

Bar-Ilan University  
Department of Computer Science

# LEXICAL ENTAILMENT AND ITS EXTRACTION FROM WIKIPEDIA

by

Eyal Shnarch

Advisor: Dr. Ido Dagan

Submitted in partial fulfillment of the requirements for the Master's degree  
in the department of Computer Science

Ramat-Gan, Israel  
July 2008 Tamuz 5768  
Copyright 2008

## Abstract

This work investigates the lexical entailment relation and develops a Wikipedia-based resource of lexical entailment rules. Lexical entailment is a semantic relation that holds between lexical elements when the meaning of one element can be inferred from the meaning of the other. This relation can be represented as a rule in which the left hand side (LHS) entails the right hand side (RHS), denoted by  $LHS \Rightarrow RHS$ . This relation is very useful in many Natural Language Processing (NLP) applications. For instance, a Question Answering system may use the lexical entailment rule *cruiser*  $\Rightarrow$  *ship* in order to find the answer to the question “*What is the name of the Russian ship that sunk at Port Arthur?*” in the sentence “*Russian cruiser Pallada sunk at Port Arthur*”. Even though NLP systems apply lexical semantic inference, there is no common definition for this relation and no resource was built as dedicated to include lexical entailment rules.

We suggest deriving the missing definition from the framework of Textual Entailment, a recent paradigm for semantic inference. We present two definitions which bridge the gap between textual entailment and lexical entailment. These definitions help understanding the intended meaning of the lexical entailment relation and deciding which rules should be included in a lexical entailment rule base. We utilize these definitions to investigate the utility of several state-of-the-art lexical semantic resources as potential sources for lexical entailment rules. In that process we reveal the strengths and weaknesses of these resources, with respect to the task of recognizing lexical entailment, and the different types of the lexical entailment relations that each resource covers.

A second contribution of this work is the development of a large-scale lexical entailment rule base, the first one designed to contain lexical entailment rules, which was extracted from Wikipedia. We present extraction methods geared to cover the broad range of the lexical entailment relation and evaluate them under this target criterion. We conduct both an internal evaluation, by comparing results to human judgments, and an external evaluation, by using our rule base within

a real NLP task. By filtering rules, according to the type of method which extracted them, one can choose different recall-precision tradeoffs varying from a precision of 0.87 for almost 2 million rules up to 8 million rules with a precision of 0.66. On the text categorization evaluation our resource performs better than previous automatically-created resources, and performs comparably to WordNet, a lexicon in which relations between terms were manually crafted by experts, even though the rules in our resource were automatically extracted from texts written for human consumption.

# Acknowledgments

I would like to thank my advisor, Ido Dagan, for his guidance throughout the way. Through long discussions and by his concern in all details of my work, he exposed me to the fascinating subjects of our research field and taught me its methodology. From day one he let me take a part in our lab projects which helped me develop scientific skills.

I would also like to thank my colleagues and friends. Idan Szpektor, Roy Bar Haim and Shachar Mirkin for the joint work, a lot of help and support, guidance and many many valuable advices. Also, I wish to thank Libby Barak for providing me with her text categorization system used in this work for one of the evaluations and mainly for being a friend.

Finally I thank my parents for their guidance and moral support which help me make the decision of following my interest and fully dedicate my time to research. Your mentoring and education took a vital role in my accomplishments.

# Contents

<b>1</b>	<b>Introduction</b>	<b>6</b>
<b>2</b>	<b>Background</b>	<b>10</b>
2.1	Types of Lexical Semantic Relations . . . . .	10
2.2	Usage of Lexical Semantic Relations . . . . .	12
2.3	Resources for Lexical Semantic Relations . . . . .	13
2.3.1	Manually Constructed Lexical Resources . . . . .	13
2.3.2	Automatic Learning of Lexical Relations . . . . .	14
<b>3</b>	<b>Lexical Entailment Definitions</b>	<b>20</b>
3.1	Entailment of Sub-sentential Hypotheses . . . . .	21
3.2	Entailment between Lexical Elements . . . . .	22
<b>4</b>	<b>Learning Lexical Entailment Rules from Wikipedia</b>	<b>24</b>
4.1	Extraction Methods . . . . .	24
4.2	Supervised Classification of Rules . . . . .	30
<b>5</b>	<b>Evaluation and Results</b>	<b>33</b>
5.1	Direct Rule-level Evaluation . . . . .	33
5.1.1	Extraction Methods . . . . .	33
5.1.2	Error Analysis . . . . .	35
5.1.3	Supervised Learning Classification . . . . .	38
5.2	Indirect Evaluation within a Text Categorization Task . . . . .	39
<b>6</b>	<b>Comparative Evaluation of Lexical Rule bases</b>	<b>44</b>
6.1	Evaluation Methodology . . . . .	45

6.2	Dataset and Annotation . . . . .	47
6.3	Lexical-Semantic Resources . . . . .	48
6.4	Comparison Results . . . . .	50
<b>7</b>	<b>Conclusions and Future Work</b>	<b>55</b>

# List of Figures

1.1	An example for compositional inference process . . . . .	8
4.1	Few extracted rules from a Wikipedia article . . . . .	29
5.1	Error analysis: type and origin of incorrect rules . . . . .	37
6.1	Comparative evaluation methodology flow chart . . . . .	46

# List of Tables

2.1	Commonly used patterns for hyponym extraction . . . . .	16
4.1	Examples of the different extraction methods . . . . .	27
5.1	Direct rule based evaluation . . . . .	34
5.2	Various types of lexical entailment relations . . . . .	36
5.3	Supervised classification results . . . . .	39
5.4	Unsupervised Text Categorization results . . . . .	41
5.5	Expansions from Wikipedia that assist Text Classification . . . . .	43
6.1	Comparison between lexical resources . . . . .	51
6.2	Rules derived from lexical semantic resources . . . . .	54
7.1	Natural Types . . . . .	58

# Chapter 1

## Introduction

Currently most Web search engines implement a *keyword based search*. Given a user's query, *The Beatles*, for instance, the search engine looks for Web pages that contain that exact string. While this method may look sufficient for the common user as there are plenty of Web pages containing the name of the famous band, this may not suffice for a more sophisticated user with more complex information needs. There are texts containing relevant information that will not be retrieved that way. For instance, it would be helpful to know that the album "Abbey Road" *refers to* the user's query and therefore is relevant for it. A search engine that would be able to recognize the semantic relation between these two terms would find more information relevant to "The Beatles". This known technic is called query expansion and is a step towards *semantic search*, a search that analyses the meaning of the information need rather than just looks for the exact query of the user.

Recognizing such semantic relations between terms is important not just for search, but rather for most Natural Language Processing (NLP) applications. A Question Answering (QA) system may encounter the following question: "*Which British luxury car was the best selling of 2007?*". The answer may appear in a sentence like: "*Rolls Royce was the best selling of its class in 2007*". Note that the question term "British luxury car" does not appear in this answer sentence at all. A QA system that would not be able to recognize that "Rolls Royce" *implies the meaning* of "British luxury car" would not be able to identify the above sentence

as the answer for the given question.

Relations between lexical terms, such as *refers to* and *implies the meaning*, are modeled by *Lexical Entailment* which is part of a generic framework for semantic inference called *Textual Entailment*. Textual entailment is a directional relation between two text fragments, a text  $t$  and a textual statement (hypothesis)  $h$ . We say that  $t$  entails  $h$  if the meaning of  $h$  can be inferred from the meaning of  $t$ . For instance, from the following sentence: “*All tickets for the Beatles’ concert in Liverpool were sold*” we can infer that “*The Beatles gave a concert in Liverpool*”. An entailment relation between specific language expressions may be represented as a rule whose left hand side (LHS) entails its right hand side (RHS), denoted  $LHS \Rightarrow RHS$ . The sides of a rule may be templates with variables, as in  $X$ ’s concert in  $Y \Rightarrow X$  gave a concert in  $Y$ , or lexical terms, as in *Rolls Royce*  $\Rightarrow$  *British luxury car*.

As elaborated in Chapter 3, the textual entailment relation is defined in terms of truth values, assuming that  $h$  is a complete sentence (proposition). However, there are cases in which entailment inference is applied to the sub-sentential level. First, in certain applications the hypotheses are often sub-sentential. Most notably, search queries in Information Retrieval (IR), which play the hypothesis role from an entailment perspective, typically consist of a single term (e.g. *The Beatles*, *French cars drawbacks*). A second case can be found in many inference systems which decompose the hypothesis sub-sentential parts and use a compositional process to infer each part of it from the text (Giampiccolo et al., 2007). In the example in Figure 1.1 three steps are taken in order to infer  $h$  from  $t$ : in the first step a syntactic transformation is applied, the second step uses a lexical-syntactic entailment rule and the last one utilizes a lexical entailment rule.

In both cases such sub-sentential hypotheses (*French cars drawbacks* or *Romania*) cannot be regarded naturally in terms of truth values and therefore do not fit well within the scope of the textual entailment definition. To address this issue, one contribution of our work is to bridge the theoretical gap between textual entailment and entailment relation at the sub-sentential level, termed *Lexical Entailment*. The proposed definition is a precondition for characterizing what kinds of relations between terms a knowledge base of lexical entailment rules should hold. We then leverage the new definition to analyze the utility of several state-of-

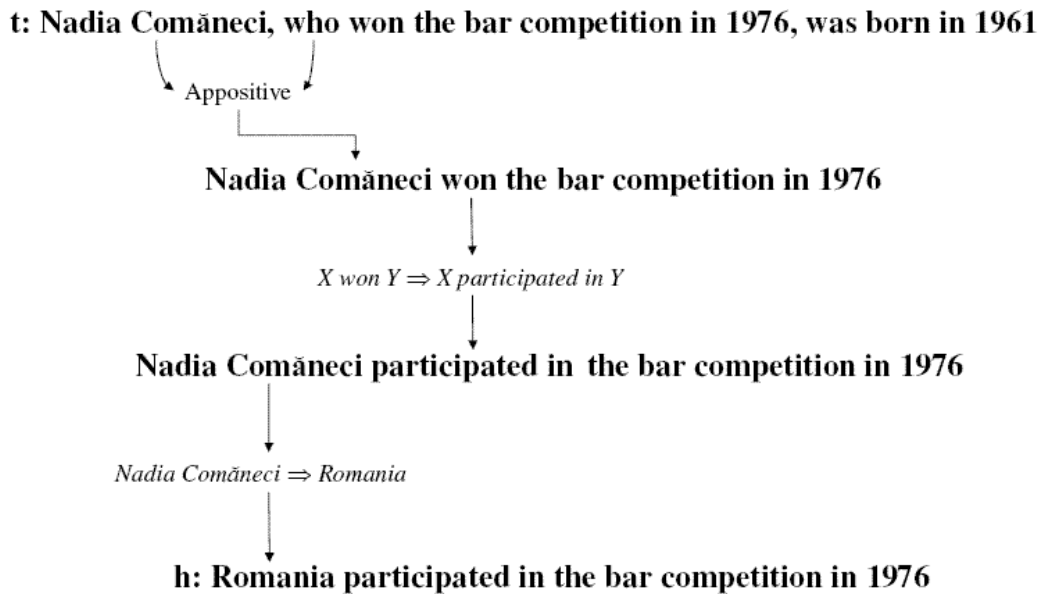


Figure 1.1: An example for compositional inference process. The last step utilizes lexical entailment rule.

the-art lexical semantic resources as potential sources for lexical entailment rules. The second contribution of this work is to add a new resource for the pool of lexical entailment resources. We develop a first rule base which is dedicated to hold rules that aim to cover the broad spectrum of lexical entailment as needed for NLP applications. No other lexical resource was built specifically to cover this important relation, and thus when a system needs to apply inference at the sub-sentential level it uses portions of existing lexical semantic resources that were not designed specifically for entailment modeling.

In order to recognize that a meaning of term can be implied from another term, inference systems need background knowledge of entailment relations between terms. Such knowledge can be represented as a list of entailment rules between terms. In order to be effective such a rule base must be very large and therefore an automatic method for constructing it is needed. We suggest utilizing definition sentences for that purpose. Definitions were recognized as an important source for knowledge about relations between terms (e.g. (Chodorow et al., 1985; Moldovan and Rus, 2001)), and we suggest to explore their potential for our goal of deriving a large lexical entailment rule base.

Definitions can be found either in dictionaries or in encyclopedias. Dictionaries describe the common language, which does not change too often. Encyclopedias, on the other hand, contain knowledge from various domains, including proper names, events and technical terms that do not appear in dictionaries. These types of concepts are updated and augmented much faster than general language concepts in a dictionary, therefore encyclopedias grow rapidly. On the other hand, encyclopedias do not cover all terms in the language and mainly contains noun phrases (dictionaries also cover verbs, adjectives and more). Therefore dictionaries and encyclopedias complement each other with respect to the scope of knowledge they cover.

We chose to use an encyclopedia as our source of definitions, from which lexical entailment rules will be derived, and specifically chose Wikipedia <sup>1</sup> since it covers a vast range of domains and is constantly growing. We focused on developing generic methods for rule extraction from a general Web knowledge base, thus aiming not to use Wikipedia-specific attributes.

Following a background section about the usage and definitions of lexical semantic relations and methods for recognizing them in Chapter 2, we set the needed definitions for entailment at the sub-sentential level in Chapter 3. Chapters 4 and 5 present the construction and evaluation of the rule base built for lexical entailment. We extracted a very large rule base from Wikipedia and evaluated its rules both directly, comparing them to human annotated gold standard, and indirectly within a text categorization application. In Chapter 6 we compare the utility of several lexical semantic resources (including ours) in an information retrieval setting<sup>2</sup>. During our work on Wikipedia we recognized few promising directions off the main line of this work. We summarize our initial findings and describe briefly the work we have done in these directions, as well as conclusions and future work, in Chapter 7.

---

<sup>1</sup>[www.wikipedia.org/](http://www.wikipedia.org/)

<sup>2</sup>Chapters 3 and 6 cover joint work with Shachar Mirkin.

# Chapter 2

## Background

In this chapter we introduce various types of lexical semantic relations that are valuable in natural language processing and their utility for applications. We also explore various methods which learn instances of such relations, and aiming to acquire knowledge bases of lexical semantic relations.

