Grammatical Functions and Possibilistic Reasoning for the Extraction and Representation of Semantic Knowledge in Text Documents

by

Richard Khoury

A thesis
presented to the University of Waterloo
in fulfillment of the
thesis requirement for the degree of
Doctor of Philosophy
in
Electrical and Computer Engineering

Waterloo, Ontario, Canada, 2007

I hereby declare that I am the sole author of this thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

Abstract

This study seeks to explore and develop innovative methods for the extraction of semantic knowledge from unlabelled written English documents and the representation of this knowledge using a formal mathematical expression to facilitate its use in practical applications.

The first method developed in this research focuses on semantic information extraction. To perform this task, the study introduces a natural language processing (NLP) method designed to extract information-rich keywords from English sentences. The method involves initially learning a set of rules that guide the extraction of keywords from parts of sentences. Once this learning stage is completed, the method can be used to extract the keywords from complete sentences by pairing these sentences to the most similar sequence of rules. The key innovation in this method is the use of a part-of-speech hierarchy. By raising words to increasingly general grammatical categories in this hierarchy, the system can compare rules, compute the degree of similarity between them, and learn new rules.

The second method developed in this study addresses the problem of knowledge representation. This method processes triplets of keywords through several successive steps to represent information contained in the triplets using possibility distributions. These distributions represent the possibility of a topic given a particular triplet of keywords. Using this methodology, the information contained in the natural language triplets can be quantified and represented in a mathematical format, which can be easily used in a number of applications, such as document classifiers.

In further extensions to the research, a theoretical justification and mathematical development for both methods are provided, and examples are given to illustrate these notions. Sample applications are also developed based on these methods, and the experimental results generated through these implementations are expounded and thoroughly analyzed to confirm that the methods are reliable in practice.

Acknowledgements

First and foremost, I would like to express my deep gratitude towards Professor Fakhreddine Karray, for his availability and his help during my years as a doctoral student. By his good counsel, he has guided me through all the stages of this work.

I would also like to give my sincerest thanks to the members of my committee, Professors Krzysztof Czarnecki, Chrysanne DiMacro, Keith Hipel, and Gerald Penn, for their very valuable and constructive suggestions which contributed to improve the quality of my thesis.

Most of all, I would especially like to thank my parents for their support and encouragement.

Table of Contents

List of Figures

List of Tables

List of Symbols

Symbol	Meaning	First Use
$\alpha(n_i, w_k)$	Correlation between noun n_i and word w_k.	Equation (5-3)
$\alpha(v_j, w_k)$	Correlation between verb v_j and word w_k.	Equation (5-3)
$\alpha(w_i, w_j)$	Correlation between words w_i and w_j.	Equation (5-2)
Δ_i	Information gain of noun n_i.	Equation (3-1)
Δ'_i	Information gain of a noun sub-category.	Section 3.2.3
Δ_{ij}	Pointwise mutual information of nouns n_i and n_j.	Equation (3-2)
δ	Semantic distance.	Figure 5-11
$\delta(n_i, v_j)$	Semantic distance between noun n_i and verb v_j.	Equation (5-3)
$\overline{\delta}_i$	Average semantic distance of the noun-verb pairs using noun n_i.	Section 5.3.1
δ_{ij}	Alternative writing of $\delta(n_i, v_j)$.	Section 5.2.4
δ_{ik}	Semantic distance between noun n_i and verb v_k.	Section 5.4
δ_M	Maximum possible semantic distance between a noun-verb pair.	Equation (5-25)
δ_m	Minimum possible semantic distance between a noun-verb pair.	Equation (5-25)
λ	Linguistic probability value of a statement.	Equation (5-14)
$\Pi_{d_t n_i}$	Possibility distribution of d_t given n_i.	Figure 5-10
$\Pi_{P(X \text{ is } F)}$	Possibility distribution of the probability distribution of F.	Equation (5-14)
Π_X	Possibility distribution of variable X.	Equation (5-12)
$\pi(d_t)$	Possibility of domain d_t given an entire testing document.	Equation (6-2)
π_1	Possibility of E_1 occurring.	Section 5.4
$\pi_1(d_1)$	Possibility of d_1 given LO_1.	Figure 6-2
$\pi_1(d_2)$	Possibility of d_2 given LO_1.	Figure 6-2
$\pi_2(d_1)$	Possibility of d_1 given LO_2.	Figure 6-2
$\pi_2(d_2)$	Possibility of d_2 given LO_2.	Figure 6-2

π_M	Possibility of the domain with the highest possibility.	Equation (6-3)
π_{M-1}	Possibility of the domain with the second-highest possibility.	Equation (6-4)
$\pi_{n_iv_j}(d_t)$	Possibility of d_t given n_iv_j.	Section 5.4
$\pi^k_{n_iv_j}(d_t)$	Possibility of d_t given n_iv_j as perceived at δ_{ik}.	Equation (5-25)
$\pi_{n_iv_j \cup n_iv_k}(d_t)$	Possibility of d_t given either n_iv_j or n_iv_k.	Equation (5-24)
$\pi_{n_iv_j \cap n_kv_j}(d_t)$	Possibility of d_t given both n_iv_j and n_kv_j.	Equation (5-29)
$\pi_{n_iv_jn_k}(d_t)$	Possibility of d_t given $n_iv_jn_k$.	Equation (5-29)
$\pi_{n_iv_k}(d_t)$	Possibility of d_t given n_iv_k.	Equation (5-24)
$\pi'_{n_iv_k}(d_t)$	Updated possibility of d_t given n_iv_k.	Equation (5-26)
$\pi_{n_kv_j}(d_t)$	Possibility of d_t given n_kv_j.	Equation (5-29)
$\pi_{P(n_iv_j \text{ is } d_t)}$	Possibility of the probability that the observed pair n_iv_j belongs to d_t.	Section 5.4
π_X	Possibility distribution function corresponding to Π_X.	Section 5.3.1
$\pi_X(u)$	Possibility that $X = u$.	Section 5.3.1
$\pi_z(d_t)$	Possibility of domain d_t given the triplet τ_z.	Equation (6-1)
ρ	Parameter of the discourse in the Zipf-Mandelbrot law.	Equation (5-1)
σ_{δ_i}	Standard deviation of the semantic distance of all pairs using noun n_i.	Section 5.3.1
τ_1	Triplet 1.	Table 6-6
τ_2	Triplet 2.	Table 6-6
τ_3	Triplet 3.	Table 6-6
τ_z	z^{th} triplet of the test document.	Section 6.3.3
$AP(n_i, d_t)$	Normalized average probability of all noun-verb using noun n_i pairs in a region of the plot of domain d_t.	Section 5.3.1
B	Parameter of the discourse in the Zipf-Mandelbrot law.	Equation (5-1)
C	Confidence of the classification.	Equation (6-5)
C_C	Confidence of the documents classified correctly.	Table 6-7

C_I	Confidence of the documents classified incorrectly.	Table 6-7
d_1	Domain 1.	Table 6-6
d_2	Domain 2.	Table 6-6
d_3	Domain 3.	Table 6-6
d_k	Domain k in the training corpus.	Never used
d_t	Domain t in the training corpus.	Section 3.2.3
E_1	An event.	Section 5.4
E_2	An event.	Section 5.4
e_{it}	Expected number of occurrences of word w_i in domain d_t.	Equation (5-2)
e_{jt}	Expected number of occurrences of word w_j in domain d_t.	Equation (5-2)
F	A fuzzy subset of U.	Section 5.3.1
f	Frequency of a word in the Zipf-Mandelbrot law.	Equation (5-1)
FN	False negative (untagged keyword).	Section 4.2
FP	False positive (tagged non-keyword).	Section 4.2
G	Neighbourhood size.	Equation (5-25)
H_i	Total number of occurrences of the word w_i in the training corpus.	Section 5.2.4
L	Length of the training corpus, or in other words the number of triplets representing the training corpus.	Section 5.2.4
LO_1	Learning object 1.	Figure 6-2
LO_2	Learning object 2.	Figure 6-2
l_k	Length of domain d_k, or in other words the number of triplets representing domain d_k.	Equation (5-9)
l_t	Length of domain d_t, or in other words the number of triplets representing domain d_t.	Section 5.2.4
N_C	Number of documents classified correctly.	Table 6-7
N_I	Number of documents classified incorrectly.	Table 6-7
N_{it}	Number of occurrences of noun sub-category i in domain d_t.	Section 3.2.3
N_{Sim}	Number of elements contained within V_{Sim}.	Equation (5-20)
N_v	Number of different verbs in the training corpus.	Equation (5-16)
N_w	Number of different words in the training corpus.	Equation (5-3)

n_i	Noun i.	Section 3.2.3
n_{it}	Number of occurrences of noun n_i in domain d_t.	Equation (3-1)
n_iv_j	Pair composed of noun n_i and verb v_j.	Section 5.2.5
$n_iv_jn_k$	Triplet composed of noun n_i as subject, verb v_j, and noun n_k as object.	Section 5.6
n_iv_{jt}	Number of occurrences of pair n_iv_j in domain d_t.	Equation (5-8)
n_iv_k	Pair composed of noun n_i and verb v_k.	Section 5.4
n_j	Noun j.	Section 3.2.3
n_{jt}	Number of occurrences of noun n_j in domain d_t.	Equation (3-2)
n_k	Noun k.	Section 5.6
n_kv_j	Pair composed of noun n_k and verb v_j.	Section 5.6
P	Parameter of the discourse in the Zipf-Mandelbrot law.	Equation (5-1)
$P(d_t)$	Prior, probability of domain d_t.	Equation (5-4)
$P(d_t \mid n_iv_j)$	Posterior, probability of domain d_t given the pair n_iv_j.	Equation (5-4)
$P(d_t \mid w_iw_j)$	Probability of domain d_t given the pair w_i and w_j.	Equation (5-9)
$P(n_iv_j)$	Normalizing constant, probability of the pair n_iv_j.	Equation (5-4)
$P(n_iv_j \mid d_t)$	Likelihood function, probability of the pair n_iv_j given domain d_t.	Equation (5-4)
$P(n_iv_j$ is $d_t)$	Probability that the observed pair n_iv_j belongs to d_t.	Section 5.4
$P(n_iv_jd_t)$	Frequency of the pair n_iv_j in domain d_t divided by the frequency of n_iv_j in the entire corpus.	Equation (5-20)
$P(X$ is $F)$	Probability of the statement "X is F"	Equation (5-14)
p	A discrete probability value.	Section 5.4
r	Rank of a word in the Zipf-Mandelbrot law.	Equation (5-1)
$R(X)$	A fuzzy restriction on the variable X.	Equation (5-11)
RF_j	Relative frequency of verb v_j.	Section 5.3.1
STD	Standard deviation.	Table 6-11
T	Total number of domains in the training corpus.	Equation (5-2)
TP	True positive (correctly-marked keyword).	Section 4.2
u	A value in U.	Section 5.3.1
U	A universe of discourse.	Section 5.3.1

v_j	Verb j.	Section 5.2.4
v_{jt}	Number of occurrences of verb v_j in domain d_t.	Equation (5-6)
V_{Sim}	Verbs whose semantic distance with noun n_i is within one standard deviation σ_{δ_i} of the average semantic distance $\overline{\delta}_i$.	Equation (5-20)
VW_j	Verb weight of verb v_j.	Section 5.3.1
w_1	Domain-specific word originating from domain d_1.	Section 5.8
w_2	Domain-specific word originating from domain d_2.	Section 5.8
w_3	Domain-specific word originating from domain d_1.	Section 5.8
w_E	Extraction error occurring in domain d_1.	Section 5.8
w_F	Extraction error occurring in a domain other than d_1.	Section 5.8
w_G	General word.	Section 5.8
w_i	Word i.	Equation (5-2)
w_{it}	Number of occurrences of word w_i in domain d_t.	Equation (5-2)
w_{ik}	Number of occurrences of word w_i in domain d_k.	Equation (5-9)
w_j	Word j.	Equation (5-2)
w_{jt}	Number of occurrences of word w_j in domain d_t.	Equation (5-2)
w_{jk}	Number of occurrences of word w_j in domain d_k.	Equation (5-9)
w_k	Word k.	Equation (5-3)
X	A variable in U.	Section 5.3.1
Z	Number of triplets in a testing document.	Equation (6-2)

Chapter 1
Introduction

1.1 Problem Background and Motivation

In 2002, laboratories from six Canadian universities, including the Pattern Analysis and Machine Intelligence (PAMI) laboratory at the University of Waterloo, teamed up to work on the Learning Objects Repositories Network (LORNET) project. This project aims to develop the Semantic Web: the next generation of the Web in which information will be organized semantically, or by meaning [9]. The specific focus of the PAMI lab within this project is on the development of knowledge extraction and representation techniques, and the implementation of software that operates based on these techniques. Once integrated into the overall LORNET framework, these tools will enable the development of software that semantically organizes, navigates through, and retrieves information.

Our research is part of the PAMI lab contribution to the LORNET project. Our purpose is to develop two new Natural Language Processing (NLP) methods, one designed to extract semantic knowledge from otherwise untagged written English documents, and the other to represent this knowledge using a formal mathematical expression. These methods will facilitate the use of this knowledge in practical applications such as document classifiers and search engines. As this course of study will shown, these methods provide flexible frameworks that can be taken separately or together and adapted to fit the needs of any aspect of the LORNET project.

1.2 Definitions

To establish a clear framework for our study, we will define from the outset a number of expressions that play an important role in the study.

In English, a sentence in its simplest form is composed of two parts: the subject and the predicate [63]. The subject is typically a noun phrase, while the predicate is a verb phrase that contains a verb and zero or more objects. This gives rise to the *subject-verb-object*

sentence structure, which is fundamental in this thesis. Using this structure causes us to discard a lot of information such as the semantic distinction between the agent and patient of the verb, and the grammatical distinction between the various types of objects. However, if the methods presented in this dissertation are proven to be effective using only simple subject-verb-object information, future work can build upon and refine these methods by reintroducing the discarded grammatical and semantic information. In our work, the subject-verb-object structure is represented by a triplet of keywords, which we sometimes refer to simply as the *triplets*.

A *corpus* (plural: *corpora*) is a collection of texts of any nature. Two different kinds of corpora will be used in this thesis, namely a training corpus and a testing corpus. The *training corpus* is a set of texts that a program can use to discover patterns, rules or statistical features. The *testing corpus*, on the other hand, is used to test a system after it has been trained.

A *domain* is a part of the training corpus. It can comprise either a long text or a collection of short texts that have been merged. A domain represents a topic of discourse such as the medical domain or the business domain. In this thesis, the word "domain" can refer to either the original English text, the corresponding list of triplets extracted from it, or the topic it represents.

A *document*, or *testing document*, is a part of the testing corpus. Documents are classified into a domain according to their content. In this thesis, "document" refers to either the original English text or the corresponding list of triplets extracted from it.

Polysemy refers to the situation in which one word has several different meanings. For example, the word "bank" is polysemous, since it can refer to a financial institution or the rising ground bordering a body of water. Polysemous words are difficult to understand in isolation, but their meaning usually becomes clear when they are interpreted in their given context. Polysemous words therefore pose a major problem for applications of natural language processing that ignore or discard contextual information.

The *data sparseness problem*, also referred to in the literature as the *sparse data problem* and the *rare events problem*, is an important challenge in natural language processing. This problem is encountered when researchers extrapolate the probability of observing pairs of words (or triplets, or more) in reality, based on their frequency of occurrence in a training corpus. Because the training corpus has a limited size, it contains a finite number of word pairs. Pairs that are not present have a frequency of zero. Assigning zero probabilities is erroneous, as many of the pairs are not impossible, but are very rare. Instinctively, the solution to this problem might be to augment the size of training corpus to include a representative sample of the missing pairs. However, researchers in statistical NLP have shown that all representative corpora have this problem; there are simply too many rare events [113], [55]. This observation has led to the development of a number of algorithms to make the probabilities more accurate, such as Laplace's law [48], Lidstone's law of successions [51], the Expected Likelihood Estimation [12], the held out estimator [43], and the famous Good-Turing estimation [35]. The data sparseness problem becomes more severe when longer sequences of words are studied. This is because a given training corpus contains more word pairs than triplets of words. Therefore, the sample of pairs is normally more complete than the sample of triplets, and the probabilities extrapolated for the pairs will be more reliable. Likewise, the corpus will contain more triplets of words than quadruplets of words, more quadruplets than quintuplets, and so on. A good presentation of the data sparseness problem, as well as of some popular statistical smoothing techniques used to compensate for it, can be found in Manning and Schütze [56].

Possibility theory was proposed by L. A. Zadeh as a middle-ground between fuzzy set theory and probability theory [110]. In that seminal paper, he makes the case that possibilities are related to fuzzy sets through fuzzy restrictions, and weakly connected to probabilities through the possibility/probability consistency principle [110]. He also argues that, while probability theory is unquestionably useful for measuring information, the proper framework to handle the meaning of information should instead be possibility theory. For this reason, we used possibility theory as the mathematical foundation for our work in Chapter 5. Consequently, we will present a more complete introduction to the theory later in this thesis.

1.3 Previous Work and Evolution

The basic concept of this research has developed from previous work by the PAMI Lab in the field of fuzzy natural language understanding (NLU) [84], [85]. This research had culminated in the development of a three-stage approach to NLU and implementation in a prototype system [85]. The first stage of this approach consists of obtaining a semantic representation of the sentences in the text being analyzed. This is achieved by matching each sentence to one of the semantic patterns defined in the system. These patterns are created manually, and have a format similar to "{company | person} acquire {company | person | financial-instrument} {for | at} money". As this example shows, the semantic patterns are specialized not only to specific domains but also to specific actions in each domain. The formats of the patterns are moreover based on the assumption that the sentences of the text have simple, straightforward structures. The approach of [85] also requires, at this first stage, the creation of a custom-built dictionary to match the words in the sentence to the elements identified in the pattern, so as to distinguish, for example, a company from a person. At the second stage, the approach of [85] computes statistical information about the elements detected previously in the process of matching the patterns to the sentences in the first stage. More specifically, the statistics computed at this stage are the semantic distance between each pair of elements, as well as the frequency of occurrence of each of these pairs. Finally, membership functions are constructed in the third stage. These functions represent typical sentences for each action in each domain. The statistics computed in the previous stage are used here for the construction of the membership functions. In addition, rules are defined to assign a specific weight to each statistic based on which elements originated from which pattern in the first stage.

Even though the general idea of our research traces its roots to the project described above, it becomes quite evident on closer inspection that this thesis presents considerable changes and makes important intellectual improvements to the original approach. Further insight into the evolution of our research from [84] to its final state presented in this thesis comes from our work in [45]. At that point, the project was between these two stages of development and illustrated the first innovations that laid the foundation for our subsequent expansion in this

study. First, and of great importance in this regard, is the fact that the domain-specific semantic patterns of the first stage of the original approach in [84] have been eliminated and replaced in [45] with general heuristics designed to isolate the nouns and verbs in the sentences. Later on, motivated by a desire to make the approach more practical, these heuristics were further refined into the triplet extraction method illustrated in Chapter 3 of this thesis. Secondly, of the two statistics computed in the second stage of the early work, only the semantic distances have been retained in [45], while the frequency of occurrences have been replaced by conditional probabilities to take into account the different lengths of the documents we can work with. Thirdly, the fuzzy membership functions found in [84] have been drastically modified in our version of the method presented in [45]. Whereas these functions were designed to model typical sentences in [84], they have been transformed in [45] to represent instead the membership degree of noun-verb pairs in each domain. As a result, mathematical computation of the membership function was completely overhauled to eliminate the idea of rules in favour of a set of equations (covered in Section 5.3 of this thesis).

Drawing on these three main improvements, our research moved on to develop, most notably, a keyword extraction method using a part of speech hierarchy that is more versatile than the heuristics-based method of Chapter 3, and a framework founded on possibility theory that is more mathematically rigorous than the fuzzy set method it replaced. Putting all these features together led to the emergence of an integrated approach for knowledge extraction and representation, which is expounded and successfully tested in this thesis.

1.4 Organization of the Dissertation

With the above concepts now defined, this study takes the current research on natural language processing a step further by developing new methods for the extraction and representation of actions in text documents. Accordingly, Chapter 2 provides an overview of the existing literature in the field of natural language processing that deals specifically with the extraction of semantic information from documents. Although the information extraction

techniques reviewed work well under the specific conditions for which they were designed, our analysis shows that their applicability is too limited for the purpose of this study.

The methods developed in this research present several key features. For clarity and ease of presentation, the research is subdivided into two main phases, knowledge extraction and knowledge representation. The knowledge extraction phase is the subject matter of Chapters 3 and 4. The discussion in these two chapters focuses on the methodological processes developed to extract from the input documents a list of word triplets that represent the information contained in those documents. Chapter 3 presents the first of two triplet extraction processes. The analysis and experimental results reveal that this process, while simple and functional, bears a number of shortcomings that severely limit its applicability. Consequently, Chapter 4 suggests a second and more robust triplet extraction method. The subsequent analysis of this method and the experimental results generated from it demonstrate its reliability and usefulness. The keyword extraction method described in this chapter is one of the key contributions of this research.

The knowledge representation phase of our research is investigated in Chapters 5 and 6. To begin with, a method developed for this purpose is presented in Chapter 5, along with its mathematical development. This method of knowledge representation is another contribution of this study. Following this method, the triplets undergo several successive processing steps to represent the information they contain using possibility distributions. To provide more insight into the structure of the system, each step is detailed with relevant concepts, theoretical justification, and the mathematical development of that step. And as befits a new approach, we verify its validity by conducting a detailed study of the mathematical principles underlying it. The mathematical foundation of each step of the method is examined in this chapter, and parallels between these theoretical outcomes and the practical results of Chapter 6 are drawn. An example is also supplied to illustrate the practical operation of the system. Chapter 6 brings additional insight to the discussion with five different practical applications that use the proposed method. The first application is a domain classifier specifically devised to illustrate the implementation process step by step. Several tests are conducted with this classifier and the results generated are thoroughly analyzed. The discussion then moves on to

an overview of four more possible practical applications that could be implemented with the same method, including two that were designed to tackle problems encountered in other areas of the LORNET project.

To sum up the main features of the new methods for the extraction and representation of knowledge developed in this study, Chapter 7 gives a brief retrospective of the main line of reasoning of this research and suggests directions for future work.

1.5 Overview of the Contributions

The key contributions to the field of natural language processing that will emerge from this research are as follows. The first contribution is the design of a part-of-speech hierarchy. This hierarchy is the foundation of the rule-learning and knowledge extraction method described in Chapter 4.

The second main contribution, as detailed in Chapter 5, is to support the theory that most of the important semantic information of a sentence is contained in actions, i.e. subject-verb-object triplets. In the same vein, our third main contribution is the mathematical development of possibility distributions as a means to represent this action-based semantic information.

The fourth contribution is the implementation of our theoretical methods in practical applications. The experimental results obtained from these implementations confirm that the methods will be reliable in practice outside of our study.

1.6 Conclusions

The motivation for this thesis is to develop reliable methods for the purpose of extracting and representing semantic knowledge in texts. Two methods are developed and investigated in this thesis, one for knowledge extraction in Chapters 3 and 4, and the other for knowledge representation in Chapters 5 and 6.

Chapter 2
Literature Review and Background

2.1 Introduction

The loss of semantic information has been a limiting factor in many applications of natural language processing (NLP). This is the case for information retrieval, text summarization and document translation. Consequently, several researchers have focused on the challenge of extracting semantic information from text documents, and have come up with interesting solutions. The methods underlying these solutions have evolved from the purely statistical methods advocated originally [50] to the methods of soft computing and machine learning used later. This chapter presents a survey of some of the most significant developments in this field of research. As befits this scientific domain, the discussion begins with a general overview of the field with emphasis on the philosophical shift from statistical to fuzzy NLP. This chapter then moves on to outline the topic from a different perspective, that of contrasting statistical NLP and symbolic NLP. The analysis then proceeds to recent developments in two particular areas of application of NLP methods, namely semantic tagging and domain classification. Finally, the chapter concludes with a general outline of the method we have developed to extract and represent semantic information from text documents.

2.2 Overview of Natural Language Processing

2.2.1 Statistical Natural Language Processing

Modern studies involving text representation often rely on a statistical language model such as the document vector used in [54], [99], [61]. The assumption underlying this approach as well as other similar approaches is that any text document can adequately be represented by a document vector, which contains the frequency count of words selected as the most significant by a certain metric [1]. These approaches have been labelled *bag-of-words* techniques because in a sense, they treat the document as if it were a bag of words: it is

assumed that the order of the words is irrelevant and only their occurrence frequency matters. Despite their successes, document vector approaches have been criticized for numerous shortcomings, such as their inability to recognize words with similar or related meanings [61], [88]. For example, this method would fail to recognize that a document containing the words “learner” and “pupil” is similar to a document containing “student”.

To overcome the limitations of the bag-of-words approach, some NLP researchers have enhanced it by disambiguating the semantic meaning of words using a variety of machine-readable dictionaries such as WordNet [88], [22], [40], the Oxford Advanced Learner’s Dictionary [50], [7], the Longman Dictionary of Contemporary English [47], or the Funk and Wagnalls Dictionary [95]. The first, and arguably the most well-known, of these semantic techniques became known as Lesk’s algorithm, after its creator [50]. Lesk proposed to compare the dictionary definitions of the words in the text. In the case of words with several definitions, the correct one is assumed to be the one with the most terms in common with the definitions of other words. In essence, instead of treating the document as a bag-of-words, Lesk replaces the words of the document with their dictionary definitions and treats each of those definitions as a bag-of-words. Lesk’s simple setup can correctly classify on average 50% of the ambiguous words it encounters, but when enhanced with collocates or specialized dictionaries such as those mentioned above, Lesk’s algorithm can reach 70% accuracy.

Alternatively, some researchers have tried to incorporate semantic knowledge into their systems by using predefined syntactic patterns [49], [79], [80]. The basic method consists in matching a sentence to a pattern, then using words from that sentence to fill slots in the pattern [49]. A set of conditions is defined for each slot and must be met by the word considered for that slot. More advanced techniques to identify the patterns and the conditions take one sentence as an initial strict pattern and generalize it to encompass similar sentences in a training corpus [79], [80]. However, the use of predefined patterns implies a very precise *a priori* knowledge of the syntactic structures utilized in the document being analyzed, and as such is inherently limited to a domain-specific approach.

To represent syntactic and semantic information despite the limitations of domain-specific approaches such as the afore-mentioned syntactic patterns, some studies have explored the possibility of modelling the relationship between individual words [103], [104], [105], [106]. Indeed, modelling the relationships between all parts of a sentence may require specialized knowledge, but the semantic relationships between individual nouns, adjectives and adverbs are mostly independent of domain. However, because of the fuzziness present in human knowledge and consequently in the semantics of human languages [103], as in the case of English, word relationships are rarely governed by clear, crisp rules, and so are difficult to represent statistically. For this reason, researchers have turned their attention to other approaches. One of the most notable among these, and the one most related to our work, is fuzzy-based NLP.

2.2.2 Fuzzy-Based Natural Language Processing

In his paper on the foundations of fuzzy logic, Zadeh states that fuzzy logic has four major facets [101]. Its first and most common facet is called set-theoretic. In this perspective, fuzzy logic is used to define fuzzy sets, or sets with non-crisp boundaries. As a matter of fact, this aspect of fuzzy logic is the first one on which researchers have focused on when they initially explored this field, and it is the one on which most current applications of fuzzy logic rely. Secondly, fuzzy logic can be seen as a multiple-valued logical system with its own set of inference rules. These rules allow it to represent and manipulate information that is uncertain or partially true. A third aspect of fuzzy logic is its epistemic facet, which is related to the logical-system perception. This facet focuses rather on knowledge applications such as knowledge representation and information systems, where the knowledge is incomplete or uncertain. Finally, there is a relational aspect to fuzzy logic, which is mainly concerned with the representation and manipulation of fuzzy relations. It is this last facet that deals with the imprecise or uncertain relationships that exist between crisp or fuzzy elements. In practice, these fuzzy relations are most commonly represented as fuzzy if-then rules. This ability to deal with imprecision and approximate reasoning is intrinsic to fuzzy logic, and more importantly the usefulness of this characteristic in handling uncertain relationships makes

fuzzy logic an ideal tool to model and handle the vagueness and uncertainty of natural languages [102].

In early attempts to apply fuzzy logic to the field of NLP, researchers focused on modelling the impact of adverbs on the meaning of associated words in a sentence using fuzzy sets. A good deal of this work was pioneered by Zadeh. His initial research showed that adverbs could be treated as operators acting on fuzzy sets [103]. He went on to define a series of fuzzy operations that could be implemented using adverbs and laid the theoretical groundwork needed to match specific adverbs to the appropriate operations. These accomplishments were the first steps towards fuzzy representation and semantic evaluation of a composite sentence, but they were limited to representing adverbs. Representing the meaning of nouns is a far more arduous task. In fact, numerous psychologists, linguists and philosophers have studied the notion of word meaning without coming to a single definitive definition of this concept and what it means to humans [104]. For his part, Zadeh observed that words such as "green" or phrases such as "large integers" have an imprecise and subjective meaning. As such, he proposed to quantify the meaning of these words as fuzzy subsets of a universe of discourse, and developed a mathematical framework to represent and evaluate the value of these subsets [104]. He later expanded this concept by creating a new fuzzy method for representing the meaning of words in a natural language, which he called *Possibilistic Relational Universal Fuzzy* (PRUF) [105], [106]. According to Zadeh, PRUF is to fuzzy logic what predicate calculus is to propositional logic. That is to say that he developed a method to translate natural-language premises into PRUF expressions. These expressions can then be transformed using fuzzy rules of inference to yield new PRUF expressions. These new PRUF expressions can then be translated back to natural language as conclusions inferred from the original premises. PRUF not only differs from other methods of representing knowledge because it uses fuzzy logic, but also improves on other fuzzy-logic-based approaches by making the basic assumption that the vagueness intrinsic to natural languages is possibilistic in nature as opposed to the probabilistic nature that is assumed by other meaning representation techniques. By taking possibility distributions as its starting point, the PRUF method allows for a uniform treatment of the intended meaning,

probable meaning and possible meaning of propositions and makes it possible to manipulate them all in a manner similar to predicate calculus [106].

Zadeh's next breakthrough followed from his theory of fuzzy information granulation (TFIG) [107]. He started from the idea that information granulation (IG) is one of the fundamental mechanisms of human cognition, along with organization and causation. Presented briefly, granulation is the division of a whole into its parts, while organization is the opposite mechanism of joining a set of parts into a whole, and causation is the understanding of the relationship of cause to effect. Zadeh studied the implementation of crisp IG, which is commonplace in fields such as interval analysis, rough set theory, machine learning from example and cluster analysis. He then extended the concept of IG to fuzzy logic and by so doing laid the foundation for TFIG. A key difference between crisp IG and fuzzy IG is that in the former case granules are precisely defined, while fuzzy IG allows the creation of fuzzy granules, which have fuzzy attributes and values. It is clear that this theory can be easily applied to NLP by defining words as fuzzy granules. Next, Zadeh combined TFIG with his previous work on generalized constraints (GC) [108], in which he defined relationships of the general form "*X isr R*", where *X* is a variable being constrained, *R* is the relation constraining *X*, and *isr* is a copula containing the discrete values of *r*, the variable that defines how *R* constrains *X*. By combining TGIF and GC, Zadeh introduced the concept of precisiated natural language (PNL) [109]. A statement expressed in PNL is a translation of a natural language sentence into a mathematical protoform in which the words are fuzzy granules and the relationships between them are governed by the rules of generalized constraints. Furthermore, it is possible to manipulate PNL statements with a mathematical precision that is absent from natural languages with the help of operators defined in the generalized constraints. These operators allow for the application of inference rules to extract new information from a database of PNL statements and then answer simple queries. In spite of all these promising applications, Zadeh's techniques remain limited in scope since they can only represent words that can be ranked on a measurement scale such as "very", "somewhat", "red", "tall", and simple sentences such as "most Swedes are honest" or "getting there takes about 20-25 minutes".

More recently, researchers have considered the possibility of representing semantic information in the form of fuzzy relations [2][83]. Rather than using Zadeh's approach of representing the meaning or effect of a word directly with fuzzy sets, fuzzy relations define a word's meaning in terms of its relationships with other words and uses fuzzy logic to quantify the degree of those relationships. The resulting database of words and relationships is usually called a *fuzzy thesaurus* because the act of looking up a word in this database will return a list of words with related meaninga. In [2], researchers built a fuzzy thesaurus of soccer-related words using the following three relationships:

1) Equivalence: two words are synonymous;

2) Inclusion: an asymmetric relationship in which one word is a generalisation of a more specific word;

3) Association: two words are loosely related.

In [83], the authors took adapted the concept by focusing on affect-related information such as emotions, feelings, attitudes, temperaments, humours, frames of mind, moods, spirits, morale and dispositions. They reasoned that since affect-related information fundamentally pervades all natural language documents and human thinking, capturing such information will extract important information from texts, which can then be usefully integrated into all types of NLP applications. They went on to define 83 basic affect categories and created a fuzzy thesaurus of affect-related words using these two relationships:

1) Centrality: the extent to which a word belongs to a category;

2) Intensity: the suitability of the word in the category.

Fuzzy thesauri such as those presented above are valid means of representing words within a specific domain, such as the soccer domain of [2], or words related to a specific set of concepts, such as affects in [83]. However, they are of limited usefulness in most general, non-domain-specific applications.

That said, fuzzy-based natural language applications have been successfully implemented in and highly beneficial to several other fields such as database handling. Indeed, it is more

intuitive and productive for users to query the database using natural language questions rather than domain-specific keyword searches. To this end, a method is needed that would parse the sentences, recognize keywords, then reorganize those keywords into the appropriate database queries. One such system has been proposed in [90]. This system relies on a fuzzy grammar to parse possibly incomplete sentences. The fuzzy grammar allows the system to classify the words of the query according to three levels of importance, to detect that words are missing in queries, and to automatically correct grammatically incomplete queries if the missing words belong to one of the two lowest levels of importance. To work properly, this system requires a complete list of grammar rules along with a fuzzy value representing the importance of each rule. These fuzzy rules must be written manually, and often relate to the specific domain in which the query will be asked. In yet another contribution to database applications, a fuzzy data mining and inference engine was proposed [77] to detect entries in natural-language databases that represent the same concept. Such a system requires fuzzy inference rules to perform the comparison and judge the results because highly dissimilar natural-language expressions can describe the same concept in two databases. However in this application, selecting what part of each entry to consider and the construction of the inference rules used to compare them are both done manually by a domain expert. Thus, these two applications are limited by the tedious labour they impose on the system designers.

2.2.3 Relationship to our Work

The methods discussed above represent important milestones in the field of NLP; in particular, that of fuzzy NLP. However, while they offer valuable insights into the NLP problems they set out to solve, these methods also present significant constraints that limit their applicability.

The original bag-of-words approaches, though interesting, are limited by their inherent disregard for all syntactic and semantic information in the text, which prevents any unambiguous understanding of the document. Instead of solving this problem, dictionary-based methods such as the Lesk algorithm attempt to alleviate it by introducing new semantic information in the form of the dictionary definitions of the words that appear in the text.

