was employed and only 1 when Wikipedia was the chosen knowledge base. At a maximum, for the WordNet graph misses, there is a hit 43% of the times another DKG is used for the RTE3 and SICK datasets, which are semantically simpler. When we consider the BPI and GHS datasets, which

are semantically more complex and require a larger amount of world knowledge, the other DKGs maximum hit rate falls to 16% of the WordNet graph misses. This confirms that WordNet is the best knowledge base for querying semantic relationships, but also suggests that a combination of resources could boost accuracy. Nevertheless, combining resources for increasing information coverage also leads to increased graph size, so more advanced graph query methods are necessary for not impairing the system performance.

At the other end of the false negative error spectrum, misclassifications due exclusively to the right path not being found by the DGN account for the smallest portion of the proposed approach failures. There is a single occurrence of this kind of error for the RTE3, BPI and GHS datasets each, and 14 for the SICK dataset (this dataset contains many similar pairs with very slight syntactic variations among them, which could be solved by the same path, so a single path that could not be found ends up accounting for multiple false negatives). For most of these occurrences, we observed that the main hindrance is the lower-than-appropriate (considering the context set by the sentences) similarity scores returned by the DSM. This shows that the Distributional Graph Navigation algorithm is a robust and reliable model for exploring external knowledge bases, which succeeds at least 95% of the times when provided with the right inputs.

As evidenced by Figure 6.2, false positives are predominantly caused by a wrong semantic relationship assumption between T and H. As described earlier, this kind of error happens for pairs with a *neutral* relationship, that is, nor entailment neither contradiction, so no marked context information which can clearly indicate a non-entailment is available. This is the most challenging scenario for the proposed entailment approach, and points to the need of the identification of further, subtler syntactic and semantic evidence that can be extracted from T and H so a negative entailment can be classified as so even when the sentences are very semantically related.

6.4 Summary

In this chapter a systematic error analysis focused on the Distributional Graph Navigation model was presented, including a classification of the types of errors found during the experiments and a quantitative evaluation of the impact of each type across all the tested datasets.

Errors leading to false negatives and false positives were divided into syntactic and semantic factors and further classified as internal or external (or a mix of both) aspects. The main error roots were also identified: for internal errors, the Context Analysis and DGN core algorithms, or the utility sentence splitting function are the main error sources, while external errors are caused by the Stanford parser and POS tagger, the DSM (word2vector), the knowledge base, or the dataset.

The quantitative analysis showed that wrong Context Analysis decisions due to misinterpretations of context information, along with knowledge base incompleteness are the most common problems leading to false negatives. False positives, on the other hand, are mainly caused by wrong semantic relationship assumptions, when, despite the non-entailment relationship between T and H, the DGN can still find a semantic relationship between two entities $e_1 \in T$ and $e_2 \in H$, because no evident context information preventing the misclassification can be detected.

Opportunities for future work, such as the exploration of more advanced graph analysis techniques to enable the aggregation of multiple knowledge base for better information coverage, and the use of further resources and features for capturing more complex or subtler context information, were also highlighted. These points indicate possible paths for increasing the accuracy of semantic, world knowledge-demanding text entailments.

Chapter 7

Conclusion

The advances that have been helping to popularize Artificial Intelligence applications owe much to the Natural Language Processing research field. NLP enables smooth human-machine communication, making it possible for users to interact with smart applications and devices using their own language, as if talking to another person. Processing human language involves many challenges, because computers should not only interpret what a text says, but also understand what it means beyond what is explicitly said, requiring not only natural language text interpretation at both syntactic and semantic levels, but also inference capabilities so implicit meanings that are essential to the correct interpretation of the message can be uncovered.

Text entailment, the task of determining whether a piece of text logically follows from another piece of text, is a key component in NLP. It provides input for many semantic applications such as question answering, text summarization, information extraction, and machine translation, among others, dealing with inference issues so other applications can cope with language variability. The work presented in this thesis aimed at filling some of the gaps in the text entailment area by proposing a way for dealing with the differences among the phenomena present in entailment, developing a method for better addressing entailments involving more complex relationships, including the knowledge acquisition necessary for performing this task, and including explainability features to make the final system interpretable.

Chapter 2 described the main developments in both the text entailment recognition and the semantic interpretability areas. The text entailment review showed that there is a substantial variety of approaches, but most of them

heavily rely on sentence syntactic analysis, using bag-of-words or dependency parse as a representation schema, and shallow semantic information, such as synonym and hypernym links, verb frames, and semantic role labeling. Logic-based and rule-based approaches try to incorporate further knowledge through logical axioms or entailment rules, which are, though, usually generated from the same shallow semantic relations. Large scale resources provide paraphrase-style rules, which cover mostly the equivalence relationship, and hand-crafted axioms, which encode more complex inference rules, are sometimes included in a small number. Structured content from Wikipedia, which covers a larger variety of relationships between (mostly named) entities, is also used by a few approaches. It was also shown that the Natural Language Inference task, a subtask of textual entailment, has brought great advancements, with deep leaning models achieving high accuracy, but whose performance nevertheless is being influenced by linguistic bias in the datasets which allows supervised models to learn patterns in hypotheses but not the correlation between premises and hypotheses, many times correctly classifying examples without actually performing inference.

The semantic interpretability review showed that different AI fields are now concerned about developing more interpretable models, but most of the times the focus is on transparency, and only a few models provide post-hoc explanations, in the form of text or visual cues. From the usability point of view, it was argued that such explanations are more user-friendly, because they allow non-experts (i.e., non-developers) to understand the rationale behind a system's prediction without having to go through its internal operations. A further analysis, focused on text entailment systems' interpretability features, showed that, as approaches moved from alignment and transformation strategies to more complex, multiple-features classification models (including NLI systems), transparency has decreased, and providing explanations is still not a concern. The publication associated with this chapter is (Silva, Freitas, & Handschuh, 2019b).

Chapter 3 focused on the knowledge acquisition aspect of the proposed approach, describing how dictionaries definitions are processed to be represented in a structured way so they can be used by a reasoning mechanism intended to recognize entailments involving semantic relationships. The conceptual model proposed aims at capturing the underlying semantic structure of lexical definitions, describing their main components and the relationships among them. The proposed construction methodology allows the automatic conversion of a set of natural language definitions into an RDF graph, which will provide definitional knowledge, with all the variety of relationships it can cover, to the text en-

tailment proposed approach. The publications associated with this chapter are (Silva, Handschuh, & Freitas, 2016) and (Silva, Freitas, & Handschuh, 2018a).

Chapter 4 detailed the development of the proposed composite interpretable text entailment approach. It showed how the Distributional Graph Navigation model explores a definition knowledge graph to look for semantic relationships between a text and a hypothesis and then generates a natural language justification explaining this relationship. This model, aimed at solving semantic entailments, was then integrated into a complete system, where syntactic entailments are solved by a Tree Edit Distance model, and a routing mechanism analyzes each entailment pair to identify the most relevant phenomena (syntactic or semantic) and sent it to the suitable module. It was also described how a complementary module extract further syntactic and semantic features from the entailment pair so additional context information can be retrieved to better inform the final entailment decision. The publications associated with this chapter are (Silva, Freitas, & Handschuh, 2018b), (Silva, Freitas, & Handschuh, 2019a), and (Silva, Freitas, & Handschuh, 2020).

Chapter 5 reported the experiments carried out to evaluate the proposed approach. It showed that the Distributional Navigation model performs better than syntactic-oriented approaches for more knowledge-demanding datasets, and that the composite approach, which employs different methods for solving syntactic and semantic entailment pairs, outperforms single-technique approaches, be it syntactic-only, like the transformation-based and classification-based baselines, or semantic-only, like the DGN model, besides being less dependent on training data. A qualitative evaluation assessed the correctness of the justifications, showing that, despite the challenges imposed by some datasets, especially those fully composed of real world data, the proposed strategy is a viable way of generating explanations in an unsupervised manner. A further analysis compared different knowledge resources, showing that dictionaries are better for the entailment task than encyclopedic resources, and that lexicons built by experts deliver better recall and slightly higher quality justifications than those built collaboratively by lay users. The publications related to this chapter are (Silva et al., 2018b), (Silva et al., 2019a), and (Silva et al., 2020).