### 2.1 Types of Lexical Semantic Relations

#### Linguistic Semantic Relations

Several linguistic semantic relations between terms were recognized as relevant for natural language processing (Miller et al., 1990):

- **Synonym** - A term which has the same, or at least highly similar, meaning as another term. For instance, *automobile* is a synonym of *car*. This relation is most important since synonyms can be used interchangeably.
- **Hypernym** - A term  $w$  is a hypernym of another term  $u$  if  $w$  has an extensive meaning that forms a category under which  $u$  and other more specific terms fall. For instance, *vehicle* is a hypernym of *car*.
- **Hyponym** - A term that is more specific than another term (the opposite direction of hypernym), For instance, *taxi*, *limousine*, *sports car* and (Ford's) *Model T* are all hyponyms of *car*.

If  $u$  is a hyponym of  $w$  (and  $w$  is a hypernym of  $u$ ) it means that  $u$  is a kind of  $w$  and therefore these relations are also called the *IS-A* relation. In this relation the hypernym  $w$  can often replace its hyponyms while preserving and generalizing the meaning. For instance, replacing *limousine* with its hypernym *car* in the sentence “*The movie star stepped out of the limousine*” still preserves the meaning of the sentence, even though some details are lost.

- Meronym - A term that names a part of a larger whole or a member of a group. *Petrol engine* and *wheel* are meronyms of *car* and *Israel* is a meronym of *Middle East*.
- Derivation - The process whereby new terms are formed from existing terms or bases by affixation, For instance, *footballer* is derived from *football*.

The last two relations are also important for inference as they indicate that different terms are semantically related, but these relations do not necessarily enable simple replacement. If one visited *Israel*, we know he or she also visited the *Middle East*, but if an object has a *wheel* it does not necessarily mean that it is a *car*.

### **Distributional Similarity**

Another paradigm which presents computational methods to measure semantic relatedness (also called association) between terms is distributional similarity (e.g. (Lin, 1998)). Such methods do not search for known linguistic relations but are satisfied with rather loose semantic relationships which are identified between terms that appear in similar contexts. This paradigm is inspired by Harris’ distributional hypothesis (Harris, 1968), which states that semantically similar terms tend to appear in similar contexts (see 2.3.2 for details).

### **Substitutable Lexical Entailment**

(Geffet and Dagan, 2005) made the first step towards defining lexical entailment by defining substitutable lexical entailment between terms  $w$  and  $u$  if (i) the meaning of a possible sense of  $w$  implies a possible sense of  $u$  and (ii)  $w$  can substitute

$u$  in some naturally occurring sentence, such that the meaning of the modified sentence would entail the meaning of the original one. This definition only covers the substitutable case of lexical entailment. We are interested in a broader definition that will capture also non-substitutable cases such as *Crime and Punishment*  $\Rightarrow$  *Dostoevsky*.

## 2.2 Usage of Lexical Semantic Relations

The knowledge of lexical semantic relations is often used in two types of inferences. The first is making similarity-based generalizations for smoothing term co-occurrence probabilities, in applications such as language modeling and disambiguation. For instance, in supervised text classification, Scott and Matwin (1998) grouped terms in a document according to their hypernyms in order to gain more statistics and emphasize relevant concepts. Thus in a document mentioning *limousine*, *Model T* and other car models or types, these terms are generalized to their common hypernym *car*. The value of the new feature is the sum of the values of its hyponyms, making it a stronger feature and aiding a learner to identify that the topic of the document is *Cars*. Another kind of smoothing is achieved when test documents contain terms different from those of the training set and thus are not classified. By generalizing terms on both the training and test sets the test documents may have common features with the training documents and may be classified appropriately.

The second usage of lexical semantic relations is in term expansion, a commonly used technique for enhancing the recall of natural language processing systems by coping with lexical variability. Few examples are (Voorhees, 1994) for information retrieval (IR), (Harabagiu et al., 2000; Hovy et al., 2001; Moldovan and Rus, 2001) in question answering (QA) and (de Buenaga Rodríguez et al., 1997) in text classification (TC). Given a query in IR or a question in QA, terms of similar meaning are added to the query, thus expanding the search for texts that are relevant for the original request. For instance, consider the query *Water Pollution* which we can expand to include also *lake pollution* or *river pollution* and thus find more relevant documents and increase recall.

## 2.3 Resources for Lexical Semantic Relations

A knowledge base of lexical relations between terms can be assembled manually by linguistic experts or can be constructed automatically (or semi-automatically), involving some learning algorithm. Clearly, the former manual methods are tedious, costly and time consuming while the latter methods are less accurate.

### 2.3.1 Manually Constructed Lexical Resources

#### WordNet

WordNet (Fellbaum, 1998)<sup>1</sup>, an electronic lexical database, is considered to be the most important resource available to researchers in computational linguistics, text analysis, and many related areas. Its design is inspired by current psycholinguistic and computational theories of human lexical memory. English nouns, verbs, adjectives, and adverbs are organized into synonym sets, called *synsets*, each representing one underlying lexicalized concept. For instance, *car*, *auto*, *automobile* and *motorcar* are gathered under one synset. Each synset is provided with a dictionary gloss defining it and often including few examples of its usage within a sentence. Different semantic relations link the synsets, including hypernyms/hyponyms, meronyms and derivations, while all terms inside a synset are considered synonyms of each other. WordNet is an ongoing project developed over two decades at the Cognitive Science Laboratory at Princeton university. Recently the third version was published which contains more than 155,000 English terms. Other projects around the world developed versions of WordNet in 30 other languages<sup>2</sup>.

#### Nomlex

Nomlex (Macleod et al., 1998)<sup>3</sup> (nominalization lexicon) is a dictionary of English nominalizations (forming a noun from a verb or an adjective) developed at New York University. Nomlex holds information about nominalization forms of verbs and how to map their arguments when transforming from one lexical form to an-

---

<sup>1</sup><http://wordnet.princeton.edu>

<sup>2</sup>[www.globalwordnet.org/gwa/wordnet\\_table.htm](http://www.globalwordnet.org/gwa/wordnet_table.htm)

<sup>3</sup>[www.nlp.cs.nyu.edu/nomlex](http://www.nlp.cs.nyu.edu/nomlex)

other. Using Nomlex one can transform a verbal phrase like “*Microsoft acquired Yahoo*” to the nominal phrase “*the acquisition of Yahoo by Microsoft*”. Nomlex contains 1025 distinct entries.

### **CatVar**

CatVar (Habash and Dorr, 2003)<sup>4</sup> (Categorical Variation) is a database of clusters of uninflected words (lexemes) and their categorial, i.e. part-of-speech (POS), variants. For example, the cluster of *developing* is *develop* (verb), *developer* and *development* (nouns), *developed* (noun and adjective) and *developing* (adjective). The database was developed for English using a combination of resources and algorithms. In its second release, the database includes 63,146 clusters of 109,807 words.

The motivation for constructing both resources is a computational linguistic one and they were made for computer consumption. The disadvantages of manually constructed resources (apart of the large amount of required time and cost) are (i) inconsistency in classification since they are constructed by many lexicographers and are edited every few years, (ii) bias in coverage - some concepts are better covered than others and (iii) limited coverage, especially for infrequent or domain-specific words. Additionally, WordNet’s relations, Nomlex nominalizations and CatVar POS variations do not cover the entire spectrum of lexical entailment relations.

## **2.3.2 Automatic Learning of Lexical Relations**

### **Distributional Similarity**

As mentioned above, distributional similarity between terms is based on Harris’ Distributional Hypothesis, suggesting that the semantic similarity of meaning of terms can be predicted from their distributional similarity (similarity of the contexts in which they appear). For example *company*, *firm* and *government* co-occur in many common contexts. The context of a term *w* consist of other terms which co-occur with it. For instance, *announced* and *spokesman of* are typical contexts

---

<sup>4</sup><http://clipdemos.umiacs.umd.edu/catvar>

for the above words. All contexts of a word  $w$  construct the feature vector which represent it. The value of each feature  $f$  is determined by some weight function  $weight(w,f)$ , which quantifies the degree of statistical association between the feature and the corresponding term. Typical feature weighting functions are the log of the frequency of term-feature co-occurrence, the conditional probability of the feature given the term and Point-wise Mutual Information (PMI):

$$PMI(w, f) = \log_2 \frac{P(w, f)}{P(w)P(f)}$$

Probably the most widely used association weight function is PMI. It calculates the ratio between the probability of finding  $w$  and  $f$  together and the probability of finding them together given that they are independent of each other.

Given two terms represented as feature vectors, the distributional similarity between them is defined by some vector similarity metric, such as cosine, weighted Jaccard and various information theoretic measures (see (Dagan, 2000; Weeds et al., 2004)). The widely used similarity measure from (Lin, 1998) measures the similarity between two terms  $w$  and  $v$  by the ratio between the amount of information needed to state their commonality (common feature), and the information needed to fully describe each of them:

$$sim_{Lin}(w, v) = \frac{\sum_{f \in F(w) \cap F(v)} PMI(w, f) + PMI(v, f)}{\sum_{f \in F(w)} PMI(w, f) + \sum_{f \in F(v)} PMI(v, f)}$$

where  $F(w)$  and  $F(v)$  are features of the terms with a positive PMI value (active features). As two terms have more overlapping features, the numerator is closer to the denominator, thus the measure assigns a higher score for terms that have many common contexts. But as can be seen from the example in the beginning of this section, the loose semantic relation found by distributional similarity methods may result in non-entailing associations such as *company* and *government*. (Geffet and Dagan, 2004) suggest to reduce this problem by recalculating features weights and promoting those features that characterize many of the terms that are most similar to the term meaning.

Distributional similarity methods tend to find synonymy relations and co-hyponym terms (terms which have a common ancestor in the hypernym hierarchy,

such as *orthopedist*, *pediatrician* and *surgeon*). The disadvantage of distributional similarity, with respect to lexical entailment, is its low precision due to the loose unspecified relation which it finds, often resulting in association of terms from a common topic, such as *doctor* and *nurse*, which do not necessarily have a direct semantic relation.

### Indicative Text Patterns

A fluent reader of English can infer from a phrase like “*aquatic plants such as hydrilla*” that *hydrilla* is an *aquatic plant* even if he or she has never encountered this term before. We can do this since particular patterns indicate a semantic relation between their terms. The pattern used in the above example is  $NP_1$  *such as*  $NP_2$  (NP stands for noun phrase), indicating that  $NP_1$  is a hypernym of  $NP_2$ . As can be seen in the first sentence of this paragraph, which includes the pattern  $NP_1$  *like*  $NP_2$ , natural texts often define relations between terms even though the author has not deliberately intended to define them. Therefore such patterns can be used to extract semantic relations from arbitrary texts.

(Hearst, 1992) was the first to use lexico-syntactic patterns for automatic extraction of hyponyms from a large corpus. She used a bootstrapping algorithm, starting with three manually crafted patterns to discover three more patterns from a raw text corpus. (Pantel et al., 2004) extended her algorithm for very large (terascale) corpora. Table 2.1 presents the commonly used patterns for hyponym identification from these two works. (Berland and Charniak, 1999) applied a method based on this technique to extract meronym relations, but found that patterns are less commonly used for that semantic relation. Since synonyms rarely co-occur within short patterns in texts, pattern-based methods are less suitable for their extraction.

$NP_1$ such as $NP_2$	Such $NP_1$ as $NP_2$	$NP_1$ or other $NP_2$
$NP_1$ and other $NP_2$	$NP_1$ including $NP_2$	$NP_1$ especially $NP_2$
$NP_1$ like $NP_2$	$NP_1$ <i>adverb</i> known as $NP_2$	
$NP_1$ a/an $NP_2$	$NP_1$ is a/an $NP_2$	$NP_1$ are $NP_2$

Table 2.1: Commonly used patterns for hyponym extraction

Patterns were also used to acquire factual instances of concrete semantic relations for information extraction (IE) and question answering, such as *Author-Title*, *Person-Birthday*, *Person-Invention*, and *Film-Actor* (e.g. (Brin, 1998; Ravichandran and Hovy, 2002)).

Several works automatically learn patterns. (Riloff and Jones, 1999) use a bootstrapping algorithm, starting from manually provided seeds. They apply a filter mechanism between the algorithm's cycles to limit the deterioration of performance, which may be caused by adding wrong instances to the seeds set. They managed to identify 100-200 new members for the evaluated categories with precision of 50-75%. (Davidov and Rappoport, 2006) do not require any manually provided seeds. Instead they identify symmetric patterns in short sequences of frequent and infrequent words. Words in such patterns are assumed to be semantically similar. *from NP<sub>1</sub> to NP<sub>2</sub>* and *neither NP<sub>1</sub> nor NP<sub>2</sub>* are examples of new patterns they found. (Snow et al., 2005) do not search for specific highly indicative patterns, but rather they use all syntactic patterns as features in a hypernym classifier. They collect noun pairs from corpora and represent a pair by all syntactic patterns in which it occurs. Labeled examples for a supervised hypernym classifier were automatically assembled using WordNet. As positive examples they take all pairs in which one noun is an ancestor of the other in the WordNet hypernyms hierarchy, while negative examples are taken as pairs in which neither noun is an ancestor of the other.

Indicative patterns provide an accurate method for extracting relations. Patterns also provide the directionality of the relation (for instance, we can know which NP is the hypernym and which is its hyponym). The disadvantage of the pattern method is its low coverage. Not all knowledge falls into patterns, and many related pairs may simply not occur in the same sentence.

Finally, (Mirkin et al., 2006) use supervised learning to exploit the complementary information provided by pattern-based and distributional similarity methods to identify lexical entailment.

### **Definitions and Wikipedia**

Definitions have long been identified as a valuable source for semantic relations between words, as they describe words in terms of other words (Ide and Jean,

1993). Definitions typically describe the defined concept by giving a genus term which is usually a hypernym of it. For example, the definition of *car* from *Webster's Dictionary* is: a vehicle moving on wheels. *Vehicle* serves as the genus term (hyponym of car), while *moving on wheels* differentiates cars from other hyponyms of vehicle (e.g. tanker). Many works on extracting information from machine readable dictionaries (MRD) have tried to make this implicit information available for use by programs. (Chodorow et al., 1985) observed that the genus term is typically the head of the defining phrase and suggested simple heuristics to find it. Other methods use a specialized parser or a set of regular expressions tuned to a particular dictionary (Nichols et al., 2005; Wilks et al., 1996). (Moldovan and Rus, 2001) parse the definitions in WordNet and transform them into logical form representation in which arguments are explicitly connected to their predicates. They used this representation to incorporate world knowledge into a question answering system.

