RECOGNIZING TEXTUAL ENTAILMENT USING DESCRIPTION LOGIC AND SEMANTIC RELATEDNESS

Reda Siblini

A thesis

in

The Department

of

Computer Science and Software Engineering

Presented in Partial Fulfillment of the Requirements

For the Degree of Doctor of Philosophy

Concordia University

Montréal, Québec, Canada

April 2014

CONCORDIA UNIVERSITY

School of Graduate Studies

This is to certify that the thesis prepared

By: **Mr. Reda Siblini**

Entitled: **Recognizing Textual Entailment using Description Logic and Semantic Relatedness**

and submitted in partial fulfillment of the requirements for the degree of

Doctor of Philosophy (Computer Science)

complies with the regulations of this University and meets the accepted standards with respect to originality and quality.

Signed by the final examining committee:

______________________	Dr. A. Hammad,	Chair
______________________	Dr. H. Saggion,	External Examiner
______________________	Dr. L. Kosseim,	Thesis Supervisor
______________________	Dr. V. Haarslev,	Examiner
______________________	Dr. S. Bergler,	Examiner
______________________	Dr. F. Khendek,	Examiner

Approved ______________________________

Chair of Department or Graduate Program Director

__________ 20 _____ ______________________________

Dr. Christopher Trueman, Dean

Faculty of Engineering and Computer Science

Abstract

Recognizing Textual Entailment using Description Logic and Semantic Relatedness

Reda Siblini, Ph.D.

Concordia University, 2014

Textual entailment (TE) is a relation that holds between two pieces of text where one reading the first piece can conclude that the second is most likely true. Accurate approaches for textual entailment can be beneficial to various natural language processing (NLP) applications such as: question answering, information extraction, summarization, and even machine translation. For this reason, research on textual entailment has attracted a significant amount of attention in recent years. A robust logical-based meaning representation of text is very hard to build, therefore the majority of textual entailment approaches rely on syntactic methods or shallow semantic alternatives. In addition, approaches that do use a logical-based meaning representation, require a large knowledge base of axioms and inference rules that are rarely available. The goal of this thesis is to design an efficient description logic based approach for recognizing textual entailment that uses semantic relatedness information as an alternative to large knowledge base of axioms and inference rules.

In this thesis, we propose a description logic and semantic relatedness approach to textual entailment, where the type of semantic relatedness axioms employed in aligning the description logic representations are used as indicators of textual entailment. In our approach, the text and the hypothesis are first represented in description logic. The representations are enriched with additional semantic knowledge acquired by using the web as a corpus. The hypothesis is then merged into the text representation by learning semantic relatedness axioms on demand and a reasoner is then used to reason over the aligned representation. Finally, the types of axioms employed by the reasoner are used to learn if the text entails the hypothesis or not. To validate our approach we have implemented an RTE system named AORTE, and evaluated its performance on recognizing textual entailment using the fourth recognizing textual entailment challenge. Our approach achieved an accuracy of 68.8% on the two way task and 61.6% on the three way task which ranked the approach as 2^{nd} when compared to the other participating runs in the same challenge. These results show that our description logical based approach can effectively be used to recognize textual entailment.

Acknowledgments

This thesis reflects my long and rewarding research journey, during which I came across many wonderful people to whom I express my thanks.

First, I would like to express my deepest graduate to my supervisor, Dr. Leila Kosseim, whose spirit of quest in regard to research inspired me to freely explore new ideas. This thesis would not have been possible without her persistent support and guidance.

In addition, I wish to extend my appreciation for the members of my doctoral committee, Professors Volker Haarslev, Sabine Bergler, Ferhat Khendek, and Amin Hammad for their valuable time and feedback. I am particularly thankful to Professor Bergler, whose enthusiasm for linguistics is contagious, for her thorough comments on my thesis. I am also grateful to Professor Haarslev, who introduced me and gave me access to his work in Description Logic.

I want to specially thank the external examiner, Professor Horacio Saggion for his participation in my defense committee and for his valuable feedback on my thesis.

Finally, I would like to thank my colleagues at the CLaC Lab, my family, and loved ones for their continuous moral support and encouragement throughout my studies.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Textual entailment (TE) is a relation that holds between two pieces of text where one reading the first piece can conclude that the second is most likely true. Accurate approaches for textual entailment can be beneficial to various natural language processing (NLP) applications such as question answering, information extraction, summarization, and even machine translation. For this reason, research on textual entailment has attracted a significant amount of attention in recent years. This can be clearly observed by the large number of workshops and challenges on textual entailment, and also by the high number of participants in such challenges ([Dagan, Glickman, and Magnini, 2005, Bar Haim, Dagan, Dolan, Ferro, Giampiccolo, Magnini, and Szpektor, 2006, Giampiccolo, Magnini, Dagan, and Dolan, 2007, Giampiccolo, Dang, Magnini, Dagan, and Dolan, 2008, Bentivogli, Dagan, Dang, Giampiccolo, and Magnini, 2009, Bentivogli, Clark, Dagan, Dang, and Giampiccolo, 2010, 2011, Dzikovska, Nielsen, Brew, Leacock, Giampiccolo, Bentivogli, Clark, Dagan, and Dang, 2013]).

In this chapter we introduce the concept of textual entailment and more specifically the problem of automatically *recognizing* textual entailment. Then, we present the motivation of such a task and its usefulness in various natural language processing applications. We follow this discussion with an overview of the thesis and its intended contributions, and we conclude with the thesis organization.

1.1 Problem Statement

Textual entailment is defined as *"a relationship between a coherent text T and a language expression, which is considered as a hypothesis, H. We say that T entails H (H is a consequent of T), if the meaning of H, as interpreted in the context of T, can be inferred by a human from the meaning of T"* [Dagan and Glickman, 2004].

A simple example of textual entailment is:

(T1): *Jurassic Park is a novel written by Michael Crichton and published in 1990.*

T1 entails the following hypothesis (among others):

(H1): *Michael Crichton is an author.*

(H2): *Michael Crichton is a writer.*

(H3): *Michael Crichton is a human being.*

(H4): *Jurassic Park is a book.*

(H5): *Jurassic Park is a fiction.*

(H6): *Jurassic Park is a literary work.*

(H7): *Michael Crichton is a creative writer.*

(H8): *Michael Crichton created the Jurassic Park novel.*

(H9): *Mr. Crichton is the writer of the book Jurassic Park.*

(H10): *Michael Crichton is the author of Jurassic Park, which is a fictional and creative writing.*

Natural language is a complex and expressive communication system. Various properties of natural language attribute to its expressiveness, and make it difficult for computers to recognize textual entailment relations. Each language expression can invoke various linguistic phenomena to create many entailed expressions. The above hypotheses for example, include syntactic phenomena such as passive to active voice construction *(written by Crichton = Crichton wrote)*. They also show examples of semantic phenomena, such as hypernym relations *(writer → human being, novel → fiction, writer → creator)*, synonymy relations *(writer=author)*, or named entity relations *(Mr. Crichton = Michael Crichton)*. In addition, they include discourse phenomena, as in the coreference *(which = Jurassic Park)*, and domain specific knowledge *(writer of literary work = creative writer)*. All these make the recognition of textual entailment a difficult task.

In addition, given very similar hypotheses, it may be possible to conclude entailment for some, but not possible for others as in the following examples:

(U1): *Michael Crichton is the writer of the book and movie Jurassic Park.*

(U2): *Michael Crichton wrote Jurassic Park in the United States.*

(U3): *The Jurassic Park novel which was written by Michael Crichton was published in November 1990.*

We can also have contradictory or incorrect hypothesis, as in the following examples:

(C1): *Michael Crichton did not write Jurassic Park.*

(C2): *Michael Crichton never wrote Jurassic Park.*

(C3): *Michael Crichton wrote the non-fiction book Jurassic Park.*

This further complicates the problem, as even the ability to detect high similarity between a text and a hypothesis does not guarantee that the text entails the hypothesis.

Many researchers have worked on the problem of textual entailment, most focused on recognizing textual entailment, some on generating textual entailment, and others on extracting pairs of textual entailment.

Recognizing textual entailment (RTE) is the task concerned with deciding whether a text entails a hypothesis or not. An example input would be the text (T1) and hypothesis (H1), and the output would be a judgment. For the T1-H1 pair the judgment would be true for entailment. As of today, the bulk of research on TE has focused on the task of recognizing textual entailment. It is usually considered as a classification problem, with the majority of work concerned with a two way classification (entailment or not) [Dagan, Glickman, and Magnini, 2005], [Bar Haim, Dagan, Dolan, Ferro, Giampiccolo, Magnini, and Szpektor, 2006], [Giampiccolo, Magnini, Dagan, and Dolan, 2007]. Other challenges are concerned with a three way classification (entailment, contradiction, unknown) [Giampiccolo, Dang, Magnini, Dagan, and Dolan, 2008], [Bentivogli, Dagan, Dang, Giampiccolo, and Magnini, 2009], [Bentivogli, Clark, Dagan, Dang, and Giampiccolo, 2010], or [Bentivogli, Clark, Dagan, Dang, and Giampiccolo, 2011]. And more recently, there has been an attempt at a five way classification task (bidirectional, entailment, contradiction, irrelevant, not in the domain)[Dzikovska, Nielsen, Brew, Leacock, Giampiccolo, Bentivogli, Clark, Dagan, and Dang, 2013]. A detailed survey of the various approaches on recognizing textual entailment will be given in Section 2.1.

On the other hand, *generating textual entailment* is creating possible hypotheses given a text. For example, given the text (T1) above, the task would be to generate possible hypotheses as in (H1-H10). Recent work on generating textual entailment include the work of [Sonntag and Sacaleanu, 2010] that uses an association based word alignment to generate possible questions hypothesis for a question answering system.

Finally, *extracting textual entailment* consists of extracting pairs of text and hypothesis given a corpus. Recent work on extracting textual entailment include the work of [Lin and Pantel, 2001] that proposed an unsupervised method to discover binary inferences from text (such as if "X is author of Y" that entails "X wrote Y") using similarity of dependency trees paths from a parsed corpus. And the work of [Szpektor, Tanev, Dagan, Coppola, et al., 2005] that proposed a an unsupervised method to extract entailment relations from the web by searching the web for related syntactic entailment templates for a list of verbs, for example the extracted entailment templates for “X Prevent Y” includes “X provides protection against Y”, “X reduces Y”...

Others have focused on *bidirectional textual entailment* or *paraphrasing*, which can be seen as a special case of textual entailment where the text entails the hypothesis and the hypothesis entails the text. Much work has been done on this topic, such as the work of the participants in the International Workshop on Paraphrasing [Inui and Hermjakob, 2003].

Some have worked on *cross-lingual textual entailment*, where the text and the hypothesis are in different languages. Recent challenges that have focused on this task include: Recognizing Inference in TExt challenge [Shima, Kanayama, Lee, Lin, Mitamura, Miyao, Shi, and Takeda, 2011] and the Cross-lingual Textual Entailment challenge [Negri, Marchetti, Mehdad, Bentivogli, and Giampiccolo, 2012].

The scope of this thesis is limited to the problem of recognizing textual entailment and more precisely recognizing monolingual textual entailment.

1.2 Motivation

The ability to recognize textual entailment is a fundamental task to many NLP applications such as question answering, information extraction, information retrieval, machine translation, and summarization.

Question Answering (QA): QA is concerned with answering questions asked in natural language from text. Recognizing textual entailment can be used as part of validating candidate answers of a QA approach. For example: if the question is *“Who painted the Scream?”* and a candidate text includes: *“Norway’s most famous painting, The Scream by Edvard Munch”* then RTE can be used to test whether the question (in affirmative form) *“Edvard Munch painted the Scream”* is a consequent of the text or not. Researchers who have applied RTE in QA, include [Harabagiu and Hickl, 2006] and [Kouylekov, Negri, Magnini, and Coppola, 2007].

Information extraction (IE): The main goal of IE is to extract specific structured information from a collection of documents. Various subtasks of IE focus on entity property extractions, relations extractions, or even scenario production. An example of a specific relation extraction task is *"gathering the profits of companies from company reports"*. The classical approach to such a task is usually trying to fill a set of templates representing the relation that we are interested in. In our example those templates could be: X reported Y in profit, X profits hit Y, X record profits of Y... Recognizing textual entailment can be used in IE by reformulating the needed information as a hypothesis and testing this hypothesis on relevant extracted texts. One researcher who has used this approach successfully for IE is [Kouylekov, 2006].

Information Retrieval (IR): Information retrieval is concerned with retrieving relevant information from a collection of documents. A query must be matched against a large number of documents. Recognizing textual entailment can be applied to information retrieval by evaluating if the document retrieved does entail the information we are looking for. One attempt to use RTE for IR is the work of [Clinchant, Goutte, and Gaussier, 2006].

Machine translation (MT): Machine translation from one natural language to another can also use RTE by either validating an automated translation to a manually created one, or by finding corresponding terms missing from the translation database. Researchers who have used RTE in machine translation, include [Mirkin, Specia, Cancedda, Dagan, Dymetman, and Szpektor, 2009] and [Padó, Galley, Jurafsky, and Manning, 2009].

Summarization: The main goal of automated summarization is to reduce the length of a document by retaining its most important points while avoiding redundancy. Redundancy can be avoided by recognizing textual entailment. Many researchers such as those who participated in the textual entailment search pilot task [Bentivogli, Dagan, Dang, Giampiccolo, and Magnini, 2009] have used RTE for the task of summarization.

The main goal of recognizing textual entailment is to provide a common generic framework targeting semantic inference that can be used by various NLP tasks. In the next section, we will present a brief of overview of the thesis including our main approach to RTE, the methodology we used, and its evaluation.

1.3 Overview of the Thesis

A robust logical-based meaning representation of text is very hard to build, therefore the majority of textual entailment approaches rely on syntactic methods or shallow semantic alternatives. Approaches that do use a logical-based meaning representation (e.g. [Tatu, Iles, Slavick, Novischi, and Moldovan, 2006], [de Salvo Braz, Girju, Punyakanok, Roth, and Sammons, 2006], [Clark and Harrison, 2008]), require a large knowledge base of axioms and inference rules that are rarely available. Our goal in this thesis is to design an efficient description logic based approach for recognizing textual entailment that uses semantic relatedness information as an alternative to the needed large knowledge base of axioms and inference rules, and then evaluate it experimentally using current benchmarks. As a full logical based meaning representation is still a very difficult and challenging problem, we started first with an investigation of the use of description logic as a surface meaning representation of text, knowledge querying in natural language, and semantic relatedness to recognize textual entailment (detailed in Chapter 3). An analysis of the description logic baseline approach helped us identify the main problems of the shallow logical approaches with semantic relatedness to recognizing textual entailment, and led us to investigate the following:

1. How the web can be used as a corpus for enriching a meaning representation of a text.
2. How the semantic relatedness between concepts can be used to learn axioms on demand, as an alternative to using a predefined set of axioms, to recognize textual entailment.
3. If the type of logical statements used to align textual representations can be used as an indicator to textual entailment.

In this thesis, we propose a description logic and semantic relatedness approach to textual entailment, where the type of semantic relatedness axioms employed in aligning the description logic representations are used as indicators of textual entailment. In our approach, the text and the hypothesis are first represented in description logic. The representations are enriched with additional semantic knowledge by using the web as a corpus. The hypothesis is then merged into the text representation by learning semantic relatedness axioms on demand and a reasoner is then used to reason over the aligned representation. Finally, the types of axioms employed by the reasoner are used to learn if the text entails the hypothesis or not.

To validate our approach we have implemented an RTE system named AORTE (described in Chapter 4), and evaluated its performance on recognizing textual entailment using the fourth recognizing textual entailment challenge [Giampiccolo, Dang, Magnini, Dagan, and Dolan, 2008] (described in Section 2.2). The system classified 1000 T-H pairs into the three way task (*Entailment, Contradiction,* or *Unknown*). The evaluation was done automatically, where the classifications returned by the system were compared to the human annotated gold standard and the returned score is the accuracy or the percentage of matching judgments. Our approach achieved an accuracy of 68.8% on the two way task and 61.6% on the three way task which ranked the approach as 2^{nd} when compared to the other participating runs in the same challenge.

The alignment of meaning representation relies heavily on the semantic relatedness between concepts. Consequently, the accuracy of the alignment is directly related to the precision of the semantic relatedness measurement. To improve that accuracy, we have then investigated a new approach to measure semantic relatedness. This approach (described in Chapter 5) is based on the assumption that the type of semantic relations in a lexicon can be a good indicator of semantic relatedness. We evaluated our lexicon semantic relatedness approach intrinsically using correlation with human ranking of semantic relatedness, and synonymy tests. The results show a Pearson's correlation of 0.93 with human ranking of semantic relatedness and an accuracy of 91.25% on the TOEFL synonym set. This result significantly improves the state of the art of lexicon-based approaches (described in details in Section 5.2). We have also applied our semantic relatedness measure extrinsically to RTE, and achieved an accuracy of 56% on the three way task of the fourth recognizing textual entailment challenge (shown in Section 5.3). Although this approach resulted in a lower accuracy than using the web, we foresee that combining different sources of measurements could lead to much better accuracy (discussed in Section 6.2).

1.4 Intended Contributions

In this thesis, we show that the type of semantic relatedness axioms used to align meaning representations can be successfully used as indicators of textual entailment. The proposed development in this thesis contributes to research in Natural Language Processing in the following ways:

Development of a Recognizing Textual Entailment Approach

Our main contribution is the development of a novel approach to recognizing textual entailment based on description logic and semantic relatedness, which improves the current state of the art. We have developed a method for representing text in description logic, that was published in [Siblini and Kosseim, 2008a] and is described in Chapter 3.

We have also used the types of logical statements used to align textual representations as features in a machine learning algorithm to recognize textual entailment. This was published in [Siblini and Kosseim, 2009] and is described in Chapter 4. To show how our approach can be used in RTE, we have built a prototype (called AORTE) and evaluated it on the available benchmarks.

This approach has led to other contributions that are described below.

Development of a Method to Natural Language Querying

We have designed a novel approach to query a knowledge base in natural language. This approach (described in Section 3.3) is based on predicate selectional preferences to answer queries in natural language, and was published in [Kosseim, Siblini, Baker, and Bergler, 2006]. To test the proposed approach and its usefulness, we have developed a natural language querying prototype (called ONLI) and evaluated it on the Fungal Web Ontology.

Development of a Web Based Named Entities Recognition Approach

We have developed an approach to extract classes of named entities by exploring the web linguistically as a corpus. This approach (described in Section 4.2) is based on lexical preferences of grammatical relations in addition to a set of grammatical patterns. This has led to the publication [Siblini and Kosseim, 2008b].

Development of a Semantic Relatedness Approach

Finally, we have developed a lexicon based method to measure semantic relatedness. This approach is based on the types of semantic relations between concepts as an indicator of relatedness, and has improved the current state of the art of lexicon based semantic relatedness measures. This was published in [Siblini and Kosseim, 2013b] and is described in details in Section 5.1. An extension of this method, which was devised to detect phrasal similarity, has been published in [Siblini and Kosseim, 2013a] and is described in details in Section 5.2.3.

1.5 Thesis Organization

The thesis is organized as follows: Chapter 2 introduces the state of the art in recognizing textual entailment. Section 2.1 reviews the different approaches for recognizing textual entailment categorized by the method of representation. Section 2.2 presents the available benchmarking data in the field, followed by an overview of the resources used and their impact on recognizing textual entailment.

Chapter 3 introduces our baseline approach for recognizing textual entailment that is based on description logic and knowledge querying. The goal of this chapter is to introduce a baseline method, and show how far we can go using a shallow logical based method and without a huge set of axioms and inference rules. This chapter starts with an overview of the approach, and describes our method for automatically representing text in description logic. Section 3.3 of this chapter explains our approach for querying a knowledge representation in natural language (ONLI) and its evaluation on a large knowledge base. Section 3.4 demonstrates how knowledge representation querying can be used for recognizing textual entailment, followed by its evaluation and result analysis.

Based on what we have learned from Chapter 3, we describe our second approach and main contribution to recognizing textual entailment in Chapter 4. The chapter introduces our method of using the Web as a corpus for extracting semantic type of named entities (Section 4.2). The chapter also shows our approach for aligning knowledge representations (Section 4.3), and demonstrates how the type of semantic relations used in aligning the representations can be a good indicator of textual entailment (Section 4.4). This chapter ends with the evaluation of this approach on the available benchmarks (Section 4.5) and an analysis of the results.

In Chapter 5, we revisit the RTE approach described in Chapter 4 and more specifically the alignment of knowledge representation. We try to investigate a novel approach for measuring lexical semantic relatedness based on a weighted semantic network (Section 5.1) and evaluate it with the available benchmarks (Section 5.2). Section 5.3 presents the evaluation of the RTE approach with the novel semantic relatedness method, followed by an analysis of the results (Section 5.4).

The last chapter provides a discussion and analysis of the dissertation, and concludes with our major contributions, and possible further work.

Chapter 2

State of the Art

In recent years, there has been much interest in recognizing textual entailment. This can be observed in particular in the popularity of the Recognizing Textual Entailment (RTE) challenges. In the following chapter we will examine the different approaches for recognizing textual entailment as described in the literature, we will look at the benchmarks available for evaluating the different methods, and review the resources used by those approaches.

2.1 Approaches to Recognizing Textual Entailment

Recognizing textual entailment is the task concerned with deciding whether a text entails a hypothesis or not. A typical approach for recognizing textual entailment is usually made up of three main components: a representation component, a comparison component, and a decision component. The representation component involves the representation of the text and the hypothesis in a way to facilitate the comparison between the two. A text can be represented as a tree ([Iftene, 2008, de Marneffe, MacCartney, Grenager, Cer, Rafferty, and Manning, 2006, Marsi, Krahmer, Bosma, and Theune, 2006, Bar-Haim, Berant, and Dagan, 2009]), a set of predicate argument structures ([Sammons, Vydiswaran, Vieira, Johri, Chang, Goldwasser, Srikumar, Kundu, Tu, Small, et al., 2009, Krestel, Witte, and Bergler, 2009]), a logical form ([Tatu, Iles, Slavick, Novischi, and Moldovan, 2006, Clark and Harrison, 2008, de Salvo Braz, Girju, Punyakanok, Roth, and Sammons, 2006]), or with other representations (e.g. [Wang and Neumann, 2008, Bos and Markert, 2006]). The comparison methods will then vary depending on the selected representation. The decision component decides if the hypothesis is entailed from the text or not based on the comparison results. This component is usually either a set of predefined thresholds or rules over the comparison method or more commonly based on a machine learning technique.

In the following subsections we categorize recognizing textual entailment approaches in terms of their levels of representation. In addition, we mention the evaluation of those approaches on the Recognizing Textual Entailment task, which is the main benchmark task for recognizing textual entailment (described in Section 2.2).

2.1.1 Lexical Based Methods

Numerous approaches to recognizing textual entailment rely directly on the text and the hypothesis surface strings, without creating any kind of further representation. Such approaches operate solely on a string comparison between the text and the hypothesis. The comparison component of such methods can be a simple counting of word overlap, the computation of the Levenshtein edit distance [Levenshtein, 1966] such as the work of [Adams, 2006] and [Castillo and Alemany, 2008], or other lexical similarity measures as in the approaches of [Settembre, 2007, Perini, 2009, Pakray, Bandyopadhyay, and Gelbukh, 2009]. The decision component in this case is either a simple set of rules on the resulted computation, or machine learning algorithm trained on similar data.

An approach worth describing that uses a lexical method is [Adams, 2006]'s approach. The comparison component of this method operates on a combination of word similarity measures, a web based word similarity method, and the lexical edit distance for comparing T and a H. The word similarity method used is the [Hirst and St-Onge, 1998] method that uses a lexical database relations as a similarity measurement. The web based method is based on [Glickman, Dagan, and Koppel, 2006] that uses web frequencies to count similarities. The lexical edit distance simply counts the number of words that were not identical from H to T relative to the length of H, which is seen as insertion from an editing perspective. Then the computed comparison measurements are used as an input to a J48 decision tree classifier that was trained on the development set. The classifier decides whether H is entailed on H or not. The approach achieved a relatively high accuracy of 0.63 on the RTE2 challenge (described in Section 2.2).

[Settembre, 2007]'s comparison approach uses a lexical similarity metric ratio, and synonym and antonym replacement. The first metric counts word overlap normalized by the total number of words in the hypothesis. The second metric seeks for synonyms and antonyms when a word in the hypothesis is not matched. For the decision making a probabilistic model is used which essentially calculates the probability that a feature will appear in a dataset. This approach achieved an accuracy of 0.62 on the RTE3 challenge. [Perini, 2009]'s comparison approach is based on the word relatedness score of [Corley and Mihalcea, 2005] between the text and the hypothesis, and a gene expression decision component to decide whether the T-H pair is an entailment or not. The approach achieved an accuracy of 0.615 on the RTE5 challenge.

[Breck, 2009]'s comparison approach is based on the observation that for a text to entail a hypothesis, the text must mention all the information in the hypothesis. The comparison is done through string matching, string edit distance that considers that two words match if they have 80% of the letters in one or more adjacent text words, and finally lexicon based similarity. The approach achieved an accuracy of 0.61 on the RTE5 challenge[1]. [Bayer, Burger, Ferro, Henderson, and Yeh, 2005]'s MITRE system approach is based on string alignment using a statistical machine translation model trained on news corpus headlines. The module is designed to find correspondence between pairs of sentences for machine translation purposes, and was used in comparing a text and hypothesis. For decision making, a k-nearest neighborhood was used to classify T-H pairs. The approach achieved an accuracy of 0.58 on the RTE1 challenge.

[Pakray, Bandyopadhyay, and Gelbukh, 2009]'s approach used a combination of unigram matching, bigram matching, and longest common sub-sequence, and named entity matching for comparing T and H. Then, they used simple rules of ranking the most matched pair as entailment in a set. The approach achieved an accuracy of 0.58 on the RTE5 challenge.

[1] It should be noted that results across challenges cannot be compared as the tasks and difficulty may vary across years.

[Castillo and Alemany, 2008]'s approach is based on four metrics: edit distance, WordNet similarity measurements, and longest common substring for comparing T and H. Then it uses a support vector machine to make a decision about the T-H pairs. The approach achieved an accuracy of 0.57 on the RTE4 challenge.

[Perez and Alfonseca, 2005]'s comparison approach is based on the BiLingual Evaluation Understudy (BLEU) algorithm [Papineni, Roukos, Ward, and Zhu, 2002] that is usually used to automatically evaluate machine translation. The BLEU algorithm calculates the percentage of overlapping n-grams between T and H, and using different n values (unigram, bigram). The approach achieved an accuracy of 0.49 on the RTE1 challenge, which is less than the baseline (see Section 2.2).

In general, lexical based methods perform poorly on the task of recognizing textual entailment. The main reason of this poor performance is that textual entailment is a directional relation, where the text contains more information than the hypothesis. In addition supervised machine learning based methods of alignment require large training corpora, which is usually not available. Another reason for the poor performance is that entailment knowledge does not always appear at the surface level, and additional knowledge is needed to infer textual entailment. For example, the text *If you help the needy, God will reward you* entails the hypothesis *Giving money to a poor man has good consequences*. A pure lexical based match for this example, may not yield a positive result, without the knowledge that *giving money is helping*, *the poor are the needy*, and that *a reward is a good consequence*. Our approach for recognizing textual entailment uses a deeper approach based on description logic representations and logical reasoning for comparison, and machine learning for decision making.

2.1.2 Syntax Based Methods

The most popular types of approaches for recognizing textual entailment are syntax based, and specifically the use of tree based representation. Syntactic information is usually represented as a tree or a graph and then the comparison becomes a tree searching, tree alignment, or other graphical based method of comparison.

[Iftene, 2008]'s approach starts by parsing the text and the hypothesis into dependency trees using the Minipar parser [Lin, 2003]. Then the comparison tries to map every node in the hypothesis tree to one in the text tree. The mapping can be done either directly when the entities are available between the two, or indirectly by using rules extracted from external resources such as the 12 million rules extracted from DIRT (described in Section 2.3.9). A score is kept for the type and availability of a mapping between the two and is used to decide whether the text entails the hypothesis or not. This approach has achieved an accuracy of 0.72 on the RTE4 challenge.

Another method that used parse trees for representation but a graphical similarity method for comparison is [Zanzotto, Moschitti, Pennacchiotti, and Pazienza, 2006]'s method. To measure the similarity between the two trees, they capture the number of common sub-trees that share the same anchoring scheme. Anchors are links between words that are equal or similar based on a similarity function. A support vector machine is trained using positive examples from the development set that decided whether the text entails the hypothesis or not. This approach has achieved an accuracy of 0.63 on the RTE2.

[de Marneffe, MacCartney, Grenager, Cer, Rafferty, and Manning, 2006]'s approach represents the text and the hypothesis as a typed dependency graphs. These graphs contain a node for each word, and labeled edges representing grammatical relations between words. To compare the graphs, an alignment heuristic is used and searches the graph for the most similar sub-graphs. The semantic comparison uses external resources including WordNet [Fellbaum, 1998] and special purpose gazetteers. To make a decision about entailment, they use a logistic regression classifier that is trained on a development set to make the decision. This approach has achieved an accuracy of 0.60 on the RTE2.

[Marsi, Krahmer, Bosma, and Theune, 2006]'s approach represents the text and the hypothesis as dependency trees, and compared the two by first aligning them using an algorithm that matches nodes in a dependency trees. The similarity score of each pair of nodes depends on their own similarity and the similarity of the best matching pairs of their descendants. To decide on the entailment, the authors look at whether the top node of the hypothesis dependency tree is aligned, and whether the alignment strength exceeds a certain threshold value. This approach has achieved an accuracy of 0.60 on the RTE2.

[Bar-Haim, Berant, and Dagan, 2009]'s approach represents the text and the hypothesis as compact forests that are basically a set of dependency trees. The matching part is based on rules and tree kernels computed over compact forests. The entailment decision is made by a support vector machine generated from the training set over feature extracted from the matching phase. This approach has achieved an accuracy of 0.60 on the RTE4.

The advantage of comparing representations at a syntactic level is that it may reveal similarities that might not be evident at the linear surface level. The created trees or graphs can relate distant words, and can make the discovery of similar meaning easier even if the sentences have different word order. However, those advantages can be offset by the inaccuracies of the parsers.

2.1.3 Semantic Based Methods

Semantic representations are able to reveal similarities which cannot be detected by a surface or syntactic level. For example, a semantic representation can offset grammatical variabilities of the same textual meaning as in the case of one semantic representation that denotes both the active and passive voice of the same text. Semantic representation may also capture implicit information such as information about the roles an argument plays in relation to its predicate. On the other hand, the semantic analysis process might introduce inaccuracies in the representation, which might offset the advantage of revealing semantic similarities that are hard to be revealed with a syntactic based representation or a surface level one. Semantic based methods, those that use some sort of semantic representation of the text and the hypothesis, are usually based on a predicate argument structure.

[Hickl, Williams, Bensley, Roberts, Rink, and Shi, 2006]'s Groundhog system creates a semantic representation of the text and the hypothesis by following several steps. The system performs a lexical analysis, syntactic parsing, coreference resolution, and semantic parsing. Then semantic role labeling are added using a system trained on the predicate-argument annotations found in PropBank (described in Section 2.3.6). The comparison phase first performs a lexical alignment of the semantic representation, and then the generation of possible paraphrases using the web as a corpus. The decision phase is made by the extraction of dependency features, paraphrase features, and semantic features from the representation and feed it to a decision tree based machine learning classifier (an implementation of C5.0), which classifies the representations as either entailment or not. This approach has achieved a high accuracy of 0.75 on the RTE2 challenge.

[Sammons, Vydiswaran, Vieira, Johri, Chang, Goldwasser, Srikumar, Kundu, Tu, Small, et al., 2009] also create predicate argument representations of the text and the hypothesis. They integrate multiple resources to create the representation with multiple levels of annotations. For the comparison phase, a set of entailment metrics have been devised that score the similarity of two semantic constituents. Those metrics differ from the type of annotation they are comparing. For the decision phase, the entailment metrics are used as features to a support vector machine that decides whether the text entails the hypothesis or not. This approach has achieved an accuracy of 0.64 on the RTE5 challenge.

[Krestel, Witte, and Bergler, 2009]'s approach represents the text and the hypothesis as a predicate argument structure. For comparison, a set of rules are used to compare the created representations. The decision of entailment is made if the resulted similarity score is higher than a threshold. This approach has achieved an accuracy of 0.54 on the RTE4 challenge.

2.1.4 Logical Form Based Methods

A logical meaning representation is able to expose similarities that are not seen in lower representation levels. Such a representation is a meaning representation that is backed with a sound and understandable formal semantics and can take advantage of formal reasoning algorithms to derive information. The disadvantage of a logical based approach is the required knowledge resources in the creation of such a representation, and more specifically the need for large number of axioms and inference rules that are not generally available. Most of the approaches rely on a manually created set of rules to do so. In addition to the resources needed, the computational processing power required is much more important than the lexical, syntactic, or semantic based approaches. The higher the representation level, the additional processing and knowledge needed to create them and the higher the possibilities of inaccuracies. In addition, the efficiency of reasoning over a representation is directly related to the expressiveness of the representation, as such, a balance should be struck between the representation power and the reasoning capabilities. Logical form based methods are one of the most knowledge intensive set of approaches and that relies on a logical meaning representation. The comparison phase of such approaches relies on logical entailment usually using a theorem prover and the decision is then based on the prover's results. The main difficulty of this type of approach is the creation of the meaning representation. Below are some of the approaches that have used a logical representation in recognizing textual entailment.

[Tatu, Iles, Slavick, Novischi, and Moldovan, 2006]'s approach shows a high accuracy using a logical representation and logical proving system. The logic representation is derived from a full syntactic parse, semantic parse, and a temporal representation. They then use the COGEX [Moldovan, Clark, Harabagiu, and Maiorano, 2003] natural language prover originating from OTTER [McCune, 1994] to proove the hypothesis from the represented text. A large number of axioms have been created from various external knowledge bases and used by the prover. The entailment decision is then based on the proof's computed score, which is a measurement of the kinds of axioms used in the proof and the significance of the dropped arguments and predicates. This approach has achieved a high accuracy of 0.73 on the RTE2 challenge.

[de Salvo Braz, Girju, Punyakanok, Roth, and Sammons, 2006] presents an interesting approach that involves the induction of the representation of T and H into a hierarchical knowledge representation. The representation used is the Extended Feature Description Logic (EFDL) language. The representation is induced by machine learning based resources, including a tockenizer, lemmatizer, part of speech tagger, syntactic parser, semantic parser, named entity recognizer, and a name coreference system. In additional, a set of rewrite paraphrasing rules were used to create 300 inference rules. An inference procedure is recursively applied to match the nodes in the representation. The matching information is then used to reformulate the recognizing of textual entailment problem in an equivalent Integer Linear Programming (ILP) problem. This approach has achieved an accuracy of 0.56 on the RTE4 challenge.

[Clark and Harrison, 2008]'s BLUE (Boeing Language Understanding Engine) system creates a logic based representation of the text and the hypothesis. The system uses a syntactic parser and a logical form generator to generate a semi-formal structure between a parse and full logic. The structure is a normalized tree structure with logic type elements generated by grammar rules. The semi-formal structure is then used to generate ground logical assertions by applying a set of syntactic rewrite rules recursively to the structure. The entailment task is then reduced to inferring if the hypothesis subsumes the text. The system tries to do the inference on the created representation based on inference rules that are generated from the word's logical definitions and a paraphrasing database. This approach has achieved an accuracy of 0.56 on the RTE4 challenge.

2.1.5 Hybrid Methods

Hybrid methods cover approaches that use a combination of methods to recognize textual entailment. Hybrid approaches are usually based on only two methods with one acting as a primary strategy and the other as a backup. However, some are based on multiple methods with a voting mechanism to select the best result. Below is a description of some of the methods that follow a hybrid approach.

[Wang and Neumann, 2008] creates multiple modules that each work on a specific RTE problem, and then combine the results using a voting mechanism. The modules created include: a time anchoring module that detects entailment relationships between temporal expressions, a named entity oriented module that detects entailment relationships between named entities, and a tree skeleton module that uses a kernel based machine learning method to make the entailment prediction on dependency trees. Different confidence values assigned to each module are used by the voting mechanism to decide on the result. If all modules fail to return a result, a backup module that is based on a simple bag of word approach is then used. This approach achieved a high accuracy of 0.70 on the RTE4 challenge.

[Bos and Markert, 2006]'s approach combines two modules, one based on a bag of words and the other based on logical representation and inference. For the first method word overlap and word weight that are calculated as the inverse document frequency from the web are used to compute relatedness. On the other hand, the second module uses a first order fragment of the DRS language used in Discourse Representation [Kamp and Reyle, 1993] and the Vampire 7 [Riazanov and Voronkov, 2002] theorem prover. A decision tree model is then created to decide which result of the two different modules to use. This approach has achieved a precision of 0.61 on the RTE2 challenge.

This category of approach has the capabilities of aggregating several methods and choosing the best of their results. Consequently, it should have an advantage over individual methods, yet the decision of which method result to choose from is not an easy problem to solve and require a great deal of training to be able to do so.

2.1.6 Analysis

As seen in the previous section, the most accurate approaches used in RTE seem to be based on a either on a semantic representation (Section 2.1.5), or a logical form

(Section 2.1.6). [Hickl, Williams, Bensley, Roberts, Rink, and Shi, 2006]'s approach, based on semantics, attained the highest accuracy of 0.75 at the RTE2 challenge. At the same challenge, the next highest accuracy belongs to [Tatu, Iles, Slavick, Novischi, and Moldovan, 2006]'s approach, from the logical form based category, with an accuracy of 0.73. Considering that the logical form of a sentence is a formal representation which also offers proper formal semantics and reasoning, we believe that they provide a significant advantage over semantic forms which lack the ability of having proper inferences. Therefore, we have chosen to concentrate our efforts on a logical form based method. In spite of all the advantages of logical form based methods, they suffer from one common shortcoming: the demand for a large number of axioms and word knowledge. For example, [de Salvo Braz, Girju, Punyakanok, Roth, and Sammons, 2006] logical based approach relies on 12 million paraphrase rules in the DIRT database (described in Section 2.3.9) that are transformed into logical axioms, in addition to 300 manually generated inference rules. [Clark and Harrison, 2008]'s approach relies on 100,000 inference rules created from WordNet glosses (described in Section 2.3.9) in addition to the 12 million paraphrase extracted from DIRT and all transformed into logical form. Similarly, [Tatu, Iles, Slavick, Novischi, and Moldovan, 2006]'s approach relies on an undisclosed large number of axioms divided into the following categories: lexical chains axioms, ontological axioms, linguistic axioms, semantic calculus axioms, and temporal axioms. Most of these axioms were either extracted from various external hand crafted knowledge bases or have been manually created by the authors. This lead us to our research question: how far can we go with a surface logical representation and semantic relatedness to recognize textual entailment? We want to examine the possibility of using a logical form based methods but without the prerequisite large knowledge base. Instead, we believe that the semantic relatedness of concepts would be enough for the specific task of recognizing textual entailment.