Chapter 6 presented the results of a systematic error analysis, showing that misclassifications are caused by both wrong algorithm decisions and external factors, such as syntactic parser errors or knowledge base incompleteness. The error quantification showed that most errors are caused by misinterpretation of context information or absence of relevant information in the knowledge base,

while the core graph navigation algorithm is the source of only a very small number of errors, showing that it works satisfactorily well when provided with the right inputs.

The work presented in this thesis introduces additions for the text entailment recognition, knowledge extraction and representation, and semantic interpretability areas. The main contributions of this thesis can be summarized as follows:

- A conceptual model for representing natural language definitions in a structured way.
- A methodology and a set of tools for filtering, labeling, and structuring natural language definitions, allowing the automatic conversion of dictionaries or other domain-specific glossaries into RDF graphs.
- A set of four publicly available definition knowledge graphs that can be used as knowledge sources not only in text entailment but also in many other NLP applications.
- A method for traversing such definition knowledge graphs using distributional semantic models for identifying semantic relationships in entailment pairs.
- A method for telling syntactic entailments apart from semantic ones, allowing different methods to be used separately inside the same system, without the need of specialized datasets.
- A complete entailment recognition system that can solve both syntactic and semantic entailments using only the most relevant features of each of them, and explain what the semantic relationship (if any) between the text and the hypothesis is.
- A method for generating natural language explanations using knowledge from external world knowledge bases in an unsupervised manner.
- A study of the characteristics of different lexicons and their impact on inference and language processing tasks, which can inform other NLP areas about their usefulness and usability.

7.1 Hypotheses Confirmation

Recalling the hypotheses presented in Chapter 1 and considering the developments described throughout this thesis, especially the quantitative and qualitative results reported in the evaluation in Chapter 5, the following conclusions can be drawn:

H1: *The use of different methods for addressing different (syntactic or semantic) entailment phenomena increases the accuracy of the overall entailment approach.*

The experiments show that the proposed approach, materialized in the XTE system, outperforms all of the three single-technique baselines for all of the tested datasets. For the RTE3 dataset, XTE shows an increase of 6% in F1-score when compared to the best performing syntactic baseline, and of 25% when compared to the semantic-only approach. For the SICK dataset, the gain is of 4% and 26% regarding the best performing syntactic algorithm and the semantic-only baseline, respectively. For the BPI dataset, XTE shows an increase of 22% in F1-score over the best performing syntactic algorithm, and 6% when compared to the semantic approach. For the GHS dataset, there is a gain of 27% and 7% in relation to the best performing syntactic approach and the semantic algorithm, respectively. These results, therefore, confirm hypothesis H1.

H2: *Solving semantic entailments by searching for the key semantic relationship between T and H in a knowledge graph (a knowledge base structured as a set of concepts linked by semantic relationships) increases the accuracy of the system, especially for world knowledge-demanding datasets.*

As already shown for hypothesis H1, XTE outperforms both syntactic-oriented approaches, but the improvement is even more noticeable for the BPI and GHS datasets, which are the most knowledge-demanding ones: the increase in F1-score is of 22% over the Edit Distance algorithm and of 42% over the Maximum Entropy Classifier model when the BPI dataset is tested, and of 27% over the Maximum Entropy Classifier model and of 34% over the Edit Distance algorithm when the GHS dataset is used, also confirming hypothesis H2.

H3: *Natural language dictionary definitions, extracted from lexical resources, can provide the commonsense knowledge necessary to solve semantic entail-*

ments.

Definitional knowledge, as described in Chapter 2, does not cover all the knowledge requirements for the textual entailment task, but has the potential to cover the largest amount of different semantic relationships between entities. With this in mind, and considering the overall quantitative results, it is possible to affirm that the definition graph satisfactorily addresses the knowledge needs for a large number of semantic entailments. This is supported by the fact that, on average, only around 20% of the false negatives are due to absent or insufficient information in the knowledge graph and, even so, as pointed in Chapter 6, knowledge base incompleteness can have more to do with a specific linguistic resource coverage than with the nature of definitional knowledge itself. Justifications provide further evidence that a considerable number of semantic relationships between the relevant entities in the text and the hypothesis are being found in the graph, confirming hypothesis H3.

H4: *By traversing a definition knowledge graph to find the key semantic relationship between T and H, it is possible to generate a natural language justification from the retrieved path, making the system decision interpretable.*

Given that, as detailed in Chapter 3, nodes in the definition graph usually enclose a self-contained and comprehensible amount of information, the concatenation of the nodes in the retrieved path is enough for generating intelligible sentences. A simple rule-based formatting procedure, which takes into account the definition semantic roles represented by the edges between nodes produces the final justification without the need for supervision. The qualitative evaluation presented in Chapter 5 shows that incorrect justifications are due to a wrong choice of source-target word pairs, which is done at the preprocessing stage, therefore, when provided with the right inputs, the graph traversal routine will always return a path (for positive entailments) that allows the generation of the correct explanation, containing all the steps that led from the text to the hypothesis, confirming hypothesis H4.

7.2 Future Directions

The text entailment approach proposed in this work introduces important contributions, especially regarding the identification of semantic relationships by using external world knowledge and the inclusion of post-hoc explanations as

an interpretability feature. Nevertheless, as evidenced by the error analysis presented in Chapter 6, there are some limitations that could be further explored in future work. A point that deserves attention is the preprocessing stage, where, among other tasks, the source-target word pairs which will be sent as input to the Distributional Graph Navigation model are identified. Finding the right pairs is especially challenging when the sentences are very long and there are multiple candidates due to the large number of semantically related words. Wrong source-target pair selection affects not only the overall system accuracy but also the validity of the justification. Developing better ways to deal with long sentences without being limited to syntactic analysis, filtering the portions in T that are the most relevant in relation to H regarding their meaning so the semantic relationship can be sought for the right concepts is a path worth of investigation.

The coverage of the knowledge bases has also shown to impose limitations on the system accuracy. For performance reasons, knowledge graphs were tested only in isolation, but a combination of resources, increasing the knowledge coverage, could doubtless improve accuracy. Investigating advanced graph storage and querying mechanisms to deal with resources whose dimensions can reach the order of millions of nodes in an efficient manner is another opportunity for future development.

Last but probably most important is the association of the knowledge injection and interpretability features presented in this thesis with more advanced learning techniques, such as the deep neural networks employed by Natural Language Inference models. Such models achieve very high accuracy, but current implementations, as observed in Chapter 2, are very vulnerable to bias in the training data, which sometimes even prevent them from performing actual inference. Their use of external knowledge, which could help to soften the negative effects of biased data, is still very limited. Combining the discovery of semantic relationships through the traversal of knowledge graphs with statistical learning methods could leverage a model's inference capabilities, while also providing inputs for the generation of explanations for making the prediction understandable. This is a promising direction for overcoming the trade-off between accuracy and interpretability, and that could result in models that deliver not only accurate but also intelligible and verifiably reliable predictions.

Appendix A

POS and Non-Terminal Symbols

Lists of part-of-speech (POS) and non-terminal tags from the Penn Treebank, provided by (Taylor, Marcus, & Santorini, 2003).

CC	Coordinating conjunction	TO	Infinitival *to*
CD	Cardinal number	UH	Interjection
DT	Determiner	VB	Verb, base form
EX	Existential there	VBD	Verb, past tense
FW	Foreign word	VBG	Verb, gerund/present participle
IN	Preposition	VBN	Verb, past participle
JJ	Adjective	VBP	Verb, non-3rd person singular present
JJR	Adjective, comparative	VBZ	Verb, 3rd person singular present
JJS	Adjective, superlative	WDT	Wh-determiner
LS	List item marker	WP	Wh-pronoun
MD	Modal	WP$	Possessive *wh*-pronoun
NN	Noun, singular or mass	WRB	Wh-adverb
NNS	Noun, plural	#	Pound sign
NNP	Proper noun, singular	$	Dollar sign
NNPS	Proper noun, plural	.	Sentence-final punctuation
PDT	Predeterminer	,	Comma
POS	Possessive ending	:	Colon, semi-colon
PRP	Personal pronoun	(	Left bracket character
PP$	Possessive pronoun	)	Right bracket character
RB	Adverb	″	Straight double quote
RBR	Adverb, comparative	‘	Left open single quote
RBS	Adverb, superlative	“	Left open double quote
RP	Particle	’	Right close single quote
SYM	Symbol	”	Right close double quote

Table A.1: The Penn Treebank POS tags.