We use Wikipedia as our source of definitions for learning lexical entailment relations. Wikipedia is a collaboratively written online encyclopedia and is therefore very dynamic, constantly growing and covers a vast range of domains. Today Wikipedia is available in more than 250 languages. The Hebrew version contains almost 80,000 articles and it is ranked 26<sup>th</sup> in size amongst all languages. The English version is the largest with almost 2.5 million articles and 40,000-50,000 new articles added per month<sup>5</sup>. (Giles, 2005) showed that the quality of Wikipedia articles is comparable to those of the Internet encyclopedia Britannica, making it the largest reliable encyclopedia. Wikipedia, in contrast to WordNet and Nomlex, is a resource that was built for human consumption to answer human needs. We thus utilize an existing knowledge base rather than relying on a resource that was built manually specifically for natural language tasks.

Several previous works utilized Wikipedia to build an ontology. (Ponzetto and Strube, 2007) identify the subsumption (IS-A) relation between Wikipedia category tags using basic syntactic methods, connectivity methods and pattern-based methods. (Suchanek et al., 2007) use these category tags to extract entities and facts from Wikipedia which are then unified with WordNet by rule-based and heuristic methods. They predefine a fixed set of 14 relations which they identify

---

<sup>5</sup>Growth statistics are from the beginning of 2008.

from the WordNet hyponym hierarchy, redirect links in Wikipedia and patterns in category tags. The lexical entailment relation we address subsumes most relations found in these works, while our extractions are not limited to a fixed set of predefined relations. While these works utilize only the category tags, we avoid using Wikipedia specific features and learn lexical entailment relations from articles' texts.

Other works examine Wikipedia texts, rather than just its structural features. (Kazama and Torisawa, 2007) explore the first sentence of an article and identify the first noun phrase following a *be* verb as a label for the article title. We reproduce this part of their work as one of our baselines. (Toral and Muñoz, 2007) use all nouns in the first sentence. (Gabrilovich and Markovitch, 2007) utilize Wikipedia-based concepts as the basis for a high-dimensional meaning representation space for texts. Both works do not aim to construct a rule base of relations and use these extractions as a first step for other purposes (gazetteers for named entity recognition and text relatedness respectively). We also explore the first sentence and use all nouns in it, but we consider a syntactic analysis of the sentence rather than taking all nouns as equally useful.

## Chapter 3

# Lexical Entailment Definitions

Textual entailment emerged in recent years as a modeling paradigm for applied semantic inference largely through the series of the Recognizing Textual Entailment (RTE) challenges (Giampiccolo et al., 2007). While providing a generic application-independent framework, it captures the needs of a broad range of text understanding applications such as Question Answering (QA), Information Extraction (IE) and Information Retrieval (IR).

Textual entailment is a directional relation between two texts fragments, a text  $t$  and a textual statement (hypothesis)  $h$ . We say that  $t$  entails  $h$ , denoted  $t \Rightarrow h$ , if humans reading  $t$  will infer that  $h$  is most likely true (Dagan et al., 2005). For instance, from the following text fragment: “*Sunscreen is used to protect from getting sunburned*” we can infer that “*Sunscreen prevents sunburns*”. As opposed to the linguistic definition of entailment, which requires that  $h$  will be true in every circumstance (possible world) in which  $t$  is true, this applied definition is based on human judgment and only requires that entailment will most likely hold. Human judgment is a common judgment method in natural language processing tasks and being less strict than the linguistic definition makes the application of this definition feasible in real world situations. Based on this definition, if a text understanding system recognizes that  $t$  entails  $h$  it can infer the meaning of  $h$  from  $t$ . (Dagan et al., 2005) describe systems used for the RTE challenge that reduce semantic inference needs to the textual entailment task.

While the textual entailment relation is defined in terms of truth values, sub-

sentential levels are not regarded in such terms, thus making the current definition inadequate for them. We propose extensions for the textual entailment definition in order to cover entailment at the sub-sentential level.

### 3.1 Entailment of Sub-sentential Hypotheses

We first seek a definition that would capture the entailment relationship between a complete text and a sub-sentential hypothesis. Our first step in this direction was taken in (Glickman et al., 2006) where we defined *lexical reference* from a text to a term as *an explicit or implied reference from a set of terms in the text to a possible meaning of the term*. We used this definition to manually annotate a data set of pairs of sentences and terms from the Recognizing Textual Entailment challenge. Achieving a substantial agreement (Landis and Koch, 1997) between our annotators reinforced our confidence in the new definition. We also observed that it is typically a necessary, but not sufficient, condition for textual entailment between  $t$  and  $h$  that all non-compositional terms in the hypothesis  $h$  would be lexically referenced in the given text  $t$ . For example, in order to infer from a text the hypothesis “*a dog bit a man*”, it is a necessary that the concepts of *dog*, *bite* and *man* would be referenced by the text, either directly or in an implied manner. This finding suggests that the lexical reference definition is consistent with the original definition of textual entailment for sentential hypotheses and can thus model compositional entailment inferences for sub-parts of hypotheses. Of course, for proper (sentential) entailment to hold, it is further needed that the right relationships would hold between these concepts<sup>1</sup>.

A slight adaptation of this definition is suitable to capture the notion of entailment for sub-sentential hypotheses, obtaining the following definition:

**Definition 1** *A sub-sentential hypothesis  $h$  is entailed by a text  $t$  if there is an explicit or implied reference in  $t$  to a possible meaning of  $h$ .*

For example, the sentence “*crude steel output is likely to fall in 2000*” entails the sub-sentential hypothesis “*production*”.

---

<sup>1</sup>Quoting the known journalism saying “*Dog bites man*” are not news, but “*Man bites dog*” are.

## 3.2 Entailment between Lexical Elements

In the majority of cases, the reference to an “atomic” (non-compositional) lexical element  $e$  in  $h$  stems from a particular lexical element  $e'$  in  $t$ . In the example above, it is the word *output* that implies the meaning of the hypothesis *production*.

To identify this relationship, an inference system needs a knowledge resource that would specify that the meaning of  $e'$  implies the meaning of  $e$ , at least in some contexts. We thus suggest the following definition to capture this relationship between  $e'$  and  $e$ :

**Definition 2** *A lexical element  $e'$  entails another lexical element  $e$ , denoted  $e' \Rightarrow e$ , if there exist some natural (non-anecdotal) texts containing  $e'$  which entail  $e$ , such that the reference to the meaning of  $e$  can be implied solely from the meaning of  $e'$  in the text.*

(Entailment of  $e$  by a text is according to Definition 1.)

We will refer to this relationship in this work as *lexical entailment*, and call  $e' \Rightarrow e$  a *lexical entailment rule*. The definition suggests a specification for the rules that should be provided by a lexical entailment resource, following an operative rationale: A rule  $e' \Rightarrow e$  should be included in an entailment rule base if it would be needed, as part of a compositional process, to infer the meaning of  $e$  from some natural texts. A rule need not apply in all contexts, as long as it is appropriate for some texts. For example, the rule *produce*  $\Rightarrow$  *lay* is valid in contexts where the producer is poultry and the product is egg, and therefore should be included in a lexical entailment resource.

As mentioned earlier, currently there are no rule bases that were designed specifically for lexical entailment modeling. Indeed, various lexical semantic resources, like those reviewed in Chapter 2 and used in our experiment (Chapter 6), do contain useful lexical entailment rules. However, the types of lexical relationships they capture do not coincide with the needed lexical entailment relationship. For example, WordNet synonyms and hyponyms are likely to satisfy the lexical entailment definition in most cases. Meronyms, however, lexically entail their holonym only if they are unique for it, e.g. *chest* entails *body*, but *wheel* does not

entail *car*. Other entailment rules, like *breastfeeding*  $\Rightarrow$  *baby*, do not correspond to any WordNet relation at all.

Thus, lexical entailment captures a broader notion of semantic relation between terms than the common lexicographic relations. Lexical entailment is well defined, as opposed to the association relation found by distributional similarity methods (described in 2.3.2). Definition 2 is intended to help us understand which rules derived from current resources should be regarded as valid and useful entailment rules, as well as how to design dedicated lexical entailment resources.

## Chapter 4

# Learning Lexical Entailment Rules from Wikipedia

Directed by the above notion of lexical entailment we aim to develop high-precision methods for rule extraction from Wikipedia, which will mostly yield correct entailment rules. Section 4.1 presents the various types of methods for rule extraction from Wikipedia that we developed. Section 4.2 examines supervised classification of the extracted rules as correct or incorrect, considering multiple features per rule.

### 4.1 Extraction Methods

Each Wikipedia article provides a definition for the concept denoted by its *title*. As the most concise definition we take the first sentence of each article, following (Kazama and Torisawa, 2007). Our preliminary evaluations showed that taking the entire first paragraph as the definition rarely introduces new valid rules while harming extraction precision significantly.

Since a concept definition usually employs more general terms than the defined concept (Ide and Jean, 1993), the concept title is more likely to entail terms appearing in its definition rather than the other way around. Therefore, when extracting rules from a definition, the title is taken as the left hand side (LHS) and the extracted definition term is taken as the right hand side (RHS) of the rule. As

Wikipedia’s titles are mostly noun phrases, the terms we extract as RHS are the nouns and noun phrases in the definition.

The first step of our algorithm is preprocessing. We clean each article from Mediawiki Markup tags and some HTML tags. Next we identify the first paragraph of the article and apply a sentence splitter in order to extract the first sentence of the article as the definition of the title. We parse the definition sentence using Minipar (Lin, 1998). Our preliminary experiments showed that parse-based extraction is more accurate than chunk-based extraction as taken by (Kazama and Torisawa, 2007). It also enables extracting additional rules by splitting conjoined noun phrases and by taking both the head noun and the complete base noun phrase as RHS for separate rules. We noticed that Minipar does not handle well long clauses (between two commas) and parenthesis (which are very common in definitions, e.g. for dates of birth or for writing the title name in its original language). Therefore we parse each definition twice, once as is and again after removing parenthesis and long clauses.

The next step is to identify the title (or part of it) in the definition sentence. Verifying its existence and identifying its syntactic role in the sentence is important for the first three extraction methods described next.

The remainder of this section describes methods for extracting rules from the definition sentence, and from the general structure of Wikipedia as a Web knowledge resource:

**1. Be-Complement** - Following the general idea in (Kazama and Torisawa, 2007), we extract nominal complements of the verb ‘be’ after verifying that the article title is its subject. This process resembles identifying occurrences of *IS-A* patterns (see the last two patterns in Table 2.1). We take the complete base noun phrase of such a complement as the RHS of a rule whose LHS is the article title (see example 1 in Table 4.1). We further extract the head of that complement as the RHS of a second rule whose LHS is the title (see example 2). When we encounter a conjunction in this process, we extract rules for all the conjoined noun phrases (see example 3).

There are cases in which the syntactic head of a noun phrase does not bear the semantic meaning of it, and there is another part of the noun phrase which

serves as the semantic head. This phenomenon is called *transparent head* (Fillmore et al., 2002; Grishman et al., 1986). In example 4, the term “genus” is the syntactic head found by the parser, while “catfish” is the semantic head which describes the subject “Rita”. In such cases we want to extract the semantic, rather than the syntactic, head as the LHS of the rule. This phenomenon usually happens with phrases like “group of”, “name of”, “variant of” and more. A customary way to overcome transparent heads is defining a list of such phrases, and when encountering one, the noun that follows it is taken as the semantic head. We also implemented this solution, while avoiding including too many phrases in the list as this could harm performance since not all occurrences of such phrases are necessarily cases of transparent heads. For instance, from the definition “*Danny is a name of a boy*” we would like to learn that “Danny” IS-A “name”.

**2. Indirect Be-Complement** - During data analysis we noticed that many complements of the verb ‘be’ were not extracted since, for those cases, the title was not found in the subject position for the verb. This may happen due to parsing errors or since the title is connected to the verb indirectly through discourse structure or as a result of inability to identify the title in the definition (see example 5 in which the title was not identified as the subject of the verb ‘be’ due to parse error). Our data analysis revealed that many of these nominal complements form valid rules and therefore should be extracted.  $\text{Be-Complement}_{\textit{indirect}}$  is the extraction method which does not enforce identifying the title as the subject of the verb ‘be’.

**3. All-Nouns** - The two *Be-Complement* extraction methods yield mostly hyponym relations, which do not exploit the full range of lexical entailments within the concept definition. Therefore, we further consider all head nouns and base noun phrases in the definition (see example 6). Yet, we observed that not all nouns in the definition have the same likelihood to be entailed by the concept title. The likelihood of such entailment depends greatly on the syntactic path connecting the title and the considered noun (which was exploited also in (Snow et al., 2006)). For instance, the path of example 5 is  $\textit{title} \xleftarrow{\textit{subj}} \textit{album} \xrightarrow{\textit{vrel}} \textit{released} \xrightarrow{\textit{by-subj}} \textit{by} \xrightarrow{\textit{pcomp-n}} \textit{noun}$ , which indicates an entailment relation between an album name and its cre-

No.	Extraction Method	Rule
<i>James Eugene "Jim" Carrey is a Canadian-American actor and comedian</i>		
1	<i>Be-Complement</i>	<i>Jim Carrey ⇒ Canadian-American actor</i>
2	<i>Be-Complement</i>	<i>Jim Carrey ⇒ actor</i>
3	<i>Be-Complement</i>	<i>Jim Carrey ⇒ comedian</i>
<i>Rita is a genus of catfishes of the family Bagridae</i>		
4	<i>Be-Complement</i>	<i>Rita ⇒ catfish</i>
<i>Brian Mulroney was the eighteenth Prime Minister of Canada and was leader of the Progressive Conservative Party</i>		
5	<i>Be-Complement<sub>indirect</sub></i>	<i>Brian Mulroney ⇒ leader</i>
<i>Abbey Road is an album released by The Beatles</i>		
6	<i>All-Nouns</i>	<i>Abbey Road ⇒ The Beatles</i>
7	<i>Redirect</i>	<i>CPU ⇔ Central processing unit</i>
8	<i>Redirect</i>	<i>Receptors IgG ⇔ Antibody</i>
9	<i>Redirect</i>	<i>Hypertension ⇔ Elevated blood-pressure</i>
10	<i>Link</i>	<i>pet ⇒ Domesticated Animal</i>
11	<i>Link</i>	<i>Gestaltist ⇒ Gestalt psychology</i>
12	<i>Parenthesis</i>	<i>Graph ⇒ mathematics</i>
13	<i>Parenthesis</i>	<i>Graph ⇒ data structure</i>

Table 4.1: Examples of the different extraction methods

ator (since many texts mentioning *Abbey Road* make an implicit reference to *The Beatles*).