Consequently the different approaches that we will present in this dissertation all rely on a decidable formal logical representation and semantic relatedness to recognize textual entailment.

2.2 Recognizing Textual Entailment Benchmarks

The main benchmark task for recognizing textual entailment is the Recognizing Textual Entailment task dedicated for this problem. Since 2005, there have been seven challenges for recognizing textual entailment. Those challenges have made it easier for researchers to compare their work and learn as a community. These challenges provide common test collections, a common evaluation procedure, and a medium to share and discuss researchers' work.

The RTE challenges were originally organized by PASCAL[2], the Pattern Analysis, Statistical Modelling and Computational Learning Network of Excellence funded by the European Union. Afterwards, they became part of the National Institute of Standards and Technology[3] (NIST)'s Text Analysis Conference[4] (TAC). In the following sections, we will give an overview of each of the challenges, a description of the methods used in recognizing textual entailment, and the various resources used by researchers in the field.

2.2.1 The First Recognizing Textual Entailment Challenge (RTE1)

The first recognizing textual entailment challenge[5] was held in 2005 as an attempt to promote an abstract generic task that captures major semantic inference needs across

[2] http://www.pascal-network.org/
[3] http://www.nist.gov
[4] http://www.nist.gov/tac/
[5] http://pascallin.ecs.soton.ac.uk/Challenges/RTE

applications [Dagan, Glickman, and Magnini, 2005]. This first attempt was prompted after suggestions by the scientific community to have a separate empirically evaluated task for textual entailment recognition ([Monz and de Rijke, 2001, Condoravdi, Crouch, De Paiva, Stolle, and Bobrow, 2003, Dagan and Glickman, 2004]). This first challenge dataset consisted of small text snippets from the general news domain of Text-Hypothesis pairs. Each pair was labeled by human annotators as either the "text (T) entails the hypothesis (H)" or not. The dataset was balanced in terms of entailment and not entailment. The dataset consisted of 567 text-hypothesis pairs as the development set and 800 as the test set. All pairs are categorised into one of the following seven subsets corresponding to settings in different applications:

Information Retrieval Subset (IR):

The annotators created web queries as hypothesis based on prominent sentences in news stories, and selected candidate texts from search engines retrieved documents that either entail or not the created query. The following is a text-hypothesis pair example from the RTE1 development data set:

```
<pair task="IR" value="TRUE" id="20">
  <t>Eating lots of foods that are a good source of fiber may keep your blood
    glucose from rising too fast after you eat.</t>
  <h>Fiber improves blood sugar control.</h>
</pair>
```

The data set is represented in the eXtensible Markup Language (XML). The main tag <pair>has a "task" attribute that shows the subset that the pair belongs to. In this example, "IR" indicates the Information Retrieval task. The <pair> tag also includes the "value" attribute which is either true or false, for entailment or not. In this example, the vale true indicates that the hypothesis is entailed from the text. The <t> tag contains the text string, and the <h> tag contains the hypothesis string.

Comparable Document Subset (CD):

The annotators examined news articles on common stories with common lexical overlap to extract text-hypothesis pairs. For example:

<pair task="CD" value="TRUE" id="778">
<t>Voting for a new European Parliament has been clouded by apathy.</t>
<h>Apathy clouds EU voting.</h>
</pair>

Reading Comprehension Subset (RC):

For the RC data set, the annotators created text-hypothesis pairs as if they were creating a reading comprehension test for high school students. For example:

<pair task="RC" value="TRUE" id="153">
<t>The Mona Lisa, painted by Leonardo da Vinci from 1503-1506, hangs in Paris' Louvre Museum.</t>
<h>The Mona Lisa is in France.</h>
</pair>

Question Answering Subset (QA):

The annotators used the Cross Language Evaluation Forum (CLEF)[6] QA clusters of questions as a resource for questions. The question is run on the TextMap Web Based Question Answering system [Echihabi, Hermjakob, Hovy, Marcu, Melz, and Ravichandran, 2003], which provides relevant text snippets as potential answers to the question. Those potential answers along with the question itself are used to create the text and the hypothesis. For example, the CLEF-QA question *"Who painted the Scream?"*, returned the text snippet *"Norway's most famous painting, 'The Scream' by Edvard Munch"*, which is used as the text. Then the question is transformed to an affirmative form as the hypothesis. The resulting pair is:

[6] `http://clef.isti.cnr.it/`

<pair task="QA" value="TRUE" id="568">
<t>Norway's most famous painting, "The Scream" by Edvard Munch, was recovered Saturday, almost three months after it was stolen from an Oslo museum.</t>
<h>Edvard Munch painted "The Scream".</h>
</pair>

Information Extraction Subset (IE):

The annotators used the UIUC[7] information extraction relations as hypotheses, and the potential sentences from new stories answering the IE relation as the texts. For example, given the information extraction task of identifying the acquirer of an acquisition relation we get the following example pair:

<pair task="IE" value="TRUE" id="955">
<t>C&D Technologies announced that it has closed the acquisition of Datel, Inc.</t>
<h>C&D Technologies acquired Datel Inc.</h>
</pair>

Machine Translation Subset (MT):

The annotators used the Document Understanding Conferences (DUC)[8] 2004 evaluation data, from the National Institute of Standards and Technology (NIST)[9], automatic and human translations as either the text or hypothesis. The correctness of the translation determines whether the text entails or not the hypothesis. For example:

<pair task="MT" value="FALSE" id="363">
<t>Baghdad had announced that it will stop cooperating with UNSCOM completely but indicated that it will not ask for their departure.</t>

[7] http://cogcomp.cs.illinois.edu/Data/ER/
[8] http://www-nlpir.nist.gov/projects/duc/
[9] http://duc.nist.gov/duc2004/

<h>Baghdad announced the complete halt in their cooperation with UNSCOM, and said also, that it will ask them to leave.</h>
</pair>

Paraphrase Acquisition Subset (PP):

The annotators used multiple paraphrase acquisition systems, such as the Corpus of Sentence Alignment in monolingual comparable corpora[10] to acquire text-hypothesis pairs. For example:

<pair task="PP" value="TRUE" id="521">
<t>California voters recall Gray Davis and elect Arnold Schwarzenegger as their governor.</t>
<h>California voters dumped Gov. Gray Davis and replaced him with Arnold Schwarzenegger.</h>
</pair>

Each text-hypothesis pair was annotated by at least two annotators with an average of 80% between each pair of annotators, and an average Kappa[11] level of 0.6. Sixteen teams submitted their systems' results to the challenge. The systems' results were compared to the gold standard, and the percentage of matching judgments was used as the accuracy of the system. Accuracy in this case is simply the total number of correctly classified pairs over the total number of all pairs.

The overall accuracies of all systems were between 50 and 60 percent, a relatively low accuracy (considering that the baseline is about 50%), which goes to show the challenges of such task.

[10] `http://www.cs.columbia.edu/~noemie/alignment/`

[11] Kappa coefficient is a statistical measure of agreement between annotators [Carletta, 1996]

2.2.2 The Second Recognizing Textual Entailment Challenge (RTE2)

In 2006, the second RTE challenge[12] was focused on providing a more realistic and larger data set. The data set consisted of 800 text-hypothesis pairs for development and 800 for testing. However, the categories of the examples for this challenge were only four: Information Retrieval (IR), Information Extraction (IE), Question Answering (QA), and multi-document summarization (SUM) [Bar Haim, Dagan, Dolan, Ferro, Giampiccolo, Magnini, and Szpektor, 2006].

The information retrieval, information extraction, and question answering data were comparable to those in RTE1, and the multi-document summarization subset is what used to be the comparable document subset (CD).

These are some examples from the RTE2 dataset:

<pair task="IR" entailment="YES" id="5">
<t>Scientists have discovered that drinking tea protects against heart disease by improving the function of the artery walls.</t>
<h>Tea protects from some diseases.</h>
</pair>
<pair task="IE" entailment="NO" id="8">
<t>Mangla was summoned after Madhumita's sister Nidhi Shukla, who was the first witness in the case.</t>
<h>Shukla is related to Mangla.</h>
</pair>
<pair task="QA" entailment="YES" id="4">
<t>A Chilean expert points out that the 1987 Montreal Protocol has not been effective in arresting the destruction of the ozone layer in the earth's atmosphere, resulting in the unprecedented size of the ozone hole this year, and causing 120,000 people in Chile to be on yellow alert for ultraviolet radiation (meaning a fair-skinned person would get burnt within 10 minutes of being exposed to the sun).</t>
<h>The ozone layer is in the earth's atmosphere.</h>
</pair>
<pair task="SUM" entailment="YES" id="1">

[12] `http://pascallin.ecs.soton.ac.uk/Challenges/RTE2`

<t>The news comes as doctors in Hong Kong warned that people who survive Sars may suffer permanent lung damage and may suffer a relapse.</t>
<h>Those who recovered from Sars might have permanent lung damage.</h>
</pair>

Another difference in this challenge to the previous one is the additional filtering on the data set for pairs with annotators' disagreement. RTE2 only provided data with an average agreement of 89.2% between each pair of annotators, and an average Kappa level of 0.78 on the test set.

In addition, a secondary task was added to rank H-T pairs according to the system's confidence. This was evaluated using the average precision measure [Voorhees, 2001]. Twenty three teams submitted their system's results to the challenge. The overall accuracies of all systems were between 53% and 75%.

2.2.3 The Third Recognizing Textual Entailment Challenge (RTE3)

The third RTE challenge[13] in 2007 basically followed the same structure as the previous one, with the main difference of the introduction of longer texts, up to a paragraph long for some of the texts, compared to a one sentence from the previous years. The data set consisted of 800 text-hypothesis pairs for development, and 800 for testing. 17% of the tests had a longer text and were marked by a "long" length attribute. For example:

<length="long" task="IE" entailment="YES" id="82"> <t>Jerry Reinsdorf (born February 25 1936 in Brooklyn, New York) is the owner of Chicago White Sox and the Chicago Bulls. Recently, he helped the White Sox win the 2005 World Series and, in the process, collected his seventh championship ring overall (the first six were all with the Bulls in the 1990s), becoming the third owner in the history of North American sports to win a championship in two different sports.</t>
<h>Jerry Reinsdorf has won 7 championships.</h>

[13] `http://pascallin.ecs.soton.ac.uk/Challenges/RTE3`

</pair>

Twenty six teams submitted their system's results to the challenge [Giampiccolo, Magnini, Dagan, and Dolan, 2007]. The overall accuracies of all systems were between 49% and 80%. In addition to the main task, an optional pilot task was also introduced. This pilot task, called "Extending the Evaluation of Inferences from Texts", was set to differentiate between unknown entailment and contradiction. This task allows for each textual entailment three possible answers: "YES" (entails), "NO" (contradicts), and "UNKNOWN", with the goal to drive for more precise informational distinctions. The pilot data set consisted of 1600 text-hypothesis. Eight teams participated in the pilot task and the overall accuracies of all systems were between 35% and 73%. Our baseline knowledge querying system was tested on the RTE3 challenge and performed below the average with a result of 49% on the two-way task (entailment, no entailment).

2.2.4 The Fourth Recognizing Textual Entailment Challenge (RTE4)

The fourth RTE challenge[14] became part of the Text Analysis conference (TAC) under the auspice of the National Institute of Standards and Technology (NIST). In terms of dataset, RTE4 was similar to the previous one with one major difference: the addition of a three way classification of entailment (which was a suggested pilot task in RTE3). The three way classification added an "unknown" value for entailment, where the truth of the hypothesis cannot be determined by the text. So the entailment attribute value contains "Entailment", "Contradiction", and "Unknown". Another difference is related to the text length, where most texts were as long as a paragraph. For example:

<task="IR" entailment="UNKNOWN" id="56">

<t>In a bad-tempered outburst last week, Steven Crawshaw, chief executive of Bradford & Bingley, accused analysts of looking for "communists under every bed"

[14] http://www.nist.gov/tac/publications/2008/papers.html

when questioned about the bank's £300m rights issue. The normally breezy boss added that, after years of criticism for not being enough like its fast-growing rival from the North East, Northern Rock, now it was being blamed "for the sins of being too much like Northern Rock".</t>
<h>Bradford & Bingley falls into the red.</h>
</pair>

The testing data set of this challenge was also larger than the previous year, having 1000 pairs compared to 800 from the previous years, with more pairs in the QA and SUM categories than in the IE and IR categories as the former proved to be more difficult. The distribution of the 3 way task on the test set was 50% entailment, 35% unknown, and 15% contradiction. The three way systems submission values were also automatically converted to two way values, where contradiction and unknown were conflated as no entailment, hence the systems that submitted for the three way task were automatically also submitting to the two way task.

Twenty six teams submitted their system's results to the challenge [Giampiccolo, Dang, Magnini, Dagan, and Dolan, 2008]. The overall accuracies of all systems that participated in the two-way task were between 49.7% and 74.6% with an average accuracy of 58%. As for the three way task, the accuracy was between 30.7% and 68.5% with an average accuracy much lower than the two way task of 51%. Our knowledge alignment system was tested on the RTE4 three-way task and resulted in an accuracy of 61.6%, which was ranked 2^{nd} when compared to other system that participated in the same task. The RTE challenge automatically converts the three way submitted runs into two way runs by automatically conflating *"Contradiction"* and *"Unknown"* to *"No Entailment"*. Our results on the two way run scored 68.8%, which ranked 3^{nd} when compared to all the system that participated in the 2-way challenge and the 3-way challenge conflating to two way results.

2.2.5 The Fifth Recognizing Textual Entailment Challenge (RTE5)

In 2009, the fifth RTE challenge[15] was similar to the previous one with the addition of a search task. The search task consisted of searching for all sentences in a corpus that entail a given hypothesis. In addition, ablation testing of all knowledge resources used by participating systems was a mandatory requirement for all participants. The ablation test was aimed at examining the importance of each resources used in recognizing textual entailment.

As for the testing dataset, the text became even longer (up to 100 words) and consisted of 1200 text-hypothesis pairs, compared to 1000 from the previous year.

Twenty teams submitted their system's results to the challenge's main task [Bentivogli, Dagan, Dang, Giampiccolo, and Magnini, 2009]. The overall accuracies of the systems were between 50% and 73% on the two way task, and between 43% and 68% on the three way task.

After RTE5, both the sixth RTE challenge[16] [Bentivogli, Clark, Dagan, Dang, and Giampiccolo, 2010] and the seventh RTE challenge[17] [Bentivogli, Clark, Dagan, Dang, and Giampiccolo, 2011] focused mainly on the search task. The challenge changed at this stage from recognizing textual entailment to the search and extraction of textual entailment task situated mainly in the summarization application setting. This completely changed the task from a classification one, to a retrieval one. With a retrieval task, the context of the topic is important for the text to be interpreted as the text may rely on implicit references to other information in the corpus. This made the task a lot more challenging, as evidenced from the drop in accuracy in the average result of the RTE6-7 participants.

[15] `http://www.nist.gov/tac/publications/2009/papers.html`
[16] `http://www.nist.gov/tac/publications/2010/papers.html`
[17] `http://www.nist.gov/tac/publications/2011/papers.html`

2.2.6 The Student Response Analysis and The Eighth Recognizing Textual Entailment Challenge (SRA-RTE8)

In 2013, the Join Student Response Analysis and Eight Recognizing Textual Entailment Challenge[18] was part of the International Workshop on Semantic Evaluation (SemEval 2013) [Dzikovska, Nielsen, Brew, Leacock, Giampiccolo, Bentivogli, Clark, Dagan, and Dang, 2013]. This challenge's main task was to assess student answers to exercise questions that can be useful in a tutorial or e-learning setting. The accuracy of student's answers is assessed as a textual entailment to a known correct reference answers. The following is an example entry from the training dataset:

Question: Why does measuring voltage help you locate a burned out bulb?

Reference Answer: Measuring voltage indicates the place where the electrical state changes due to a gap.

Student Answer: because if there is a difference in electrical states then there is a gap. That will located the burned out bulb.

The assessment is performed on different levels of granularity: as a 2-way task (correct, incorrect), a 3-way task (correct, contradictory, or incorrect), or as a 5-way task, where the approaches are required to classify the student answer according to one of the following:

Correct: if we have a bi-directional textual entailment, where the reference answer entails the student answer and the student answer entails the reference answer (as shown in the example above).

Partially correct or incomplete: if the reference answer entails the student answer, but the student answer does not entail the reference. This means that the student answer contains some but not all the information from the reference answer. An example of a partially correct student answer for the example question above is: "electrical states will determine whether there is a gap or connection in the circuit."

Contradictory: if the student answer explicitly contradicts the reference answer. An example of a contradictory student answer for the example question above is: "a bulb causes a difference in electrical state."

Irrelevant: if the students answer is talking about domain content but not providing the necessary information. An example of an irrelevant student answer for the example

[18] `http://www.cs.york.ac.uk/semeval-2013/task7/index.php?id=data`

question above is: “a battery uses a chemical reaction to maintain different electrical states between two terminals.”

Non domain: if the student answer is not talking about domain content but expressing a request for help, frustration, or lack of domain knowledge. An example of non-domain student answer for the example question above is: “i do not know.”

The provided data consists of two distinct subsets, one that contains 56 questions in the electronics domain requiring 1 or 2 sentence answers and 3000 student answers, and the other consisting of 197 assessment questions in 15 different science domains with 10000 student answers.

Nine teams submitted their system’s results to this challenge. The overall accuracy of all systems in the 5-way task ranged from 12% up to 71%.

2.2.7 Analysis of Benchmarking

Because textual entailment covers various linguistic phenomena and different sets of inference tasks, it is very difficult to create a benchmark which addresses all these phenomena. This was obvious from the evolution of the RTE challenges, and their expansion each year to take into consideration additional phenomena. For example, the transition from RTE2 to RTE3 introduced longer texts which required anaphora resolution. In addition, the entailment started from a two classes of entailment and progressed to a five classes of entailment. One suggestion that might be better for benchmarking is [Sammons, Vydiswaran, and Roth, 2010]’s who suggest an explanation based analysis of RTE data. [Sammons, Vydiswaran, and Roth, 2010] proposed the annotation of RTE with inference steps to reach a decision, and hence give the ability to address more focused inference tasks. Perhaps even dividing the task into simpler subtasks, each involving a specific inference task, would be a better way to evaluate recognizing textual entailment.

Another difficulty with the current benchmarks, which is true in all NLP bake-offs in general, is related to the difficulty of evaluating the contribution of individual components or resources of an RTE approach. In RTE5, the ablation testing of all knowledge resources used by participants was mandatory. This was aimed at examining the importance of each resources used in RTE separately. Many NLP challenges have recognized the importance of such testing and today require such ablation. Information about the different resources used in RTE and the ablation tests will be described in more details in the next section.

2.3 Resources used in Recognizing Textual Entailment

In this section, we will survey the available resources typically used in recognizing textual entailment. We have limited the survey to resources that are either available for public use or accessible for research, and that have already been used by researchers to recognize textual entailment. Some of the resources include information about ablation performance as part of various RTE approaches. As described in Section 2.2, the ablation tests were aimed at examining the importance of each resources used in recognizing textual entailment. Ablation tests consist of removing one component or resource at a time from a system and re-running it on the same test set with the rest of the component and resources unchanged. Usually most of the resources used in RTE4 have a positive impact, but the reported impact of most of the resources is not statistically significant (between 0.10 and 6% accuracy) and some resources even have a negative impact. Consequently, determining the actual impact of a resource is not a straightforward task. It must be noted that a low impact may be due to many factors, for example, the coverage and precision of the resource itself, the manner in which it is used or even the accuracy of the base system in which it is used. Furthermore, many of the resources were not tested separately usually because the resource is an integral part of one's approach and testing without it would not be possible.

2.3.1 Acronym Lists

Acronym lists contain abbreviations that reference an actual phrase or sometimes just a word. The availability of such lists is important in textual entailment in order to be able to compare an acronym to its actual reference, or further expand the acronym with additional semantic information. The following presents some of the lists that were often used by researchers in recognizing textual entailment:

- The Acronym Guide[19]: is a set of acronym and abbreviation lists for English. The set contains 21 lists, and over a thousand acronyms and abbreviations in total. An example entry in the business acronym set: *CEO = Chief Executive Officer*. This list has been used by [Iftene and Balahur-Dobrescu, 2007] and [Varma, Bysani, Kranthi Reddy, Santosh GSK, Kovelamudi, Kiran Kumar, and Maganti, 2009] for recognizing textual entailment. An ablation test by removing the acronym module from the RTE system has been done for the RTE5 challenge and yielded a low positive impact of 0.17% on precision for the two way task for [Iftene and Balahur-Dobrescu, 2007]'s system but no impact on [Varma, Bysani, Kranthi Reddy, Santosh GSK, Kovelamudi, Kiran Kumar, and Maganti, 2009]'s system.

- BADC Acronym and Abbreviation List[20]: is a list of acronyms and abbreviations by the British Atmospheric Data Centre (BADC). An example entry in the BADC list is *KBPS= kilobits per second*. This list has been used by [Castillo and Alemany, 2008] for recognizing textual entailment, but its impact was not

[19] http://www.acronym-guide.com/
[20] http://badc.nerc.ac.uk/help/abbrevs.html

reported.

2.3.2 Nominalization Databases

Nominalization databases contain a collection of uninflected words and their related variants from different parts of speech. The importance of such databases in recognizing textual entailment is in relating different lexemes to the same concept. For example, “writer”, “writers”, and “writing” all relate to the concept “writing”. The following list contains some of the nominalization databases that were used often by researchers in recognizing textual entailment:

- NOMLEX (NOMinalization Lexicon)[21]: is a database of English nominalizations that describe the allowed complements for a nominalization in addition to their relation to the arguments of the corresponding verb [Macleod, Grishman, Meyers, Barrett, and Reeves, 1998]. The database contains 1025 entries selected from lists of frequently appearing nominalizations [Macleod, Grishman, Meyers, Barrett, and Reeves, 1998]. The following is a sample entry from the NOMLEX database:

```
(NOM ORTHOGRAPHY “writer”
     PLURAL “writers”
     VERB “write”
     NOM-TYPE ((SUBJECT))
     VERB-SUBCAT ((NOM-NP OBJECT ((N-N-MOD)
                                  (PP PVAL (“of”)))
                                  REQUIRED ((OBJECT)))
                  (NOM-INTRANS)))
```

  This entry for the noun ”writer”, is a nominalization of the verb “write”, that can have the plural form “writers”. The type of nominalization is (or NOM-Type) is a subject of a verb. The VERB-SUBCAT feature list the complements

[21] http://nlp.cs.nyu.edu/nomlex/

of the corresponding verb and provide information about how the verbal complement is realized as a nominal complement, in this example “writer” has one verbal complement NP as an object, that can appear as a noun modifier (N-N-MOD) or a post noun to the preposition “of” (PP). NOMLEX-plus from [Meyers, Reeves, Macleod, Szekeley, Zielinska, Young, and Grishman, 2004a] is a 7050 entry extension of NOMLEX that includes in addition to the original NOMLEX entries, 4900 entries for nominalizations of verbs, 550 entries for nominalizations of adjectives and 1600 entries that fall into 16 classes for argument taking nouns (such as PARTITIVE nouns, RELATIONAL nouns, ATTRIBUTE nouns, among others). NOMLEX-plus has been used by [Bar-Haim, Berant, and Dagan, 2009] as part of their lexical syntactic resource for recognizing textual entailment, but its impact has not been reported.

- CATVAR (Categorical Variation)[22]: is a database of uninflected words and their categorical variants [Habash and Dorr, 2003]. The database contains 63,146 clusters and 109,807 words and was created by combination of resources and algorithms. The following is an example cluster entry from the CATVAR database for the word “writer”:

  (Variants - Part Of Speech
  “write” - Verb
  “writer” - Noun
  “writing” - Noun
  “writings”- Noun
  “written” - Adjective)

  CATVAR has been used by [Shnarch, 2008] for recognizing textual entailment, but its impact has not been reported.

[22] http://clipdemos.umiacs.umd.edu/catvar/

2.3.3 Gazetteers

A gazetteer is a list that includes geographical information. It is useful in recognizing textual entailment, as it relates geographic features to their relevant information, such as the feature type, location, elevation, population. The following list contains some of the gazetteers used by researchers for recognizing textual entailment:

- GNIS - Geographic Names Information System[23]: is a database of 2 million records of geographic features in the United States that was created by the U.S. Geological Survey. The following is an example entry from the database:

Feature Name	Class	County	State	Latitude	Longitude	Elevation
Amherst Museum	Building	Erie	NY	430502N	0784342W	581 ft

  This database has been used by [Ageno, Farwell, Ferres, Cruz, and Rodríguez, 2008] for recognizing textual entailment, however its impact has not been reported.

- Geonames[24]: is a geographic database containing over 10 million geographical names. It includes integrating geographical information such as names of places in various languages, elevation, and population. The following is an example entry from the database:

Names	Country	Population	Latitude	Longitude
Montreal,Montréal...	Canada- QC	3,268,513	N 45 30 31	W 73 35 16

  Geonames has been used by [Ageno, Farwell, Ferres, Cruz, and Rodríguez, 2008] for recognizing textual entailment, however its impact has not been reported.

2.3.4 N-gram Models

N-gram models are probabilistic language models that have been widely used in various natural language processing applications, from language identification to machine

[23] http://nhd.usgs.gov/gnis.html
[24] http://www.geonames.org/

translation [Manning and Schütze, 1999]. They have also been used in recognizing textual entailment, mainly in assessing the similarity of terms based on the probabilistic information of the language model.

To date, the largest n-gram model is the Web 1T 5-grams from Google[25]. It contains English word n-grams and their observed frequencies. The n-gram counts were generated from approximately 1 trillion word tokens of text from publicly accessible Web pages [Brants and Franz, 2006]. The corpus contains in total about 13.5 million unique words, 314.8 million bigrams, 977 million trigrams, 1.3 billion four grams, and 1.1 billion five grams after discarding sequences appearing less than 40 times. The following is an example entry from the 4-gram dataset:

4 gram	Count
serve as the indication	72
serve as the indicator	120
serve as the indicators	45

The Web 1T 5-grams model has been used by [Yatbaz, 2008] for measuring the relevance of word pairs as part of a recognizing textual entailment system, however its impact has not been reported.

2.3.5 Semantic Networks

A semantic network is a network of concepts that are related to one another by semantic relations. Semantic networks between concepts are usually used in recognizing textual entailment to compute the similarity between two concepts available in the text and the hypothesis. Another use for semantic networks is to augment the knowledge representation of the text or hypothesis to further infer possible entailment between the two. The following list contains some of the semantic networks used by researchers for recognizing textual entailment. In our approaches for RTE, we have

[25] `http://www.ldc.upenn.edu/Catalog/CatalogEntry.jsp?catalogId=LDC2006T13`

used the WordNet and VerbOcean semantic networks as the main resources from this list.

- WordNet[26]: is a network of semantic relations between English words grouped into sets of synonyms (called synsets) [Fellbaum, 1998]. The semantic relations include hypernyms, hyponyms, holonyms, meronym, verb entailment, similar adjective, and other lexical relations. Each synonym includes a dictionary definition (or gloss), and an example sentence. In addition, nouns and verbs are organized into hierarchies using the hypernym relations. The following is an example entry from WordNet:

Word	Synonyms	Gloss		
Writer	(Writer, author)	writes books or stories professionally...		
	->Hypernym	(Communicator)		
		->Hypernym	(Person, individual...)	
			->Hypernym	(Organism)

  WordNet is probably one of the most used resources in recognizing textual entailment (and many other NLP applications). It is used either to recognize semantic relations or to compute the similarity between words. Out of 20 participants in RTE5's main task, 17 have used WordNet in their approach. The WordNet impact ranged from a negative to a positive one depending on the resource use, with the highest positive impact reported by [Clark and Harrison, 2008] and [Breck, 2009]. [Clark and Harrison, 2008] used WordNet's semantic relations to recognize equivalence between the text and the hypothesis, and reported a positive impact of the resource of 4.00% on the two way task and 5.67% on the three way task of RTE5. [Breck, 2009] used WordNet for lexicon based matching based on path distance between two words, and reported a positive impact of 4.00% on the three way task of RTE5.

[26] `http://wordnet.princeton.edu/`

- Augmented WordNet[27]: is the result of the application of a learning algorithm for inducing semantic taxonomies from parsed texts [Snow, Jurafsky, and Ng, 2004]. The algorithm automatically acquires items of world knowledge, and uses these to produce significantly enhanced versions of WordNet, adding up to 40,000 more synsets. For example, the algorithm learns that "deuterium" is a type of atom after reading the phrase "heavy water rich in the doubly heavy hydrogen atom called deuterium". Augmented WordNet has been used by [Bar-Haim, Berant, and Dagan, 2009] as a source of entailment rules in the task of recognizing textual entailment, however its impact has not been reported.

- Extended WordNet[28]: is another extension of WordNet which contains a logical representation of WordNet glosses [Harabagiu, Miller, and Moldovan, 1999]. For example, the gloss or definition of the word "excellent" is "of the highest quality", which results in the following logical form entry:

Left hand side	Right hand side
Excellent: JJ(x1)	->of:In(x1,x2) highest:JJ(x1) quality:NN(x1)

  Here, the argument *'x1'* refers to the adjective "Excellent" from the left hand side, and to the adjective "highest" and noun "quality" from the right hand side. Extended WordNet has been used by [Bar-Haim, Berant, and Dagan, 2009] as a source of entailment rules, and by [Iftene, 2008] who reported a positive impact with Extended WordNet of 1.33% on the three way task of recognizing textual entailment.

- VerbOcean[29]: is a broad coverage semantic network between verbs [Chklovski and Pantel, 2004]. The semantic relations covered include: similarity, opposite-of, stronger-than, can-result-in, and happens-before relations. The network

[27] `http://ai.stanford.edu/~rion/swn/`
[28] `http://xwn.hlt.utdallas.edu/`
[29] `http://demo.patrickpantel.com/demos/verbocean/`

contains a total of 29,165 strongly associated verb pairs, and has been created using a set of 35 manually created lexico-syntactic patterns. The following is an example entry from VerbOcean:

Word1	Semantic Relation	Word2	Score
Own	[happens-before]	sell	10.788965
Produce	[similar]	create	11.312437

VerbOcean has been used by many researchers in recognizing textual entailment with various results. [Wang, Zhang, and Neumann, 2009] reported a positive impact between 0.17% and 0.33% on the two-way task and 0.17% and 0.5% on the three-way task in RTE5. On the other hand, [Mehdad, Matteo, Elena, Milen, and Magnini, 2009] reported a negative impact of the resource of 0.16% on the two way task ablation test on RTE5.

2.3.6 Frames

Frames are knowledge representation schemes derived from semantic networks that are used to represent a typical situation [Minsky, 1995]. Several types of frames have been used by researchers for recognizing textual entailment. The most common ones include:

- FrameNet[30]: is based on the idea that the meanings of most words can be understood on the basis of a semantic frame: a description of a type of event, relation, or entity and the participants in it [Baker, Fillmore, and Lowe, 1998]. The following is an example entry from FrameNet:

  Frame: Reading

Frame Element	Explanation
Reader (Core element)	The one who examines a Text to understand it.

[30] https://framenet.icsi.berkeley.edu/

Text (Core element)	The entity that contains linguistic symbols.
Degree	Degree to which event occurs.
Manner	Manner of performing an action.
Place	Where the reading event takes place.
Purpose	The reason for which the Reader reads the Text.
Time	When the reading event takes place.
...	

FrameNet has been used by [Mirkin, Dagan, and Padó, 2010] to perform frame to frame similarity measurements in recognizing textual entailment. Its use had a positive impact on the two way task of 1.16% at RTE5. FrameNet has also been used by [Ofoghi and Yearwood, 2011] to judge if two concepts are equal if they belong to the same frame. For example, the two concepts "fly" and "pace" belong to the same frame (Self motion), and therefore are considered semantically similar. However, its impact has not been reported by [Ofoghi and Yearwood, 2011] with this particular usage.

- PropBank[31]: is a corpus containing verb frames from the Penn Treebank annotated with argument role labels [Kingsbury and Palmer, 2002]. The following is an example verb frame from PropBank:

Roleset id	Argument	Description
Read.01		
	Arg0	reader
	Arg1	thing read
	Arg2	benefactive or direction

Although the PropBank frames are similar to FrameNet's, they are more centered around verbs whereas a FrameNet frame may include several verbs. For example, the verbs: "write", "draft", and "compose" belong to the same FrameNet frame "Text Creation". Another difference is that PropBank arguments are

[31] http://verbs.colorado.edu/~mpalmer/projects/ace.html

closer to the syntactic level, whereas FrameNet frame elements are closer to the semantic level. PropBank has been used by [Ren, Ji, and Wan, 2009] as part of the syntactic parse, with a positive impact on the two way task of 2.00% and 3.17% for the three way task.

- NomBank[32]: is a corpus of noun frames [Meyers, Reeves, Macleod, Szekely, Zielinska, Young, and Grishman, 2004b]. Similarly to the verb frames from PropBank, NomBank provides argument structures for nouns from the Penn Treebank corpus. The following is an example noun frame from NomBank:

Noun	Argument	Description
Writer		
	Arg1	thing written
	Arg2	beneficiary
	(Example)	(an editorial writer for the Rocky Mountain News)
	(Arg1)	(editorial)
	(Arg2)	(for the Rocky Mountain News)

NomBank has been used by [Bar-Haim, Berant, and Dagan, 2009] as part of their lexical syntactic resources. [Bar-Haim, Berant, and Dagan, 2009] did not perform a separate ablation test for NomBank, but as part of the lexical syntactic resource, it had a positive impact of 0.70% on the RTE4 challenge.

- VerbNet[33]: is a corpus of verbs organized into classes, where each class is described by thematic roles, selectional restrictions on arguments, and frames [Kipper, Korhonen, Ryant, and Palmer, 2006]. The following is an example verb class from VerbNet:

Class	Roles	Restrictions
Create-26.4		
	Agent	animate or machine
	Result	

[32] http://nlp.cs.nyu.edu/meyers/NomBank.html
[33] http://verbs.colorado.edu/~mpalmer/projects/verbnet.html

	Material	
	Beneficiary	animate
	Attribute	
Frames		
NP V NP		
	example	“David constructed a house.”
	syntax	Agent V Result
. . .		

In this example, the class “Create”, can take the roles: “Agent” (in addition to the other shown above) with a restriction of “animate” or “machine” type of agents. One possible frame (or usage) of this class is the noun phrase/verb/noun phrase (NP V NP), as in the example sentence “David constructed a house.”, where the first NP has the role of an “agent” and the second NP has the role of a “result”. Similarly to FrameNet, a class may include several verbs. For example, the class “Create” in the example above includes 27 verbs, such as: “construct”, “create”, and “write”. VerbNet has been used by [Mirkin, Dagan, and Padó, 2010] to find correspondences between verbs, but they did not report its impact. [Roth and Sammons, 2007] also have used VerbNet but to pair verb argument patterns and they too have not reported its impact.

2.3.7 Thesauri and Encyclopaedias

Thesauri and encyclopaedias have been used as a resource for recognizing textual entailment, either as part of semantic similarity matching or for creating lexical entailment rules. The following list contains some of the thesauri and encyclopaedias used by researchers for recognizing textual entailment:

- Roget’s Thesaurus[34]: is an English thesaurus composed of six primary classes, each class composed of multiple divisions and sections [Roget, 1911]. Those

[34] `http://www.gutenberg.org/ebooks/10681`

classes contain semantically linked words (i.e. words or phrases of similar meaning). Although these classes can be seen as representin a hypernym relation, there are no explicitly defined semantic relations in this thesaurus. The following is an example entry from the Roget's Thesaurus:

Class	Title
Class IV	Words relating to the intellectual faculties
Division II	Communication of ideas
Section III	Means of communicating ideas
#3	Written Language
#590	Writing
	Vb. push the pen, push the pencil, write, pen...
	N. chirography, stelography, cerography...
	...

In this example, the set of words and phrases with similar meaning (push the pen, write, pen), belong to class #590 titled "writing", which is a subclass of class #3 titled "Written language" ... This thesaurus has been used by [Delmonte, Tonelli, and Tripodi, 2009] in recognizing textual entailment, and specifically for semantic similarity matching. It had a positive impact of 2.83% on the RTE5 two-way task.

- Dekang Lin's Theasurus[35]: is an automatically constructed thesaurus from a parsed corpus based on the distributional similarity score [Lin, 1998b]. The following is an example entry from Lin's thesaurus:

Word1	Word2	Similarity
brief (noun)	affidavit	0.13
	petition	0.05
	memorandum	0.05
	motion	0.05
	lawsuit	0.05
	...	

[35] `http://webdocs.cs.ualberta.ca/~lindek/downloads.htm`

In this example, the noun “brief” is similar to the word “affidavit” with a similarity score of 0.13. This thesaurus has been used by [Breck, 2009] in recognizing textual entailment, but they did not report its impact.

- Wikipedia[36]: is a free encyclopedia that has been used in recognizing textual entailment by various researchers and in different ways. For example, [Iftene and Moruz, 2009] have used it to identify the distance between named entities and they reported a low positive impact of 0.17% on the RTE5 two-way task. [Cabrio and Magnini, 2010] have used Wikipedia to extract rules using latent semantic analysis, and reported a positive impact of 1.00% on the RTE5 two way task.

2.3.8 Ontologies

An ontology is a formal and explicit specification of a shared conceptualisation [Gruber et al., 1993]. It formally represents knowledge as a set of concepts and relationships between those concepts and can be used to support reasoning over those concepts. The following list contains some of the ontologies used by researchers for recognizing textual entailment. In our baseline approach for RTE, we have used the Freebase ontology from this list.