ADJP	Adjective phrase
ADVP	Adverb phrase
NP	Noun phrase
PP	Prepositional phrase
S	Simple declarative clause
SBAR	Subordinate clause
SBARQ	Direct question introduced by *wh*-element
SINV	Declarative sentence with subject-aux inversion
SQ	Yes/no questions and subconstituent of SBARQ excluding *wh*-element
VP	Verb phrase
WHADVP	Wh-adverb phrase
WHNP	Wh-noun phrase
WHPP	Wh-prepositional phrase
X	Constituent of unknown or uncertain category
*	"Understood" subject of infinitive or imperative
0	Zero variant of *that* in subordinate clause
T	Trace of *wh*-constituent

Table A.2: The Penn Treebank non-terminal tags.

Appendix B

RDF Model Properties

List of properties and namespaces for the RDF representations of the definition knowledge graphs.

B.1 Namespaces

Noun and verb entity nodes are identified by *synsets*, a set of synonym words which share the same definition. Synsets can have 1 to n words, and each word is represented by a *rdf:label* element associated to the entity node. Noun and verb entity nodes, as well as role nodes have each their own namespaces, where $<xx>$ stands for the acronym of the lexicon that gave origin to the graph:

- **wn:** WordNet
- **wb:** Webster's Dictionary
- **wt:** Wikitionary
- **wp:** Wikipedia

Similarly, $<Lex>$ in the full namespace URI is one of the four lexicon names: WordNet, Websters, Wikitionary or Wikipedia.

Namespace	Full Namespace URI	Usage
dsr	http://nlp/resources/DefinitionSemanticRoles#	Model properties
<xx>n	http://nlp/resources/synsets/<Lex>NounSynset#	Resources denoting the <Lex> noun entity nodes
<xx>v	http://nlp/resources/synsets/<Lex>VerbSynset#	Resources denoting the <Lex> verb entity nodes
<xx>e	http://nlp/resources/expression/<Lex>Expression#	Resources denoting the definition role nodes, which can range from a single word to a whole sentence

Table B.1: List of namespaces for the RDF graphs.

B.2 Properties

RDF properties for the definition knowledge graphs are named after the definition semantic roles.

Property	Usage
dsr:has_supertype	Links the entity node to a supertype role node
dsr:has_diff_qual	Links a supertype role node to a differentia quality role node
dsr:has_diff_event	Links a supertype role node to a differentia event role node
dsr:at_time	Links a differentia event role node to its event time role node
dsr:at_location	Links a differentia event role node to its event location role node
dsr:has_qual_modif	Links a differentia quality role node to its quality modifier role node
dsr:has_origin_loc	Links a supertype role node to a origin location role node
dsr:has_purpose	Links a supertype role node to a purpose role node
dsr:has_assoc_fact	Links a supertype role node to a associated fact role node
dsr:has_acc_qual	Links a supertype role node to a accessory quality role node
dsr:has_acc_det	Links a supertype role node to a accessory determiner role node

Table B.2: List of properties for the RDF graphs.

References

Agarwal, B., Mittal, N., Bansal, P., & Garg, S. (2015). Sentiment analysis using common-sense and context information. *Computational intelligence and neuroscience*, *2015*, 30.

Agirre, E., Banea, C., Cardie, C., Cer, D., Diab, M., Gonzalez-Agirre, A., ... others (2015). Semeval-2015 task 2: Semantic textual similarity, english, spanish and pilot on interpretability. In *Proceedings of the 9th international workshop on semantic evaluation (SemEval 2015)* (pp. 252–263).

Agirre, E., Gonzalez-Agirre, A., Lopez-Gazpio, I., Maritxalar, M., Rigau, G., & Uria, L. (2015). Ubc: Cubes for english semantic textual similarity and supervised approaches for interpretable sts. In *Proceedings of the 9th international workshop on semantic evaluation (SemEval 2015)* (pp. 178–183).

Allemang, D., & Hendler, J. (2011). *Semantic web for the working ontologist: effective modeling in rdfs and owl*. Elsevier.

Alonso, J. M., Castiello, C., & Mencar, C. (2015). Interpretability of fuzzy systems: Current research trends and prospects. In *Springer handbook of computational intelligence* (pp. 219–237). Springer.

Alonso, J. M., & Magdalena, L. (2011). Special issue on interpretable fuzzy systems. *Information Sciences*, *20*(181), 4331–4339.

Alonso, J. M., Mencar, C., Castiello, C., & Magdalena, L. (2014). Some insights on interpretable fuzzy systems. *Mathware & Soft Computing*, *21*(1), 12–14.

Androutsopoulos, I., & Malakasiotis, P. (2010). A survey of paraphrasing and textual entailment methods. *Journal of Artificial Intelligence Research*, *38*, 135–187.

Baker, C. F., Fillmore, C. J., & Lowe, J. B. (1998). The Berkeley Framenet project. In *Proceedings of the 17th international conference on computational linguistics-volume 1* (pp. 86–90).

Balázs, K., & Kóczy, L. T. (2013). A stochastic model for analyzing the interpretability-accuracy trade-off in interpretable fuzzy systems using nested hyperball structures. In *8th conference of the european society for fuzzy logic and technology, EUSFLAT 2013.*

Banjade, R., Niraula, N. B., Maharjan, N., Rus, V., Stefanescu, D., Lintean, M., & Gautam, D. (2015). Nerosim: A system for measuring and interpreting semantic textual similarity. In *Proceedings of the 9th international workshop on semantic evaluation (SemEval 2015)* (pp. 164–171).

Bar-Haim, R., Dagan, I., Greental, I., Szpektor, I., & Friedman, M. (2007). Semantic inference at the lexical-syntactic level for textual entailment recognition. In *Proceedings of the ACL-PASCAL workshop on textual entailment and paraphrasing* (pp. 131–136).

Baroni, M., Murphy, B., Barbu, E., & Poesio, M. (2010). Strudel: A corpus-based semantic model based on properties and types. *Cognitive Science*, *34*(2), 222–254.

Bentivogli, L., Dagan, I., & Magnini, B. (2017). The recognizing textual entailment challenges: Datasets and methodologies. In *Handbook of linguistic annotation* (pp. 1119–1147). Springer.

Berant, J., Chou, A., Frostig, R., & Liang, P. (2013). Semantic parsing on Freebase from question-answer pairs. In *Proceedings of the 2013 conference on empirical methods in natural language processing* (pp. 1533–1544).

Berg, J. (1982). Aristotle's theory of definition. *ATTI del Convegno Internazionale di Storia della Logica*, 19–30.

Biçici, E. (2015). Rtm-dcu: Predicting semantic similarity with referential translation machines. In *Proceedings of the 9th international workshop on semantic evaluation (SemEval 2015)* (pp. 56–63).

Biran, O., & Cotton, C. (2017). Explanation and justification in machine learning: A survey. In *IJCAI-17 workshop on explainable AI (XAI)* (pp. 8–13).

Bizer, C., Cyganiak, R., & Gauß, T. (2007). The RDF Book Mashup: From web APIs to a web of data. *SFSW*, *248*.

Blei, D. M. (2012). Probabilistic topic models. *Communications of the ACM*, *55*(4), 77–84.

Blei, D. M., & Lafferty, J. D. (2005). Correlated topic models. In *Proceedings of the 18th international conference on neural information processing systems* (pp. 147–154).