In order to estimate the likelihood that a syntactic path indicates lexical entailment we collected from Wikipedia all paths connecting a title to a noun phrase in the definition sentence. We note that since there is no available lexical resource which covers the full breadth of our target entailment relation we could not obtain large-scale supervised training data for learning which paths correspond to correct entailments. This is in contrast to (Snow et al., 2005) who focused only on hyponymy and synonymy relations and could therefore extract positive and negative examples from WordNet. However, as our entailment relation is much broader, we do not want to train a classifier that would be focused on the currently available types of semantic relations.

We therefore propose the following unsupervised entailment likelihood score for a syntactic path  $p$  within a definition, based on two counts: the number of times the path  $p$  connects an article title with a noun in its definition, denoted by  $C_T(p)$ , and the total number of  $p$ 's occurrences in Wikipedia definitions,  $C(p)$ . The score of a path  $p$  is then defined as:

$$score(p) = \frac{C_T(p)}{C(p)}$$

The rationale for this score is that  $C(p) - C_T(p)$  corresponds to the number of times in which the path connects two nouns within the definition, none of which is the title. These instances are likely to be non-entailing, since a concise definition typically does not contain terms that can be inferred from each other. For instance, the path of example 5 obtained a score of 0.98, as it is used almost solely in definitions of musical creations.

We use this score to sort the large set of rules extracted by the *All-Nouns* method (as each rule was extracted from a path with such score) and we split the sorted list into 3 thirds: *All-Nouns<sub>top</sub>*, *All-Nouns<sub>middle</sub>* and *All-Nouns<sub>bottom</sub>*. As will be shown in Section 5.1 this split is indeed indicative for rule reliability. It may be worthwhile to investigate in future research additional path scoring formulas and methods to utilize them.

**4. Redirect** - As any dictionary and encyclopedia, Wikipedia contains a *Redirect* mechanism which, like URL redirect, redirect different search queries to a single canonical title. For instance, there are 86 different queries that redirect the user to *United States* (e.g. *U.S.A.*, *America*, *Yankee land*). Redirect links are hand coded, reflecting an assumption that both terms refer to the same concept. We therefore generate a bidirectional entailment rule for each redirect link (examples 7–9).

**5. Link** - Wikipedia texts contain hyper-links which point at articles. We scan all links in the entire text of each article. For each link we generate a rule whose LHS is the linking text and RHS is the title of the linked article (examples 10–11). In this case we do not generate a bidirectional rule since links do not necessarily

connect semantically equivalent entities.

**6. Title Parenthesis** - A common convention in Wikipedia for disambiguating ambiguous titles is adding a descriptive term in parenthesis at the end of the title, as in *The Siren (Musical)*, *The Siren (sculpture)* and *Siren (amphibian)*. From such titles with Parenthesis we extract rules in which the descriptive term inside the parenthesis is the RHS and the rest of the title is the LHS (examples 12–13).



## E.T. the Extra-Terrestrial

From Wikipedia, the free encyclopedia

(Redirected from [E.T. \(film\)](#))

Redirect

Parenthesis

Be-  
Complement

All-  
Nouns

*E.T. the Extra-Terrestrial* is a 1982 science fiction film co-produced and directed by [Steven Spielberg](#), written by [Melissa Mathison](#) and starring [Henry Thomas](#), [Robert MacNaughton](#), [Drew Barrymore](#), [Dee Wallace](#) and [Peter Coyote](#). It tells the story of Elliott (played by Thomas), a lonely boy who befriends a friendly [alien](#), dubbed "E.T.", who is stranded on [Earth](#). Elliott and his [Extraterrestrial life](#)

Link

Figure 4.1: Few extracted rules from a Wikipedia article

While the last three extraction methods depend on Wikipedia structure, they are not Wikipedia specific, as we expect any Web-like knowledge base to contain a redirect mechanism, hyper-links and some disambiguation means. Wikipedia has additional structural features such as category tags that place each article in

a topical hierarchy, infoboxes which are structured summary tablets for specific semantic classes (like countries or animals), and articles containing lists. We preferred a generic methodology and focused on the extraction methods described above, which are likely to be relevant and could be implemented for many Web-like knowledge bases in various domains.

Figure 4.1 visually exemplify the extraction of the following rules from one Wikipedia article: The title *E.T. the Extra-Terrestrial* entails *science fiction film* (Be-Complement), *Steven Spielberg* and *Drew Barrymore* (All-Nouns) and *E.T.* (Redirect). Since *Redirect* extracts bidirectional rules, *E.T.* entails the title as well. From *Parenthesis* we can learn that *E.T.*  $\Rightarrow$  *film* and from *Link* we can deduce that *alien*  $\Rightarrow$  *extraterrestrial life*.

After all rules were extracted we perform a postprocessing step in which we remove rules whose LHS contain their RHS (e.g. *E.T. the Extra-Terrestrial*  $\Rightarrow$  *E.T.*) since such rules are usually useless as their RHS does not add any information to their LHS. A rule can be extracted more than once, either by several extraction methods or from several articles. We merge duplicate extractions of the same rule, keeping a list of the different methods which extracted it. Finally we remove rules whose either sides is a stop word, being a very frequent word such as pronouns or determiners.

## 4.2 Supervised Classification of Rules

As will be shown in Section 5.1.1, the different extraction methods yield different precision and coverage levels. This may allow an application to utilize only extraction methods whose empirical precision is above a desired level, and thus choose between several possible recall-precision tradeoff points. As an alternative approach for selecting rules out of all extracted rules we investigated a supervised classification approach. Our goal was to find out whether representing each rule by multiple features and classifying rules for correctness can provide better recall-precision tradeoffs.

Each rule is represented by the following features:

**Extraction methods** - 8 binary features corresponding to the 8 extraction meth-

ods described above, allowing the classifier to prefer rules obtained by multiple extraction methods.

**Named entity** - Two binary features indicate for each side of the rule whether it is a named entity, based on Minipar's proper nouns tags. This additional information may assist classification. For instance, we expect to find many rules whose LHS is a named entity and their RHS is a common nouns (since titles are often proper names), while rules in which a common noun, which is the more general term, entails a named entity, which is a specific term, are less likely to be correct.

**Co-occurrence** - A numerical feature which records the number of Wikipedia articles in which both sides of the rule appear. In order to normalize this number, we divide it by the number of articles in which the RHS appears. Therefore this feature estimates the strength of the connection of each term in the set of LHSs entailing a single RHS.

**Extraction count** - The number of times the rule was extracted. This numerical feature is calculated during the duplicated merging process in the post processing step described above.

**Path score** - If the rule was extracted by one of the *All-Nouns* methods then this feature holds the score of the path connecting its sides as explained above, otherwise it is set to zero.

A training set was obtained by manually judging for correctness a random sample of extracted rules, as described in the next chapter. We then trained an SVM<sup>light</sup> classifier<sup>1</sup> and varied its  $J$  parameter to obtain different recall-precision tradeoffs, which can be compared with the basic strategy of choosing only few of the (more accurate) extraction methods. The  $J$  parameter defines the ratio by which training errors on positive examples outweigh errors on negative examples. As this ratio is higher the learning algorithm will try to avoid false negative judgments (determine that an example is wrong when it is actually correct). This will

---

<sup>1</sup>[www.svmlight.joachims.org](http://www.svmlight.joachims.org)

lead to preferring positive judgments over negative and thus to preferring recall over precision.

# Chapter 5

## Evaluation and Results

We applied our rule extraction methods over a version of Wikipedia available in a database constructed by (Zesch et al., 2007)<sup>1</sup>. The extraction yielded about 8 million rules altogether, with about 2.5 million distinct right hand sides and more than 2.8 million distinct left hand sides. As expected, the extracted rules involve mostly named entities and specific concepts, as typically covered in encyclopedias. For comparison, Snow’s published extension to Wordnet<sup>2</sup>, which covers similar types of terms but is restricted to synonyms and hyponyms, includes 400,000 relations.

We next present evaluations of the quality of our rule base. First, in order to understand the behavior of the different extraction methods we directly evaluate rule correctness relative to human judgment. Then, we indirectly evaluate the utility of the rule base for lexical expansion in keyword-based text categorization, and compare it to several baselines.

### 5.1 Direct Rule-level Evaluation

#### 5.1.1 Extraction Methods

We randomly sampled 800 rules from our rule base and manually annotated them for correctness, according to the lexical entailment notion specified in Chapter 3.

---

<sup>1</sup>English version from February 2007, contains 1.6 million articles. Can be found at [www.ukp.tu-darmstadt.de/software/JWPL](http://www.ukp.tu-darmstadt.de/software/JWPL). Other languages are also available there

<sup>2</sup><http://ai.stanford.edu/~rion/swn/>

In cases which were too difficult to judge, the annotator was allowed to abstain, which happened for 20 rules out of the 800. 66% of the remaining rules were annotated as correct. 200 rules from the sample were judged by another annotator for agreement measurement. The resulting Kappa score was 0.7 (corresponding to substantial agreement (Landis and Koch, 1997)), either when considering all the abstained rules as correct or as incorrect.

The middle columns of Table 5.1 present, for each extraction method, the obtained percentage of correct rules (precision) and their estimated absolute number. The number of correct rules is estimated by multiplying the number of annotated correct rules for the extraction method by the sampling proportion. The right part of Table 5.1 shows the performance figures for accumulated rule bases, created by adding the extraction methods one at a time in descending order of their precision. % used is the percentage of correct rules in each rule base out of the total number of correct rules extracted jointly by all methods (the union set).

Extraction Method	Per Method		Accumulated	
	Precision	Estimated # of Correct Rules	Precision	%used
<i>Redirect</i>	0.87	1,851,384	0.87	31
<i>Be-Complement</i>	0.8	1,083,903	0.84	52
<i>Be-Complement<sub>indirect</sub></i>	0.73	535,010	0.82	60
<i>Parenthesis</i>	0.71	94,155	0.82	60
<i>Link</i>	0.7	485,528	0.8	68
<i>All-Nouns<sub>top</sub></i>	0.6	684,238	0.76	83
<i>All-Nouns<sub>middle</sub></i>	0.46	380,572	0.72	90
<i>All-Nouns<sub>bottom</sub></i>	0.41	515,764	0.66	100

Table 5.1: Direct rule based evaluation: precision and estimated number of correct rules per extraction method, and precision and % of correct rules used of rule-sets accumulated by methods.

We can see that excluding the *All-Nouns* methods all extraction methods reach quite high precision levels of 0.7-0.87, with accumulated precision of 0.8. Furthermore, by selecting only a subset of the extraction methods, according to their precision, one can choose different recall-precision tradeoff points that suit application requirements. For instance, taking rules extracted by the six most precise methods (filtering only *All-Nouns<sub>middle</sub>* and *All-Nouns<sub>bottom</sub>*) uses 83% of

the total number of correct rules in our rule base and their accumulated precision is 0.76. We can see that *Be-Complement* precision is much higher than *Be-Complement<sub>indirect</sub>* and that it covers two thirds of the rules extracted by these two methods. The difference in their precision emphasizes the importance of using a full parser for our extraction, as opposed to using patterns or a chunker (shallow parser) which only approximate the syntactic relations between parts of the definition.

Precision for the unified set of all parts of *All-Nouns* rules was 0.49. By splitting this set into three sub-types we could obtain a reasonably high precision for the top third of these rules, relative to the other extraction types. The less accurate *All-Nouns<sub>middle</sub>* and *All-Nouns<sub>bottom</sub>* types may be used when high recall is important, accounting for 17% of the correct rules. The precision difference between these three types shows that our proposed path score provides useful information about rule reliability.

An examination of the paths in the *All-Nouns* types reveals, beyond standard hyponymy and synonymy, various semantic relations that satisfy lexical entailment, such as *Location*, *Occupation* and *Creation*, as illustrated in Table 5.2 (see Chapter 7 for further discussion). Typical relations covered by *Redirect* and *Link* rules include synonyms (*NY State Trooper*  $\Rightarrow$  *New York State Police*), morphological derivations (*irritate*  $\Rightarrow$  *irritation*), different spellings or naming (*Pythagoras*  $\Rightarrow$  *Pythagoras*) and acronyms (*AIS*  $\Rightarrow$  *Alarm Indication Signal*).

### 5.1.2 Error Analysis

We sampled 100 rules which were annotated as incorrect and examined the cause of errors. We also examined the distribution of extraction methods yielding the incorrect rules. Figure 5.1 presents our error analysis results. The pie chart on the right (A) shows the distribution of types of errors:

**Wrong NP part.** The most common error, accounting for 36% of the errors, is not taking the correct part of a noun phrase (NP) as described in Section 4.1. From each extracted NP we create two rules, by taking both the head noun and the complete base NP as the right hand side. While both rules are usually correct,

Relation	Rule	Path Pattern
<i>Location</i>	<i>Lovek</i> ⇒ <i>Cambodia</i>	<i>Lovek</i> <b>city in</b> <i>Cambodia</i>
<i>Occupation</i>	<i>Thomas H. Cormen</i> ⇒ <i>computer science</i>	<i>Thomas H. Cormen</i> <b>professor of</b> <i>computer science</i>
<i>Occupation</i>	<i>Stephen Hawking</i> ⇒ <i>University of Cambridge</i>	<i>Stephen Hawking</i> <b>Professor of Mathematics at</b> <i>University of Cambridge</i>
<i>Creation</i>	<i>Crime and Punishment</i> ⇒ <i>Fyodor Dostoevsky</i>	<i>Crime and Punishment</i> <b>novel by</b> <i>Fyodor Dostoevsky</i>
<i>Creation</i>	<i>Alice</i> ⇒ <i>Woody Allen</i>	<i>Alice</i> <b>film directed by</b> <i>Woody Allen</i>
<i>Origin</i>	<i>Willem van Aelst</i> ⇒ <i>Dutch</i>	<i>Willem van Aelst</i> <i>Dutch</i> artist
<i>Alias</i>	<i>Dean Moriarty</i> ⇒ <i>Benjamin Linus</i>	<i>Dean Moriarty</i> is <b>an alias of</b> <i>Benjamin Linus</i> on <i>Lost</i> .
<i>Spelling</i>	<i>Egushawa</i> ⇒ <i>Agushaway</i>	<i>Egushawa</i> , <b>also spelled</b> <i>Agushaway</i> ...