Although these dictionary methods enjoy greater popularity than bag-of-words approaches, they still disregard too much information. To address this issue, a number of researchers have proposed statistical methods designed to represent semantic information through the use of predefined semantic patterns. On closer inspection, these patterns are only effective for texts that follow a specific structure, which implies an *a priori* in-depth knowledge of the documents being analyzed. Since this condition is rarely met in practice, the scope of these methods narrows considerably. To represent natural languages using more flexible tools, some researchers turned their attention to fuzzy-based approaches. Zadeh pioneered a good deal of this work and created predicate languages such as PRUF and PNL that can represent words and sentences and manipulate them using mathematical operators or even inference rules. Unfortunately, these languages can only handle the simplest of sentences. Other researchers studying fuzzy-based NLP proposed that words can be represented in terms of their relationships to selected key concepts. They constructed fuzzy thesauri in which each word is represented by the fuzzy quantification of its relationship to the key concepts. This line of reasoning appears lead to a very narrow field of applications. The choices of key concepts and their corresponding fuzzy quantifications are inherently domain-dependent, which makes the thesauri limited in usefulness to general applications. Finally, several researchers have designed fuzzy-based NLP systems specialized to certain applications such as database handling. These systems rely, however, on large and often domain-specific, manually-constructed fuzzy rule bases that render their implementation long and tedious.

The purpose of Chapter 5 is to develop a new NLP method to address all the problems mentioned above. Instead of using the semantically barren bag-of-words approach or the strict semantic patterns, our method for semantic information representation will focus on representing the relationship between the subjects, verbs and objects of each sentence. As we will demonstrate in that chapter, the subject-verb-object triplets contain most of the semantic information of the sentence, so representing these triplets imparts a significant level of understanding to the system. Moreover, the subject-verb-object structure is fundamental to English language usage. As such, our method assumes no *a priori* knowledge of the text to be analyzed and remains general and domain-independent, unlike the semantic patterns and

fuzzy thesauri. The subject-verb-object relationships will be represented mathematically using principles founded in possibility theory, which we will present in the next section. Zadeh proposed this theory as a framework to represent and handle the intrinsic uncertainty of natural languages [110]. In his work, he argued that much of the knowledge that humans transfer through language is possibilistic in nature rather than probabilistic or fuzzy, and that possibility theory should therefore be used to represent knowledge. Following Zadeh, we will rely on the mathematical framework of possibility theory to develop the methods used in our study.

2.3 Overview of Possibility Theory

2.3.1 Introduction to Possibility Theory

Possibility theory, first introduced by Zadeh [110], can be seen as a middle ground between probability theory and fuzzy set theory. Although it can be difficult in some cases to draw a clear distinction between these three theories, each one represents a different concept. More precisely, the probability of an event is assigned a value between 0 and 1 depending on the likelihood of that event occurring. A probability of 0 means a certainty that the event will never occur, while a probability of 1 is a certainty that the event will occur. Fuzzy set theory on the other hand, deals with the membership of an event in a set. A membership value of 0 means that the event does not belong at all in the set, while a membership value of 1 means that the event epitomizes of the set. Finally, possibility theory expresses the ease with which an event can occur, or belong to a set. A possibility of 0 means that an event cannot occur, while a possibility of 1 means that the event is completely allowed to occur. Some examples will help to clarify these notions.

To distinguish between possibilities and probabilities, consider the simple scenario of tossing a coin. When the coin is fair and the toss is carried out with no intention of influencing the results, the coin is as likely to land on one side as on the other. Within the framework of probability theory, the probability of the coin coming up heads on any individual toss is 0.5, and the probability of it coming up tails is 0.5. However, under similar

conditions, in the context of possibility theory, we can say that the coin could just as easily land on one side or the other. Hence, the possibility of an outcome of heads is 1 and the possibility of an outcome of tails is also 1.

Possibilities can take on values between 0 and 1 to represent events that are restricted but not impossible. To illustrate this situation, consider the quantity of water Canadians drink in a single day. In this case, the probabilities of the consumption of various quantities of water can be estimated by conducting a survey of the drinking habits of Canadians or based on our degree of belief that Canadians will drink a given quantity of water. Possibilities, on the other hand, will express the degree of ease with which Canadians can drink a given quantity of water. If the recommended quantity is three litres per day and people drink less than that quantity, they will likely feel thirsty and dehydrated, but if they drink too much more they will feel bloated and uncomfortable. Table 2-1 presents the probabilities and possibilities estimated for the water consumption of Canadians in the range from 1 to 8 litres per day.

Table 2-1: The probability and possibility distributions of water consumption of Canadians per day.

Litres of water	1	2	3	4	5	6	7	8
Probability	0.2	0.3	0.4	0.1	0.0	0.0	0.0	0.0
Possibility	0.9	1.0	1.0	0.8	0.5	0.2	0.1	0.0

Table 2-1 shows that the possibility and probability values are not necessarily in sync. For example, while it is quite possible for a person to drink 4 litres of water, the probability of anyone doing so is very small. But overall, there is a heuristic connection between possibilities and probabilities. This connection can be expressed intuitively, with observations such as “if an event is almost impossible, it is bound to be improbable”. Zadeh refers to this as the possibility/probability consistency principle [110].

The relationship between fuzzy sets and possibility distributions can be illustrated using the notion of temperature. To keep the example simple, assume that someone divides a temperature scale into three fuzzy sets, namely “cold”, “warm” and “hot”, as shown in Figure 2-1. Each temperature is associated with a membership value in each set, which in turn

represents how compatible that temperature is with that set. For example, a temperature of 10 degrees has a membership of 0.8 in the “cold” set and 0 in the “warm” and “hot” sets, which means that it is very compatible with our notion of cold but not at all compatible with that of warm and hot. Conversely, if we think in terms of a statement such as “this room is cold”, then the fuzzy set “cold” works as a fuzzy restriction on the temperature of the room, and a temperature of 10 degrees is said to satisfy the constraint to 0.8. Alternatively, we can take this situation to mean that there is a possibility of 0.8 that the room’s temperature is 10 degrees given the fact that the room is cold. In this situation, we can think of the value of 0.8 as representing the degree of ease with which someone would call a room with a temperature of 10 degrees “cold”. From this perspective, possibility distributions can thus be regarded as a new interpretation of fuzzy sets and fuzzy restrictions. While the membership value represents to what extent a temperature of 10 degrees belongs to the fuzzy set “cold”, the possibility value, by contrast, represents the extent to which a temperature of 10 degrees corresponds with our archetypal idea of cold [27].

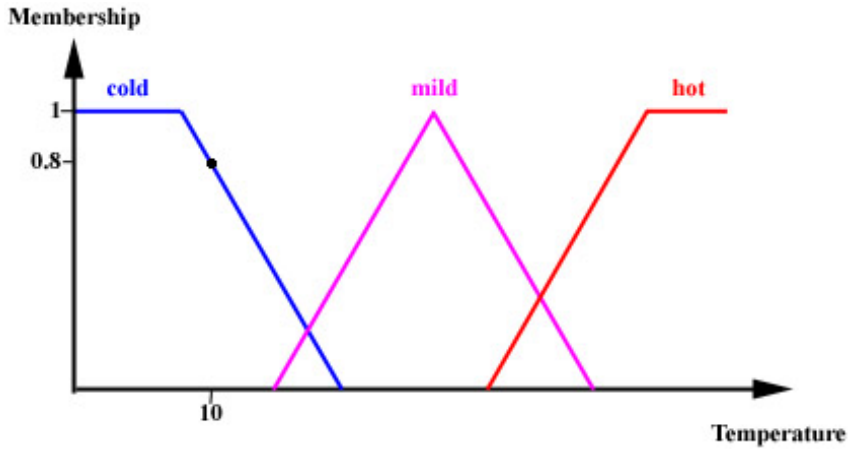


Figure 2-1: Fuzzy sets dividing a temperature scale.

Possibility distributions sometimes refer to the ease with which figurative events might occur. Indeed, the degree of ease represented by the possibility distribution may or may not be physically real. In the first two examples, the possibility values represent the ease with which a coin can land on one of its sides and the ease with which someone can drink a given quantity of water, respectively. In both cases the notions are real and measurable. However, in the third example, the possibility values stand for the ease with which a temperature of 10

degrees can be assigned to the concept of cold. In that case, the notion is figurative and has no corresponding physical significance.

2.3.2 Possibility Theory in NLP

The statistical analysis of information has contributed immensely to the development of modern information representation and handling systems. However, as Zadeh observed in [110], its usefulness is limited to measuring and quantifying information. As such, statistical analysis is not a suitable tool when we are focussed on the meaning of information. In that case, the appropriate framework for information analysis is a possibilistic one, because much of the intrinsic fuzziness of natural languages and much of the knowledge humans gather from linguistic information is possibilistic, not probabilistic, in nature. For example, given a written statement such as "this room is cold" and a specific temperature such as 10 degrees, the reader will get an idea about the room's temperature by pondering the possibility that the room is 10 degrees rather than the probability of the room being 10 degrees. In other words, he will weigh how easy it is for a room to be 10 degrees given that we know it is cold, rather than how likely it is to be 10 degrees given that it is cold or how strongly he believes that a room can be 10 degrees given that it is cold. This important distinction begins to explain the different scopes of probability theory and possibility theory.

Consider the statement "Bob is young". In a probabilistic framework, one would assign different values various ages that correspond to how likely we believe or observe it is for someone of that age to be called "young". This set of probabilities will then become the probability of Bob's age, given that we know he is young; any given age will have a known probability of being the age of Bob. On the other hand, in a possibilistic framework, "young" becomes a fuzzy restriction on Bob's age. Any given age will have a known membership degree to the fuzzy set "young". The statement "Bob is young" will have the effect of converting this membership degree into a possibility degree, specifically the possibility that Bob's age is this figure given the restriction of the fuzzy set "young", or the degree of ease with which Bob's age can take this value given the elasticity of the knowledge that Bob is

“young” [110]. In this sense, we say that probabilities measure the information given by sentences, while possibilities capture the meaning of the sentence.

An important advantage of possibility theory is its unbiased ability to represent ignorance [26]. Consider for example the two scenarios described in [29]. In the first, a gambler has the option to bet on several events, which are all known to be equally probable. A rational gambler would of course bet exactly the same amount of money on each event. In the second scenario, a gambler has to bet on events while being completely ignorant regarding which is more or less likely. In that case, the most prudent strategy is again to bet an equal amount of money on each event. From a probabilistic point of view, these two scenarios are equivalent. Probability theory is thus an inappropriate framework for handling ignorance, and particularly the extreme case of total ignorance. Indeed, by assigning an equal probability of (1 / number of events) to each event in the case of total ignorance, we implicitly assume it is possible to compile an exhaustive list of all events. In a situation of complete ignorance, this assumption is questionable.

In contrast, possibility theory gives us a more flexible and non-biased framework for representing ignorance [26]. This is because consequence of the fact that, unlike probabilities, possibilities are not additive [29] but follow instead axioms referred to by [28] as “maxitivity” and “minitivity”.

Assume two events E1 and E2, each with a different probability and a different possibility of being true, which we will note as P(E1), P(E2), π(E1) and π(E2). Following [110], we can define the union and the intersection of these two events using equations (1) and (2) respectively:

$$\pi(E1 \textbf{ » } E2) = \max(\pi(E1), \pi(E2)) \quad (1)$$

$$\pi(E1 \textbf{ ... } E2) = \min(\pi(E1), \pi(E2)) \quad (2)$$

Following [26], we can also define the necessity of an event as follows:

$$N(E1) = 1 - \pi(\sim E1) \quad (3)$$

Where ~E1 is the complement of E1. Necessity is a measure of possibility that is defined as function of the complement of the event. For an event to have a necessity of 1, its

complement must have a possibility of 0. In other words, if an event must necessarily occur, then its complement must be absolutely impossible. Necessity obeys the axioms of maxitivity and minitivity, and in particular it obeys the following relation:

$$N(E1 \,\mathbf{...}\, \sim E1) = \min(N(E1), N(\sim E1)) = 0 \quad (4)$$

which prohibits an event and its complement from being both in any way necessary at the same time. Likewise, the possibility of an event and its complement obey the following relation:

$$\pi(E1 \,\mathbf{»}\, \sim E1) = \max(\pi(E1), \pi(\sim E1)) = 1 \quad (5)$$

which means that given two contradictory events, at least one of them will be completely possible.

Probability theory offers a similar axiom to express the probability of two complementary events:

$$P(E1 \,\mathbf{»}\, \sim E1) = P(E1) + P(\sim E1) = 1 \quad (6)$$

This can be seen as a counterpart to equations (4) and (5). However, as [29] points out, equations (4) and (5) only imply the weaker relations:

$$N(E1) + N(\sim E1) \leq 1 \quad (7)$$

$$\pi(E1) + \pi(\sim E1) \geq 1 \quad (8)$$

This is one of the key differences between probabilities and possibilities. While the probability of an event completely defines the probability of the complementary event, the possibility and necessity of an event are only weakly tied to those of the complementary event. This gives the possibilistic framework more flexibility in representing the uncertainty of an event. Moreover, to thoroughly express this uncertainty requires knowledge of both its possibility and its necessity. Possibility is a more natural way of handling uncertainty and reflects the human tendency to evaluate an uncertain event in terms of the evidence for it (the necessity) and the evidence against it (one minus the possibility) [29].

Based on the previous analysis, possibility theory is related to the Dempster-Shafer theory of evidence with the measures of possibility and necessity being cautious versions of belief and plausibility respectively. This is a problem for some authors who see probability theory

and the Dempster-Shafer theory as fundamentally incompatible and clashing. For example, in [66], Pearl develops a case where the probability of two mutually-contradicting premises can be used to correctly compute the probability of a consequent. Meanwhile, he shows that computing the level of belief using the Damper-Shaffer theory overestimates the value of the consequent [66], and takes that as evidence of the problematic limits of that theory compared to probability. However, in light of our analysis in this section, this is simply a natural consequence of the difference between the strict equality of equation (6) and the flexible inequality of equation (8). Moreover, this argument omits an important point that Pearl mentions a few pages earlier [66 p.421]: possibilities are designed to handle the very different situation in which we are ignorant about some of the information needed to compute the probabilities of the premises. The Bayesian probabilistic model simply cannot account for this ignorance [26] and requires that we make assumptions about the missing information [66 p.425], which introduces a personal bias to the model. This makes probability theory and possibility theory complementary rather than opposites; two theories that can be used to model different states of knowledge, ignorance and uncertainty.

Since the information in natural language sentences is necessarily in a summarized and therefore imprecise form [110], a certain degree of ignorance is unavoidable in our models. We chose to build our method in Chapter 5 using possibility theory as it provides us with the greatest flexibility in handling this uncertainty.

2.4 Comparison of Statistical and Symbolic NLP

Two popular branches of natural language processing are those of statistical and symbolic NLP. In the first branch, tools and methods are designed based on statistical regularities of language observed in a training corpus. This allows these methods to infer information that will be correct on average. On the other hand, techniques in the branch of symbolic NLP seek to design patterns or templates to apply to the text, and infer information based on how (or if) a given sentence fits each pattern. The creation of these templates usually requires both a training corpus and background knowledge about the topics of the texts the templates will be applied to. The main differences between statistical and symbolic NLP methods can be

effectively analyzed in terms of the width of their coverage and the depth of their understanding [78]. By virtue of the fact that they do not normally require semantic information or reasoning and that they only rely on the observation of word occurrences, statistical NLP methods are the more general of the two and can be applied to a wide array of domains with little modification. However, they are by nature confined to retrieving superficial statistical data such as word counts and co-occurrences, and extrapolating from them. It thus appears that statistical NLP methods have a large width but a limited depth. On the other hand, symbolic NLP methods are more precise on account of their reliance on background knowledge and semantic information. They are thus capable of extracting precise information and relations from within text documents. In spite of their distinct advantage in the specific domain for which they were developed, the reliance of symbolic NLP methods on domain-specific knowledge makes it difficult to port them to other domains. In this sense, it can be said that symbolic NLP methods have a large depth but a limited width. This distinction between width and depth limits, in more cases than not, the practical applications of these methods, as will become evident in the following paragraphs.

A popular application of statistical NLP is the creation of web filters. These tools are developed to detect and block web pages with inappropriate (usually erotic) content while allowing access to appropriate pages (such as sexual health resources). Given that acceptable and unacceptable pages often use the same vocabulary, filtering tools that rely on simple keyword detection can easily be misled. This observation underscores the need for more sophisticated statistical tools to fill this gap. To address this issue, the authors in [97], for example, suggested refining keyword detection systems by sorting keywords into three classes that represent three different frequencies of words occurrences within acceptable and unacceptable pages. The importance of an observed keyword is then expressed numerically both in relation to its class and to the presence of related words of other classes. Other researchers [25] followed a different route and explored the possibility of comparing a web page to a corpus of unacceptable pages. By representing each page as a vector containing the frequencies of occurrence of common words, it becomes possible to compute the similarity between the pages as the distance between their vectors using such common measures as the

cosine method. The filtering task is then accomplished simply by denying access to web pages that are too similar to one of those in the corpus of unacceptable pages.

Tools that make use of symbolic NLP methods are favoured when there is a need to retrieve some very specific information from a corpus of specialized documents. Two typical examples of such tools can be found in the biomedical domain and are described in [37] and [62]. The first example is a system designed to extract information on protein-protein interactions from scientific papers [37]. This system builds chains of words that are strongly semantically related in each document, isolates the chains that are relevant to the topic of biology, and finally ranks the selected chains according to the strength of the relationships between the words they contain. To recognize relevant chains and evaluate the strength of word relationships in a biological context, this system relies on several specialized linguistic resources including biological and medical ontologies as well as an extension of WordNet for protein-related terms. The second example is an indexing tool designed not only to list the references found in a biomedical paper, but also to provide information about the relationship of each reference to the contents of the paper in question [62]. This system must therefore detect and understand linguistic cues at several levels of textual granularity. At a coarse level of granularity, the specific positioning of a reference in the document carries noteworthy significance that could be exploited. Indeed, references cited early in the text would likely represent links to previous work in the field, whereas references mentioned further on are likely related to the experimental results of the study. At a finer level of granularity, the words surrounding a reference usually convey more specific information about its purpose, as for instance if the reference confirms, explains, or rather contradicts the particular argument to which it refers. In this regard, the aforementioned linguistic cues are obtained from a corpus of scientific documents through a systematic learning process. On this point, the authors remark that to extract more sophisticated cues, the corpus should be limited to papers related to the same field.

The various examples presented above illustrate the advantages and shortcomings of statistical and symbolic NLP methods. The main advantage of tools engineered on the basis of statistical NLP is that they can be applied to documents from a wide array of domains with

little or no modification. Indeed, were it not for issues of scalability, the tools presented in [97] and [25] could conceivably be successfully applied to the entire Internet at once. However, statistical tools are fundamentally confined to conduct a very coarse level of information analysis, which restrains their depth of semantic usefulness. To highlight this fact, recall that the tools discussed in these examples can only perform a binary classification of websites based on their overall content. On the other hand, tools designed using symbolic NLP methods provide more complex results. In particular, they have the distinct advantage of detecting and extracting very fine-grained information from documents such as the nature of protein-protein interactions [37] or the relationship between a paper and a reference it contains [62]. However, while the two symbolic tools mentioned above have a definitive lead over statistical tools in the biomedical domain, their reliance on domain-specific information restricts the width of their coverage. Their applicability to other domains hinges upon the introduction of major modifications in their knowledge base and a renewed training in that updated context.

The examples presented above illustrate the advantages and shortcomings of statistical and symbolic NLP methods. Most notably, they reveal that statistical NLP approaches have the advantage of being general enough to suit a wide array of domains with little modification. However, they are inherently limited to the extraction of superficial data. By contrast, symbolic NLP tools can perform a fine-grain analysis of a text, but they cannot easily be exported to other documents of a different nature. In short, statistical NLP methods demonstrate large width but limited depth, whereas symbolic methods exhibit large depth but limited width.

The method we propose in Chapter 4 seeks to integrate the width of coverage and depth of understanding of statistical and symbolic NLP tools. As we will show in that chapter, our method learns rules that are precise enough to extract specific information from individual sentences, which matches the depth of other symbolic NLP methods. At the same time, these rules are based only on sentence structure with no reliance on the domain-specific information that limits the width of symbolic NLP methods. Furthermore, since recent studies indicate that information regarding syntactic structure can be transported from one

domain to another [19], we can say that our method has a width comparable to that of statistical NLP approaches.

Our method of Chapter 4 bears some similarity with work done in the field of Natural Language Understanding, specifically the semantic entailment system described in [14]. Much like our method, the system described in [14] uses a knowledge representation hierarchy. However, their hierarchy is very different from ours. While the part-of-speech hierarchy we present in this chapter is designed to only handle individual words, the hierarchy in [14] handles both words and phrases. Moreover, while our hierarchy is for parts-of-speech only, the hierarchy in [14] includes a variety of information, such as semantic role and syntactic parse tags of words, the phrase types of arguments, and the structure of complex sentences. This abundance of information makes the hierarchy of [14] a lot more complex to handle than the one we have designed.

2.5 Overview of Grammatical Function Labelling and Semantic Tagging

Semantic tagging classifies words according to their meaning or role in the phrase. This problem has been extensively studied, and several interesting approaches have been proposed. For example, the FrameNet Project [6] has constructed a database that indexes the descriptions of lexical items. These descriptions include the context in which each item is used as well as the semantic role of co-occurring lexical items. This database is a valuable tool for several NLP applications like word sense disambiguation [31], text understanding [30], and even machine translation [11]. However, the construction of such a database is a time-consuming endeavour that requires considerable effort on the part of human annotators and reviewers [6]. Another project along the same vein is that of PropBank [46], which is a descendent of the Penn TreeBank [57], [58]. The objective of PropBank is to add a layer of semantic annotation to the Penn TreeBank to extract relational data from the text. The semantic information added, which is in the form of verb-argument relations, is similar to the information we seek to extract from our documents. However, PropBank, like FrameNet, relies on manual annotation of the data.

Numerous other small-scale semantic annotation projects also exist in the research framework of the Semantic Web [9]. But, given that the manual annotation of the Internet is an unthinkable task, these projects invariably focus on the automation of the annotation process. To accomplish this automation, the projects then rely on various pre-existing ontologies, such as the ontology of concepts used by [64], the ontology of annotated poems proposed in [82], the sharable domain ontology encouraged by [81], or the bridge-ontology designed to link multi-ontologies and presented in [89].

Recently, several ingenious algorithms have been proposed to address the challenges of automated grammatical function and semantic role tagging. These algorithms seek to discover information about the role of nouns or noun phrases in a sentence. These roles are defined in relation to the predicate of the sentence; while the predicate is the action, a noun can represent the manner, benefactor, location or direction of this action, to name only a few examples from [10]. This information in turn allows a much deeper level of text understanding, as we show in Chapter 5.

Although the algorithms proposed in the literature are rich in variety and originality, they tend to follow the same underlying principle. Their basic design is to prepare a list of semantic tags or grammatical tags, as well as a corpus of correctly-tagged documents. The algorithm then uses the documents to discover linguistic features that are observed in the texts when each tag is encountered, and based on these observations it computes the probability of each tag given the observed features. Finally, the implemented tagger works by detecting the keywords to tag as well as the known features surrounding it in the document to be analysed, and assigns to each keyword its most probable tag.

This methodological structure is obvious in the semantic role tagger proposed in [34]. In this paper the authors define a set of 12 semantic domains for their tagger to recognise, and rely for their training corpus on a subset of the British National Corpus that was hand-annotated by the FrameNet project. During its training phase, the system computes the probability of each semantic domain given various features, such as the surrounding words and the parse tree of the sentence. In empirical tests, these features allow the system to

correctly tag the semantic domain of keywords with an 82% accuracy level. A similar algorithm based on the work done in the Penn Treebank project rather than in FrameNet was also proposed for grammatical function tagging in [10]. The grammatical tag set used in this algorithm is the one defined in the Penn Treebank, some examples of which have been given previously, and the training and testing data is a section of the data used in the Penn Treebank project. As in [34], the features detected in [10] are Boolean-valued functions of the parse tree of the sentences, which combine both lexical and part-of-speech information, and which allow it to achieve an 87% accuracy level. As the authors note in [10], this mix of lexical and part-of-speech features is the source of one of the weaknesses of the method. Indeed, the grammatical tags used in that project are quite fine-grained, but the parts-of-speech are too coarse to accurately predict the correct tags to use, while the lexical information is too sparse to yield precise statistics. This latter problem, however, could be solved by grouping the sparse lexical items into larger and more general categories. That, in fact, is the intuition behind another grammatical tagger, presented in [23]. This tagger, however, has a more specialized purpose. It seeks to disambiguate the sense of nouns used as arguments of predicates. For example, given the sentence "Fred ate strawberries with a spoon", it would detect that "with a spoon" is an argument of "ate" rather than "strawberries". Nevertheless, the tagger follows the same basic methodological structure as before. It defines a set of tags, which represent argument positions between a word and a predicate, and selects a training corpus, namely the British National Corpus. The features it detects are simply the occurrences of noun-predicate-argument triplets. As mentioned earlier, this type of lexical data is very sparse, a problem that the authors of [23] solve by moving the nouns up to more general categories in the WordNet lexical hierarchy. They then focus mainly on estimating the optimal level of generalization of nouns in WordNet, in order to compute probabilities that are based on data that is neither too sparse nor too general. At that optimal level, their tagger achieves a 74% accuracy level.

The variety in semantic role and grammatical function tagging stems in part from the particular tag set defined, as demonstrated by the examples given previously. In these examples, tag sets encompass 20 specific grammatical functions [10] or 12 general semantic

domains [34] or two basic argument positions [23]. The analysis conducted in Section 2.5 show that pre-existing resources such as FrameNet and PropBank are a mixed blessings. On one hand, they provide researchers with a versatile framework and an annotated corpus, two elements crucial to designing a new algorithm. On the other hand, restrictions resulting from design decisions of the original resource are automatically inherited by the new algorithm. For example, by virtue of being built based on the Penn Treebank project, the tagger of [10] gained a large and correctly-annotated training corpus but was limited to using the Penn Treebank tag set. Algorithms built using specialized ontologies such as those of [81], [82], [89], inherit even more restrictions, as they can only ever be applied in the domain for which the ontology is valid. To avoid inheriting such restrictions, some authors [23] simply avoid these resources.

With this in mind, Chapter 3 presents the first of two original taggers developed for this research. Our approach in that chapter most resembles that of [23]. As in [23], we limit our scope to the basic subject and object argument positions. As we will show in that chapter, our method reveals these arguments by using rules that combine both part-of-speech information and lexical information and in one case a more general lexical category. We also make the same decision as [23], to forego the use of pre-existing semantic resources or ontologies.

2.6 Overview of Text Classification

Text classification refers to the task of assigning new, previously-unseen documents to the most similar of several predefined topics based on the documents' contents. This task has been studied extensively over the past few years as reported in [111], [38], [3], [94], [44]; several approaches have been developed to tackle this complex issue.

Most of the techniques proposed for text classification in the literature rely on a document vector to represent the texts be classified. Such a vector may simply contain the frequency of certain important keywords, or if it is designed in a more sophisticated way, it may quantify relevant text features [1]. The document vector is then compared with vectors representing archetypal documents of each domain. Several methods designed to perform these comparisons have been proposed and analysed. A popular method is *k*-nearest-neighbours

(kNN) [111], in which the entire vector is compared to all vectors in the training corpus, and the *k* most similar vectors are retained. The domains represented by those *k* vectors are then voted upon with weights corresponding inversely to their distance, and the domain with the highest score is assumed to be the correct one. The main drawbacks of this approach arise from the fact that the comparison of the vector to the entire training corpus imposes inherently stringent requirements in terms of memory and computation. An improvement on the kNN algorithm, called the ε-approximate nearest neighbour (ε-ANN), has recently been proposed to address this problem [18]. But while the ε-ANN algorithm is efficient in one- and two-dimensional cases, it still has serious obstacles to overcome before it can be efficiently applied to multidimensional problems such as those faced in NLP [18]. Other comparison techniques have been proposed to circumvent this problem. For example, using a decision tree as proposed in [38] and [3] allows the classifier to compare key elements of the vector one by one in a specified order and to terminate the search once it encounters a combination of features that only occurs in a particular domain. This technique is subject to its own drawbacks, namely the requirement to learn the tree from a training corpus while avoiding the problem of overfitting. This problem occurs when a decision tree has been built using overly-specific, irrelevant or coincidental features of the domains found in the training corpus. In these cases it generates a very accurate classification on the training data, but has a very poor performance on unseen test data.

Given the non-crisp nature of human knowledge and of the knowledge conveyed in natural language sentences [103], it seems intuitive to use fuzzy logic tools to deal with text classification. Some researchers have explored this possibility, and in so doing they have come up with some interesting classifiers. One such system worth noting is based on a fuzzy similarity metric [94]. At its core, this system is an implementation of the Rocchio algorithm, which clusters the vectors of each domain in the training corpus and replaces them by a single vector representing the center of each cluster. In the classic Rocchio algorithm, the correct domain of a test vector is then computed as the nearest cluster center using a Euclidian metric such as the cosine coefficient [71]. In [94], the authors improved on this algorithm by replacing the crisp values of the cluster center vectors with fuzzy sets to

represent the membership of each term in the domain. They also computed the similarity between a test vector and the cluster centers using ordinary fuzzy operators [94]. Their results show that using fuzzy similarity gives a better classification than using crisp measures. Some researchers undertook the more general task of comparing the performance of classification using crisp and fuzzy clustering methods [44]. They found a notable improvement in the quality of the results when fuzzy clustering is used.

2.7 Conclusions

This chapter has presented a survey of a representative sample of modern techniques that we developed either for the general field of NLP, or more particularly for the specialized branches of semantic tagging and domain classification. Although these techniques work well under the specific conditions for which they were designed, our analysis shows that their applicability is limited, which makes them inappropriate for the purpose of this study.

The purpose of the research that will unfold in the following chapters is to develop new NLP methods that which will address the problems mentioned throughout in this chapter. The approaches we develop in this study build upon our previous work in this field as documented in the publications listed in Appendix C. One of the key innovations in our method for semantic information representation is its focus on modelling the relationship between the subjects, verbs and objects of each sentence. As we will illustrate, the subject-verb-object structure contains most of the semantic information of the sentence, so that modelling it imparts a useful level of understanding into a text-analysis system.

Extracting the subject-verb-object triplets from the documents could easily become the limiting factor of our method. To address this problem, we propose in the following chapters another method, designed to learn autonomously the rules needed to identify the subjects, verbs and objects in sentences. With this second method, implementing systems based on our research does not impose a heavy workload on the system designer in the form of rule base programming, as other applications often do.

The remainder of this thesis presents key characteristics of the proposed methods that highlight their differences with traditional approaches found in the literature. Accordingly, the first key concept consists in the extraction of the subject-verb-object triplets from the documents under analysis. A first approach to accomplish this task will be presented in Chapter 3, while a second, more robust method will be put forward in Chapter 4. The semantic information contained in the triplets will then be represented using a formal mathematical expression in Chapter 5. Experimental results obtained using an implementation of the representation method of Chapter 5 will be presented in Chapter 6.

Chapter 3
Syntactic Heuristics

3.1 Introduction

It is a common scientific practice to make simplifying assumptions in order to get a better understanding of the important relationships on which to concentrate. In this research, we assume that most of the semantic information in a sentence lies in the action described by that sentence. For example, the meaning of the sentence "the excited man drives the red car" is best captured by "man drives car", rather than "excited man" or "red car". Consequently, we focus on identifying and modelling relationships between words where those relationships represent actions. In more precise terms, our focus in this chapter will be on identifying and modelling the relationships between the subject, verb and object of the sentences.

To minimize the workload for the user, our complete system is designed to accept un-annotated English documents as training or testing data, and then automatically extract the subject-verb-object triplets from them. In Chapter 5, we propose to use a possibilistic method to represent the semantic information these triplets contain. Chapter 5 illustrates how this method computes the possibility of the different domains given the subject-verb pairs and the verb-object pairs. The possibility of a domain given a triplet will then be computed on the basis of its subject-verb and verb-object pairs. This will tell the system to which domain the triplet belongs to, and the degree to which that domain classification is reliable. The possibility values of various domains given the noun-verb pairs will be computed using information extracted from domain-specific training corpora. This will allow our method to be specialized for a specific domain or generalized to various domains through the use of a general corpus such as the Brown corpus [32].

This chapter presents the first method to extract subject-verb-object triplets mentioned above, and establishes the discussion of the possibilistic method to be presented in Chapter 5.

To accurately find the verbs and their subjects and objects, the system must annotate the document text. To this end, several annotation systems, or taggers, have been proposed in the literature. Some of these taggers have been examined previously in Section 2.5. The rest of Chapter 3 presents the tagger we developed for our study.

3.2 The Triplet Extraction Method

In view of the limits of existing semantic annotation systems and the specific needs of our research, we have developed an appropriate process specifically designed to identify the subjects, verbs and objects in each sentence. It is worth noting that the most accurate results can be obtained only if the triplet extraction is done manually. However, an automated system can provide fairly accurate results nonetheless. In the system we developed, the triplets are extracted from the text automatically, using a three-step process. This process begins by performing a part-of-speech tagging to find the verbs and nouns in the sentence. It then applies some semantic rules to recognise the useful verbs, their subjects and their objects. Finally, it removes some information we deemed irrelevant from the triplets, so that later parts of the method may focus only on the most meaningful data.

3.2.1 Part-of-Speech Tagging

The first step of our triplet extraction process is to perform the part-of-speech tagging. This step is intended to transform an English sentence into a simple sequence of part-of-speech tags, which are a lot easier for a computer to handle. Indeed, while the English language features millions of words, it has only a limited number of parts-of-speech, which are typically grouped in only eight main classes: noun, verb, adjective, adverb, pronoun, preposition, conjunction, and interjection [63]. For our research, we are only interested in keeping the verbs and the nouns. Thus, performing part-of-speech tagging allows the system to focus on the relevant parts of the sentence and to discard unnecessary information.

To perform the tagging, we rely on an implementation of the Brill tagger [15]. The Brill tagger is a simple yet efficient English-language tagger that operates by assigning each word its most commonly used part-of-speech tag, and then uses 9 transformation rules to correct

common tagging mistakes. Brill reports that this system achieves a tagging accuracy of 95.6% [16].