Blei, D. M., Ng, A. Y., & Jordan, M. I. (2003). Latent dirichlet allocation. *Journal of Machine Learning Research*, *3*(Jan), 993–1022.

Bollacker, K., Evans, C., Paritosh, P., Sturge, T., & Taylor, J. (2008). Freebase: a collaboratively created graph database for structuring human knowledge. In *Proceedings of the 2008 ACM SIGMOD international conference on management of data* (pp. 1247–1250).

Bos, J., & Markert, K. (2005). Recognising textual entailment with logical inference. In *Proceedings of the conference on human language technology and empirical methods in natural language processing* (pp. 628–635).

Boteanu, A., & Chernova, S. (2015). Solving and explaining analogy questions using semantic networks. In *Twenty-ninth AAAI conference on artificial intelligence.*

Bouchoucha, A., He, J., & Nie, J.-Y. (2013). Diversified query expansion using ConceptNet. In *Proceedings of the 22nd ACM international conference on information & knowledge management* (pp. 1861–1864).

Bovi, C. D., Telesca, L., & Navigli, R. (2015). Large-scale information extraction from textual definitions through deep syntactic and semantic analysis. *Transactions of the Association for Computational Linguistics*, *3*, 529–543.

Bowman, S. R., Angeli, G., Potts, C., & Manning, C. D. (2015). A large annotated corpus for learning natural language inference. In *Conference on empirical methods in natural language processing (EMNLP).*

Braz, R. d. S., Girju, R., Punyakanok, V., Roth, D., & Sammons, M. (2005). An inference model for semantic entailment in natural language. In *Machine learning challenges workshop* (pp. 261–286).

Brinton, C. (2017). A framework for explanation of machine learning decisions. In *IJCAI-17 workshop on explainable AI (XAI)* (pp. 14–18).

Burek, P. (2004). Adoption of the classical theory of definition to ontology modeling. In *International conference on artificial intelligence: Methodology, systems, and applications* (pp. 1–10).

Cabrio, E., Cojan, J., Palmero Aprosio, A., Magnini, B., Lavelli, A., & Gandon, F. (2012). QAKiS: an open domain QA system based on relational patterns. In *International semantic web conference 2012 posters & demonstrations track.*

Cabrio, E., & Magnini, B. (2014). Decomposing semantic inferences. *LiLT (Linguistic Issues in Language Technology)*, *9*.

Calzolari, N. (1991). Acquiring and representing semantic information in a lexical knowledge base. In *Workshop of siglex (special interest group within acl on the lexicon)* (pp. 235–243).

Cambria, E., Havasi, C., & Hussain, A. (2012). SenticNet 2: A semantic and affective resource for opinion mining and sentiment analysis. In *Twenty-fifth international FLAIRS conference.*

Cambria, E., & White, B. (2014). Jumping nlp curves: A review of natural language processing research. *IEEE Computational intelligence magazine*, *9*(2), 48–57.

Camburu, O.-M., Rocktäschel, T., Lukasiewicz, T., & Blunsom, P. (2018). e-SNLI: natural language inference with natural language explanations. In *Advances in neural information processing systems* (pp. 9539–9549).

Canbek, N. G., & Mutlu, M. E. (2016). On the track of artificial intelligence: Learning with intelligent personal assistants. *Journal of Human Sciences*, *13*(1), 592–601.

Central Intelligence Agency. (2009). *The CIA world factbook 2010.* Skyhorse Publishing Inc.

Chang, J., Gerrish, S., Wang, C., Boyd-Graber, J. L., & Blei, D. M. (2009). Reading tea leaves: How humans interpret topic models. In *Advances in neural information processing systems* (pp. 288–296).

Chen, D., & Manning, C. (2014). A fast and accurate dependency parser using neural networks. In *Proceedings of the 2014 conference on empirical methods in natural language processing (emnlp)* (pp. 740–750).

Chen, Q., Zhu, X., Ling, Z.-H., Inkpen, D., & Wei, S. (2018). Neural natural language inference models enhanced with external knowledge. In *Proceedings of the 56th annual meeting of the association for computational linguistics (volume 1: Long papers)* (pp. 2406–2417).

Chen, Q., Zhu, X., Ling, Z.-H., Wei, S., Jiang, H., & Inkpen, D. (2017). Enhanced LSTM for natural language inference. In *Proceedings of the 55th annual meeting of the association for computational linguistics (volume 1: Long papers)* (pp. 1657–1668).

Chklovski, T., & Pantel, P. (2004). Verbocean: Mining the web for fine-grained semantic verb relations. In *Proceedings of the 2004 conference on empirical methods in natural language processing* (pp. 33–40).

Chowdhury, G. G. (2003). Natural language processing. *Annual review of information science and technology*, *37*(1), 51–89.

Clark, P., Fellbaum, C., & Hobbs, J. (2008). Using and extending WordNet to support question-answering. In *Proceedings of the 4th global wordnet conference (gwc'08).*

Clark, P., & Harrison, P. (2009). An inference-based approach to recognizing entailment. In *Proceedings of the text analysis conference TAC.*

Clark, P., Murray, W. R., Thompson, J., Harrison, P., Hobbs, J., & Fellbaum, C. (2007). On the role of lexical and world knowledge in rte3. In *Proceedings of the ACL-PASCAL workshop on textual entailment and paraphrasing* (pp. 54–59).

Conde-Clemente, P., Alonso, J. M., & Trivino, G. (2013). Interpretable fuzzy system allowing to be framed in a profile photo through linguistic expressions. In *8th conference of the european society for fuzzy logic and technology, EUSFLAT 2013* (pp. 463–468).

Copestake, A. (1991). The LKB: a system for representing lexical information extracted from machine-readable dictionaries. In *Proceedings of the acquilex workshop on default inheritance in the lexicon, cambridge.*

Cowie, J., & Wilks, Y. (2000). Information extraction. *Handbook of Natural Language Processing*, *56*, 241–260.

Dagan, I., Dolan, B., Magnini, B., & Roth, D. (2009). Recognizing textual entailment: Rational, evaluation and approaches. *Natural Language Engineering*, *15*(4).

Dagan, I., & Glickman, O. (2004). Probabilistic textual entailment: Generic applied modeling of language variability. *Learning Methods for Text Understanding and Mining*, *2004*, 26–29.

Dagan, I., Glickman, O., & Magnini, B. (2006). The PASCAL recognising textual entailment challenge. In *Machine learning challenges: evaluating predictive uncertainty, visual object classification, and recognising textual entailment* (pp. 177–190). Springer.

Dagan, I., Roth, D., Sammons, M., & Zanzotto, F. M. (2013). Recognizing textual entailment: Models and applications. *Synthesis Lectures on Human Language Technologies*, *6*(4), 1–220.

Damljanovic, D., Agatonovic, M., & Cunningham, H. (2011). FREyA: An interactive way of querying linked data using natural language. In *Extended semantic web conference* (pp. 125–138).

Datta, A., Sen, S., & Zick, Y. (2017). Algorithmic transparency via quantitative input influence. In *Transparent data mining for big and small data* (pp. 71–94). Springer.

Denkowski, M., & Lavie, A. (2014). Meteor universal: Language specific translation evaluation for any target language. In *Proceedings of the ninth workshop on statistical machine translation* (pp. 376–380).

Dolan, W., Vanderwende, L., & Richardson, S. D. (1993). Automatically deriving structured knowledge bases from on-line dictionaries. In *Proceedings of the first conference of the pacific association for computational linguistics* (pp. 5–14).

Dong, L., Wei, F., Zhou, M., & Xu, K. (2015). Question answering over Freebase with multi-column convolutional neural networks. In *Proceedings of the 53rd annual meeting of the association for computational linguistics and the 7th international joint conference on natural language processing (volume 1: Long papers)* (pp. 260–269).

Doshi-Velez, F., & Kim, B. (2017). A roadmap for a rigorous science of interpretability. *arXiv preprint arXiv:1702.08608*.