Table 5.2: *All-Nouns* rules exemplifying various types of lexical entailment relations

there are cases in which the left hand side entails the NP as a whole but not its head alone. For example, a British location entails *United Kingdom* but not *Kingdom*.

**Related but not Entailing.** Although all terms in a definition are highly related to the defined concept, not all are entailed by it. For 16% of the incorrect rules, the two sides are related but do not comply to the lexical entailment definition. Few examples: the origin of a person (*\*The Beatles* ⇒ *Liverpool*<sup>3</sup>), family ties such as ‘daughter of’ or ‘sire of’, predicates such as ‘founder of’, ‘manufacture in’ and ‘developed by’. It is worth noting that 62% of these wrong rules were extracted by the *All-Nouns<sub>bottom</sub>* method, and are thus identified by our method as being less reliable.

**All-Nouns pattern errors.** Most authors of articles in Wikipedia comply to the common convention to start an article with a short sentence which serves as a concise definition. However, some articles start with a longer sentence which may include information which is not directly relevant to the title of the article. For instance, *\*Pilar Primo de Rivera* ⇒ *Falange* was extracted from “*Pilar Primo de Rivera was the sister of José Antonio Primo de Rivera, founder of the Falange, a political movement of Spain*”. Some incorrect rules of this type result from taking

<sup>3</sup>The asterisk marks an incorrect rule

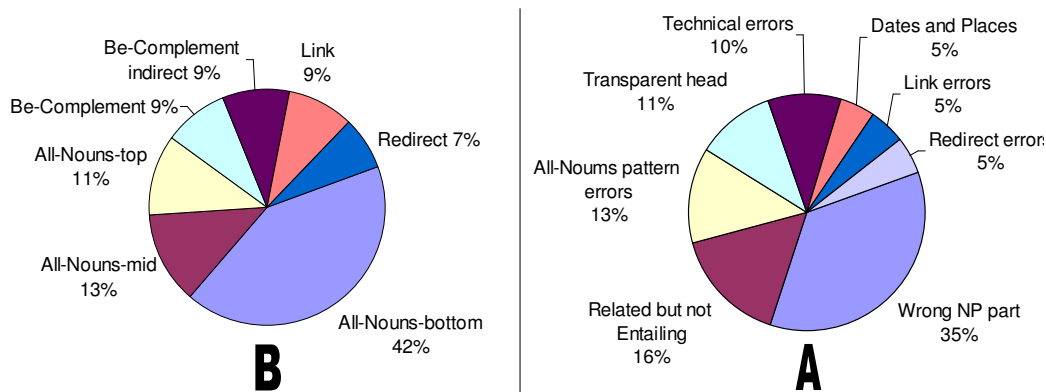


Figure 5.1: Error analysis: type and origin of incorrect rules

location names that appear in the description of the title. Many times the location is indeed entailed by the title, but this is not always the case. For instance, consider *\*Interstate 80 ⇒ California* from “*Interstate 80, the second-longest U.S. Interstate highway, runs from California to New Jersey*”. 57% of these incorrect rules were extracted by the *All-Nouns<sub>bottom</sub>* method.

**Transparent head.** This is the phenomenon in which the syntactic head of a noun phrase does not bear the semantic meaning of it, and there is another part of the noun phrase which serves as the semantic head mentioned in Section 4.1. Since parsers identify the syntactic head, in such cases we extract an incorrect rule. For instance, deriving *\*Prince William ⇒ member* instead of *Prince William ⇒ British Royal Family* from “*Prince William is a member of the British Royal Family*”. Even though we implemented the common solution for this phenomenon (i.e. constructing a list of phrases of transparent head) it is partial, as there is no close set of such phrases.

**Technical errors.** 10% of the incorrect rules result from technical extraction errors mainly due to wrongly identifying the appearance of the title in the definition or not handling well different languages than English.

**Dates and Places.** Dates and places where a certain person was born at, lived in or worked at often appear in definitions but do not comply to the lexical entailment definition (*\*Galileo Galilei ⇒ 15 February 1564*). A convention in Wikipedia is to write these dates in parenthesis. We remove such parenthesis

in the preprocessing step of our algorithm, thus avoiding most of these incorrect extractions. In our sample, all such rules were extracted from the *All-Nouns<sub>bottom</sub>* method.

**Link errors.** These are usually the result of connecting the sides of the rule in the wrong direction. Such errors mostly occur when a general term, i.e. *revolution*, links to a more specific (albeit typical) concept, i.e. *French Revolution*.

**Redirect errors.** These may occur in rare cases in which we should not have created a bidirectional rule (e.g. *\*Anti-globalization*  $\Rightarrow$  *Movement of Movements* is wrong but the other way around is correct, as *Movement of Movements* is a popular term in Italy for *Anti-globalization*).

The second pie chart (B) in Figure 5.1 shows the part of each one of the extraction methods in the sample of incorrect rules. We first removed the errors which resulted from technical errors and from taking the wrong NP part, as these errors are not a result of particular extraction method. Not surprisingly, the less accurate methods of *All-Nouns* are responsible for most of the errors (65%).

### 5.1.3 Supervised Learning Classification

As described in Section 4.2, we examined the possibility of achieving better recall-precision tradeoffs using a supervised classifier, which considers several features per rule. With SVM<sup>light</sup> we can control recall-precision tradeoff by varying the  $J$  parameter. Table 5.3 summarizes averaged rule classification results for a 5-fold cross validation on the annotated sample described in the previous sub-section, with various  $J$  values. Comparing this table with Table 5.1 shows almost identical ability to rank the rules by their reliability. This result indicates that although our classifier considers additional information than just the extraction method, it does not yield better recall-precision tradeoffs relative to relying on the extraction method alone. Further research is thus needed to improve our current feature set and classification performance.

J	0.3	0.4	0.5	0.6	0.9	1.5
Precision	0.86	0.83	0.81	0.79	0.72	0.66
% used	34	54	59	69	82	100

Table 5.3: Different usage of correct rules and precision tradeoff points for supervised classification of rule correctness.

## 5.2 Indirect Evaluation within a Text Categorization Task

In addition to the direct rule-level evaluation it is desired to assess whether our extracted rules are indeed useful and can be applied effectively within a concrete application. To that end we evaluated the utility of our and other baseline rule bases on lexical expansion in an available keyword-based text categorization (TC) system<sup>4</sup>.

Keyword-based text categorization methods aim at topical categorization of documents based on sets of terms, without requiring a supervised training set of labeled documents (McCallum and Nigam, 1999; Ko and Seo, 2004; Liu et al., 2004). Generally speaking, such systems operate in two phases: (i) a setup phase, in which a set of characteristic terms for the category is assembled, constituting the category’s feature vector, and (ii) a classification phase, in which the term-based feature vector of a classified document is compared with the feature vectors of all categories.

Our basic categorization system implements the above two phases as follows. Taking a textual entailment perspective, we assume that the characteristic terms in a category’s vector should entail the term (or terms) denoting the category name (this perspective is somewhat analogous to the use of information retrieval queries as hypotheses in the textual entailment datasets (Giampiccolo et al., 2007)). Accordingly, we construct the category’s feature vector by taking first the category name itself, and then expanding it with all left hand sides of lexical entailment rules whose right hand side is identical to the category name. For example, the category *Cars* is expanded by rules such as *Ferrari F50*  $\Rightarrow$  *car*. During classifi-

---

<sup>4</sup>Developed by Libby Barak

cation cosine similarity is measured between the feature vector of the classified document and the vectors of all categories, assigning the document to the category which yields the highest similarity score, following a single-class classification approach (Liu et al., 2004).

It should be noted that various keyword-based text categorization systems employ additional steps, such as bootstrapping, which generalize to multi-class settings and further improve performance. Our basic system suffices though to evaluate comparatively the direct impact of expansion resources on intermediate performance. We also note that our categorization setting is similar to query expansion in information retrieval (IR). Yet, the advantage of using a text categorization test set in our case is that it includes exhaustive annotation for *all* documents. Typical IR datasets, on the other hand, are partially annotated through a pooling procedure. Thus, some valid lexical expansions might retrieve non-annotated documents, which were missed by the previously pooled systems. This is particularly likely to happen when using novel types of expansion methods such as our Wikipedia-based resource.

For our evaluation we used the 20-News Groups collection in the “bydate” version,<sup>5</sup> which contains 18941 documents partitioned (nearly) evenly over the 20 categories<sup>6</sup>. Classification results are reported for the test set part of the collection, containing 40% of the documents. Document features consist of POS-tagged lemmata of single words and bigrams, limited to nouns, verbs, adverbs and adjectives, with term frequency as the feature value.

We compare the quality of our rule base expansions to 5 baselines (Table 5.4). The basic one avoids any expansion, classifying documents based on cosine similarity with the category names only. Such method would classify a document to a category only if the category name explicitly appears in it. As expected, this baseline yields relatively high precision but low recall, indicating the need for lexical expansion. Then we utilized 3 lexical semantic resources to expand the category name:

---

<sup>5</sup>[www.ai.mit.edu/people/jrennie/20Newsgroups](http://www.ai.mit.edu/people/jrennie/20Newsgroups).

<sup>6</sup>The keywords we used as category names are: atheism; graphic; microsoft windows; ibm,pc,hardware; mac,hardware; x11,x-windows; sale; car; motorcycle; baseball; hockey; cryptography; electronics; medicine; outer space; christian(noun & adj); gun; mideast,middle east; politics; religion

Rule Base		Precision	Recall	F <sub>1</sub>
Baselines	No Expansion	0.53	0.19	0.28
	Snow <sup>400K</sup>	0.56	0.13	0.21
	WikiBL	0.53	0.19	0.28
	WordNet <sub>first_sense</sub>	0.5	0.28	0.36
	WordNet <sub>all_senses</sub>	0.48	0.31	0.38
Extraction Methods	Redirect only	0.54	0.21	0.3
	above + Be-Complement	0.50	0.22	0.3
	above + Be-Complement <sub>indirect</sub>	0.55	0.21	0.3
	above + Parenthesis and Link	0.41	0.3	0.34
	above + All-Nouns <sub>top</sub>	0.39	0.29	0.35
	above + All-Nouns <sub>middle</sub>	0.38	0.3	0.35
	above + All-Nouns <sub>bottom</sub> (all rules)	0.39	0.31	0.35
Classifier	j = 0.3	0.55	0.21	0.31
	j = 0.5	0.55	0.21	0.31
	j = 1.1	0.31	0.28	0.3
Union	WN <sub>first_sense</sub> + Wiki <sub>all_rules</sub>	0.39	0.33	0.36
	WN <sub>all_senses</sub> + Wiki <sub>all_rules</sub>	0.39	0.34	0.36
	WN <sub>first_sense</sub> + Wiki <sub>top_precise</sub>	0.52	0.3	0.39
	WN <sub>all_senses</sub> + Wiki <sub>top_precise</sub>	0.49	0.33	0.4

Table 5.4: Results of different methods for category name expansion

**WordNet** - We use WordNet to expand category names by their derivations and synonyms. Each obtained term is then expanded by its immediate hyponyms, or by its meronyms if it has no hyponyms. Finally, these terms are further expanded by their derivations and synonyms, constructing the category vector from all obtained terms<sup>7</sup>. There are two common approaches regarding choosing which senses of a term in WordNet to use, either taking the first sense only (the most common sense of the term) or taking all its senses. The first approach yields better precision while the latter achieves better recall. We tested both options and present WordNet<sub>all\_senses</sub> which uses all the senses of the category name, and WordNet<sub>first\_sense</sub> which uses only its first sense<sup>8</sup>. WordNet expansions improve substantially both Recall

<sup>7</sup>We also tried expanding by the entire hyponym hierarchy, but the method described above achieved the highest performance.

<sup>8</sup>We found that for the hyponyms and meronyms expansions it is better to take all senses than considering only the first one.

and  $F_1$  (the harmonic mean of recall and precision) relative to no expansion, while decreasing precision.

**Wiki Baseline** - We implemented the relevant part of (Kazama and Torisawa, 2007), taking the first noun after a *be* verb in the Wikipedia definition sentence, denoted as WikiBL. This baseline does not improve at all over no expansion.

**Snow**<sup>400k</sup> - We also examined Snow's extension to WordNet which was mentioned earlier, but its expansions did not yield any improvement, either over the No Expansion baseline or over WordNet when joined with the WordNet expansions.

We then used for expansion different subsets of our Wikipedia rule base, producing different recall-precision cutoff points either by progressively including the different extraction methods or by using a couple different  $J$  values in the SVM classifier. Taking rules from the *Redirect* and both *Be-Complement* extraction methods yields the highest precision, even slightly higher than using the category name only (in the *No Expansion* baseline). Using any subset of rules yields better performance than Snow's extension and WikiBL. Using the entire rule base, our resource's  $F_1$  is only slightly lower than WordNet's, even though it was constructed automatically. As Wikipedia continues to grow, the recall of our resource is expected to increase while its precision should be maintained, or further improved by future research.

Finally, since a dictionary and an encyclopedia are complementary in nature, we applied the union of all WordNet and Wikipedia expansions. The union of WordNet<sub>first\_sense</sub> with all rules from Wikipedia does not improve WordNet's  $F_1$ . Further investigation revealed that given the high precision of WordNet expansions, joining them with the rules extracted by the less precise methods from Wikipedia causes some harm. On the other hand, joining WordNet<sub>all\_senses</sub> only with the output of the *top precise* types of *Redirect* and both *Be-Complement* methods improves performances over WordNet alone, yielding the best performing configuration.

In order to better understand the contribution of using the Wikipedia rule base over WordNet for category name expansion we investigated which correct expanding terms were found by Wikipedia and were not found by WordNet. We consider an expanding term as correct if it was found in a document which belongs to the category whose name it expands (e.g. *coalition* was found in a *Politics* document and assisted in identifying its topic). Table 5.5 presents few correct expanding terms found only by Wikipedia for several category names.

Category Name	Expanding Terms
Politics	democracy, opposition, coalition, alliance, libertarianism, whip <sup>9</sup>
Cryptography	adversary, crypto, cryptosystem, key, secure
Mac	PowerBook, Grab <sup>10</sup> , Radius <sup>11</sup>
Religion	Jesus, heaven, abyss, creation, religious, baptism, miracle, belief, missionary
Motorcycles	biker, rider, cruiser <sup>12</sup> , Nighthawk <sup>13</sup> , Norton <sup>14</sup> , swingarm <sup>15</sup>
Medicine	doctor, physician, treatment, clinical, MD
computer graphics	rendering , radiosity <sup>16</sup> , CorelDRAW <sup>17</sup> , palette, drawing, SIGGRAPH <sup>18</sup>

Table 5.5: Terms found by Wikipedia and not by WordNet that assist correct classification

The marginal contribution of Wikipedia rule base is thus not just in finding proper names that entail the category name (e.g. PowerBook and Nighthawk), but also in finding common nouns that entail is (e.g. opposition, adversary and belief).