- DBPedia[37]: is an open community ontology describing millions of concepts extracted from Wikipedia infoboxes [Auer, Bizer, Kobilarov, Lehmann, Cyganiak, and Ives, 2007]. The English version of DBPedia describes 2.35 million instances that are classified in an ontology. Those instances include 764,000 instances of persons, 573,000 instances of places, 333,000 instances of creative works, and 192,000 instances of organizations (as per the latest release of DBPedia #3.7).

[36] `http://www.wikipedia.org/`
[37] `http://dbpedia.org`

The ontology covers 359 classes which form a subsumption hierarchy, where each class may have one or more super classes. It uses the Resource Description Framework (RDF) to represent the information, and can also be queried via the DBPedia SPARQL Protocol and RDF Query Language (SPARQL) endpoint. This allows for the querying of complex queries such as *"Give me all cities in New Jersey with more than 10,000 inhabitants"* or *"Give me all German musician who were born in Berlin"*. For the later query DBPedia returns the following among others: "Alexander Marcus", "Andy Malecek", "Drafi Deutscher", . . .

DBPedia was used by [Delmonte, Tonelli, and Tripodi, 2009] in RTE5 in combination with other ontologies to confirm anaphoric links in bridging coreference, but no ablation test was performed to show its impact.

- YAGO[38] (Yet Another Great Ontology): is an ontology having more than 10 million entities and more than 120 million facts [Suchanek, Kasneci, and Weikum, 2007]. The information of YAGO was extracted from Wikipedia, WordNet, and GeoNames[39]. YAGO attaches temporal and spatial dimensions to many of its facts and entities. Similarly to DBPedia, YAGO is also represented in RDF and can be queried using a SPARQL end point. The following is an example query that can be asked on YAGO: *"Politicians who are also scientists, born nearby Hamburg, after the year 1900"*, which resulted in *"Helmut Schmidt", "Robert Heilbroner", "Angel Merkel", . . .*

  YAGO was used by [Delmonte, Tonelli, and Tripodi, 2009] in RTE5 in combination with other ontologies also to confirm anaphoric links in bridging coreference, but no ablation test was performed to show its impact.

[38] `http://www.mpi-inf.mpg.de/yago-naga/yago/`
[39] `http://www.geonames.org/`

- Freebase[40]: is a large collaborative ontology containing data harvested from online resources as well as individually contributed data from its users. Freebase data structure consists of set of nodes and a set of links that establish relationships between them, and holding over 125 million tuples, and more than 4000 types, and more than 7000 properties [Bollacker, Evans, Paritosh, Sturge, and Taylor, 2008]. We have used Freebase in our baseline approach described in Chapter 3.

- Umbel[41]: is a vocabulary and reference concept ontology, short for Upper Mapping and Binding Exchange Layer. It is an extracted subset of the OpenCyc project, providing data in an RDF ontology based on OWL2. It provides a reference structure of 25,000 concepts and 60,000 relationships among those concepts. Umbel was used by [Delmonte, Tonelli, and Tripodi, 2009] in RTE5 in combination with other ontologies to confirm anaphoric links in bridging coreference, but no ablation test was performed to show its impact.

2.3.9 Inference Rules

Inference rules are generalizations that are considered to be true if their premises are true. Inference rules have been used by many researchers in recognizing textual entailment. The following describe some publicly available resources of inference rules:

- DIRT[42]: is a method for collecting inference rules from text, and a knowledge collection of paraphrase expressions [Lin and Pantel, 2001]. The algorithm learns rules based on the distributional hypothesis over paths of dependency trees. DIRT collects rules of equivalent paths when they tend to link to the same set of words. The DIRT algorithm extracted 7 million paths from 231

[40] http://www.freebase.com/
[41] http://www.umbel.org/
[42] http://demo.patrickpantel.com/demos/lexsem/paraphrase.htm

000 parse tree of a 1GB newspaper text and resulted in 12 million rules. The following is an example of the top paraphrases formed using DIRT:

X solves Y ->

X solution to Y

Y is resolved in X

Y is solved through X

...

DIRT has been used by many researchers in recognizing textual entailment, such as [Iftene, 2008] in RTE4 to map relations between words with a positive performance of 0.70% on the two way task, and [Bar-Haim, Berant, and Dagan, 2009] in RTE5 with a positive impact of 1.33% on the two way task.

- WikiRules[43]: is a database of 8 million lexical reference rules extracted from Wikipedia [Shnarch, Barak, and Dagan, 2009]. A lexical reference rule is a directional relation that is more general than a regular lexical relation (such as hypernym or synonym). The following is an example rule from WikiRules:

Left hand side	Right hand side
Bentley	->luxury car
Abbey Road	->The Beatles

In the first example rule above, the left hand side of the rule "Bentley" is equivalent to the right hand side of the rule "luxury car". WikiRules have been used by [Bar-Haim, Berant, and Dagan, 2009] with mixed impacts. In RTE4, it gave a low positive impact of 1.00%, but in RTE5 a low negative impact of 1.00% on the accuracy of the two way task.

[43] `http://u.cs.biu.ac.il/~nlp/downloads/WikiRules.html`

2.4 Conclusion

In this chapter, we have surveyed the different approaches for recognizing textual entailment which have been used in past work. Our approaches to RTE (Chapter 3, 4, and 5) belong to the category of logical form based methods (described in Section 2.1.6), our main differentiator with the rest of the approaches of that category is that we use semantic relatedness information as an alternative to a large knowledge base of axioms and inference rules. This will be described in details in Section 4.3

We have also looked at the available benchmarks for evaluating the different RTE approaches. In our RTE approaches, we have used the RTE1 and RTE2 data sets (described in Sections 2.2.1 and 2.2.2) for development, and have used the RTE3 and RTE4 data sets (described in Sections 2.2.3 and 2.2.4) for evaluating and benchmarking our approaches. We have chosen the data sets of the first 4 RTE challenges as they fit the scope of our thesis, of recognizing textual entailment, whereas the subsequent challenges have evolved from recognizing textual entailment to the search and extraction of textual entailment situated mainly in the summarization application setting.

Finally, this chapter described the different resources used by most RTE approaches. As we will see in Chapters 3 and 4, our approaches use the WordNet and VerbOcean semantic networks (described in Section 2.3.5) and the Freebase ontology (described in Section 2.3.8). In the next chapter, we will present our first approach for recognizing textual entailment.

Chapter 3

Baseline Knowledge Querying Approach

In this chapter we present our first approach to recognize textual entailment. This approach is based on a logical representation, knowledge querying, and semantic relatedness. As we mentioned in Chapter 2, the main disadvantage of logical based approaches is that they typically require a large knowledge base of axioms to be able to show that a hypothesis is a consequent of a certain text. The goal of this chapter is to introduce a baseline method, and show how far we can go using a shallow logical method in the task of recognizing textual entailment, without the prerequisite of a large knowledge base to recognize textual entailment. We propose an approach for recognizing textual entailment that is based on a logical form for representation, knowledge querying for comparison, and a set of rules for decision making. The next section will present an overview of this approach, followed by a detailed description of its prototype implementation, and an evaluation of the implementation.

3.1 Overview of the Baseline Knowledge Querying Approach

Similarly to most textual entailment approaches, our knowledge querying approach can be divided into three main components: a representation component, a comparison component, and a decision component. First, the representation component represents the text in description logic. Then, the comparison component compares the created representations with the purpose of creating a knowledge base query. Finally, the decision component classifies the knowledge querying results into entailment results. Figure 1 shows an illustration of our approach. The different components are shown in dashed boxes, the inputs and outputs in ovals, and the sub-components in rectangles. The representation component involves the representation of the text in description logic to facilitate the comparison and decision making (described in Section 3.2). This component comprises a logical form creator that transforms a text into logical form. The creator follows a three step procedure: syntactic analysis, semantic analysis, and ontological analysis. The syntactic analysis step creates a syntactic representation of the text using a dependency parser and a set of transformation rules. This is followed by the semantic analysis step, which creates a semantic representation of the text using a named entity recognizer, a noun compound interpreter, and a set of rules. Finally, an ontological analysis step creates a description logic based representation (DL) of the text, using a set of transformation rules and the WordNet semantic network.

As shown in Figure 1, the comparison component will then compare the description logic representation of the text with the hypothesis, and collect comparison information to formulate a knowledge base query (described in Section 3.3). This component is made of a query formulator that compares the hypothesis to the created description logic representation in order to create a knowledge base specific query. The comparison relies on a reasoner and a semantic relatedness calculator as a substitute for additional knowledge to compare the text representation with the hypothesis. The purpose of this comparison is to create a query from the hypothesis, using the concept and properties of the created description logic representation.

Finally, the decision component will use the top query result to decide whether the text entails the hypothesis or not (described in Section 3.4). The answers of that query on the created knowledge representation will be classified with a set of decision rules to decide on a textual entailment result.

In order to evaluate this approach we have created a prototype system. Figure 2 provides an overview of the system architecture, specifically showing the use of specific resources in the implementation. The main components are shown in dashed boxes, the inputs and outputs are shown in ovals, the sub-components are shown in rectangles, and the sub-component's resources in cylinders. In our implementation, we have chosen Minipar [Lin, 1998c] as a dependency parser, Freebase [Bollacker, Cook, and Tufts, 2007] as the source of named entity types, Nakov's approach [Nakov and Hearst, 2006] for extracting noun compound types, and the WordNet [Fellbaum, 1998] semantic network with a set of manually created transformation rules for the logical form creator part. In addition, we have used the RACER reasoner [Haarslev and Möller, 2003] and WordNet based similarity measures in formulating the query and comparing it with the text. Finally, a set of manually created rules is used in the decision making. In the following sections, we will describe each component in detail, starting with the knowledge representation component.

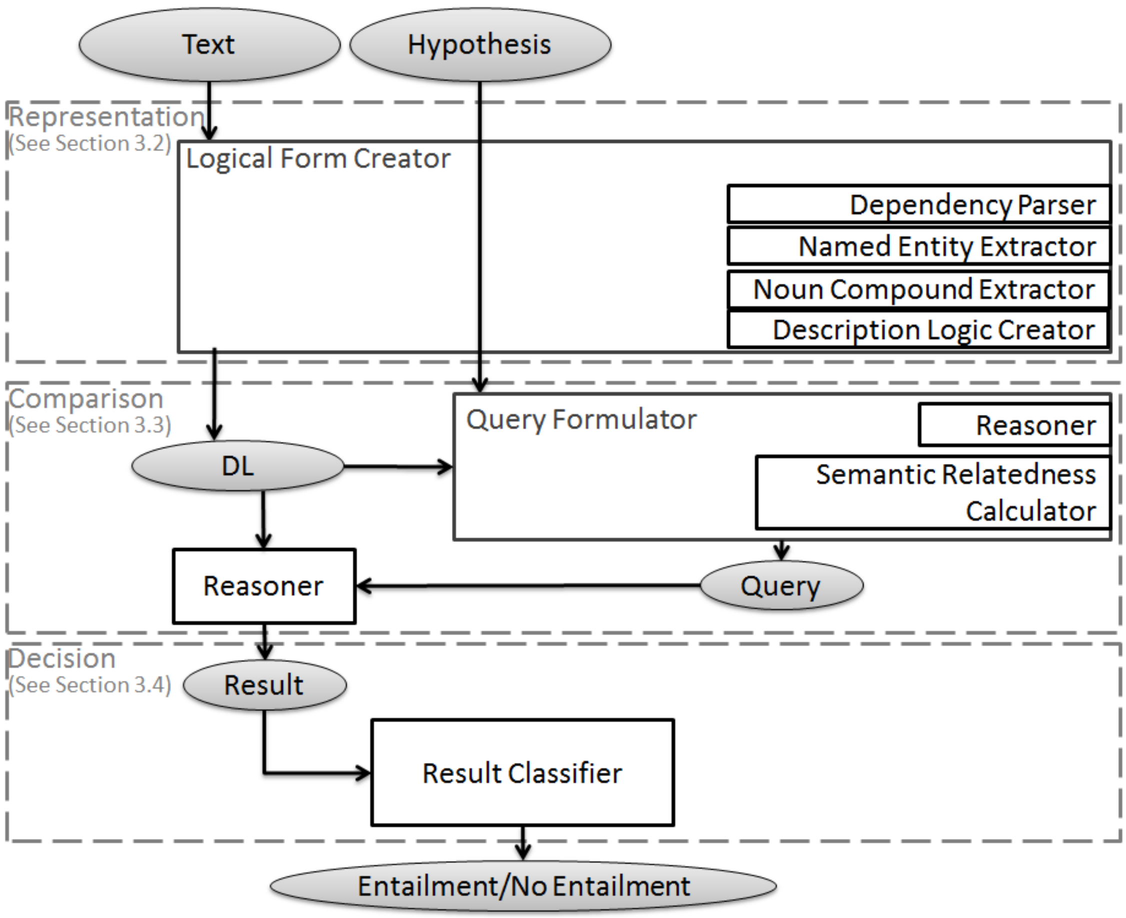

Figure 1: Baseline Knowledge Querying Approach

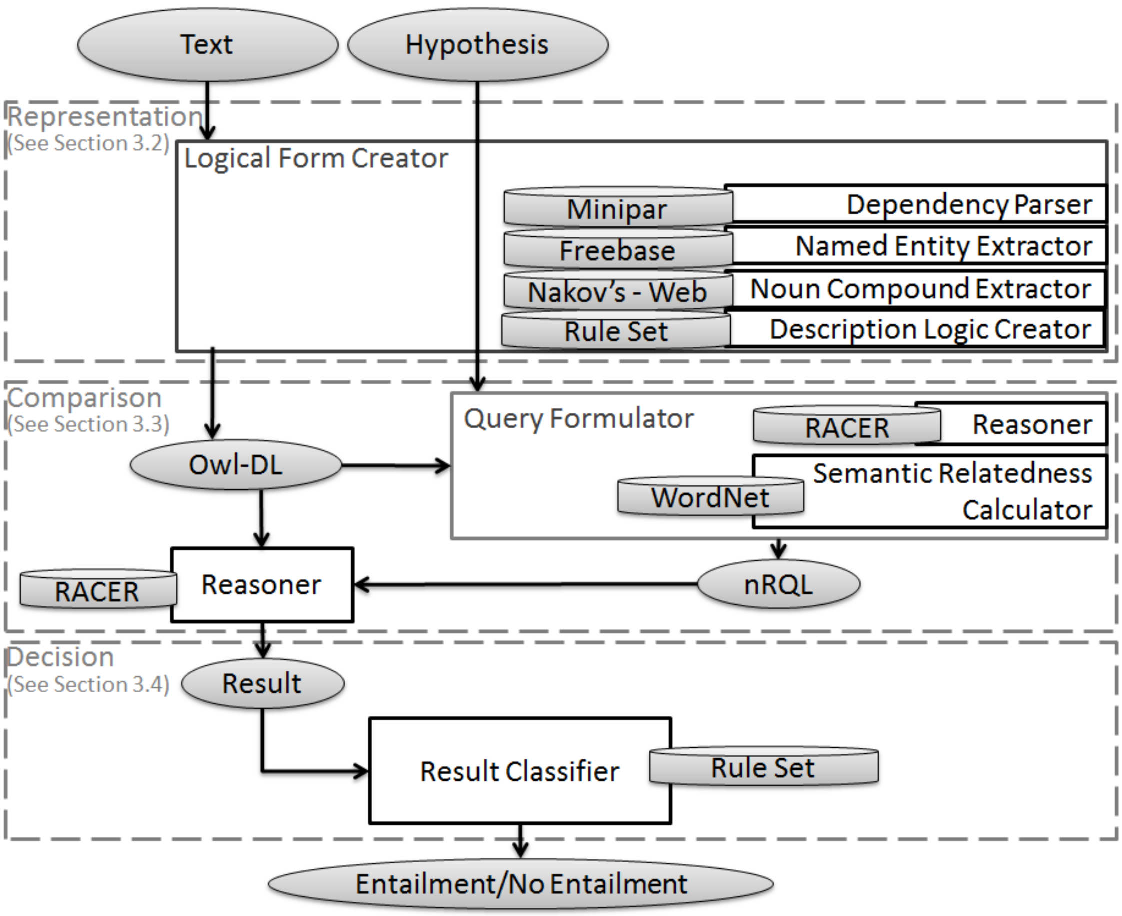

Figure 2: Prototype of the Baseline Knowledge Querying Approach

3.2 Representation

In order to recognize textual entailment, we need a representation that bridges the gap from linguistic inputs to knowledge of the world. As we have seen from Chapter 2, that logical based approaches have one of the best performance in RTE. The representation that we are focusing on, is a formal logical based representation which is expressive enough yet decidable (i.e. has an effective method to determine the truthfulness of a statement). One type of representation that adheres to our requirements is Description logic (DL) [Baader, 2003].

DL is a logical based knowledge representation formalism descendant of semantic networks, that can describe a domain in terms of concepts (sets of objects), roles (a binary relation between individuals), individuals (or instances of a concept), and their relations. The fundamental modeling concept of description logic, are logical statements that relate concepts or roles (axioms), which allow the building of complex concepts and roles from simple or atomic ones. A DL knowledge base consists of a TBox (terminological box) describing concepts and their relations, and an ABox (assertional box) describing ground sentences between individuals. DL is distinguished by a formal semantics (typically model theoretic), with a decidable fragment of First Order Logic (FOL), having a sound and complete decision procedure and highly optimised implemented inference systems.

The Semantic Web is a collaborative effort led by the World Wide Web Consortium (W3C) that provides a framework for making the World Wide Web content processable by machines [Berners-Lee, 1998]. W3C endorsed the web ontology language (OWL) [McGuinness, Van Harmelen, et al., 2004] as the language for the Semantic Web [Dean, Schreiber, et al., 2004]. OWL is a semantic markup language for defining and instantiating web ontologies, and is based on description logic. It is a vocabulary extension of RDF (the Resource Description Framework) [Brickley and Guha, 2004], derived from the DAML+OIL (DARPA Agent Markup Language and Ontology Interchange Language) [Horrocks et al., 2002], and based on XML (Extensible Markup Language) [Bray, Paoli, Sperberg-McQueen, Maler, and Yergeau, 1997]. As described in Section 2.3.8, an ontology is an "explicit specification of a conceptualization" [Gruber, 1995]. A conceptualization is a simplified view of the world that we wish to represent, and is explicitly specified by a set of vocabulary (concepts, roles, individuals, and other entities) to describe the domain of interest. An OWL ontology includes descriptions of **classes** (or concepts) that represent a collection of objects, e.g. person, **properties** (or role) that represents a binary directed relation between instances of a domain and range class, e.g. has-father, **individuals** that represents objects in the world belonging to a class, e.g. Tom type Person, and a set of axioms. OWL can be viewed as expressive description logic with an ontology being equivalent to DL knowledge base. We have selected OWL-DL[1], as an alternate decidable notation of DL language SHOIN(D) [Horrocks and Patel-Schneider, 2003], as the structure for representing the text knowledge. A main feature of DL is that it is a formal language with well-defined semantics. The standard way for specifying the semantic of DL is using a model theoretic semantics that can explain the relation between the DL syntax and the intended model of a domain. A model consists of domain 1 (set of objects) and an interpretation function I (a mapping from individual, class, and property names to elements of the domain). This interpretation function provides the necessary bridge between the representation and the domain being considered. Figure 3 (taken from [Horrocks and Patel-Schneider, 2004]) shows the construct names, syntax, and model semantics of the SHOINQ description logic, where A is a concept name, C and D are concepts, R and S are roles, RC is the set of transitive roles, o is an individual name, P is a simple role, and n is a non-negative integer. So for every

[1]urlhttp://www.w3.org/TR/owl-guide/

Construct Name	Syntax	Semantics	
atomic concept	A	$A^{\mathcal{I}} \subseteq 1^{\mathcal{I}}$	
atomic role	R	$R^{\mathcal{I}} \subseteq 1^{\mathcal{I}} \times 1^{\mathcal{I}}$	
transitive role	$R \in \mathbf{R}_{\mathcal{C}}$	$R^{\mathcal{I}} \mathcal{D} (R^{\mathcal{I}})^{\mathcal{C}}$	
conjunction	$C \sqcap D$	$C^{\mathcal{I}} \cap D^{\mathcal{I}}$	
disjunction	$C \sqcup D$	$C^{\mathcal{I}} \cup D^{\mathcal{I}}$	$\mathcal{S}$
negation	$\neg C$	$1^{\mathcal{I}} \setminus C^{\mathcal{I}}$	
exists restriction	$\exists R.C$	$\{x \mid \exists y.\langle x \rangle y \in R^{\mathcal{I}}$ and $y \in C^{\mathcal{I}}\}$	
value restriction	$\forall R.C$	$\{x \mid \forall y.\langle x \rangle y \in R^{\mathcal{I}}$ implies $y \in C^{\mathcal{I}}\}$	
role hierarchy	$R \sqsubseteq S$	$R^{\mathcal{I}} \subseteq S^{\mathcal{I}}$	$\mathcal{H}$
nominal	$\{o\}$	$\{o^{\mathcal{I}}\}$	$\mathcal{O}$
inverse role	R^{-}	$\{\langle x \rangle y \mid \langle y \rangle x \in R^{\mathcal{I}}\}$	$\mathcal{I}$
number	$\geqslant n\, P$	$\{x \mid \sharp\{y.\langle x \rangle y \in P^{\mathcal{I}}\} \geqslant n\}$	$\mathcal{N}$
restrictions	$\leqslant n\, P$	$\{x \mid \sharp\{y.\langle x \rangle y \in P^{\mathcal{I}}\} \leqslant n\}$	
qualifying number	$\geqslant n\, P.C$	$\{x \mid \sharp\{y.\langle x \rangle y \in P^{\mathcal{I}}$ and $y \in C^{\mathcal{I}}\} \geqslant n\}$	$\mathcal{Q}$
restrictions	$\leqslant n\, P.C$	$\{x \mid \sharp\{y.\langle x \rangle y \in P^{\mathcal{I}}$ and $y \in C^{\mathcal{I}}\} \leqslant n\}$	

Figure 3: Syntax and Semantics of *SHOINQ* Description Logic from [Horrocks and Patel-Schneider, 2004]

concept A, an interpretation function assigns a set $A^I \subseteq$ to 1^I. In addition for every atomic role R, an interpretation function assigns a binary relation $R^I \subseteq$ to 1^I X 1^I. The interpretation is then extended by inductive definitions which is summarized in the figure.

The main reason for selecting OWL-DL for our representation is that in addition to being based on a formal decidable logic, it also provides a well-studied set of reasoners that can be used to reason over the created knowledge base schema and instances, the accessibility of well-studied powerful query formalism, the possibility of applying rules to the created knowledge base using backward or forward chaining, and the ability to integrate multiple knowledge bases. These features are very useful for the task of RTE that, to our knowledge, most other approaches have not used in the past. In addition, the OWL language was used because of its design that was motivated by practical considerations, which emphasises on readability and general ease of use for the ontology language, making it very easy to read and understand when compared with other logical-based languages.

The logical form creator is our process for automatically creating an OWL-DL representation from text, and for the specific purpose of recognizing textual entailment. The basis of an OWL data model is based upon the idea of making statements about resources in the form of *property(Domain class, Range class)* expressions (or triples in RDF terminology). The property expresses a relationship between the domain class and the range class. The logical form creator follows a syntax-driven pipeline architecture, which is augmented by knowledge on demand, to transform a text to OWL. A text is first passed through a syntactic analysis process to derive its dependency parse. This dependency parse is then passed as input to a semantic analyser to produce a literal meaning representation. Finally, this meaning representation is passed as input to an ontological analyser that produces a formal logical-based meaning representation in OWL-DL. The transformation of the structure at each stage in the pipeline is shown below:

1. Syntax analysis: transform a sentence into a set of dependency relations: *dependency-relation(Governing content word, Modifying content word)*

2. Semantic analysis: transform the set of dependency relations into a set of semantic roles relations: *semantic-role(Predicate, Argument).*

3. Ontological analysis: transform a set of semantic role relations into a set of OWL properties: *property(Domain class, Range class).*

The driving idea of creating the final representation (and the intermediate ones) was to focus on precision as opposed to recall. In other words, in our view, it is more important to create a correct but incomplete representation as opposed to a more complete representation that may contain errors. Hence the driving force of all our transformation rules is to keep and represent only information in which we are confident.

The following sections will explain in more details how each step performs its transformation, starting from the syntactic analysis step. In order to present and demonstrate the prototype's steps in details, we will use the following text as a running example throughout the following sections:

Example 1:*Jurassic Park is a novel written by Michael Crichton and Published in 1990.*

For this example, our main goal is to create an OWL-DL representation as shown graphically in Figure 4. The figure shows a graphical illustration of part of the resulting OWL representation for example 1 with the added WordNet axioms. In this figure the classes are shown in ovals, the properties as arrows, the individuals in rectangle, and equal for equivalent axioms. A complete version of the resulting representation of example 1 is given in Appendix A.

3.2.1 Syntactic Analysis

The first step in our pipeline is the syntactic analysis phase, with the main purpose of transforming each sentence into a set of dependency relations. As the meaning of a sentence is not only based on the meaning of its words but also on the relations between those words in the sentence, a syntactic analysis is required to make explicit those relations. Various syntactic parsers that produce a grammatical representation

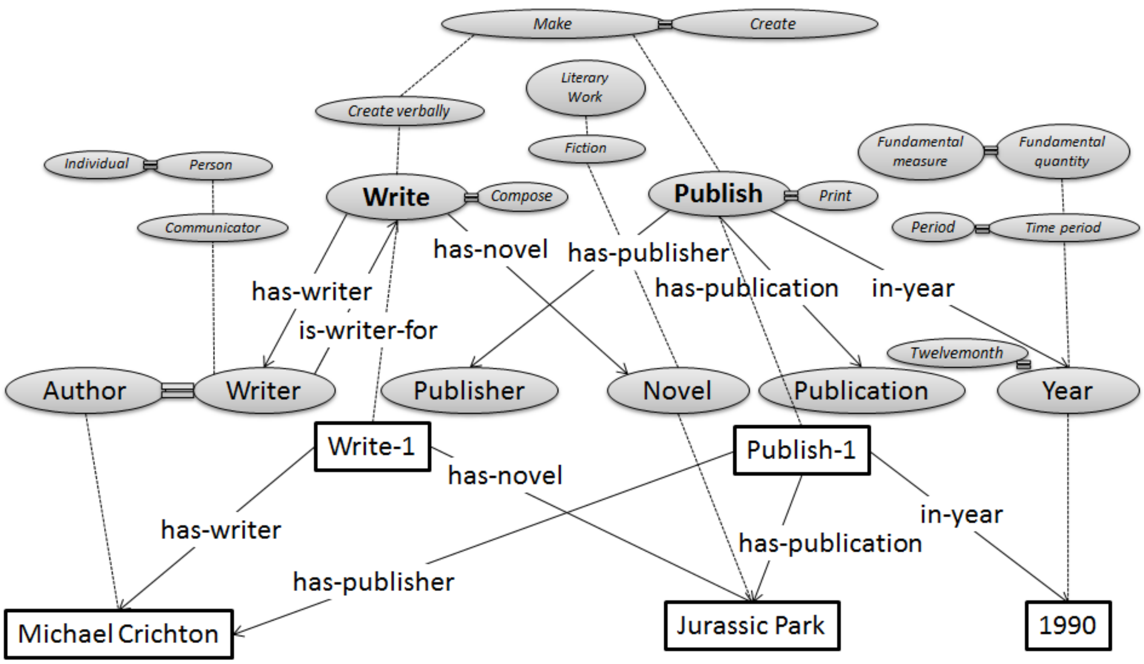

Figure 4: A Graphical Illustration of Part of the Representation Created for the Text "*Jurassic Park is a novel written by Michael Crichton and Published in 1990.*" with WordNet axioms.

Dependency relationship	Governor	Modifier	Stem	POS
lexical modifier	Jurassic Park	Jurassic	-	undefined
subject	be	Park	Jurassic Park	noun
independent clause	-	is	be	Verb to be
subject	novel	-	Jurassic Park	noun
determiner	novel	a	-	determiner
predicate	be	novel	-	noun
rel	novel	written	write	verb
object	write	-	novel	noun
by-subject	write	by	-	preposition
lexical modifier	Michael Crichton	Michael	-	undefined
preposition complement	by	Crichton	Michael Crichton	Noun
punctuation	write	and	-	undefined
conjunction	write	published	publish	verb
subject	publish	-	novel	noun
modifier	publish	in	-	preposition
preposition complement	in	1990	-	noun

Table 1: Dependency Parse of "*Jurassic Park is a novel written by Michael Crichton and Published in 1990.*"

from a sentence are available. Although most of the syntactic parsers use a context-free grammar, we are more interested with dependency based parsers as their resulting structure is described only in terms of words and binary relations between them. Most relations represent grammatical relations, but others also represent semantic relations. [Hays, 1964] define a dependency relationship as a binary relationship between a word called the "head" and another called the "modifier". A dependency based representation limited to a binary relations between words is more appropriate to our problem as constituents and phrase structures do not play any role in our representation. One of the dependency parser that parses a sentence into a set of dependency relations is the Minipar parser [Lin, 1998c]. Minipar is a very efficient dependency parser for the English language that was able to achieve about 88% precision and 80% recall with respect to dependency relationships on the SUSANNE corpus [Sampson, 2002].

Table 1 shows the Minipar parse of example 1. The table shows the dependency parse as a set of dependency relationships between a governor (or head) and a modifier, in addition to the stem of the modifier (Stem) and Part Of Speech (POS). A dependency relation is a relation between a word (a governor) and its dependents (a modifier), and the Stem refers to the part of the word that is common to all it inflected variants, and the POS refers to the linguistic category (noun, adjective, verb...) of the modifier. A record from the table shows that an "Object" dependency relation exist between the governing word "Write" and the modifier having the stem "novel", which is a "noun". The result is a set of dependency relations of the following form *dependency-relation (Governing Word, Modifying Word).* In this representation, the governing and modifying words can be either a content word (verbs, nouns, adjectives, and adverbs), or non-content words. Non-content words (or function words such as prepositions, conjunctions, auxiliary verbs, articles ...) are words that mainly serve to express grammatical relationships between content words in the sentence and have little meaning attached to them. Consequently, a transformation of the dependency relation set is required to further restrict the variability of the presentation and to represent the underlying meaning as precisely as possible yet be general enough to allow for reasoning. The underlying meaning will be represented as classes in an OWL representation and the relationships between them, as such the following transformations are performed:

- Transformation 1: We further restrict the governing and modifying words to the content words and discard any other word except for prepositions and conjunctions.

- Transformation 2: Prepositions are rewritten into dependency relations as they are a grammatical class that represent relations between content words. In example 1, the preposition "In" is related to the word "1990", and to the governor "Publish". Those two dependency relations, will be merged into one relating

Relation	**Governor**	**Modifier**
lexical modifier	Jurassic Park	Jurassic
subject-is	Jurassic Park	novel
object	write	novel
lexical modifier	Michael Crichton	Michael
subject	write	Michael Crichton
subject	publish	Michael Crichton
object	publish	novel
in	publish	1990

Table 2: Result of the Syntactic Analysis for "*Jurassic Park is a novel written by Michael Crichton and Published in 1990.*

the governor "Publish" to "1990" by the relation "In".

- Transformation 3: Conjunctions are re-written into a set of dependency relations as conjunctions may connect two larger syntactic structures to each other, and not only two words. In order to capture the semantics of the sentence we duplicate the dependency relations that exist between a word and its conjunction. In example 1, the conjunction "and" relates the word "Write" and "Publish", in this case, "Michael Crichton" the "Subject" of "Write" is propagated to become the subject of "Publish" as well.

The result of these transformations is similar to collapsed dependencies as we are collapsing conjunctions and prepositions into a single relation. The main difference with collapsed dependencies is that the first transformation restricts the governing and modifying words to content words, so dependency relations such as determiner or punctuation, for example, are discarded in our syntactic representation. Table 2 shows the set of relations that result after performing the above transformations. The table shows a set of *dependency-relation (Governing Word, Modifying Word)* for the text, where all governing and modifying words are content words.

3.2.2 Semantic Analysis

The main purpose of the semantic analysis process is to further transform the set of related dependency relations into a set of semantic role relations; more specifically, transform the set of dependency relations from the previous step:

dependency-relation(Governing content word, Modifying content word)

into a set of semantic roles relations:

semantic-role(Predicate, Argument).

This is an important step toward building a deeper text meaning, by abstracting from the syntactic structure and into related predicate argument structure, and where each argument plays a specific semantic role. The importance of this step is that it further explains the semantic role an argument plays with respect to the action described by a predicate.

Human languages can be represented by predicate-argument structures at the core of their semantic structure [Jurafsky, Martin, Kehler, Vander Linden, and Ward, 2000]. Verbs in particular, can be thought of as logical predicates and their lexical relations as the predicate logical arguments. In addition, as verbs have sub-categorization requirements, we can link the lexical relation with a semantic role and further restrict those roles to certain conditions. For our example, we can define the semantic predicate for the verb *Publish* with 3 arguments:

Subject(Michael Crichton), *Object(Novel)*, and *In(1990).*

We can then link a verb dependency relation with a semantic role that an argument plays in the representation. For example, the *subject* can be linked to the role of a *publisher* or *author*. The resulting predicate-argument structure can become the basis of the semantic representation. However, in OWL a *property* is a binary relation that links two individuals to each other, but as we have seen in natural language, a predicate can relate to more than just two arguments. To be able to represent a text in a natural way, but still use OWL, we decided to use an n-ary relation pattern where each *property* is a relation between a predicate and its argument, or more precisely an OWL *property* can represent the semantic role that an argument plays with respect to a predicate in a sentence. Consequently, a sentence is transformed into a set of *semantic roles (predicate, Argument)* triplets, which can create the basis of the ontological representation of that text. This transformation further restricts the dependency relations, from the previous step, by limiting the governing word to a predicate (except for named entities), and transforming the "dependency -relation" into a "semantic-role"; i.e.:

from *dependency-relation(Governing content word, Modifying content word)*

to

semantic-role(Predicate, Argument).

To do so, a set of transformation rules have been created to deal with the following cases: a named entity modifying a verb, a noun or adjective modifying a noun, and transforming a dependency relation to a semantic role.

Named Entities: Named entities are terms which designate an instance of a concept (as in the name of a person). They basically will represent an individual of an OWL class in an OWL representation. Finding the classes that individuals belong to will help in adding implicit information into our representation. In some cases, named entity types information is available in the actual text, such when there are predicates or appositives that describe a named entity. This could be seen in our example, where "Jurassic Park is novel (*"subject-is(Jurassic Park, novel)"*), define the semantic type or class of "Jurassic Park" to be a "novel". If the class of a named entity is not present in the text, we use a shared online ontology of structured knowledge called FreeBase to search for it. FreeBase (described in Section 2.3.8) contains named entities with their general and specific types. We have chosen the Freebase ontology because it is the largest ontology with over 125 million instances at the time. For example, according to FreeBase, the named entity: *"Michael Crichton"* belongs to the types *"author, director, producer ..."*. The first type from the FreeBase result is selected as the semantic type of the named entity; in this example, *"author"* is selected. Once we know the semantic type of a named entity, we replace every occurrence of that named entity in the set of dependency relations to the named entity type. This assumes that in a single text, all occurrences of the same named entity will refer to the same extra linguistic entity. For example, *subject(write, Michael Crichton)* becomes *subject(write, author(Michael Crichton))*. In this case, a "subject" relation exist between the predicate "write" and the argument "Michael Crichton" of type "author".

Noun Relations: As we have restricted the governing word to the verb class, in an effort to limit syntactic variations and give more meaning to the structure, we need to find the relation (predicate) that is usually available between the noun (or adjectives) modifying other nouns. A predicate will preserve the structure that we are looking for, and will make explicit the implicit common-sense knowledge available between the compounds. In some cases, a simple transformation of a nominalized verb into a verb, or a genitive relationship to verb is sufficient. However, in most cases, we need to extract the verb through other methods. To do this, we used the unsupervised method described by [Nakov and Hearst, 2006]. This method allows for the extraction of predicates from the Web that explain a noun relations. More specifically, this method relies on the use of relative clause based patterns of the form *"noun2 THAT * noun1"*[2] as web queries. For example, for the noun-noun relation *"fruit tree"*, a set of Web queries will be created following the relative clause pattern such as *"tree that * fruit"*. The result's most frequent verb characterizing *"fruit tree"* is *"bear"*, which could be further explained as *"a fruit tree is a tree that bear fruit"*.

[2] The * is a search engine wildcard that could be filled by one or more words.

Semantic Role: To transform dependency relations to semantic roles, we have created a set of transformation rules that aim to add more knowledge to the relation that exists between a predicate and its arguments. The semantic role in our case is meant to take into consideration the syntactic information of the dependency relation, the semantic information of the argument type from the previous step, and WordNet cross part of speech relations. WordNet's cross part of speech relations include the links that hold among semantically similar words sharing a stem with the same meaning. Following the same decision made for named entities and noun relations, we do not limit the semantic role to a specific set. The following is the set of transformation rules:

1. For a subject/object relation, we use the WordNet cross part of speech relations that is most related to the argument type. For example, *write (verb)* and *writer (noun)*. We then use the argument type to select the cross part of speech relation that is mostly related to the argument we are labeling. In our example, the dependency relation: *subject(write, author(Michael Crichton))* becomes *has-writer*[3]*(write, author(Michael Crichton))* as *writer* is the most semantically related concept to *author* from the list of cross part of speech relations to *write* in WordNet.

2. If the relation is a preposition, then the semantic role becomes the preposition followed by the argument type. In our example, the dependency relation: *in(publish, year(1990)* becomes *in-year(publish, year(1990)).*

3. Otherwise, the semantic role is simply the actual argument type. In our example, the dependency relation *object(write, novel)*, the argument is of type *novel* so the relation becomes *has-novel(write, novel).*

[3]We add the verb HAVE for readability purposes.

semantic-role	Predicate	Argument(Type)
has-novel	write	novel(Jurassic Park)
has-writer	write	author(Michael Crichton)
has-publisher	publish	author(Michael Crichton)
has-publication	publish	novel(Jurassic Park)
in-year	publish	year(1990)

Table 3: Result of the Semantic Analysis for the "*Jurassic Park is a novel written by Michael Crichton and Published in 1990.*

Table 3 shows the resulting set of semantic-role relations for the example text.