Duan, Y., Edwards, J. S., & Dwivedi, Y. K. (2019). Artificial intelligence for decision making in the era of Big Data – evolution, challenges and research agenda. *International Journal of Information Management*, *48*, 63–71.

Faralli, S., & Navigli, R. (2013). A java framework for multilingual definition and hypernym extraction. In *Proceedings of the 51st annual meeting of the association for computational linguistics: System demonstrations* (pp. 103–108).

Fellbaum, C. (1998). *Wordnet*. Wiley Online Library.

Ferrucci, D., Brown, E., Chu-Carroll, J., Fan, J., Gondek, D., Kalyanpur, A. A., ... Welty, C. (2010). Building Watson: An overview of the DeepQA project. *AI magazine*, *31*(3), 59–79.

Firth, J. R. (1957). A synopsis of linguistic theory 1930-1955. *Studies in Linguistic Analysis (Special volume of the Philological Society)*, *1952-59*, 1-32.

Fowler, A., Hauser, B., Hodges, D., Niles, I., Novischi, A., & Stephan, J. (2005). Applying COGEX to recognize textual entailment. In *Proceedings of the PASCAL challenges workshop on recognising textual entailment* (pp. 69–72).

Freitas, A., Curry, E., & O'Riain, S. (2012). A distributional approach for terminological semantic search on the linked data web. In *Proceedings of the 27th annual ACM symposium on applied computing* (pp. 384–391).

Freitas, A., da Silva, J. C. P., Curry, E., & Buitelaar, P. (2014). A distributional semantics approach for selective reasoning on commonsense graph knowledge bases. In *International conference on applications of natural language to data bases/information systems* (pp. 21–32).

Fyshe, A., Talukdar, P. P., Murphy, B., & Mitchell, T. M. (2014). Interpretable semantic vectors from a joint model of brain-and text-based meaning. In *Proceedings of the association for computational linguistics meeting* (Vol. 2014, pp. 489–499).

Fyshe, A., Wehbe, L., Talukdar, P. P., Murphy, B., & Mitchell, T. M. (2015). A compositional and interpretable semantic space. In *Proceedings of the 2015 conference of the north american chapter of the association for computational linguistics: Human language technologies* (pp. 32–41).

Gabrilovich, E., & Markovitch, S. (2007). Computing semantic relatedness using wikipedia-based explicit semantic analysis. In *IJCAI* (Vol. 7, pp. 1606–1611).

Ganitkevitch, J., Van Durme, B., & Callison-Burch, C. (2013). PPDB: The paraphrase database. In *Proceedings of the 2013 conference of the north american chapter of the association for computational linguistics: Human language technologies* (pp. 758–764).

García-Silva, A., Szomszor, M., Alani, H., & Corcho, O. (2009). Preliminary results in tag disambiguation using DBpedia. In *Proceedings of the first international workshop collective knowledge capturing and representation (CKCaR09).*

Ghahramani, Z. (2015). Probabilistic machine learning and artificial intelligence. *Nature*, *521*(7553), 452–459.

Glickman, O., & Dagan, I. (2005). A probabilistic setting and lexical coocurrence model for textual entailment. In *Proceedings of the ACL workshop on empirical modeling of semantic equivalence and entailment* (pp. 43–48).

Gomaa, W. H., Fahmy, A. A., et al. (2013). A survey of text similarity approaches. *International Journal of Computer Applications*, *68*(13), 13–18.

Gong, Y., Luo, H., & Zhang, J. (2017). Natural language inference over interaction space. *arXiv preprint arXiv:1709.04348*.

Goodman, B., & Flaxman, S. (2016). Eu regulations on algorithmic decision-making and a right to explanation. In *ICML workshop on human interpretability in machine learning (WHI 2016)*.

Granger, E. H. (1984). Aristotle on genus and differentia. *Journal of the History of Philosophy*, *22*(1), 1–23.

Graves, A., & Schmidhuber, J. (2005). Framewise phoneme classification with bidirectional lstm and other neural network architectures. *Neural networks*, *18*(5-6), 602–610.

Guarino, N., & Welty, C. (2002). Evaluating ontological decisions with OntoClean. *Communications of the ACM*, *45*(2), 61–65.

Gunning, D. (2017). Explainable artificial intelligence (XAI). *Defense Advanced Research Projects Agency (DARPA)*.

Gururangan, S., Swayamdipta, S., Levy, O., Schwartz, R., Bowman, S., & Smith, N. A. (2018). Annotation artifacts in natural language inference data. In *Proceedings of the 2018 conference of the north american chapter of the association for computational linguistics: Human language technologies, volume 2 (short papers)* (pp. 107–112).

Hänig, C., Remus, R., & De La Puente, X. (2015). Exb themis: Extensive feature extraction from word alignments for semantic textual similarity. In *Proceedings of the 9th international workshop on semantic evaluation (SemEval 2015)* (pp. 264–268).

Harabagiu, S., Hickl, A., & Lacatusu, F. (2007). Satisfying information needs with multi-document summaries. *Information Processing & Management*, *43*(6), 1619–1642.

Harmeling, S. (2009). Inferring textual entailment with a probabilistically sound calculus. *Natural Language Engineering*, *15*(4), 459–477.

Hassan, B., AbdelRahman, S., & Bahgat, R. (2015). Fcicu: The integration between sense-based kernel and surface-based methods to measure semantic textual similarity. In *Proceedings of the 9th international workshop on semantic evaluation (SemEval 2015)* (pp. 154–158).

Heilman, M., & Smith, N. A. (2010). Tree edit models for recognizing textual entailments, paraphrases, and answers to questions. In *Human language technologies: The 2010 annual conference of the north american chapter of the association for computational linguistics* (pp. 1011–1019).

Herrera, J., Penas, A., & Verdejo, F. (2006). Textual entailment recognition based on dependency analysis and WordNet. In *Machine learning chal-*

lenges. evaluating predictive uncertainty, visual object classification, and recognising textual entailment (pp. 231–239). Springer.

Hickl, A., & Bensley, J. (2007). A discourse commitment-based framework for recognizing textual entailment. In *Proceedings of the ACL-PASCAL workshop on textual entailment and paraphrasing* (pp. 171–176).

Hirasawa, T., Aoyama, K., Tanimoto, T., Ishihara, S., Shichijo, S., Ozawa, T., ... Tada, T. (2018). Application of artificial intelligence using a convolutional neural network for detecting gastric cancer in endoscopic images. *Gastric Cancer*, *21*(4), 653–660.

Hoffart, J., Yosef, M. A., Bordino, I., Fürstenau, H., Pinkal, M., Spaniol, M., ... Weikum, G. (2011). Robust disambiguation of named entities in text. In *Proceedings of the conference on empirical methods in natural language processing* (pp. 782–792).

Hofmann, T. (1999). Probabilistic latent semantic indexing. In *Proceedings of the twenty-second annual international sigir conference.*

Hosny, A., Parmar, C., Quackenbush, J., Schwartz, L. H., & Aerts, H. J. (2018). Artificial intelligence in radiology. *Nature Reviews Cancer*, *18*(8), 500–510.

Hsu, M.-H., Tsai, M.-F., & Chen, H.-H. (2008). Combining WordNet and ConceptNet for automatic query expansion: a learning approach. In *Asia information retrieval symposium* (pp. 213–224).

Hua, W., Song, Y., Wang, H., & Zhou, X. (2013). Identifying users' topical tasks in web search. In *Proceedings of the sixth acm international conference on web search and data mining* (pp. 93–102).

Hulpus, I., Hayes, C., Karnstedt, M., & Greene, D. (2013). Unsupervised graph-based topic labelling using DBpedia. In *Proceedings of the sixth acm international conference on web search and data mining* (pp. 465–474).

Iftene, A., & Balahur-Dobrescu, A. (2007). Hypothesis transformation and semantic variability rules used in recognizing textual entailment. In *Proceedings of the ACL-PASCAL workshop on textual entailment and paraphrasing* (pp. 125–130).

Im, J., & Cho, S. (2017). Distance-based self-attention network for natural language inference. *arXiv preprint arXiv:1712.02047*.