3.2.2 Syntactic Heuristics

Once the part-of-speech tags are known, we can extract the triplets. We assume that the sentence follows a "subject verb object" structure, which is a common sentence structure in English. Thus, for each verb in the sentence, we assume the subject is the first noun preceding it and the objects are all the nouns following it. However, in the realm of English common parlance, this assumption can run into some problematic situations. These situations occur when extra words, such as extra verbs, extra nouns or extra sub-sentences, are inserted into the main sentence in order to give more information about the main action depicted. Consider for example the sentence "The seller agreed to meet the client". Obviously, the triplet extracted from this sentence should be "seller meet client" while the extra verb "agree" should be ignored, since the main point of the sentence relates to a meeting between the seller and client. Likewise, in the sentence "Education Minister Kennedy attended a meeting", the noun "education" and the proper noun "Kennedy" should be ignored, and the triplet extracted should be limited to "minister attended meeting", to reflect the fact that the most important attribute of the attendee is that he is a minister, as opposed to being in charge of education or having the last name Kennedy. To address such common problematic situations, nine simple syntactic heuristics are implemented into our system. Their role is to detect these situations, isolate and eliminate the extra words and allow the system to successfully extract the real actions. It should be noted that these nine heuristics by no means cover all possible problematic situations. Yet taken together, they can correctly handle a wide range of common cases, and thus help minimize the number of incorrect triplets that will be introduced in the training and testing data. It is worth noting that, while we chose to use heuristics in order to maintain direct control over the processing of the sentences and the creation of the triplets, a system designer could just as easily replace them with a simple parser that performs the same task automatically. It is also important to point out that, while it remains true that we only wish to keep the nouns and verbs in the end, more information,

such as punctuation and prepositions, is needed for our heuristics to work. The syntactic heuristics implemented in our system are:

- The "*verb* to *verb*" rule: This rule handles combinations of two verbs, such as "we *decided to visit* France". When the system encounters this structure in a sentence, it considers that only the second verb is meaningful and ignores the first one. This is the rule used in the first example given previously.
- The "*verb verb*" rule: This rule handles combinations of two verbs, such as "the story *is based* on historical facts". Like in the previous case, when the system encounters this structure in a sentence, it considers that only the second verb is meaningful and ignores the first one.
- The "*word-verb*" rule: This rule handles hyphenated words that are composed of one verb, such as "this site offers *web-related* advice". The words are separated by the Brill tagger in the previous step of the method, hence the second part is wrongfully detected as one of the verbs of the sentence. This rule allows the system to skip such verbs.
- The "of *nation*" rule: This rule handles names that contain "of nation", such as "the government *of Canada* passed a law". When such a structure is encountered, the system considers that the "of nation" part is unimportant and ignores it.
- The "*noun noun*" rule: This rule handles double-nouns, such as "the *company president* announced the sale". Normally, the second noun is the most meaningful one of the pair, and the first one serves to add details. Consequently, when such a structure is encountered in a sentence, the system ignores the first noun.
- The "*noun-noun*" rule: This rule handles double-nouns that are joined by a hyphen, such as "the famous *actor-director* visited the set". These words were separated by the Brill tagger in the previous step of the method, and are wrongfully detected as two distinct nouns. This rule allows the system to merge them back into a single noun.
- The "*noun propernoun*" rule: This rule handles proper names preceded by a title, such as "Prime *Minister Harper* attended a meeting". In this case, the system considers that

the title is the most meaningful word of the pair, and ignores the proper noun. This is the rule used in the second example given previously.

- The "comma *noun* comma" rule: This rule handles portions of sentences placed between two comas, such as "the president*, who is at a summit this month,* answered the question". The segment between the commas gives additional information on the subject (president). However, since the search for a subject is done by the system by moving leftward from the verb (answered), the first noun encountered is necessarily found in the segment between commas (month), which is evidently not the subject sought after. Consequently, when the system searches for the subject of a verb, it skips portions of sentences between two commas.
- The "in *noun* to *verb*" rule: This rule handles sentence structures such as "students have to read the book *in order to write* a report". When such a structure is encountered, the system considers that the noun is unimportant and ignores it.

3.2.3 Reducing Complexity

Given the rich English lexicon and the use of synonyms, a system that retains in the triplets all nouns and verbs exactly as they appear in the sentences will face serious complexity issues. To begin with, such a system would make irrelevant distinctions between different forms of the same word, such as singular and plural endings, or American and British spellings. Moreover, it has been shown that a typical natural-language document or corpus will be made up of a few frequently-occurring words and many rarely-used words [113], [55]. The frequently-occurring words will be general, and have such a broad meaning that they will be downright worthless in an NLP application. On the other hand, words that have a specialized or specific meaning and would therefore be of great importance in an NLP application are too infrequently used to be of any help. After all, what kind of reliable information could be derived from a word observed only once or twice in a corpus? This situation is a major obstacle in NLP.

In order to address the issues described above, the triplet extraction process in our system includes a complexity reduction phase. This segment of the process is based on the assumption that the most informative part of an action is the act itself, rather than the objects performing the act or those being acted upon, and rather than the time (past, present or future) in which the act takes place. For this reason, we eliminate the verb tense by substituting all verbs in the triplets with their infinitive form. In the same vein, nouns with a similar meaning to each other are replaced with a more general noun category. This is done to avoid the problem of handling many significant but rarely-used words. Instead of having, for example, one hundred nouns with a similar meaning used once each, the system will have one noun category with a roughly equivalent meaning used a hundred times. However, because of the importance we place on verbs, we will not group them in categories like we do for the nouns.

The noun categories have to be designed and specialized for the particular domain in which the tagger will be used. To this end, domain-specific nouns must be represented by specific, fine-grained categories, while general nouns or nouns from other domains can be grouped together in general, broad categories. The justification for this design philosophy is that domain-specific nouns will carry vital details that could be lost if the clustering is too broad, while general and out-of-domain nouns are probably incidental, and unlikely to carry important domain-specific information. Moreover, domain-specific nouns can be expected to abound in their respective domains, which enables us to use fine categories without the risk of encountering data sparseness, while out-of-domain nouns are comparatively quite rare. To illustrate these distinctions, consider a sample of specialized medical terms. In the medical domain, the differences between viruses and bacteria, or enzymes and proteins, are very meaningful. For the purpose of this domain, each of these words should correctly be allocated to its own category, as each one carries a different medical significance. For the business domain however, the differences in the specific significance of these terms are irrelevant, and they can all be grouped under a single “medical terms” category with no loss of relevant information. Moreover, distinguishing between these terms could potentially

introduce distinctions between the documents that are irrelevant for the business domain, and which only serve to confuse the system with unnecessary details.

To conduct this study, we have developed a partially-automated process for the specific purpose of creating the noun categories. The first step in this process is to count n_{it}, the number of occurrences of each noun n_i in each domain d_t of the training corpus. Then, we compute the information gain Δ_i of each noun using an equation we have formulated for this research:

$$\Delta_i = \frac{\max_t(n_{it}) - \min_t(n_{it})}{\sum_{t=1}^{T} n_{it}}. \qquad (3\text{-}1)$$

In this equation, words that frequently occur in all domains will have a low information gain value, while words that occur a lot more in one domain than in any other will have a high value. The second step in our process is to manually sort out each noun into a broad noun category. Although no formal framework exists to help us perform this task, the general guideline we follow is to have as few categories as possible, while separating words with important differences in meaning into different categories. For example, in the implementation we will present in Chapter 6, the 3773 nouns encountered were sorted into 375 noun categories, ranging from specific "worker" and "building" categories to general "individual" and "location". This manual sorting is the reason why we labelled our method as "partially-automated". However, this manual process could be replaced by an automated tool, such as in this case the CBC algorithm [65]. The nouns are then clustered according to the domain in which they occur most often. In the third step, the noun categories are split into several sub-categories, one per domain for the nouns that occur only in that domain, and one for the nouns that occur in several different domains. Then, by applying Equation (3-1), the information gain Δ'_i of each noun sub-category is computed by using N_{it}, the number of occurrences of each sub-category in each domain, instead of n_{it}. Finally, we set a threshold value for the information gain. If the information gain Δ'_i of a sub-category is below that threshold, the noun with the lowest information gain Δ_i is removed from that sub-category and filtered out of the corpus, and the new value of Δ'_i is computed. This process is repeated

until all sub-categories have an information gain value higher than the threshold. Obviously, this last step of the method is skipped for the sub-categories containing nouns that occur only in one domain, since the information gain value of these sub-categories and of all nouns comprised in them is always 1.

It is worth noting that other techniques could have been used, based on more standard information measures rather than the one we chose to devise for our work. One such alternative, for example, could be the pointwise mutual information (PMI) score described in [87]. We adapted the central equation of PMI to our work in equation (3-2). In that equation, n_{it} and n_{jt} are the number of occurrences of words n_i and n_j, respectively, in domain d_t, while the expression $n_{it} \wedge n_{jt}$ represents the number of occurrences of both words together. We can thus see that equation (3-2) computes the information between a pair of words n_t and n_t. Consequently, using this equation will require us to modify the rest of the category generation process. This could be easily accomplished. For example, while previously the last step of the process eliminated the words with the lowest information gain as computed using equation 3-1, our adapted process would eliminate the noun with the lowest average PMI with all other nouns in the category.

$$\Delta_{ij} = \log_2 \frac{n_{it} \wedge n_{jt}}{n_{it} * n_{jt}}. \tag{3-2}$$

As mentioned above, our system considers the verbs to be the most significant part of the triplet. For this reason, we choose not to make use of verb classes. However, not all verbs are significant. Some very common verbs, such as *to be* and *to have*, carry little significance, yet they will have a major impact on the results by virtue of the fact that they are used very frequently in all domains. It is thus necessary to detect and filter out these verbs from the triplets. Consequently, we similarly apply the notion of information gain to the verbs, in order to detect and filter out these common and meaningless verbs. The information gain of each verb is computed by introducing the number of occurrences of the verb in each domain into Equation (3-1). We then set a threshold value, which doesn't have to be the same as for the noun sub-categories, and verbs whose information gain value is below that threshold are considered too common and are consequently filtered out.

3.2.4 Illustrative Example

In order to illustrate the steps of our triplet extraction process, a simple example is presented in Table 3-1.

The sentence to be analysed is presented in the first row of the table. The first step of the algorithm is to generate the equivalent sequence of part-of-speech tags using the Brill tagger, as mentioned in Section 3.2.1. The output of this step is presented in the second row. The implementation of the tagger we took for our research uses the 46 parts-of-speech of the Penn Treebank, which are discussed in detail in [74] and reproduced in Appendix A for convenience. It should be noted that, although "have" and "doubled" are both verbs, they serve different functions in the sentence, and consequently the Brill tagger assigns them different tags.

The next step of the algorithm is to extract the triplets. By using the part-of-speech tags, it's easy to pick out the verbs and the nouns. The verbs are identified first, as the words with a part-of-speech that includes the letters VB. In our example, only "doubled" is marked; "have" has the part-of-speech RB, which labels it as an adverb. Then, since we assume that the sentence follows the "subject-verb-object" structure, the first noun left of the verb is assumed to be its subject, and the nouns right of the verb are assumed to be objects. This leads to the tagging shown in the third row of Table 3-1. However, as we explained in Section 3.2.2, this assumption can lead to the extraction of erroneous triplets in the cases when sentences do not strictly adhere to the "subject-verb-object" structure. To address this problem, we apply the heuristics of Section 3.2.2 to the sentence to eliminate wrong verbs, subjects or objects. This step is accomplished by finding which heuristic, if any, has the same structure as the sentence, and performing the action dictated by that heuristic. In our current example, since the sentence is fairly straightforward, the system finds no suitable heuristic, and the subject, verb and objects extracted remain those of the third row.

Finally, the last step of the process is the one of Section 3.2.3, in which irrelevant information is discarded. As we can see in the last row of Table 3.1, the verb "doubled" has been transformed to its infinitive form "double", and the nouns have been replaced by

general categories. Hence, the sentence in this example would generate two triplets: “finance double measure” and “finance double date”.

Table 3-1: Tagging example.

The	shares	have	almost	doubled	in	value	since	the	start	of	2005	.
DT	NNS	VRB	RB	VBN	IN	NN	IN	DT	NN	IN	CD	.
	subject			verb		object			object			
	finance			double		measure			date			

3.3 Experimental Results

The data used to test our extraction heuristics comes from the Brown Corpus [32]. This dataset is a corpus of texts written in American English and compiled in 1961. It is composed of 500 sample documents selected to reflect the spread of domains that the American public were likely exposed to at that time. The documents in the corpus thus cover a wide range of topics, from news coverage to religious texts, from industrial reports to detective fiction. We decided to set off the demonstration by initially limiting the scope of the rule-learning system to the business domain of the corpus (samples A26-A28) and the science-fiction domain (samples M01-M06).

As a benchmark for comparison, we analysed the selected samples and manually identified the correct keywords to extract. We then applied the extraction method of Section 3.2 on the samples. On the business domain, it obtains a precision of 86% and a recall of 91%, while on the science-fiction domain it obtains a precision of 90% and a recall of 91%. These good results are not surprising; indeed, we designed the heuristics of Section 3.2.2 by studying these same samples and identifying the most general form of the necessary rules. Moreover, they were obtained at the cost of imposing a number of important limitations on the method, which we will now present.

3.4 Limitations of the Extraction Method

The triplet extraction technique presented in this chapter does by no means guarantee the extraction of all the triplets, even less so the extraction of only accurate triplets from every

sentence it deals with. As mentioned before in section 3.2, the best extraction results can only be achieved if an individual were to perform the task manually; and even then, the outcome would not be perfect. Indeed, some ambiguous sentences can be interpreted differently by different people, who will depict them through different triplets. For example, given the sentence “time flies like an arrow”, some readers would annotate the triplet “time fly arrow”, while others would choose the triplet “flies like arrow”, or even a different triplet because of the many different interpretations of this expression that could be imagined.

Ambiguous and complex sentences are the major problem faced by our extraction system. Indeed, the system operates on the core assumption that all sentences follow a straightforward subject-verb-object structure. This is obviously an oversimplification of reality, and it ignores many real sentence structures, such as the post-modification of subjects for example. Consequently, sentences with more sophisticated structures lead to the extraction of erroneous triplets, or cause the system to miss good triplets. We acknowledged this failing and attempted to limit its impact by using some syntactic heuristics to counter common problematic structures. However, there is no limit to the number of different sentence structures that can be used in natural texts, and as a result, it is impossible to devise heuristic rules to handle all possible scenarios. Sentences with unusual structures will always be a source of errors in this extraction process.

Another source of error comes from the system’s reliance on the Brill tagger. This tagger boasts an accuracy rate of 95.6% [16], in the sense that any single word in the sentence has a probability of 95.6% of having been tagged correctly. This means that the longer the sentence, the more likely a tagging error will occur somewhere in it. For example, the sentence presented in Table 3-1 has 13 taggable elements, hence it has only a 55.7% chance of being entirely tagged correctly. At the moment, only tagging errors that cause nouns and verbs to be mislabelled, or that cause other words to be mislabelled as nouns or verbs, can have a detrimental effect on the system. However, as new and more elaborate syntactic heuristics are added to the system for the purpose of handling more complicated sentences, we can expect that the negative impact of these tagging errors will become more important for the system’s performance.

The use of anaphora in sentences brings about a loss of information for our system. Anaphoric words are placeholders that can stand for other words, expressions, or even complete noun phrases. However, they are meaningless in and of themselves, and can only make sense when we know what they stand for. The task of disambiguating their meaning is called *anaphora resolution*, and it remains to this day a major challenge in the field of NLP [42]. Our system is not designed to address this problem. As a result, encountering anaphora as subject or object in a sentence leads to the extraction of meaningless, useless triplets.

Finally, the method does require a non-negligible amount of manual work. Indeed, we designed the syntactic heuristics of Section 3.2.2 manually after studying the test corpora that will be used in Chapter 6, and the process to generate the noun categories in Section 3.2.3 requires the manual sorting of the nouns into coarse categories that the process then refines. This limits the portability of the method.

The shortcomings of this method mentioned above are serious enough to justify the search for a more robust method for the extraction of the triplets. This is the subject matter of the following chapter.

3.5 Conclusions

In this chapter, we focused on the triplet extraction method of our research. We provided an analysis of the triplet extraction process we developed for this study. We showed that our process follows three sequential steps. It begins by identifying the part-of-speech tags of the sentence. It then isolates the relevant nouns and verbs using simple semantic rules, and associates each verb with its subject and objects. Finally, it removes superfluous information by replacing nouns with general noun categories, eliminating verb tense, and filtering out common verbs. An example illustrated the functioning of this process and exposed the theoretical limits of the method.

Although simple and efficient, the method's demonstrated shortcomings severely limit its applicability and justify striving toward a more robust alternative. Chapter 4 will present our alternative approach for triplet extraction.

Chapter 4
Part-of-Speech Hierarchy

4.1 Introduction

Earlier research [78] has demonstrated that tools based on statistical natural language processing (NLP) approaches are general enough to generate a coarse-level understanding of almost any text. However, statistical NLP tools are rather limited when it comes to extracting finer details. Conversely, symbolic NLP tools can perform a fine-grain analysis of a text, but they cannot easily be exported to other documents of a different nature. This study expands the current NLP literature by developing a new method that combines the advantages of both statistical and symbolic NLP. This method permits a fine-grain analysis comparable to that of symbolic NLP applications while maintaining the generality that characterises statistical NLP algorithms.

More specifically, we present in this chapter an original rule-learning algorithm for symbolic NLP that was designed to learn rules for extracting keywords marked in its training sentences. These keywords represent information that a potential user or system designer seeks to mine from the sentences. As such, the information sought varies with each application. The algorithm developed in this study is designed to extract the main verb of the sentence along with its subject and objects. Our method differs from other recent developments in the field of symbolic NLP in the implementation of a hierarchy of parts-of-speech at the core of the learning algorithm. This new approach makes the rules dependent only on sentence structure rather than on content as in ontology-based methods [80]. Similarly, our method employs rules that are independent of context- and domain-specific information.

To put in perspective the design and scope of our research, this chapter is structured as follows. The next section presents a complete description of our method and makes the case

for its original contribution. Experimental results generated through an implementation of this method are elaborated and thoroughly analysed in Section 4.3. We compare this extraction method with that of Chapter 3 in Section 4.4, and give directions for future work in Section 4.5. Finally, we end the chapter with concluding remarks in Section 4.6.

4.2 The Rule-Learning Algorithm

4.2.1 Part-of-Speech Hierarchy

It is commonplace to define a *part-of-speech* as a linguistic category of lexical items (generally words) that share common syntactic or morphological characteristics. This definition is however quite vague and underscores the fact that there is no agreement among grammarians on what exactly constitutes an independent part-of-speech. To illustrate this point, consider for a moment that the Penn Treebank, with its set of 46 parts-of-speech, is a simplification of the Brown Corpus set, that has 87 parts-of-speech, and is in turn dwarfed by other corpora, such as the Lancaster-Oslo/Bergen tagset and the London-Lund tagset, that contain 135 and 197 parts-of-speech respectively [57]. At the other extreme, the "correct" English grammar taught in schools asserts that there are only eight parts-of-speech in the English language: the noun, verb, adjective, adverb, pronoun, preposition, conjunction, and interjection. To be sure, this list is modelled after Latin grammar, which was designed with eight parts-of-speech to resemble closely the more civilized (at the time) Greek grammar, that included eight parts-of-speech on account of the early and influential grammatical work of Dionysius Thrax in the Second Century BC [63]. Thus it appears that the eight part-of-speech classes carried over to English grammar originate from tradition rather than from grammatical necessity.

This study's core contribution to the field of NLP is the design of an original part-of-speech hierarchy. Indeed, while the design of hierarchical systems, such as ontologies or WordNet, has undeniably been gaining importance and recognition over the past years, no part-of-speech hierarchy has ever been developed before. The need for such a hierarchy does exist, as evidenced by the fact that some work in the field has been subtly pointing to it. For

example, as was pointed out earlier, the Penn Treebank is a simplification of the Brown Corpus tagset; it merges some of the Brown Corpus tags together and reduces the size of the tagset almost by half, but the lost tags can be recovered [57]. On the other hand, similar tags in the Penn Treebank use common prefixes. For example, the various parts-of-speech representing adjectives start with JJ, and those for verbs start with VB. This setup can be seen as a three-level hierarchy, with the prefixes as the highest, most general level, the Penn Treebank as an intermediate level, and the Brown Corpus tagset as the lowest, most precise level. However, the authors of the Penn Treebank never actually make this logical step, and the need for a part-of-speech hierarchy is never fulfilled or explicitly recognized. This study seeks to fill that gap.

To help visualise the different levels of our part-of-speech hierarchy, a graphical representation of the complete hierarchy developed in this section is presented in Appendix B. To be sure, the purpose of the hierarchy originating in this research is to group specific words into more and more general part-of-speech groups. In this regard, it is logical to say that the lowest, most basic and specific level of the hierarchy is comprised of the lexical items taken from the sentence. The second level of the hierarchy corresponds to the first stage of the generalization of the lexical items. At this level, lexical items are grouped in the 46 parts-of-speech of the Penn Treebank, discussed in detail in [74]. This set of part-of-speech tags is listed in Appendix A for future reference.

As can be seen in the appendix, we have made one notable modification to the Penn Treebank, which is the addition of a special “blank” symbol that stands for a missing word. This modification will allow rules and segments of different lengths to be compared and matched together, as the extra words in the longer sentence will be paired with “blank” words in the shorter one.

At the second stage of generalisation, parts-of-speech that represent morphological variations of the same words, such as “singular common noun” and “plural common noun”, are brought together into *morphological categories* that constitute the third level of the hierarchy. Moving two levels beyond this third one, we reach the level in the hierarchy where

items are grouped into *lexical categories*. We have defined nine such categories for words, and three for the various symbols and non-native words that are commonly encountered in English texts. Each lexical category also includes the special "blank" symbol. Our set of lexical categories reads as follows:

- Adjective: A word that modifies the attributes of a noun.
- Adverb: A word that modifies a verb, adjective, another adverb, a clause or a sentence, but not a noun.
- Auxiliary verb: A verb whose function is to give information about the following verb.
- Conjunction: A function word that serves as a connector between two parts of a sentence. This category includes the preposition and the more general adposition, which are sometimes considered separate but overlapping categories.
- Determiner: A function word that modifies the reference or quantity of a noun.
- Interjection: A word that expresses emotion or represents an exclamation on the part of a speaker.
- Noun: The name of a person, place or thing.
- Pronoun: A function word that substitutes for a noun phrase.
- Verb: An action, occurrence or state of being. This category excludes auxiliary verbs.
- Punctuation: A symbol creating a division within a sentence or between sentences.
- Numeric: Numbers and symbols related to numbers.
- Foreign word: Any word in a language other than English.

The level of the hierarchy that falls between the morphological and lexical categories defined above will be called the *sub-lexical category*. This level puts together items that belong to the same lexical category and which form a sub-group within this category. For example, the morphological categories "present participle verb" and "past participle verb"

can be seen as the sub-lexical category “participle tense verb” within the “verb” lexical category.

The level of the hierarchy above that of the lexical category will be designated as the *super-lexical category*, because it is the one where the lexical categories are grouped into four super-categories, namely

- Core word: A word that carries its meaning independently of any other (noun, verb and foreign word).
- Modifier word: A word that modifies another word (adjective, adverb, interjection and auxiliary verb).
- Function word: A word with no real meaning, that serves a practical function within the sentence (determiner, conjunction and pronoun).
- Non-word: A symbol that is not a word at all (numeric and punctuation).

Finally, the most general level of the hierarchy is the *universe* level, where all items are grouped into a single category.

4.2.2 Similarity between Words and Sentences

One way to compare two words or two sentences, to compute their similarity or merge them, consists in relaxing the constraints on one of them until it includes the second one, as suggested by Soderland [80]. In the case of two words, this is accomplished by finding their lowest common ancestor in an ontology. Then, the similarity between the words is the number of levels between each one and the common parent, while the merging is simply accomplished by replacing the words with their common ancestor. Likewise, for two sentences, the comparison is accomplished by pairing their words together, then finding the lowest common ancestor of each pair in an ontology. The similarity between the sentences is then computed as the average similarity between the pairs of words, and merging both sentences yields a sentence composed of the common ancestors of the original pairs of words.

However, it remains true that for the ontology to be valid, the rule-learning system must be confined to its specific domain, as in the case of Soderland's study [80]. On the other hand, although our research is not intended to be limited to a single domain, it is still possible for us to define a general hierarchy of categories. To this end, we have designed the novel part-of-speech hierarchy described in the previous section. Using this hierarchy, the merger of two rules is done by raising the level to which each word belongs in this hierarchy towards more and more general grammatical categories. For example, the rules "the sites" and "the dogs" can be merged into "the (plural common noun)", and that rule can then be merged with "a house" to become "(determiner) (common noun)". Moreover, in this proposed hierarchy, the similarity between two rules simply becomes the average cost of raising each of their words to the levels in the hierarchy needed to merge the rules. This cost is a function of the number of grammatical elements represented by the category, and of the semantic importance of the category. For example, nouns and verbs are more semantically important in a sentence, and therefore more expensive to merge, than adjectives, which are in turn more important and expensive than punctuation marks.

It is worth noting that the notion of cost we defined is not the only way to measure the distance between concepts in a hierarchy. In fact, the growing importance of the hierarchical lexicon WordNet in NLP research has led to the development of a number of new distance measures for this purpose, such as the Jiang-Conrath distance and Lin's universal similarity measure. Several of these measures are reviewed in [17]. Future work can investigate how to adapt these techniques from WordNet to our part-of-speech hierarchy, as well as which is most appropriate for our purpose.

4.2.3 The Rule-Learning Algorithm

The rule-learning algorithm adopted in this research is a supervised learning system based on the work of Stephen Soderland [80]. The fundamental idea behind Soderland's method is to devise the strictest rule possible to handle a sentence, and then relax it gradually in order to handle other, similar sentences. By applying the same principle, we have developed a new a

method to learn the rules needed to extract the main verb of a sentence, along with its subject and objects.

Our learning system relies on a training corpus of sentences, in which the correct words to be extracted are identified. Since our method is centered on a part-of-speech hierarchy that will be explained in the next section, the first operation that the algorithm must naturally execute is the part-of-speech tagging of the sentence. This operation associates each word in the sentence with its correct part-of-speech tag from the Penn Treebank set [74], and is performed using the Brill tagger presented in Section 3.2.1. Once tagged, the training sentences are manually divided into short segments containing at most one word to extract each. Each of these segments thus becomes the strictest extraction rule possible for that word in that context. The set of all the aforementioned segments constitutes the training corpus of rules for the rule-learning system.

As with any step in any system which involves manually handling data, the fore-mentioned splitting of the sentences into segments is a bottleneck in the implementation process. It is important to note, however, the fact that this is neither a hard nor knowledge-intensive task. Indeed, sentences present many "natural" split points, such as commas and conjunctions. Moreover, the split does not need to be very precise, and can be off by a word or two without negatively influencing the system. Indeed, as similar rules are merged together in the rule base using the process described in Section 4.2.2, the extra or missing words in some segments will be evened out in the merged rule, and replaced by more general categories of parts-of-speech. Important words that are sometimes missing in the segments will be replaced by lexical-level categories, which will allow the use of our blank part-of-speech to handle cases where the word is absent. On the other hand, extra words in a segment which are truly irrelevant to the rule will be generalized up to the Universe category, meaning that they can be anything at all. It thus appears that the splitting process is a simple one; so much so, in fact, that it will be possible to automate it. Future work can focus on implementing an automated sentence divider using based on these "natural" split points.

When the learning system receives a new sentence from the training corpus, it

immediately proceeds to add the sentence's rules to its internal rule base, and then computes the similarity of all pairs of rules. The next step is for the most similar pair of rules to be merged together. This merger is accomplished by generalising one of the rules in order to incorporate the second. After the merger is completed, the keywords are extracted twice from the corpus of sentences, first by applying the pre-merger rule base, and secondly by using the post-merger rule base including the merged rule. Once that is done, the precision and recall ratios of both extraction processes are compared in order to determine the best rule base. If the first extraction turns out to have the higher ratios, the merged rule is rejected and the two rules are kept in the rule base without modification. By contrast, if the second extraction has equal or higher precision and recall ratios, the merged rule is retained in lieu of the original pair and its similarity to each of the other rules in the rule base is computed. Both outcomes, however, lead the algorithm back to the initial point on the loop, explained above, from where the learning process will continue running until all pairs of rules have been considered. To provide a deeper understanding of the learning algorithm presented above, Figure 4-1 illustrates its structure and scope from an operational perspective.

```
A. Select the next sentence in the training corpus
B. Add all the sentence's rules to the rule base
C. Create the list of pairs of rules, with their similarity
D. While all pairs of rules haven't been considered
      a. Find the most similar pair of rules
      b. Merge the rules
      c. Create RuleBaseA, the rule base including the two rules
      d. Create RuleBaseB, the rule base excluding the two rules and
         including the merged rule
      e. Compute PrecisionA and RecallA, the precision and recall
         obtained when applying RuleBaseA to the training sentences
      f. Compute PrecisionB and RecallB, the precision and recall
         obtained when applying RuleBaseB to the training sentences
      g. If PrecisionB ≥ PrecisionA and RecallB ≥ RecallA
            i. Replace the pair of rules with the merged rule in the
               rule base
           ii. Discard all pairs of rules that used one of the old
               rules from the list
          iii. Add all pairs of rules using the merged rule to the
               list of pairs of rules to consider
      h. Else
            i. Discard the merged rule
```

Figure 4-1: Structure of the learning algorithm

In order to compute the precision and recall ratios, the system compares the keywords extracted by applying the rules to those correctly marked in the sentences. If a keyword is correctly extracted by the rules, it counts as a true positive (*TP*). However, a keyword that is extracted by the rules yet does not appear as a keyword in the marked sentences counts as a false positive (*FP*), whereas a keyword missed by the rules counts as a false negative (*FN*). Furthermore, if the system is trained to recognise different types of keywords, and encounters a keyword extracted as the wrong type, it counts it as both a *FP* and a *FN*. Once these three values are calculated, the precision and recall ratios are computed as per Equations (4-1) and (4-2) below.

$$precision = \frac{TP}{TP + FP} \tag{4-1}$$

$$recall = \frac{TP}{TP + FN} \tag{4-2}$$

To be sure, in the context of this research, the term "applying the rule base" refers to the process of extracting the subjects, objects and verbs from a corpus of sentences according to the rules. As part of this process, the algorithm we develop matches each part of the sentence to the most similar rule in the base. For instance, if a sentence contains the words "a duck", the algorithm would recognise that it must match it to the rule "(determiner) (common noun)" instead of, say, "to (infinitive verb)". This matching component of the extraction process will be described in greater detail in a later section. Once the matching is done, the subjects, objects and verbs that our system must find and that are already marked in the rules, can simply be extracted by identifying the matching words in the sentences. In the special case of a training corpus of sentences where the correct words to be extracted are known, the application of a rule base yields results that could be compared to the correct solution in order to estimate the precision and recall of the rule base.

4.2.4 Applying the Rules

The next major building block in our method is the procedure to apply the rules to a corpus of sentences, in order to extract the desired keywords. This particular procedure serves a dual

purpose. In the first place it serves to compute the precision and recall values of the various rule bases generated by the learning algorithm described previously, in order to compare them and pick the best one. Secondly, it serves to analyze real sentences using the final rule base learned by the algorithm.

It should be noted that, in the context of this study, the rules that make up the rule base are only a few words long, which makes them much shorter than normal sentences. In the circumstances, it can easily be seen that it will take a combination of several rules, in most cases, to cover an entire sentence. Obviously, a large number of such combinations is possible, and the best combination of rules to apply is the one that is most similar to a sentence, using the measure of similarity defined before. In this way, the problem of applying the rules becomes one of finding the most similar sequence of rules to cover a sentence.

The most similar sequence of rules is discovered by computing the similarity between each rule and the sentence, first at the initial word of the sentence, and then after the end of the previous rule, until the sentence is completely covered. This problem can be likened to that of searching a tree, in which each node is the choice of a rule, the cost of that node is the similarity of that rule to the part of the sentence to which it is being compared, and the cost of the path through the tree up to that node is the similarity of the sequence of rules up to that point. The tree is searched using a uniform-cost search algorithm. This algorithm finds the cheapest node (i.e. finds the sequence of most similar rules up to a point in the sentence), expands it (i.e. applies all rules at that point), and repeats this process until the cheapest node is at the end of the sentence. The structure of this search algorithm is illustrated in Figure 4-2. The advantages of the uniform-cost search are that it is optimal and complete [72], which means that it is guaranteed to find the cheapest sequence of rules for each sentence.

4.2.5 Rule-Learning Example

An example can help illustrate the rule generalisation process. This example, taken from the training of our prototype described in Section 4.3, will illustrate the successive merging of three similar rules into one.

```
A. Create the initial node at the start of the sentence, cost = 0
B. Create the node list, containing only the initial node
C. While the cheapest node in the list is not at the end of the
   sentence
      a. Remove the cheapest node from the list
      b. Expand the cheapest node
            1. While there are rules in the base
                  i.   Apply the rule at that point in the sentence
                  ii.  Create a child node containing this information
                  iii. Add the cost of applying the rule to the child
                       node's cost
      c. Add the children nodes into the list
D. Return the cheapest node
```

Figure 4-2: Structure of the uniform-cost tree-searching algorithm

The first two rules found to be similar in our example are "to the public" and "during the past week". Their respective part-of-speech sequences are "TO DT NN" and "IN DT JJ NN", the interpretation of which can be found in Appendix A, and in both cases the last noun of the rule is the one marked for extraction. These two rules are merged together using our algorithm presented in Figure 4-2 and our part-of-speech hierarchy illustrated in Appendix B. The new generalized rule obtained at the end of this process is "_ the _ _", where each underscore represents a word, and its part-of-speech sequence is "Connector DT Adjective NN". It is interesting to note that the adjective "past" in the second rule has no equivalent in the first rule. It is therefore matched to a blank symbol, and generalized in the merged rule to the higher-up lexical category "Adjective" which can represent any adjective or no word at all. Likewise, the prepositions "to" and "during", which have different parts-of-speech, are generalized to their common sub-lexical parent "Connector" in the hierarchy. On the other hand, the article "the", which is common to both sentences, remains unchanged in the merged rule. Similarly, the nouns "public" and "week", which both have the part-of-speech "NN", are simply generalized as any word with that same part-of-speech.

Next, the newly-created rule is merged with a third rule. This third rule is "in years", "IN NNS", and the word to extract is "years". The preposition "in" is paired with the "Connector" of the new rule and "years" is paired with the "NN" noun, which means that the other two words of the new rule, "the" and "Adjective", have to be paired with blank symbols. The final merged rule is thus "_ _ _ _", with the parts-of-speech "Connector Determiner Adjective CommonNoun" and with the final noun tagged for extraction. Since

the category “Connector” already includes the preposition “in” and the category “Adjective” already includes the blank symbol, these two elements are not changed during the merging. The part-of-speech NN, for singular common noun, is merged with NNS, plural common noun, and the word is generalized to the lowest parent in the hierarchy, which is the morpheme category CommonNoun. Finally, the determiner “the” is generalized to the lowest parent that includes the blank symbol it is being merged with, which is the Determiner lexical category in our hierarchy.

To finish, it is important to note that either one of these mergings could have been rejected by our algorithm if the merged rule caused a drop in the precision or recall of the extraction process when compared to using the two original rules, as shown in Figure 4-1. However, this does not occur in our example, and in both cases the merged rule is accepted and used to replace the two original rules in the rule base.

4.3 Experimental Results

4.3.1 Setup

As noted earlier, the algorithm developed in this study can be trained to extract any desired information from a text, provided the information could be obtained in the form of keywords in the sentences. To demonstrate this feature, the system was trained so as to extract the main verb of a sentence, along with the nouns that are its subjects and objects. However, the fact that most sentences typically contain many verbs and nouns that must be ignored constitutes a serious impediment to this extraction process. For example, in the sentence “John is prepared to leave for Berlin”, the verb “leave” is the verb that the system must extract, while the other two verbs should be ignored. Similarly, in the sentence “The president of the country made a speech”, the correct subject to extract is president, not country. Hence, the distinction between the nouns and verbs that should be retained and those that should be discarded is one of the main challenges that the extraction process must be capable of handling. The following section will demonstrate how our algorithm can effectively address this challenge.