<sup>9</sup>A legislator who ensures party members attend and vote as the leadership desires.

<sup>10</sup>A Macintosh screen capture software.

<sup>11</sup>A computer hardware firm specializing in Macintosh equipment.

<sup>12</sup>A style of motorcycles, e.g. Harley-Davidson.

<sup>13</sup>A motorcycle created by Honda.

<sup>14</sup>A British motorcycle manufacturer.

<sup>15</sup>A part of a motorcycle (a.k.a swing fork).

<sup>16</sup>An illumination algorithm.

<sup>17</sup>A graphics editor.

<sup>18</sup>A conference on computer graphics.

## Chapter 6

# Comparative Evaluation of Lexical Rule bases

As a part of our investigation of the lexical entailment relation we compare the utility of several lexical-semantic resources (including ours) as a source for applying lexical entailment rules. The evaluation provides a comparative assessment of current resources with respect to different lexical entailment types. It reveals which types of relations are covered by which resources and the strength and weaknesses of each resource. Thus the evaluation process and the resulting analysis expose the broader nature of lexical entailment.

A rule is typically applied within an inference system by finding a text that matches its left hand side, from which we then infer the rule's right hand side. The utility of a lexical-semantic resource thus depends upon the actual performance of its rule applications, rather than just on the proportion or absolute number of correct rules that it contains. Therefore we use an "instance-based" evaluation methodology, in which we simulate rule applications by collecting texts that contain rules' left hand side and manually assessing the correctness of their applications.

As exemplified by the rule *produce*  $\Rightarrow$  *lay* (mentioned in Section 3.2), correct rules might yield incorrect applications in inappropriate contexts. Another example is *have*  $\Rightarrow$  *born* where the problem is even more severe since the verb *have* has a very dominant meaning which is different than the one valid for the rule (i.e

have children). This is a well known phenomenon (observed, e.g. in (Voorhees, 1994) for IR) that may be avoided in two different occasions. First, systems often avoid attempting to apply rules in inappropriate contexts, when successful matching of other parts of the hypothesis implicitly validates the context for the rule application. For example, the rule *waterside*  $\Rightarrow$  *bank* will most likely not be applied incorrectly when trying to infer the hypothesis *bank loans* since texts matching *waterside* are unlikely to contain the meaning of *loan*. Second, inference systems may employ explicit context matching modules such as Word Sense Disambiguation (WSD) or sense matching (Dagan et al., 2006), in order to avoid rule applications that do match the hypothesis left hand side but are inappropriate in the given context.

We chose to incorporate only the first setting since context matching algorithms are not sufficiently standardized yet within inference systems. This makes our methodology consistent for different rule bases and enables system-independent comparison of resources. Systems that do employ explicit context matching modules will be able to avoid some wrong rule applications and reach a higher level of performance than we measured in our simulation.

Next we describe our lexical resource evaluation methodology and in Section 6.3 we present seven state-of-the-art lexical resources and the methods by which we extract lexical entailment rules from them. In Section 6.4 we present the results of the comparative evaluation.

## 6.1 Evaluation Methodology

In this section we present a methodology which we developed to evaluate practical performance of inference methods, based on their application over instances extracted from corpora. We used this methodology in (Szpektor et al., 2007) to evaluate entailment rule acquisition and in Bar-Haim et al. (2007) in the process of evaluating semantic inference.

One input of our evaluation methodology is a lexical-semantic resource  $R$ , which contains lexical inference rules. Our goal is to evaluate the utility of  $R$  in inferring hypotheses from the sentences in a corpus. An additional input,  $H$ , is a sample of test hypotheses that contain more than one lexical element, in order to

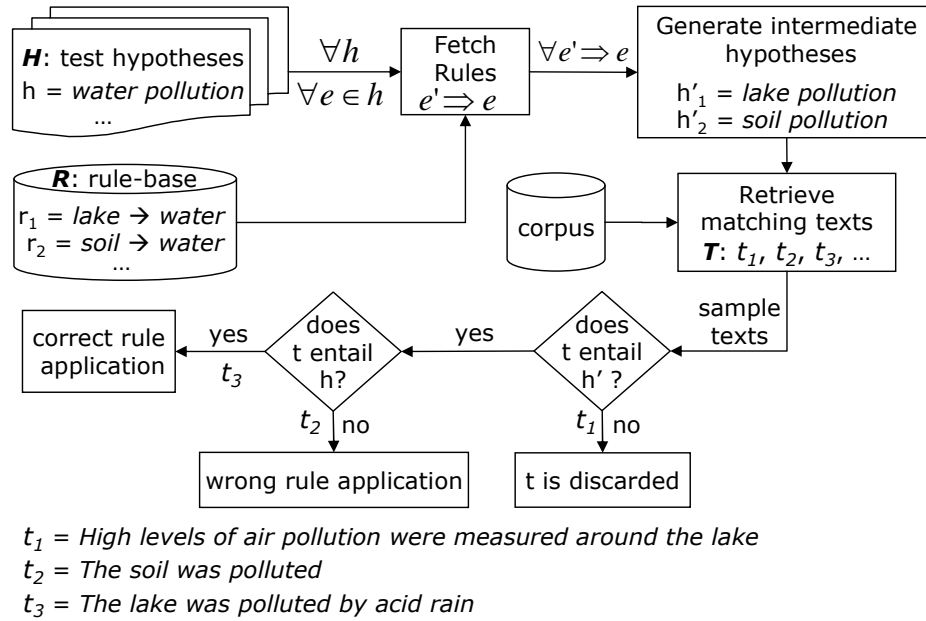


Figure 6.1: Comparative evaluation methodology flow chart

provide some context for rule applications, such as  $h = \text{water pollution}$ . Figure 6.1 illustrates the flow of our evaluation methodology as explained below:

**Fetch rules** - For each  $h \in H$  and each lexical element  $e \in h$  (e.g. *water*) we fetch from  $R$  all the rules that might be applied to entail  $e$ . These are all rules of the form  $e' \Rightarrow e$ , such as *lake*  $\Rightarrow$  *water*.

**Generate intermediate hypotheses  $h'$**  - In order to isolate the application of a single rule  $r$ , we utilize an intermediate hypothesis  $h'$ , constructed by replacing  $e$  in  $h$  with  $e'$  from  $r$ . This way, applying  $r$  on the intermediate hypothesis  $h'$  will result in the original hypothesis  $h$ . From a text  $t$  entailing  $h'$ ,  $h$  can be further entailed by the single application of the rule  $r$ , relying on the transitivity of entailment. In our example,  $h'_1 = \text{lake pollution}$ . We thus simulate the process by which an entailment system would infer  $h$  from  $t$  using  $r$ .

**Retrieve matching texts** - For each  $h'$  we retrieve from a corpus all texts that contain the lemmatized words of  $h'$ , but do not contain  $e$ . These texts may entail  $h'$ , but do not match  $h$  directly. We discard texts that match  $h$  since entailing  $h$  from them does not require the application of the evaluated rule  $r$ . In our example,

the retrieved texts contain *lake* and *pollution* but not *water*. We thus obtain a large set of text instances  $T$  which may entail the various intermediate hypotheses created for the rules.

A sample of the matching texts is presented to a human annotator and the following questions are presented, which simulate the typical inference of an entailment system:

**Does  $t$  entail  $h'$ ?** The annotator is asked whether the retrieved  $t$  entails  $h'$ . If not, the text is discarded, as it would not provide a relevant example for the application of  $r$ . For example,  $t_1$  (“*High levels of air pollution were measured around the lake*”) does not entail  $h'_1$  (*lake pollution*) and thus we cannot deduce  $h$  from it by applying the rule  $r$ .

**Does  $t$  entail  $h$ ?** If  $t$  is annotated as entailing  $h'$ , and assuming that this entailment would be recognized by the entailment system, the system would then infer  $h$  from  $h'$  by applying  $r$ . We thus ask the annotator whether  $t$  indeed entails  $h$ . If the annotator determines that  $h$  is not entailed from  $t$  even though  $h'$  is entailed, then the rule application would be considered invalid. For instance,  $t_2$  (“*The soil was polluted*”) does not entail  $h$  even though it entails  $h'_2$  (*soil pollution*). Indeed, the application of  $r_2 = *soil \Rightarrow water$ <sup>1</sup>, from which  $h'_2$  was constructed, yields incorrect inference. If the answer is ‘yes’, as in the case of  $t_3$  (“*The lake was polluted by acid rain*”), the application of  $r$  for  $t$  is considered valid.

The above process yields a sample of annotated rule applications for each test hypothesis, from which we can measure resources performance, as described in Section 6.4.

## 6.2 Dataset and Annotation

Current state-of-the-art lexical-semantic resources mainly deal with nouns, therefore we used nominal hypotheses. We chose the corpora of TREC 1-8 , Information Retrieval benchmarks<sup>2</sup> as our corpus and randomly sampled 25 ad-hoc

---

<sup>1</sup>The asterisk marks an incorrect rule

<sup>2</sup><http://trec.nist.gov/>

queries of two-noun compounds as our hypotheses<sup>3</sup> We did not use longer hypotheses to ensure that enough texts containing the intermediate hypotheses are found in the corpus.

While  $h$  is a query used in TREC and therefore is a coherent phrase,  $h'$  is not always so (e.g. *water dumping* created from the rule *dumping*  $\Rightarrow$  *pollution*). In such cases the annotators were allowed to construct a simple phrase from the nouns in  $h'$  using function words such as prepositions and support verbs (creating *dumping into water*). For annotation simplicity, we used individual sentences as our texts.

While we used Information Retrieval queries, such hypotheses can be used in any natural language processing application (e.g. *railway accidents* and *water pollution* can be Information Extraction events).

Applying our methodology, for each rule applied for an original hypothesis  $h$ , we retrieved all sentences in which the words of the intermediate hypothesis  $h'$  were found. As a baseline, we also retrieved all sentences in which the words of  $h$  were found directly. We sampled 10 sentences retrieved by the rules of each resource for each hypotheses, and additional 10 sentences retrieved for the hypothesis itself (the baseline). In total, 1945 unique sentences were sampled and annotated by two fluent English speaking annotators.

## 6.3 Lexical-Semantic Resources

Additionally to our Wikipedia rule base, we evaluated the following resources:

**WordNet ( $WN^d$ )** - Given a term  $e$ , we consider all its synonyms and direct hyponyms in WordNet 3.0 for all senses of  $e$ . As there is no clear agreement regarding which set of WordNet relations correspond to lexical entailment, we took a conservative approach and used only synonyms and direct hyponyms which by definition comply with the entailment relationship. For each word  $e$  we created

---

<sup>3</sup>We used TREC 1-8 excluding TREC 4 (which contains only descriptions but no queries). We used the following queries: airbus subsidies, Antarctica exploration, antibiotics ineffectiveness, bank failures, cigarette consumption, computer security, Euro opposition, gun control, hydrogen energy, implant dentistry, journalist risks, marine vegetation, nuclear proliferation, official corruption, outpatient surgery, paper cost, police deaths, pope beatifications, railway accidents, Schengen agreement, ship losses, stirling engine, surrogate motherhood, welfare reform, wildlife extinction

a rule  $e' \Rightarrow e$  for each  $e'$  amongst its synonyms or direct hyponyms. See examples 1–4 in table 6.2.

**Snow (*Snow*<sup>30k</sup>)** - (Snow et al., 2006) presented a probabilistic model for taxonomy induction, which considers as features paths in parsed trees between related taxonomy nodes. They show that the best performing taxonomy was the one adding 30,000 hyponyms to WordNet<sup>4</sup>. We created a hyponymy rule for each new hyponym added to WordNet by their algorithm (examples 5–7).

**LCC's extended WordNet (*XWN*<sup>\*</sup>)** (Moldovan and Rus, 2001) - In this resource WordNet glosses were transformed into logical form axioms. From this representation we created a rule  $e' \Rightarrow e$  for each  $e'$  in the gloss tagged as referring to the same entity as  $e$  (examples 8–11).

**CBC** - A knowledge base of labeled clusters resulted from the statistical clustering and labeling algorithms in (Pantel and Lin, 2002; Pantel and Ravichandran, 2004)<sup>5</sup>. Given a cluster label  $e$ , an entailment rule  $e' \Rightarrow e$  is created for each member  $e'$  of the cluster (examples 12–14).

**Lin Dependency Similarity (*Lin-dep*)** - A syntactic-dependency based distributional word similarity resource described in (Lin, 1998)<sup>6</sup>. Given a term  $e$  and its list of similar terms we construct for each  $e'$  in the list the rule  $e' \Rightarrow e$  (examples 15–16). This resource was previously used for lexical matching in textual entailment engines, e.g. (Roth and Sammons, 2007)

**Lin Proximity Similarity (*Lin-prox*)** - A knowledge base of terms with their co-occurrence-based distributionally similar terms. Rules are created from this resource as for the previous one (examples 17–20).

**Wikipedia baseline (*WikiBL*)** - (Kazama and Torisawa, 2007) used Wikipedia as external knowledge to improve Named Entity Recognition. We constructed a rule base from Wikipedia using the first step of their algorithm, extracting for each page title a noun, from the first sentence in the page, that appears in an *is-a* pattern referring to the title. For each such pair we constructed a rule  $title \Rightarrow noun$  (examples 21–23).

The above resources represent various methods for detecting lexical relations:

---

<sup>4</sup>Available at <http://ai.stanford.edu/~rion/swn/>

<sup>5</sup>Kindly provided to us by Patrick Pantel.

<sup>6</sup>Lin's resources are available at <http://www.cs.ualberta.ca/~lindek/demos.htm>

manually and semi-automatically constructed ( $WN^d$  and  $XWN^*$ , respectively), automatically constructed based on a lexical-syntactic pattern ( $WikiBL$ ), distributional methods ( $Lin-dep$  and  $Lin-prox$ ) and a combination of pattern-based and distributional methods ( $CBC$  and  $Snow^{30k}$ ).