Jimenez, S., Dueñas, G., Baquero, J., & Gelbukh, A. (2014). UNAL-NLP: Combining soft cardinality features for semantic textual similarity, relat-

edness and entailment. In *Proceedings of the 8th international workshop on semantic evaluation (semeval 2014)* (pp. 732–742).

Jolliffe, I. T. (1986). Principal component analysis and factor analysis. In *Principal component analysis* (pp. 115–128). Springer.

Jurafsky, D., & Martin, J. H. (2000). *Speech and language processing: An introduction to natural language processing, computational linguistics and speech recognition*. Prentice-Hall.

Karumuri, S., Vuggumudi, V. K. R., & Chitirala, S. C. R. (2015). Umduluth-blueteam: Svcsts-a multilingual and chunk level semantic similarity system. In *Proceedings of the 9th international workshop on semantic evaluation (SemEval 2015)* (pp. 107–110).

Kim, B., Shah, J. A., & Doshi-Velez, F. (2015). Mind the gap: A generative approach to interpretable feature selection and extraction. In *Advances in neural information processing systems* (pp. 2260–2268).

Kim, S., Kang, I., & Kwak, N. (2019). Semantic sentence matching with densely-connected recurrent and co-attentive information. In *Proceedings of the aaai conference on artificial intelligence* (Vol. 33, pp. 6586–6593).

Kipper, K., Korhonen, A., Ryant, N., & Palmer, M. (2006). Extending VerbNet with novel verb classes. In *LREC* (pp. 1027–1032).

Klema, V., & Laub, A. (1980). The singular value decomposition: Its computation and some applications. *IEEE Transactions on automatic control*, *25*(2), 164–176.

Kobilarov, G., Scott, T., Raimond, Y., Oliver, S., Sizemore, C., Smethurst, M., ... Lee, R. (2009). Media meets semantic web–how the BBC uses DBpedia and linked data to make connections. In *European semantic web conference* (pp. 723–737).

Kordjamshidi, P., Moens, M.-F., & van Otterlo, M. (2010). Spatial role labeling: Task definition and annotation scheme. In *Proceedings of the seventh conference on international language resources and evaluation (lrec'10)* (pp. 413–420).

Kotov, A., & Zhai, C. (2012). Tapping into knowledge base for ConceptNet feedback: leveraging ConceptNet to improve search results for difficult queries. In *Proceedings of the fifth ACM international conference on web search and data mining* (pp. 403–412).

Kotsiantis, S. B., Zaharakis, I., & Pintelas, P. (2007). Supervised machine learning: A review of classification techniques. *Emerging artificial intelligence applications in computer engineering*, *160*, 3–24.

Kouylekov, M., & Magnini, B. (2005). Recognizing textual entailment with tree edit distance algorithms. In *Proceedings of the first challenge workshop recognising textual entailment* (pp. 17–20).

Kuang, C. (2017). Can A.I. be taught to explain itself? *The New York Times Magazine*. Retrieved from `https://www.nytimes.com/2017/11/21/magazine/can-ai-be-taught-to-explain-itself.html`

Lakkaraju, H., Bach, S. H., & Leskovec, J. (2016). Interpretable decision sets: A joint framework for description and prediction. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining* (pp. 1675–1684).

Lehmann, J., Isele, R., Jakob, M., Jentzsch, A., Kontokostas, D., Mendes, P. N., ... Bizer, C. (2015). DBpedia–a large-scale, multilingual knowledge base extracted from wikipedia. *Semantic Web*, *6*(2), 167–195.

Lenci, A. (2008). Distributional semantics in linguistic and cognitive research. *Italian journal of linguistics*, *20*(1), 1–31.

Letham, B., Rudin, C., McCormick, T. H., & Madigan, D. (2013). An interpretable stroke prediction model using rules and bayesian analysis. In *AAAI (late-breaking developments)*.

Levy, R., & Andrew, G. (2006). Tregex and Tsurgeon: tools for querying and manipulating tree data structures. In *Proceedings of the fifth international conference on language resources and evaluation (LREC 2006)* (pp. 2231–2234).

Lin, C.-Y., & Och, F. J. (2004). Automatic evaluation of machine translation quality using longest common subsequence and skip-bigram statistics. In *Proceedings of the 42nd annual meeting on association for computational linguistics* (p. 605).

Lin, D., & Pantel, P. (2001). DIRT - discovery of inference rules from text. In *Proceedings of the seventh ACM SIGKDD international conference on knowledge discovery and data mining* (pp. 323–328).

Lipton, Z. C. (2016). The mythos of model interpretability. *arXiv preprint arXiv:1606.03490*.

Liu, H., & Singh, P. (2004). Conceptneta practical commonsense reasoning tool-kit. *BT technology journal*, *22*(4), 211–226.

Liu, Y., Sun, C., Lin, L., & Wang, X. (2016). Learning natural language inference using bidirectional LSTM model and inner-attention. *arXiv preprint arXiv:1605.09090*.

Lloyd, A. C. (1962). Genus, species and ordered series in Aristotle. *Phronesis*, 67–90.

Lloyd, J. R., Duvenaud, D. K., Grosse, R. B., Tenenbaum, J. B., & Ghahramani, Z. (2014). Automatic construction and natural-language description of nonparametric regression models. In *Aaai* (pp. 1242–1250).

LoBue, P., & Yates, A. (2011). Types of common-sense knowledge needed for recognizing textual entailment. In *Proceedings of the 49th annual meeting of the association for computational linguistics: human language technologies* (pp. 329–334).

Lopez, V., Fernández, M., Motta, E., & Stieler, N. (2012). Poweraqua: Supporting users in querying and exploring the semantic web. *Semantic Web*, *3*(3), 249–265.

Ma, H., Yang, H., Lyu, M. R., & King, I. (2008). Sorec: social recommendation using probabilistic matrix factorization. In *Proceedings of the 17th acm conference on information and knowledge management* (pp. 931–940).

MacCartney, B., Galley, M., & Manning, C. D. (2008). A phrase-based alignment model for natural language inference. In *Proceedings of the conference on empirical methods in natural language processing* (pp. 802–811).

MacCartney, B., & Manning, C. D. (2007). Natural logic for textual inference. In *Proceedings of the ACL-PASCAL workshop on textual entailment and paraphrasing* (pp. 193–200).

Magnini, B., Zanoli, R., Dagan, I., Eichler, K., Neumann, G., Noh, T.-G., ... Levy, O. (2014). The excitement open platform for textual inferences. In *Acl (system demonstrations)* (pp. 43–48).

Manning, C. D., Surdeanu, M., Bauer, J., Finkel, J. R., Bethard, S., & McClosky, D. (2014). The Stanford CoreNLP natural language processing toolkit. In *ACL (System Demonstrations)* (pp. 55–60).

Marelli, M., Menini, S., Baroni, M., Bentivogli, L., Bernardi, R., & Zamparelli, R. (2014). A SICK cure for the evaluation of compositional distributional semantic models. In *LREC* (pp. 216–223).

Màrquez, L., Carreras, X., Litkowski, K. C., & Stevenson, S. (2008). Semantic role labeling: an introduction to the special issue. *Computational linguistics*, *34*(2), 145–159.

Mehdad, Y., Negri, M., Cabrio, E., Kouylekov, M., & Magnini, B. (2009). EDITS: An open source framework for recognizing textual entailment. *Proceedings of the Text Analysis Conference TAC*.

Mencar, C., Castiello, C., Cannone, R., & Fanelli, A. M. (2011). Interpretability assessment of fuzzy knowledge bases: A cointension based approach. *International Journal of Approximate Reasoning*, *52*(4), 501–518.

Mencar, C., & Fanelli, A. M. (2008). Interpretability constraints for fuzzy information granulation. *Information Sciences*, *178*(24), 4585–4618.

Mencar, C., Lucarelli, M., Castiello, C., & Fanelli, A. M. (2013). Design of strong fuzzy partitions from cuts. In *8th conference of the european society for fuzzy logic and technology, EUSFLAT 2013.*

Mendes, P. N., Jakob, M., García-Silva, A., & Bizer, C. (2011). DBpedia spotlight: shedding light on the web of documents. In *Proceedings of the 7th international conference on semantic systems* (pp. 1–8).