The data used to train our system comes from the Brown Corpus [32]. This dataset is a corpus of texts written in American English and compiled in 1961. It is composed of 500 sample documents selected to reflect the spread of domains that the American public were likely exposed to at that time. The documents in the corpus thus cover a wide range of topics, from news coverage to religious texts, from industrial reports to detective fiction. By adopting this entire corpus as training data, our system should be able to learn rules to handle a wide variety of sentences reflecting many different writing styles. From a practical perspective, we decided to set off the demonstration by initially limiting the scope of the rule-learning system to the business domain of the corpus (samples A26-A28), and to train our system with a random sub-sample constituted from 10% of that domain. This sub-sample represents 269 training rules, or 27 sentences.

4.3.2 Results and Discussion

It is interesting to examine in the first place the behaviour of the number of rules in the rule base even as the learning process evolves. Figure 4-3 illustrates the size of the rule base after each training sentence has been processed. As the figure shows, the rule base quickly grows to the 20-rules range, but its rate of growth slows down considerably after this. To be sure, the rule base still gains new rules after reaching the 20-rule level, but it does so at a slow and irregular pace, and even eliminates rules at some points in the training process. In short, after learning 20 rules from the first 50 training rules, the learning process only discovers 13 additional ones from the next 220 training rules. This result seems to indicate that there is a limited number of rules to be learned, or alternatively that there is a limited number of atomic sentence structures that are combined in various ways as needed to form complete sentences.

To gain a deeper understanding of the learning process, it would be informative to analyze the behaviour of the precision and recall values of the rule base after each training sentence has been processed. To compute these values, we applied the rule base on the unseen 90% portion of the Brown Corpus' business domain that was not used in the training process. Overall, the results, which are illustrated in Figure 4-4, show that both the precision and recall values increase as the system becomes more and more trained. More specifically,

the recall value starts off higher than the precision value but falls during the processing of the first 80 training rules, as that of the precision increases. Figure 4-4 also reveals a sharp temporary rise of a few percentage points in the recall value in the 150-220 training rules range, which is matched by a corresponding drop in the precision value. This is a typical trade-off in classification systems, where an increase in precision is usually offset by a decrease in recall, or vice-versa [80] [76]. It is however interesting to note in this regard that, by the end of the learning process, the recall value has returned approximately to its initial value. The precision value itself has a slow and irregular rate of increase, but gains nonetheless 10% from the start to the finish of the training process. These results indicate that, with further training, both the precision and recall values of our system might be improved.

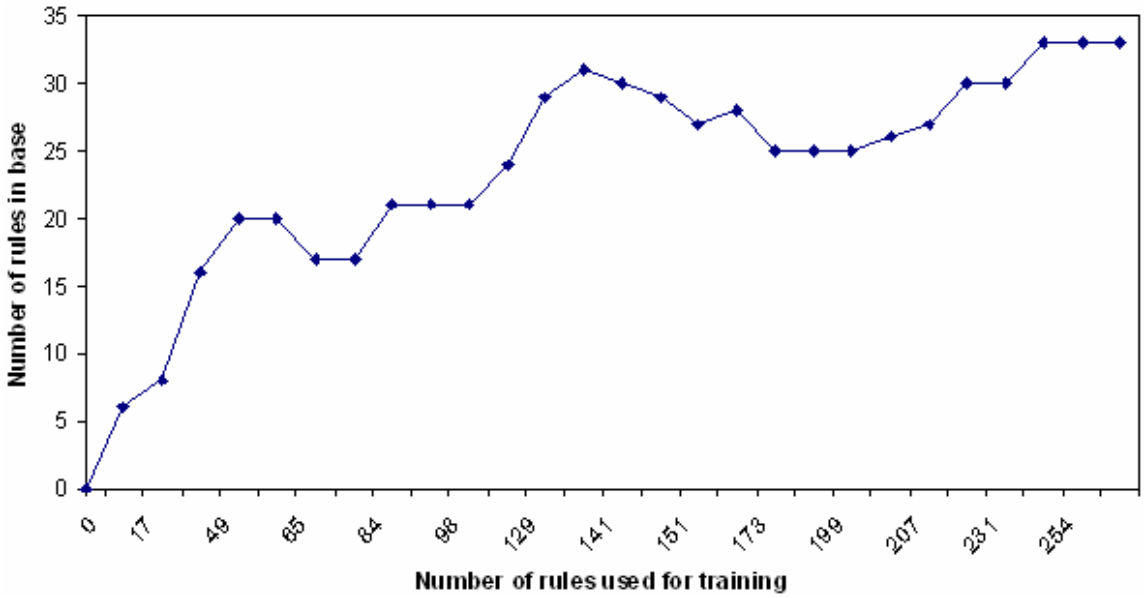


Figure 4-3: Size of the rule base during training.

The final results of the test of our algorithm using the useen 90% portion of the Brown Corpus' business news domain are provided in Table 4-1. For comparison purposes, the table also includes the precision and recall values of both a statistical and a symbolic NLP algorithm. The symbolic algorithm represented here is the one developed by Soderland [80] which, as we noted, underlies our work. The experiments done in [80], the results of which are include in Table 4-1, were performed on a corpus of business news articles similar in

nature to the portion of the Brown Corpus we selected, and thus their results are comparable with our study. The statistical algorithm selected is the Minipar parser, a statistical dependency parser which was chosen because it was trained on the Brown Corpus to extracts, among other things, the same subjects, verbs and objects for which our algorithm has been trained, and also because it has been thoroughly studied and evaluated in [52]. It is worth noting the fact that both Soderland's algorithm and Minipar were trained using much bigger training corpora than the one used for our algorithm.

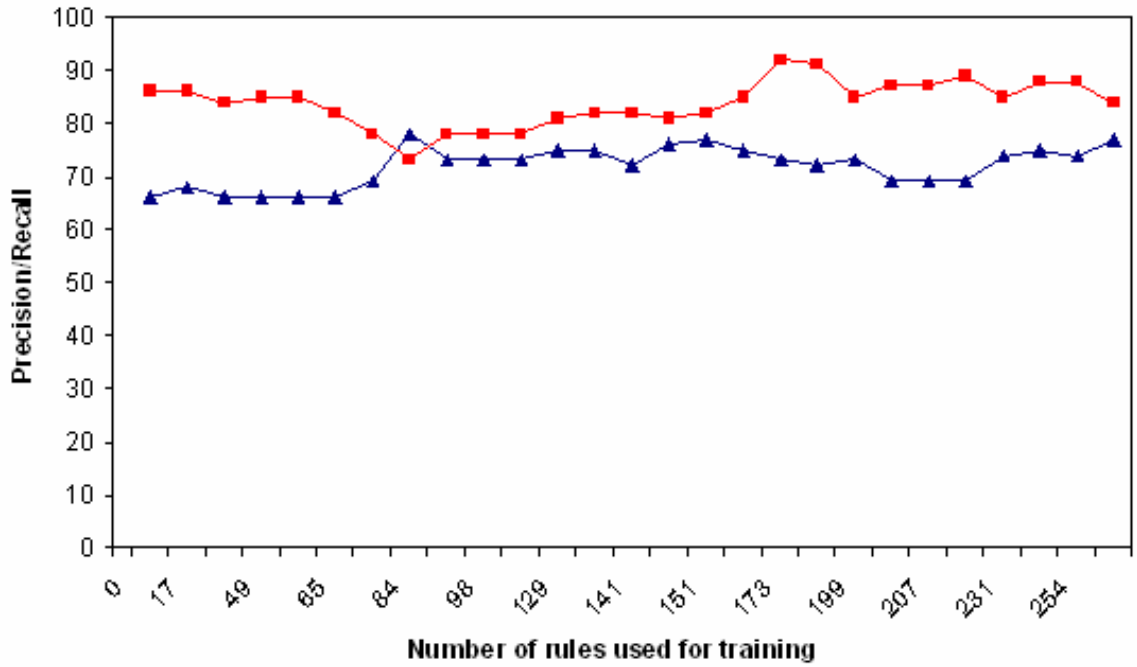


Figure 4-4: Precision (blue triangles) and recall (red squares) of the rule base during training.

Table 4-1: Experimental results of our algorithm and two reference algorithms.

	Our algorithm	Minipar	Soderland
Precision	77%	88%	70%
Recall	84%	75%	56%
Limited width	No	No	Yes
Limited depth	No	Yes	No

As can be observed in Table 4-1, our method compares quite favourably with the two reference algorithms. Our system exhibits a better recall than the two references, and a better

precision than Soderland's algorithm. Moreover, our algorithm's field of application seems to be neither limited in width nor in depth as is the case for the two algorithms in reference. Indeed, while Minipar is quite good at the task of finding the subjects and objects of sentences, its performance degrades severely when it is used to extract information that requires a deeper understanding of the text. For example, on the task of finding conjunctions, Minipar only achieves a precision and recall of 67% and 50%, while for relative clauses those values drop even more to 52% and 56%. This rather poor performance is due to the fact that discovering this information often requires resolving attachment ambiguities, which Minipar tries to do by using decision rules that are often, but not always, correct [52]. In other words, Minipar relies on statistical information, which by its very nature only gives an overall view of the text and is not always well-suited for the extraction of specific details from sentences (as explained in Section 2.4). This limitation of Minipar contrasts with the capacity of our algorithm to successfully extract more specific information from the sentences, by simply adding new and more specialized rules to its rule base and without a significant drop in precision or recall. In the particular case described above, for example, our method would learn extraction rules representing how to handle each individual ambiguous case, and would then match a new ambiguous sentence to the most similar rule in order to know how to resolve it.

On the other hand, the width limitation of symbolic algorithms, such as Soderland's, comes from their reliance on domain-specific ontologies or other specialized sources of information. This type of information cannot be ported to other domains, which severely limits the scope of algorithms that are based on it. By contrast, our algorithm does not rely on any such constrained information, and the rules generated with it should therefore be applicable in any domain. In order to fully illustrate this fact, the final rule base was subjected to two more tests beyond the one described above, in which the unseen 90% portion of the Brown Corpus' business news domain was used. To conduct the second and third tests, we selected the Brown Corpus' science-fiction domain (samples M01-M06, 830 sentences) and the transcript of a live television debate between three people (358 sentences). The reason for this selection of domain is that they are quite different from the business news

domain that was used to initially train the system. By design, the purpose of these tests is to demonstrate that the rules learned with our method are not domain-dependent. We manually tagged the correct keywords that should be extracted in our two test corpora, in order to evaluate the performance of the algorithm. In the tests, our method exhibits a precision of 83% and a recall of 87% on the science-fiction corpus, and a precision of 61% and a recall of 91% on the debate transcript. The positive results we obtain when we apply the rule base extracted from the business news domain to sentences from very different domains clearly demonstrate that our method is not domain-specific. Taken together with the previous analysis performed above, these results suggest that the algorithm developed in this study not only outperforms statistical and symbolic NLP algorithms, but also exhibits a wider field of application then symbolic algorithms and a deeper field of application then statistical algorithms.

It is worth noting the main drawback of our method at this point, which is its requirement in computational time. Indeed, in the worst case, each word of a sentence segment will be compared to each word of each rule in the rule base, leading to a cubic algorithmic complexity. In fact, it took over a week for our implementation of the system to train itself using the portion of the Brown Corpus' business news domain that we used as a training corpus. Clearly, optimising the method will be an important focus in future research.

Our analysis also provides insight into the reason why the rule base actually performs better on the science-fiction dataset than on that of the business domain. Notably, this outcome is contrary to the expectation that the rule base should perform better on unseen in-domain data than on out-of-domain data. The explanation for this surprising result is that the sentences found in science-fiction literature are typically simpler than those in business-related news articles. Indeed, the sentences of the science-fiction domain are on average 17 words long and contain five keywords to extract, while those in the business domain are on average 25 words long and have seven keywords to extract. Moreover, a closer look at the rules that were applied to extract the keywords reveals that the processing of the science-fiction domain relied predominantly on seven short and simple rules, which indicates that the sentences were generally simple and straightforward. By contrast, the analysis of the business

domain's sentences required a much greater variety of rules from the rule base, indicating that the sentences had a more varied and complicated structure. These observations could prove quite useful in future research.

Using similar reasoning, we can explain why applying the rule base on the televised debate transcript yields a very high recall but a low precision compared to the business news corpus. Indeed, spoken sentences are typically shorter and more straightforward than written sentences. This allows the rules to recognise the keywords more reliably and to miss fewer of them, thus improving the recall value. Unfortunately, spoken sentences also make use of sentence structures that are not found in business news articles, and for which the rule base has no corresponding rule. For example, spoken sentences can make heavy use of interjections. Speakers also often repeat words for emphasis, or stop in mid-sentence when they are interrupted by someone else. Without the correct rule to handle such sentences, the system will try to apply the most similar rule found in the rule base. This most similar rule, however, may not be appropriate, and may mark the wrong word as keyword. This causes the drop in precision value.

4.4 Limitations of the Extraction Method

Before concluding this presentation, it would be instructive to contrast the limitations of the new extraction process discussed in this chapter with those of the extraction process we presented in Chapter 3.

As we saw earlier, the first and foremost limit of the extraction process discussed in Section 3.4 comes from its overly-simplistic underlying assumption that all sentences follow a straightforward subject-verb-object structure. To be sure, we made an attempt to mitigate the incidence of this assumption by using a number of heuristics, but quickly realized that this correction is by its very nature constrained to handle a limited number of cases in the potentially limitless universe of sentence structures. By contrast, the new extraction process expounded in this chapter has the advantage of not relying on any such over-simplistic assumption. Indeed, the main assumption behind this new extraction process is that sentences can be reduced to a combination of atomic structures, each a few words long. And as shown

in Section 4.3, our experimental results seem to confirm the validity of this assumption. Furthermore, these atomic structures can be combined in any order and any number of times, thus allowing our system to handle a limitless variety of complex sentence structures. Taken together, these features are quite superior to those of the triplet extraction system of Chapter 3 which, it will be recalled, was limited to the few complex sentence structures it was pre-programmed to handle using its heuristics.

The second limit of the extraction process of Chapter 3 comes from its reliance on the Brill tagger, which can mislabel the words in a sentence. Given that the heuristics were applied directly to the tagged sentence returned by the Brill tagger, the mislabelled words were consequently a major concern for that process. To be sure, our new extraction process also makes use of the Brill tagger, and is therefore subject to the same mislabelling problem. On closer inspection however, we saw in the case of the process presented in this chapter, that words are generalized to more and more general categories in our part-of-speech hierarchy. This means that mislabelled words can be generalized away. Indeed, a sentence segment with one mislabelled word can still be associated with its most similar rule, but only with that specific word replaced by the lowest category common to it and to the correct word in the rule. It follows that in the new process, this mislabelling problem is no longer the major limiting factor it was in Chapter 3.

The third shortcoming of the process of Chapter 3 exposed in Section 3.4 is the absence of a way to handle anaphora resolution. Unfortunately, this problem is also present in the new system discussed in this chapter. Furthermore, as explained in Section 3.4, the task of anaphora resolution is a major challenge in NLP that lies outside the scope of this research. Nevertheless, it is interesting to note that our new system could potentially be used for this purpose, by learning rules that can match an anaphor with the word it stands for within a sentence segment. Such a major undertaking could be the subject matter of future research.

The final limit of the extraction process of Chapter 3 was its need for manual work, a problem which limited the portability of the method. The process described in this chapter does not suffer from this limitation. Indeed, the only manual task in the method is that of

splitting the sentences of the training corpus into rules for the learning algorithm, and we have explained in Section 4.2.3 how that task can easily be automated. In addition, both the work of Chelba [19] and our own theoretical and experimental analysis in this chapter show that the rule base learned can be ported to other domains. This means that the learning process, and the manual work it requires, only needs to be done once, and makes this method much more portable than the one presented in Chapter 3.

In the discussion of this chapter, we have demonstrated that our new part-of-speech method obtains high precision and recall results, while combining the unlimited width of statistical algorithms with the unlimited depth of symbolic algorithms. For comparison, in experimental tests, we found that the triplet-extraction method of Chapter 3 obtains slightly better precision and recall values in the business and science-fiction corpora introduced in Section 4.3. We should bear in mind however, that these are two of the corpora that the process of Chapter 3 was specifically designed to handle, and for which the heuristics of Section 3.2.2 were fine-tuned; the good performance of these heuristics in those corpora should therefore come as no surprise. Moreover, these results were obtained by sacrificing both the width and depth of the algorithm. Indeed, as is the case for any symbolic algorithm that relies on a finely-tuned rule set designed for a specific corpus, our triplet-extraction technique cannot be expected to perform well on different corpora. And the heuristics used are designed for the sole purpose of extracting the key verbs and nouns from sentences. They cannot be easily adapted for another purpose, and in fact, if there was a need to extract more precise information, it would be necessary to come up with a completely new set of heuristics. Taken together, these observations show that the method of Chapter 3 has both a limited width and a limited depth.

In light of this analysis, it is clear that the new extraction process expounded in this chapter is robust enough to overcome the major shortcomings that plague the process of the previous chapter.

4.5 Future Developments

4.5.1 Citation Classification

It is interesting to note that our study bears some resemblance to that of Garzone [33], in which the author presents an innovative citation classification scheme. Garzone's research is motivated by the need to develop informational tools that can automatically extract and organize the citations found in scientific papers. Indeed, the general tendency in scientific papers is to present the various approaches taken by previous studies related to the topic under examination in order to provide a sketch of the setting in which the research is developed, and to give methodological details, experimental results, counter-arguments, and a host of other relevant information to the reader. All this wealth of information is lost however in the paper's "references" section, where studies are simply listed either in alphabetical order or according to their order of appearance in the paper. The search tool proposed by Garzone is designed to overcome this shortcoming by generating automatically a more informative classification of the references used in a study in order to show their relevance to the topic under examination.

The citation classifier put forward in [33] operates in a succession of steps. The initial step consists in converting the article to be analysed from its original format, such as postscript, to a regular text file. In the second step, the sentences containing references to other studies are extracted from the text, and the section of the paper where such sentences appear is noted. Following that, each sentence is analysed using a regular part-of-speech tagger as well as a syntactic parser, in order to accurately identify the part-of-speech of each word. In the final step, the information about the sentence, its position in the article, and its parts of speech, are fed as inputs into a semantic parser. The role of this semantic parser is to determine the exact classification of the references mentioned in the sentence. It does so by comparing the sentence to a set of semantic grammar rules, and then assigning to it the corresponding category of the semantic grammar rule that matches the sentence.

Still according to [33], a semantic grammar differs from a syntactic grammar in that the latter is general enough to be applicable to any context. The level of flexibility that

characterises syntactic grammars is not however necessary when designing an application for a specific domain, such as citations in scientific literature for example. Indeed, the syntactic rules in this case could be formulated more specifically by replacing their general lexical elements with domain-specific semantic elements. This switching of elements makes the rules much easier to use in a natural language application, as domain-specific elements are easier to recognise and match in the sentences.

The rules of the semantic grammar used in [33] are learned from example. In this context, a number of sentences are provided for each class of reference use. The key part of each sentence, which contains the reference, a verb, and class-specific cue words, is isolated, and the rest of the sentence is discarded. Then, the rules are formed by merging together those key segments that use the same verb, regardless of verb tense and of class. In the process of this merger, identical words are maintained, while differing words are generalized into groups of words and unnecessary words are dropped. This yields a rule general enough to recognise similar sentences, but specialized to a limited number of classes. The fine-grained distinction between these classes can still be recovered, however. When a sentence in an article is matched to a rule, the words of the sentence that correspond to the generalized words of the rule provide the necessary information needed to determine to which of the classes covered by the rule the sentence actually belongs. By using this two-level classification system, Garzone is able to limit his semantic grammar to 12 rules.

At first glance, it looks as if some similarity exists between our part-of-speech rules and Garzone's semantic grammar rules. Indeed, both sets of rules are learned from example, and are formulated by generalizing similar sentence segments into single rules. On closer inspection, however, it becomes clear that our method provides a far more complete and rigorous framework for mathematically computing the similarity between sentence segments and for merging them together. For example, our algorithm is not limited to merging rules that use the same verb, and our part-of-speech categories are more versatile than groups of words observed in training examples. To be sure, our method does not allow for a two-level classification system, and would generate individual rules for each class, even if some of these rules differ only by one keyword. By contrast, in Garzone's system, such rules are

merged together and the different classes are recognized at a later stage with the help of the different keywords. For all practical purposes, our method thus appears to be equivalent to that of Garzone in this specific regard. This observation leads to an interesting suggestion, namely to substitute Garzone's semantic grammar with our part-of-speech rules. This modification would give the citation classification system a new level of flexibility, which will include for example the ability to generalize verbs, or to generalize nouns into a noun part-of-speech instead of constraining them to a limited group of nouns observed in the training data. This change would in all probability enhance the performance of the overall system, although the precise degree to which it would do so is hard to evaluate in advance. Still, we believe this to be a very interesting and promising development to investigate in future research.

4.5.2 Incorporating Other Information

Although we've shown in this chapter that our method can get good results using only parts-of-speech, other authors [10] believe that this information alone can be too coarse to extract finer grammatical or semantic relationships. Consequently, an interesting future development for our method would be the addition of grammatical information. This information is more precise than part-of-speech tags, and taking it into account would improve the precision and recall of our method, as well as make it capable of extracting more fine-grained information from the sentences. In fact, the Penn Treebank already defines a grammatical function tag set, so including them in our method appears to be the next logical step for our work.

One possible way of adding this information, which would be consistent with our current method, would be to create a new grammatical-function-tag hierarchy. Each word in the rules could then be represented by two tags, one for its part-of-speech and one for its grammatical function in the rule. Next, comparing two words in two rules would be done as described in this chapter, by raising the words to their lowest common ancestor, except this would now be done in both hierarchies independently. The similarity value between the words would take into account the distance from the word to their common ancestor in both hierarchies. This

setup would leave the merging of rules in the learning stage of Section 4.2.3 unchanged. Likewise, applying the rules would still be done as in Section 4.2.4, but by taking into account both hierarchies.

4.5.3 Other Developments

From a practical perspective, this study's method has implications for other projects dealing with common NLP challenges. For instance, in situations where keywords marked in the training corpus only represent the central ideas of the texts making up the corpus, our proposed system becomes an appropriate tool for document summarization. Alternatively, by computing the frequency with which each rule of a rule base is used in dissimilar domains or in texts written by different people, our new method can be effectively applied in a domain classifier or in an author identification system. Another interesting application is in relation to a word processing software's automated correction system. In that context, the method can provide a formalized mechanism for comparing the user's text to a rule base of proper English sentence structures. By so doing, it can help identify bad sentences as those that are very different from the rule base, signal mistakes to the user and even suggest more syntactically-correct revisions of the sentences. These examples underscore the idea that the proposed method could be effectively applied to various NLP problems and tools. It will befall future research to explore the practicality of these possibilities.

4.6 Conclusions

This chapter introduced an NLP method designed to extract information-rich keywords from English sentences. The method involves first learning a set of rules that guide the extraction of keywords from parts of sentences. Once this learning stage is complete, the method can be used to extract the keywords from complete sentences by matching these sentences to the most similar sequence of rules. The key innovation in our method is the part-of-speech hierarchy that lies at its core. By raising words to more and more general grammatical categories in this hierarchy, the system can compare rules, compute the similarity measure, and perform the learning. The theoretical development and the experimental results discussed

in this study confirm that our method can be used to perform in-depth analyses of texts from any domain; but this comes at the cost of a higher computational complexity. These arguments appear to indicate that our proposed method will outperform both traditional statistical and symbolic NLP methods at their respective tasks, albeit in a more computationally-intensive manner.

At the close of the extraction process, each free-text English document in the training and testing corpora that were provided to the system is replaced by a series of triplets that represent the actions described in that document. In the next phase of our research, the triplets of the training corpus will be used to compute the possibility distribution corresponding to each domain. This will be the focus of Chapter 5.

Chapter 5
Possibility Distribution

5.1 Introduction

In the previous two chapters, the system was given a corpus of un-annotated English documents sorted by domains and transformed it into a collection of triplets that represent the actions described in those texts. Henceforth, each domain is associated with a series of subject-verb-object triplets. Some of these are common triplets present in several or all domains; others are specialized triplets found in only a few domains or in a single domain. Some triplets have virtually the same meaning and should be lumped together while others have very different meanings. The focus of this chapter documents our efforts to develop a method to extract the meaning and domain information contained in the triplets and to represent this information in a formal mathematical expression to facilitate the use of the information in practical applications.

For most of this chapter, each subject-verb-object triplet is split into a subject-verb pair and an object-verb pair. This was done to avoid data sparseness: pairs of words occur far more frequently than triplets of words. This subdivision was done on the assumption that the meaning of the subject-verb pair and that of object-verb pair are discrete; although both pairs influence the meaning of the triplet, each one carries its own meaning independently of the other. To simplify the notation, we will refer to these two pairs as *noun-verb pairs* in the remainder of this thesis with the understanding that we will distinguish whether a noun category is used as a subject or an object.

The mathematical expression we propose here is computed using a three-step method. In the first step, we compute the semantic distance between each noun-verb pair using a technique similar to that proposed in the literature by Rieger [70]. The second step consists of computing the conditional probabilities of the domains given each pair. To accomplish this, we adapted Bayes' theorem to manipulate the data in our research. Finally, in the last step, a possibility distribution is computed from the semantic distances and the probabilities of each

domain using a mathematical development we devised for this purpose. This three-step method will be examined in detail in the remainder of this chapter.

In the following sections, we present the mathematical foundation of each step in our method, and the possible outcomes are presented and illustrated graphically whenever feasible. Just as importantly, the close parallel between this theoretical discussion and its real-world implications are highlighted with examples drawn from the SchoolNet dataset used in the practical implementation presented in Chapter 6. Considering both these theoretical and experimental results permits an evaluation of the proposed method's validity.

5.2 Mathematical Foundations

As is the case with any scientific advance, our research is firmly rooted in the work of other scientists. Specifically, we adapted notions from the research of Rieger [68], [69], [70], and Zadeh [110], [26], in addition to using Bayes' Theorem. In all cases, we introduced modifications to the original material which we will present in the remainder of this chapter.

In this section, we will present the mathematical principles on which our work is founded. Accordingly, we will begin by introducing some necessary notions regarding the Zipf-Mandelbrot law and semantic understanding on which Rieger based his work, before moving on to present the steps for computing Rieger's semantic distances. Next, we will briefly touch upon Bayes' theorem. Finally, we will give a two-part introduction to Zadeh's possibility theory, focusing first on some conceptual notions, and second on the mathematical development. In all cases, we will pay special attention to how these mathematical notions are applied in our research.

5.2.1 The Zipf-Mandelbrot law

To set off this mathematical analysis, we will make the assumption that each domain of the training corpus is made up of a very large number of triplets. Since our method learns its possibility distributions from these training examples, this assumption simulates the method's ideal operating conditions. Moreover, under this assumption, we know that the

frequency of occurrence of the individual words in each domain will follow the Zipf-Mandelbrot law [55].

The Zipf-Mandelbrot law finds its origin in [113], when Zipf observed that people, in general, seek to minimize the predicted amount of effort they will have to make over the long term. This observation has very significant consequences in the field of NLP. To visualise its impact, we can imagine a conversation between two people, a speaker and a listener. The speaker makes an effort by choosing the words that can best represent the ideas and concepts he is trying to convey. His efforts would be minimized if his vocabulary were composed of only a few words of broad meaning, or at the limit of only one word that can mean everything. On the other hand, the listener makes an effort to figure out the meaning of the words he hears, in order to understand the speaker without ambiguity. His efforts would be minimized if his vocabulary were composed of words that each have one and only one very specific meaning. Such a vocabulary must necessarily be comprised of a lot of words. In the real world, of course, no one is bound to be speaker or listener all the time, and people seek to minimise their efforts by reaching a compromise between these two extremes. In this balanced state, a normal spoken or written discourse will be composed, in major part, of a few often-repeated words with broad meaning, complemented by several rarely-used words with a precise meaning. Follow-up research [55] showed that when we rank the words in a discourse according to their frequency of occurrence, we obtain the following hyperbolic relationship:

$$f = P(r + \rho)^{-B} . \tag{5-1}$$

This equation is known as the Zipf-Mandelbrot law, where f is the frequency of a word, r is its rank, and P, B and ρ are the parameters of the discourse analysed. When applied to the Brown Corpus for example, the values of the parameters are $P = 10^{5.4}$, $B = 1.15$ and $\rho = 100$ [56], and the resulting relationship is illustrated in Figure 5-1.

We can now make educated statements about the nature of the words with low and high ranks in the Zipf-Mandelbrot distribution. In line with our earlier discussion, we know that words with the highest frequencies will be general words of broad meaning. Moreover, in

this analysis, we consider domain-specific distributions. Hence, it stands to reason that important domain-specific words will also be used frequently and will be assigned a high rank. On the other hand, words with low ranks are those that appear rarely within a domain. Logically, they will be mainly composed of domain-specific words originating from other domains, and possibly of a few erroneous words resulting from errors encountered in the triplet extraction stage.

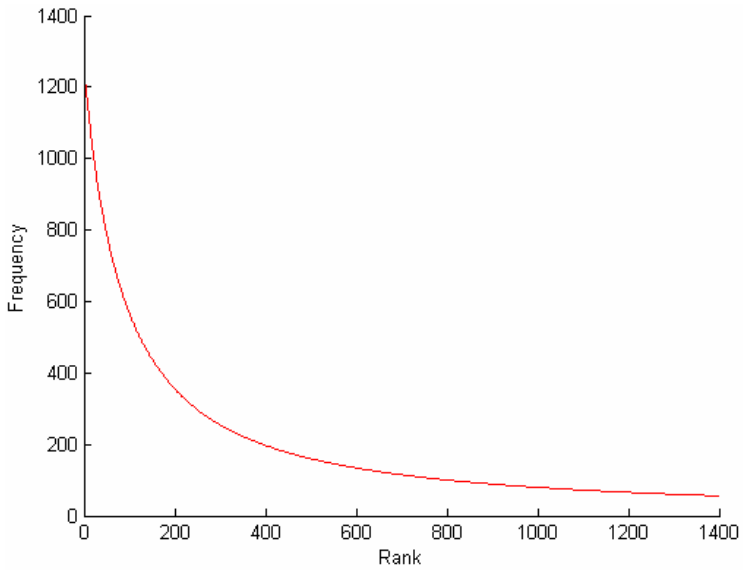
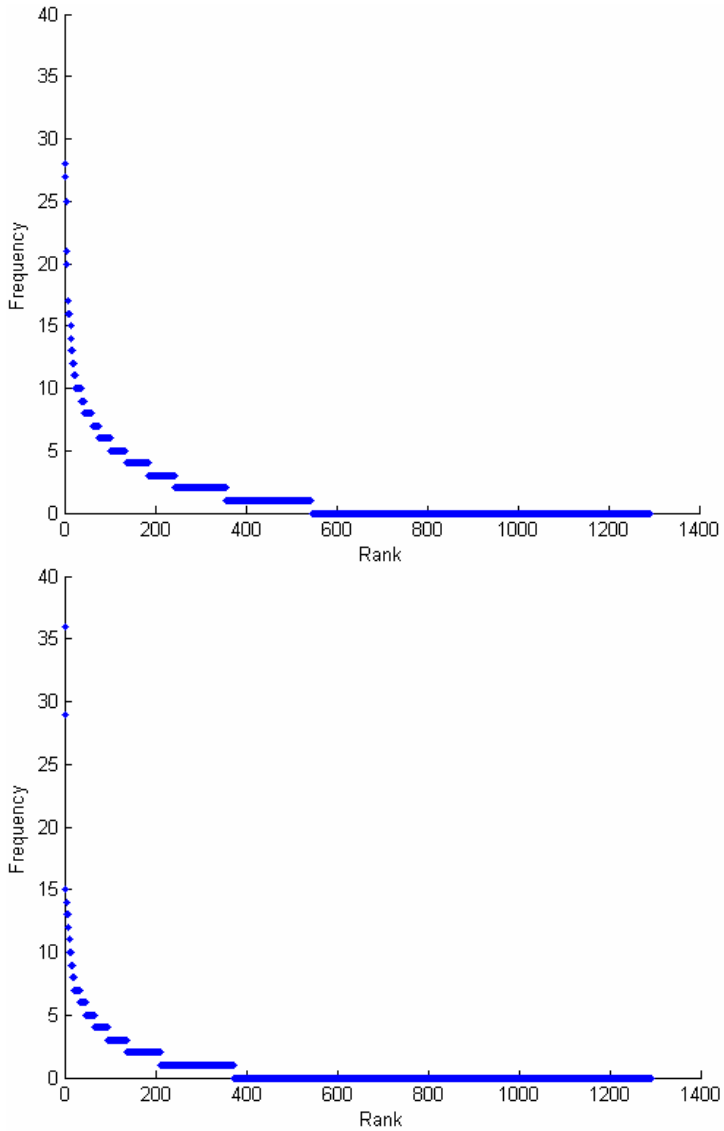


Figure 5-1: Ideal plot of the Zipf-Mandelbrot law.

To be sure, we can examine how the training corpus used in our implementation of the method compares with this theory. To this end, the frequency-rank plots of individual words, specifically of the noun categories and the verbs, in two different domains of our SchoolNet training corpus are shown in Figure 5-2. The two domains selected are the "social studies" domain and the "sciences" domain. It can be seen from the figure that both distributions follow the shape predicted by the Zipf-Mandelbrot law. Furthermore, an examination of the specific noun categories in each domain shows that the higher-ranked words include, as expected, both general words and important domain-specific words. For example, among the higher-ranking words and categories of both domains is a noun category containing general research-related words that commonly occur in all academic domains, and which also ranks



(a)

(b)

Figure 5-2: Frequency-rank plot of the words in the social studies domain (a) and the sciences domain (b) of the training corpus.

high in several other domains of the corpus. We also observe that domain-specific noun categories, such as the one for artistic terms in the social studies domain and the one for scientific terms in the sciences domain, rank high in their respective domains but poorly in the other domain. However, it should be mentioned that, while the shapes of the curves in both examples of Figure 5-2 below undeniably follow the shape of the Zipf-Mandelbrot law, they are a lot steeper than the ideal curve presented in Figure 5-1. This is a consequence of the problem of data sparseness that hinders this implementation of our method. The size of

the training corpus of our system is simply too small, and as a result the statistical information computed from it is imprecise and noisy. As the training corpus grows in size, we expect the results to converge towards their theoretical ideal values.

5.2.2 Introduction to Semantic Understanding

The first step of the training process of our method is to represent the meaning of each noun-verb pair. This is done using a technique similar to the one proposed by Rieger [68], [69], [70]. The basis of Rieger's work is two-fold. To start with, he draws on the notion of *situation semantics* [8], which holds that the meaning of an expression is based on two situations:

1) Discourse situation: This notion of situation is what allows the expression's meaning to be interpreted;

2) Described situation: This notion of situation allows the expression's truth-value to be evaluated.