## 6.4 Comparison Results

We applied our comparison methodology (from Section 6.1) on the seven lexical resources described above and on our *Wikipedia* rule base. Annotation agreement, on the two questions in our methodology, was measured on 220 sentences. Our two judges achieved Kappa coefficient scores of 0.74 and 0.64 for judging the entailment of  $h'$  and  $h$ , respectively, which corresponds to substantial agreement (Landis and Koch, 1997). Analyzing the disagreements between the judges, we observe that quite a few can be attributed to a different interpretation of the concept inferred by the hypothesis. For example, the annotators disagreed on whether “*regulations regarding toy guns*” is entailing *gun control*.

We evaluated each resource using two measures, precision and recall-share, macro averaged over all hypotheses. We chose macro average to give equal weight to the various queries. *Precision* is the percentage of texts entailing  $h$  from those that entailed  $h'$ :

$$Precision = \frac{count(h = 1)}{count(h' = 1)}$$

We note that absolute recall cannot be measured since the number of texts entailing each hypothesis is unknown. Instead, *recall-share* measures the relative additional contribution of the resource’s rules to recall. We denote by  $yield(h)$  the number of texts annotated as entailing  $h$  obtained by matching the original hypothesis  $h$  directly. This figure is estimated as the number of texts retrieved for  $h$  and annotated as entailing it, multiplied by the sampling proportion. In the same fashion we estimate the number of texts entailing  $h$  obtained through entailment rules from each resource  $R$ , denoted  $yield_R(h)$ . *Recall-share* of a resource is then the proportion of the yield obtained through rules relative to the sum of yields

obtained by the original  $h$  and the rules:

$$recall - share = \frac{yield_R(h)}{yield(h) + yield_R(h)}$$

The results achieved for each resource are summarized in Table 6.1.

Resource	Precision	Recall-share
Original $h$	0.72	-
<i>Snow</i> <sup>30k</sup>	0.56	0.04
<i>WN</i> <sup>d</sup>	0.55	0.2
<i>XWN</i> <sup>*</sup>	0.51	0.09
<i>WikiBL</i>	0.45	0.07
<i>Wikipedia</i>	0.43	0.28
<i>CBC</i>	0.33	0.09
<i>Lin-dep</i>	0.28	0.43
<i>Lin-prox</i>	0.24	0.36

Table 6.1: Comparison between lexical resources

A clear distinction between *Lin-dep*, *Lin-prox* and *CBC* and the other resources is reflected in their precision scores. These statistical methods yield noticeably lower precision, especially *Lin-dep* and *Lin-prox*. The two Lin resources yield, on the other hand, the highest recall-share. An analysis of incorrect rule applications shows that for these resources, incorrect rules have a major negative impact on the resources' precision, whereas for *WN*<sup>d</sup>, *XWN*<sup>\*</sup>, *Snow*<sup>30k</sup>, *WikiBL* and *Wikipedia*, this is not a significant factor at all. In fact, it turns out that these resources highly conform with the lexical entailment definition, yielding very few applications of incorrect rules. Incorrect applications of rules from these sources mainly result from application of a correct rule in an inappropriate context (see the discussion below). The table reveals that *Lin-dep* and *Wikipedia* have the same best combination of recall and precision values with opposite strengths, as *Wikipedia* is more accurate and *Lin-dep* has a higher recall-share. Last, we can see that our new resource, *Wikipedia*, increases recall share by a factor of 4 compared to its baseline, *WikiBL*, almost without harming precision.

Table 6.2 illustrates correct and incorrect rule applications derived from the different resources for the various queries. The annotation column indicates whether the application of the rule was correct (i.e. both  $h'$  and  $h$  were annotated as entailed by the sentence), while an asterisk marks an incorrect rule. Note that there are cases of correct rules yielding an incorrect application, due to several reasons. Following we describe these reasons as well as few more insights from the table:

- One case of correct rule yielding an incorrect application is when the right hand side (RHS) of the rule refers to a different meaning of the term than the one used in the query (see example 24 where *stock* entails *security* as a bond, or a certificate of stock).
- Example 1 exemplifies a different meaning mismatch, this time in the left hand side (LHS) of the rule. In the domain of computers, *client* and *guest* (as well as *host*) are hyponyms of *computer* and therefore entail it. However, all these terms have a more commonly used meaning which does not fit the sense specified in the query.
- The disadvantage of the distributional similarity method, as a resource for entailment, is revealed in examples 19–20. Many co-hyponyms<sup>7</sup> such as *air-marine* (natural environment) and *doctor-journalist* (occupation) are found by this method.
- Both *WikiBL* and *Wikipedia* find many correct but rare rules that result in incorrect applications. For instance, *work*, in physics, is a difference between energies, and *ionization potential* is the energy required to remove an electron from the isolated atom, but these terms do not suit the query and therefore result in an incorrect application (examples 26 and 21 respectively).

Our comparison reveals that the standard WordNet synonyms and hyponyms account for a limited portion of the needed entailment relations. One type of complementary information is relations for proper names that can be found in

---

<sup>7</sup>a.k.a sister terms or coordinate terms, terms that share a common father in the hypernym hierarchy

rules from *Snow*<sup>30k</sup>, *WikiBL* and *Wikipedia*. Other less standard lexical relations can be found in distributional similarity resources such as *Lin-prox* and *Lin-dep*, which achieved the highest recall-share in our evaluation. Since such resources acquire a rather loose notion of semantic similarity, further work is needed to extract more precise lexical entailment rules from them.

No.	Resource	Query	rule	Annotation
1	<i>WN<sup>d</sup></i>	computer security	<i>client, guest ⇒ computer</i>	0
2		welfare reform	<i>*health ⇒ welfare</i>	0
3		official corruption	<i>degeneracy ⇒ corruption</i>	1
4		cigarette consumption	<i>use ⇒ consumption</i>	1
5	<i>Snow<sup>30k</sup></i>	paper cost	<i>Newsday ⇒ paper</i>	0
6		computer security	<i>IBM ⇒ computer</i>	1
7		ship losses	<i>USS Stark, Achille Lauro ⇒ ship</i>	1
8	<i>XWN*</i>	marine vegetation	<i>*growth ⇒ vegetation</i>	0
9		gun control	<i>*ammunition ⇒ gun</i>	0
10		official corruption	<i>bureaucrat ⇒ official</i>	1
11		computer security	<i>workstation ⇒ computer</i>	1
12	<i>CBC</i>	bank failures	<i>*error ⇒ failure</i>	0
13		cigarette consumption	<i>*alcohol ⇒ cigarette</i>	0
14		ship losses	<i>cruiser, battleship ⇒ ship</i>	1
15	<i>Lin-dep</i>	Antarctica exploration	<i>*south China sea ⇒ Antarctica</i>	0
16		wildlife extinction	<i>Amphibian, owl, rhino ⇒ wildlife</i>	1
17	<i>Lin-prox</i>	gun control	<i>arm ⇒ gun</i>	1
18		railway accidents	<i>locomotive ⇒ railway</i>	1
19		marine vegetation	<i>*air ⇒ marine</i>	0
20		journalist risks	<i>*doctor ⇒ journalist</i>	0
21	<i>WikiBL</i>	hydrogen energy	<i>ionization potential ⇒ energy</i>	0
22		computer security	<i>laptop ⇒ computer</i>	1
23		journalist risks	<i>reporter ⇒ journalist</i>	1
24	<i>Lin-dep, WN<sup>d</sup></i>	computer security	<i>stock ⇒ security</i>	0
25	<i>WN<sup>d</sup>, Lin-prox</i>	pope beatifications	<i>pontiff ⇒ pope</i>	1
26	<i>Wikipedia</i>	hydrogen energy	<i>work ⇒ energy</i>	0
27		bank failures	<i>*institution ⇒ bank</i>	0
28		Euro opposition	<i>single European currency ⇒ Euro</i>	1
29		welfare reform	<i>Perestroika ⇒ reform</i>	1
30		railway accidents	<i>train ⇒ railway</i>	1

Table 6.2: Rules derived from lexical semantic resources and the annotation of their application query expansion. The asterisk marks an incorrect rule

## Chapter 7

### Conclusions and Future Work

In this work we investigated the lexical entailment relation which is very important for many natural language processing (NLP) applications. Starting from the textual entailment definition, we first identified the difficulty in applying the definition to the lexical level, for which truth values do not apply naturally. We then suggest two definitions which place the lexical entailment relation within the textual entailment framework. These definitions help understanding the intended meaning of the lexical entailment relation and recognizing term pairs for which it holds, represented as lexical entailment rules. The definitions provide an operative method for deciding which rules should be included in a lexical entailment rule base. We utilize these definitions to investigate the utility of several lexical semantic resources as sources for lexical entailment rules.

One of these resources is a new rule base we extract from Wikipedia. This is the first effort to build a resource that is designed to contain lexical entailment rules. This is also one of the first works to explicitly analyze the text of Wikipedia articles. We introduced 8 methods for extracting lexical entailment rules: 5 are based on syntactic analysis of definitions from Wikipedia, and 3 are based on general structural features of a Web knowledge base. The rule base we obtain contains about 8 million rules, which is a magnitude larger than available lexical semantic resources. We thoroughly evaluated the extracted rules both internally, by comparing results to human judgments, and externally, by using it within a real NLP task of unsupervised text categorization. Our evaluations identify the precision

and coverage of each extraction type, allowing applications to choose which rules to use according to their recall-precision tradeoff preference. For instance, almost 70% of the relative recall can be obtained with an accumulative precision of 80%. On the text categorization task, our rule base was shown to perform comparably to WordNet, even though its rules were automatically extracted from texts made for human consumption as opposed to rules manually crafted by experts in WordNet. The union of both resources achieved the best performance on this task.

For the comparative investigation of the available lexical semantic resources (including ours), we developed an evaluation methodology to assess the practical utility of each resource. This methodology extracts text fragments from corpora on which it applies rules from the investigated resource, in order to entail an input hypothesis. It presents these texts to an annotator and through two binary questions, identifies whether a rule application was correct. Our investigation is one of the only comparative evaluation of these state-of-the-art resources, and the first one to be done within the entailment paradigm. It reveals the strengths and weaknesses of each resource, as regards to the task of providing lexical entailment rules. We found that our Wikipedia-based resource to achieve the best recall-precision combination, together with *Lin-dep*, and that it covers various types of the lexical entailment relation.

### **Future Work**

We recognize two directions for future work. The target of the first is to increase recall by discovering new lexical entailment rules from our rule base, while the second aims to increase precision.

Our rule base can be viewed as a graph, whose nodes are the terms participating in its rules, and the directed edges are the rules connecting them. Currently we only extract individual edges as rules, but this can be expanded to extract paths in the graph, hence providing new rules. Graph mining for rules is based on the transitivity of the lexical entailment relation. For instance, given the directed edges (rules) *Adi Shamir*  $\Rightarrow$  *Cryptographer* (extracted by the *Be-Complement* method) and *Cryptographer*  $\Rightarrow$  *Cryptography* (a *Redirect* rule), we can add a new edge (i.e. deducing a new rule) *Adi Shamir*  $\Rightarrow$  *Cryptography*. We can estimate the reliability of a new rule by the precision of the edges we traversed in the process of

assembling it. Another example of a graph path is *RSI*  $\Rightarrow$  *Repetitive Strain Injury*  $\Rightarrow$  *Disorder*  $\Rightarrow$  *Disease*.

Analyzing the errors of applications which use our rule base, we learnt that the richness of knowledge covered by Wikipedia often produces rules with an idiosyncratic meaning, which are correct but will rarely be used in this sense. For example, in the text categorization evaluation, described in Section 5.2, one of the category names is “mac”, meaning the Macintosh computer brand. Our rule base contains many correct left hand sides that entail this term but in a very idiosyncratic sense. Some examples are the name of a sitcom character, some acronyms, e.g. “Musical Arts Conference” and “Mudiad Amddiffyn Cymru” (a Welsh freedom-fighting organization), a novel by that name and more. Having such rules in our rule base harms practical precision. The way to cope with this challenge is to integrate context constraint to our rules. One way to integrate context is by adding a vector with all the nouns of the definition sentence (or the entire first paragraph) as the appropriate context for that rule. Before applying a rule the application will compare this vector with the context suitable for its input and will only apply rules that match the needed context, as in (Szpektor et al., 2008).

### **Additional Initial Findings**

While learning lexical entailment rules from Wikipedia, we identify some promising research directions. We have already put some efforts in following these leads, but as they deviated from our main line of research we did not continue these efforts. Next we briefly review our work on these directions and the initial findings we discovered.

While the focus of this work is the lexical entailment relation which covers a broad range of subclass relations, we recognized that an inference process can benefit from identifying the fine grained classes of lexical entailment. For instance, it is very important to know in which circumstances the two sides of a rule are substitutable. A first step towards this end is utilizing the syntactic paths learned in the process of extracting *All-Nouns* rules. In the work describe earlier these paths are obtained merely as a by product of the unsupervised entailment likelihood score, and are used to divide the list of rules extracted by this method by their predicted accuracy. We noticed that these paths often hold a predicate whose

arguments are the terms of the rule (see Table 5.2 in Section 5.1.1). Such predicates add information about the fine grained semantic relation between the argument terms of the rules. For instance, *Thomas H. Cormen*  $\Rightarrow$  *computer science* is a valid lexical entailment rule, but adding the information that this rule was learnt from the pattern given below assists the inference system to decide when it should be used. For this example, the terms *Thomas H. Cormen* cannot be substituted by *computer science* even though there is a valid entailment relation between these terms, but rather can be substituted by *professor of computer science*. We can learn it from the pattern it was extracted from: *noun* $\xleftarrow{subj}$ *professor* $\xrightarrow{prep}$ *of* $\xrightarrow{prep}$ *noun*.