Mesnil, G., Dauphin, Y., Yao, K., Bengio, Y., Deng, L., Hakkani-Tur, D., ... Zweig, G. (2015). Using recurrent neural networks for slot filling in spoken language understanding. *IEEE/ACM Transactions on Audio, Speech and Language Processing (TASLP)*, *23*(3), 530–539.

Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., & Dean, J. (2013). Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems* (pp. 3111–3119).

Miller, T., Howe, P., & Sonenberg, L. (2017). Explainable ai: Beware of inmates running the asylum. In *IJCAI-17 workshop on explainable AI (XAI)* (pp. 36–42).

Murphy, B., Talukdar, P., & Mitchell, T. (2012). Learning effective and interpretable semantic models using non-negative sparse embedding. *Proceedings of COLING 2012*, 1933–1950.

Niklaus, C., Bermeitinger, B., Handschuh, S., & Freitas, A. (2016, December). A sentence simplification system for improving relation extraction. In *Proceedings of coling 2016, the 26th international conference on computational linguistics: System demonstrations* (pp. 170–174). Osaka, Japan: The COLING 2016 Organizing Committee.

Nilsson, N. J. (2014). *Principles of artificial intelligence*. Morgan Kaufmann.

Palmer, M., Gildea, D., & Kingsbury, P. (2005). The proposition bank: An annotated corpus of semantic roles. *Computational Linguistics*, *31*(1), 71-106.

Palmer, M., Gildea, D., & Xue, N. (2010). Semantic role labeling. *Synthesis Lectures on Human Language Technologies*, *3*(1), 1–103.

Pancho, D. P., Alonso, J. M., Alcalá-Fdez, J., & Magdalena, L. (2013). Interpretability analysis of fuzzy association rules supported by fingrams.

In *8th conference of the european society for fuzzy logic and technology, EUSFLAT 2013.*

Papineni, K., Roukos, S., Ward, T., & Zhu, W.-J. (2002). Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting on association for computational linguistics* (pp. 311–318).

Parikh, A., Täckström, O., Das, D., & Uszkoreit, J. (2016). A decomposable attention model for natural language inference. In *Proceedings of the 2016 conference on empirical methods in natural language processing* (pp. 2249–2255).

Pawlik, M., & Augsten, N. (2016). Tree edit distance: Robust and memory-efficient. *Information Systems*, *56*, 157–173.

Pennington, J., Socher, R., & Manning, C. D. (2014). GloVe: Global vectors for word representation. In *EMNLP* (Vol. 14, pp. 1532–1543).

Poliak, A., Naradowsky, J., Haldar, A., Rudinger, R., & Van Durme, B. (2018). Hypothesis only baselines in natural language inference. In *Proceedings of the seventh joint conference on lexical and computational semantics* (pp. 180–191).

Poplin, R., Varadarajan, A. V., Blumer, K., Liu, Y., McConnell, M. V., Corrado, G. S., ... Webster, D. R. (2018). Prediction of cardiovascular risk factors from retinal fundus photographs via deep learning. *Nature Biomedical Engineering*, 158–164.

Punyakanok, V., Roth, D., & Yih, W.-t. (2005). The necessity of syntactic parsing for semantic role labeling. In *IJCAI* (Vol. 5, pp. 1117–1123).

Raina, R., Ng, A. Y., & Manning, C. D. (2005). Robust textual inference via learning and abductive reasoning. In *AAAI* (pp. 1099–1105).

Ramage, D., Manning, C. D., & Dumais, S. (2011). Partially labeled topic models for interpretable text mining. In *Proceedings of the 17th ACM SIGKDD international conference on knowledge discovery and data mining* (pp. 457–465).

Recski, G. (2016). Building concept graphs from monolingual dictionary entries. In *Proceedings of the tenth international conference on language resources and evaluation (LREC 2016).* European Language Resources Association (ELRA).

Ribeiro, M. T., Singh, S., & Guestrin, C. (2016). Why should I trust you?: Explaining the predictions of any classifier. In *Proceedings of the 22nd*

ACM SIGKDD international conference on knowledge discovery and data mining (pp. 1135–1144).

Riid, A., & Sarv, M. (2013). Determination of regional variants in the versification of Estonian folksongs using an interpretable fuzzy rule-based classifier. In *8th conference of the european society for fuzzy logic and technology, EUSFLAT 2013.*

Robertson, S. (2004). Understanding inverse document frequency: on theoretical arguments for IDF. *Journal of documentation.*

Rocktäschel, T., Grefenstette, E., Hermann, K. M., Kočiskỳ, T., & Blunsom, P. (2016). Reasoning about entailment with neural attention. In *Proceedings of the international conference on learning representations.*

Russell, S., & Norvig, P. (2010). *Artificial intelligence: A modern approach* (3rd ed.). Prentice Hall.

Sales, J. E., Souza, L., Barzegar, S., Davis, B., Freitas, A., & Handschuh, S. (2018). Indra: A word embedding and semantic relatedness server. In *Proceedings of the eleventh international conference on language resources and evaluation (LREC 2018).* Miyazaki, Japan: European Language Resources Association (ELRA).

Salton, G. (1971). The SMART system. *Retrieval Results and Future Plans.*

Sammons, M., Vydiswaran, V., & Roth, D. (2010). Ask not what textual entailment can do for you... In *Proceedings of the 48th annual meeting of the association for computational linguistics* (pp. 1199–1208).

Sammons, M., Vydiswaran, V. V., Vieira, T., Johri, N., Chang, M.-W., Goldwasser, D., ... Roth, D. (2009). Relation alignment for textual entailment recognition. In *Proceedings of the text analysis conference TAC.*

Shen, W., Wang, J., Luo, P., & Wang, M. (2012). Linden: linking named entities with knowledge base via semantic knowledge. In *Proceedings of the 21st international conference on world wide web* (pp. 449–458).

Shen, W., Wang, J., Luo, P., & Wang, M. (2013). Linking named entities in tweets with knowledge base via user interest modeling. In *Proceedings of the 19th ACM SIGKDD international conference on knowledge discovery and data mining* (pp. 68–76).

Silva, V. S., Freitas, A., & Handschuh, S. (2018a). Building a knowledge graph from natural language definitions for interpretable text entailment recognition. In *Proceedings of the eleventh international conference on language resources and evaluation (LREC 2018).*

Silva, V. S., Freitas, A., & Handschuh, S. (2018b). Recognizing and justifying text entailment through distributional navigation on definition graphs. In *Thirty-second AAAI conference on artificial intelligence (AAAI-18)*.

Silva, V. S., Freitas, A., & Handschuh, S. (2019a). Exploring knowledge graphs in an interpretable composite approach for text entailment. In *Thirty-third AAAI conference on artificial intelligence (AAAI-19)*.

Silva, V. S., Freitas, A., & Handschuh, S. (2019b). On the semantic interpretability of artificial intelligence models. *arXiv preprint arXiv:1907.04105*.

Silva, V. S., Freitas, A., & Handschuh, S. (2020). XTE: Explainable text entailment. *arXiv preprint arXiv:2009.12431*.

Silva, V. S., Handschuh, S., & Freitas, A. (2016). Categorization of semantic roles for dictionary definitions. In *Cognitive aspects of the lexicon (CogALex-V), workshop at COLING 2016* (pp. 176–184).

Simonite, T. (2018). When it comes to gorillas, Google Photos remains blind. *Wired*. Retrieved from `https://www.wired.com/story/when-it-comes-to-gorillas-google-photos-remains-blind/`

Song, Y., Wang, H., Wang, Z., Li, H., & Chen, W. (2011). Short text conceptualization using a probabilistic knowledgebase. In *Proceedings of the twenty-second international joint conference on artificial intelligence* (pp. 2330–2336).