3.2.3 Ontological Analysis

The main purpose of the ontological analysis is to transform the semantic representation of the semantic analysis into a formal logical representation with formal semantics in OWL. More specifically, the purpose is to create an OWL DL representation from the semantic representation of the previous step (a set of *semantic-role (Predicate, Argument))*), into a set of OWL properties of the form *property (Domain class, Range Class).* Although the OWL language is easy to read, it is rather verbose in nature. Throughout this section we will use the OWL functional-style linear syntax. The functional-style syntax is suggested by the World Wide Web Consortium (W3C) as an easier to read syntax for humans that abstracts from exchange syntax and facilitates access to the language[4]. The following is a summary of the syntax that is relevant to this section:

- The OWL main entities are presented as follows: **datatype**, **owlClass**, **objectProperty**, **dataProperty**, and **individual**.

[4]urlhttp://www.w3.org/TR/owl11-syntax/

- OWL descriptions:

  The description construct **objectUnionOf** forms a disjunction of a set of descriptions, **objectIntersectionOf** is a conjunction of a set of descriptions, **objectComplementOf** is a negation of a description, and **objectOneOf** is a description that contains exactly the objects denoted by the set of specified individuals.

- OWL object properties restrictions:

  The construct **objectAllValuesFrom** denotes the set of objects that are connected via the given object property only to instances of the given description, **objectSomeValuesFrom** denotes the set of objects that are connected via the given object property to at least one instance of the given description, and **objectHasValue** denotes the set of objects that are connected via the given object property to the object denoted by the given individual.

- OWL class axioms:

  The **subClassOf** axiom states that one description is a subclass of another description. The **equivalentClasses** axiom states that a set of descriptions are all equivalent. The **disjointClasses** axiom states that a set of descriptions are pair-wise disjoint. Finally, the **disjointUnion** axiom defines a class as a union of descriptions, all of which are pair-wise disjoint.

- OWL property axioms:

  The **subObjectPropertyOf** axiom states that one description is a subproperty of another description.The **equivalentObjectProperties** axiom states that a set of object properties are all equivalent, and the **disjointObjectProperties** axiom states that a set of object properties are pair-wise disjoint. Furthermore, **objectPropertyDomain** and **objectPropertyRange** specify the domain and the range description, respectively, of an object property. Finally, **inverseObjectProperties** axiomatizes two properties to be inverse of each other.

- OWL facts axioms:

  The **sameIndividual** axiom states that each of the individuals from a given set denotes the same object, whereas the **differentIndividuals** axiom states that each of the individuals from a given set denotes a different object. The **classAssertion** axiom states that the object denoted by the given individual is an instance of the given description.

The formal meaning in model-theoretic semantics and their equivalence in description logic of the OWL constructions above is given in Figures 5 and 6 (taken from [Horrocks, Patel-Schneider, and Van Harmelen, 2003]). Full details on this model theory can be found in the OWL Semantics and Abstract Syntax [Patel-Schneider, Hayes, Horrocks, et al., 2004]. The main purpose of the ontological analysis is to create an OWL DL representation from the semantic-role relations of the previous step, to an OWL properties *property (Domain class, Range Class).* This is performed through a set of transformation rules that first convert the semantic role into an *ObjectProperty*, the predicate and argument into an *owlClass*, and named entities into *individuals.* The following is a detailed description of the transformation rules:

Abstract Syntax	DL Syntax	Semantics
Descriptions (C)		
A (URI reference)	A	$A^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}}$
`owl:Thing`	$\top$	$\texttt{owl:Thing}^{\mathcal{I}} = \Delta^{\mathcal{I}}$
`owl:Nothing`	$\bot$	$\texttt{owl:Nothing}^{\mathcal{I}} = \{\}$
`intersectionOf(`C_1 C_2 `...)`	$C_1 \sqcap C_2$	$(C_1 \sqcap D_1)^{\mathcal{I}} = C_1^{\mathcal{I}} \cap D_2^{\mathcal{I}}$
`unionOf(`C_1 C_2 `...)`	$C_1 \sqcup C_2$	$(C_1 \sqcup C_2)^{\mathcal{I}} = C_1^{\mathcal{I}} \cup C_2^{\mathcal{I}}$
`complementOf(`C`)`	$\neg C$	$(\neg C)^{\mathcal{I}} = \Delta^{\mathcal{I}} \setminus C^{\mathcal{I}}$
`oneOf(`o_1 `...)`	$\{o_1, \ldots\}$	$\{o_1, \ldots\}^{\mathcal{I}} = \{o_1^{\mathcal{I}}, \ldots\}$
`restriction(`R `someValuesFrom(`C`))`	$\exists R.C$	$(\exists R.C)^{\mathcal{I}} = \{x \mid \exists y.\langle x, y\rangle \in R^{\mathcal{I}} \text{ and } y \in C^{\mathcal{I}}\}$
`restriction(`R `allValuesFrom(`C`))`	$\forall R.C$	$(\forall R.C)^{\mathcal{I}} = \{x \mid \forall y.\langle x, y\rangle \in R^{\mathcal{I}} \rightarrow y \in C^{\mathcal{I}}\}$
`restriction(`R `hasValue(`o`))`	$R : o$	$(\forall R.o)^{\mathcal{I}} = \{x \mid \langle x, o^{\mathcal{I}}\rangle \in R^{\mathcal{I}}\}$
`restriction(`R `minCardinality(`n`))`	$\geqslant n\, R$	$(\geqslant n\, R)^{\mathcal{I}} = \{x \mid \sharp(\{y.\langle x, y\rangle \in R^{\mathcal{I}}\}) \geqslant n\}$
`restriction(`R `minCardinality(`n`))`	$\leqslant n\, R$	$(\geqslant n\, R)^{\mathcal{I}} = \{x \mid \sharp(\{y.\langle x, y\rangle \in R^{\mathcal{I}}\}) \leqslant n\}$
`restriction(`U `someValuesFrom(`D`))`	$\exists U.D$	$(\exists U.D)^{\mathcal{I}} = \{x \mid \exists y.\langle x, y\rangle \in U^{\mathcal{I}} \text{ and } y \in D^{\mathbf{D}}\}$
`restriction(`U `allValuesFrom(`D`))`	$\forall U.D$	$(\forall U.D)^{\mathcal{I}} = \{x \mid \forall y.\langle x, y\rangle \in U^{\mathcal{I}} \rightarrow y \in D^{\mathbf{D}}\}$
`restriction(`U `hasValue(`v`))`	$U : v$	$(U : v)^{\mathcal{I}} = \{x \mid \langle x, v^{\mathcal{I}}\rangle \in U^{\mathcal{I}}\}$
`restriction(`U `minCardinality(`n`))`	$\geqslant n\, U$	$(\geqslant n\, U)^{\mathcal{I}} = \{x \mid \sharp(\{y.\langle x, y\rangle \in U^{\mathcal{I}}\}) \geqslant n\}$
`restriction(`U `maxCardinality(`n`))`	$\leqslant n\, U$	$(\leqslant n\, U)^{\mathcal{I}} = \{x \mid \sharp(\{y.\langle x, y\rangle \in U^{\mathcal{I}}\}) \leqslant n\}$
Data Ranges (D)		
D (URI reference)	D	$D^{\mathbf{D}} \subseteq \Delta_{\mathbf{D}}^{\mathcal{I}}$
`oneOf(`v_1 `...)`	$\{v_1, \ldots\}$	$\{v_1, \ldots\}^{\mathcal{I}} = \{v_1^{\mathcal{I}}, \ldots\}$
Object Properties (R)		
R (URI reference)	R	$R^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$
	R^{-}	$(R^{-})^{\mathcal{I}} = (R^{\mathcal{I}})^{-}$
Datatype Properties (U)		
U (URI reference)	U	$U^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}} \times \Delta_{\mathbf{D}}^{\mathcal{I}}$
Individuals (o)		
o (URI reference)	o	$o^{\mathcal{I}} \in \Delta^{\mathcal{I}}$
Data Values (v)		
v (RDF literal)	v	$v^{\mathcal{I}} = v^{\mathbf{D}}$

Figure 5: Syntax and Semantics of OWL DL Entities [Horrocks, Patel-Schneider, and Van Harmelen, 2003]

Abstract Syntax	DL Syntax	Semantics
Class(A partial C_1 ...C_n)	$A \sqsubseteq C_1 \sqcap \ldots \sqcap C_n$	$A^{\mathcal{I}} \subseteq C_1^{\mathcal{I}} \cap \ldots \cap C_n^{\mathcal{I}}$
Class(A complete C_1 ...C_n)	$A = C_1 \sqcap \ldots \sqcap C_n$	$A^{\mathcal{I}} = C_1^{\mathcal{I}} \cap \ldots \cap C_n^{\mathcal{I}}$
EnumeratedClass(A o_1 ...o_n)	$A = \{o_1, \ldots, o_n\}$	$A^{\mathcal{I}} = \{o_1^{\mathcal{I}}, \ldots, o_n^{\mathcal{I}}\}$
SubClassOf(C_1 C_2)	$C_1 \sqsubseteq C_2$	$C_1^{\mathcal{I}} \subseteq C_2^{\mathcal{I}}$
EquivalentClasses(C_1 ...C_n)	$C_1 = \ldots = C_n$	$C_1^{\mathcal{I}} = \ldots = C_n^{\mathcal{I}}$
DisjointClasses(C_1 ...C_n)	$C_i \sqcap C_j = \bot, i \neq j$	$C_i^{\mathcal{I}} \cap C_j^{\mathcal{I}} \{\}, i \neq j$
Datatype(D)		$D^{\mathcal{I}} \subseteq \Delta_{\mathbf{D}}^{\mathcal{I}}$
DatatypeProperty(U super(U_1)...super(U_n)	$U \sqsubseteq U_i$	$U^{\mathcal{I}} \subseteq U_i^{\mathcal{I}}$
domain(C_1) ...domain(C_m)	$\geqslant 1\, U \sqsubseteq C_i$	$U^{\mathcal{I}} \subseteq C_i^{\mathcal{I}} \times \Delta_{\mathbf{D}}^{\mathcal{I}}$
range(D_1) ...range(D_l)	$\top \sqsubseteq \forall U.D_i$	$U^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}} \times D_i^{\mathcal{I}}$
[Functional])	$\top \sqsubseteq \leqslant 1\, U$	$U^{\mathcal{I}}$ is functional
SubPropertyOf(U_1 U_2)	$U_1 \sqsubseteq U_2$	$U_1^{\mathcal{I}} \subseteq U_2^{\mathcal{I}}$
EquivalentProperties(U_1 ...U_n)	$U_1 = \ldots = U_n$	$U_1^{\mathcal{I}} = \ldots = U_n^{\mathcal{I}}$
ObjectProperty(R super(R_1)...super(R_n)	$R \sqsubseteq R_i$	$R^{\mathcal{I}} \subseteq R_i^{\mathcal{I}}$
domain(C_1) ...domain(C_m)	$\geqslant 1\, R \sqsubseteq C_i$	$R^{\mathcal{I}} \subseteq C_i^{\mathcal{I}} \times \Delta^{\mathcal{I}}$
range(C_1) ...range(C_l)	$\top \sqsubseteq \forall R.C_i$	$R^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}} \times C_i^{\mathcal{I}}$
[inverseOf(R_0)]	$R = (^{-}R_0)$	$R^{\mathcal{I}} = (R_0^{\mathcal{I}})^{-}$
[Symmetric]	$R = (^{-}R)$	$R^{\mathcal{I}} = (R^{\mathcal{I}})^{-}$
[Functional]	$\top \sqsubseteq \leqslant 1\, R$	$R^{\mathcal{I}}$ is functional
[InverseFunctional]	$\top \sqsubseteq \leqslant 1\, R^{-}$	$(R^{\mathcal{I}})^{-}$ is functional
[Transitive])	$Tr(R)$	$R^{\mathcal{I}} = (R^{\mathcal{I}})^{+}$
SubPropertyOf(R_1 R_2)	$R_1 \sqsubseteq R_2$	$R_1^{\mathcal{I}} \subseteq R_2^{\mathcal{I}}$
EquivalentProperties(R_1 ...R_n)	$R_1 = \ldots = R_n$	$R_1^{\mathcal{I}} = \ldots = R_n^{\mathcal{I}}$
AnnotationProperty(S)		
Individual(o type(C_1) ...type(C_n)	$o \in C_i$	$o^{\mathcal{I}} \in C_i^{\mathcal{I}}$
value(R_1 o_1)...value(R_n o_n)	$\langle o, o_i \rangle \in R_i$	$\langle o^{\mathcal{I}}, o_i^{\mathcal{I}} \rangle \in R_i^{\mathcal{I}}$
value(U_1 v_1)...value(U_n v_n))	$\langle o, v_i \rangle \in U_i$	$\langle o^{\mathcal{I}}, v_i^{\mathcal{I}} \rangle \in U_i^{\mathcal{I}}$
SameIndividual(o_1 ...o_n)	$o_1 = \ldots = o_n$	$o_i^{\mathcal{I}} = o_j^{\mathcal{I}}$
DifferentIndividuals(o_1 ...o_n)	$o_i \neq o_j, i \neq j$	$o_i^{\mathcal{I}} \neq o_j^{\mathcal{I}}, i \neq j$

Figure 6: Syntax and Semantics of OWL DL Axioms and Facts [Horrocks, Patel-Schneider, and Van Harmelen, 2003]

- Classes:

  Every predicate and argument in the set of semantic-role relations is transformed into an *owlClass*. For our example, we will therefore create five owlClasses: *owlClass(write)*, *owlClass(novel)*, *owlClass(author)*, *owlClass(publish)*, and *owlClass(year)*.

- Properties:

  Every semantic role is transformed into an *objectProperty*, with the *predicate* as a restriction of the *objectPropertyDomain* and the *argument* as a restriction of the *objectPropertyRange*.

  For example, the relation *has-novel(write, novel)* will be transformed into:

  ObjectProperty(has-novel)

  ObjectPropertyDomain(has-novel Write)

  ObjectPropertyRange(has-novel Novel)

  In addition, an *InverseObjectProperties* is created for each *objectProperty* of the form *is-objectProperty-for* ("is" replaces "has").

  Therefore in our example; for *ObjectProperty (has-novel)* we will also create:

  InverseObjectProperties (is-novel-for has-novel).

- Individuals:

  Individuals are created indirectly as an assertion of classes and properties. So every predicate is transformed into an instance by simply adding an index to the argument value being of type *owlClass(argument)*. For example, *ClassAssertion(Write Write-1)*. Each semantic-role relation is transformed into an instance of an object, having the predicate individual as its domain and either an individual of the argument (a classAssertion to the argument similarly to the predicate) or a named entity if it exists. So in our example, *has-novel(write, novel)* will be transformed into:

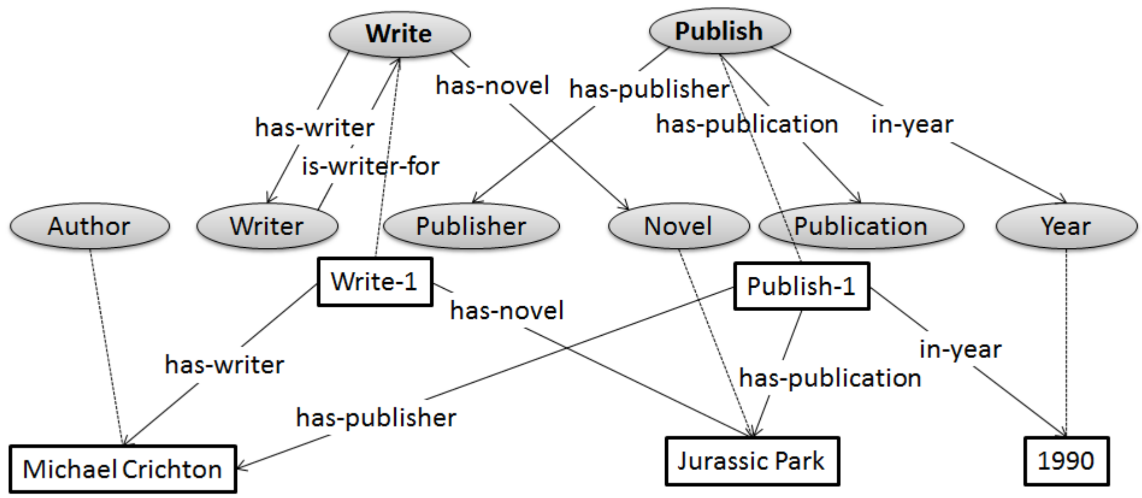

Figure 7: A Graphical Illustration of Part of the Representation Created for the Text "*Jurassic Park is a novel written by Michael Crichton and published in 1990.*"

ObjectPropertyAssertion(has-novel Write-1 Jurassic-Park).

Figure 7 shows a graphical illustration of the representation from our example. The figure shows the classes in ovals, the properties as arrows, and the individuals in rectangles. From the semantic analysis, the classes *novel*, *author*, and *year* are extracted from the argument type, the classes *write* and *publish* from the created predicates, while the classes *writer*, *publisher*, *publication*, and the roles *has-novel*, *has-writer*, *has-publisher*, and *in-year* are extracted from the semantic role relations.

Word knowledge is then added to the representation by extracting hypernym, synonym, and anotonym relations from the WordNet semantic network [Fellbaum, 1998] of the most frequent synset and transforming them into OWL axioms as follows:

- Hypernyms:

  For each class created in the representation, the hypernyms are retrieved from the WordNet semantic network, and transformed into a *subClassOf* axioms. Only 2 levels up are considered in order not to include too general information, For our example, the "Novel" class has the hypernym "Fiction" which itself has the hypernym "Literary-Work". This result in the following:

SubClassOf(Novel Fiction)

SubClassOf(Fiction Literary-Work)

A similar transformation is performed on the properties. For our example, the property *has-novel* will have:

SubObjectPropertyOf(has-novel has-fiction)

SubObjectPropertyOf(has-fiction has-literary-work)

- Synonyms:

  For each class created in the representation, the synonym relations are retrieved from the WordNet semantic network, and transformed into a *equivalentClasses* axioms. For example, the word "Write" has the word "Compose" as a synonym, this relation will be transformed into *EquivalentClasses(Write Compose)*. Similarly, the synonym relations are retrieved for the properties, and transformed into a *equivalentObjectProperties* axioms.

- Complex Classes Definition:

  Complex classes can be defined in OWL using set operations, such union, intersection, and complement. For example, a complex class "Write" can be defined as the intersection of the class "Make" and the property "has-novel". So that if one is "making a novel" it is equivalent to "writing a novel". To create complex classes, we take advantage of properties of classes created from the text. The property of a class created from the text is similar to a differentiae of a definition of that class, where the genus of that class is its hypernym; For example, the concept "Write" is a subclass of "Make" and has the property "has-novel" which can be seen as what differentiate "Write" from "Make". A complex class axiom can then be created to define a class that is equivalent to its parent if that subclass is an intersection between the parent and having a property of that class. So for our example, if the class *Make* is related with a *has-novel* property to the class *Novel*, then this class is equivalent to the class *Write*. In OWL the following axiom is created: *EquivalentClasses(Write ObjectIntersectionOf(ObjectSomeValuesFrom(has-novel Novel) Make))*.

The result is an OWL representation that can be illustrated graphically as in Figure 4 (shown at the beginning of the chapter). This figure shows a graphical illustration of part of the resulting OWL representation for our example with the added WordNet axioms. Classes are shown in ovals, properties as arrows, individuals in rectangle, and equal for equivalent axioms. A complete version of this example representation is shown in Appendix A.

This small representation, that was automatically generated from the sentence *Jurassic Park is a novel written by Michael Crichton and published in 1990*, allows us to infer knowledge that was not specifically mentioned in that text, and to draw several conclusions based on that representation. The following are some of the conclusions that can be inferred from the created representation using a reasoner (see Section 3.3):

The result of all classes that the individual "Michael Crichton" belongs to using a reasoner is:

Michael Crichton is an author.
Michael Crichton is a writer.
Michael Crichton is a communicator.
Michael Crichton is a person.
Michael Crichton is an individual.

....

The result of all classes that the individual "Jurassic Park" belongs to using a reasoner is:

Jurassic Park is a novel.
Jurassic Park is a fiction.
Jurassic Park is a literary work.
....

The result of all range classes filling the role "has-novel" having the domain class "write" related to the writer "Michael Crichton" using a reasoner is:

Michael Crichton (has-writer) wrote (has-novel) Jurassic Park.
Michael Crichton (has-writer) wrote (has-novel) a novel.
Michael Crichton (has-writer) wrote (has-novel) a fiction.
Michael Crichton (has-writer) wrote (has-novel) a literary work.
....

"Write" and "Publish" role fillers (roles omitted for readability):

Michael Crichton published Jurassic Park.
Michael Crichton pen Jurassic Park.
Michael Crichton composed Jurassic Park.
Michael Crichton made Jurassic Park.
Michael Crichton created verbally Jurassic Park.
....

Notice how some inferences (already mentioned as possible hypothesis of this text in Chapter 1), were not mentioned explicitly in the original text, but they were inferred by the reasoner. Of course not all created inferences are necessary entailments from the text, for example, *"Jurassic park was published in twelve months* is one of the created inferences as "twelve months" is an equivalent class to a "year".

3.3 Comparison

In order to be able to infer that a hypothesis is entailed from a text, we need to formulate the hypothesis as a query over the created knowledge representation. The query of knowledge representation formalisms such as ontologies is a central requirement of the Semantic Web. Existing tools that allow users to query and reason over ontologies (e.g. [Haarslev and Möller, 2001, Pellet, FaCT]) use custom designed query languages [Wessel and Molle, 2005] with complex syntax which are reportedly difficult for domain experts to master [Smith, Ceusters, Klagges, Kohler, Kumar, Lomax, Mungall, Neuhaus, Rector, and Rosse, 2005]. The reasoner that we are using is RACER [Haarslev and Möller, 2003]. The RACER system (an acronym for Renamed ABox and Concept Expression Reasoner) is a reasoner that implements tableau calculus for description logic (DL) and supports the web ontology languages DAML+OIL, RDF, and OWL. In this section, we present our approach to querying an ontology in a natural language that uses the semantic restrictions imposed by the ontology design to map terms in the question to the content of the ontology. The hypothesis, formulated in unrestricted natural language, is mapped into the new RACER query language syntax and presented to the description logic automated reasoner RACER which returns the query results. A description of the ideas presented below appears in [Kosseim, Siblini, Baker, and Bergler, 2006].

Semantic-role	Predicate	Argument (Type)
has-book	craft	book(Jurassic Park)
has-crafter	craft	author(Michael Crichton)
in-year	craft	year(1990)

Table 4: Result of the Semantic Analysis for "*Michael Crichton crafted Jurassic Park in 1990.*"

Our approach for knowledge querying consists of creating a query from the hypothesis by mapping the content of the hypothesis into classes and roles of the ontology using the selectional restrictions imposed by the ontology. The highest scoring query is processed using a reasoner over the ontology. Our approach was evaluated on a set of human created question over an OWL ontology, achieving a mean-reciprocal rank (MRR[5]) score of 72%. In order to present and demonstrate the comparison's steps in details, we will use the following hypothesis as a running example throughout the following sections:

Michael Crichton crafted Jurassic Park in 1990.

3.3.1 Query Creation

To transform the hypothesis into a query, we first perform the same semantic analysis step described in Section 3.2. Therefore, the input to query creation is the hypothesis transformed into a set of *semantic-role(Predicate, Argument)* relations. Table 4 shows the resulting set of semantic-role relations for our running example text. Next, we attempt to match each constituent of the structure to variables, classes, individuals or properties in the representation (built from the text - see Section 3.2). This can be seen as a classical categorization problem, in particular, word sense disambiguation. The task here is to find a function to map the linguistic expressions to particular

[5]Mean Reciprocal Rank (MRR) is measured as the average of the multiplicative inverse of the rank for the first correct answer of a query.

senses (classes, individuals or properties in the representation).

To select the correct mapping we were inspired by the selectional restriction-based disambiguation approach used in word-sense disambiguation [Resnik and Yarowsky, 1997]. Indeed, in a text, a semantic role represents the restrictions imposed by a predicate on its arguments which allow one to disambiguate its sense, and in turn, the sense of its arguments. For example, in its transitive form, the verb *drink* imposes that its direct object be a *liquid*. The correct sense of an ambiguous direct object can therefore be identified through this semantic constraint. With an ontology, this same strategy can be used as the roles in the ontology impose constraints on the domain and range of the concepts they can relate. In turn, correctly identifying the concepts or instances involved in the question can help us identify an ambiguous role. To map a semantic role of the hypothesis to a property in the ontology, we compute the semantic distance between a semantic role and a candidate property, using a semantic relatedness measurement. In our implementation we used the WordNet Similarity path length method [Pedersen, Patwardhan, and Michelizzi, 2004a, Patwardhan, Banerjee, and Pedersen, 2003]. The result of this step is that each property is mapped to each semantic role, with a different confidence score based on the semantic relatedness measurement. Once the semantic role is matched to a set of possible properties, we try to match its arguments to a variable, a class, or an individual in the domain and range of this property. This is where semantic restrictions come into play.

If the argument is empty, then a new variable is created as a placeholder, but the domain and range of the property already mapped are kept as constraints on the variable. The Cartesian product of all possible mappings for the semantic role and all possible mappings for the arguments is then computed. The overall confidence score of the final mapping is computed as the product of the individual mappings. For each predicate structure, the result of the semantic analysis is a list of possible

predicate structure	mapping	score
semantic-role: *has-crafter* Predicate: *Craft* Argument: *Author*	property: `has-person` Domain: `Make` Range: `Author`	0.33 $\times$ 0.33 $\times$ 1 0.108
semantic-role: *has-crafter* Predicate: *Craft* Argument: *Author*	property: `has-author` Domain: `Make` Range: `Author`	0.25 $\times$ 0.33 $\times$ 1 0.082
semantic-role: *has-crafter* Predicate: *Craft* Argument: *Author*	property: `has-person` Domain: `Create` Range: `Author`	0.33 $\times$ 0.20 $\times$ 1 0.066

Table 5: Examples of semantic mapping for the predicate structure *has-crafter (Craft, Author)*

properties, classes, and individual in the representation along with a confidence score. Table 5 shows an example, where the predicate structure *has-crafter(Craft, Author)* is mapped to three different ontological structures with different mapping scores. In this table, we can see that the mapping with the highest confidence score of 0.108 for *has-crafter(craft, Author)* is *has-person(Make, Author)*. The confidence score of 0.108 is the product of the WordNet Similarity path length (Sim) measurement as follows: (Sim(has-crafter and has-person)= 0.33) x (Sim(craft and make)= 0.33) x (Sim(author and author)= 1). Once a set of possible mappings is built for each predicate structure of the hypothesis, we need to make sure that variables that should refer to the same entities actually do. This, in effect, allows us to process the predicate structures of a hypothesis as a single semantic unit, rather than a conjunction of unrelated predicate structures. To identify which variables should co-refer to the same entities, we use the semantic constraints we set when we used the semantic relations to bind our variables. If the constraints of two variables can be unified, then we consider the variables to co-refer. For each candidate semantic role relation, the list of the possible mappings is finally ranked according to the overall confidence score and the best mappings are sent to be translated to an ontology querying language.

3.3.2 Querying with a Reasoner

The last step of the ontology querying is responsible for creating the nRQL queries and sending them to the RACER reasoner. Since the establishment of the Ontology Web Language (OWL), design specifications for Description Logic (DL) based query languages have been proposed and existing languages contrasted, highlighting their advantages and limitations [Glimm and Horrocks, 2004]. nRQL emerges as a prominent and highly expressive DL-query language and extends the existing capabilities of RACER with a series of query atoms. nRQL uses a Lisp based syntax; the general structure of a query is composed of a query head e.g. `retrieve(?x)` upon which variables used in the body are projected e.g. `(?x Author)`, where `(retrieve (?x)(?x Author))` queries for instances of the concept `Author`. In our approach, we employ conjunctive queries where the atoms are simple class or property assertions and where the variables in the body of the query match the corresponding individuals in the ontology that satisfy all query conditions. A detailed description of nRQL is given in [Haarslev, Moeller, and Wessel, 2004]. The two main types of nRQL queries that have been considered are: unary class queries and binary property queries. A unary class query tries to find instances of a particular class (e.g. *Find all Authors* ⇒ `(retrieve (?x) (?x Author))`) or to determine if an entity is an individual of a class (e.g. *Is Michael Crichton an Author?* ⇒ `(retrieve () (Michael Crichton Author))`). A unary class query can therefore have one of the two forms:

1. `(retrieve (?x) (?x <class>))`
2. `(retrieve () (<individual> <class>))`

A binary property query searches for the binding between 2 classes or individuals

Predicate Structure Pattern	nRQL query
semantic-role(Predicate, Argument)	`(?x <Predicate>) (?y <Argument>) (?x ?y <semantic-role>)`
semantic-role(Predicate, Argument (named-entity))	`(?x <Predicate>) (?x named-entity <semantic-role>)`
semantic-role(Predicate, ⊘)	`(?x <Predicate>) (?x ?y <semantic-role>)`
...	...

Table 6: Examples of Predicate Structure Patterns and Corresponding nRQL Queries

(e.g. *What has-author what?* ⇒ `(retrieve (?y ?x) (?y ?x has-author))`. A binary role can specify particular classes or individuals instead of specifying a variable. For example, *What happened in-year 1990?* ⇒ `(retrieve (?x) (?x <1990> in-year))`. A binary property query can therefore take the 4 following forms:

3. `(retrieve (?y ?x)(?y ?x <property>))`
4. `(retrieve (?x)(?x <individual> <property>))`
5. `(retrieve (?x)(<individual> ?x <property>))`
6. `(retrieve ()(<individual> <individual> <property>))`

In order to create the nRQL queries from the mappings we found in the query creation step (see Section 3.3.1), we created a set of rules to map the semantic role relations into nRQL queries. Table 6 shows a sample of these patterns and the corresponding nRQL expression.

For our example hypothesis *Michael Crichton crafted Jurassic Park*, the following predicate structures were created by the query creation step (see Section 3.3.1):

has-book(craft, book(Jurassic Park))

has-crafter(craft, author(Michael Crichton))

The corresponding highest ranking ontological matches are:

property:has-novel(class:Make,class:Novel(Jurassic-Park))

property:has-person(class:Make,class:Author(Michael-Crichton))

The above two triplets correspond to the case of [`semantic-role(Predicate, Argument (named-entity))`] shown in Table 6 which create the following nRQL statements:

1. `(?x <Make>))`
2. `(?x Jurassic-Park <has-novel>)`
3. `(?x Michael-Crichton <has-person>)`

Notice how they are related using the variable `x`, which co-refer to the same concept *Make*. Individual nRQL expressions are then connected with an AND operator to create the final nRQL query: `(retrieve (?x) (and (?x <Make>) (?x Jurassic-Park <has-novel>) (?x Michael-Crichton <has-person>)))`

3.4 Decision

Once the text is represented in OWL as explained in Section 3.2, then the hypothesis is transformed into an nRQL query as explained in Section 3.3, and we can then decide whether the text entails the hypothesis or not. The purpose of creating the queries from the hypothesis is interpreting it as a yes/no question, and a returned result for those questions is considered as a hypothesis entailment from the text representation. So the decision making process is rather straight forward: the created OWL text representation is loaded into the reasoner to answer the top generated nRQL hypothesis query. If the reasoner answers the query, then we conclude that the hypothesis is entailed by the text, otherwise it is not entailed. We could have well chosen to test more than only the top nRQL queries, but because we had to limit the overall time of the execution, our current implementation is restricted to the single top most generated nRQL. In our example above, the top generated nRQL query for the hypothesis *Michael Crichton crafted Jurassic Park* is: `(retrieve (?x) (and (?x <Make>) (?x Jurassic-Park <has-novel>) (?x Michael-Crichton <has-person>)))`. When run through the RACER reasoner, it returns the individual "Write-1", which is an individual of the class *Make*, correctly answering the query, and thus this hypothesis is entailed from the created text representation.

3.5 Evaluation and Analysis

We first evaluated the knowledge querying prototype intrinsically on a separate domain (see Section 3.5.1), then evaluated it extrinsically on the task of recognizing textual entailment (see Section 3.5.2). We then performed an analysis of the major issues encountered with this approach (see Section 3.5.3). The evaluations and analysis are described next in details.

3.5.1 Evaluation of the Knowledge Querying Approach

The prototype natural language interface was intrinsically tested on a separate domain: the FungalWeb Ontology [Shaban-Nejad, Baker, Butler, and Haarslev, 2004]. The FungalWeb Ontology is a prototype bio-ontology, scripted in the OWL formalism. It is an integrated conceptualization of multiple scientific domains. These overlapping domains include taxonomies of fungi and enzyme reaction mechanisms as well as enzyme substrates and industrial specifications describing the applications and benefits of enzymes. The FungalWeb Ontology contains 3616 concepts and 11,163 instances related by 142 roles. The conceptualization was designed so that fungal species, enzyme names, enzyme product names, enzyme vendor names and chemical names are modeled as instances. Free text segments describing enzyme applications, industrial benefits of enzymes were also modeled as instances. The scope of the ontology has been further illustrated in a series of application scenarios [Shaban-Nejad, Baker, Haarslev, and Butler, 2005, Baker, Shaban-Nejad, Xu, Haarslev, and Butler, 2006] demonstrating the range of query capabilities afforded by the conceptualization.

To test the knowledge querying prototype, we used a corpus of 180 pairs of hypothesis and their associated nRQL queries. The material was created by 4 different casual users in order not to be influenced by the writing style of one particular person. The users were all knowledgeable in the domain and the content of the ontology, but did not necessarily know its structure and property names. We used 80% randomly selected pairs for developing the query generation and the remaining 20% were used for the evaluation. We compared the prototype-generated results with the human composed queries as gold-standard. The comparison was based on query equivalence. If the generated query was not exactly the same as the one in the gold-standard, it was considered wrong. For each hypothesis, we generate a set of possible queries ranked in order of confidence. For each hypothesis h, we therefore computed the final score as the reciprocal rank of their first correct answer. If none of the generated queries was equivalent to the gold-standard, a score of 0 was given. Otherwise, the score is equal to the reciprocal of its rank. For example, if a hypothesis generated 4 ranked queries, and the 3^{rd} one is correct, the question received a score of $\frac{1}{3}$. The overall score is the average $RR(q)$ for all questions q. This methodology is called the mean-reciprocal rank (MRR) score as used in question-answering [Voorhees, 2001], and the resulting MRR on the test set is 0.72, which is in line with the current accuracy of other ontology based question answering approaches [Lopez, Motta, Uren, and Sabou, 2007]. Details of this work are described in [Kosseim, Siblini, Baker, and Bergler, 2006]. An extrinsic evaluation will be described next on the task of recognizing textual entailment.

3.5.2 Evaluation of the Recognizing Textual Entailment Approach

For the development of the recognizing textual entailment prototype, we have used the dataset of the RTE-1 and RTE-2 Recognizing Textual Entailment competitions. Recall from Section 2.3 that the dataset of Text-Hypothesis (T-H) pairs for the first challenge (RTE-1) [Dagan, Glickman, and Magnini, 2005] consists of 1367 T-H pairs, the second challenge (RTE-2) consists of 1600 pairs [Bar Haim, Dagan, Dolan, Ferro, Giampiccolo, Magnini, and Szpektor, 2006] totalling 2967 T-H pairs. To evaluate the performance of the system we performed the test on the RTE-3 Recognizing Textual Entailment third challenge (RTE-3) test data. As described in Chapter 2, the corpus consists of 800 Text-Hypothesis (T-H) pairs of text snippets and manually labeled for entailment by human annotators. Twenty six groups participated in the RTE-3 challenge. Overall, the systems achieved an average accuracy of 61%, and the best accuracy of 80%. Our baseline system performed below the average with an accuracy of 49% on the two-way task (entailment, no entailment). As specified in the beginning of this chapter, this approach constitutes our baseline system for a description logic approach to recognize textual entailment, which will be improved in Chapters 4 and 5.

3.5.3 Analysis

When analysing the performance of our approach we realized that two major issues could be improved:

- The first major issue is related to the comparison component and the decision component. 60% of all the created nRQL were incorrectly mapped, which lead to an incorrect result from the reasoner. The reasons range from missing information, to the optimistic setting in our use of the semantic relatedness measure which always resulted in a mapping. This feature is quite acceptable in a question answering setting as any result is better than no result, but not at all acceptable in recognizing textual entailment as a result means an entailment, which is not always the case. To illustrate this, our approach incorrectly entailed the hypothesis *Michael Crichton copied the fiction Jurassic Park*, because the path length semantic relatedness distance between the pair "Copy" and "Write" is similar to the semantic relatedness distance between "Create" and "Write". So the incorrectly created nRQL from this hypothesis is :

  1. `(?x <Write>))`
  2. `(?x Jurassic-Park <has-novel>)`
  3. `(?x Michael-Crichton <has-person>)`

  Our intuition at this stage is to transform the semantic relatedness information into axioms which is needed to facilitate the comparison and give those axioms to a machine learning algorithm to decide on the inferred information if the texts entail the hypothesis or not. This will be described in Chapter 4.

- The second issue is related to the named entities information. Over 30% of the named entities were not found in the Freebase database; an alternative is needed. One possibility is to explore the web for additional knowledge related to named entities. The intuition is that the Web could be considered as the largest corpus, and will contain information that may otherwise not be available in handmade lexicons. This will be discussed in details in Chapter 4.

In the next chapter, we will describe how the lessons learned from our first experience at the RTE challenge where used to develop a better performing approach.

3.6 Conclusion

In this chapter, we have presented our first approach at recognizing textual entailment. This approach has investigated the use of description logic as a surface meaning representation of text, knowledge querying in natural language, and semantic relatedness to recognize textual entailment. The implementation of this approach has resulted in a low accuracy of 49% in the RTE3 challenge. An analysis of the results helped us identify the problems of this shallow logical approach in recognizing textual entailment. The major issues encountered are either related to missing implicit information, or a weakness in the semantic relatedness comparison scheme.