Natural Type	length of set	Accuracy
Album	29,000	1
Politician	22,000	0.9
Car	2,700	0.65
Film Director	800	0.95
American Film Director	284	1

Table 7.1: Natural types set found in our rule base, their sets length and accuracy evaluation

A second type of fine grained lexical entailment relations were found in the form of Natural Typing (NT). Natural types are naturally phrased entity types. We define a natural type as a phrase that defines a set of terms that bear the same meanings. For instance, “Rolls Royce”, “Jaguar” and “Mercedes ” are members of the set defined by the natural type *Luxury Car*, while only the first two terms suit the natural type *British Luxury Car*. Natural types should be more flexible than common named entity types. The advantage of NTs is that they are naturally phrased and are not restricted to a pre-defined set of types, as defined by named entity recognizers (NER) (e.g. (Sekine et al., 2002) defined 200 entity types<sup>1</sup>). We noticed that many possible NTs serve as right hand side (RHS) of rules in our rule base. Therefore, we can provide exhaustive lists for such NTs by taking their left hand sides (LHS). For instance, there are more than 22,000 terms entailing *Politician*, most of them are indeed names of current and past politician. Even for a more specific NT, such as *American Politician*, we can find 1700 terms that

<sup>1</sup><http://nlp.cs.nyu.edu/ene/>

entail it. However, not all terms entailing a NT are indeed members of the set it define (e.g. *limousine* is not a car model). Our first suggestion for recognizing natural types in our rule base, and extracting their lists, is to follow this process: recognize a RHS as NT if it has a long list of LHS in which the proportion of named entities is high. We ran some preliminary experiments in order to decide what a sufficiently long list and the desired named entity proportion in it, but have not reached conclusive results yet. Table 7.1 presents some natural types found in our rule base, the number of terms in their entailing sets and an accuracy evaluation based on a random sample of 20 instances from each set.

In conclusion, even though our resource is at its first stage of development, it has already become useful and is being integrated in most of the projects of our lab: Text Categorization, the 4<sup>th</sup> Recognizing Textual Entailment challenge, Contextual Preferences, Ontology construction for the medical and archeological domains and Co-reference Resolution. The reason our lexical entailment rule base is used in these projects is that it assists handling the common challenge of lexical variability which is addressed by many natural language processing applications.

# Bibliography

1. Roy Bar-Haim, Ido Dagan, Iddo Grental, and Eyal Shnarch. Semantic inference at the lexical-syntactic level. In *Proceedings of the Twenty-Second AAAI Conference on Artificial Intelligence (AAAI'07)*, pages 871–870, Vancouver, British Columbia, Canada, 2007.
2. Matthew Berland and Eugene Charniak. Finding parts in very large corpora. In *Proceedings of the 37th Annual Meeting of the Association for Computational Linguistics*, pages 57–64, College Park, Maryland, USA, 1999. Association for Computational Linguistics.
3. Sergey Brin. Extracting patterns and relations from the world wide web. In *WebDB Workshop at 6th International Conference on Extending Database Technology, EDBT'98*, 1998.
4. Martin S. Chodorow, Roy J. Byrd, and George E. Heidorn. Extracting semantic hierarchies from a large on-line dictionary. In *Proceedings of the 23rd annual meeting on Association for Computational Linguistics*, pages 299–304, Chicago, Illinois, 1985. Association for Computational Linguistics.
5. Ido Dagan. *Contextual Word Similarity*, chapter 19, pages 459–476. Handbook of Natural Language Processing. Marcel Dekker Inc, 2000.
6. Ido Dagan, Oren Glickman, and Bernardo Magnini. *The PASCAL Recognising Textual Entailment Challenge*, volume 3944 of *Lecture Notes in Computer Science*. Springer, 2005.
7. Ido Dagan, Oren Glickman, Alfio Gliozzo, Efrat Marmorshtein, and Carlo Strapparava. Direct word sense matching for lexical substitution. In *Pro-*

- ceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*, pages 449–456, Sydney, Australia, July 2006. Association for Computational Linguistics.
8. Dmitry Davidov and Ari Rappoport. Efficient unsupervised discovery of word categories using symmetric patterns and high frequency words. In *Proceedings of the 44th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 297–304, Sydney, Australia, 2006. Association for Computational Linguistics.
  9. Manuel de Buenaga Rodríguez, José María Gómez-Hidalgo, and Belén Díaz-Agudo. Using WordNet to complement training information in text categorization. In *Proceedings of RANLP-97, 2nd International Conference on Recent Advances in Natural Language Processing*, pages 150–157, Tzigrav Chark, Bulgaria, 1997.
  10. Christiane Fellbaum, editor. *WordNet: An Electronic Lexical Database*. MIT Press, Cambridge, Massachusetts, 1998.
  11. Charles J. Fillmore, Collin F. Baker, and Hiroaki Sato. Seeing arguments through transparent structures. In *Proceedings of the 3rd International Conference on Language Resources and Evaluation (LREC 2002)*, pages 787–791, Las Palmas, Spain, 2002.
  12. Evgeniy Gabrilovich and Shaul Markovitch. Computing semantic relatedness using wikipedia-based explicit semantic analysis. In *Proceedings of the 20th International Joint Conference on Artificial Intelligence (IJCAI07)*, pages 1606–1611, Hyderabad, India, 2007.
  13. Maayan Geffet and Ido Dagan. Feature vector quality and distributional similarity. In *Proceedings of the 20th international conference on Computational Linguistics (COLING '04)*, pages 247–253, Geneva, Switzerland, 2004. Association for Computational Linguistics.

14. Maayan Geffet and Ido Dagan. The distributional inclusion hypotheses and lexical entailment. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL'05)*, pages 107–114, Ann Arbor, Michigan, June 2005. Association for Computational Linguistics.
15. Danilo Giampiccolo, Bernardo Magnini, Ido Dagan, and Bill Dolan. The third pascal recognizing textual entailment challenge. In *Proceedings of the ACL-PASCAL Workshop on Textual Entailment and Paraphrasing*, pages 1–9, Prague, June 2007. Association for Computational Linguistics.
16. Jim Giles. Internet encyclopedias go head to head. In *Nature*, pages 438: 900–901, 2005.
17. Oren Glickman, Eyal Shnarch, and Ido Dagan. Lexical reference: a semantic matching subtask. In *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing*, pages 172–179, Sydney, Australia, July 2006. Association for Computational Linguistics.
18. Ralph Grishman, Lynette Hirschman, and Ngo Thanh Nhan. Discovery procedures for sublanguage selectional patterns: Initial experiments. *Computational Linguistics*, 12(3):205–215, 1986.
19. Nizar Habash and Bonnie Dorr. A categorial variation database for english. In *Proceedings of the North American Association for Computational Linguistics (NAACL '03)*, pages 96–102, Edmonton, Canada, 2003. Association for Computational Linguistics.
20. Sanda M. Harabagiu, Dan Moldovan, Marius Paşca, Rada Mihalcea, Mihai Surdeanu, Razvan C. Bunescu, Roxana Gîrji, Vasile Rus, and Paul Morărescu. Falcon: Boosting knowledge for answer engines. In *Proceedings of the ninth text retrieval conference (TREC-9)*, Gaithersburg, Maryland, 2000. NIST.
21. Zelig S. Harris. *Mathematical Structures of Language*. New York, 1968.
22. Marti A. Hearst. Automatic acquisition of hyponyms from large text corpora. In *Proceedings of the 14th conference on Computational linguistics*, pages 539–545, Nantes, France, 1992. Association for Computational Linguistics.

23. Eduard H. Hovy, Ulf Hermjakob, and Chin-Yew Lin. The use of external knowledge of factoid qa. In *Proceedings of the ninth text retrieval conference (TREC-10)*. NIST, 2001.
24. Nancy Ide and Véronis Jean. Extracting knowledge bases from machine-readable dictionaries: Have we wasted our time? In *Proceedings of KB & KS'93 Workshop*, pages 257–266, Tokyo, 1993.
25. Jun'ichi Kazama and Kentaro Torisawa. Exploiting Wikipedia as external knowledge for named entity recognition. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pages 698–707, 2007.
26. Youngjoong Ko and Jungyun Seo. Learning with unlabeled data for text categorization using a bootstrapping and a feature projection technique. In *Proceedings of the 42nd Meeting of the Association for Computational Linguistics (ACL'04), Main Volume*, pages 255–262, Barcelona, Spain, July 2004.
27. J. Richard Landis and Gary G. Koch. The measurements of observer agreement for categorical data. In *Biometrics*, pages 33:159–174, 1997.
28. Dekang Lin. Dependency-based evaluation of minipar. In *Proceedings of the Workshop on Evaluation of Parsing Systems at LREC*, 1998.
29. Dekang Lin. Automatic retrieval and clustering of similar words. In *Proceedings of the 17th international conference on Computational linguistics*, pages 768–774, Montreal, Quebec, Canada, 1998. Association for Computational Linguistics.
30. Bing Liu, Xiaoli Li, Wee Sun Lee, and Philip S. Yu. Text classification by labeling words. In *Proceedings of the American Conference of Artificial Intelligence*, pages 425–430, 2004.
31. Catherine Macleod, Ralph Grishman, Adam Meyers, Leslie Barrett, and Ruth Reeves. Nomlex: A lexicon of nominalizations. In *Proceedings of the Eighth*

- International Congress of the European Association for Lexicography*, pages 187–193, Liege, Belgium, 1998.
32. Andrew McCallum and Kamal Nigam. Text classification by bootstrapping with keywords, em and shrinkage. In *Proceedings of the ACL Workshop for unsupervised Learning in Natural Language Processing*, pages 52–58, 1999.
  33. George A. Miller, Richard Beckwith, Christiane Fellbaum, Derek Gross, and Katherine J. Miller. Introduction to wordnet: An on-line lexical database. *Journal of Lexicography*, 3(4):235–244, 1990.
  34. Shachar Mirkin, Ido Dagan, and Maayan Geffet. Integrating pattern-based and distributional similarity methods for lexical entailment acquisition. In *Proceedings of the COLING/ACL 2006 Main Conference Poster Sessions*, pages 579–586, Sydney, Australia, 2006. Association for Computational Linguistics.
  35. Dan I. Moldovan and Vasile Rus. Logic form transformation of wordnet and its applicability to question answering. In *Proceedings of the 39th Annual Meeting on Association for Computational Linguistics*, pages 402–409, Toulouse, France, 2001. Association for Computational Linguistics.
  36. Eric Nichols, Francis Bond, and Daniel Flickinger. Robust ontology acquisition from machine-readable dictionaries. In *Proceedings of the International Joint Conference on Artificial Intelligence IJCAI-2005*, pages 1111–1116, Edinburgh, 2005.
  37. Patrick Pantel and Dekang Lin. Discovering word senses from text. In *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 613–619, 2002.
  38. Patrick Pantel and Deepak Ravichandran. Automatically labeling semantic classes. In *Proceedings of HLT-NAACL*, pages 321–328, Boston, Massachusetts, USA, 2004. Association for Computational Linguistics.

39. Patrick Pantel, Deepak Ravichandran, and Eduard Hovy. Towards terascale semantic acquisition. In *Proceedings of Coling 2004*, pages 771–777, Geneva, Switzerland, 2004. COLING.
40. Simone Paolo Ponzetto and Michael Strube. Deriving a large scale taxonomy from wikipedia. In *Proceedings of the 22nd National Conference on Artificial Intelligence (AAAI-07)*, pages 1440–1447, Vancouver, B.C., 2007.
41. Deepak Ravichandran and Eduard Hovy. Learning surface text patterns for a question answering system. In *Proceedings of 40th Annual Meeting of the Association for Computational Linguistics*, pages 41–47, Philadelphia, Pennsylvania, USA, 2002. Association for Computational Linguistics.
42. Ellen Riloff and Rosie Jones. Learning dictionaries for information extraction by multi-level bootstrapping. In *Proceedings of the Sixteenth National Conference on Artificial Intelligence (AAAI'99)*, pages 474–479, Orlando, Florida, United States, 1999. American Association for Artificial Intelligence.
43. 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, June 2007. Association for Computational Linguistics.
44. Sam Scott and Stan Matwin. Text classification using WordNet hypernyms. In *Proceedings of the Workshop on Usage of WordNet in Natural Language Processing Systems (Coling-ACL'98)*, pages 45–51, Montreal, Canada, 1998. Association for Computational Linguistic.
45. Satoshi Sekine, Kiyoshi Sudo, and Chikashi Nobata. Extended named entity hierarchy. In *Proceedings of 3<sup>rd</sup> International Conference on Language Resources and Evaluation (LREC'02)*, pages 1818–1824, Canary Islands, Spain, 2002.
46. Rion Snow, Daniel Jurafsky, and Andrew Y. Ng. Learning syntactic patterns for automatic hypernym discovery. In Lawrence K. Saul, Yair Weiss, and Léon Bottou, editors, *Advances in Neural Information Processing Systems 17*, pages 1297–1304. MIT Press, Cambridge, MA, 2005.

47. Rion Snow, Daniel Jurafsky, and Andrew Y. Ng. Semantic taxonomy induction from heterogenous evidence. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*, pages 801–808, Sydney, Australia, July 2006. Association for Computational Linguistics.
48. Fabian M. Suchanek, Gjergji Kasneci, and Gerhard Weikum. Yago: A core of semantic knowledge - unifying WordNet and Wikipedia. In *Proceedings of the 16th International World Wide Web Conference (WWW07)*, pages 1440–1447, Banff, Canada, 2007.
49. Idan Szpektor, Eyal Shnarch, and Ido Dagan. Instance-based evaluation of entailment rule acquisition. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pages 456–463, Prague, Czech Republic, June 2007. Association for Computational Linguistics.
50. Idan Szpektor, Ido Dagan, Roy Bar-Haim, and Jacob Goldberger. Contextual preferences. In *Proceedings of ACL-08: HLT*, pages 683–691, Columbus, Ohio, June 2008. Association for Computational Linguistics.
51. Antonio Toral and Rafael Muñoz. A proposal to automatically build and maintain gazetteers for named entity recognition by using wikipedia. In *Proceedings of Workshop on New Text, 11th Conference of the European Chapter of the Association for Computational Linguistics NAACL/HLT*, Trento, Italy, 2007.
52. Ellen M. Voorhees. Query expansion using lexical-semantic relations. In *Proceedings of the 17th Annual International ACM-SIGIR Conference on Research and Development in Information Retrieval*, pages 61–69, Dublin, Ireland, 1994.
53. Julie Weeds, David Weir, and Diana McCarthy. Characterising measures of lexical distributional similarity. In *Proceedings of the 20th international conference on Computational Linguistics (COLING '04)*, pages 1015–1021, Geneva, Switzerland, 2004. Association for Computational Linguistics.

54. Yorick A. Wilks, Brian M. Slator, and Louise M. Guthrie. *Electric words: dictionaries, computers, and meanings*. MIT Press, Cambridge, MA, USA, 1996.
55. Torsten Zesch, Iryna Gurevych, and Max Mühlhäuser. *Analyzing and Accessing Wikipedia as a Lexical Semantic Resource*, pages 197–205. *Data Structures for Linguistic Resources and Applications*. Gunter Narr, Tübingen, Tuebingen, Germany, 2007.