Following situation semantics, it appears that the meaning of an expression can be discovered by recognising similarities and invariants between situations in which the expression appears. In other words, the important information we must extract from the situations are the regularities in word usage. In the second place, Rieger linked the notion of situation to the idea of *language games* [96]. Wittgenstein had previously introduced language games, or the contextual-usage-meaning view, as an explanation for the way children begin to understand and use words. In his theory, children learn not by assimilating the definitions of words, but by discovering patterns and regularities in their usage. For example, when a child hears the word "blue" for the first time, he will not open a dictionary to look it up. Rather, he will hear adults around him describe various objects as "blue", and eventually he will figure out that the only thing constant every time he hears that word is a particular colour. However, Wittgenstein's work on language games was done at the level of the philosophical discourse. By linking it to the formally-defined notion of situation semantics, Rieger laid the theoretical basis for an empirical approach to learn word meaning using its observed usage in a normal text.

In essence, the assumption behind Rieger's work, as well as ours, is that the same learning technique that children use to discover the meaning of words can also be applied to computers. If there is a target word a computer needs to understand, rather than looking up its definition in an online dictionary, the computer could search for its occurrences in a corpus of typical English documents, find which words regularly co-occur with it, and understand the meaning of the target word only on the basis of that information.

Starting from the assumption that the analysis of a number of texts in order to isolate regularities in a word's usage can reveal essential information regarding the word's meaning, Rieger [70] developed an empirical model to discover these regularities. In fact, since Rieger wanted his model to rely only on observable regularities in the text, he was in essence trying to isolate lexical items that are regularly associated with the target word and may give insight into its meaning. The core of his method, which we have adapted for our purposes, was a two-level abstraction process that produced a set of *usage regularities* and a set of *meaning points*.

5.2.3 Correlation Coefficient

To extract the usage regularities, Rieger used a modified correlation coefficient. This coefficient computes the interdependence between any two lexical items on the basis of their frequencies in the texts of the training corpus. Rieger then used this correlation coefficient to estimate the relationship between a target word whose meaning he wanted to discover and any other relevant word. For the purpose of this study, the relevant words are the nouns and verbs found by our triplet extraction process. The equation we use to compute each word pair's correlation coefficient is given by:

$$\alpha(w_i, w_j) = \frac{\sum_{t=1}^{T}(w_{it} - e_{it})(w_{jt} - e_{jt})}{\left(\sum_{t=1}^{T}(w_{it} - e_{it})^2 \sum_{t=1}^{T}(w_{jt} - e_{jt})^2\right)^{\frac{1}{2}}}, \tag{5-2}$$

where $\alpha(w_i, w_j)$ is the correlation coefficient of the pair composed of words w_i and w_j; where T denotes the total number of domains forming the training corpus; and w_{it} and w_{jt} denote the

total number of occurrences of w_i and w_j in domain d_t, respectively. The expected number of occurrences e_{it} is defined as $e_{it} = \frac{H_i}{L} l_t$, where H_i is the total number of occurrences of the word w_i in the training corpus; l_t is the length of domain d_t, or, said differently, the number of triplets representing domain d_t; and L is the length of the training corpus. Mathematically, word pairs that have similar occurrences in the training corpus, in the sense that they are both present in or absent from the same texts, have a positive correlation coefficient. These are called *affined* pairs. On the other hand, word pairs with different occurrences, in the sense that one word often appears in texts without the other, will have a negative correlation coefficient. Those noun-verb pairs are called *repugnant*.

One way to understand the behaviour of the correlation coefficients would be to look at the difference between the frequency of a word and its expected frequency, or $w_{it} - e_{it}$. The result of this subtraction, which we will call the *frequency deviation*, or simply the *deviation*, can fall into three broad ranges of values which we will define hereafter. The first range is $w_{it} - e_{it} \approx 0$, when w_{it} is roughly equal to e_{it}. This first range of deviation values can only be observed if the word in question appears almost evenly throughout all domains of the corpus, regardless of the actual topic of each specific domain. This means that this particular word is definitely not domain-specific, but is most likely a word of general meaning. It is also possible, however, that such a word could come from a triplet extraction error. Such error words can only occur rarely and are confined to only one domain. Consequently, w_{it} tends to become very small, e_{it} tends to converge towards zero, and as a result the value of the deviation tends to be very low. The second range of deviation values that we define is $w_{it} - e_{it} > 0$, which occurs when a word appears in a domain more often than expected. For this to happen, the word cannot be distributed evenly throughout the training corpus, but must appear more often in some domains than in others. It must therefore be a domain-specific word. Moreover, the domain d_t is one where the frequency of occurrence of w_i is higher than average. Consequently, the word w_t must be relevant to that domain. We call such words *in-domain*. The third and final range of deviation values that we define is $w_{it} - e_{it} < 0$. This range is characteristic of a situation that is at the opposite extreme of the

previous one. In this case, w_i is domain-specific, but its frequency of occurrence in d_t is less than average. Under these conditions, it is quite evident that the word w_i does not belong in this specific domain. We call such words *out-of-domain*.

The next step in the analysis is to examine the range of values that the correlation coefficients can take as a function of the three ranges of frequency deviation values defined above. To be sure, these three ranges of deviation values could be combined into four unique pairs of words. The first pair we consider is composed of two general words. Referring to Equation (5-2) and to our earlier discussion, we can see that in this case, the correlation coefficient will be close to zero. Indeed, in this case, each of the two deviations in the numerator of the equation will have a low value, which means that the numerator itself, and hence the correlation coefficient, will be rather small. The second pair is made up of a general word and a domain-specific word. Based on our previous discussion, we know that $w_{it} - e_{it}$ can take a high positive value when the domain-specific word is in-domain and a high negative value when it is out-of-domain. However, according to Equation (5-3), this deviation will then be multiplied by the low-value of the general word's deviation, resulting in a small correlation coefficient, albeit not as close to zero as in the general-word-pair case discussed earlier. The third pair under consideration is that composed of two domain-specific words belonging to the same domain. Since they will always be either in-domain or out-of-domain for the same domains, their deviations will always yield matching positive or negative numbers, and the multiplication in the numerator of Equation (5-2) will therefore always be positive. This will result in a correlation coefficient that is always greater than zero. The fourth and final pair of words being considered is the one composed of two domain-specific words belonging to two different domains. This pair is the opposite case of the previous one. Since the words belong to different domains, they will be either in-domain or out-of-domain for different domains. It follows that if one deviation is positive, the other will be negative, and the multiplication in the numerator of Equation (5-2) will therefore always give a negative value. The correlation coefficient obtained from this calculation will likewise always be less than zero.

Taken together, the features of the frequency deviation mentioned above have important implications for the discussion. It is thus of importance to realize that if we were to plot the pairs of words encountered in the training corpus according to their frequency deviation in each domain, we would find that they are divided in regions in a manner similar to the model presented in Figure 5-3. In this theoretical plot, region A contains general pairs, which have two low deviation values and a correlation coefficient around zero. Region C contains pairs composed of one general word and one domain-specific word, and whose correlation will be a little higher than those of region A. Finally, region B contains pairs of domain-specific words, which have large positive or negative deviation values. The pairs in regions BI and BIII are those for which the words belong to the same domain, and are both either in-domain in the case of BI or out-of-domain in the case of BIII. Those pairs have a high positive correlation. The pairs in regions BII and BIV, on the other hand, are composed of words belonging to different domains, and have a high negative correlation.

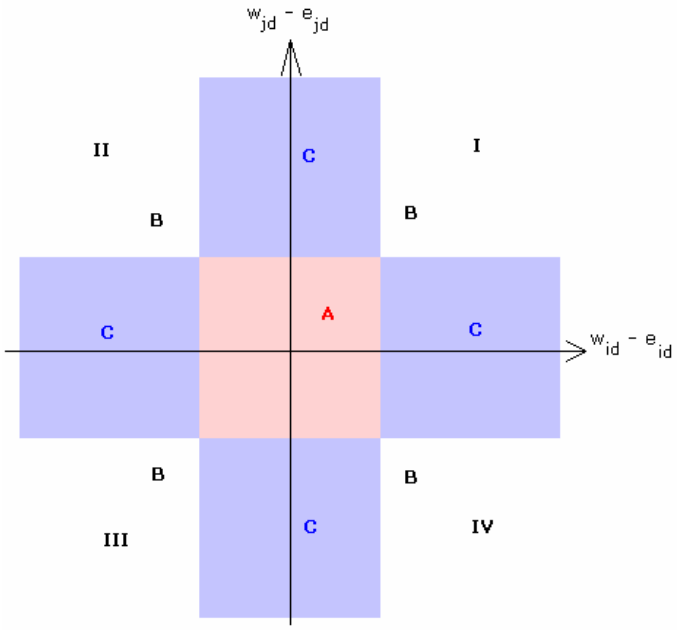


Figure 5-3: The main regions of the values of the correlation.

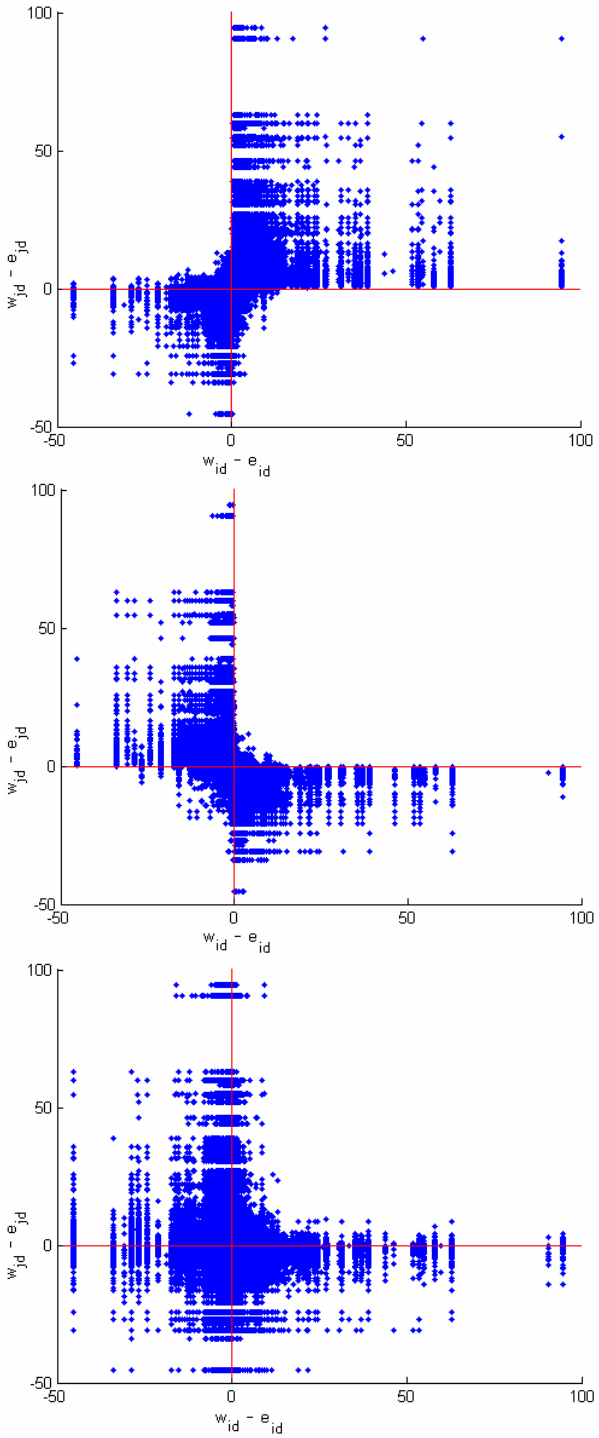
Our predictions can be compared with results gathered from the experimental data of Chapter 6. For clarity, we subdivided the word pairs of our training corpus into three groups, namely: those with a correlation greater than +0.5, those with a correlation less than –0.5,

and those with a correlation that lies between –0.5 and +0.5. We then proceeded to generate a plot similar in nature to that of Figure 5-3 for each of these three subgroups. The resulting three plots, which are presented in Figure 5-4, seem to confirm our theoretical development. Indeed, the figure shows that word pairs with a correlation greater than +0.5 are concentrated in quadrants I and III, while those with a correlation lesser than –0.5 are mostly found in quadrants II and IV, and most of those with a correlation between –0.5 and +0.5 are found in regions A and C. However, while a majority of the points follow our predicted distribution, the graphics do not show the clearly-defined regions A, B and C that we had hoped to find. The reason for this can be traced back to the data sparseness problem highlighted in the previous section. Indeed, we have shown that while the frequency statistics follow the Zipf-Mandelbrot distribution we had anticipated, they are rather sparse and imprecise. As a result, the correlation coefficients we computed here using those statistics cannot be entirely accurate.

5.2.4 Semantic Distance

Once the usage regularities are known, the second level of Rieger's model uses them to extract the meaning points. Rieger defined these meaning points as the difference between the usage regularity of two lexical items. The smaller this difference, the more similar the usage of the two lexical items is, and therefore the closer their meaning must be. This distance can be measured using an Euclidian metric, that will compute the difference of usage regularities of a lexical item against all other lexical items. Although we could compute the distance between any pair of words, in our method we are solely interested in the difference between noun-verb pairs. Thus, we have defined the semantic distance between noun n_i and verb v_j using the following equation:

$$\delta(n_i, v_j) = \left(\sum_{k=1}^{N_w} \left(\alpha(n_i, w_k) - \alpha(v_j, w_k) \right)^2 \right)^{\frac{1}{2}}, \tag{5-3}$$



(a)

(b)

(c)

Figure 5-4: The frequency deviation plot of 94000 pairs of words with a correlation greater than +0.5 (a), 67000 pairs of words with a correlation less than –0.5 (b), and 551000 pairs of words with a correlation between –0.5 and +0.5 (c).

where N_w is the number of different words in the training corpus. The semantic distance $\delta(n_i,v_j)$ can take values between 0 and $2\sqrt{N_w}$, with the most similar noun-verb pairs having the lowest values.

For convenience, we will shorten the expression $\delta(n_i,v_j)$ to δ_{ij} in the rest of this document. We will use the convention that the first subscript is the noun of the pair, and the second subscript is the verb.

The notion of semantic distance represents mathematically the difference in meaning between two words. Indeed, as can be understood from the mathematical development in this section, two words w_i and w_j will have a low semantic distance if their meanings are similar, but a high semantic distance if their meanings are different. But moreover, we can use semantic distances to represent the meaning of two words relative to a third one. For example, if two pairs of words w_i-w_j and w_i-w_k have very different semantic distances, then it appears that w_j and w_k have very different meanings, with one having a meaning more similar to that of w_i and the other having a meaning rather different from that of w_i. On the other hand, if the two pairs have a similar semantic distance, then it appears that w_j and w_k have a similar meaning relative to that of w_i. That understanding of semantic distances is the one we will be using when we build our possibility distributions in Section 5.4.

In light of our earlier analysis of the correlation coefficients, we can foresee four main ranges of values for the semantic distance, corresponding to the four main regions of the correlation plot of Figure 5-3. The first corresponds to the distance between two domain-specific words that belong to the same domain, or regions BI and BIII. As explained before, the correlation between a domain-specific word and another word could be either positive or negative, depending on whether or not the latter word belongs to the same domain as the former. And since we are computing Equation (5-3) using two words w_i and w_j that belong to the same domain, their correlations will always be either positive or negative simultaneously and their values should therefore cancel out when they are subtracted. It follows that the semantic distance in this first scenario will be close to zero. The opposite situation arises

when we compute the distance between two domain-specific words originating from different domains. In this second case, which corresponds to regions BII and BIV, it is logical to presume that a third word belonging to the same domain as one of these two must necessarily come from a domain different from that of the other one. This means that one of the correlation values in Equation (5-3) will be positive while the other will be negative and, instead of cancelling out as before, they will always add up and yield a rather large semantic distance, up to a maximum of $2\sqrt{\text{Number of words}}$ [68], [69], [70]. The third range of semantic distance values we can foresee is the one covering the distance between two general words, and corresponds to region A in Figure 5-3. In that third case, each of the correlation coefficients in Equation (5-3) will be either between a pair of general words, or between a general and a domain-specific word. Given that both of these pairs have small correlation coefficients, the semantic distance will consequently be small. The fourth and final range of values occurs when we compute the semantic distance between a general word and a domain-specific word, and was region C in our correlation analysis. While the correlation of the pairs that make use of the general word will remain small, the pairs that make use of the domain-specific word will take on positive or negative correlation values of different magnitudes depending on the other word in the pair. These correlation values will partially, though not completely, cancel each other out through the subtraction operation in Equation (5-3), thus resulting in a medium semantic distance figure – larger than the distance between two general words, but smaller than that between two domain-specific words of different domains. The resulting correlation coefficient-semantic distance graph, with the four regions positioned as per our analysis, is presented in Figure 5-5.

The experimental data of Chapter 6 can now be used to demonstrate graphically the validity of the preceding theoretical observations, by illustrating the relationship between the correlation coefficients and the semantic distances of pairs of words. To carry out this verification, recall that two domain-specific words belonging to the same domain have a large positive correlation and a semantic distance near zero. On the other hand, pairs composed of two general words or of a general word and a domain-specific word both have small correlations, and either a small or a medium semantic distance, respectively. Finally,

pairs of domain-specific words originating from different domains have a large negative correlation and a large semantic distance. By plotting the relationship between the correlation coefficients computed in Section 5.2.3 and the semantic distance computed for the pairs of words, we obtain the graphic shown in Figure 5-6. Given the imprecision of the correlation coefficients computed in Section 5.2.3, this graphic is naturally noisy. However, it can readily be seen that the general shape of the relationship traced in this figure confirms the validity of our theoretical analysis. As predicted in Figure 5-5, the distribution goes from the negative correlation coefficient-high semantic distance range to the positive correlation coefficient-low semantic distance range, and is wider in the central region near the zero correlation coefficient.

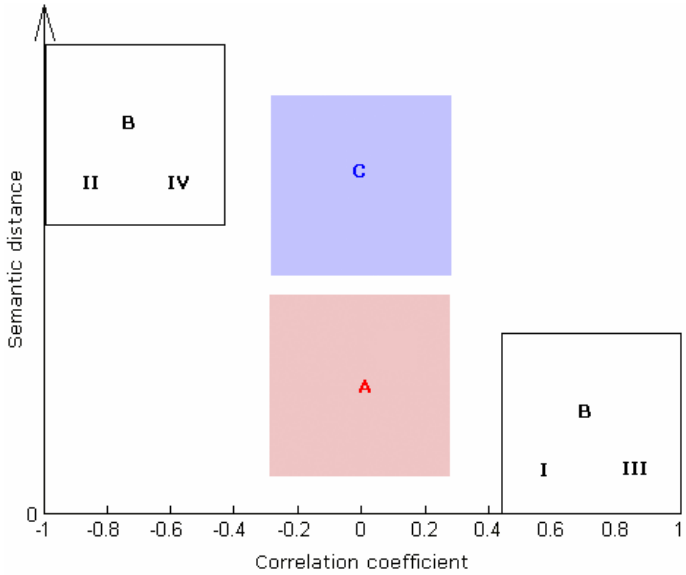


Figure 5-5: The main regions of the values of the semantic distance.

5.2.5 Conditional Probability

Apart from the two above concepts that we have adapted from Rieger's work, our method still requires a third statistical piece of information. It consists of the conditional probability of a domain given each noun-verb. Indeed, while the semantic similarity gives us a statistical insight into the combined meaning of a noun-verb pair, the domain-pair probability, on the other hand, provides us with a statistical relationship between the noun-verb pair and the domains of the training corpus.

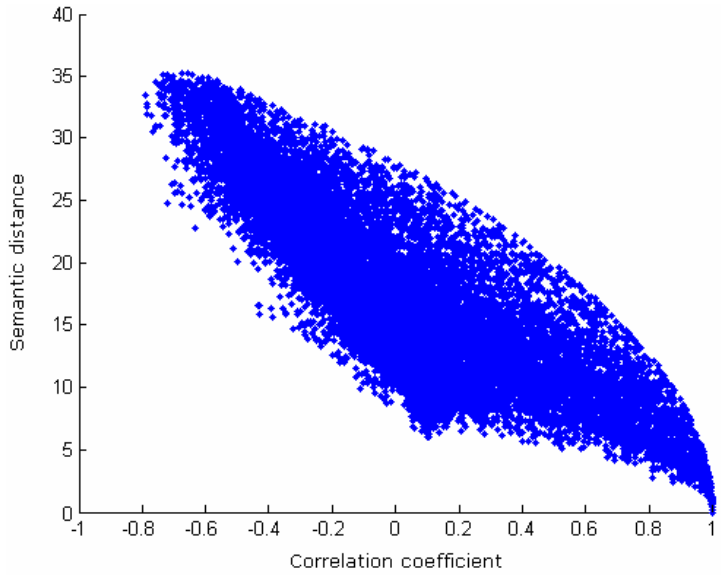


Figure 5-6: The relationship between the correlation and semantic distance of pairs of words.

The probability of domain d_t given the pair composed of noun n_i and verb v_j, $P(d_t \mid n_i v_j)$, can be computed following Bayes' theorem as:

$$P(d_t \mid n_i v_j) = \frac{P(n_i v_j \mid d_t) P(d_t)}{P(n_i v_j)}, \tag{5-4}$$

where $P(d_t)$ is the prior probability of domain d_t, $P(n_i v_j)$ is the normalizing constant, and $P(n_i v_j \mid d_t)$ is the likelihood function. In our system, we assume that the probability of a domain is not function of its length, but that each of the T domains of the training corpus is equally probable, hence:

$$P(d_t) = \frac{1}{T}. \tag{5-5}$$

The normalizing constant is the probability of the pair $n_i v_j$ in the entire training corpus. This will be computed as the sum of the probabilities of n_i and v_j in each of the domains, as follows:

$$P(n_i v_j) = \sum_{t=1}^{T} \frac{n_{it}}{l_t} \frac{v_{jt}}{l_t} P(d_t), \tag{5-6}$$

where n_{it} and v_{jt} are the number of occurrences of n_i and v_j in domain d_t, respectively, and l_t is the length of domain d_t. Finally, the likelihood is the probability of the pair $n_i v_j$ occurring in

domain d_t, which is computed using the probabilities of n_i and v_j, in the same manner as in Equation (5-6):

$$P(n_i v_j \mid d_t) = \frac{n_{it}}{l_t} \frac{v_{jt}}{l_t}. \tag{5-7}$$

For sure, we considered using probability of the pair $n_i v_j$ in each domain to compute Equations (5-6) and (5-7), instead of taking the probabilities of the noun and verb individually. The modified equations would be:

$$\begin{aligned} P(n_i v_j \mid d_t) &= \frac{n_i v_{jt}}{l_t}, \\ P(n_i v_j) &= \sum_{t=1}^{T} \frac{n_i v_{jt}}{l_t} P(d_t), \end{aligned} \tag{5-8}$$

where $n_i v_{jt}$ is the number of occurrences of the pair $n_i v_j$ in domain d_t. In theory, such a measure would be more accurate than the one we used in Equations (5-6) and (5-7). However, in practice, the data sparseness problem will cause most pairs to be absent from most domains, thus resulting in a probability of zero. Using the probability of n_i and v_j individually instead of the probability of the pair $n_i v_j$ allows us to avoid this problem.

In light of the mathematical development presented above, Equation (5-4) can be simplified to:

$$P(d_t \mid w_i w_j) = \frac{\frac{w_{it}}{l_t} \frac{w_{jt}}{l_t}}{\sum_{k=1}^{T} \frac{w_{ik}}{l_k} \frac{w_{jk}}{l_k}}. \tag{5-9}$$

In words, Equation (5-12) can be written as:

$$P(d_t \mid w_i w_j) = \frac{\text{Percentage of domain } d_t \text{ made up by pair } w_i w_j}{\text{Sum of (percentage of each domain made up by pair } w_i w_j \text{)}}. \tag{5-10}$$

Like any other probability, the value of $P(d_t \mid w_i w_j)$ will vary between 0 and 1. From Equations (5-9) and (5-13), we can infer that it will be closer to 1 when a pair of words $w_i w_j$ represents a significant portion of a domain d_t yet an insignificant portion of other domains.

Such pairs are clearly composed of domain-specific words, and the more domain-specific each of the two words of the pair is, the closer to 1 the probability gets. However, the probability will only reach 1 if one of the words in the pair occurs exclusively in that domain. Although such a word could be an exclusive domain-specific word, it could also be the single occurrence of a triplet extraction error. On the other hand, Equations (5-9) and (5-10) also show that the probability will fall close to 0 in the opposite situation, namely when the pair $w_i w_j$ represents an insignificant portion of a domain d_t but a significant portion of the other domains. In this case, although the pair is strongly domain-specific, it belongs to another domain besides d_t. We can further infer that the closer to 0 the probability gets, the more domain-specific and out-of-domain each of the two words is. However, if the probability is exactly 0, the only inference that can be made is that one of the two words never occurs in domain d_t, and of course nothing can be inferred about the other word. Finally, it is worth mentioning that the conditional probability of a domain given a pair of general, non-domain-specific words, will lie somewhere between 0 and 1, with a perfectly-evenly-distributed word pair yielding a probability equal to 1 / number of domains.

5.2.6 Possibility Theory

Zadeh has laid down the mathematical foundations of possibility theory in [110]. In this section, we will outline some key concepts of that theory that are needed for our work. More details about the theory are found in [110].

Let F be a fuzzy subset of a universe of discourse U, and X be a variable that can take values in U. A proposition such as "X is F" can be expressed as:

$$X \text{ is } F \rightarrow R(X) = F, \tag{5-11}$$

where $R(X)$ is a fuzzy restriction on the variable X. We can then associate a possibility distribution Π_X to X, which we postulate to be equal to $R(X)$, i.e.:

$$\Pi_X = R(X), \tag{5-12}$$

with the corresponding possibility distribution function denoted as π_X. In this context, $\pi_X(u)$ is the possibility that $X = u$. By combining Equations (5-11) and (5-12), we can write:

$$X \text{ is } F \rightarrow \Pi_X = F \,. \tag{5-13}$$

Moving one step further, if we assign a linguistic probability value λ to the statement "X is F", we can write:

$$X \text{ is } F \text{ is } \lambda \rightarrow \Pi_{P(X \text{ is } F)} = \lambda \,, \tag{5-14}$$

where $P(X \text{ is } F)$ is the probability distribution of F. This rule follows directly from the assumption that the propositions "X is F is λ" and "$P(X \text{ is } F) = \lambda$" are semantically equivalent:

$$X \text{ is } F \text{ is } \lambda \leftrightarrow P(X \text{ is } F) = \lambda \,. \tag{5-15}$$

This assumption, made by Zadeh in [110], is one of the basic elements of our work. However, despite our reliance on the pioneering work of Zadeh in this field, it is worth pointing out that our application of possibility theory is quite different from his own. As mentioned in Section 2.2.2, Zadeh uses possibility theory to devise mathematical tools, such as PRUF, to be used in translating natural language information into fuzzy sets. For our part, we will show in the following section that we use possibility theory as an end in itself to represent natural language information.

5.3 Fuzzy Sets

The next step of the method is to combine the semantic distances and the probabilities into a single unified measure of the relationship between the word pairs and the domains. The first direct treatment of this question we proposed at an early stage of this study was accomplished using fuzzy set theory [45]. The main focus of this line of reasoning is to generate fuzzy sets that represent the membership of the pairs in each domain, and to provide the tool to defuzzify the membership degree of a specific pair in a specific domain.

As we've mentioned in Section 1.3, the idea of using fuzzy sets in this way was proposed in the previous work done on this project in [85]. Naturally, the first stage of our research consisted in completing and refining the approach into the final form presented in this section. Over the course of our research however, we decided to forego the use of fuzzy set theory in favour of a leap forward into an original mathematical development based on

possibility theory that is more appropriate for our study and opens the way to significant practical applications. In order to follow these two lines of analysis, we shall first deal briefly with fuzzy sets in this section, and then at greater length with the application of possibility theory in Sections 5.5 to 5.8.

5.3.1 Fuzzification

Our line of analysis in applying fuzzy set theory relies on building fuzzy membership functions that represent and combine the information contained within the semantic distances and the probabilities. To this end, the system generates a 2D graph for each noun-domain combination, where the X-axis of the graph is the noun-verb similarity and the Y-axis is the noun-verb probability in each domain. The system can then position each verb in every graph with its appropriate coordinates, and is ready to compute an equivalent fuzzy membership function to represent this data.

Studies have shown that the shape and tuning of the membership functions play an important role in the behaviour of the fuzzy controller [53]. Regarding the shape, we have opted for a trapezoid membership function, which was the one advocated in the previous work [85]. The tuning of the membership function's parameters was done based on each graph's semantic distance and conditional probability values, as we will show in the following paragraphs.

The top plateau of the trapezoid function is centered on the average semantic distance of the noun-verb pairs, $\overline{\delta}_i$, which is computed as follows:

$$\overline{\delta_i} = \frac{1}{N_v} \sum_{j=1}^{N_v} \delta_{ij} \ , \tag{5-16}$$

where N_v is the number of different verbs in the training corpus. The plateau extends one standard deviation of the semantic distance, σ_{δ_i}, on each side, as indicated in the following equation:

$$\sigma_{\delta_i} = \left(\frac{1}{N_v} \sum_{j=1}^{N_v} \left(\delta_{ij} - \overline{\delta_i} \right)^2 \right)^{\frac{1}{2}}. \tag{5-17}$$

The height of the plateau will be the normalized average probability ($AP(n_i, d_t)$) of all noun-verb pairs in that region. It is necessary to normalize that measure, because the high number of noun-verb pairs whose probability is zero would otherwise drag the height of all the membership functions near zero. These noun-verb pairs come from two sources: domain-dependent verbs of one domain that do not appear in other domains, and rare verbs that occur once in the training corpus, thus generating a probability of 1 in one domain and 0 in all others. Obviously, the latter source will skew the results, but it must be dealt with in a way that doesn't affect the former source, which contains some very important information. The solution is to give each verb v_j a weight depending on its frequency in the training corpus. This verb weight VW_j is computed in two steps. In the first step we compute the relative frequency RF_j of the verb, by dividing its frequency in the corpus by the total frequency of all verbs:

$$RF_j = \frac{\sum_{t=1}^{T} v_{jt}}{\sum_{k=1}^{N_v} \sum_{t=1}^{T} v_{kt}}. \tag{5-18}$$

Since there are several thousand verbs in total, that relative frequency will be quite close to zero. To avoid this result, the second step is to normalize the relative frequency by dividing it by the most frequent verb's relative frequency:

$$VW_j = \frac{RF_j}{\max(RF_k)}. \tag{5-19}$$

Thus, the most frequent verbs will have a verb weight close to 1, while verbs that seldom appear in the corpus will have a verb weight around 0. We are now ready to compute the normalized average probability. We define this measure as the sum of the probabilities of the noun-verb pairs in the region of the plateau, divided by the total weight of those verbs. This is given mathematically in Equation (5-20):

$$AP(n_i, d_t) = \frac{\sum_{j=1, j \in V_{Sim}}^{N_{Sim}} P(n_i v_j d_t)}{\sum_{j=1, j \in V_{Sim}}^{N_{Sim}} VW_j}, \tag{5-20}$$

where:

$$V_{Sim} = \left\{ v_j \mid \left| \delta_{ij} - \overline{\delta_i} \right| \leq \sigma_{\delta_i} \right\}, \tag{5-21}$$

and:

$$P(n_i v_j d_t) = \frac{n_i v_{jt}}{\sum_{t=1}^{T} n_i v_{jt}}. \tag{5-22}$$

The ramp sections of the trapezoid extend to three standard deviations on each side of the mean. In Equation (5-20), N_{Sim} represents the number of elements contained within V_{Sim}. A graphical representation of this setup is shown in Figure 5-7.

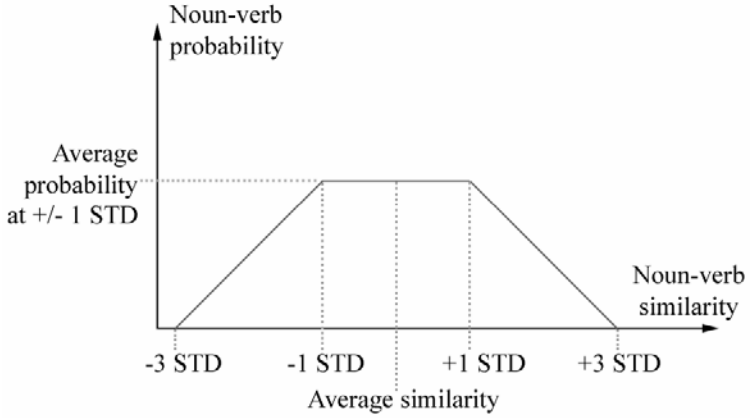


Figure 5-7: Graphical representation of a membership function.

5.3.2 Defuzzification

Once the set of membership functions has been fully trained, it can be used to compute the membership of unseen triplets. The membership of a triplet composed of n_i as subject, v_j as verb and n_k as object in domain d_t is computed by defuzzifying the value of both pairs, $n_i v_j$ and $n_k v_j$, from the corresponding membership functions. Numerous defuzzification techniques have been proposed and tested in the literature [73], and though the centroid

method is generally considered more effective than the other methods, it is not by any means the best method to use in all situations. Indeed, different situations call for different defuzzification methods. More specifically, the centroid method is not appropriate for our work. The problem with the centroid method is that it minimises the importance of the pair-domain probabilities (the Y-axis of the trapezoid plots) and focuses on the noun-verb similarities (the X-axis of the trapezoid plots). However, the similarity of a noun-verb pair is the same in all domains, and its height is what distinguishes one domain from another. Using the centroids eliminates almost completely this distinction, leaving very little information to differentiate one domain from another. Since our method requires a defuzzification method that will not suffer from such information loss, we have decided to defuzzify the membership functions by computing their area, using a process that we outline here and present graphically in Figure 5-8.

For each of the two noun-verb pairs in domain d_t, the algorithm begins by performing the α-cut of the noun's membership function at the height corresponding to the noun-verb similarity abscissa coordinate (Figure 5-8a). The cropped membership functions of the subject-verb pair and object-verb pair of the triplet are then merged together using an AND function (Figure 5-8b), and the height of the resulting triplet membership function is normalized (Figure 5-8c). In our early work, the normalisation was done by multiplying the membership degrees by the relative importance of the verb in the domain, or the number of occurrences of the verb over the number of verbs in the domain, and dividing them by the relative importance of the domain in the corpus, or l_t / L. Finally, the triplet's membership function is defuzzified by computing its area in order to obtain the crisp membership value of the triplet (Figure 5-8d).