In the following chapter we will present our knowledge alignment approach for recognizing textual entailment. This approach takes into consideration the limitations and issues discussed in Section 3.5 and improves upon them. It specifically investigates the use of the Web as a corpus for enriching a meaning representation of a text, and the use of semantic relatedness between concepts to learn axioms on demand for the purpose of aligning textual representations and as an indicator of textual entailment.

Chapter 4

Knowledge Alignment Approach

In Chapter 3 we have introduced a query based logical inference approach for recognizing textual entailment. An analysis of its evaluation helped us identify some of the main problems faced by a shallow logical approach. These are either related to missing implicit information, or a weakness in the semantic relatedness comparison scheme. In this chapter we will address both of these issues by first investigating the use of the Web as a corpus for enriching a meaning representation (Section 4.2), and the use of semantic relatedness between concepts to learn axioms on demand (Section 4.3), as an alternative to a predefined set of axioms, and as an indicator to textual entailment (Section 4.4).

In this chapter, we propose a description logic and semantic relatedness approach to textual entailment, where the type of semantic relatedness axioms employed in aligning the description logic representations are used as indicators of textual entailment. This approach is based on the automatic acquisition of a representation from the text T, another representation from the hypothesis H, and the alignment of the acquired representations. The automatic acquisition of representations includes a new component that uses the Web as a corpus and machine learning. The textual entailment problem is therefore reduced to a classification one based on the resulting aligned representation. Transforming the problem into classification one should make the approach more flexible than a hard logical inference one.

In the next section we will first introduce the architecture of our approach and the representation acquisition module of the system. We will then describe the main differences of this module compared to our previous approach presented in Chapter 3. The rest of the sections will introduce the representation alignment in more details as it is the major difference with our previous approach, and an evaluation of the overall approach. A description of this approach has appeared in [Siblini and Kosseim, 2008a] and [Siblini and Kosseim, 2009].

4.1 Overview of the Knowledge Alignment Approach

As with our baseline approach (Chapter 3), our knowledge alignment approach can be divided into three main components: a representation component, a comparison component, and a decision component. The representation component follows the same method as the previous approach (see Section 3.2) but this time we represent both the text and the hypothesis in description logic. In addition, instead of using a database to find named entities types, a new approach is used that uses the Web as a corpus to do so. Then, the comparison component compares and aligns the two created representations. Finally, the decision component classifies the information collected in the representation alignment into an entailment result. Figure 8 shows an overview of this approach. The different components are shown in dashed boxes, the inputs and outputs in ovals, and the sub-components in rectangles. The differences between this approach and the baseline one from Chapter 3 are highlighted in bold and with a light grey background. Note that Figure 8 differs from Figure 1 in

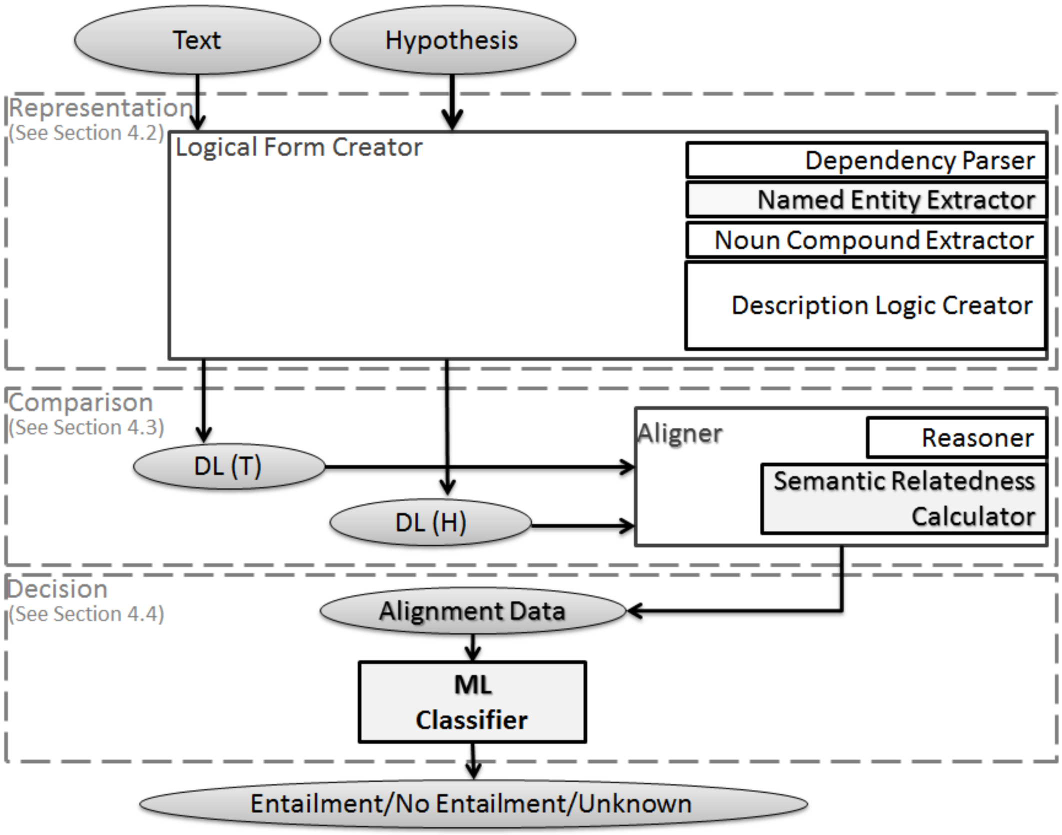

Figure 8: Knowledge Alignment Approach

the noun compound extractor component, the semantic relatedness calculator component, and the machine learning classifier.

The representation component involves the representation of the text in description logic to facilitate the comparison and decision making (described in Section 4.2). This component comprises a logical form creator that transforms both a text and a hypothesis into their own logical forms. In this component we will be introducing a new approach for using the Web as a corpus to extract types of named entities. This novelty has been introduced to deal with the high number of unrecognized named entities of our baseline approach (see Section 3.5).

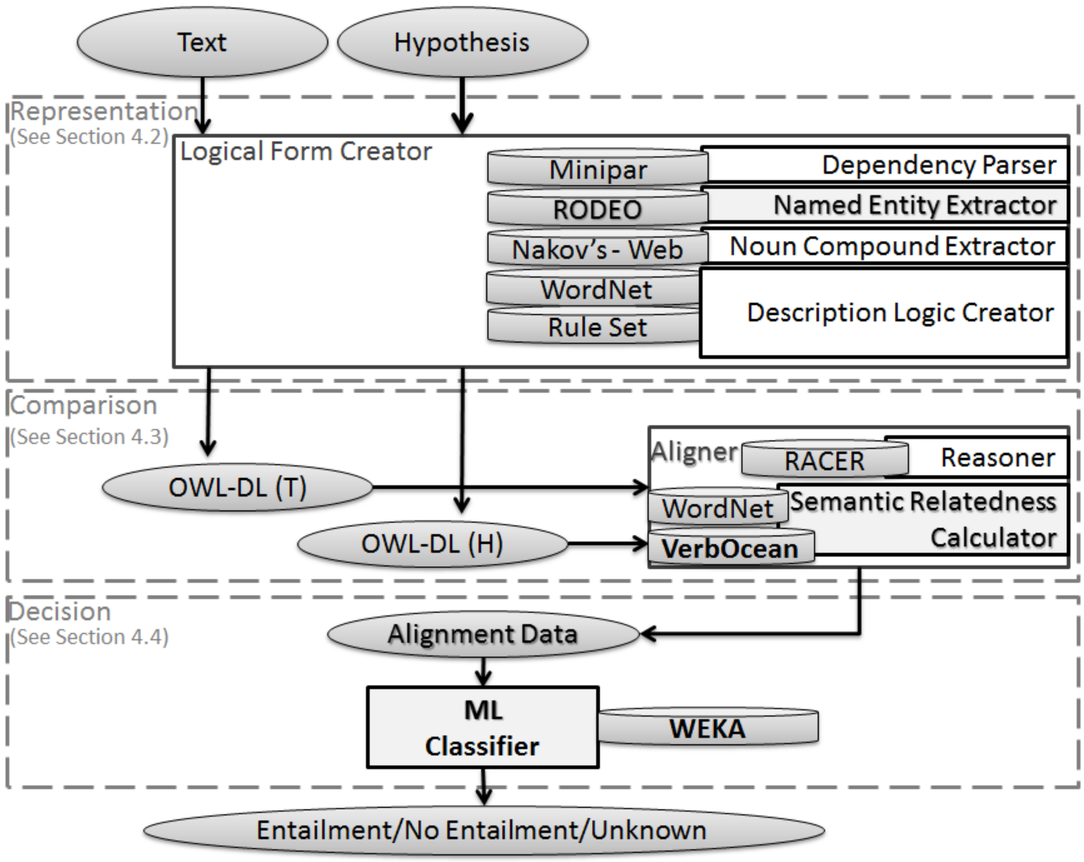

Figure 9: Prototype of the Knowledge Alignment Approach

The comparison component will then compare and align the representation of the text with the representation of the hypothesis, and collect alignment information that will be used in the decision making (described in Section 4.3). This is the major difference with our previous approach. It has been introduced to deal with the first major issue of that approach of incorrectly mapping an nRQL query from the hypothesis (see Section 3.5). The comparison relies on a reasoner and a semantic relatedness calculator as a substitute for additional knowledge to compare the representation with the hypothesis. The alignment process learns new axioms on demand, based on the semantic relatedness calculator. The purpose of this comparison is to collect alignment information that will be used in the decision making.

Finally, the decision component will use the alignment data to decide whether the text entails the hypothesis or not (described in Section 4.4). The approach relies on machine learning that uses the alignment data as features to decide on a textual entailment result.

In order to evaluate this approach, we have created a prototype system. Figure 9 provides an overview of the system architecture, specifically showing the use of specific resources in the implementation decisions. As in the previous figures, the main components are shown in dashed boxes, the inputs and outputs are shown in ovals, the sub-components are shown in rectangles, and the sub-component's resources in cylinders. As shown in Figure 9, for our implementation, we have chosen the Minipar parser [Lin, 1998c] as a dependency parser, the RODEO system which is our own implementation of a tool that uses the Web as a corpus [Siblini and Kosseim, 2008b] as the source of named entity types (see Section 4.2.2), Nakov's approach [Nakov and Hearst, 2006] for extracting noun compound types, and the WordNet [Fellbaum, 1998] semantic network with a set of manually created transformation rules for the logical form creator module. In addition, we have used the RACER reasoner [Haarslev and Möller, 2003] and VerbOcean lexical patterns [Chklovski and Pantel, 2004] in addition to WordNet [Fellbaum, 1998] to perform the semantic alignment of classes and properties. Finally, the alignment data produced in the comparison component are used as features in several machine learning algorithms using the WEKA tool [Dimov, Feld, Kipp, Ndiaye, and Heckmann, 2007] for the decision making. In the following sections, we will go over each component in details, starting with the knowledge representation component.

4.2 Representation

The representation procedure follows the same approach described in Section 3.2. The logical form creator is the process of automatically creating an OWL-DL representation. The process follows a syntax-driven pipeline architecture, which is augmented by knowledge on demand. A text is first passed through a syntactic analysis to derive its dependency parse. This dependency parse is then passed as input to a semantic analyser to produce a literal meaning representation. Finally, this meaning representation is passed as input to an ontological analyser that produces a formal logical-based meaning representation in OWL-DL.

The only differences with our baseline approach are the following:

1. The representation is created for both the text and the hypothesis. This is one of the differences with our baseline approach, and it has been introduced to deal with the first major issue of incorrectly mapping an nRQL query from the hypothesis (see Section 3.5).

2. The use of the Web as a corpus to recognize the semantic types of named entities. This method has been used in the logical form creator semantic analysis step to replace the use of the Freebase database for named entity recognition. This is needed to deal with the large number of named entities that were not found in the Freebase database.

Let us describe our new approach for named entity recognition using the web.

4.2.1 Named Entity Recognition Using the Web

Named Entity Recognition (NER), as described by the Message Understanding Conferences (MUC)-7 [Chinchor, 1998], is the task of identifying and classifying entities that are considered to belong to one of the following semantic classes: person, location, organization, temporal entities and numeric quantities. Different approaches have been introduced to deal with NER, however two approaches are mainly adopted. The first uses resources, such as gazetteers, and hand crafted rules to match expressions to the resources, and the other uses supervised machine learning techniques on annotated corpora in order to learn a set of patterns or to discriminate features such as the work of [Bikel, Schwartz, and Weischedel, 1999]. One of the major issues encountered when we analysed our RTE baseline approach (see Section 3.5) is related to the named entities information. Over 30% of the named entities were not found in the Freebase lexicon; and an alternative is needed.

In this section, we explore using the web for additional knowledge related to named entities. Our aim is to automatically extract more fine-grained class(es) of named entities compared to the general classes described in MUC conferences. For example, we need the ability to extract the classes that *"Paul Krugman"* belongs to; in this case a general class would be a "person", but a more specific one would be a "columnist". To accommodate a larger coverage in selecting fine-grained classes of named entities, many researchers have used the Web. Most of the techniques used rely on a set of patterns, and the main difference between one technique and the other is usually the type of patterns used. Some use text patterns such as the work of [Etzioni, Cafarella, Downey, Popescu, Shaked, Soderland, Weld, and Yates, 2005], while others uses wrappers or HTML patterns such as the work of [Nadeau, Turney, and Matwin, 2006]. On the other hand, [Etzioni, Cafarella, Downey, Popescu, Shaked, Soderland, Weld, and Yates, 2005]'s KNOWITALL system aims to automate the extraction of instances of classes such as the names of scientist from the Web by using a set of text patterns. We have built upon [Etzioni, Cafarella, Downey, Popescu, Shaked, Soderland, Weld, and Yates, 2005]'s work; however, instead of using a set of text patterns over the Web, we have used grammatical patterns, and instead of extracting named entities that belong to a class (for example, an example named entity that belongs to the *columnist* class is *Paul Krugman*), we extract classes of a named entity (for example, the *Paul Krugman* belong to the classes *economist, columnist...*). It is necessary in our work to be able to categorize the types of each named entity mentioned in a text.

Crawling the entire World Wide Web and annotating it with grammatical information would be an ideal solution for our needs. However, as this is not feasible, we needed to find an alternative. Web search engines provide an unstructured query language to search the Web, but using the content words of a grammatical query may return millions of documents, and not all the returned documents will satisfy the grammatical query we are looking for. Therefore, the main challenge is to be able to narrow down the returned results to a manageable and relevant subset. To narrow down the returned results, our approach is to create specific search engine queries that relies on the non-content words selections preferences of the grammatical query. To test this approach, a tool called RoDEO has been implemented and included in our Knowledge Alignment RTE approach to extract named entities from the web.
The following are details of the RoDEO approach (more information can be found in [Siblini and Kosseim, 2008b]).:

Grammatical Query to Web Queries: The first part of this approach is to transform a grammatical query to a set of specific Web queries. In order to do so, we have developed the following method to generate sentences from a grammatical query. We have used the open American English Corpus [Ide and Macleod, 2001] and parsed with the Minipar dependency parser [Lin, 1998c], in order to collect a large set of parsed sentences. This allowed us to create a corpus of the most frequent non-content word preferences for each type of dependency grammatical relation. The corpus is used to translate grammatical queries into web queries which are made of specific search phrases. A search phrase is a sequence of words that must co-occur. For example, a *subject* relation's most frequent non-content word preferences includes the following search phrases:

a NOUN(Subject) can VERB a

the NOUN(Subject) that VERB the

a NOUN(Subject) who VERB a

a NOUN(Subject) to VERB

VERB by the NOUN(Subject)

So if we are searching for the possible subjects of verb *drive*, one possible search phrase will be "*a **[1] *can drive*". In this example, the subject has a preference to be preceded with a non-content word "a" and followed with "can", which is followed by a verb.

Web Query Results to Grammatical Query Results: Each Web query generated is then run through a web search engine, and the relevant sentence snippets are then grammatically parsed. The parse that match the original grammatical query is then returned. For our example of finding the possible subjects of the verb *drive*, the created search phrase "*a * can drive a*" returns the following relevant sentences, among others:

at 16 years old a person can drive a 3-wheeler...

a person can drive a bus with a seating...

a driver can drive a minibus...

a woman can drive a car...

a man can drive a motorboat...

Parsing those sentences and finding the subjects of *drive* returns the following:

[1]The star is a placeholder for one or more unknown terms.

person, driver, woman, man...

For the task of extracting classes of named entities from the web, we created a set of specific grammatical patterns[2] and used RoDEO to query the web for those patterns.

One of the grammatical patterns that we will introduce here is the **predicative nominative pattern**: Named-Entity (subj) V-BE (predicate) Noun. Grammatically, the predicate nominative is the noun that follows a form of the verb *to be*, like in the sentence: *"Margaret Thatcher was the Prime Minister"*. In this example *"Margaret Thatcher"* is the subject of verb *to be* and *"Prime Minister"* is its predicate noun. As a result, it would be an appropriate pattern to extract classes of named entities.

Another pattern is the **appositive pattern**: Named-Entity (appositive) Noun. Appositive is noun or noun phrase that describes another noun phrase that is placed besides it, as in the following example:

"Robert Fisk, a journalist, said ...". The noun *"Journalist"* is an appositive to *"Robert Fisk"*.

We also have derived other grammatical patterns based on lexico-syntactic patterns available in the literature (e.g. [Hearst, 1992, Pearson, 1998, Saggion, 2004], such as:

1. NP[3] such as NP, (or/and) NP.

   Example: *Columnist, such as Paul Krugman.*

2. Such NP as NP, (or/and) NP.

   Example: *Work by such columnist as Paul Krugman, and Paul Romer.*

3. NP, or other NP.

   Example: *Paul Krugman, or any other columnist in the N.Y. Times.*

[2]The grammatical patterns are expressed as a set of dependency relations.

[3]NP stands for noun phrase.

4. NP, and other NP.

   Example: *Read Paul Krugman and other economists and healthcare experts....*

The first pattern, for example, is translated to the following grammatical query: Noun (modified) such-as (Pcomp-n) Named-Entity, where Pcomp-n stands for a nominal complement of a preposition, in this case a named entity complement of the preposition *such-as* that is modifying a noun.

Using RoDEO, a grammatical pattern will return a list of nouns that conforms to the selected grammatical relations from the Web for a selected named entity. For each pattern, a set of web queries are created. For example, the following queries are automatically generated by RoDEO for pattern 1:

*" a * such as the * or the *"*
*" when * such as * and *"*
*" the * such as * and *"*
*" and * such as * and *"*
*" from * such as * and *"*
*" from a * such as * who"*
*" from a * such as * of"*
*" * such as * he would"*
*" * such as * here at"*
*" * such as * not an"*
*" * such as * in the"*
*" * such as * should"*
*" a * such as * and *"*
*" a * such as * or *"*
*" a * such as the *"*
*" * such as * can"*
...

For each pattern, we replace the named entity with the one that we are looking to classify, and run the result through RoDEO, which will return a set of nouns corresponding to possible classes that a named entity can belong to.

This mechanism of extracting named entity types is used as part of the semantic analysis phase in representing the text and the hypothesis in our RTE approach. It is used as an alternative to the Freebase database, which was causing a major issue in recognizing textual entailment (as mentioned in Section 3.6) and the rest of the representation component was kept as the baseline approach introduced in Section 3.2.

4.2.2 Evaluation of the Named Entity Recognizer

To evaluate the approach described in the previous section, we used a set of 1019 named entities extracted from the shared online FreeBase database [Bollacker, Cook, and Tufts, 2007]. FreeBase contains named entities with their general and specific types. For example, according to FreeBase, the named entity: *"Al Franken"* belongs to the following types *"Person, author, writer, and actor"*. The evaluation scoring has been done by comparing our extracted types to the FreeBase types. As RoDEO returns classes that not exactly match the FreeBase types, we used the WordNet::Similarity [Pedersen, Patwardhan, and Michelizzi, 2004b] Path Length method to compare the types. The path length method is a simple node-counting scheme, which returns a relatedness score between two concepts. The score is inversely proportional to the number of nodes along the shortest path between the synsets in WordNet. The shortest possible path occurs when the two synsets are the same, in which case the length is 1. If the compared types had a relatedness score that was over a threshold t, (t=0.21 in our setting), we considered it as correct. The threshold has been selected after manually comparing a set of 50 classes. For example, if the returned class is an *"Actor"* for a named entity, and its FreeBase corresponding type is an *"Artist"*, the WordNet::Similarity Path Length method returns a relatedness of 0.25 for the two concepts. As such we assume that the returned class is correct. We have evaluated a total of 1019 named entities chosen at random from FreeBase. The total number of different FreeBase types that these entities belong to is 69 types. The total number of classes returned by our approach for the 1019 named entity is 678 types. That shows that our approach is returning far more fine grained results than the FreeBase types, which only returned 69 types. For example, the *"Athlete"* FreeBase type has been matched to *"Blocker, bowler, boxer, cornerback, cricketer, footballer, keeper, receiver, scorer, skater, swimmer, tackle..."*. We want fine grained types for the task of recognizing textual entailment.

To compute the accuracy of the extracted classes of a single named entity we use the following metric:

$$\text{Accuracy} = \frac{\text{Number of correct types}}{\text{Total number of types}}$$

The accuracy of RoDEO is computed as the average of the accuracy for all the evaluated named entities.

Overall, RoDEO achieved an accuracy of 0.7. Table 7 shows some of the accuracy results grouped by types and sub-types. For example, for the high level *"Person"* type, the accuracy achieved is up to 0.87, whereas the type *"Company"* achieved an accuracy of 0.62. The person type can be subdivided into several subtypes, for example the *"Actor"* type which achieved an accuracy of 0.78.

There are many related work throughout the named entity recognition and classification field; however most of the available work falls under the initial task set at the MUC conference for identifying and classifying named entities into five very broad classes, which is much easier than classifying named entities into more fine grained

Types	Accuracy	Subtypes	Accuracy
Person	0.87	...	...
		Actor	0.78
		Athlete	0.76
		Author	0.75
		...	...
		Publisher	0.14
		...	...
Company	0.62	...	...
		Airline	0.66
		Employer	0.34
		Owner	0.31
		...	...
		Chain	0.16
		...	...

Table 7: Sample of the Evaluation Results

classes. Most methods that classify named entities into five classes achieve an accuracy well above 90%. However, this has not been the case when classifying named entities into more fine grained classes. As such, we will focus our comparison to some of the approaches that classify named entities into more than five classes. Table 8 shows a comparison of some of these approaches ordered by the number of classes they consider. [Cimiano and Staab, 2004]'s PANKOW system is a lexico-syntactic pattern based system that uses the web frequency to select the appropriate class from a set of 59 classes. The PANKOW system achieved an accuracy of 24.9%. [Nadeau, 2007]'s BaLIE system uses semi-supervised machine learning and the web to classify named entities into 100 classes. It achieved an accuracy of 57.4%. BaLIE creates large gazetteers of named entities, using a hand crafted HTML markup in web pages and a seed of named entities, and then uses a simple heuristic to identify and classify named entities. [Sekine, 2004]'s system achieved 72% by classifying named entities into 200 classes, however they used about 1,400 handcrafted rules and a dictionary of

130,000 instances that are classified into the 200 classes. Another interesting system is [Alfonseca and Manandhar, 2002]'s system that adopted a vector space model having syntactic dependencies as vector features, and compared the named entity vector into the most similar vector. They considered 1200 classes and achieved an accuracy of 17.39% using the verb/object dependencies as a feature. Although we are extracting

Systems	Types	Accuracy
PANKOW	59	24%
BaLIE	100	57%
Sekine's tagger	200	72%
RoDEO	678	70%
Alfonseca's system	1200	17%

Table 8: Comparison of Various NE Recognition systems

a large number of fine grained classes, we are not classifying the named entities into these sets of classes, but rather extracting the most frequent classes associated with each named entity. As Table 8 shows, RoDEO's accuracy is comparable to the systems using hand crafted rules, even though we are extracting a much larger number of classes.

4.2.3 Named Entity Recognizer and RTE

Let us now return to the question of RTE and see how RoDEO, our new approach to NER, can improve our baseline RTE system. Recall that in the representation component of our baseline approach (see Section 3.2), and more specifically in the semantic analysis phase, we try to find the classes that individuals belong in order to help in adding implicit information into our representation. If the class of a named entity is not present in the text, we use the FreeBase to search for it. However, in our new approach we used the Web as a corpus to recognize the semantic types of named entities, and specifically the set of grammatical patterns using RoDEO as described in the previous section. This is needed to deal with the problem of the large number of named entities that were not found in the Freebase database in the evaluation of our baseline RTE approach (as described in Section 3.5). In addition to the NER changes in the representation component of the RTE approach the next section will go over the changes of the comparison component.

4.3 Comparison

Once the representation component of our RTE approach creates a representation for the text (representation-T) and a representation for the hypothesis (representation-H), the comparison component then aligns the two representations into a single representation (representation-A), that will be the basis of an entailment decision. The importance of this component is in its ability to learn semantic relatedness knowledge specifically related to the represented content and transforming this additional knowledge into axioms, as an alternative to a large predefined set of axioms. Those learned axioms, in addition to the aligned representation-A, will be the basis for the decision component to make an entailment decision.

The following highlights the main steps followed to compare the two representations and to learn content related axioms:

1. The first step is to find semantic relations between concepts. This step is performed using our own implementation of the S-Match algorithm [Giunchiglia, Shvaiko, and Yatskevich, 2004]. The S-match algorithm produces a semantic mapping between two graph-like structures using the WordNet semantic network. The algorithm produces mappings among the nodes that correspond semantically to each other, through an equivalence (using WordNet Synonyms), a

generality (using WordNet Hypernyms), or disjoint (using WordNet Antonyms) relations. A matrix is produced of all concepts of the two representations, and the strongest relations holding between concepts of nodes is selected. In case we are not able to match concepts or properties using WordNet, we use the VerbOcean web lexical patterns [Chklovski and Pantel, 2004] to add semantic relations on demand. Whereas WordNet semantic relations were created by hand, VerbOcean detect fine-grained semantic relations using lexico-syntactic patterns over the web, with an average accuracy of 65.5%. For this reason we give the VerbOcean detected relations a lower confidence.

2. If a relation is found, we translate it into one or more axiom as follows:

Equivalent : WordNet based similarities are translated to an equivalentClasses(X Y) or equivalentObjectProperties(X Y) axiom in the representation (depending if they are classes matching other classes or properties matching other properties).

Possibly Equivalent : Similarly, VerbOcean based similarity are translated to an equivalentClasses(X Y) or equivalentObjectProperties(X Y) axiom in the representation, but those axioms will be annotated with a "Possibly Equivalent" label to differentiate them with the WordNet related ones and will be used later in the decision phase.

Opposite : Semantically related relations that are found through an antonym WordNet relation or a VerbOcean antonym pattern are also translated to an equivalentClasses(X Y) or equivalentObjectProperties(X Y) axiom, but are annotated with an "Opposite" label (which will be used later in the decision phase).

This step is what we refer to as learning axioms on demand. We do not have to rely on a predefined set of hand-coded axioms, instead we use the semantic relation information from a lexicon or using the Web for specific concepts or properties and we translate them into axioms.

Our hypothesis is that the created axioms through this comparison phase will be good indicators of textual entailment between the two representations.

In order to illustrate the 2 steps above, let us take the following example:

(T): *Jurassic Park is a novel written by Michael Crichton.*

(H): *Michael Crichton crafted the book Jurassic Park.*

Figure 10 shows a graphical illustration of the resulting OWL representation from the example text. The figure shows the classes in ovals, the properties in arrows, and the individuals in rectangles. Similarly, Figure 11 shows a graphical illustration of the resulting OWL representation for the example hypothesis. Recall that these representations have been created using the approach described in Section 4.2. In this example, we can notice from the two graphs that these representations share several common concepts and properties. Only the three classes: *Book*, *Craft*, and *Crafter* which are available in representation-H, are missing from representation-T. We first try to semantically match those concepts with all the concepts in representation-T, by calculating their semantic relatedness using WordNet, or if no relation exist with WordNet then we try the VerbOcean method. *Book* for example has the strongest semantic match with *Novel* through a VerbOcean semantic match. *Crafter* has a hypernym relation with *Person*, and *Craft* has a strongest match as a WordNet hypernym with *Make*.

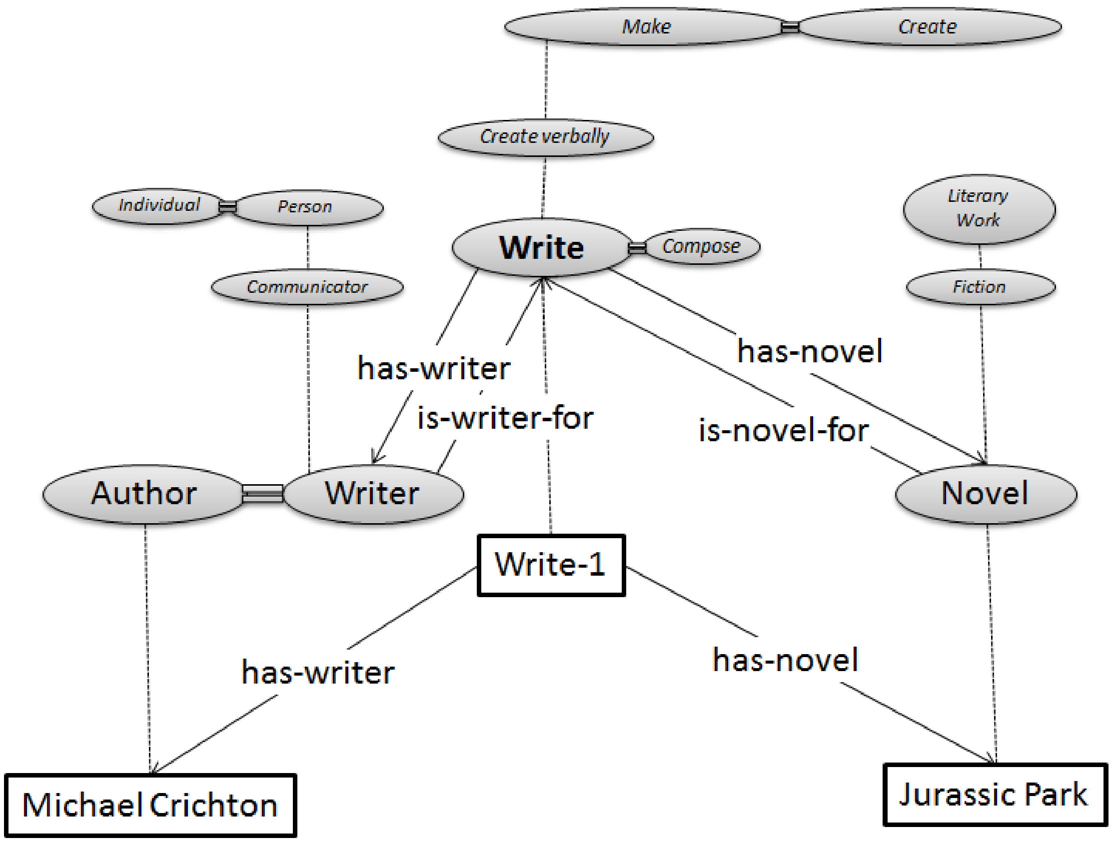

Figure 10: A Graphical Illustration of representation-T Created for the Hypothesis "*Jurassic Park is a novel written by Michael Crichton*"

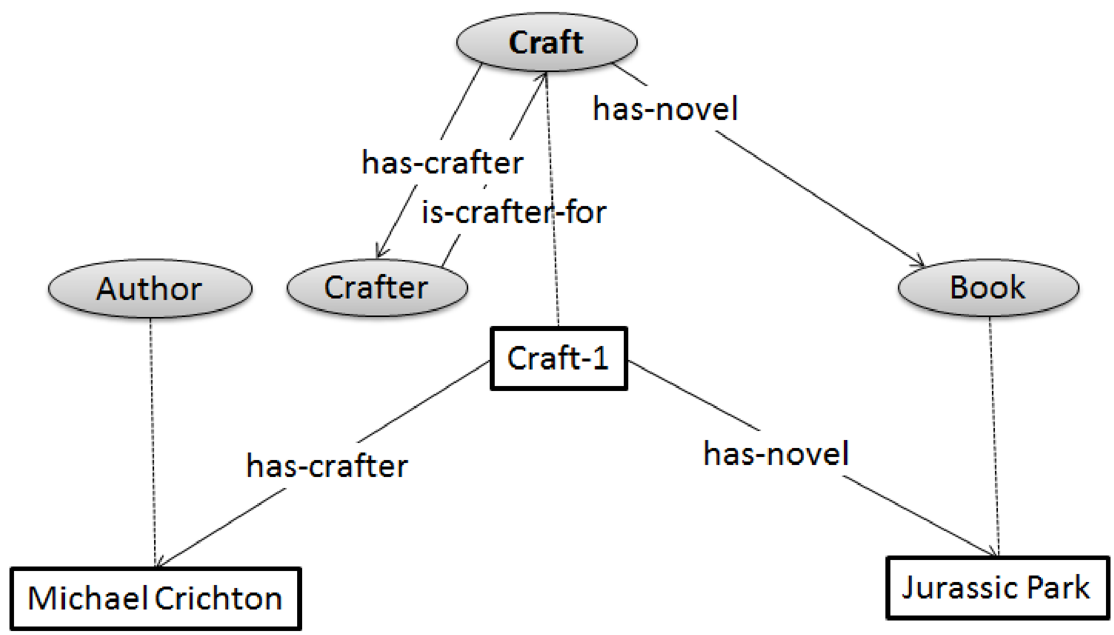

Figure 11: A Graphical Illustration of representation-H Created for the Hypothesis "*Michael Crichton crafted the book Jurassic Park*"

Figure 12 shows a graphical representation of the aligned representation example, where the equality sign represents semantic equivalence (retrieved through the WordNet method), the approximate sign represent a possible semantic relatedness relation (retrieved using the VerbOcean method), and the inequality sign represent complement classes (through an antonym path or pattern). Note that in this specific example, all the classes and properties have been aligned in the resulted representation, but this may not always be the case. The type of axioms created in this alignment step is the basis of our hypothesis for recognizing textual entailment, where we take the resulted alignment axioms as features for classifying textual entailment. In this case the class *Author* is already available in representation-T and is equivalent to the class *Writer*, *Book* is possibly equivalent to the *Novel* class, *Crafter* is possibly equivalent to the *Person* class, and *Craft* is possibly equivalent to the *Make* class. Properties are also matched in a similar fashion (not displayed in the graph due to lack of space),

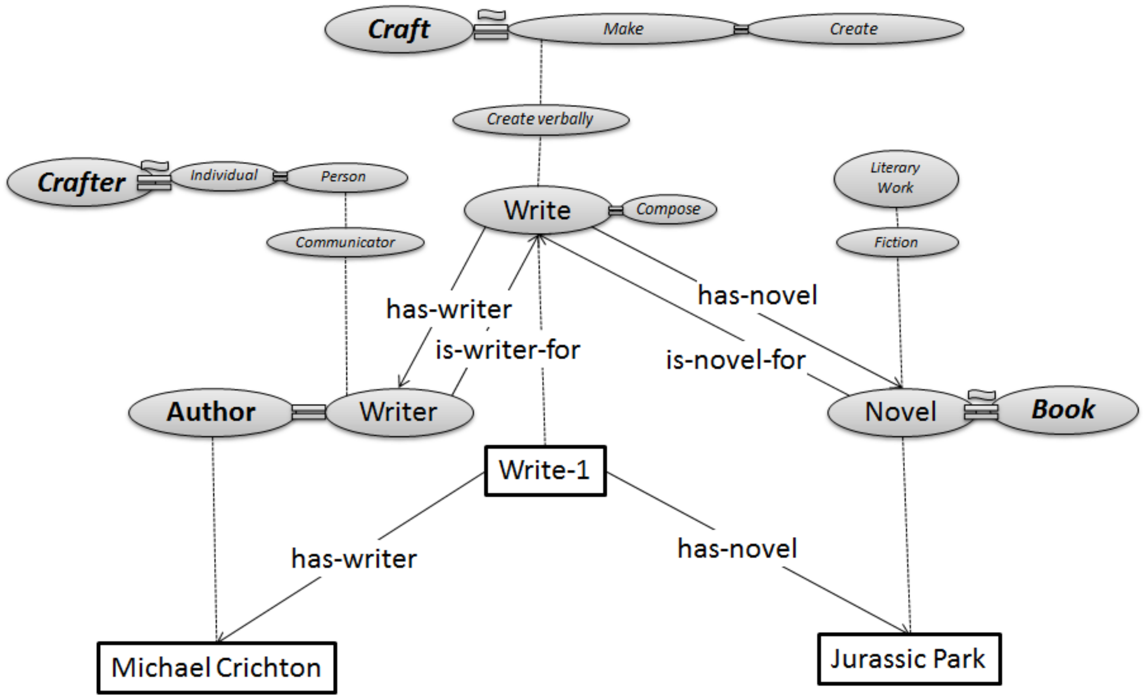

Figure 12: A Graphical Illustration of an Alignment representation-A

has-crafter has a possible equivalent relation to *has-person*, and *has-novel* has a possible equivalent to *has-book*. Reasoning over the aligned representation-A, taking into consideration the type of axioms used in the alignment of this step, will enable the decision component to decide whether the text entails the hypothesis or not.

4.4 Decision

Our hypothesis for recognizing textual entailment is that the type of logical statements used to align textual representations can be used as an indicator of textual entailment. In particular, if a high proportion of classes and properties between the two created representations are shared then most probably we have an entailment. To test this hypothesis, we turned to supervised machine learning. We created a set of features based on this hypothesis to train a machine learning algorithm to classify textual entailment when trained on an RTE training set. If those features can be used successfully in a machine learning algorithm to predict textual entailment, then we can conclude that the type of logical statements used to align textual representations can be used as an indicator of textual entailment.

The features that we used include:

F1: Available Classes This feature represents the percentage of the classes in rep resentation-H that are also present in representation-T. For example, the class *Author* is already available in both representations.

F2: Available Properties This feature represents the percentage of the properties in representation-H that are also present in representation-T. For example, the property *has-novel* is already available in both representations.