Stadler, C., Lehmann, J., Höffner, K., & Auer, S. (2012). LinkedGeoData: A core for a web of spatial open data. *Semantic Web*, *3*(4), 333–354.

Stern, A., & Dagan, I. (2011). A confidence model for syntactically-motivated entailment proofs. In *Proceedings of the international conference recent advances in natural language processing 2011* (pp. 455–462).

Suchanek, F. M., Kasneci, G., & Weikum, G. (2007). YAGO: a core of semantic knowledge. In *Proceedings of the 16th international conference on world wide web* (pp. 697–706).

Sundermeyer, M., Schlüter, R., & Ney, H. (2012). Lstm neural networks for language modeling. In *Thirteenth annual conference of the international speech communication association*.

Swartz, N. (1997). Definitions, dictionaries, and meanings. *Simon Fraser University*. Retrieved from `http://www.sfu.ca/~swartz/definitions.htm`

Tas, O., & Kiyani, F. (2007). A survey automatic text summarization. *PressAcademia Procedia*, *5*(1), 205–213.

Tatu, M., Iles, B., Slavick, J., Novischi, A., & Moldovan, D. (2006). Cogex at the second recognizing textual entailment challenge. In *Proceedings of the second PASCAL challenges workshop on recognising textual entailment* (pp. 104–109).

Taylor, A., Marcus, M., & Santorini, B. (2003). The penn treebank: an overview. In *Treebanks* (pp. 5–22). Springer.

Thorne, J., Vlachos, A., Christodoulopoulos, C., & Mittal, A. (2019). Generating token-level explanations for natural language inference. In *Proceedings of the 2019 conference of the north american chapter of the association for computational linguistics: Human language technologies, volume 1 (long and short papers)* (pp. 963–969).

Tsai, A. C.-R., Wu, C.-E., Tsai, R. T.-H., & Hsu, J. Y.-j. (2013). Building a concept-level sentiment dictionary based on commonsense knowledge. *IEEE Intelligent Systems*, *28*(2), 22–30.

Turney, P. D., & Pantel, P. (2010). From frequency to meaning: Vector space models of semantics. *Journal of artificial intelligence research*, *37*, 141–188.

Unger, C., Bühmann, L., Lehmann, J., Ngonga Ngomo, A.-C., Gerber, D., & Cimiano, P. (2012). Template-based question answering over RDF data. In *Proceedings of the 21st international conference on world wide web* (pp. 639–648).

Usbeck, R., Ngomo, A.-C. N., Röder, M., Gerber, D., Coelho, S. A., Auer, S., & Both, A. (2014). AGDISTIS-graph-based disambiguation of named entities using linked data. In *International semantic web conference* (pp. 457–471).

Vellido Alcacena, A., Martin Guerrero, J. D., & Lisboa, P. J. (2012). Making machine learning models interpretable. In *Proceedings of the european symposium on artificial neural networks, computational intelligence and machine learning (ESANN 2012)* (pp. 163–172).

Voorhees, E. M. (2008). Contradictions and justifications: Extensions to the textual entailment task. In *46th annual meeting of the association for computational linguistics: Human language technologies (ACL 2008)*.

Vossen, P. (1991). Converting data from a lexical database to a knowledge base. *Esprit BRA-3030 ACQUILEX Working Paper No 27*.

Vossen, P. (1992). The automatic construction of a knowledge base from dictionaries: a combination of techniques. In *Euralex* (Vol. 92, pp. 311–326).

Vossen, P., & Copestake, A. (1994). Untangling definition structure into knowledge representation. In *Inheritance, defaults and the lexicon* (pp. 246–274).

Wang, C., Chakrabarti, K., He, Y., Ganjam, K., Chen, Z., & Bernstein, P. A. (2015). Concept expansion using web tables. In *Proceedings of the 24th international conference on world wide web* (pp. 1198–1208).

Wang, J., Wang, H., Wang, Z., & Zhu, K. Q. (2012). Understanding tables on the web. In *International conference on conceptual modeling* (pp. 141–155).

Wang, P., Wu, Q., Shen, C., Dick, A., & van den Hengel, A. (2018). Fvqa: Fact-based visual question answering. *IEEE transactions on pattern analysis and machine intelligence*, *40*(10), 2413–2427.

Wang, R., & Neumann, G. (2008a). An accuracy-oriented divide-and-conquer strategy for recognizing textual entailment. In *Proceedings of the text analysis conference TAC.*

Wang, R., & Neumann, G. (2008b). A divide-and-conquer strategy for recognizing textual entailment. In *Proceedings of the text analysis conference, gaithersburg, md.*

Wang, S., & Jiang, J. (2016). Learning natural language inference with LSTM. In *Proceedings of the 2016 conference of the north american chapter of the association for computational linguistics: Human language technologies* (pp. 1442–1451).

Wang, X., Kapanipathi, P., Musa, R., Yu, M., Talamadupula, K., Abdelaziz, I., ... Witbrock, M. (2019). Improving natural language inference using external knowledge in the science questions domain. In *Proceedings of the AAAI conference on artificial intelligence* (Vol. 33, pp. 7208–7215).

Wang, X., McCallum, A., & Wei, X. (2007). Topical n-grams: Phrase and topic discovery, with an application to information retrieval. In *Seventh IEEE international conference on data mining (ICDM 2007)* (pp. 697–702).

Wang, Z., Zhao, K., Wang, H., Meng, X., & Wen, J.-R. (2015). Query understanding through knowledge-based conceptualization. In *Twenty-fourth international joint conference on artificial intelligence.*

Williams, A., Nangia, N., & Bowman, S. (2018). A broad-coverage challenge corpus for sentence understanding through inference. In *Proceedings of the 2018 conference of the north american chapter of the association for computational linguistics: Human language technologies, volume 1 (long papers)* (Vol. 1, pp. 1112–1122).

Wishart, D. S., Feunang, Y. D., Guo, A. C., Lo, E. J., Marcu, A., Grant, J. R., ... Wilson, M. (2017). DrugBank 5.0: a major update to the DrugBank database for 2018. *Nucleic acids research*, *46*(D1), D1074–D1082.

Wu, F., & Weld, D. S. (2007). Autonomously semantifying wikipedia. In *Proceedings of the sixteenth acm conference on conference on information and knowledge management* (pp. 41–50).

Wu, F., & Weld, D. S. (2008). Automatically refining the wikipedia infobox ontology. In *Proceedings of the 17th international conference on world wide web* (pp. 635–644).

Wu, W., Li, H., Wang, H., & Zhu, K. Q. (2012). Probase: A probabilistic taxonomy for text understanding. In *Proceedings of the 2012 acm sigmod international conference on management of data* (pp. 481–492).

Yao, X., & Van Durme, B. (2014). Information extraction over structured data: Question answering with Freebase. In *Proceedings of the 52nd annual meeting of the association for computational linguistics (volume 1: Long papers)* (pp. 956–966).

Yih, W.-t., Chang, M.-W., He, X., & Gao, J. (2015). Semantic parsing via staged query graph generation: Question answering with knowledge base. In *Proceedings of the 53rd annual meeting of the association for computational linguistics and the 7th international joint conference on natural language processing (volume 1: Long papers)* (pp. 1321–1331).

Zhang, K., Chen, E., Liu, Q., Liu, C., & Lv, G. (2017). A context-enriched neural network method for recognizing lexical entailment. In *AAAI* (pp. 3127–3134).

Zhang, K., & Shasha, D. (1989). Simple fast algorithms for the editing distance between trees and related problems. *SIAM journal on computing*, *18*(6), 1245–1262.

Zhao, J., Wang, T., Yatskar, M., Ordonez, V., & Chang, K.-W. (2017). Men also like shopping: Reducing gender bias amplification using corpus-level constraints. In *Proceedings of the 2017 conference on empirical methods in natural language processing* (pp. 2979–2989).

Zhao, J., Zhu, T., & Lan, M. (2014). ECNU: One stone two birds: Ensemble of heterogenous measures for semantic relatedness and textual entailment. In *Proceedings of the 8th international workshop on semantic evaluation (semeval 2014)* (pp. 271–277).