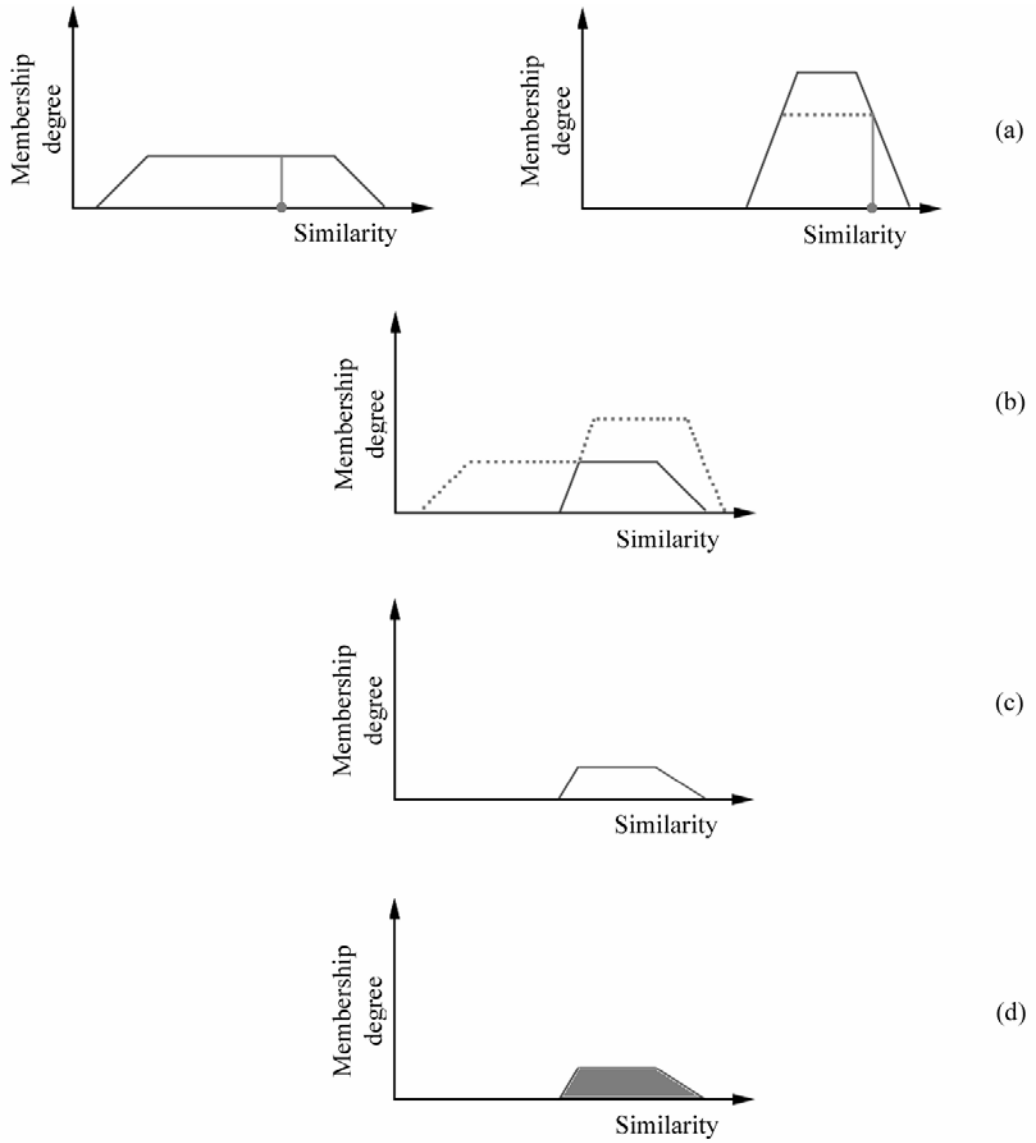


Figure 5-8: Illustration of the defuzzification technique.

5.3.3 Implementation and Experimental Results

The method described in this section was used to implement a general fuzzy text classifier. The classifier is designed to recognise specific text domains that are present in a training corpus. The training corpus is simply composed of un-annotated texts and of their matching domain. Once trained, the system will accurately classify unseen texts in one of the domains it was trained for, and will also give a measure of the certainty of its classification.

The training data used in this experiment comes from three domains of the Brown Corpus [32], and examples of actual 2D graphs and their membership functions are shown in Figure 5-9, with the noun category *biochem* acting as subject in all three domains the classifier is trained for. The 20 documents used to test the classifier were taken from various sources on the Internet. A complete presentation of this training and testing data will be done in Chapter 6. For now, we will limit ourselves to an overview of the results, to demonstrate the usefulness of our fuzzy set-based representation technique.

It is interesting to note that the membership function in Figure 5-9 reaches the highest membership degree and covers the greatest area for the second domain. As we have already mentioned, the noun category used for this example is *biochem*, a category comprised of various bio-chemical nouns, that extends from the names of molecules to the names of body parts. It is no surprise then, that noun-verb pairs making use of nouns from this category have a higher membership degree in the medical domain than in the other two.

Next, we classified the 20 test documents in the three domains the system was trained for. Each document is classified in the domain in which it exhibits the highest membership. This membership is computed by taking the sum of the membership of each triplet of the document in each domain. For comparison purposes, we computed the membership of the triplets first by using the defuzzification technique proposed in Section 5.3.2, then by using the classic centroid defuzzyfication method, and finally by simply calculating the probability $P(n_i v_j d_t)$. Table 5-1 presents the classification results, along with the average membership (or probability) of the correctly- and incorrectly-classified documents in each case.

For the reasons explained in Section 5.3.2, the centroid defuzzification method performs, as expected, very poorly for our setup. Meanwhile, the results obtained using our defuzzification method and those obtained by using the probabilities directly show some significant differences. The average probability of the documents classified correctly using the probabilities is significantly higher than the average membership obtained by using the defuzzification, but so is the that of the documents classified incorrectly. Furthermore, the

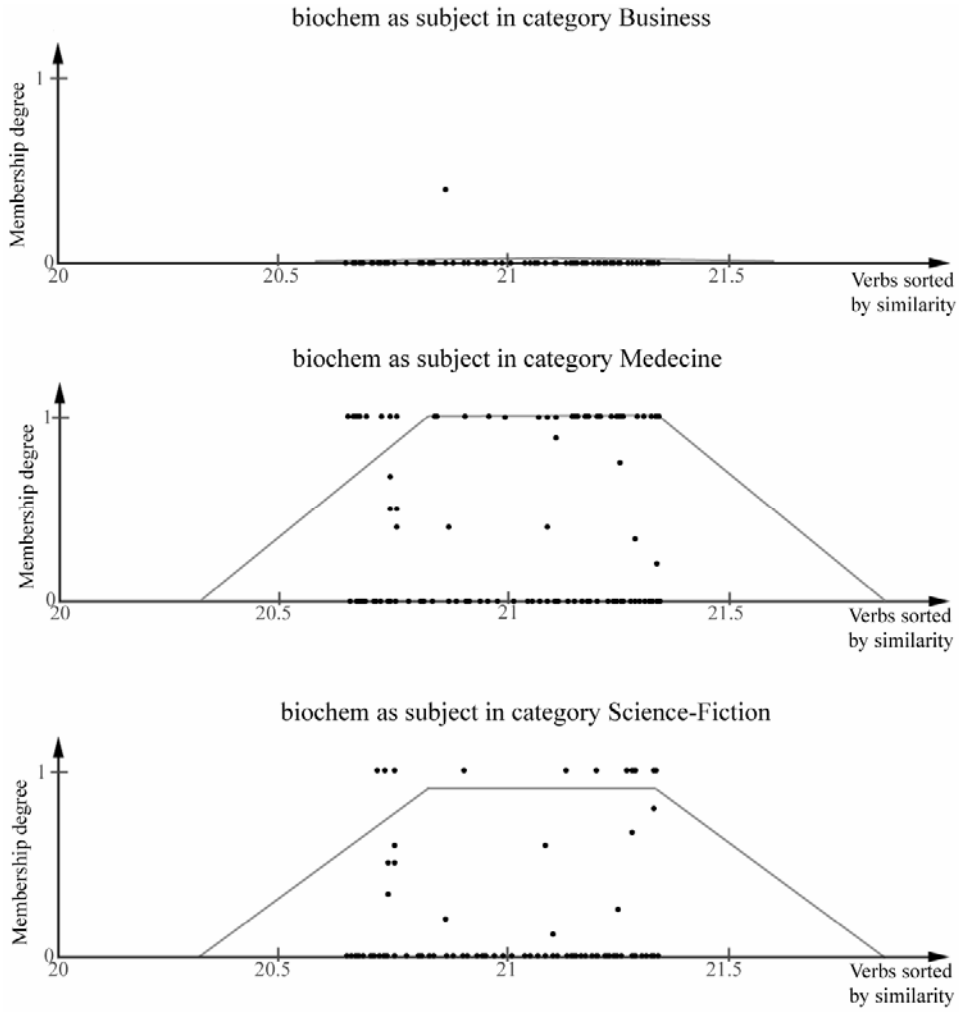


Figure 5-9: 2D graphs and membership functions for the noun category *biochem* as subject in the domain *business* (top), *medicine* (middle) and *science-fiction* (bottom). Each dot represents a verb.

standard deviation of the documents classified incorrectly is much larger when the probabilities are used. These two findings negate the gains of having a higher correct probability, and indicates that it is preferable to use membership functions than the probabilities directly. This is due to the fact the probabilities can vary wildly, as shown by the very high standard deviation of the incorrectly-classified documents. This problem is commonly encountered in statistical NLP, and is due to the sparseness of the training data. Indeed, some rare noun-verb pairs occur only once or twice in the training data, which generates 100% spikes of probabilities in one domain, while a large number of rare noun-

verb pairs will not be encountered at all, which generates a large number of 0% probability points in all domains. These 100% and 0% probabilities are obviously false, since no noun-verb combination in English is used only in one domain, or is never used at all. They only occur because examples of a specific combination are lacking in our training corpus. Instinctively, the solution to this problem seems to consist in increasing the size of the training corpus in order to include a representative sample of the rare pairs. However, researchers in statistical NLP have shown that all training corpora have this problem, regardless of the size. This observation has led to the development of a number of algorithms to smooth the co-occurrence probabilities, such as Laplace's law [48], Lidstone's law of successions [51], the Expected Likelihood Estimation [12], the held out estimator [43], and the famous Good-Turing estimation [35]. These algorithms operate by subtracting a portion of the probability mass of pairs encountered in the training corpus, and distributing it to absent pairs that would otherwise have had a zero-probability. In our system, the construction of the fuzzy membership functions serves as a smoothing algorithm for the probabilities. It distributes the probability mass to the pairs depending on their similarity, decreasing the 100% spikes and increasing the zero-probability pairs. This smoothing significantly improves the results of our method, as our discussion of the results of Table 5-1 indicated.

Table 5-1: Classification results.

Experiment	Documents correctly classified	Average membership and deviation of correct classifications	incorrect classifications
Section 5.3.2	15 / 20	0.42 ± 0.23	0.11 ± 0.09
Centroid	5 / 20	0.04 ± 0.04	0.19 ± 0.13
Probability	16 / 20	0.84 ± 0.23	0.36 ± 0.46

5.4 Possibility Distributions

5.4.1 Theoretical Development

In the system we have developed, we use possibility distributions in order to represent the possibility of a domain d_t given a noun-verb pair $n_i v_j$. This is implemented by using an individual possibility distribution for each domain-noun combination $\Pi_{d_t n_i}$, as illustrated in

Figure 5-10. Within this framework, when given the pair $n_i v_j$, a user can look up the possibility distribution of each domain d_t given n_i, and obtain the value of the distribution at v_j. This permits him to know the possibility of the domain d_t given $n_i v_j$.

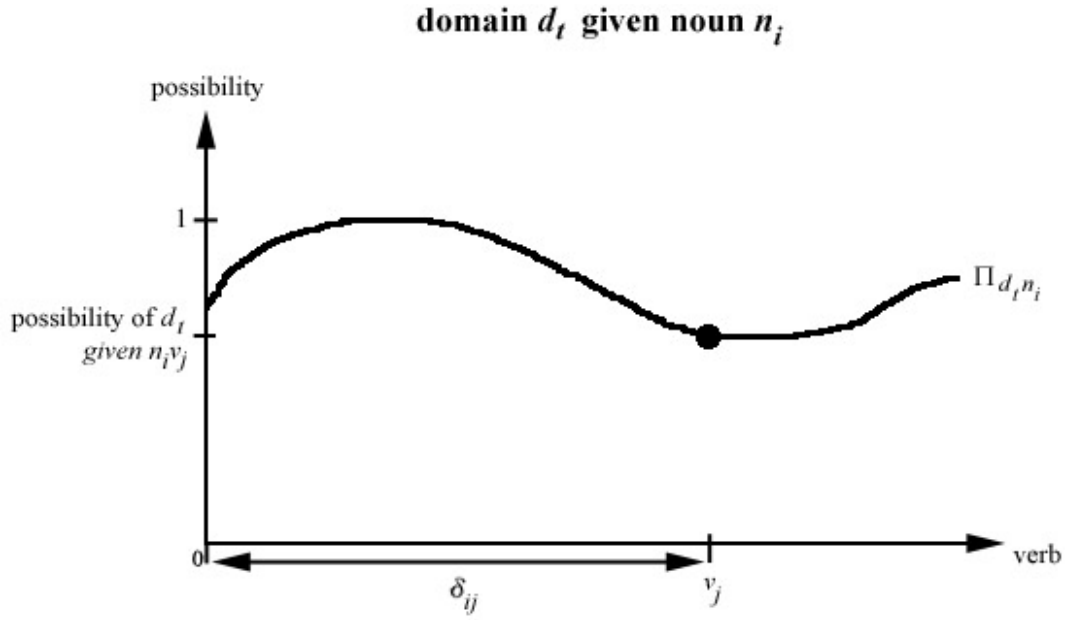


Figure 5-10: Plot of the possibility distribution for domain d_t given noun n_i, and the possibility of d_t given the pair $n_i v_j$.

Zadeh [110] showed that a probability distribution and a possibility distribution could be equivalent. Furthermore, it has been shown in Section 5.2 that each domain is related to each noun-verb pair by a discrete probability value. By using these results, we can adapt Zadeh's work to our situation as follows:

$P(d_t \mid n_i v_j) = p$	*From Section 5.2.*
"The probability of d_t given $n_i v_j$ is p"	*English equivalent of previous line.*
"The probability that the observed pair $n_i v_j$ belongs to d_t is p"	*Rewriting of previous line.*
$P(n_i v_j$ is $d_t) = p$	*Mathematical equivalent of previous line.*
$n_i v_j$ is d_t is p	*From Equation (5-19).*

$\pi_{P(n_iv_j \text{ is } d_t)} = p$ *From Equation (5-18). Because p is a discrete probability instead of a linguistic probability λ, it corresponds to a single possibility π instead of a possibility distribution* Π.

$\pi_{n_iv_j}(d_t) = p$ *Equivalent rewriting of previous line.* (5-23)

This development shows in what way the probability of d_t given n_iv_j is equivalent to $\pi_{n_iv_j}(d_t)$, the possibility of d_t given n_iv_j. However, transforming a set of these possibilities into a possibility distribution is not as straightforward as it seems. Indeed, these possibilities are based on a relatively small training corpus, and because of data sparseness they are incomplete and inaccurate. Some form of data smoothing is therefore required, in order to create an accurate possibility distribution.

An example can help illustrate this situation. Suppose that our system observes two noun-verb pairs, namely "company buy" (as in, "a company bought stocks") and "company purchase" (as in, "a company purchased stocks"). Since the meaning of these two pairs is nearly identical, it is expected that a domain will have the same possibility given the knowledge of either one. However, if the first pair is often observed in one domain while the second pair is not, then this domain will only have a high possibility given the first pair. Such an erroneous result must be corrected.

The correction we propose is based on the fact that two pairs n_iv_j and n_iv_k with a similar meaning will be at a very close, or possibly at the same, semantic distance. If they have the same semantic distance, we will consider that the two pairs are synonymous and give them a single possibility value. Following [26], this is done by taking the maximum of the two values:

$$\pi_{n_iv_j \cup n_iv_k}(d_t) = \max\left(\pi_{n_iv_j}(d_t), \pi_{n_iv_k}(d_t)\right), \quad (5\text{-}24)$$

where $\pi_{n_iv_j \cup n_iv_k}(dt)$ is the possibility of d_t given either one of the pairs. This rule is in accordance with our intuition: if the possibility that an event E_1 occurs is π_1, then the possibility that either events E_1 or E_2 occur cannot be less than π_1.

The same principle can be applied to correct the possibility associated with two pairs n_iv_j and n_iv_k with different semantic distances δ_{ij} and δ_{ik}, respectively. However, we must allow for the fact that the pairs have different meanings, by dampening the possibility of one of the pairs in function of the difference between δ_{ij} and δ_{ik}. This is accomplished by multiplying one of the possibilities by the relative difference between the pairs. In other words, the possibility of d_t given the pair n_iv_j will be perceived at the semantic distance δ_{ik} as $\pi^k_{n_iv_j}(d_t)$:

$$\pi^k_{n_iv_j}(d_t) = \pi_{n_iv_j}(d_t) * \left(1 - \frac{\left|\delta_{ij} - \delta_{ik}\right|}{(\delta_M - \delta_m) * \frac{1}{G}} \right), \tag{5-25}$$

where δ_M and δ_m are the maximum and minimum possible semantic distances between a noun-verb pair, respectively, and G is the neighbourhood size. The role of this last parameter is to further dampen the possibility of pairs at a large distance to δ_{ij}, up to the point where past a certain limit, pairs will have no influence whatsoever in the update function of Equation (5-24). In other words, the role of the parameter G is to limit the update to a neighbourhood around the semantic distance of pair n_iv_j. In so doing, it allows us to manually control the trade-off between having the pairs at each semantic distance be completely independent of all others, and having pairs at any semantic distance influence all others. When $G = 1$, which is its minimum value, then all pairs are part of a single neighbourhood, and a pair at δ_m can affect the update of another pair at δ_M. However, higher values of G create smaller neighbourhoods. For instance, if $G = 2$, a neighbourhood can cover at most half the range of semantic distances, and a pair at δ_m can only affect the update of pairs up to $\delta_M/2$. To get a better grasp of this relationship, Figure 5-11 gives a graphical illustration of the impact of three different values of G on the perceived possibility of a pair at δ_m. One possible direction for future work could be to estimate the optimal value of this parameter.

Next, we can update the possibility of d_t given n_iv_k by modifying Equation (5-24) as follows:

$$\pi'_{n_iv_k}(d_t) = \max\left(\pi^k_{n_iv_j}(d_t), \pi_{n_iv_k}(d_t)\right). \tag{5-26}$$

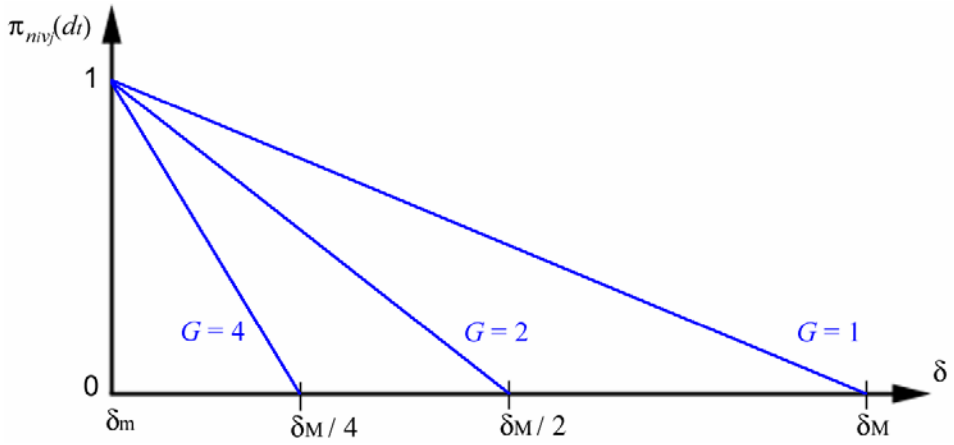


Figure 5-11: Impact of three neighbourhood values.

Equation (5-26) updates $\pi_{n_i v_k}(d_t)$ by taking into account the possibility of one other pair. For more accuracy and completeness however, all other pairs composed of n_i and a verb should also be incorporated. To this end, Equation (5-26) must be modified in the following way:

$$\pi'_{n_i v_k}(d_t) = \max_{j \in N_v} \left(\pi^k_{n_i v_j}(d_t) \right), \tag{5-27}$$

where N_v is the number of different verbs in the training corpus. Finally, the possibility distribution $\Pi_{d_t n_i}$ of a combination of d_t and n_i is the set of possibilities of d_t given n_i and each of the verbs. Equation (5-28) computes this possibility distribution as:

$$\Pi_{d_t n_i} = \left\{ \pi'_{n_i v_k}(d_t), k \in N_v \right\}. \tag{5-28}$$

5.4.2 Validity of the Method

It would be interesting to subject the possibility distributions to the same validity analysis as we conducted for the correlation coefficients, semantic distances and conditional probabilities in Section 5.2. More specifically, since the possibility distributions are constructed from the conditional probabilities and the semantic distances, it is therefore of importance to get a deeper understanding of the relationship between these two notions. Since the distribution of a specific domain is computed under the assumption of a specific noun and any verb, we can limit our study to three main situations, namely: that where the

noun is domain-specific and is in-domain, that where the noun is domain-specific but is out-of-domain, and that where the noun is general and not domain-specific.

The first situation to be considered is that where we compute the possibility distribution of a domain given an in-domain domain-specific noun. This situation is depicted in Figure 5-12(a). In this graph as well as all the others of Figure 5-12, the verbs are sorted on the X axis according to the semantic distance of the noun-verb pair, while the conditional probability is measured on the Y axis. As shown in Section 5.2.4, the noun-verb pairs with the lowest distance will be the ones that belong to the same domain, and since the noun is in-domain, it follows that the pair is both strongly domain-specific and belongs to the current domain as well. Under these conditions, and as shown in Section 5.2.5, the conditional probability of the domain given the pair will have a high value. The next highest value of semantic distance will be that of pairs in which the noun is matched with a general verb. As argued in Section 5.2.5, the probability of a domain given a domain-specific word and a general word will be lower than that of two domain-specific words, mentioned before. Finally, pairs where the noun is matched to a domain-specific verb from a different domain will yield the highest semantic distance, and the lowest probability. These relationships trace the decreasing line shown in Figure 5-12(a).

The second situation under analysis is the inverse of the preceding one. In this case, we compute the possibility distribution of a domain given an out-of-domain domain-specific noun. Once again, the noun-verb pairs with the lowest semantic distance will be those that belong to the same domain, followed by those pairs with general verbs, and finally by the pairs that belong to different domains. However, in this instance, the lowest-distance noun-verb pairs belong to a domain different from the one being plotted, and their conditional probability will be close to zero. In the same vein, the second group, namely that of noun-verb pairs that include a general verb, will exhibit a slightly higher conditional probability than that of the first group. Finally, in the case where the noun and verb belong to different domains, it is possible that the verb will belong to the domain being plotted. If this is the case, the conditional probabilities given these pairs will reach the highest values. This relationship between the semantic distance of the three groups of pairs and their condition

probability traces a positive curve as depicted in Figure 5-12(b). It should be mentioned that this graph is valid only if the verbs with the greatest distance from the noun are the most in-domain ones. However, if there are several domains, then other less distant verbs may be more in-domain than these. In that case, the probability will peak for some moderate value of the semantic distance, and will fall again as the verbs become increasingly distant from that peak, indicating that they belong to more and more dissimilar domains. This setup is illustrated in Figure 5-12(c).

The final situation is the one that uses a general, non-domain-specific noun. This case is notably different from the previous two. Indeed, as shown in Section 5.2.4, only domain-specific noun-verb pairs can have very small or very large semantic distances, whereas pairs that include a general word will show small to medium distance values. Moreover, the conditional probability of a domain given general noun-verb pairs is rather small, while that of a domain given mixed general and domain-specific pairs will only be a little higher when the domain-specific word belongs to the domain being plotted, and lower otherwise. The resulting distribution will therefore simply consist of a central hump, much like the one illustrated in Figure 5-12(d).

The relationships described above can be observed in the experimental data of Chapter 6. We have selected from these data two examples to illustrate these relationships. These two examples have been chosen on account of their clarity and comprehensiveness.

The first example is composed of the probability values of two domains given a noun category of scientific terms and all verbs. The plot of the probabilities of the sciences domain (in blue) and that of the social studies domain (in red) are shown in Figure 5-13(a). These two plots can be superposed as in this figure, since the semantic distances composing the X-axis are not domain-dependent. This example illustrates the first two theoretical cases discussed earlier. As shown in Figure 5-13(a), the probability of the sciences domain given pairs composed of a science noun and a similar verb is very high at first, but falls gradually as the distance between the verbs and the noun increases. By contrast, in the social studies domain, the probability given pairs of science-related words starts very low, but increases as

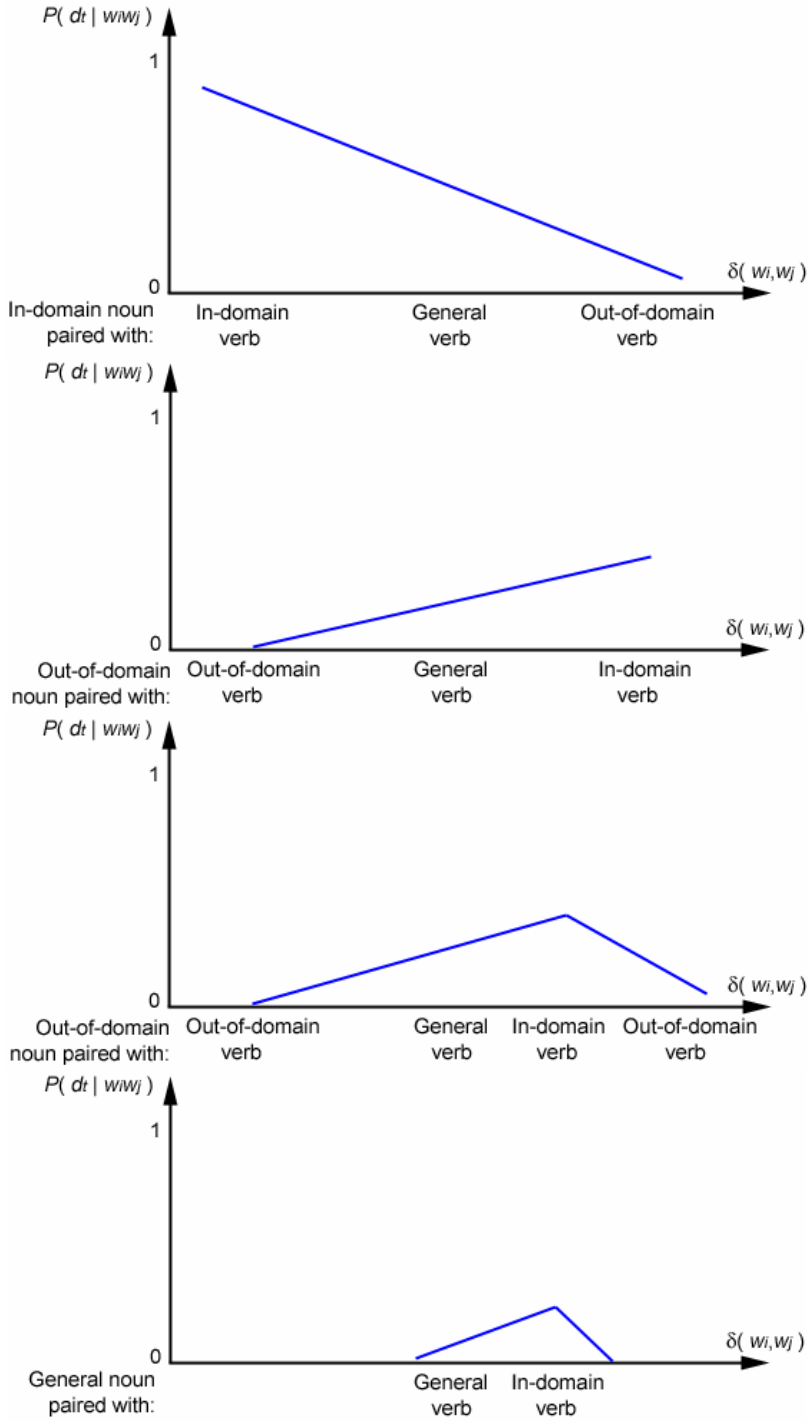


Figure 5-12: The four possible shapes of the probability-distance relationship. Illustrated is the conditional probability of a domain given a noun of that domain (a), given a noun of a different domain (b and c), and given a general noun (d).

the verbs become more and more distant from the science noun category. It is worth noting that, since the science noun category selected only occurs in those two domains, the probabilities in both domains are exactly complementary. In this regard, the second example selected is a lot more sophisticated. In this example, we selected an art-related noun category that occurs in no less than six different domains. For ease of presentation, the three most interesting domains of this example are illustrated in Figure 5-13(b). They are the art domain in green, the language art domain in red, and the social studies domain in blue. This graph also shows that, with respect to the conditional probability of the art domain, the art-related noun-verb pairs in the first half of the semantic distance axis dominate those of the other two domains and indeed those of the remaining three domains not depicted graphically here. And even though they are overtaken by the conditional probabilities of other domains in the second half of the distance axis, the art domain still has a strong showing in that region. The social studies domain, on the other hand, is unrelated to the noun category, and its probability is therefore high only for pairs with the highest distances, namely those pairs where the verb is the most unrelated to the art noun. To be sure, these two situations are similar to the ones examined in the previous example. However, the novelty in this example comes from the language art domain. This particular domain is somewhat related to the arts noun category, and Figure 5-13(b) shows, its conditional probability distribution peaks near the middle of the plot, when the probability of the art domain begins to fall, and when the distance between the art noun category and the verbs has a medium value. That medium value means that the verbs are not exactly art-related, yet not completely different either. This last case of the language art domain illustrates the situation depicted in Figure 5-12(c).

Although the scatter diagram of the second example presented above follows the theoretical distributions of three graphs of Figure 5-12, it has rather an irregular shape, and far more so than in the simpler first example. For instance, the probability of the art domain can drop from nearly 1 to almost 0 and jump back again to a high value over three consecutive noun-verb pairs. Since the conditional probability distributions depicted in Figure 5-13 are used as background to generate the possibility distributions, the need to

smooth them becomes evident. The resulting possibility distributions which correspond to the smoothed version of Figure 5-13 are presented in Figure 5-14.

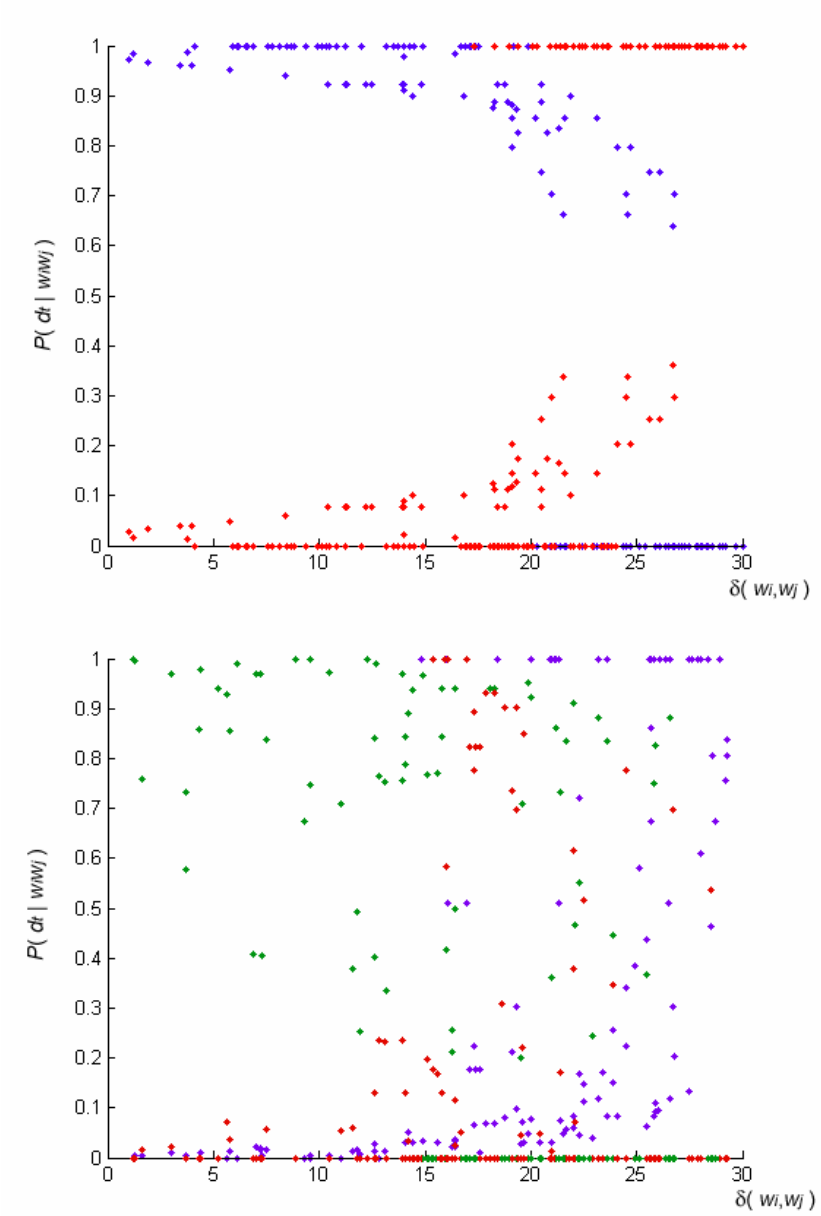
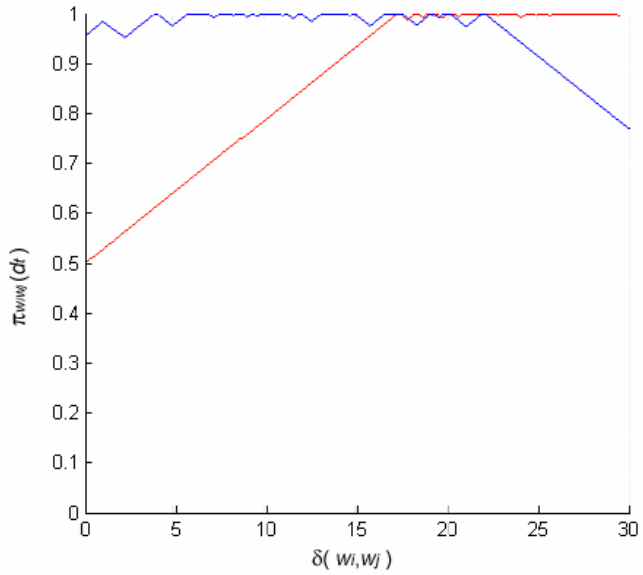
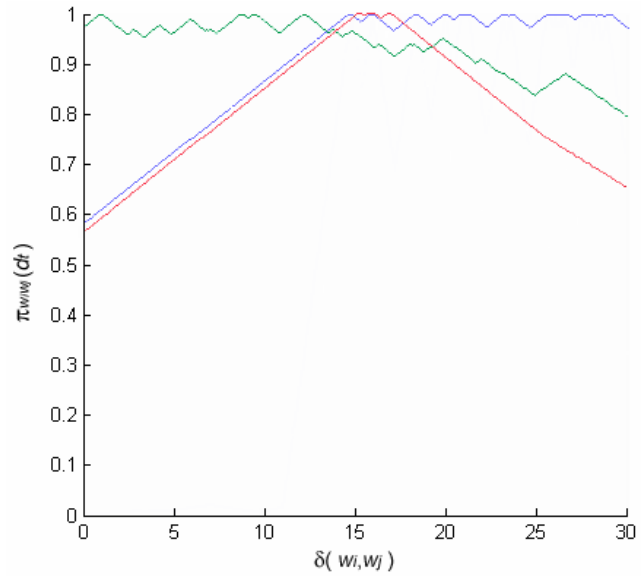


Figure 5-13: Experimental results of the relationship conditional probability and semantic distance. Shown in (a) is the science domain (blue) and social studies domain (red) given a science noun category. Shown in (b) is the art domain (green), the language art domain (red) and the social studies domain (blue) given an art noun category.