F3: Available Sub-Classes The percentage of subclass relationships between classes in representation-H that are present in representation-A. For example, the class *Novel* is a subclass of the class *Fiction*.

F4: Equivalent Classes The percentage of equivalent classes present in represe ntation-A and representation-H; where an equivalent class is one having an *equivalentClasses* axiom in the representation. For example, the class *Author* is equivalent to the class *Writer*.

F5: Possible Equivalent Classes The percentage of possibly equivalent classes present in representation-A and representation-H; where a possibly equivalent class is one having an *equivalentClasses* axiom in the representation but has been annotated with a *Possibly Equivalent* annotation by the comparison phase (Section 4.3). For example, the class *Craft* which is possibly equivalent to the class *Create*.

F6: Equivalent Properties The percentage of equivalent properties present in rep resentation-A and representation-H; where an equivalent property is one having an *equivalentProperties* axioms in the representation. For example, the property *has-writer* is equivalent to the property *has-author*.

F7: Possible Equivalent Properties The percentage of possible equivalent properties present in representation-A and representation-H; where a possibly equivalent property is one having an *equivalentProperties* axiom in the representation but has been annotated with a *Possibly Equivalent* annotation by the comparison phase (Section 4.3). For example, the property *has-novel* is possibly equivalent to the property *has-book*.

F8: Disjoint Classes The percentage of disjoint classes present in representation-A and representation-H. The term disjoint classes in OWL refers to two classes that do not have members in common; However, in our work we use the term to refer to two classes that represent content words that are antonyms, these disjoint classes are the ones that have an *equivalentClasses* axiom but have been annotated with an *Opposite* annotation by the comparison phase (Section 4.3).

F9: Possible Disjoint Classes The percentage of possible disjoint classes present in representation-A and representation-H; where those classes are have an *equivalentClasses* axiom but have been annotated with a *Possibly Equivalent* and an *Opposite* annotation by the comparison phase (Section 4.3).

F10: Disjoint Properties The percentage of disjoint properties present in represen tation-A and representation-H. This notion is similar to disjoint classes, but for properties. The disjoint properties are the ones that has an *equivalentProperties* axiom but has been annotated with an *Opposite* annotation by the comparison phase (Section 4.3).

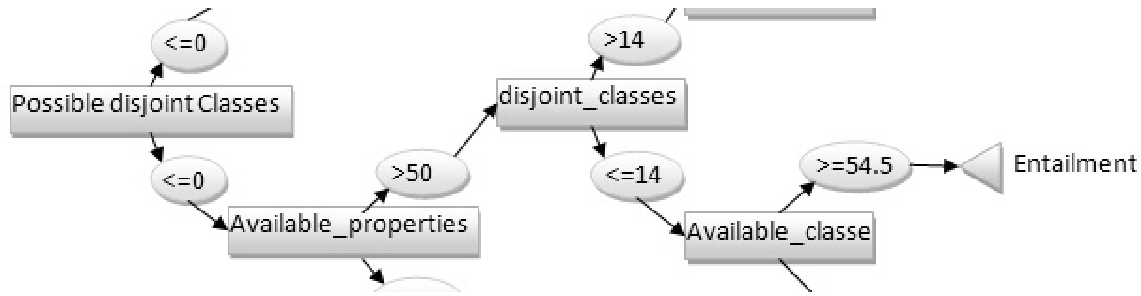

Figure 13: Part of the Resulting Decision Tree

Using these ten features, we then used the B40 decision tree classifier based on ID3 (implemented in the WEKA tool [Dimov, Feld, Kipp, Ndiaye, and Heckmann, 2007]) and trained it over the 800 text-hypothesis pairs of the RTE3 pilot task dataset [Giampiccolo, Magnini, Dagan, and Dolan, 2007]. The main reasoning for using decision tree is that they are simple to interpret, which allowed us to learn which features are more discriminatory and which one are the least. As presented in Section 2.2.3, the RTE3 pilot task focused on recognizing textual entailment; where the dataset is annotated into three decisions: *yes* for entailment, *no* for no entailment, and *unknown*. We ran the our representation acquisition and alignment prototype over the data set and created representation-T, representation-H, and representation-A for all 800 pairs. For each pair of hypothesis and text, we extracted the 10 features described above for the created representations, and used them to feed the decision tree classifier.

An analysis of the resulting decision tree, part of which is shown in Figure 13, indicates that the most discriminating features are the following:

1. F9: Possible disjoint classes (root of the tree).

2. F5, F2, and F3: Possible equivalent classes, available properties, and available sub-classes.

3. F1, F4 and F8: Available classes, equivalent classes and disjoint classes.

4. F6 and F7: Equivalent properties and possible equivalent properties.

5. F10: Disjoint properties.

This tree shows that the feature “disjoint classes” is a more discriminating feature than “equivalent classes”, which is more discriminating than “equivalent properties”. In addition, it shows that the axioms learned through VerbOcean were as important as the ones learned from WordNet.

Traversing the relevant part of the learned decision tree shows that the example feature from the representation of the example (T): *Jurassic Park is a novel written by Michael Crichton.* and hypothesis (H): *Michael Crichton crafted the book Jurassic Park.* is classified as entailment according to the created decision tree.

4.5 Evaluation and Analysis

To evaluate our approach, we participated in the TAC-RTE4 challenge (described in Chapter 2), and more specifically the three way task of recognizing textual entailment. The test set includes 1000 T-H pair to be classified into (*Entailment, Contradiction,* or *Unknown*). The evaluation is done automatically, where the classifications returned by a system are compared to human annotated golden standard, and the returned score is the accuracy or the percentage of matching judgments. As the RTE4 task did not provide a development set, we used the RTE3 pilot dataset introduced in the previous section for training. Twenty six teams submitted their system's results to the challenge [Giampiccolo, Dang, Magnini, Dagan, and Dolan, 2008]. The overall accuracies of all systems that participated in the two-way task were between 49.7% and 74.6% with an average accuracy of 58%. As for the three way task, the accuracy

Group	Accuracy
UAIC [Iftene, 2008]	0.685
OAQA (our approach)	**0.616**
DFKI [Wang and Neumann, 2008]	0.614
...	
CERES1 [Glinos, 2008]	0.405
boeing1 [Clark and Harrison, 2008]	0.377
IIITSum082 [Varma, Pingali, Katragadda, and Krisha, 2008]	0.307

Table 9: Results of the RTE4 3-way Task Participants

Group	Accuracy
LCC [Bensley and Hickl, 2008]	0.746
UAIC [Iftene, 2008]	0.720
OAQA (our approach)	**0.688**
...	
Emory3 [Agichtein, Askew, and Liu, 2008]	0.510
cambridge1 [Bergmair, 2008]	0.510
KUNLP3 [Yatbaz, 2008]	0.497

Table 10: Results of the RTE4 2-way Task Participants

was between 30.7% and 68.5% with an average accuracy much lower than the two way task of 51%. Our system achieved an accuracy of 61.6% using the B40 decision tree classifier and was ranked 2^{nd} when compared to the other systems that participated in the same challenge. Table 9 shows the results of the top 3 and bottom 3 systems in the RTE4 3-way task. The RTE challenge automatically converts the three way submitted runs of each system into two way runs by automatically conflating *"Contradiction"* and *"Unknown"* to *"No Entailment"*. The B40 decision tree classifier on the two way run achieved an accuracy of 68.8%, which ranked 3^{nd} when compared to all the systems that participated in the 2-way challenge and the 3-way challenge conflating to two way results. Table 10 shows the results of the top 3 and bottom 3 systems in the RTE4 2-way task.

The main weakness of this approach, as with any supervised machine learning approach, is its need for annotated corpora to train the system. The main strength of this approach compared to our baseline approach is its use of machine learning to decide on textual entailment result. This feature gave our approach the flexibility needed to deal with cases that were never encountered in the training set. Another strength of the approach is its use of the Web as a corpus to learn on demand additional information (with RoDEO and the types of named entities in the representation phase), and to learn axioms (using the measure of semantic relatedness in the comparison phase). In order to further investigate the impact of those two features, we tested two different settings with the current approach.

First, we tested the impact of using the Web to recognize named entities. We re-tested the same approach, with the same settings, but with one exception of using the Freebase database (as with our previous approach, see Section 3.2) instead of the RoDEO generated named entities. This ablation test resulted in a decrease of accuracy by 4%. This seems to show that using the Web as a corpus to enrich the knowledge base does result in a significant improvement. Second, we tested the impact of using the Web to learn axioms on demand that is based on semantic relatedness (introduced in the comparison phase). We re-tested the same approach, with the same settings but with one exception of using the WordNet path length semantic similarity method (as with our previous approach, see Section 3.3) instead of the combination of WordNet semantic match and VerbOcean Web patterns. This setting resulted in a statistically significant increase of accuracy of 9%. Again, this shows that the accuracy of this approach is directly affected by the coverage of the semantic relatedness calculator. One possible improvement of the current approach is to further improve the accuracy of the semantic similarity calculator, an improvement that we further experiment with in the next chapter.

4.6 Conclusion

In this chapter we investigated the use of the Web as a corpus for enriching a meaning representation (Section 4.2). We also investigated the use of semantic relatedness between concepts to learn axioms on demand (Section 4.3), as an alternative to a predefined set of axioms, and as an indicator to textual entailment (Section 4.4). Our proposed textual entailment approach, based on description logic and semantic relatedness showed that the type of semantic relatedness axioms employed in aligning the description logic representations are good indicators of textual entailment. Using a decision tree, this approach classifies textual entailment based on alignment semantic relatedness features. The implementation performance of the approach was evaluated using the Recognizing Textual Entailment (RTE-4) challenge, resulting in an accuracy of 68% on the two-way task, and 61% on the three way task, which ranked 2^{nd} when compared to the other systems that participated in the challenge. This approach and the other enhancements discussed earlier show a significant improvement over our previous approach (discussed in Chapter 3). We also have shown that the accuracy of this approach is very much affected by the accuracy of the semantic relatedness calculator. In the next chapter we will further discuss this issue and propose a new approach to calculate semantic relatedness, which we believe can improve the accuracy of recognizing textual entailment.

Chapter 5

Recognizing Textual Entailment and Lexical Semantic Relatedness

In Chapter 4 we have introduced a knowledge alignment based approach for recognizing textual entailment. An analysis of its evaluation showed that the accuracy of this approach is directly affected by the accuracy of the semantic relatedness calculator. Recall that to compute semantic relatedness, we used the WordNet Similarity path length method [Pedersen, Patwardhan, and Michelizzi, 2004a, Patwardhan, Banerjee, and Pedersen, 2003]. In this chapter we will investigate a new method for measuring lexical semantic relatedness (Section 5.1), to be used in the decision component of the RTE approach (Section 5.3).

In this chapter, we will introduce the problem of semantic relatedness (Section 5.1), our proposed approach to this problem (Section 5.1.2), and an intrinsic evaluation of our approach (Section 5.2). We follow this section, with an integration of our semantic relatedness method to our previous approach of recognizing textual entailment (Section 5.3), and an evaluation of this integration on the RTE4 dataset (Section 5.4).

A version of this approach has appeared in [Siblini and Kosseim, 2013b] and in [Siblini and Kosseim, 2013a].

5.1 Lexical Semantic Relatedness

Lexical semantic relatedness try to measure how two words are related in meaning. Many natural language processing applications such as textual entailment, question answering, or information retrieval require a robust measurement of lexical semantic relatedness. Current approaches to address this problem can be categorized into three main categories: those that rely on a lexicon and its structure, those that use the distributional hypothesis on a large corpus, and hybrid approaches.

Lexicon-based methods use the features of a lexicon to measure semantic relatedness. The most frequently used lexicon is Princeton's WordNet [Fellbaum, 1998] which groups words into synonyms sets (called synsets) and includes various semantic relations between those synsets, in addition to their definitions (or glosses). WordNet contains 26 semantic relations that include: hypernymy, hyponymy, meronymy, and entailment.

To measure relatedness, most of the lexicon-based approaches rely on the structure of the lexicon, such as the semantic link path [Patwardhan, Banerjee, and Pedersen, 2003] (which we used in Chapter 4), depth [Leacock and Chodorow, 1998, Wu and Palmer, 1994], direction [Hirst and St-Onge, 1998], or type [Tsatsaronis, Varlamis, and Vazirgiannis, 2010]. Most of these approaches exploit the hypernym/hyponym relations, but a few approaches have also included the use of other semantic relations. [Leacock and Chodorow, 1998] for example, computed semantic relatedness as the length of the shortest path between synsets over the depth of the taxonomy. [Wu and Palmer, 1994] also used the hyponym tree to calculate relatedness by using the depth of the words in the taxonomy and the depth of the least common superconcept between the two words. [Hirst and St-Onge, 1998], on the other hand, used the lexical chains between words based on their synsets and the semantic edges that connect them. In addition to using the hypernym relations, they classified the relations into classes: "extra strong" for identical words, "strong" for synonyms, "medium strong" for when there is a path between the two, and "not related" for no paths at all. The semantic measurement is then based on the path length and the path direction changes. [Tsatsaronis, Varlamis, and Vazirgiannis, 2010] used a combination of semantic path length, node depth in the hierarchy, and the types of the semantic edges that compose the path. The lexical semantic relatedness approach we used in the previous chapters was semantic link path [Patwardhan, Banerjee, and Pedersen, 2003] using the WordNet taxonomy.

On the other hand, corpus-based approaches rely mainly on distributional properties of words learned from a large corpus to compute semantic relatedness. Such as the work of [Finkelstein, Gabrilovich, and Matias, 2001] that used Latent Semantic Analysis, and the work of [Strube and Ponzetto, 2006] and [Gabrilovich and Markovitch, 2007], which both used the distributional hypothesis on Wikipedia.

Finally, hybrid approaches use a combination of corpus-based and lexicon-based methods. For example, the approach proposed by [Hughes and Ramage, 2007] used a random walk method over a lexicon-based semantic graph supplemented with corpus-based probabilities. Another example is the work of [Agirre, Alfonseca, Hall, Kravalova, Pasca, and Soroa, 2009] that used a supervised machine learning approach to combine three methods: WordNet-based similarity, a bag of word based similarity, and a context window based similarity.

The approach presented in this chapter belongs to the lexicon-based category. However, as opposed to the typical lexicon-based approaches described above and the

Category	**Weight**	**Semantic Relations in WordNet**
Similar	α	antonym, cause, entailment, participle of verb, pertainym, similar to, verb group
Hypernym	$2 \times \alpha$	derivationally related, instance hypernym, hypernym
Sense	$4 \times \alpha + \beta$	lemma-synset
Gloss	$6 \times \alpha$	lemma-gloss content words
Part	$8 \times \alpha$	holonym (part, member, substance), inverse gloss, meronym (part, member, substance)
Instance	$10 \times \alpha$	instance hyponym, hyponym
Other	$12 \times \alpha$	also see, attribute, domain of synset (topic, region, usage), member of this domain (topic, region, usage)

Table 11: Relations Categories and Corresponding Weights.

ones used in the previous chapters, the novelty of our approach is that we use all 26 semantic relations found in WordNet in addition to information found in glosses. These relations are used to create an explicit semantic network, where the edges of the network representing the semantic relations are weighted according to the type of the semantic relation. The semantic relatedness is computed as the lowest cost path between a pair of words in the network.

5.1.1 Our Approach to Semantic Relatedness

Our method to measure semantic relatedness is based on the idea that the types of relations that relate two concepts are a suitable indicator of the semantic relatedness between the two. The type of relations considered includes not only the hyponym/hypernym relations but also all other available semantic relations found in WordNet in addition to word definitions.

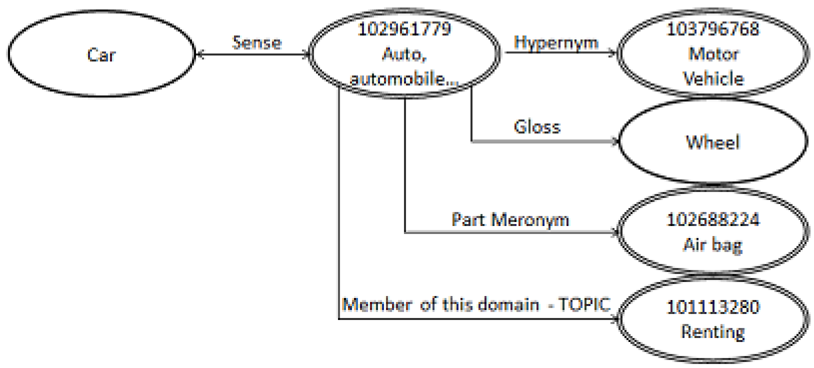

Figure 14: Example of the Semantic Network Around the Word *car*.

5.1.1.1 WordNet's Semantic Network

To implement our idea, we created a weighted and directed semantic network based on the content of WordNet. To build the semantic network, we used WordNet 3.1's words and synsets as the nodes of the network. Each word is connected by an edge to its synsets, and each synset is in turn connected to other synsets based on the semantic relations included in WordNet. In addition each synset is connected to the content words contained in its gloss. For example, Figure 14 shows part of the semantic network created around the word *car*. In this graph, single-line ovals represent words, while double-line ovals represent synsets.

By mining WordNet entirely, we created a network of 265,269 nodes connected through a total of 1,919,329 edges. The nodes include all words and synsets, and the edges correspond to all 26 semantic relations in WordNet in addition to the relation between a synset and every content word of a synset definition.

5.1.1.2 Semantic Classes of Relations

To compute the semantic relatedness between nodes in the semantic network, it is necessary to take into consideration the semantic relation involved between two nodes.

Indeed, WordNet's 26 semantic relations do not contribute equally to the semantic relatedness between words. The *hypernym* relation (relation #2), for example, is a good indicator of semantic relatedness; while the relation of *member of this domain - topic* (relation #15) is less significant. This can be seen in Figure 14, for example, where the word *car* should be more closely related to *Motor vehicle* than to *Renting*. In order to determine the contribution of each relation, we compared a manually created set of 210 semantic relations for their degree of relatedness. For example, for the concept *car*, we have compared the sense of *automobile* with the hypernym *motor vehicle*, the gloss word *wheel*, the part meronym *air bag*, the member of this topic *renting*, and another sense of *car* such as a *cable car*. From the annotated data of direct semantic relations we learned that Synonymy is on average more related than hypernymy, which is more semantically related to meronymy. This has lead us to classify the relations into seven categories, and rank these categories from the most related category to the least related ones. By classifying WordNet's relations into these classes, we are able to weight the contribution of a relation based on the class it belongs to, as opposed to assigning a contributory weight to each relations. For example, all relations of type *Similar* will contribute equally to the semantic relatedness of words, and will contribute more than any relations of the class *Hypernym*. Table 11 shows the seven semantic categories that we defined, their corresponding weight, and the WordNet relations they include. The weights[1] were simply assigned as a multiple of a small value α, representing the lowest weight, and an addition of 2 for each multiplier in the list in order to represent a higher cost of the less related categories. Let us describe each category in detail.

The category *Similar* includes WordNet's relations of *antonym*, *cause*, *entailment*,

[1]The weight can be seen as the cost of traversing an edge; hence a lower weight is assigned to a highly contributory relation.

similar to, *participle of verb*, *pertainym* and *verb group*. This class of relations includes relations that are the most useful to compute semantic relatedness as per our manual corpus analysis and are the rarest available relations in the semantic network and hence was assigned the lowest weight of all categories of relations: α.

The second category of semantic relations is the *Hypernym* which includes WordNet's relations of *hypernym*, *instance hypernym* and *derivationally related*. Being less important than the *similar* relations to compute relatedness, as shown in Table 11, the *Hypernym* category was assigned a weight of $(2 \times \alpha)$.

The *Sense* category represents the relationship between a word and its synset. Because a word can belong to several synsets, in order to favor the most frequent senses as opposed to the infrequent ones, the weight of this category is modulated by a factor β. Specifically, we use $(4 \times \alpha + \beta)$, where β is computed as the ratio of the frequency of the sense number in WordNet over the maximum number of senses for that word.

The fourth category of semantic relations is the *Gloss* that covers the relation between synsets and their glosses. A synset gloss contains a brief definition of the synset, which usually consists of a genus (or type) and one or more differentia (or what distinguishes the term from the genus). The genus relations is explicitly defined in WordNet as a hypernym relation, however the differentia is most of the time not defined. The differentia includes essential attributes of the synset being defined. For this reason, we explicitly included those relations in the semantic network. For example, the gloss of the synset *#102961779 car, auto, automobile ...* is *a motor vehicle with four wheels*, the hypernym of this synset is *motor vehicle*, and the differentia is *four wheel*. There is no semantic relation explicitly defined in WordNet between *car* and *four wheel*, nor is there a relation with *wheel*. Even if a meronymy relation between *car* and *wheel* existed in WordNet, it also should be more related to it than the rest of the meronymy relations as it is a defining attribute. To include such relations to the semantic network, we create an edge between every content word in the gloss and the synset, but only consider words that have an entry in the lexicon. As this is a simplistic approach of adding the gloss relations, we gave it a high weight of $(6 \times \alpha)$, but less than the next category covering meronymy relations. The inverse of this edge (from a gloss word to a synset) is also included, but is considered to be less related and thus included in the next category.

The fifth category is the *Part* category that includes *holonymy*, *meronymy*, and *inverse gloss* relations which are all weighted as $(8 \times \alpha)$.

The sixth category, the *Instance* category, only includes the *hyponymy* and *instance of hyponymy* relations that are weighted as $(10 \times \alpha)$.

Finally, all others relations available in WordNet are grouped under the last category *Other* and given the maximum weight of $(12 \times \alpha)$.

5.1.1.3 Calculation of Semantic Relatedness

Given the weighted semantic network extracted from WordNet, the semantic relatedness, $S(w_1, w_2)$, between two words w_1 and w_2 is computed essentially as the weight of the lowest cost path[2] between the two words. However, because the network is directed, the lowest cost from w_1 to w_2, $P_{min}(w_1, w_2)$, may be different than from w_2 to w_1, $P_{min}(w_2, w_1)$. To account for this, we therefore consider the semantic relatedness $S(w_1, w_2)$ to be equal to the highest relatedness score in either direction. More formally, the semantic relatedness between w_1 and w_2 is defined as:

$$S(w_1, w_2) = \max\left(\frac{M - (P_{min}(w_1, w_2) - K)}{M}, \frac{M - (P_{min}(w_2, w_1) - K)}{M}\right)$$

[2]The lowest cost path is based on an implementation of Dijkstras graph search algorithm [Dijkstra, 1959]

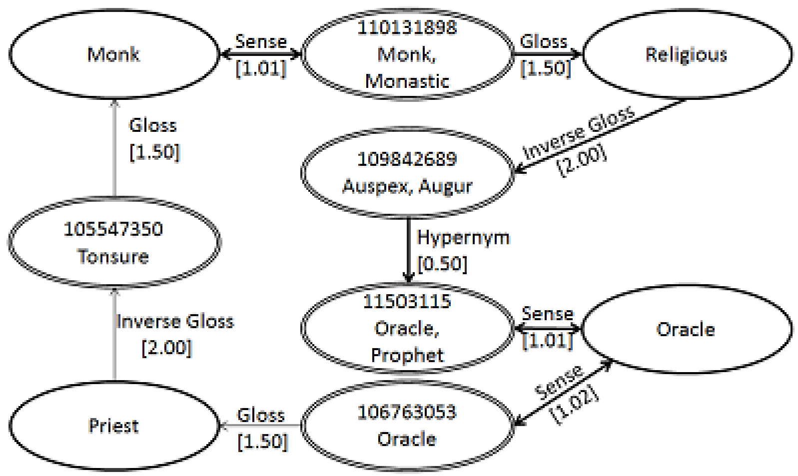

Figure 15: Lowest Cost Path Between the Words *Monk* and *Oracle*.

Where, M is a constant representing the weight after which two words are considered unrelated, and K is constant representing the weight of true synonyms. In our implementation, we have set $M = 2 \times (12 \times \alpha)$ corresponding to the maximum of traveling twice the relation with the highest weight, and $K = 2 \times (4 \times \alpha)$ corresponding to the minimum of traveling from a word to its sense and back to the word itself.

5.1.1.4 An Example

Figure 15 shows an extract of the network involving the words *Monk* and *Oracle*. The lowest cost path from *Monk* to *Oracle* in highlighted in bold. As the figure shows, the word *Monk* is connected with a *Sense* relation to the synset *#110131898 [Monk, Monastic]*. As indicated in Table 1, the weight of this relation is computed as $(4 \times \alpha + \beta)$. Because this synset is the first sense (the most frequent sense given by WordNet) for the word *Monk*, then ($\beta = 1/75 = 0.01$, where 75 is the maximum number of senses for a word in WordNet. If α is set to 0.25, then, as shown in Figure 15, the weight of this edge is computed $(4 \times 0.25 + 0.01 = 1.01)$.

The synset *#11013898 [Monk, Monastic]* is connected to the word *Religious* through a *Gloss* relation type. In WordNet, the gloss of this synset is: *a male religious living in a cloister and devoting himself to contemplation and prayer and work*. The content words are: *male, religious, live, cloister, devote, contemplation, prayer*, and *work*, which are each related to this synset with the weight set to ($6 \times \alpha = 1.5$).

Overall, the weight of the lowest cost path $P_{min}(Monk, Oracle)$ is hence equal to the sum of the edges shown in Figure 1 (1.01+1.50+2.00+0.50+1.01 = 6.02). As the figure shows, in this example, $P_{min}(Monk, Oracle)$ is identical to $P_{min}(Oracle, Monk)$. With the constants M set to 6 and K to 2, $S(Monk, Oracle)$ will therefore be (6-(6.02-2))/6 = 0.33.

5.2 Intrinsic Evaluation of Semantic Relatedness

To evaluate our approach to semantic relatedness intrinsically, we used two types of benchmarks: using human ratings and using synonym tests.

5.2.1 Evaluation using Human Ratings

In their study on semantic similarity, [Miller and Charles, 1991] (M&C) gave 38 undergraduate students 30 pairs of nouns to be rated from 0, for no similarity, to 4, for perfect synonymy. The noun pairs were chosen to cover high, intermediate, and low level of similarity and are part of an earlier study [Rubenstein and Goodenough, 1965] (R&G) which contained 65 pairs of nouns. The M&C test gained popularity among the research community for the evaluation of semantic relatedness. The evaluation is accomplished by calculating the correlation between the average student's ratings and one's approach. The commonly used correlation measurement for this test is the Pearson correlation measurement [Pearson, 1900], but some have also used the

Spearman ranking coefficient [Spearman, 1904] as an evaluation measurement. Our approach achieved a Pearson correlation of 0.93 and a Spearman correlation of 0.87 with the M&C data set. In addition, it achieved a 0.91 Pearson correlation and a 0.92 Spearman correlation on the R&G data set.

For comparative purposes, Table 12 shows the Pearson correlation of several previous approaches to semantic relatedness measures against the same data set, as reported in their respective papers. For information, the table indicates the type of approach used: lexicon-based method, corpus-based method, or hybrid. As Table 12 shows, most other approaches achieve a correlation around 85%, while a few achieve a correlation above 90%. These results do not seem to be influenced by the type approach. Our approach compares favorably to the state of the art in the field on the Miller and Charles data set, with a high correlation of 93%. Our result is higher than any other lexicon based approach, however it must be noted that the Miller and Charles Data Set is quite small for empirical analysis.

WordSimilarity-353 is another set of human ratings that was introduced by [Finkelstein, Gabrilovich, and Matias, 2001]. The data set is much larger than the Miller and Charles Data Set and includes 353 pairs of words, each rated by 13 to 16 subjects who were asked to estimate the relatedness of the words on a scale of 0 for “totally unrelated words” to 10 for “very much related or identical words”. The common practice with this data set is to the use the Spearman coefficient.

Table 13 shows various approaches and their corresponding Spearman correlation as described in the literature. On this data set, our approach achieved a correlation of 0.50, which is quite lower than the current state of the art. After analysing our results, we identified several reasons why our approach did not perform as expected. First, all lexicon based methods seem to perform poorly on this data set because

Approach	Category	Pearson
[Gabrilovich and Markovitch, 2007]	Corpus	0.72
[Hirst and St-Onge, 1998]	Lexicon	0.74
[Wu and Palmer, 1994]	Lexicon	0.78
[Resnik, 1995]	Hybrid	0.80
[Leacock and Chodorow, 1998]	Lexicon	0.82
[Lin, 1998a]	Hybrid	0.83
[Bollegala, Matsuo, and Ishizuka, 2007]	Corpus	0.83
[Jiang and Conrath, 1997]	Hybrid	0.85
[Tsatsaronis, Varlamis, and Vazirgiannis, 2010]	Lexicon	0.86
[Jarmasz and Szpakowicz, 2003]	Lexicon	0.87
[Hughes and Ramage, 2007]	Lexicon	0.90
[Alvarez and Lim, 2007]	Lexicon	0.91
[Yang and Powers, 2005]	Lexicon	0.92
[Agirre, Alfonseca, Hall, Kravalova, Pasca, and Soroa, 2009]	Hybrid	0.93
Our approach	**Lexicon**	**0.93**

Table 12: Pearson Correlation of Various Approaches on the Miller and Charles Data Set.

it includes a number of named entities that are typically not available in a lexicon. For example, in the word pair: *(Maradona – football)*, the word *Maradona* does not appear in WordNet, hence favoring corpus-based and hybrid approaches. Another difficulty is the high variance of human ratings for some word pairs, which could be due to the subjectivity required for this task, or the fact that the subjects who rated the data set were not native English speakers. That being said, perhaps the most important factor for the poor performance of lexicon based methods (including ours) is that most of the pairs in that data set require general world knowledge that is not usually available in a lexicon. Nevertheless, other approaches were able to achieve a high correlation with this data set such as the machine learning approach of [Agirre, Alfonseca, Hall, Kravalova, Pasca, and Soroa, 2009] that achieved a high correlation of 0.78.

Approach	Category	Spearman
[Strube and Ponzetto, 2006]	Corpus	0.48
[Jarmasz and Szpakowicz, 2003]	Lexicon	0.55
[Hughes and Ramage, 2007]	Lexicon	0.55
[Finkelstein, Gabrilovich, and Matias, 2001]	Hybrid	0.56
[Gabrilovich and Markovitch, 2007]	Corpus	0.75
[Agirre, Alfonseca, Hall, Kravalova, Pasca, and Soroa, 2009]	Hybrid	0.78
Our approach	**Lexicon**	**0.50**

Table 13: Spearman Correlation of Various Approaches on WordSimilarity-353 Data Set.

5.2.2 Evaluation using Synonym Tests

To test the approach further, we also evaluated it on synonym identification tests. This type of test includes an initial word and a set of options from which the most synonymous word must be selected.

The first synonym test that we experimented with is the English as a Second Language (ESL) test. The test set was first used by [Turney, 2001] as an evaluation of algorithms measuring the degree of similarity between words. The ESL test includes 50 synonym questions and each having four choices. The following is an example question taken from ESL data set:

```
Text:  A rusty nail is not as strong as a clean, new one.

Stem:  rusty

Choices:

(a) corroded

(b) black

(c) dirty

(d) painted

Solution:  (a) corroded
```

Approach	Category	Accuracy
[Resnik, 1995]	Hybrid	32.66%
[Leacock and Chodorow, 1998]	Lexicon	36.00%
[Lin, 1998a]	Hybrid	36.00%
[Jiang and Conrath, 1997]	Hybrid	36.00%
[Hirst and St-Onge, 1998]	Lexicon	62.00%
[Turney, 2001]	Corpus	74.00%
[Terra and Clarke, 2003]	Corpus	80.00%
[Jarmasz and Szpakowicz, 2003]	Lexicon	82.00%
[Tsatsaronis, Varlamis, and Vazirgiannis, 2010]	Lexicon	82.00%
Our Approach	**Lexicon**	**84.00%**

Table 14: Results with the ESL Data Set.

The results of our approach, along with other standard approaches, on the 50 ESL questions are shown in Table 14. The results are measured in terms of accuracy - the percentage of correct responses by each approach. Our approach has achieved an accuracy of 84% on the ESL test, which is slightly better than the reported approaches in the literature. It should be noted that sometimes the difference between two approaches belonging to the same category are merely a difference in the data set used (Corpus or Lexicon) rather than a difference in the algorithms. Also, the ESL question set includes a sentence to give a context for the word, which some approaches (e.g. [Turney, 2001]) have used as an additional information source; we on the other hand, did not make use of the context information in our approach.

The second synonym test that we used is the Test of English as a Foreign Language (TOEFL) test. The test was first used by [Landauer and Dumais, 1997] as an evaluation for the algorithm measuring the degree of similarity between words. The TOEFL test includes 80 synonym questions each having four choices. The following is an example taken from the TOEFL data set:

```
Stem: levied
Choices:
(a) imposed
(b) believed
(c) requested
(d) correlated
Solution: (a) imposed
```

The results on the 80 TOEFL questions are shown in Table 15, which also includes the results of other approaches for comparative purposes. Here again, the results are reported in terms of accuracy. As with the previous experiments, the category of the approach does not seem to have an impact on the results. It should be noted, however, that some of the approaches have been tuned specifically for the TOEFL questions. Table 15 also includes an entry for the "Average non-English US college applicant" of 64.5%. The score that was originally reported in [Landauer and Dumais, 1997] is 52.5% for college applicants, however this figure penalizes random guessing by subtracting a penalty of 1/3. To provide a more fair comparison, this penalty has been removed leading to a score of 64.5%. Our approach has achieved an accuracy of 91.25% on the TOEFL test, which is better than any of the reported lexicon based approaches.

5.2.3 Evaluation on a Word-Phrase Semantic Relatedness

We also evaluated our approach for word-phrase semantic relatedness on the recent SemEval-2013 Task 5: Evaluating phrasal semantics [Korkontzelos, Zesch, Zanzotto,

Approach	Category	Accuracy
[Resnik, 1995]	Corpus	20.31%
[Leacock and Chodorow, 1998]	Lexicon	21.88%
[Lin, 1998a]	Hybrid	24.06%
[Jiang and Conrath, 1997]	Hybrid	25.00%
[Landauer and Dumais, 1997]	Corpus	64.38%
Average non-English US college applicant	*Human*	*64.50%*
[Padó and Lapata, 2007]	Corpus	73.00%
[Hirst and St-Onge, 1998]	Lexicon	77.91%
[Jarmasz and Szpakowicz, 2003]	Lexicon	78.75%
[Terra and Clarke, 2003]	Corpus	81.25%
[Ruiz-Casado, Alfonseca, and Castells, 2005]	Corpus	82.55%
[Irina MaTveeva and Royer, 2005]	Corpus	86.25%
[Tsatsaronis, Varlamis, and Vazirgiannis, 2010]	Lexicon	87.50%
[Rapp, 2003]	Corpus	92.50%
[Turney, Littman, Bigham, and Shnayder, 2003]	Hybrid	97.50%
[Bullinaria and Levy, 2012]	Corpus	100.00%
Our Approach	**Lexicon**	**91.25%**

Table 15: Results with the TOEFL Data Set.

and Biemann, 2013], and more specifically on the sub-task of evaluating the semantic similarity between words and phrases. The task provided an English dataset of 15,628 word-phrases, 60% annotated for training and 40% for testing, with the goal of classifying each word-phrase as either positive or negative. For example, a positive example from the dataset includes the word "valuation" and the phrase "price assessment". To compute of semantic relatedness between a word and a compositional phrase, we combined the weights of the lowest cost path in the weighted semantic network between that word and every word in that phrase, normalized by the maximum path cost.

To transform the semantic relatedness measure to a semantic similarity classification one, we used JRip, WEKA's [Dimov, Feld, Kipp, Ndiaye, and Heckmann, 2007] implementation of Cohen's RIPPER rule learning algorithm [Cohen and Singer, 1999], in order to learn a set of rules that can differentiate between a positive semantic

Approach	**Recall**	**Precision**	**F-Measure**
[Waterna, 2013]	75.2%	83.7%	79.2%
[Van de Cruys, Afantenos, and Muller, 2013]	61.4%	83.8%	70.9%
[Dávila, Orquín, Chávez, and Gutiérrez, 2013]	61.3%	78.7%	68.9%
Harbin Institute of Technology	50.1%	84.0%	62.8%
Our Approach	**70.6%**	**85.5%**	**77.4%**

Table 16: Results with the Word-Phrase Data Set.

similarity and a negative one. The classifier resulted in rules for the semantic network model based relatedness that could be summarized as follows: *If the semantic relatedness of the word-phrase is over 61% then the similarity is positive, otherwise it is negative.* For the example *Interview* - *Formal meeting*, which resulted in a semantic relatedness of 66.7% with our semantic network approach, would be classified positively by the generated rule. This method was our first submitted test run to this task and resulted in a recall of 70.6%, a precision of 85.5%, and an F-measure of 77.4% on the testing set. The results on this tasks are shown in Table 16, which also includes the results of other approaches for comparative purposes. Five research teams participated in the task, and our approach was ranked 2$^{\text{nd}}$ when compared to the others [Korkontzelos, Zesch, Zanzotto, and Biemann, 2013].

5.3 Knowledge Alignment Approach to RTE Revisited

Recall that the purpose of developing a new approach for semantic relatedness was to improve our RTE approach of Chapter 4. Hence, after the intrinsic evaluations of our semantic relatedness approach (see Section 5.2), we used our new approach to replace the methods used in our RTE system. As with our previous RTE approach, our final knowledge alignment approach can be divided into three main components:

a representation, a comparison component, and a decision component. The representation component follows the same method as the previous approach (see Section 4.2). Then, the comparison component compares and aligns the two created representation similarly to the previous approach (see Section 4.3), but this time the semantic relatedness calculator used is based on our new approach for measuring semantic relatedness. Finally, the decision component is also similar to the previous approach (see Section 4.4). Basically the only change from the previous approach is the method used to calculate semantic relatedness and the method used to transform this information into axioms accordingly. Figure 16 shows an illustration of our final RTE approach discussed in this chapter. The different components are shown in dashed boxes, the inputs and outputs in ovals, and the sub-components in rectangles. The differences between this approach and the one from Chapter 4 are highlighted grey. Note that the Figure 16 differs from Figure 8 (in Chapter 4) only in the semantic relatedness component.

First, similarly to the approach of Chapter 4, the representation component of this approach creates two representations, one for the text and one for the hypothesis (as described in Section 4.2).

Then the comparison and the alignment of the created representations will result in one single aligned representation, namely representation-A. The alignment phase aligns the classes and properties of the two created representations. The alignment takes as its base the representation created from T and adds to it the classes and properties that align from the hypothesis representation. At this stage the weighted graph semantic relatedness measure described in Section 5.2 is used to perform the semantic

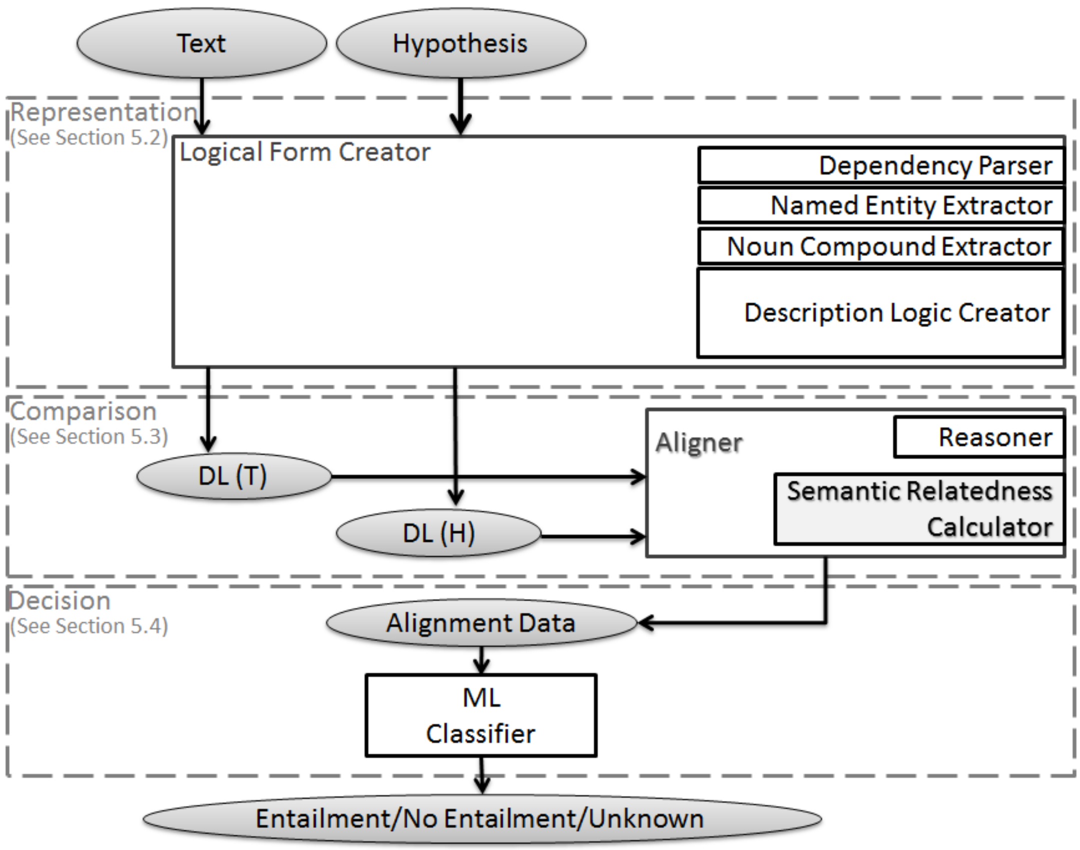

Figure 16: Knowledge Alignment Approach Augmented with New Semantic Relatedness Calculator

comparison and alignment of classes and properties. The algorithm takes the two representations and produces mapping among the classes and properties that correspond semantically to each other. For any pair of classes from the two representations, it computes the strongest semantic relation holding between each concept, which in our case is simply the highest semantic relatedness measure that exists between the two. Once the strongest relation is selected, we transform it into an axiom as follows: A semantic relatedness score of over 75% will translate to an equivalentClasses(X Y) or equivalentObjectProperties(X Y) axiom in the representation (depending on the matching type in the representation T). Otherwise, a semantic relatedness score of over 50% translate to equivalentClasses(X Y) or equivalentObjectProperties(X Y), but those axioms will be also annotated with a "Possibly Equivalent" label that will be used later in the decision phase.

Finally, the decision component is similar to Section 4.4, which decides on alignment from a set of features based on the above axioms.

5.4 Evaluation and Analysis

We evaluated this approach using the same methodology as in Chapter 4, with the RTE4 three way task of recognizing textual entailment. The test set includes 1000 T-H pair to be classified into (*Entailment, Contradiction, or Unknown*). our new approach resulted in an accuracy of 56% using the B40 decision tree classifier. Compared to the accuracy achieved by the previous system of 61.6%, our new approach suffered a significant decrease. This result was very surprising, because the only difference with the two approaches is the semantic relatedness measure used, which had very competitive accuracies with the intrinsic evaluations using several benchmark (see Section 5.2). However, the new approach did not perform as well extrinsically in the task of recognizing textual entailment, resulting in accuracy above the overall average of the other systems that participated in the challenge, but lower than our previous approach accuracy. A further analysis of our results shows that although VerbOcean created some noise (for example, according to VerbOcean, buy is similar to produce and distribute), it was able to provide further coverage. To give you an example, the following VerbOcean similar words: *"lock-barricade, loot-vandalize, heckle-boo..."* have very low similarity as measured by our lexical similarity approach. For example, the shortest path between "heckle and boo" is a path of length 9, leading to a low relatedness of 8%. One possible improvement of this approach is a hybrid between the two measurements, taking into consideration the web based method for added recall, and the WordNet semantic relatedness for added accuracy, maybe as separate features into the machine learning algorithm.

5.5 Conclusion

In this chapter, we have investigated the development of a new lexicon based method to measure semantic relatedness. This approach is based on the types of semantic relations between concepts as an indicator of relatedness, and has improved the current state of the art for lexicon based semantic relatedness measures. Our results show that this approach outperforms many lexicon-based methods to semantic relatedness, especially on the TOEFL synonym test, achieving an accuracy of 91.25%. We also learned from the human annotated data that different types of semantic relations have on average different degree of relatedness. We learned for example, that words connected with a Synonymy relations are on average more semantically related than words connected with a hypernymy, which are more semantically related to meronymy. We also investigated the use of this semantic relatedness measure in the alignment of representations and for the task of recognizing textual entailment, as the basis of the comparison component (Section 5.3). The prototype implementation performance was evaluated using the Recognizing Textual Entailment (RTE-4) challenge, resulting in an accuracy of 56% on the three-way task, above the overall average of the other systems that participated in the challenge, but lower than our previous approach accuracy of 61.6%. We believe that the high precision of our new approach was offset by its lower recall and VerbOcean's web search (used in Chapter 4) resulted in a much higher recall which ultimately resulted in a higher accuracy in the RTE task. A hybrid approach between the two semantic relatedness measures is likely to be a fruitfull research avenue.

Chapter 6

Conclusion and Future Work

As a full logical-based meaning representation of text is still a challenging problem that requires a large knowledge base of inference rules, we investigated an alternative approach of using a description logical-based meaning representation, that learns axioms on demand, and applied it to the task of recognizing textual entailment. We started our work with an initial investigation of the use of description logic to recognizing textual entailment (detailed in Chapter 3). The analysis of this baseline approach highlighted the need to 1) acquire missing information, 2) acquire missing axioms, and 3) improve the computation of semantic relatedness. For the first purpose, we showed how the Web can be used as a corpus for enriching a meaning representation of a text, and specifically for adding semantic types of named entities. This subsystem, called RoDEO, was evaluated intrinsically and as part of our RTE systems and showed competitive results in both cases.

For the second purpose, we learned axioms on demand by translating semantic relations from WordNet and from the Web to equivalence axioms with the purpose of aligning two representations to recognize textual entailment. The types of axioms employed by the reasoner are then used as features in a machine learning algorithm that learned to infer if a text entails a hypothesis or not. To validate our approach we have implemented an RTE system named AORTE (described in Chapter 4), and evaluated its performance on recognizing textual entailment using the fourth recognizing textual entailment challenge [Giampiccolo, Dang, Magnini, Dagan, and Dolan, 2008]. The system classified 1000 T-H pair into the three way task (*Entailment, Contradiction,* or *Unknown*). When compared to the human annotated golden standard, the system resulted accuracy of 61.6% and ranked as 2^{nd} when compared to the other 26 participating runs in the same challenge.

Finally, for the purpose of improving the computation of lexical semantic relatedness, we showed how we can use the types of semantic relations between concepts as an indicator of relatedness. Our approach clearly improved the current state of the art in lexicon based semantic relatedness measurement (detailed in Chapter 5) when an intrinsic evaluation was performed. However, investigating its use for the task of recognizing textual entailment resulted in a lower accuracy of 56% on the RTE-4 challenge. We suspect its lower recall as compared to VerbOcean's Web based semantic relations to be the cause of this reduced accuracy.

6.1 Main Findings and Contributions of the Thesis

In this thesis, we showed that the type of semantic relatedness axioms used to align meaning representations can be a good indicator of textual entailment. A machine learning algorithm was used to learn textual entailment from a set of features that are based on the type of axioms used to align meaning representations and showed very competitive results at RTE-4 challenge. The developments in this thesis contribute to research in Natural Language Processing in the following ways:

Development of a Recognizing Textual Entailment Approach

Our main contribution is the development of a novel approach to recognizing textual entailment based on description logic and semantic relatedness, which achieved very competitive results at RTE-4. We have developed a method for representing texts automatically in description logic, that was published in [Siblini and Kosseim, 2008a] and is described in Chapter 3.

We have also identified a set of features based on the type of logical statements used to align textual representations in a machine learning algorithm to recognize textual entailment. This tree shows that the feature "disjoint classes" is a more discriminating feature than "equivalent classes", which is more discriminating than "equivalent properties". This was published in [Siblini and Kosseim, 2009] and is described in Chapter 4. To show how our approach can be used in RTE, we have built a prototype (called AORTE) and evaluated it using the fourth recognizing textual entailment challenge [Giampiccolo, Dang, Magnini, Dagan, and Dolan, 2008]. The prototype classified 1000 T-H pair into the three way task and achieved an accuracy of 61.6% on which ranked the approach as 2^{nd} when compared to the other participating runs in the same challenge.

This approach has led to other contributions that are described below.

Development of a Method to Natural Language Querying

We have designed a novel approach to query a knowledge base in natural language. This approach is based on predicate selectional preferences to answer queries in natural language. To test the proposed approach and its usefulness, we have developed a natural language querying prototype (called ONLI). The prototype was evaluated using the Fungal Web Ontology and achieved a mean-reciprocal rank (MRR) of 0.72, this led to the publication in [Kosseim, Siblini, Baker, and Bergler, 2006].

Development of a Web Based Named Entity Recognition Approach

We have developed an approach to extract fine-grained classes of named entities by exploring the web linguistically as a corpus. This approach is based on lexical preferences of grammatical relations in addition to a set of grammatical patterns. We have evaluated the approach with 1019 named entities chosen at random and achieved an accuracy of 70% when compared with the FreeBase knowledge base. This has led to the publication in [Siblini and Kosseim, 2008b].

Development of a Semantic Relatedness Approach

The alignment of meaning representation relies heavily on the semantic relatedness between concepts. Consequently, the accuracy of the alignment is directly related to the accuracy of the semantic relatedness measurement. To improve that accuracy, we have also investigated a new approach to measure semantic relatedness. This approach is based on the assumption that the type of semantic relations in a lexicon can be a good indicator of semantic relatedness. We evaluated our lexicon semantic relatedness approach using correlation with human ranking of semantic relatedness, and standard synonymy tests. Our approach shows a Pearson's correlation of 93% with human ranking of semantic relatedness with the [Miller and Charles, 1991] (M&C) dataset and an accuracy of 91.25% on the TOEFL synonym set. This result significantly improves the state of the art of lexicon-based approaches. We also learned that different types of semantic relations have on average different degree of relatedness. For example, words connected with a synonymy relations are on average more semantically related than words connected with a hypernymy, which are themselves more semantically related to meronymy. This work (described in details in Section 5.1) was published in [Siblini and Kosseim, 2013b]. An extension of this method, which was devised to detect phrasal similarity has been published in [Siblini and Kosseim, 2013a]. We have also applied our semantic relatedness measure to RTE, and achieved an accuracy of 56% on the three way task of the fourth recognizing textual entailment challenge.

6.2 Directions for Future Research

Future work can be directed to improve the current limitations of our approach. The first limitation is that the performance of the RTE approach is directly related to the accuracy of the lexical semantic relatedness (as shown in the ablation test of Section 5.4). Although we have tried to improve that performance with our lexicon based approach described in Chapter 5, we foresee that a hybrid approach that relies both on a lexicon (for high precision) and a corpus based (for high recall) might be a better strategy.

Validating our approach was performed by evaluating it with the standard challenges of recognizing textual entailment, and comparing it to other runs on the same challenges. That being said, the current trend in the field has been more focused toward application-oriented tasks. This trend was initiated in the 5^{th} RTE challenge with the introduction of a RTE search task, and became the main focus of the next challenges. This trend was also confirmed by the latest RTE challenge (2013), whose main task is to assess the accuracy of student answers with respect to known correct reference answers. Our interest was, and still is, directed toward decreasing the scale of the problem in order to be able to better understand and improve the issues that we still face when simply trying to classify a two-way entailment.

Finally, one interesting line of work is the formal semantic annotation of inference phenomena in RTE examples. The current RTE challenges datasets only contain annotation of entailment classes (i.e. entailment, no-entailment, unknown ...), but they do not contain information about the underlying phenomena that are involved in the entailment process, such as the annotation suggested by [Toledo, Katrenko, Alexandropoulou, Klockmann, Stern, Dagan, and Winter, 2013]. The availability of such annotation would allow us to evaluate an RTE solution on a specific entailment phenomenon, instead of focusing on the application level as is the trend of the recent RTE challenges.

Bibliography

R. Adams. Textual entailment through extended lexical overlap. In *Proceedings of the Second PASCAL Challenges Workshop on Recognising Textual Entailment*, pages 128–133, Venice, Italy, April 2006.

A. Ageno, D. Farwell, D. Ferres, F. Cruz, and H. Rodríguez. TALP at TAC 2008: A semantic approach to Recognizing Textual Entailment. In *Proceedings of the Fourth PASCAL Challenges Workshop on Recognizing Textual Entailment*, Gaithersburg, Maryland, USA, November 2008.

Eugene Agichtein, Walt Askew, and Yandong Liu. Combining lexical, syntactic, and semantic evidence for textual entailment classification. In *Proceedings of the 2008 Text Analysis Conference (TAC'08)*, Gaithersburg, Maryland, USA, November 2008.

Eneko Agirre, Enrique Alfonseca, Keith Hall, Jana Kravalova, Marius Pasca, and Aitor Soroa. A study on similarity and relatedness using distributional and WordNet-based approaches. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 19–27, Boulder, Colorado, June 2009. Association for Computational Linguistics.

E. Alfonseca and S. Manandhar. Extending a lexical ontology by a combination of distributional semantics signatures. In *Proceedings of the 13th International Conference on Knowledge Engineering and Knowledge Management (EKAW 2002)*, pages 1–7, Siguenza, Spain, October 2002. Springer.

Marco A. Alvarez and SeungJin Lim. A graph modeling of semantic similarity between words. In *Proceedings of the First IEEE International Conference on Semantic Computing (ICSC 2007)*, pages 355–362, Irvine, California, September 2007. IEEE Computer Society.

Sören Auer, Christian Bizer, Georgi Kobilarov, Jens Lehmann, Richard Cyganiak, and Zachary Ives. DBPedia: A nucleus for a web of open data. *The Semantic Web*, pages 722–735, 2007.

Franz Baader. *The description logic handbook: theory, implementation, and applications*. Cambridge university press, 2003.

C.J.O. Baker, A. Shaban-Nejad, S. Xu, V. Haarslev, and G. Butler. Semantic web infrastructure for fungal enzyme biotechnologists. *Journal of Web Semantics: Special Edition on Semantic Web for the Life Sciences*, 2006.

Collin F Baker, Charles J Fillmore, and John B Lowe. The Berkeley FrameNet project. In *Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics*, pages 86–90, Montreal, Quebec, August 1998. Association for Computational Linguistics.

R. Bar Haim, I. Dagan, B. Dolan, L. Ferro, D. Giampiccolo, B. Magnini, and I. Szpektor. The second PASCAL recognising textual entailment challenge. In *Proceedings of the second PASCAL recognising textual entailment challenge*, Venice, Italy, April 2006.

R. Bar-Haim, J. Berant, and I. Dagan. A compact forest for scalable inference over entailment and paraphrase rules. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 3*, pages 1056–1065, Singapore, August 2009. Association for Computational Linguistics.

S. Bayer, J. Burger, L. Ferro, J. Henderson, and A. Yeh. MITRE's Submissions to the EU Pascal RTE Challenge. In *Proceedings of the Pattern Analysis, Statistical Modelling, and Computational Learning (PASCAL) Challenges Workshop on Recognising Textual Entailment*, Southampton, UK, April 2005.

Jeremy Bensley and Andrew Hickl. Application of LCCs GROUNDHOG System for RTE-4. In *Proceedings of the 2008 Text Analysis Conference (TAC'08)*, Gaithersburg, Maryland, USA, November 2008.

L. Bentivogli, I. Dagan, H.T. Dang, D. Giampiccolo, and B. Magnini. The fifth PASCAL recognizing textual entailment challenge. In *Proceedings of the 2009 Text Analysis Conference (TAC'09)*, volume 9, pages 14–24, Gaithersburg, Maryland, USA, November 2009.

L. Bentivogli, P. Clark, I. Dagan, H. Dang, and D. Giampiccolo. The sixth PASCAL recognizing textual entailment challenge. In *Proceedings of the 2010 Text Analysis Conference (TAC'10)*, Gaithersburg, Maryland, USA, November 2010.

L. Bentivogli, P. Clark, I. Dagan, H. Dang, and D. Giampiccolo. The seventh PASCAL recognizing textual entailment challenge. In *Proceedings of the 2011 Text Analysis Conference (TAC'11)*, Gaithersburg, Maryland, USA, November 2011.

Richard Bergmair. Monte Carlo semantics: MCPIET at RTE4. In *Proceedings of the 2008 Text Analysis Conference (TAC'08)*, pages 17–19, Gaithersburg, Maryland, USA, November 2008.

T. Berners-Lee. Semantic Web Road Map. *World Wide Web Consortium (W3C)*, 1998.

D.M. Bikel, R. Schwartz, and R.M. Weischedel. An Algorithm that Learns What's in a Name. *Machine Learning*, 1999.

K. Bollacker, R. Cook, and P. Tufts. Freebase: A shared database of structured general human knowledge. In *Proceedings of the National Conference on Artificial Intelligence*, volume 22, Vancouver, British Columbia, Canada, 2007. AAAI Press.

Kurt Bollacker, Colin Evans, Praveen Paritosh, Tim Sturge, and Jamie Taylor. Freebase: a collaboratively created graph database for structuring human knowledge. In *Proceedings of the 2008 ACM SIGMOD international conference on Management of data*, pages 1247–1250. ACM, 2008.

Danushka Bollegala, Yutaka Matsuo, and Mitsuru Ishizuka. Measuring semantic similarity between words using web search engines. In *Proceedings of the Sixteenth International World Wide Web Conference (WWW2007)*, volume 7, pages 757–786, Banff, Alberta, Canada, May 2007.

J. Bos and K. Markert. When logical inference helps determining textual entailment (and when it doesn't). In *Proceedings of the Second PASCAL RTE Challenge*, Venice, Italy, April 2006.

Thorsten Brants and Alex Franz. Web 1T 5-gram version 1. *Linguistic Data Consortium*, LDC2006T13, 2006.

Tim Bray, Jean Paoli, C Michael Sperberg-McQueen, Eve Maler, and François Yergeau. Extensible markup language (XML). *World Wide Web Journal*, 2(4): 27–66, 1997.

E. Breck. A simple system for detecting non-entailment. In *Proceedings of the 2009 Text Analysis Conference (TAC'09)*, Gaithersburg, Maryland, USA, November 2009.

Dan Brickley and Ramanathan V Guha. RDF Vocabulary Description Language 1.0: RDF Schema. *The World Wide Web Consortium (W3C)*, 2004.

John A Bullinaria and Joseph P Levy. Extracting semantic representations from word co-occurrence statistics: stop-lists, stemming, and svd. *Behavior Research Methods*, 44:890–907, September 2012. ISSN 15543528.

Elena Cabrio and Bernardo Magnini. Toward qualitative evaluation of textual entailment systems. In *Proceedings of the 23rd International Conference on Computational Linguistics*, pages 99–107, Beijing, August 2010. Association for Computational Linguistics.

Jean Carletta. Squibs and Discussions Assessing Agreement on Classification Tasks: The Kappa Statistic. *Computational Linguistics*, 22(2):249–254, 1996.

J.J. Castillo and L.A. Alemany. An approach using named entities for recognizing textual entailment. In *Proceedings of the 2008 Text Analysis Conference (TAC'08)*, Gaithersburg, Maryland, USA, November 2008.

N. Chinchor. MUC-7 named entity task definition. In *Proceedings of the 7th Message Understanding Conference (MUC-7)*, Fairfax, VA, USA, 1998.

Timothy Chklovski and Patrick Pantel. VerbOcean: Mining the Web for Fine-Grained Semantic Verb Relations. In *Proceedings of Empirical Methods in Natural Language Processing (EMNLP-04)*, Barcelona, Spain, July 2004.

P. Cimiano and S. Staab. Learning by googling. *ACM SIGKDD Explorations Newsletter*, 6(2):24–33, 2004.

P. Clark and P. Harrison. Recognizing textual entailment with logical inference. In *Proceedings of the 2008 Text Analysis Conference (TAC'08)*, Gaithersburg, Maryland, USA, November 2008.

S. Clinchant, C. Goutte, and E. Gaussier. Lexical entailment for information retrieval. *Advances in Information Retrieval*, pages 217–228, 2006.

William W Cohen and Yoram Singer. A simple, fast, and effective rule learner. In *Proceedings of the National Conference on Artificial Intelligence*, pages 335–342, Orlando, Florida, July 1999.

C. Condoravdi, D. Crouch, V. De Paiva, R. Stolle, and D.G. Bobrow. Entailment, intensionality and text understanding. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology (HLT-NAACL '03)*, volume 21, pages 38–45, Edmonton, May 2003. Association for Computational Linguistics.

C. Corley and R. Mihalcea. Measuring the semantic similarity of texts. In *Proceedings of the Association for Computational Linguistics (ACL) Workshop on Empirical Modeling of Semantic Equivalence and Entailment*, pages 13–18, Ann Arbor, Michigan, June 2005. Association for Computational Linguistics.

I. Dagan, O. Glickman, and B. Magnini. The PASCAL recognising textual entailment challenge. *Machine Learning Challenges: Evaluating Predictive Uncertainty, Visual Object Classification, and Recognising Textual Entailment*, pages 177–190, April 2005.

Ido Dagan and Oren Glickman. Probabilistic textual entailment: Generic applied modeling of language variability. *Learning Methods for Text Understanding and Mining*, 2004.

Héctor Dávila, Antonio Fernández Orquín, Alexander Chávez, and Gutiérrez. UMCC_DLSI-(EPS): Paraphrases Detection Based on Semantic Distance. In *Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 2: Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013)*, pages 108–113, Atlanta, Georgia, USA, June 2013.

M.C. de Marneffe, B. MacCartney, T. Grenager, D. Cer, A. Rafferty, and C.D. Manning. Learning to distinguish valid textual entailments. In *Proceedings of the Second PASCAL RTE Challenge Workshop*, Venice, Italy, April 2006.

R. de Salvo Braz, R. Girju, V. Punyakanok, D. Roth, and M. Sammons. An inference model for semantic entailment in natural language. *Machine Learning Challenges: Evaluating Predictive Uncertainty, Visual Object Classification, and Recognising Textual Entailment*, pages 261–286, 2006.

M. Dean, G. Schreiber, et al. OWL Web Ontology Language Reference. *W3C Recommendation*, 10, 2004.

Rodolfo Delmonte, Sara Tonelli, and Rocco Tripodi. Semantic processing for text entailment with venses. In *Proceedings of the 2009 Text Analysis Conference (TAC'09)*, Gaithersburg, Maryland, USA, November 2009.

Edsger W Dijkstra. A note on two problems in connexion with graphs. *Numerische mathematik*, 1(1):269–271, 1959.

Rossen Dimov, Michael Feld, Dr Michael Kipp, Dr Alassane Ndiaye, and Dr Dominik Heckmann. Weka: Practical machine learning tools and techniques with Java implementations. *AI Tools Seminar*, 6(07), 2007.

Myroslava O Dzikovska, Rodney D Nielsen, Chris Brew, Claudia Leacock, Danilo Giampiccolo, Luisa Bentivogli, Peter Clark, Ido Dagan, and Hoa Trang Dang.

SemEval-2013 Task 7: The joint student response analysis and 8th recognizing textual entailment challenge. In *Proceedings of the 7th International Workshop on Semantic Evaluation (SemEval 2013), in conjunction with the Second Joint Conference on Lexical and Computational Semantcis (* SEM 2013)*, Atlanta, Georgia, USA, June 2013.

A. Echihabi, U. Hermjakob, E. Hovy, D. Marcu, E. Melz, and D. Ravichandran. Multiple-engine question answering in textmap. In *Proceedings of Text REtrieval Conference (TREC)*, Gaithersburg, Maryland, USA, November 2003.

O. Etzioni, M. Cafarella, D. Downey, A.M. Popescu, T. Shaked, S. Soderland, D.S. Weld, and A. Yates. Unsupervised named-entity extraction from the web: An experimental study. *Artificial Intelligence*, 165(1):91–134, 2005.

FaCT. http://www.cs.man.ac.uk/ horrocks/fact/. last accessed 2006-01-16.

Christiane Fellbaum. *WordNet: An Electronic Lexical Database*. MIT Press, 1998. ISBN 026206197X.

Lev Finkelstein, Evgeniy Gabrilovich, and Yossi Matias. Placing search in context: The concept revisited. In *WWW '01: Proceedings of the 10th international conference on World Wide Web*, pages 406–414, New York, NY, USA, May 2001. ACM.

Evgeniy Gabrilovich and Shaul Markovitch. Computing semantic relatedness using wikipedia-based explicit semantic analysis. In *Proceedings of the 20th international joint conference on Artifical intelligence (IJCAI 2007)*, pages 1606–1611, Hyderabad, India, January 2007.

D. Giampiccolo, B. Magnini, I. Dagan, and B. Dolan. The third PASCAL recognizing textual entailment challenge. In *Proceedings of the third ACL-PASCAL Workshop* *on Textual Entailment and Paraphrasing*, pages 1–9, Prague, Czech Republic, June 2007. Association for Computational Linguistics.

D. Giampiccolo, H.T. Dang, B. Magnini, I. Dagan, and B. Dolan. The fourth PASCAL recognizing textual entailment challenge. In *Proceedings of the TAC 2008 Workshop on Textual Entailment*, Gaithersburg, Maryland, USA, November 2008.

Fausto Giunchiglia, Pavel Shvaiko, and Mikalai Yatskevich. S-Match: an Algorithm and an Implementation of Semantic Matching. *The Semantic Web: Research and Applications: First European Semantic Web Symposium, ESWS 2004*, May 2004.

O. Glickman, I. Dagan, and M. Koppel. A lexical alignment model for probabilistic textual entailment. *Machine Learning Challenges: Evaluating Predictive Uncertainty, Visual Object Classification, and Recognising Tectual Entailment*, pages 287–298, 2006.

Birte Glimm and Ian R. Horrocks. Query answering systems in the semantic web. In Sean Bechhofer, Volker Haarslev, Carsten Lutz, and Ralf Moeller, editors, *Proceedings of Applications of Description Logics (ADL 04)*, Ulm, Germany, 2004.

Demetrios G Glinos. Recognizing Textual Entailment at RTE4 with CERES. In *Proceedings of the 2008 Text Analysis Conference (TAC'08)*, Gaithersburg, Maryland, USA, November 2008.

Thomas R Gruber. Toward principles for the design of ontologies used for knowledge sharing? *International journal of human-computer studies*, 43(5):907–928, 1995.

Thomas R Gruber et al. A translation approach to portable ontology specifications. *Knowledge acquisition*, 5(2):199–220, 1993.

V. Haarslev and R. Möller. RACER System Description. In R. Goré, A. Leitsch, and

T. Nipkow, editors, *Proceedings of International Joint Conference on Automated Reasoning (IJCAR-2001)*, pages 701–705, Sienna, Italy, 2001.

V. Haarslev, R. Moeller, and M. Wessel. Querying the semantic web with racer + nrql. In Sean Bechhofer, Volker Haarslev, Carsten Lutz, and Ralf Moeller, editors, *CEUR Workshop Proceedings of KI-2004 Workshop on Applications of Description Logics (ADL 04)*, Ulm, Germany, September 2004.

Volker Haarslev and Ralf Möller. Racer: A Core Inference Engine for the Semantic Web. In *Proceedings of the 2nd International Workshop on Evaluation of Ontology-based Tools*, Sanibel Island, Fl, USA, 2003.

Nizar Habash and Bonnie Dorr. Catvar: A database of categorial variations for english. In *Proceedings of the Machine Translation Summit (MT Summit)*, pages 471–474, New Orleans, LA, September 2003.

S. Harabagiu and A. Hickl. Methods for using textual entailment in open-domain question answering. In *Annual meeting - Association for Computational Linguistics (ACL)*, volume 44, page 905, 2006.

Sanda Harabagiu, George Miller, and Dan Moldovan. Wordnet 2-a morphologically and semantically enhanced resource. In *Proceedings of Lexicon Special Interest Group (SIGLEX)*, volume 99, pages 1–8, Maryland, USA, June 1999.

D.G. Hays. Dependency theory: A formalism and some observations. *Language*, 40 (4):511–525, 1964.

M.A. Hearst. Automatic acquisition of hyponyms from large text corpora. In *Proceedings of the 14th International Conference on Computational Linguistics*, Nantes, France, August 1992.

A. Hickl, J. Williams, J. Bensley, K. Roberts, B. Rink, and Y. Shi. Recognizing textual entailment with lcc's groundhog system. In *Proceedings of the Second PASCAL Challenges Workshop*, Venice, Italy, April 2006.

Graeme Hirst and David St-Onge. Lexical chains as representations of context for the detection and correction of malapropisms. *WordNet An electronic lexical database*, pages 305–332, April 1998.

I. Horrocks and P. Patel-Schneider. Reducing OWL entailment to description logic satisfiability. *Web Semantics: Science, Services and Agents on the World Wide Web*, 1(4):345–357, 2004.

Ian Horrocks and Peter F Patel-Schneider. Reducing owl entailment to description logic satisfiability. In *The Semantic Web-ISWC 2003*, pages 17–29. Springer, 2003.

Ian Horrocks, Peter F Patel-Schneider, and Frank Van Harmelen. From SHIQ and RDF to OWL: The making of a web ontology language. *Web semantics: Science, services and agents on the World Wide Web*, 1(1):7–26, 2003.

Ian Horrocks et al. Daml+oil: A description logic for the semantic web. *IEEE Data Engineering Bulletin*, 25(1):4–9, 2002.

Thad Hughes and Daniel Ramage. Lexical semantic relatedness with random graph walks. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing - Conference on Computational Natural Language Learning (EMNLP-CoNLL)*, pages 581–589, Prague, Czech Republic, June 2007.

Nancy Ide and Catherine Macleod. The american national corpus: A standardized resource of american english. In *Proceedings of Corpus Linguistics 2001*, volume 3, 2001.

A. Iftene. UAIC Participation at RTE4. In *Proceedings of the 2008 Text Analysis Conference (TAC'08)*, Gaithersburg, Maryland, USA, November 2008.

A. Iftene and A. Balahur-Dobrescu. Hypothesis transformation and semantic variability rules used in recognizing textual entailment. In *Proceedings of the ACL-PASCAL Workshop on Textual Entailment and Paraphrasing*, pages 125–130, Prague, Czech Republic, June 2007. Association for Computational Linguistics.

Adrian Iftene and Mihai-Alex Moruz. UAIC participation at RTE5. In *Proceedings of the 2009 Text Analysis Conference (TAC'09)*, Gaithersburg, Maryland, USA, November 2009.

K. Inui and U. Hermjakob. The 2nd international workshop on paraphrasing: Paraphrase acquisition and applications. Japan, July 2003. ACL-2003 Workshop.

Ayman Farahat Irina MaTveeva, Gina-Anne Levow and ChriSTiaan Royer. Term representation with generalized latent semantic analysis. July 2005.

Mario Jarmasz and Stan Szpakowicz. Roget‘s thesaurus and semantic similarity. In *Proceedings of Recent Advances in Natural Language Processing (RANLP 2003)*, pages 212–219, Borovets, Bulgaria, September 2003.

Jay J Jiang and David W Conrath. Semantic similarity based on corpus statistics and lexical taxonomy. In *Proceedings of International Conference on Research in Computational Linguistics*, pages 19–33, Taipei, Taiwan, August 1997.

Dan Jurafsky, James H Martin, Andrew Kehler, Keith Vander Linden, and Nigel Ward. *Speech and language processing: An introduction to natural language processing, computational linguistics, and speech recognition*, volume 2. MIT Press, 2000.

Hans Kamp and Uwe Reyle. *From discourse to logic: Introduction to modeltheoretic semantics of natural language, formal logic and discourse representation theory.* Number 42. Springer, 1993.

Paul Kingsbury and Martha Palmer. From treebank to propbank. In *Proceedings of the 3rd International Conference on Language Resources and Evaluation (LREC-2002)*, pages 1989–1993, Spain, May 2002.

Karin Kipper, Anna Korhonen, Neville Ryant, and Martha Palmer. Extending verbnet with novel verb classes. In *Proceedings of the International Conference on Language Resources and Evaluation(LREC)*, Italy, May 2006.

Ioannis Korkontzelos, Torsten Zesch, Fabio Massimo Zanzotto, and Chris Biemann. Semeval-2013 task 5: Evaluating phrasal semantics. In *Proceedings of the 7th International Workshop on Semantic Evaluation (SemEval 2013)*, Atlanta, Georgia, USA, June 2013.

Leila Kosseim, Reda Siblini, Christopher Baker, and Sabine Bergler. Using Selectional Restrictions to Query an OWL Ontology. In *International Conference on Formal Ontology in Information Systems (FOIS)*, Baltimore, Maryland, USA, November 2006.

M. Kouylekov, M. Negri, B. Magnini, and B. Coppola. Towards entailment-based question answering: ITC-irst at CLEF. *Evaluation of Multilingual and Multi-modal Information Retrieval*, pages 526–536, 2007.

Milen Kouylekov. *Recognizing textual entailment with tree edit distance: Application to question answering and information extraction.* PhD thesis, DIT-University of Trento, 2006.

R. Krestel, R. Witte, and S. Bergler. Believe it or not: Solving the TAC 2009 textual entailment tasks through an artificial believer system. In *Proceedings of the 2009 Text Analysis Conference (TAC'09)*, pages 16–17, Gaithersburg, Maryland, USA, November 2009.

Thomas K Landauer and Susan T Dumais. A solution to plato's problem: The latent semantic analysis theory of acquisition, induction, and representation of knowledge. *Psychological Review*, 104(2):211–240, 1997.

Claudia Leacock and Martin Chodorow. Combining local context and wordnet similarity for word sense identification. *WordNet: An electronic lexical database*, 49(2): 265–283, 1998.

Vladimir I. Levenshtein. Binary codes capable of correcting deletions, insertions. Technical report, and reversals. Technical Report 8, 1966.

Dekang Lin. An information-theoretic definition of similarity. In *Proceedings of the 15th international conference on Machine Learning (ICML 1998)*, volume 1, pages 296–304, Madison, WI, USA, July 1998a.

Dekang Lin. Automatic retrieval and clustering of similar words. In *Proceedings of the 17th international conference on Computational linguistics-Volume 2*, pages 768–774, Montreal, Canada, July 1998b. Association for Computational Linguistics.

Dekang Lin. Dependency-based evaluation of Minipar. *Workshop on the Evaluation of Parsing Systems*, 1998c. Granada, Spain.

Dekang Lin. Dependency-based evaluation of minipar. *Treebanks*, pages 317–329, 2003.

Dekang Lin and Patrick Pantel. DIRT SBT discovery of inference rules from text. In

Proceedings of the seventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pages 323–328. ACM, 2001.

Vanessa Lopez, Enrico Motta, Victoria Uren, and Marta Sabou. State of the art on semantic question answering. *Technical Report Knowledge Media Institute (KMI)*, 2007.

C. Macleod, R. Grishman, A. Meyers, L. Barrett, and R. Reeves. Nomlex: A lexicon of nominalizations. In *Proceedings of the 8th International Congress of the European Association for Lexicography*, pages 187–193, Belgium, August 1998.

C.D. Manning and H. Schütze. *Foundations of Statistical Natural Language Processing*. MIT Press, 1999.

EC Marsi, EJ Krahmer, WE Bosma, and M. Theune. Normalized alignment of dependency trees for detecting textual entailment. April 2006.

W. McCune. Otter 3.0 reference manual and guide. Technical report, Tech. Report ANL-94/6, Argonne National Laboratory, Argonne, IL, 1994.

Deborah L. McGuinness, Frank Van Harmelen, et al. Owl web ontology language overview. *W3C recommendation*, 10(2004-03):10, 2004.

M. Mehdad, N. Matteo, C. Elena, K. Milen, and B. Magnini. Edits: An open source framework for recognizing textual entailment. In *Proceedings of the 2009 Text Analysis Conference (TAC'09)*, Gaithersburg, Maryland, USA, November 2009.

A. Meyers, R. Reeves, C. Macleod, R. Szekeley, V. Zielinska, B. Young, and R. Grishman. The cross-breeding of dictionaries. In *Proceedings of LREC*, pages 1095–1098, Portugal, May 2004a.

Adam Meyers, Ruth Reeves, Catherine Macleod, Rachel Szekely, Veronika Zielinska, Brian Young, and Ralph Grishman. The nombank project: An interim report. In *Proceedings of the 2004 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology (HLT-NAACL '04) Workshop: Frontiers in Corpus Annotation*, pages 24–31, Boston, May 2004b.