As we can see, the possibility distributions are more uniform than the probability scatter diagrams of Figure 5-13. Yet they still appear somewhat jagged – another consequence of the problem of data sparseness that has plagued this research since the beginning.



(a)



(b)

Figure 5-14: The smoothed possibility distributions of the graphics of Figure 5-13.

5.5 Illustrative Example

We will now present an example to illustrate the method described so far in this chapter. The example will show the steps required to plot a possibility distribution. We have chosen the distribution of the domain "science-fiction" given the noun "art" used as object. The data come from the first training corpus we will use in the implementation of our domain classifier, which will be presented in its entirety in Chapter 6.

The first step in the development of the method is to compute the semantic distance between the object "art" and every verb in the training corpus, using the process presented in Section 5.2.4. It should be noted that this first step is not specific to the "science-fiction" domain we are dealing with. Indeed, the semantic distance of a noun-verb pair is constant in all domains, and this step is executed only once for the entire training corpus. Given the number of different words in the training corpus, the semantic distance can take values between 0 and 55.7, although in practice it only varies between 0.1 and 44.1. It would be both impractical and uninteresting to reproduce here the complete list of semantic distances between the object "art" and each of the 684 verbs. Instead, a limited sample is shown in Table 5-2.

Table 5-2: Semantic distance between the object "art" and 25 verbs.

Verb	Distance	Verb	Distance	Verb	Distance
Work	0.1	Announce	14.3	Consider	37.6
Enjoy	4.5	Examine	18.3	Estimate	38.9
Look	4.5	Limit	21.3	Provide	40.4
Judge	4.9	Sell	23.6	Contain	42.6
Record	4.9	Buy	25.8	Identify	42.8
Carry	7.1	Design	25.8	Obtain	42.8
Write	7.1	Compare	31.7	Act	43.0
Publish	8.2	Play	34.6	Show	43.5
				Improve	44.1

The second step in the method is to compute the probability of the domain "science-fiction" given each of the 684 art-verb pairs, by applying the set of equations presented in

Section 5.2.5. Once this is done, each pair will have both a semantic distance and a probability value associated with it. These can be represented with a 2D graph, as in Figure 5-15. In this figure, black points depict single noun-verb pairs, red points represent more than one but less than 100 noun-verb pairs superposed, green points represent between 100 and 200 noun-verb pairs superposed, and blue points represent more than 200 noun-verb pairs superposed.

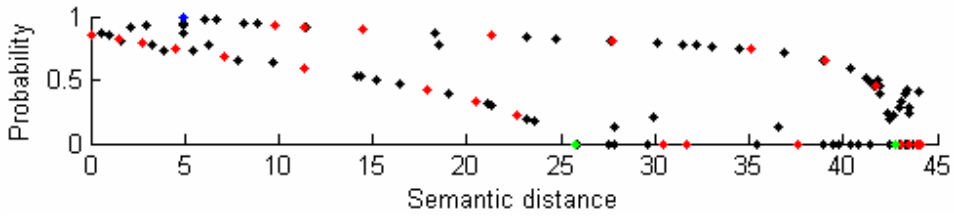


Figure 5-15: Semantic distance-probability plot of the 684 art-verb pairs.

The final step of the method is to extrapolate the possibility distribution from the probability distribution. First, art-verb pairs at the same semantic distance are given the same possibility, according to Equation (5-24). This yields the possibility plot illustrated in Figure 5-16. The possibility distribution is then smoothed, using Equation (5-25) and $G = 1$. The resulting possibility distribution is shown in Figure 5-17.

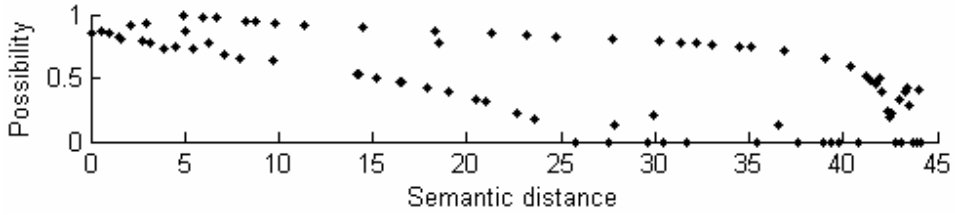


Figure 5-16: Semantic distance-possibility plot of the 684 art-verb pairs.

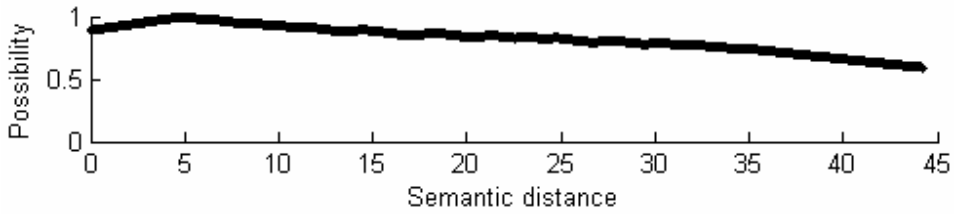


Figure 5-17: Smoothed possibility distribution of the domain "science-fiction" given the pairs composed of the object "art" and a verb.

5.6 The Possibility Given Triplets

Up till now, the method developed has only focused on noun-verb pairs, yet actions are represented using subject-verb-object triplets. The system must therefore bridge the gap between the pairs and the triplets.

As it turns out, given our earlier assumption of independence in Section 5.1, it is a simple matter to infer the possibility of a domain d_t given a subject-verb-object triplet $n_i v_j n_k$, once the system knows the possibility of the domain given $n_i v_j$ and $n_k v_j$. Since we assume that the meaning of each pair is independent, the system needs only to combine both possibility values. Following [26], this is done by taking the minimum of the two values $\pi_{n_i v_j}(d_t)$ and $\pi_{n_k v_j}(d_t)$:

$$\pi_{n_i v_j n_k}(d_t) = \pi_{n_i v_j \cap n_k v_j}(d_t) = \min\left(\pi_{n_i v_j}(d_t), \pi_{n_k v_j}(d_t)\right) \tag{5-29}$$

where $\pi_{n_i v_j}(d_t)$ and $\pi_{n_k v_j}(d_t)$ are the possibilities of d_t given the pairs $n_i v_j$ and $n_k v_j$ respectively, $\pi_{n_i v_j \cap n_k v_j}(d_t)$ is the possibility of d_t given both pairs and $\pi_{n_i v_j n_k}(d_t)$ is the possibility of d_t given the triplet $n_i v_j n_k$. This rule is in accordance with our intuition: if the possibility that an event E_1 occurs is π_1, then the possibility that both events E_1 and E_2 occur cannot be more than π_1.

Suppose for example that the system needs to know the possibility of the business domain given the triplet "company buy stock", and that it already knows the possibility of the business domain given the pairs "company buy" and "buy stock". Since the business domain will be very possible given both these pairs, the minimum possibility value, and thus the possibility given the triplet, will be very high. Now suppose that, instead of "buy stock", the second part of the triplet was something irrelevant, like "buy fruit". This triplet could come from a sentence giving background trivia on the company, such as "the company bought a variety of fruits for the employee picnic last year", or it could even come from an extraction error. Either way, the business domain will not be very possible given that pair. Hence, Equation (5-29) will insure that the business domain will not be very possible given the

triplet “company buy fruit”, despite the fact that it would be possible given the first pair “company buy”.

5.7 Complexity of the Method

To complete the presentation, it would be interesting to evaluate the complexity of an algorithm designed along the lines of the method developed in this chapter. To begin with, let us assume that the algorithm receives as input a training corpus of L triplets divided into T domains. In the worst case scenario, every single verb and noun in the triplets will be unique. There will thus be L verbs and $2L$ nouns, for a total of $3L$ words. Since each domain is composed of many triplets, it follows that the total number of domains T will always be a lot smaller than the number of triplets. For simplicity, we will follow the assumption of [13], to the effect that basic operations such as sums, subtractions, multiplications, divisions and square roots are done in unit time and are therefore constant.

As explained earlier, the method of this chapter is made up of four successive steps. The complexity of an algorithm implementing this method will therefore amount to that of the most computationally expensive step.

The first step of the method is to compute the correlation coefficient between each pair of words. The correlation equation, presented in Equation (5-2), is composed of two parts: a numerator, in which basic operations are repeated T times and which therefore has a complexity in $O(T)$, and a denominator, in which basic operations are repeated T^2 times and which is therefore in $O(T^2)$. Consequently, the overall complexity of the equation is in $O(T^2)$. Moreover, as was detailed in Section 5.2.3, the correlation is computed for each pair of words, without regard to the order of words and with no repetitions. Hence, the number of times that Equation (5-2) will be computed is equal to the number of combinations of two elements in $3L$. As can be seen from Equation (5-30) below, these combinations result in the first step of the method having a quadratic complexity in function of L, which, as was pointed out earlier, will be much greater than $O(T^2)$. The entire step thus has a complexity of $O(T^2L^2)$.

$$C_{3L}^{2} = \frac{3L!}{2!(3L-2)!} = \frac{3L(3L-1)(3L-2)!}{2(3L-2)!} = \frac{3L(3L-1)}{2} = \frac{9L^2-3L}{2} = O(L^2) \qquad (5\text{-}30)$$

The second step of the method is the computation of the semantic distances which is also discussed in Section 5.2.4. The semantic distance equation, or Equation (5-3), is made up of basic operations which are repeated for the number of different words, or $3L$ times in our current situation. The complexity of this equation will therefore be in $O(L)$. Furthermore, as explained in Section 5.2.4, the semantic distance is computed for each pair of one noun and one verb. Given the initial setting presented above, there will be $2L^2$ such pairs, and this step will consequently have a cubic complexity.

In the third step, the conditional probability of each domain given each noun-verb pair is computed using Equation (5-4). The development done in Section 5.2.5 reveals that this equation is simple enough to be executed in $O(T)$ time. However, a potential algorithm will have to compute Equation (5-4) for each domain and each noun-verb pair, for a total of $T2L^2$ times. This step of the algorithm has therefore a complexity in $O(T^2L^2)$.

The fourth and final step of the method is the smoothing of the possibility distributions, presented in Section 5.4. In this step, the algorithm has to compute for each domain given each noun the possibility distribution as per Equation (5-28). More specifically, Equation (5-28) computes the possibility value at each semantic distance in the distribution as the maximum of the set of dampened possibilities at all semantic distances, as defined in Equation (5-27). The max operator has a linear complexity given the number of elements it must consider, which is equal to the number of semantic distances. Finally, the dampening equation given in Equation (5-25) is composed entirely of basic operations and can be computed in constant time for the possibility value at each semantic distance. The overall complexity of computing the set of dampened possibility values will therefore be a linear function of the number of semantic distances. Assuming the worst case scenario in which each noun-verb pair has a unique semantic distance and given that a distribution is limited to a single noun, the number of semantic distances in the distribution will be the same as the number of verbs. Consequently, the complexity of this step will be computed as:

number of domains * *number of nouns* * *number of semantic distances* *

$$(dampening + max) = T*2L*L*(L + L) = O(TL^3). \quad (5\text{-}31)$$

In sum, an algorithm implementing our method will be made up of a sequence of four steps. Given L, which is normally much greater than T, the number of triplets provided as input at the training stage, two steps of the algorithm will have quadratic complexity while the other two will have cubic complexity. The overall algorithm will therefore have cubic complexity.

The silver lining is that, once the training is done, a testing algorithm can perform in linear time. Indeed, such an algorithm will only need to compute the possibility of each domain given each triplet of the testing corpus. Under normal circumstances the testing corpus should be much smaller than the training corpus, however, for simplicity we will assume that both corpora have the same total of L triplets. Following Equation (5-29), it appears that the possibility of each triplet is the minimum of its two composing pairs. The min operator, like the max operator, is of linear complexity given the number of elements it considers, which is a constant in this case. It thus simplifies to a complexity of $O(2)$. Likewise, looking up the possibility of a domain given a pair in our distributions is a trivial operation that can be done in unit time. The testing algorithm will perform these two operations for each domain and each triplet, for a total of TL times. Its complexity is therefore simply $O(TL)$.

5.8 Validity of the Method

Before concluding this chapter, it behoves us to briefly summarize its main findings with respect to the validity of our method. To fix ideas, we will use the domain-specific words w_1 and w_2, which originate from domains d_1 and d_2, respectively, along with the general word w_G, and an extraction error, w_E, which occurs a few times in d_1.

Let us start by focusing on the frequency of each word in each domain. Those results are fairly straightforward: the domain-specific words will occur with high frequency in their own domains but with low frequency elsewhere, while w_G will occur with high frequency everywhere and w_E will, by definition, occur rarely in one domain and never occur

elsewhere. These results support and extend the predictions of the Zipf-Mandelbrot law, as explained in Section 5.2.1. Let us turn next to the frequency deviation, as defined in Section 5.2.3. As we recall, domain-specific words will have a deviation well above zero in their own domain and well below zero in other domains, while general words and error words will have a deviation around zero. These results are presented in Table 5-3. As can be observed from that table, there is a clear distinction between the behaviour of domain-specific words, that of general words and that of error words. Indeed, even at this early stage of computation, it is possible to distinguish between these three different types of words and to filter out undesirable words, such as those resulting from triplet extraction errors.

Table 5-3: Frequency and deviation results for each word-domain pair.

Word in domain	Frequency	Deviation
w_1 in d_1	*High*	> 0
w_1 in d_2	*Low*	< 0
w_2 in d_1	*Low*	< 0
w_2 in d_2	*High*	> 0
w_G in d_1	*High*	≈ 0
w_G in d_2	*High*	≈ 0
w_E in d_1	*Low*	≈ 0
w_E in d_2	*= 0*	≈ 0

Also in the context of Section 5.2.3, the next step in our method called for the use of the frequency deviations of words to compute the correlation coefficients between pairs of words. Drawing on Equation (5-2), we showed that the correlation between domain-specific words belonging to the same domain will have a high positive value, while that of domain-specific words belonging to two different domains will have a high negative value. On the other hand, pairs composed of at least one general word will have a correlation close to zero, however if the second word were domain-specific, the pairs will have a slightly higher correlation value than if they were composed of two general words. Once calculated, these correlation coefficients are then used in Equation (5-3) to compute the semantic distances. Section 5.2.4 focuses on this aspect of the method, and reveals that pairs of domain-specific

words belonging to the same domain will have a semantic distance close to zero, while those that belong to different domains will have the highest distances. In-between these extremes are pairs that include a general word. In parallel with the results of the correlation analysis, pairs with a domain-specific word will have a slightly higher semantic distance than pairs of general words. Finally, in Section 5.2.5, the conditional probability of each domain given the pairs is computed. Most notably, this section has established that the probability of a domain given a pair of in-domain domain-specific words will be close to 1, while its probability given domain-specific words of another domain will be close to 0. The probability given other pairs will be in-between these two extremes and will vary depending on the presence of general or domain-specific words, with the probability of a domain given a pair of perfectly evenly-spread general words being equal to 1 / (number of domains in the training corpus). The results of the correlation, semantic distance and conditional probability are summarized in Table 5-4. The notation in this table adds the word w_3, which is another domain-specific word that originates from domain d_1, much like w_1.

Table 5-4: Correlation and distance of word pairs, and probability of domains given pairs.

Domain given pair	Correlation	Distance	Probability
d_1 given (w_1, w_3)	> 0	≈ 0	≈ 1
d_2 given (w_1, w_3)	> 0	≈ 0	≈ 0
d_1 given (w_1, w_2)	< 0	$\approx 2\sqrt{\text{Number of words}}$	*Medium*
d_1 given (w_1, w_G)	≈ 0 (*Medium*)	*Medium*	*High*
d_2 given (w_1, w_G)	≈ 0 (*Medium*)	*Medium*	*Low*
d_1 given (w_G, w_G)	≈ 0 (*Low*)	*Low*	≈ 1 / Number of domains

The special cases where the pairs of words include the extraction error w_E are presented separately in Table 5-5. This table includes in its bottom line the word w_F, which is another extraction error like w_E, but one that occurred in another domain. By comparing Table 5-4 and Table 5-5, it becomes apparent that pairs that include an error word have a mixed behaviour: they have the correlation and semantic distance of general pairs, but yield the conditional probability of extremely domain-specific pairs. This is a consequence of their rareness. Their low frequency and deviation, which we observed in Table 5-3, lead to a low

correlation and distance similar to that of general pairs. Furthermore, this rareness means that they appear in only one domain, which makes them akin to very domain-specific words for the purpose of the probability computation.

Table 5-5: Correlation and distance of word pairs, and probability of domains given pairs, for pairs that include an error word.

Domain given pair	Correlation	Distance	Probability
d_1 given (w_E, w_1)	≈ 0	*Medium*	$= 1$
d_2 given (w_E, w_1)	≈ 0	*Medium*	$= 0$
d_2 given (w_E, w_2)	≈ 0	*Medium*	$= 0$
d_1 given (w_E, w_G)	≈ 0	*Low*	$= 1$
d_2 given (w_E, w_G)	≈ 0	*Low*	$= 0$
d_1 given (w_E, w_F)	≈ 0	*Low*	$= 0$

To sum up, Tables 5-3 to 5-5 reveal that the semantic information representation method advocated in this research must handle four different types of words, namely domain-specific words whether they are in-domain or out-of-domain, general words, and error words. Furthermore, the method must also handle them either singly or in pairs. It does so by computing the five metrics analysed in this chapter: the frequency, the frequency variation, the correlation, the semantic distance, and the conditional probability.

Given the results presented in Tables 5-3 to 5-5, it is possible to see that no two pairs of words have the same values for all five metrics. It is thus of importance to combine these results in order to gather reliable information about the words' meaning and usage, and the nature of the documents containing them. The method developed in this study, by putting together the information obtained from the five metrics, achieves this goal in a most effective way.

5.9 Conclusions

In this chapter, we have focused on the segment of our system that allows us to represent domain information using possibility distributions. These distributions quantify the

possibility of a domain given a particular triplet, or in other words, how easy it is to use this triplet in that domain.

We began the explanation of our method by introducing the mathematical notion on which the method is founded. We discussed a way to compute the semantic distance between a pair of words and the probability of a domain given a noun-verb pair, and introduced the basics of possibility theory as it pertains to our work. Before computing the possibility distributions, we presented an alternative solution we considered, which relied on fuzzy set theory. We then showed that probabilities and semantic distances previously computed could be used to generate the possibility distribution of a domain given the noun-verb pairs. These successive steps of the method were then illustrated with an example. Next, we explained how to obtain the possibility of a domain given a particular triplet by combining the possibility of that domain given two pairs of words. Finally, we evaluated the complexity of learning and testing algorithms designed along the same lines as our method.

Using this method, the domain information contained in the English triplets can be quantified and represented in a mathematical format that can be easily used in a number of applications. Some examples of applications will be suggested in Chapter 6, and one will be examined in detail.

Chapter 6
Applications and Experimental Results

6.1 Introduction

The previous chapter explored in relative detail the theoretical underpinnings of our knowledge-representation method. Starting with subject-verb-object triplets extracted from ordinary English texts with the help of appropriate techniques, such as those presented in Chapters 3 and 4, our approach can be applied to represent domain information using possibility distributions. Chapter 5 set forth in detail the sequence of stages and equations needed to compute these possibility distributions.

The purpose of this chapter is to demonstrate the practicability of our method by presenting and analysing five suitable applications. We will first create a domain classifier based on our method. In this regard, we will detail the implementation of the method step by step. The classifier will then be trained and tested using several different corpora, and the experimental results will be analysed. Following this first illustration, four more applications will be presented. However, these applications will not be studied in the same depth as in the case of the domain classifier, since their purpose is simply to show that the method is general and flexible enough to be applied to any NLP problem.

6.2 Text Classifier

As mentioned in Section 2.6, the literature on text classification displays a large variety of techniques put forward to address this challenging task. Most of these techniques, however, follow the same fundamental pattern. They begin by representing both the classes and the documents to classify as vectors, and then they match each document to its most similar class. Accordingly, the main innovations offered by these techniques deal with refining the composition of the vectors and improving the comparison techniques.

The text classifier developed to test our method is radically different from previously proposed text classification systems: it does not rely on a document vector at all. Instead of

classifying texts using the frequencies of words or features, this text classifier bases its classification on the actions described in the text. The classifier begins by extracting the subject-verb-object triplets representing the actions from each document. It then retrieves the possibility of each domain given each of the triplets and sums them to find the most possible domain for the text document.

Two methods have been proposed for the task of triplet extraction in this thesis, namely the simpler method of Chapter 3 based on syntactic heuristics, and the more sophisticated method of Chapter 4 using our part-of-speech hierarchy and a rule-learning algorithm. Either one could be used to perform the triplet extraction for the text classifier in this chapter. We noted in Sections 3.5 and 4.5 that the method of Chapter 3 suffers from several limitations that do not appear in the more general method of Chapter 4. However, while the method in Chapter 4 is less limited, that of Chapter 3 was specifically designed and fine-tuned for our work. Consequently, while the method of Chapter 4 will be preferable in most cases, the method of Chapter 3 yields better results in this one specific case, and is therefore the one we will use to extract the triplets of words used in this chapter.

6.3 Experimental Setup

6.3.1 Training and Testing Corpora

In order to fully demonstrate the usefulness of our method, the classifier will be trained and tested thrice, using three different corpora. The rational for running three experiments is to demonstrate the generality of the method, and its usefulness in three different situations. Indeed, in the first experiment, the classifier is trained with a general text corpus divided into three different domains, whereas in the second experiment it is trained using a specialized domain-specific text corpus divided into three similar sub-domains. The third experiment goes a step further, as the classifier is trained and tested using a corpus of learning objects divided into 16 domains. Taken together, these experiments serve to show that the method can be successfully applied in a wide range of situations.

The first training corpus comes from the Brown Corpus [32]. This data source is a corpus of American English written texts compiled in 1961. It is composed of 500 sample documents, selected to reflect the spread of domains Americans were reading about at that time. Documents in the corpus thus cover a wide range of topics, from news coverage to religious texts, from industrial reports to detective fiction. To set off the research on our classifier, we have decided to limit initially our scope to three domains of the Brown Corpus, namely the business domain (samples A26-A28 in the Brown Corpus), the medical domain (samples J13-J17) and the science-fiction domain (samples M01-M06). It should be noted however that the approach presented in this paper can be expanded to cover all the domains indexed in the Brown Corpus. One of the reasons motivating the choice of this corpus is the fact that it is a general corpus that includes samples from a large variety of domains. Once the classifier is expanded to the entire corpus, it will be able to handle texts from most domains.

The second training corpus is composed of Reuters business news articles. These news articles are sorted in three domains: corporate acquisitions, company earnings, and company mergers. It should be mentioned that, since these three domains are subsets of the business domain, a good deal of overlap exists between the actions of the three domains. This makes the classification problem more difficult.

The testing corpus for the first experiment is composed of 20 documents belonging to one or the other of the three domains of the training corpus. These documents are of course not part of the training corpus, but instead they come from various online sources. Furthermore, the 11 business-domain documents of this first testing corpus form the testing corpus of the second experiment. These two testing corpora are presented in detail in Table 6-1.

Table 6-1: Composition of the testing corpora.

Document	Description	Domain	
		Experiment 1	Experiment 2
1	Business news article (Reuters)	Business	Acquisition
2	Business news article (Reuters)	Business	Acquisition
3	Business news article (Reuters)	Business	Acquisition
4	Business news article (CTV)	Business	Earnings
5	Business news article (Reuters)	Business	Earnings
6	Business news article (CP)	Business	Earnings
7	Business news article (Bloomberg)	Business	Earnings
8	Business news article (Times)	Business	Earnings
9	Business news article (CTV)	Business	Earnings
10	Business news article (Bloomberg)	Business	Earnings
11	Business news article (Washington Post)	Business	Mergers
12	F Bánhidy, RB Lowry, AE Czeizel, "Risk and Benefit of Drug Use During Pregnancy", *Int J Med Sci* 2005; 2:100-106 (abstract)	Medicine	
13	SK Das, K Sanyal, A Basu, "Study of urban community survey in India: growing trend of high prevalence of hypertension in a developing country", *Int J Med Sci* 2005; 2:70-78 (abstract)	Medicine	
14	MK Paul, AK Mukhopadhyay, "Tyrosine kinase – Role and significance in Cancer", *Int J Med Sci* 2004; 1:101-115 (abstract)	Medicine	
15	Encyclopaedia entry on antibiotics	Medicine	
16	Encyclopaedia entry on tuberculosis	Medicine	
17	Complete Short Story (Ray Bradbury, "A Sound of Thunder")	Science-Fiction	
18	Book chapter (H. G. Wells, "War of the Worlds")	Science-Fiction	
19	Book chapter (Mark Twain, "The Adventures of Tom Sawyer")	Science-Fiction	
20	Section of an unpublished short story (Mike Combs, "The Right Question")	Science-Fiction	

The data used to create the third training and testing corpora in this research comes from Canada's SchoolNet [75]. SchoolNet is an online digital repository of learning objects (LO),

that serves as a portal to thousands of educational websites. A LO is a metadata file containing information about an entity, either digital or not, that is used for educational purposes [41]. Learning object repositories (LOR) such as SchoolNet are growing in popularity, and are meant to become an integral part of the Semantic Web [9]. However, tools to represent and navigate through a LOR, such as [67], [39], or to integrate together and search through several LORs, such as [20], [21], are still in their infancy and lack a reliable process to represent and handle the information contained within the LO. Consequently, we will focus on Canada's SchoolNet in this research in order to show that our method could be helpful in that regard.

6.3.2 Triplet Extraction and Noun Categories

The necessity to replace individual nouns with noun categories was explained in Section 3.2.3. To gain a deeper understanding of the nature of the noun categories, we created the categories used in the first two experiments manually, while the process designed to generate automatically noun categories proposed in Section 3.2.3 was applied in the third experiment. This dual approach to the generation of noun categories brings into sharp focus the distinction between manually-created and automatically-generated noun categories.

In the first training corpus, each noun is represented by one of 45 categories. Although the system treats all categories in the same manner, we can nevertheless differentiate between two types of categories. Categories of the first type group together nouns that relate to specific domains or concepts, such as *finance*, *justice* or *emotion*. We list those categories in Table 6-2. Categories of the second type are "catch-all" categories, or more general categories designed to give a coarse classification to nouns that do not fit into any category of the first type. These categories are listed in Table 6-3. Although there are fewer categories of the second type than there are of the first type, each second-type category encompasses more nouns.

Table 6-2: 35 specific noun categories used in the first training corpus.

Category	Definition	Example
Art	Nouns related to art forms, art supplies, or artists.	Ballet, Poet
Biochem	Nouns related to anatomy, diseases, medical terms and chemicals.	Chloride, Kidney
Causality	Nouns representing a cause or effect.	Consequence
Clothes	Nouns representing clothes, parts of clothes, or accessories.	Crown, Pocket
Colour	Any noun used to represent the colour of an object.	Blue, Hue
Date	Names of days and months or nouns representing a lapse of time.	Hour, Future
Emotion	Any emotion or state of mind.	Enthusiasm
Family	Nouns representing family members.	Father
Finance	Nouns relating to money, economics and business transactions.	Banker, Tax
Fire	Nouns related to fire.	Flame
Food	Nouns describing food, including production and consumption.	Dairy, Snack
Geography	Any natural landscape feature or astronomical object.	Nebula, Swamp
Geometry	All manners of lines, shapes and geometric measures.	Round, Length
Group	Nouns representing groups or derived from the action of grouping.	Combination, Merge
Information	Nouns relating to information, including processing and transmitting.	News, Reasoning
Justice	Nouns relating to the court of law.	Defendant
Lifeform	Any animal or plant	Lizard
Literary	Any noun related to written text or a part of discourse.	Conjugate, Newspaper
Location	Any man-made place or proper name of a place.	Canada, City
Measure	Nouns describing measures, the act of measuring, ranks or numbers.	Estimate, Fifth
Military	Anything related to the military or warfare, including all	Battle,

	weapons.	Warrior
Movement	Nouns describing all manners of physical movement.	Orbit, Path
Opening	Any natural or man-made orifice.	Fissure
Organisation	Any corporation or business group.	Company
Politics	Anything relating to the government, including buildings and people.	Dictatorship, Voter
Religious	Any noun representing a religious item, ceremony or being, including historical figures.	Angel, Church, Moses
Resource	Nouns representing natural resources that are not biological in origin.	Dust, Iron
Science	Nouns representing a field of science or related to scientific.	Laboratory, Theory
Shopping	Anything relating to retail shopping, including places and incentives.	Ad, Marketplace
Social	Any class or social group.	Generation
Sound	Anything that can be heard.	Tenor
Transport	Any means of transportation.	Locomotive
Undertaking	Nouns representing large-scale endeavours.	Expedition, Voyage
Unit	Any unit of measure.	Kilometre
Weather	Any noun relating to the weather or environmental conditions.	Drought, Storm

Table 6-3: 10 general noun categories used in the first training corpus.

Category	Definition	Example
Abstraction	Abstract notions that do not fit in any other category.	Anything, Premium
Active	Nouns derived of active verbs that do not fit in any other category.	Care, Repair
Building	Nouns representing buildings or parts of buildings and that do not fit in any other category.	Estate, Home, Wall
Individual	Any single person that doesn't fit in any other category, and historical figures.	Friend, Napoleon
Passive	Nouns derived of passive verbs that do not fit in any other category.	Depletion, Proliferation
People	Any group of people that doesn't fit in any other category.	Crowd, Everybody
Physical	Any physical object that doesn't fit in any other category.	Debris, Teleprompter
Profession	Any occupation that doesn't fit in any other category.	Guardian, Watchmaker
Sense	Nouns relating to human senses or mechanical sensors and that do not fit in any other category.	Flavour, Glance, Stealth
Tool	Any man-made object that is used to accomplish a task and that doesn't fit in any other category.	Device, Ladder, Software

The categories presented in Tables 6-2 and 6-3 are those that were found to strike the best balance between generality and specificity given the training data used in this research. They do not by any means represent the only possible clustering of these nouns. On the other hand, the problem of polysemy was solved simply by placing nouns in the category representing their most frequently-used meaning.

The noun categories for the second training corpus are obtained by specializing those of the first training corpus to the business domain. This specialization is accomplished by splitting away nouns from some of the categories to form new, more domain-specific

categories. The specialization needed to adapt the noun categories to the business domain is set forth in Table 6-4.

Table 6-4: 8 specialized noun categories.

Category	Split from	Definition	Example
Business-Action	Active	Nouns derived of active verbs related to the business world.	Capitalization
Customer	Individual	An individual who purchases a good or service.	Participant
Deal	Abstraction	Notions relating to deals and deal-making.	Compromise
Investor	Individual	An individual who puts capital in a business enterprise.	Industrialist
Management	Profession	An administrative-level profession.	Chairman
Money	Finance	The name of currencies.	Yen
Observer	Profession	An analyst or critic type of profession.	Examiner
Worker	Profession	An employee-level profession.	Designer

The subject-verb-object triplets are extracted from the training and testing corpora using the noun categories presented above as well as the process described in Chapter 3 which, as mentioned in Section 4.4, yields slightly better results in these specific corpora than the method of Chapter 4. The total numbers of triplets extracted from each training domain and testing document are presented in Table 6-5. These results help illustrate the range of conditions under which the method must operate. Indeed, the first training corpus is composed of relatively small domains, containing less than 2,000 triplets each, while the second training corpus is composed of much larger domains, with one reaching nearly 10,000 triplets. The testing documents also vary greatly, with the smallest ones being composed of only a few dozen triplets and the largest ones featuring hundreds of triplets.

Table 6-5: Triplet counts of the training domains and testing documents.

Domain or Document	Triplets	Domain or Document	Triplets
Training domain: Business	1071	Testing document 8	90
Training domain: Medicine	1813	Testing document 9	42
Training domain: Sci-Fi	1375	Testing document 10	42
Training domain: Acquisition	9618	Testing document 11	67
Training domain: Earnings	8342	Testing document 12	27
Training domain: Mergers	3174	Testing document 13	79
Testing document 1	109	Testing document 14	34
Testing document 2	103	Testing document 15	233
Testing document 3	52	Testing document 16	381
Testing document 4	25	Testing document 17	288
Testing document 5	57	Testing document 18	141
Testing document 6	26	Testing document 19	241
Testing document 7	54	Testing document 20	60

Canada's SchoolNet is composed of 2371 learning objects sorted in a hierarchy of 150 domains. Each LO contains several metadata fields from which information could be extracted. Since the triplet extraction process described in Chapter 3 is designed to handle natural-language sentences, we have decided to limit our scope to the "description" field of the LO at this stage of the research. This field contains a few sentences written in plain English by ordinary SchoolNet users who wish to review or comment the contents of the website corresponding to each LO. Following the triplet extraction process, a total of 27746 triplets, featuring 3773 different nouns divided into 375 noun categories and 767 different verbs, are gathered from this data. The training corpus is then created at runtime, by selecting randomly 90% of the learning objects in the SchoolNet LOR. The remaining 10% constitute the testing corpus. This selection of the training and testing corpora was initially done completely randomly, but later in our research we decided to constrain the selection process to insure that 10% of each domain is included in the testing corpus and that the same ratio of long to short LO is maintained in both corpora. These two constraints are introduced to

guarantee that the testing corpus matches the training corpus more closely. These two modifications, however, did not produce any noticeable changes in the experimental results.

It is also worth noting that our classifier cannot yet handle a hierarchy of domains. As a result, it only considers the first level of the SchoolNet hierarchy, which divides the learning objects into 17 different domains. We have used 16 of these domains in our study. The 17th domain is ignored since it is composed of only two LO, and does not therefore contain enough data to be trained and tested properly.

It is important to point out that some of the remaining 16 domains cover quite similar topics, and their contents may consequently overlap. This is the case, for example, of the contents of the social studies domain, which overlap with those of the social sciences domain. The same is true for the contents of the business education domain, which overlap with those of the entrepreneurship studies domain. Also of importance is the fact that the triplets are not spread evenly throughout the domains. Indeed, while the larger domains can count thousands of triplets, the smaller ones can be limited to only a few hundreds or even less, depending on the information gain threshold set in the filtering step of Section 3.2.3. These two characteristics of the SchoolNet LOR make the classification task with this corpus a lot harder than with either the Brown Corpus or the Reuters Corpus.

6.3.3 Possibility of Testing Documents

By applying the method explained in Chapter 5 and either one of the training corpora, the classifier can be trained to generate the possibility distribution of each domain for a given triplet. When its training is completed, the classifier is set to classify the testing documents. The classification is done in a winner-takes-all fashion. For each triplet in the document, only the most possible domain is given a value, while the possibilities of the other domains are set at zero. The total possibility of each domain is then computed as the sum of all triplets. The following equations specify this process.

$$\pi_Z(d_t) = \begin{cases} \pi_{n_i v_j n_k}(d_t) & \pi_{n_i v_j n_k}(d_t) = \max_{t \in T}\left(\pi_{n_i v_j n_k}(d_t)\right) \\ 0 & \text{otherwise} \end{cases} \quad \textbf{(6-1)}$$

$$\pi(d_t) = \sum_{z=1}^{Z} \pi_Z(d_t) \quad \textbf{(6-2)}$$

In Equation (6-1), $\pi_z(d_t)$ is the possibility of domain d_t given the triplet τ_z. The triplet τ_z is the z^{th} triplet of the test document, and is composed of $n_i v_j n_k$. In Equation (6-2), the possibility of domain d_t given the entire testing document is defined as $\pi(d_t)$, and is computed as the sum of the possibility of domain d_t given each of the Z triplets in the testing document.