George A Miller and Walter G Charles. Contextual correlates of semantic similarity. *Language & Cognitive Processes*, 6(1):1–28, 1991. ISSN 01690965.

Marvin Minsky. A framework for representing knowledge. In *Computation & intelligence*, pages 163–189. American Association for Artificial Intelligence, 1995.

S. Mirkin, L. Specia, N. Cancedda, I. Dagan, M. Dymetman, and I. Szpektor. Source-language entailment modeling for translating unknown terms. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP: Volume 2*, pages 791–799, Singapore, August 2009. Association for Computational Linguistics.

Shachar Mirkin, Ido Dagan, and Sebastian Padó. Assessing the role of discourse references in entailment inference. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 1209–1219, Uppsala, Sweden, July 2010. Association for Computational Linguistics.

D. Moldovan, C. Clark, S. Harabagiu, and S. Maiorano. Cogex: A logic prover for question answering. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology-Volume 1*, pages 87–93, Edmonton, May 2003. Association for Computational Linguistics.

C. Monz and M. de Rijke. Light-weight entailment checking for computational semantics. In *Proceedings of the third workshop on inference in computational semantics (ICoS-3)*, 2001.

D. Nadeau, P.D. Turney, and S. Matwin. Unsupervised named-entity recognition: Generating gazetteers and resolving ambiguity. In *19th Canadian Conference on Artificial Intelligence*, Quebec, Canada, 2006. Springer.

David Nadeau. *Semi-Supervised Named Entity Recognition: Learning to Recognize 100 Entity Types with Little Supervision*. PhD thesis, University of Ottawa, November 2007.

Preslav Nakov and Marti Hearst. Using verbs to characterize noun-noun relations. In *Artificial Intelligence: Methodology, Systems, and Applications*, pages 233–244. Springer, 2006.

M. Negri, A. Marchetti, Y. Mehdad, L. Bentivogli, and D. Giampiccolo. semeval-2012 task 8: Cross-lingual textual entailment for content synchronization. In *Proceedings of the 6th International Workshop on Semantic Evaluation (SemEval 2012)*, Montreal, Canada, June 2012.

Bahadorreza Ofoghi and John Yearwood. Learning parse-free event-based features for textual entailment recognition. In *AI 2010: Advances in Artificial Intelligence*, pages 184–193. Springer, 2011.

S. Padó, M. Galley, D. Jurafsky, and C. Manning. Robust machine translation evaluation with entailment features. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*, pages 297–305, Singapore, August 2009. Association for Computational Linguistics.

Sebastian Padó and Mirella Lapata. Dependency-Based Construction of Semantic Space Models. *Computational Linguistics*, 33(2):161–199, 2007. ISSN 08912017. doi: 10.1162/coli.2007.33.2.161.

P. Pakray, S. Bandyopadhyay, and A. Gelbukh. Lexical based two-way rte system at rte-5. In *Proceedings of the 2009 Text Analysis Conference (TAC'09)*, November 2009.

K. Papineni, S. Roukos, T. Ward, and W.J. Zhu. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting on association for computational linguistics*, pages 311–318, Philadephia, USA, July 2002. Association for Computational Linguistics.

Peter F Patel-Schneider, Patrick Hayes, Ian Horrocks, et al. Owl web ontology language semantics and abstract syntax. *W3C recommendation*, 10, 2004.

S. Patwardhan, S. Banerjee, and T. Pedersen. Using measures of semantic relatedness for word sense disambiguation. In *Proceedings of the Fourth International Conference on Intelligent Text Processing and Computational Linguistics (CICLING-2003)*, Mexico City, 2003.

Jennifer Pearson. *Terms in context*, volume 1. John Benjamins, 1998.

Karl Pearson. Mathematical contributions to the theory of evolution. –VII. on the correlation of characters not quantitatively measurable. *Philosophical Transactions of the Royal Society of London. Series A, Containing Papers of a Mathematical or Physical Character*, 195:1–405, 1900.

T. Pedersen, S. Patwardhan, and J. Michelizzi. WordNet::Similarity - Measuring the Relatedness of Concepts. In *Proceedings of the Nineteenth National Conference on Artificial Intelligence (AAAI-04)*, San Jose, California, July 2004a.

T. Pedersen, S. Patwardhan, and J. Michelizzi. WordNet::Similarity-Measuring the Relatedness of Concepts. In *Proceedings of the Nineteenth National Conference on Artificial Intelligence (AAAI-04)*, 2004b.

Pellet. http://www.mindswap.org/2003/pellet/. last accessed 2006-01-16.

D. Perez and E. Alfonseca. Application of the bleu algorithm for recognising textual entailments. In *Proceedings of the PASCAL Challenges Workshop on Recognising Textual Entailment*, pages 9–12, Southampton, UK, April 2005.

A. Perini. Detecting textual entailment with conditions on directional text relatedness scores. In *Proceedings of the 2009 Text Analysis Conference (TAC'09)*, Gaithersburg, Maryland, USA, November 2009.

Reinhard Rapp. Word Sense Discovery Based on Sense Descriptor Dissimilarity. In *Proceedings of the Ninth Machine Translation Summit (MT Summit IX)*, pages 315–322, New Orleans, Louisiana, USA, September 2003.

Han Ren, Donghong Ji, and Jing Wan. WHU at TAC 2009: A tri-categorization approach to textual entailment recognition. In *Proceedings of the 2009 Text Analysis Conference (TAC'09)*, Gaithersburg, Maryland, USA, November 2009.

P. Resnik and D. Yarowsky. A perspective on word sense disambiguation methods and their evaluation. In *Proceedings of SIGLEX (Lexicon Special Interest Group) of the ACL*, Washington, USA, April 1997.

Philip Resnik. Using information content to evaluate semantic similarity in a taxanomy. In *International Joint Conference for Artificial Intelligence (IJCAI-95)*, pages 448–453, Montreal, Quebec, Canada, August 1995.

A. Riazanov and A. Voronkov. The design and implementation of vampire. *AI communications*, 15(2):91–110, 2002.

Peter Mark Roget. *Roget's Thesaurus of English Words and Phrases*. TY Crowell Company, 1911.

Dan Roth and Mark Sammons. Semantic and logical inference model for textual entailment. In *Proceedings of the ACL-PASCAL Workshop on Textual Entailment and Paraphrasing*, pages 107–112, Prague, Czech Republic, June 2007. Association for Computational Linguistics.

Herbert Rubenstein and John B Goodenough. Contextual correlates of synonymy. *Communications of the ACM*, 8(10):627–633, 1965.

Maria Ruiz-Casado, Enrique Alfonseca, and Pablo Castells. Using context-window overlapping in synonym discovery and ontology extension. In *Proceedings of Recent Advances in Natural Language Processing (RANLP 2005)*, Borovets, Bulgaria, September 2005.

Horacio Saggion. Identifying definitions in text collections for question answering. In *Language Resources and Evaluation (LREC)*, Portugal, May 2004.

M. Sammons, V.G.V. Vydiswaran, T. Vieira, N. Johri, M.W. Chang, D. Goldwasser, V. Srikumar, G. Kundu, Y. Tu, K. Small, et al. Relation alignment for textual entailment recognition. In *Proceedings of the 2009 Text Analysis Conference (TAC'09)*, Gaithersburg, Maryland, USA, November 2009.

M. Sammons, VG Vydiswaran, and D. Roth. Ask not what textual entailment can do for you... In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 1199–1208, Sweden, July 2010. Association for Computational Linguistics.

Geoffrey Sampson. Briefly noted-english for the computer: the susanne corpus and analytic scheme. *Computational Linguistics*, 28(1):102–103, 2002.

S. Sekine. Definition, dictionaries and tagger for Extended Named Entity Hierarchy. *Actes LREC*, May 2004.

S. Settembre. Textual entailment using univariate density model and maximizing discriminant function. In *Proceedings of the ACL-PASCAL Workshop on Textual Entailment and Paraphrasing*, pages 95–100, Prague, Czech Republic, June 2007. Association for Computational Linguistics.

Arash Shaban-Nejad, Christopher J.O. Baker, Greg Butler, and Volker Haarslev. The FungalWeb Ontology: Semantic Web Challenges in Bioinformatics and Genomics. In *4th International Semantic Web Conference (ISWC)*, Lecture Notes in Computer Science 3729, pages 1063–1066, Galway, Ireland, 2004.

Arash Shaban-Nejad, Christopher Baker, Volker Haarslev, and Greg Butler. The fungalweb ontology: Semantic web challenges in bioinformatics and genomics. *The Semantic Web–ISWC 2005*, pages 1063–1066, 2005.

H. Shima, H. Kanayama, C.W. Lee, C.J. Lin, T. Mitamura, Y. Miyao, S. Shi, and K. Takeda. Overview of RITE: Recognizing inference in text. In *Proceedings of NTCIR-8 Workshop*, Tokyo, Japan, 2011.

E. Shnarch. *Lexical entailment and its extraction from Wikipedia.* PhD thesis, Bar-Ilan University, 2008.

Eyal Shnarch, Libby Barak, and Ido Dagan. Extracting lexical reference rules from wikipedia. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*, pages 450–458, Singapore, August 2009. Association for Computational Linguistics.

Reda Siblini and Leila Kosseim. Using Ontology Alignment for the TAC RTE Challenge. In *Proceedings of the 2008 Text Analysis Conference (TAC'08)*, Gaithersburg, Maryland, USA, November 2008a.

Reda Siblini and Leila Kosseim. Rodeo: Reasoning over dependencies extracted online. In *Proceedings of the The 4th Web as Corpus: Can we do better than Google?, a workshop of the Sixth International Language Resources and Evaluation (LREC'08)*, Marrakech, Morocco, June 2008b.

Reda Siblini and Leila Kosseim. Aorte for recognizing textual entailment. In *Computational Linguistics and Intelligent Text Processing: 10th International Conference, CICLing*, Mexico City, Mexico, 2009.

Reda Siblini and Leila Kosseim. Clac: Semantic relatedness of words and phrases. In *Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 2: Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013)*, pages 108–113, Atlanta, Georgia, USA, June 2013a.

Reda Siblini and Leila Kosseim. Using a weighted semantic network for lexical semantic relatedness. In *Proceedings of Recent Advances in Natural Language Processing (RANLP 2013)*, Hissar, Bulgaria, September 2013b.

B. Smith, W. Ceusters, B. Klagges, J. Kohler, A. Kumar, J. Lomax, C. Mungall, F. Neuhaus, A.L. Rector, and C. Rosse. Relations in biomedical ontologies. *Genome Biology*, 6(5), 2005.

Rion Snow, Daniel Jurafsky, and Andrew Y Ng. Learning syntactic patterns for automatic hypernym discovery. In *In Proceedings of Neural Information Processing Systems (NIPS)*, Vancouver, Canada, December 2004.

Daniel Sonntag and Bogdan Sacaleanu. Speech grammars for textual entailment patterns in multimodal question answering. In *Proceedings of the Seventh Conference on International Language Resources and Evaluation*, Valletta, Malta, May 2010. European Language Resources Association (ELRA).

Charles Spearman. The proof and measurement of association between two things. *The American Journal of Psychology*, 15(1):72–101, 1904.

Michael Strube and Simone Paolo Ponzetto. Wikirelate! computing semantic relatedness using wikipedia. In *Proceedings of the National Conference on Artificial Intelligence*, volume 21, page 1419, Boston, Massachusetts, USA, July 2006.

Fabian M Suchanek, Gjergji Kasneci, and Gerhard Weikum. Yago: a core of semantic knowledge. In *Proceedings of the 16th international conference on World Wide Web*, pages 697–706, Banff, Alberta, Canada, May 2007. ACM.

Idan Szpektor, Hristo Tanev, Ido Dagan, Bonaventura Coppola, et al. *Scaling Web-based aquisition of entailment relations*. PhD thesis, Tel Aviv University, 2005.

M. Tatu, B. Iles, J. Slavick, A. Novischi, and D. Moldovan. Cogex at the second recognizing textual entailment challenge. In *Proceedings of the Second PASCAL Challenges Workshop on Recognising Textual Entailment*, pages 104–109, Venice, Italy, April 2006.

Egidio Terra and Charles LA Clarke. Frequency estimates for statistical word similarity measures. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology (HLT-NAACL '03)*, volume 21, pages 165–172, Edmonton, Canada, May 2003. Association for Computational Linguistics.

Assaf Toledo, Sophia Katrenko, Stavroula Alexandropoulou, Heidi Klockmann, Asher Stern, Ido Dagan, and Yoad Winter. Semantic annotation for textual entailment recognition. In *Advances in Computational Intelligence*, pages 12–25. Springer, 2013.

George Tsatsaronis, Iraklis Varlamis, and Michalis Vazirgiannis. Text relatedness based on a word thesaurus. *Journal of Artificial Intelligence Research*, 37(1):1–40, 2010.

Peter Turney. Mining the Web for Synonyms: PMI-IR versus LSA on TOEFL. In *Proceedings of the Twelfth European Conference on Machine Learning (ECML-2001)*, pages 491–502, Freiburg, Germany, September 2001.

Peter D Turney, Michael L Littman, Jeffrey Bigham, and Victor Shnayder. Combining Independent Modules to Solve Multiple-choice Synonym and Analogy Problems. In *Recent Advances in Natural Language Processing (RANLP 2003)*, pages 101–110, Borovets, Bulgaria, September 2003.

Tim Van de Cruys, Stergos Afantenos, and Philippe Muller. Melodi: Semantic similarity of words and compositional phrases using latent vector weighting. In *Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 2: Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013)*, Atlanta, Georgia, USA, June 2013.

V. Varma, P. Pingali, R. Katragadda, and S. Krisha. IIT hyderabad at TAC 2008. In *Proceedings of the 2008 Text Analysis Conference (TAC'08)*, Gaithersburg, Maryland, USA, November 2008.

V. Varma, P. Bysani, V.B. Kranthi Reddy, K.K. Santosh GSK, S. Kovelamudi, N. Kiran Kumar, and N. Maganti. IIIT hyderabad at TAC (2009). In *Proceedings of the*

2009 Text Analysis Conference (TAC'09), Gaithersburg, Maryland, USA, November 2009.

E.M. Voorhees. Overview of the TREC 2001 Question Answering Track. In *Proceedings of The Tenth Text REtrieval Conference (TREC-X)*, pages 157–165, Gaithersburg, Maryland, 2001.

R. Wang and G. Neumann. A divide-and-conquer strategy for recognizing textual entailment. In *Proceedings of 2008 Text Analysis Conference (TAC'08)*, Gaithersburg, Maryland, USA, November 2008.

R. Wang, Y. Zhang, and G. Neumann. A joint syntactic-semantic representation for recognizing textual relatedness. In *Proceedings of the 2009 Text Analysis Conference (TAC'09)*, Gaithersburg, Maryland, USA, November 2009.

Christian Waterna. Hsh: Estimating semantic similarity of words and short phrases with frequency normalized distance measure. In *Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 2: Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013)*, Atlanta, Georgia, USA, June 2013.

Michael Wessel and Ralf Molle. A high performance semantic web query answering engine. In *Proceedings of the 2005 International Workshop on Description Logics (DL2005)*, Whistler, Canada, 2005.

Zhibiao Wu and Martha Palmer. Verbs semantics and lexical selection. In *Proceedings of the 32nd annual meeting on Association for Computational Linguistics*, pages 133–138, New Mexico, June 1994.

Dongqiang Yang and David MW Powers. Measuring semantic similarity in the taxonomy of wordnet. In *Proceedings of the Twenty-eighth Australasian conference on*

Computer Science, volume 38, pages 315–322, Newcastle, Australia, January 2005. Australian Computer Society, Inc.

M.A. Yatbaz. Rte4: Normalized dependency tree alignment using unsupervised n-gram word similarity score. In *Proceedings of the Fourth PASCAL Challenges Workshop on Recognizing Textual Entailment.*, Gaithersburg, Maryland, USA, November 2008.

F.M. Zanzotto, A. Moschitti, M. Pennacchiotti, and M.T. Pazienza. Learning textual entailment from examples. In *Proceedings of the Second PASCAL Challenges Workshop on Recognising Textual Entailment*, Venice, Italy, April 2006.

Appendix A

Sample OWL Generated Representation

The following is a sample text representation that was generated automatically by the approach explained in Chapter 3. The ontology is generated from the text: `Jurassic Park is a novel written by Michael Crichton and Published in 1990.`.


```
<?xml version="1.0"?>
<!DOCTYPE Ontology [
    <!ENTITY xsd "http://www.w3.org/2001/XMLSchema#" >
    <!ENTITY xml "http://www.w3.org/XML/1998/namespace" >
    <!ENTITY rdfs "http://www.w3.org/2000/01/rdf-schema#" >
    <!ENTITY rdf "http://www.w3.org/1999/02/22-rdf-syntax-ns#" >
]>
<Ontology xmlns="http://www.w3.org/2002/07/owl#"
     xml:base="file:/C:/Users/Reed/Desktop/Research/RacerPro-20-
        Preview/examples/owl/ontology6.rdf"
     xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
     xmlns:xsd="http://www.w3.org/2001/XMLSchema#"
     xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
     xmlns:xml="http://www.w3.org/XML/1998/namespace"
```

```
 ontologyIRI="file:/C:/Users/Reed/Desktop/Research/RacerPro-20-
    Preview/examples/owl/ontology6.rdf">
<Prefix name="ns0" IRI="http://text.semanticweb.org/text#"/>
<Prefix name="owl" IRI="http://www.w3.org/2002/07/owl#"/>
<Prefix name="rdf" IRI="http://www.w3.org/1999/02/22-rdf-syntax-
   ns#"/>
<Prefix name="xml" IRI="http://www.w3.org/XML/1998/namespace"/>
<Prefix name="xsd" IRI="http://www.w3.org/2001/XMLSchema#"/>
<Prefix name="rdfs" IRI="http://www.w3.org/2000/01/rdf-schema#"/
   >
<Declaration>
    <Class abbreviatedIRI="ns0:Author"/>
</Declaration>
<Declaration>
    <Class abbreviatedIRI="ns0:Communicator"/>
</Declaration>
<Declaration>
    <Class abbreviatedIRI="ns0:Compose"/>
</Declaration>
<Declaration>
    <Class abbreviatedIRI="ns0:Create"/>
</Declaration>
<Declaration>
    <Class abbreviatedIRI="ns0:Create-Verbally"/>
</Declaration>
<Declaration>
    <Class abbreviatedIRI="ns0:Fiction"/>
</Declaration>
<Declaration>
    <Class abbreviatedIRI="ns0:Fundamental-Measure"/>
</Declaration>
<Declaration>
```

```
    <Class abbreviatedIRI="ns0:Fundamental-Quantity"/>
</Declaration>
<Declaration>
    <Class abbreviatedIRI="ns0:Indite"/>
</Declaration>
<Declaration>
    <Class abbreviatedIRI="ns0:Individual"/>
</Declaration>
<Declaration>
    <Class abbreviatedIRI="ns0:Literary-Composition"/>
</Declaration>
<Declaration>
    <Class abbreviatedIRI="ns0:Literary-Work"/>
</Declaration>
<Declaration>
    <Class abbreviatedIRI="ns0:Make"/>
</Declaration>
<Declaration>
    <Class abbreviatedIRI="ns0:Novel"/>
</Declaration>
<Declaration>
    <Class abbreviatedIRI="ns0:Pen"/>
</Declaration>
<Declaration>
    <Class abbreviatedIRI="ns0:Period"/>
</Declaration>
<Declaration>
    <Class abbreviatedIRI="ns0:Person"/>
</Declaration>
<Declaration>
    <Class abbreviatedIRI="ns0:Print"/>
</Declaration>
```

```
<Declaration>
    <Class abbreviatedIRI="ns0:Produce"/>
</Declaration>
<Declaration>
    <Class abbreviatedIRI="ns0:Publish"/>
</Declaration>
<Declaration>
    <Class abbreviatedIRI="ns0:Publication"/>
</Declaration>
<Declaration>
    <Class abbreviatedIRI="ns0:Publisher"/>
</Declaration>
<Declaration>
    <Class abbreviatedIRI="ns0:Time-Period"/>
</Declaration>
<Declaration>
    <Class abbreviatedIRI="ns0:TwelveMonth"/>
</Declaration>
<Declaration>
    <Class abbreviatedIRI="ns0:Write"/>
</Declaration>
<Declaration>
    <Class abbreviatedIRI="ns0:Writer"/>
</Declaration>
<Declaration>
    <Class abbreviatedIRI="ns0:Year"/>
</Declaration>
<Declaration>
    <Class abbreviatedIRI="ns0:Yr"/>
</Declaration>
<Declaration>
    <Class abbreviatedIRI="ns0:mortal"/>
```

```
</Declaration>
<Declaration>
    <Class abbreviatedIRI="ns0:somebody"/>
</Declaration>
<Declaration>
    <Class abbreviatedIRI="ns0:someone"/>
</Declaration>
<Declaration>
    <Class abbreviatedIRI="ns0:soul"/>
</Declaration>
<Declaration>
    <ObjectProperty abbreviatedIRI="ns0:has-author"/>
</Declaration>
<Declaration>
    <ObjectProperty abbreviatedIRI="ns0:has-communicator"/>
</Declaration>
<Declaration>
    <ObjectProperty abbreviatedIRI="ns0:has-fiction"/>
</Declaration>
<Declaration>
    <ObjectProperty abbreviatedIRI="ns0:has-literary-work"/>
</Declaration>
<Declaration>
    <ObjectProperty abbreviatedIRI="ns0:has-novel"/>
</Declaration>
<Declaration>
    <ObjectProperty abbreviatedIRI="ns0:has-person"/>
</Declaration>
<Declaration>
    <ObjectProperty abbreviatedIRI="ns0:has-publication"/>
</Declaration>
<Declaration>
```

```
    <ObjectProperty abbreviatedIRI="ns0:has-publisher"/>
</Declaration>
<Declaration>
    <ObjectProperty abbreviatedIRI="ns0:has-writer"/>
</Declaration>
<Declaration>
    <ObjectProperty abbreviatedIRI="ns0:in-time-period"/>
</Declaration>
<Declaration>
    <ObjectProperty abbreviatedIRI="ns0:in-twelvemonth"/>
</Declaration>
<Declaration>
    <ObjectProperty abbreviatedIRI="ns0:in-year"/>
</Declaration>
<Declaration>
    <ObjectProperty abbreviatedIRI="ns0:in-year-for"/>
</Declaration>
<Declaration>
    <ObjectProperty abbreviatedIRI="ns0:is-communicator-for"/>
</Declaration>
<Declaration>
    <ObjectProperty abbreviatedIRI="ns0:is-fiction-for"/>
</Declaration>
<Declaration>
    <ObjectProperty abbreviatedIRI="ns0:is-literary-work-for"/>
</Declaration>
<Declaration>
    <ObjectProperty abbreviatedIRI="ns0:is-novel-for"/>
</Declaration>
<Declaration>
    <ObjectProperty abbreviatedIRI="ns0:is-person-for"/>
</Declaration>
```

```xml
<Declaration>
    <ObjectProperty abbreviatedIRI="ns0:is-publication-for"/>
</Declaration>
<Declaration>
    <ObjectProperty abbreviatedIRI="ns0:is-publisher-for"/>
</Declaration>
<Declaration>
    <ObjectProperty abbreviatedIRI="ns0:is-writer-for"/>
</Declaration>
<EquivalentClasses>
    <Class abbreviatedIRI="ns0:Author"/>
    <Class abbreviatedIRI="ns0:Writer"/>
</EquivalentClasses>
<EquivalentClasses>
    <Class abbreviatedIRI="ns0:Compose"/>
    <Class abbreviatedIRI="ns0:Write"/>
</EquivalentClasses>
<EquivalentClasses>
    <Class abbreviatedIRI="ns0:Create"/>
    <Class abbreviatedIRI="ns0:Make"/>
</EquivalentClasses>
<EquivalentClasses>
    <Class abbreviatedIRI="ns0:Fundamental-Measure"/>
    <Class abbreviatedIRI="ns0:Fundamental-Quantity"/>
</EquivalentClasses>
<EquivalentClasses>
    <Class abbreviatedIRI="ns0:Indite"/>
    <Class abbreviatedIRI="ns0:Write"/>
</EquivalentClasses>
<EquivalentClasses>
    <Class abbreviatedIRI="ns0:Individual"/>
    <Class abbreviatedIRI="ns0:Person"/>
```

```
</EquivalentClasses>
<EquivalentClasses>
    <Class abbreviatedIRI="ns0:Literary-Composition"/>
    <Class abbreviatedIRI="ns0:Literary-Work"/>
</EquivalentClasses>
<EquivalentClasses>
    <Class abbreviatedIRI="ns0:Make"/>
    <Class abbreviatedIRI="ns0:Produce"/>
</EquivalentClasses>
<EquivalentClasses>
    <Class abbreviatedIRI="ns0:Novel"/>
    <ObjectIntersectionOf>
        <Class abbreviatedIRI="ns0:Literary-Work"/>
        <ObjectSomeValuesFrom>
            <ObjectProperty abbreviatedIRI="ns0:is-novel-for"/>
            <Class abbreviatedIRI="ns0:Write"/>
        </ObjectSomeValuesFrom>
    </ObjectIntersectionOf>
</EquivalentClasses>
<EquivalentClasses>
    <Class abbreviatedIRI="ns0:Pen"/>
    <Class abbreviatedIRI="ns0:Write"/>
</EquivalentClasses>
<EquivalentClasses>
    <Class abbreviatedIRI="ns0:Period"/>
    <Class abbreviatedIRI="ns0:Time-Period"/>
</EquivalentClasses>
<EquivalentClasses>
    <Class abbreviatedIRI="ns0:Person"/>
    <Class abbreviatedIRI="ns0:mortal"/>
</EquivalentClasses>
<EquivalentClasses>
```

```xml
    <Class abbreviatedIRI="ns0:Person"/>
    <Class abbreviatedIRI="ns0:somebody"/>
</EquivalentClasses>
<EquivalentClasses>
    <Class abbreviatedIRI="ns0:Person"/>
    <Class abbreviatedIRI="ns0:someone"/>
</EquivalentClasses>
<EquivalentClasses>
    <Class abbreviatedIRI="ns0:Person"/>
    <Class abbreviatedIRI="ns0:soul"/>
</EquivalentClasses>
<EquivalentClasses>
    <Class abbreviatedIRI="ns0:Print"/>
    <Class abbreviatedIRI="ns0:Publish"/>
</EquivalentClasses>
<EquivalentClasses>
    <Class abbreviatedIRI="ns0:Publish"/>
    <ObjectIntersectionOf>
        <Class abbreviatedIRI="ns0:Make"/>
        <ObjectSomeValuesFrom>
            <ObjectProperty abbreviatedIRI="ns0:has-publication"
               />
            <Class abbreviatedIRI="ns0:Publication"/>
        </ObjectSomeValuesFrom>
    </ObjectIntersectionOf>
</EquivalentClasses>
<EquivalentClasses>
    <Class abbreviatedIRI="ns0:TwelveMonth"/>
    <Class abbreviatedIRI="ns0:Year"/>
</EquivalentClasses>
<EquivalentClasses>
    <Class abbreviatedIRI="ns0:Write"/>
```

```
    <ObjectIntersectionOf>
        <Class abbreviatedIRI="ns0:Make"/>
        <ObjectSomeValuesFrom>
            <ObjectProperty abbreviatedIRI="ns0:has-novel"/>
            <Class abbreviatedIRI="ns0:Novel"/>
        </ObjectSomeValuesFrom>
    </ObjectIntersectionOf>
</EquivalentClasses>
<EquivalentClasses>
    <Class abbreviatedIRI="ns0:Writer"/>
    <ObjectIntersectionOf>
        <Class abbreviatedIRI="ns0:Person"/>
        <ObjectSomeValuesFrom>
            <ObjectProperty abbreviatedIRI="ns0:is-writer-for"/>
            <Class abbreviatedIRI="ns0:Write"/>
        </ObjectSomeValuesFrom>
    </ObjectIntersectionOf>
</EquivalentClasses>
<EquivalentClasses>
    <Class abbreviatedIRI="ns0:Year"/>
    <Class abbreviatedIRI="ns0:Yr"/>
</EquivalentClasses>
<EquivalentClasses>
    <Class abbreviatedIRI="ns0:Year"/>
    <ObjectIntersectionOf>
        <Class abbreviatedIRI="ns0:Fundamental-Measure"/>
        <ObjectSomeValuesFrom>
            <ObjectProperty abbreviatedIRI="ns0:is-year-for"/>
            <Class abbreviatedIRI="ns0:Publish"/>
        </ObjectSomeValuesFrom>
    </ObjectIntersectionOf>
</EquivalentClasses>
```

```
<SubClassOf>
    <Class abbreviatedIRI="ns0:Communicator"/>
    <Class abbreviatedIRI="ns0:Person"/>
</SubClassOf>
<SubClassOf>
    <Class abbreviatedIRI="ns0:Create-Verbally"/>
    <Class abbreviatedIRI="ns0:Make"/>
</SubClassOf>
<SubClassOf>
    <Class abbreviatedIRI="ns0:Fiction"/>
    <Class abbreviatedIRI="ns0:Literary-Work"/>
</SubClassOf>
<SubClassOf>
    <Class abbreviatedIRI="ns0:Novel"/>
    <Class abbreviatedIRI="ns0:Fiction"/>
</SubClassOf>
<SubClassOf>
    <Class abbreviatedIRI="ns0:Publish"/>
    <Class abbreviatedIRI="ns0:Make"/>
</SubClassOf>
<SubClassOf>
    <Class abbreviatedIRI="ns0:Time-Period"/>
    <Class abbreviatedIRI="ns0:Fundamental-Quantity"/>
</SubClassOf>
<SubClassOf>
    <Class abbreviatedIRI="ns0:Write"/>
    <Class abbreviatedIRI="ns0:Create-Verbally"/>
</SubClassOf>
<SubClassOf>
    <Class abbreviatedIRI="ns0:Writer"/>
    <Class abbreviatedIRI="ns0:Communicator"/>
</SubClassOf>
```

```
<SubClassOf>
    <Class abbreviatedIRI="ns0:Year"/>
    <Class abbreviatedIRI="ns0:Time-Period"/>
</SubClassOf>
<ClassAssertion>
    <Class abbreviatedIRI="ns0:Jurassic-Park"/>
    <NamedIndividual abbreviatedIRI="ns0:Jurassic-Park-1"/>
</ClassAssertion>
<ClassAssertion>
    <Class abbreviatedIRI="ns0:Michael-Crichton"/>
    <NamedIndividual abbreviatedIRI="ns0:Michael-Crichton-1"/>
</ClassAssertion>
<ClassAssertion>
    <Class abbreviatedIRI="ns0:Publish"/>
    <NamedIndividual abbreviatedIRI="ns0:Publish-1"/>
</ClassAssertion>
<ClassAssertion>
    <Class abbreviatedIRI="ns0:Write"/>
    <NamedIndividual abbreviatedIRI="ns0:Write-1"/>
</ClassAssertion>
<ObjectPropertyAssertion>
    <ObjectProperty abbreviatedIRI="ns0:has-publication"/>
    <NamedIndividual abbreviatedIRI="ns0:Publish-1"/>
    <NamedIndividual abbreviatedIRI="ns0:Jurassic-Park-1"/>
</ObjectPropertyAssertion>
<ObjectPropertyAssertion>
    <ObjectProperty abbreviatedIRI="ns0:has-publisher"/>
    <NamedIndividual abbreviatedIRI="ns0:Publish-1"/>
    <NamedIndividual abbreviatedIRI="ns0:Michael-Crichton-1"/>
</ObjectPropertyAssertion>
<ObjectPropertyAssertion>
    <ObjectProperty abbreviatedIRI="ns0:in-year"/>
```

```
    <NamedIndividual abbreviatedIRI="ns0:Publish-1"/>
    <NamedIndividual abbreviatedIRI="ns0:1990"/>
</ObjectPropertyAssertion>
<ObjectPropertyAssertion>
    <ObjectProperty abbreviatedIRI="ns0:has-novel"/>
    <NamedIndividual abbreviatedIRI="ns0:Write-1"/>
    <NamedIndividual abbreviatedIRI="ns0:Jurassic-Park-1"/>
</ObjectPropertyAssertion>
<ObjectPropertyAssertion>
    <ObjectProperty abbreviatedIRI="ns0:has-writer"/>
    <NamedIndividual abbreviatedIRI="ns0:Write-1"/>
    <NamedIndividual abbreviatedIRI="ns0:Michael-Crichton-1"/>
</ObjectPropertyAssertion>
<EquivalentObjectProperties>
    <ObjectProperty abbreviatedIRI="ns0:has-author"/>
    <ObjectProperty abbreviatedIRI="ns0:has-novel"/>
</EquivalentObjectProperties>
<EquivalentObjectProperties>
    <ObjectProperty abbreviatedIRI="ns0:in-twelvemonth"/>
    <ObjectProperty abbreviatedIRI="ns0:in-year"/>
</EquivalentObjectProperties>
<SubObjectPropertyOf>
    <ObjectProperty abbreviatedIRI="ns0:has-communicator"/>
    <ObjectProperty abbreviatedIRI="ns0:has-person"/>
</SubObjectPropertyOf>
<SubObjectPropertyOf>
    <ObjectProperty abbreviatedIRI="ns0:has-fiction"/>
    <ObjectProperty abbreviatedIRI="ns0:has-literary-work"/>
</SubObjectPropertyOf>
<SubObjectPropertyOf>
    <ObjectProperty abbreviatedIRI="ns0:has-novel"/>
    <ObjectProperty abbreviatedIRI="ns0:has-fiction"/>
```

```
</SubObjectPropertyOf>
<SubObjectPropertyOf>
    <ObjectProperty abbreviatedIRI="ns0:has-writer"/>
    <ObjectProperty abbreviatedIRI="ns0:has-communicator"/>
</SubObjectPropertyOf>
<SubObjectPropertyOf>
    <ObjectProperty abbreviatedIRI="ns0:in-time-period"/>
    <ObjectProperty abbreviatedIRI="ns0:in-fundamental-measure"/
       >
</SubObjectPropertyOf>
<SubObjectPropertyOf>
    <ObjectProperty abbreviatedIRI="ns0:in-year"/>
    <ObjectProperty abbreviatedIRI="ns0:in-time-period"/>
</SubObjectPropertyOf>
<SubObjectPropertyOf>
    <ObjectProperty abbreviatedIRI="ns0:is-communicator-for"/>
    <ObjectProperty abbreviatedIRI="ns0:is-person-for"/>
</SubObjectPropertyOf>
<SubObjectPropertyOf>
    <ObjectProperty abbreviatedIRI="ns0:is-fiction-for"/>
    <ObjectProperty abbreviatedIRI="ns0:is-literary-work-for"/>
</SubObjectPropertyOf>
<SubObjectPropertyOf>
    <ObjectProperty abbreviatedIRI="ns0:is-novel-for"/>
    <ObjectProperty abbreviatedIRI="ns0:is-fiction-for"/>
</SubObjectPropertyOf>
<SubObjectPropertyOf>
    <ObjectProperty abbreviatedIRI="ns0:is-writer-for"/>
    <ObjectProperty abbreviatedIRI="ns0:is-communicator-for"/>
</SubObjectPropertyOf>
<InverseObjectProperties>
    <ObjectProperty abbreviatedIRI="ns0:is-novel-for"/>
```

```
    <ObjectProperty abbreviatedIRI="ns0:has-novel"/>
</InverseObjectProperties>
<InverseObjectProperties>
    <ObjectProperty abbreviatedIRI="ns0:is-publication-for"/>
    <ObjectProperty abbreviatedIRI="ns0:has-publication"/>
</InverseObjectProperties>
<InverseObjectProperties>
    <ObjectProperty abbreviatedIRI="ns0:is-publisher-for"/>
    <ObjectProperty abbreviatedIRI="ns0:has-publisher"/>
</InverseObjectProperties>
<InverseObjectProperties>
    <ObjectProperty abbreviatedIRI="ns0:is-writer-for"/>
    <ObjectProperty abbreviatedIRI="ns0:has-writer"/>
</InverseObjectProperties>
<InverseObjectProperties>
    <ObjectProperty abbreviatedIRI="ns0:in-year-for"/>
    <ObjectProperty abbreviatedIRI="ns0:in-year"/>
</InverseObjectProperties>
<ObjectPropertyDomain>
    <ObjectProperty abbreviatedIRI="ns0:has-novel"/>
    <Class abbreviatedIRI="ns0:Write"/>
</ObjectPropertyDomain>
<ObjectPropertyDomain>
    <ObjectProperty abbreviatedIRI="ns0:has-publication"/>
    <Class abbreviatedIRI="ns0:Publish"/>
</ObjectPropertyDomain>
<ObjectPropertyDomain>
    <ObjectProperty abbreviatedIRI="ns0:has-publisher"/>
    <Class abbreviatedIRI="ns0:Publish"/>
</ObjectPropertyDomain>
<ObjectPropertyDomain>
    <ObjectProperty abbreviatedIRI="ns0:has-writer"/>
```

```
        <Class abbreviatedIRI="ns0:Write"/>
    </ObjectPropertyDomain>
    <ObjectPropertyDomain>
        <ObjectProperty abbreviatedIRI="ns0:in-year"/>
        <Class abbreviatedIRI="ns0:Publish"/>
    </ObjectPropertyDomain>
    <ObjectPropertyRange>
        <ObjectProperty abbreviatedIRI="ns0:has-novel"/>
        <Class abbreviatedIRI="ns0:Novel"/>
    </ObjectPropertyRange>
    <ObjectPropertyRange>
        <ObjectProperty abbreviatedIRI="ns0:has-publication"/>
        <Class abbreviatedIRI="ns0:Publication"/>
    </ObjectPropertyRange>
    <ObjectPropertyRange>
        <ObjectProperty abbreviatedIRI="ns0:has-publisher"/>
        <Class abbreviatedIRI="ns0:Publisher"/>
    </ObjectPropertyRange>
    <ObjectPropertyRange>
        <ObjectProperty abbreviatedIRI="ns0:has-writer"/>
        <Class abbreviatedIRI="ns0:Writer"/>
    </ObjectPropertyRange>
    <ObjectPropertyRange>
        <ObjectProperty abbreviatedIRI="ns0:in-year"/>
        <Class abbreviatedIRI="ns0:Year"/>
    </ObjectPropertyRange>
</Ontology>
```