The possibility $\pi(d_t)$ represents how easy it is for the testing document to belong to domain d_t. Once this measure has been computed for every domain, the testing document can be classified in the domain with the highest possibility, π_M, which is formally defined in Equation (6-3) as:

$$\pi_M = \max_{t \in T}\left(\pi(d_t)\right). \quad \textbf{(6-3)}$$

The rationale for using the winner-takes-all approach of Equation (6-1) instead of simply summing all possibility values is that this latter approach would give too much weight to domains that have several low-possibility triplets. Consider for instance the simplified illustration given in Table 6-6. In this example, the classifier is computing the possibility of three domains, given the three triplets of a document. If the classifier sums all the possibilities, it will identify domain d_3 as the most possible domain on account of the large number of low-possibility triplets that it contains, despite the fact that the other domains are more possible when two of these triplets are considered. Conversely, by keeping only the domain with the maximum possibility given each triplet, the classifier finds that domain d_1 is the most possible. In that scenario, the possibility of domain d_3 is greatly reduced since more possible domains win most of the triplets.

Table 6-6: A simplified classification example.

Domain	τ_1	τ_2	τ_3	Sum	Max
d_1	0.0	0.0	0.8	0.8	0.8
d_2	0.4	0.0	0.0	0.4	0.4
d_3	0.3	0.3	0.3	0.9	0.3

The foregoing method of operation also allows the classifier to compute the confidence of its classification. This is an important feature of the classifier, as it will allow it to distinguish between correct and incorrect classifications automatically. Indeed, correct classifications will be those documents classified with a high level of confidence, while erroneous classifications will be at a low level of confidence. Hence, the classifier will be able to detect and correct its classification errors automatically, thereby improving its results. In this system, we define the confidence level as the difference between the domain with highest possibility, π_M, and the domain with the second-highest possibility, π_{M-1}, defined in Equation (6-4).

$$\pi_{M-1} = \max_{t \in T, t \neq M} \left(\pi(d_t) \right). \tag{6-4}$$

Using this definition, the confidence C can be computed as in Equation (6-5).

$$C = \left(1 - \frac{\pi_{M-1}}{\pi_M} \right) \tag{6-5}$$

When a test document is classified correctly, in the sense that the domain it is classified into is the same as the one assigned by SchoolNet or in Table 6-1, it should have a high confidence value. As shown in Equation (6-5), this indicates that there is a large difference between the confidence of the correct domain and that of the runner-up domain. On the other hand, when a document is classified incorrectly, its confidence value should be low. As indicated in Equation (6-5), this result implies that the confidence values of the highest and second-highest domains are very close, or nearly equal. Hence, the classifier should be able to detect and filter out its classification errors automatically, thereby improving its results.

6.3.4 Results and Discussion

The classification results of all three experiments are presented in Table 6-7. This table starts with the proportion of documents correctly classified (N_C), i.e. those for which the domain with the highest possibility is the one given in Table 6-1 or in SchoolNet, followed by the documents incorrectly classified (N_I). It then gives the average confidence and standard deviation of the correct classifications (C_C) and of the incorrect classifications (C_I). While the setup of the first two experiments only allows us to train and test the system once, it is possible to run the third experiment several times by selecting different random training and testing corpora. Consequently, two sets of results for the third experiment are shown in Table 6-7. To begin with, the third line of the table presents the initial results obtained on the first trial run of that experiment. As the table shows, the percentage of correctly-classified documents in this first trial reaches 81%. In order to validate that these results are robust and not the product of an accidentally easy-to-classify division of the SchoolNet data, the experiment was run 20 times. The average results of these 20 runs are presented on the fourth line of the table. As indicated, the percentage of correctly-classified documents now stands at 65% only.

Table 6-7: Classification results.

Experiment	N_C	N_I	C_C	C_I
1 (Brown Corpus)	90%	10%	0.51 ± 0.20	0.19 ± 0.19
2 (Reuters Corpus)	82%	18%	0.44 ± 0.23	0.25 ± 0.05
3 (SchoolNet Corpus) First trial run	81%	19%	0.77 ± 0.32	0.59 ± 0.39
3 (SchoolNet Corpus) Average of 20 runs	65%	35%	0.94 ± 0.20	0.83 ± 0.34

In addition to the results of Table 6-7, we have computed the precision and recall values of the classifier for each domain of the third experiment. By not a dissimilar route to that of Section 4.2, we defined a true positive as a document correctly classified in domain d_t, a false positive as a document incorrectly classified into domain d_t, and a false negative as a

document belonging to domain d_t but incorrectly classified into another domain, and computed the precision and recall values using Equations (4-1) and (4-2). We then computed the average precision and recall values for all domains and all 21 runs reported in Table 6-7, and obtained an average precision value of 87% and an average recall value of 81% for the first trial run, and an average precision value of 70% and an average recall value of 46% for the other 20 runs. The observed difference between the results of the initial single run of the third experiment and the average results of the subsequent 20 runs seem to indicate that the results of the initial run are not reliable.

It is important to stress however, that the results reported in Table 6-7 confirm that the classifier can always correctly classify a majority of the testing documents. Moreover, a significant difference can be observed between the average confidences of the correctly and incorrectly classified documents in all experiments. It is worth noting that this difference in the confidence measure is greater in the first experiment than in the other two. This follows from the fact that the domains in the second and third experiments are not as clearly defined as in the first experiment, which makes the classification problem more difficult. It is also of importance to note that the above observations are true for all three experiments, despite difference in the exact values of the correct classification rate and confidence. This is a significant finding, given that each experiment has tested the classifier under different conditions. Indeed, as explained earlier, the first experiment divided the corpus into three different domains, the second one divided it into three similar domains while the third divided it into 16 various domains. Moreover, in the first two experiments the system was trained and tested using triplets extracted from complete text articles and manually-created noun categories, while in the third experiment it was trained and tested using a single field from learning objects and automatically-generated noun categories. Yet despite these major differences, the results obtained from the classifier remain of rather high quality from one experiment to the next. This observation confirms the robustness of our method for computing the possibility distributions.

It appears from Table 6-7, however, that the 20-run average results of the third experiment are noticeably worse than those of the previous two experiments. This outcome was to be

expected. Indeed, it has already been mentioned in Section 6.3.2 that the SchoolNet Corpus presents a much bigger challenge for our classifier then the other two corpora. One important difficulty stems from the fact that many of the SchoolNet domains are composed of only a few hundred triplets, and sometimes even less than that, whereas the Brown Corpus counts over a thousand triplets per domain and the Reuters Corpus counts several thousands per domain. This is an important consideration because, as is the case for many other NLP methods that are trained with examples taken from a training corpus, our possibility distributions become more accurate when they are calculated with more triplets. With only a few hundred triplets per domain, the system suffers from data sparseness, the possibility distributions are less accurate, and the quality of the classification falls accordingly. The consequences of data sparseness in the SchoolNet Corpus were studied in more detail in Chapter 5.

This problem is further compounded by the fact that several of the SchoolNet domains are similar to one another. With too few training examples, the method cannot accurately learn the subtle differences between some of the more similar domains. Consequently, test documents belonging to these domains are hard to classify correctly. To verify the above contention, the classifier was again trained and tested 20 times, but with two separate sets of domains, and according to the same experimental setup as before. The first set is composed of five of the more distinct domains of the SchoolNet Corpus, namely “entrepreneurship studies”, “social sciences”, “sciences”, “career and vocational education”, and “physical education”. By contrast, the second set consists of two pairs of similar domains, namely “social studies” and “social sciences”, and “art” and “language art”.

The average results of these two sets of 20 runs are shown in Table 6-8, alongside the results of the 20 runs using the complete SchoolNet Corpus for comparison purposes. It can be seen from that table that the results do improve when the classifier is trained with the set of five distinct domains. Indeed, the percentage of documents correctly classified increases by almost 28%, and the gap in confidence between the correctly- and incorrectly-classified documents increases considerably. The precision value also shows a large increase, while the recall value is greatly improved. On the other hand, the results of the 20 runs using the two

pairs of similar domains show a marked deterioration in comparison with those of the previous set, although they are slightly better than those obtained from the complete corpus on account of the smaller number of overlapping domains. Thus, the percentage of correctly-classified documents falls by almost 10% in comparison with that obtained from the set of distinct domains, although it is higher by about 15% compared to the result obtained from the entire corpus. The gap in confidence between the correctly- and incorrectly-classified documents is lower however, than in the other two sets of domains. The precision and recall values, though better than with the complete corpus, are still lower than those of the set of five distinct domains.

Taken together, these results confirm our contention that data sparseness and overlapping domains can hinder the ability of the method to create reliable possibility distributions. Care should therefore be taken to avoid these two pitfalls in future work.

Table 6-8: Classification results with the SchoolNet Corpus.

Set of domains	N_C	C_C	C_I	Precision	Recall
Complete corpus	65%	0.94 ± 0.20	0.83 ± 0.34	70%	46%
Five distinct domains	83%	0.86 ± 0.24	0.64 ± 0.38	85%	80%
Two pairs of similar domains	75%	0.88 ± 0.24	0.79 ± 0.32	74%	71%

It is interesting to consider how our domain classifier performs relative to other text classifiers. To this end, we can compare the classification results of our classifier to those of other classifiers reported in the literature. We should point out that this comparison is purely for indicative purposes. Indeed, the classifier we implemented in this chapter is only a first prototype built for demonstrative purposes. Therefore, the tools resulting from years of efforts and refinements presented in the literature cannot serve as strict evaluation benchmarks, but only as loose indications of our progress. We have selected as benchmarks the dynamic vector space model of [98], the improved kNN multi-level classifier developed by [100], the improved Naïve Bayes algorithm proposed by [91], the sentence-space model classification introduced in [112], the combined layered clustering and K-means algorithm put forward in [92], the Bayesian classification approach integrating compound words that

was the focus of [5], and a classifier designed using the feature selection algorithm presented in [93]. Six of these seven classifiers provide success rates, and are presented alongside the results obtained with our method in Table 6-9. In addition, five of these benchmark classifiers exhibit precision and recall statistics that are contrasted to those of our classifier in Table 6-10. The results shown in these two tables reveal that our classifier's results rank favourably compared to that of the benchmarks. This is a very positive outcome, considering the fact that our classifier is still in its prototype stage. As our method becomes more refined and our implementation more sophisticated, we can expect the performance of our classifier to improve as the experimental data becomes closer to the theoretical distributions presented in Chapter 5, and to become equal, or even surpass, the performance of the current top classifiers.

Table 6-9: Comparison of classification results with various techniques.

Classification technique	Success rate
Our method	82% to 90%
Improved kNN multi-level classifier	91%
Improved Naïve Bayes algorithm	84%
Sentence-space model classification	92%
Layered clustering and K-means algorithm	66%
Bayesian classifier with compound words	66%
Feature selection algorithm	82%

Table 6-10: Comparison of precision and recall with various techniques.

Classification technique	Precision	Recall
Our method	85%	80%
Dynamic vector space model	82%	73%
Improved Naïve Bayes algorithm	81%	89%
Layered clustering and K-means algorithm	67%	67%
Bayesian classifier with compound words	66%	74%
Feature selection algorithm	84%	82%

On the other hand, the confidence of the correctly and incorrectly classified documents can vary a lot, as is indicated by the large standard deviation values in Table 6-7. This means that, despite the difference in confidence, there will be an overlap between the correctly and incorrectly classified documents. The cause of this variation may come from the errors introduced during the triplet extraction stage of the system. Indeed, as explained in Section 3.4, the occurrence of incorrect triplets is unavoidable in the training and testing corpora. The impact of the incorrect training triplets is minimized by the subsequent steps of the method, most notably the smoothing of the possibility distribution explained in Section 5.4. It remains true however, that the incorrect triplets of the testing corpus are used directly by the classifier in order to compute the possibility of each domain in which the testing documents can belong. It is most likely therefore that it is these incorrect testing triplets that cause the confidence of the classification to vary as wildly as it does in Table 6-7.

In Section 6.3.3, we showed that the confidence level can be used to detect and eliminate classification errors. To this end, the system simply needs to determine a confidence threshold under which the classification will be considered erroneous and rejected. Two possible thresholds have been considered here. The first is defined as the confidence level of the correct classifications minus one standard deviation ($C_C - STD$), and the second as the confidence level of the incorrect classifications plus one standard deviation ($C_I + STD$). The classification results after applying each threshold in the first two experiments are shown in Table 6-11.

Table 6-11: Classification results using threshold.

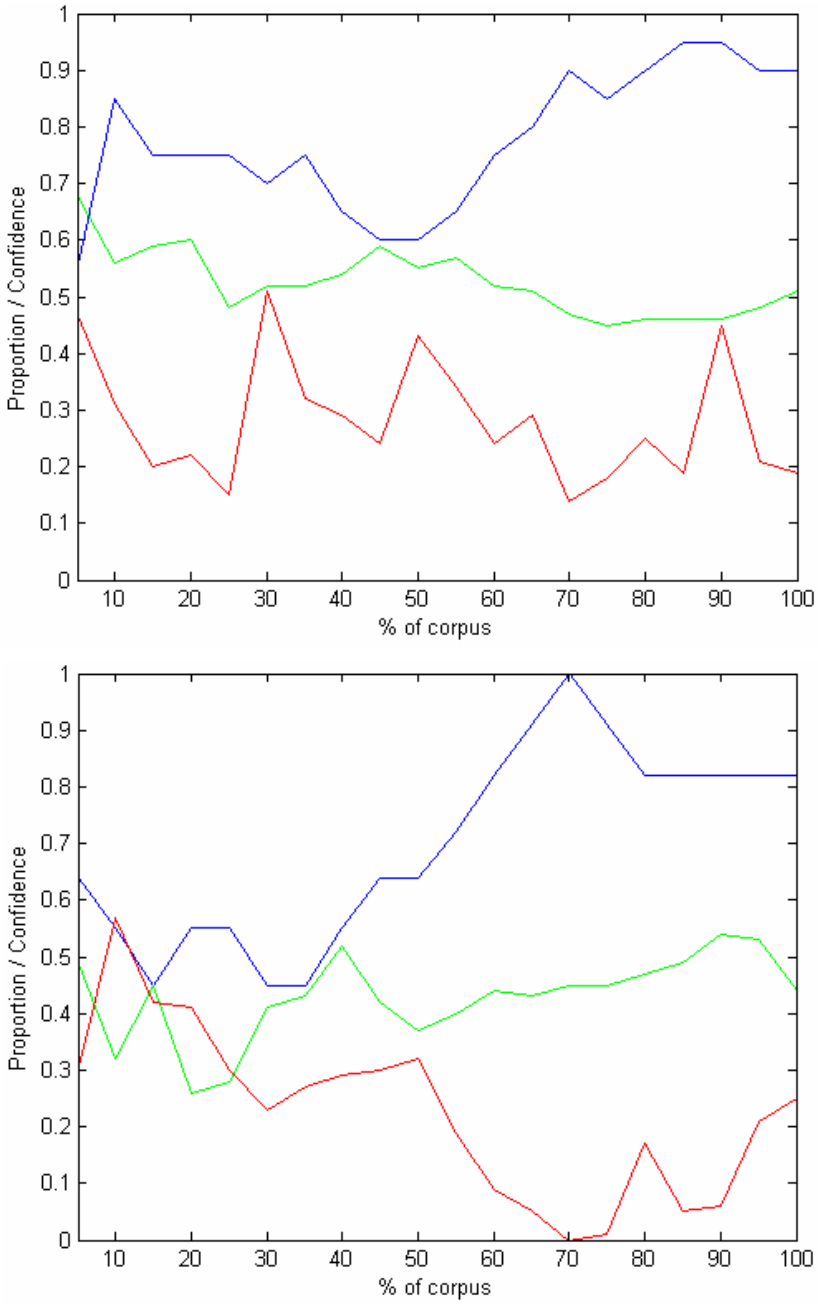
Experiment	$C_C - STD$		$C_I + STD$	
	N_C	N_I	N_C	N_I
1 (Brown Corpus)	80%	5%	75%	0%
2 (Reuters Corpus)	73%	9%	55%	0%

As expected, applying a confidence threshold allows the system to eliminate its incorrect classifications. Unfortunately, the overlap in confidence values discussed previously, leads the system to erroneously eliminate correct classifications as well. The first threshold allows

the system to eliminate half the incorrect classifications, but also reduces the proportion of correctly classified documents by 10%. The second threshold eliminates all the incorrect classifications, but entails yet another 10% loss in correct classifications. As the results of the third experiment displayed an even greater overlap in confidence values, we can expect that the loss of correctly-classified learning objects in that case will be even greater than 10%. It follows that for all practical purposes threshold levels for the classifier should be selected judiciously so as to weigh the number of correct documents needed against the number of incorrect documents that could be tolerated. This is a dilemma similar to the one encountered in normal text classifiers, whereby users have to weigh the precision against the recall of their system [76]. Note finally that, with regard to our classifier, once the number or the impact of the incorrect testing triplets is reduced, the use of the threshold can be expected to yield much more interesting results.

We pointed out in Section 6.3.2 that our method can function using training domains of varying lengths. It would be interesting at this point to study how the size of the training domains can affect the quality of the classification results. To this end, we trained the classifier using sub-samples of the first two training corpora, starting with 5% of the triplets of each domain and increasing by 5% intervals. New classification results were generated for each sub-sample, and are shown in Figure 6-1. The three results we chose to illustrate in that figure are the proportion of testing documents that were correctly classified (blue line), the average confidence of the correct classifications (C_C, green line), and that of the incorrect classifications (C_I, red line). The first thing we notice in this figure is that, save for one brief exception, C_I is always less than C_C. This is clear evidence that our method is reliable. Indeed, had it been otherwise, we would have expected to find no consistent distinction between the possibility of correct and incorrect classifications, and therefore to find that C_C and C_I have similar values. The only exception, which occurs over the first 25% of the Reuters Corpus, can be explained by keeping in mind that the domains of the Reuters Corpus are very similar to each other. Given a training sample too small in length, our system is simply not able to learn to distinguish between these domains in a reliable manner. Furthermore, we can note that the classification results mostly improve with the size of the

training corpus. Indeed, while C_C remains roughly constant overall throughout the tests, the proportion of correctly-classified documents shows a general trend upward, and C_I a general



(a)

(b)

Figure 6-1: Experimental results gathered using a given percentage of the Brown Corpus (a) and the Reuters Corpus (b).

trend downward. This result was to be expected. Having more triplets allows the method to compute semantic distances and conditional probabilities that better model the real-world usage of the noun-verb pairs, which in turn makes the possibility distributions more accurate and improves the results.

6.4 Other Applications

It is important to note that the domain classifier presented in the previous section is not the only possible application for our method. In fact, the method was not developed with a specific application in mind, but is intended to provide a flexible mathematical core that can be adapted to any NLP problem. To highlight this fact, four problems, two of them encountered in the LORNET project, will be presented hereafter and solved with the classifier. In each case, we begin with a brief description of the research project and the problem attributable to it, before giving a general indication of how our method can be modified to find a solution.

6.4.1 LORNAV

It is well known that searching and navigating through learning object repositories (LOR) with traditional two-dimensional user interfaces is a tedious and counter-intuitive task. The LORNAV system has been proposed to remedy this situation [67], [39]. LORNAV is a three-dimensional visualisation environment that enables a user to navigate through a LOR as if it were a virtual 3D world. The learning objects (LO) of the LOR are displayed as objects in the virtual world, and are positioned throughout the world in a way that reflects the objects' relationships to the domains the user is interested in.

The method proposed in this document could be successfully applied in the context of LORNAV. Indeed, as of yet, the LORNAV system possesses no formal means of computing the relationship between the LO and the domains. This relationship can be supplied by our method, in the form of the possibility of the domains given each LO. To illustrate this concept, Figure 6-2 shows the simple example of a world composed of two domains d_1 and d_2, and two learning objects LO_1 and LO_2. One way to populate such a world could be to

begin by choosing the positions of the domains, then to compute the possibility of each domain given a LO, and finally to use these possibilities as measures of the distance between the domains and the LO, in order to place the LO in the world accordingly.

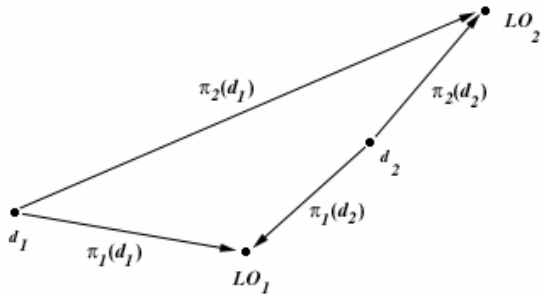


Figure 6-2: A simple LORNAV world.

6.4.2 Semantic Signatures

With the rapid popularisation of the Internet and of e-learning, many institutions rushed to create their own LORs. This situation created a need for efficient distributed search tools, to allow users to retrieve information from several LORs at once. However, the design of such tools cannot be done without addressing the problem of interoperability in a federated environment. Indeed, each of these LORs uses different semantic terms and ontologies, which reflect the institution's target audience and background. This introduces variations in the metadata values which, combined with the lack of metadata format standards, makes direct keyword queries across LORs ineffective.

One possible avenue to solve this problem is to represent each LO with a semantic signature [20], [21]. This semantic signature is computed using a three-step process. First, the system finds the most important metadata fields, such as "title" and "description", which should be present in one form or another in all LO. By focusing its scope on these fields, the system effectively bypasses the lack of metadata format standards. Next, the system detects the most meaningful keywords in the target fields. These keywords are finally replaced with their entry from a common online lexical reference, such as WordNet. These last two steps eliminate the linguistic variations present in the metadata values, and represent all LO using a

common vocabulary. Once the LO are represented by their semantic signatures, they can easily be compared with the semantic signature of a user's query, and classified accordingly.

The methods proposed in this document can offer an alternative method of computing the semantic signatures. This new method stems from the fact that our approach is made up of two roughly independent parts, namely, the triplet extraction method and the possibility distribution method. To begin with, several triplet extraction modules could be implemented, with each module specially adapted to the extraction of triplets from one or another of the LORs that the system must handle. This setup has two advantages. First it effectively bypasses the problem of lack of metadata standards, which is the source of the problem. Second, if all the extraction modules use a common list of noun categories, it eliminates the linguistic variations between LORs. In the second segment of the method, all the extracted triplets can be directed to a common possibility distribution module, which will compute the possibility of each domain given each triplet, regardless of which LOR it originated from. This procedure requires that the system identifies the presence of a domain in several LORs, in order to combine together all its triplets. Such an identification could be accomplished in two alternative ways. In the first alternative, the system can easily find recurring domains in several LORs by mapping the domains of each LOR to a common domain hierarchy. This alternative requires that the system be given rules to perform the mapping. In the second alternative, the system would compute the possibility distributions of each domain in each LOR. Given that the differences between LORs have already been eliminated in the triplet extraction stage, similar domains in two LORs will have similar possibility distributions. The system, in this second alternative, will then have to find the domains with similar possibility distributions and combine their triplets together. The structure of the resulting system is illustrated in Figure 6-3. Following this procedure, the semantic signature of a LO would be the possibility distribution of its domain given the LO's triplets. Finally, the user's query, which must include both triplets and a domain, is also transformed into a set of possibility distributions. The distributions representing the query and the LO can then be compared, using any one of the well-known signal comparison techniques available.

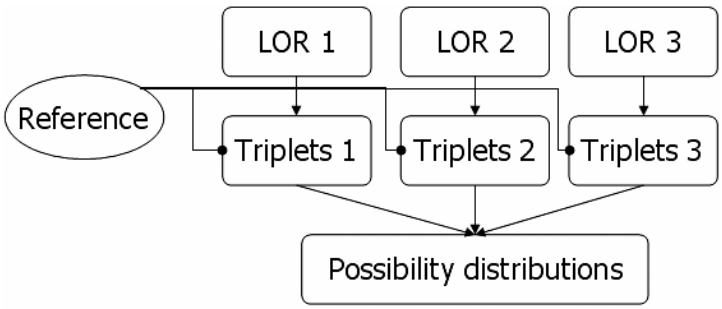


Figure 6-3: The system to compute the semantic signatures.

6.4.3 Keyphrase Extraction

Another possible application of our method that deserves to be explored is keyphrase extraction. A *keyphrase* is defined in [86] as a sentence that accurately represents the main topic of a document. Essentially, the idea is that discovering such sentences from documents can be useful in many fields of application, including information extraction, text summarization, and indexing. According to [36], keyphrase extraction methods can be divided into two broad classes, namely those designed to extract keyphrases from a single document, and those designed to handle entire sets of documents at once. A detailed overview of single-document keyphrase extraction approaches and of several related fields of research can be found in [86]. This survey highlights the fact that, although these approaches are varied and innovative, they commonly suffer from the weakness of extracting long lists of sentences that include a lot of irrelevant phrases.

With a few modifications, the domain classifier presented in this chapter could be adapted to the task of keyphrase extraction from single documents. In this context, the keyphrases are those that best represent the main topic of the document, and therefore contain a lot of information that is relevant and meaningful in that topic. Recall that, as part of the classification process, our classifier computes for all practical purposes the possibility of each domain given each triplet of a document. Suppose instead that the application already knows the correct domain to which the document belongs, either because it was successfully classified by the program or because it was specified as such by a user. Under these conditions, the computation can be limited to the calculation of the possibility of that domain given each triplet. This will allow the system to rank the triplets of the document by

possibility, or alternatively by how meaningful they are in that domain. It will then be necessary to give our keyphrase extraction system the ability to keep track of which sentence each triplet was extracted from, something that was not required for the domain classifier presented in this chapter. With this additional ability, finding the most important sentences becomes a simple matter of tracking back the most meaningful triplets to their source. Moreover, extra importance could be given to sentences from which several meaningful triplets were extracted. Finally, to avoid the problem of extracting too many sentences, the number of meaningful triplets retained can be limited in a number of ways, such as for example keeping only a pre-set proportion of the total number of triplets, or imposing a possibility threshold.

6.4.4 Search Tool

By not a dissimilar route used to build the domain classifier discussed extensively in this chapter, we can devise yet a fourth application of our method which yields an original search tool. To be sure, this application represents an expansion of the domain classifier presented earlier, rather than a completely new tool like the previous three applications discussed in this section. Indeed, a careful look at our classifier reveals that it could also be made to serve as a new type of search engine. This can be accomplished by training the classifier with a corpus that includes user-defined domains. For example, a scientist could create a domain made up of papers related to his field of research, while a voracious reader could create a domain composed of chapters taken form his favourite novels. Once the classifier is trained to recognise these special domains, it can subsequently be used to classify a large corpus, or even the entire Internet. As such, the search will turn out all the documents classified as belonging to the user's domains of interest, sorted out by their possibility value. Moreover, the confidence measure defined in Equation (6-5) can then be used to filter out false positives, thus insuring that all the documents identified in the search are relevant to the domain defined by the user.

6.5 Conclusions

In this chapter we have detailed several examples of possible applications of the method developed in this study. Its main focus has been on an in-depth study of the implementation of a domain classifier based on the proposed possibility distribution technique. The analysis has covered every step of the implementation process, and the experimental results obtained from the classifier showed that the proposed method is reliable and valid.

The usefulness of the method we propose in this study is not limited to domain classification. To illustrate this point, four NLP problems, two of them specifically encountered in the LORNET project and two of general interest, were introduced as testing grounds for our procedure. In all cases, the study has indicated the ease with which our approach could be applied to help solve these problems successfully in the real world.

Chapter 7
Conclusions and Future Work

7.1 Summary of the Study

This research has dealt with the development of new methods for extracting semantic knowledge from unannotated written English documents and to represent this information using a formal mathematical expression to facilitate its use in practical applications. The study was done in the context of the PAMI lab's contribution to the LORNET project, and its setting and motivation are presented accordingly in Chapter 1.

To set ideas, we started with a review of the most significant modern techniques relevant to our work in Chapter 2. The techniques reviewed were developed specifically to extract and handle the semantic information contained in text documents. This background analysis revealed that the techniques presented in the literature were fraught with constraints that severely limited their usefulness in this study.

This review of the literature set the stage for the development of methods designed to tackle the related issues of semantic information extraction and representation. These two goals are addressed by two separate methods, as detailed in this thesis.

Turning first to triplet extraction, Chapter 3 introduced the first method we proposed to address this task. We began the chapter with a survey of modern semantic taggers and an analysis of their relevance to our work. We then introduced the triplet extraction method developed for this research. Its mode of operation was explained in detail and illustrated with an example. Its shortcomings were also exposed and deemed significant enough to justify the search for a more robust alternative. By introducing a rule-learning algorithm, we designed a more valid method for triplet extraction, which is presented in Chapter 4. The key to the success of this method is its use of a part-of-speech hierarchy that lies at its core. The theoretical development and the experimental results discussed in this chapter confirm that our method can be used to obtain an in-depth analysis of texts without being limited to a particular domain of application.

The method we developed to represent domain information using possibility distributions is presented in Chapter 5. This method relies on the principles of semantic distances, conditional probabilities, and possibility distributions, which are all introduced in this same chapter. We discuss each of these notions in detail and provide an explanation of relevant concepts along with the theoretical justification and the mathematical development of each concept. As befits a new approach, we also provided in Chapter 5 a detailed study of the mathematical foundation underlying the proposed method. For each step of this method, we analysed the possible values of its input variables as well as the range and significance of its possible outcomes. To substantiate our examination, we contrasted the predicted theoretical outcome of each step of this new approach with the results generated through its implementation. Finally, putting together all the theoretical results reached in this chapter illustrated the complete workings of our knowledge representation method.

To add more relevance to the discussion of this possibilistic method, we illustrated its practical applicability with real-world examples in Chapter 6. Our main focus in this chapter is the in-depth study of the implementation of a domain classifier based on the proposed method. This implementation allowed us to deal with each step of the implementation process in detail, and to gather experimental results that show that the proposed method is reliable and valid. The implementation process also clarified the link between the knowledge representation method and the triplet extraction presented earlier. To illustrate the flexibility of the method, we discussed its applicability to four more NLP challenges, including notably those of standardization and navigation through learning objects, which were encountered in applications related to areas of the LORNET project.

7.2 Appreciation of the Results

Putting together the theoretical developments and the experimental results presented in this research allows us to reach definitive conclusions with regard to the validity of our methods. We established the mathematical foundation and statistical validity of both the rule-learning method of Chapter 4 and the possibilistic method of Chapter 5. We are confident that our two approaches will withstand the highest levels of scrutiny.

Turning to the implementation phase of the two methods, we began with implementing the rule-learning method into a practical algorithm. The experimental results obtained from this test, as presented in Chapter 4, reveal that the precision of our rule-learning method compares favourably with the two reference algorithms, while its recall surpasses that of the references. Moreover, the rules learned can also extract very specific information without the rule base becoming domain-specific or growing to an unmanageable size.

The possibilistic method of Chapter 5 was also successfully implemented in a domain classifier, as explained in Chapter 6. The classification precision and recall values obtained from this implementation reveal that our classifier's performance ranks favourably in comparison with state-of-the-art classifiers found in the literature. Another important result was that our experimental classifier has not only classified correctly the vast majority of its testing documents, but it also gave on average a significantly higher confidence score to the correct classifications than to the incorrect classifications. These results confirm our hypothesis that most of the semantic information of a document lies in the actions described by that document, and that our method accurately extracts and represents this semantic information. Indeed, had it been otherwise, we would have found the classification results to be much lower, and more importantly we would have found the average confidence of both the correct and incorrect classifications to be roughly equal.

Although significant, the implementation results of our two methods were not unequivocally positive. Indeed, the precision and recall values of the implementations reported in Chapters 4 and 6 show that there is still room for improvement in the methods. In the final analysis, however, the original goals of our study, which were to develop new methods to extract semantic knowledge from English documents and to represent it using a formal mathematical expression to facilitate its use in practical applications, have been achieved.

7.3 Contributions

Four key contributions to the field of natural language processing emerge from this research. The first main contribution, with important theoretical implications, is to demonstrate the

merit of representing semantic information in the form of actions, or subject-verb-object triplets.

The second contribution is the design of a rule-learning method for keyword extraction rules with a part-of-speech hierarchy at its core. Indeed, no such hierarchy has ever been devised before. Nor is there an agreement among grammarians on what exactly constitutes an independent part-of-speech, despite the fact that they are one of the basic elements of language.

The third contribution is the design of a method to represent the actions described in text documents. The central concept of this new method, and what sets it apart from other contributions to this field, is the mathematical development we formulated in order to represent the semantic information of the triplets using possibility distributions.

The fourth contribution derives from the implementation of our methods in practical applications. These implementations illustrate the way in which our theoretical ideas can be used for practical purposes. More importantly, the experimental results obtained from these implementations confirm that the methods are reliable in practice.

7.4 Future Work

7.4.1 Type-2 Fuzzy Sets

The mathematical development in Chapter 5 has demonstrated that, given enough training data, the scatter of conditional probability/semantic distance points converges to one of the four relationships illustrated in Figure 5-12. Furthermore, the method developed in that chapter showed how possibility theory can be used to approximate these relationships on the basis of statistical information derived from a training corpus. However, the precision with which the relationship can be estimated is directly related to the number of training examples from which the statistics are computed; with insufficient training examples, the statistics are inherently imprecise, as are the possibility distributions estimated from them. The impact of this lack of precision can be observed graphically: the scatter of probability/distance points becomes widespread and irregular, as shown in Figure 5-13(b).

In this study, we use a linear approximation of the scatter of probability/distance points to represent the possibility distribution, regardless of how scattered the points were originally. Although we have demonstrated both theoretically and practically that this is a sound approximation, a future question is whether a different approach using type-2 fuzzy sets (T2FS) would be equally successful.

One of the main limits of type-1 fuzzy sets (T1FS) is that they eliminate the linguistic uncertainty from the words that they represent [59]. Indeed, T1FS categorize words by relying on precise membership functions (MF). These MFs are meant to represent the uncertainty about the meaning of words; however, since they are precise functions, they actually eliminate this uncertainty. Indeed, a word can have different meanings for different people, and can thus be represented by different MFs. However, by forcing the selection of one precise MF to represent the word, T1FSs effectively remove this kind of uncertainty. T2FSs make it possible to overcome this weakness by creating MFs that are not precise functions but rather fuzzy sets – creating in effect fuzzy fuzzy sets [60]. Said differently, instead of representing a word as an exact MF, it is represented in T2FS as a continuum of precise MFs. Each individual MF represents one possible understanding of the meaning of the word, and the multiple MFs combine to form the footprint of uncertainty (FOU) of the fuzzy set.

Consider the temperature scale example discussed in Section 2.3. The words “cold”, “mild” and “hot” hold different meanings for different people, and if each individual in a sample group was asked to draw the fuzzy sets representing these words, no two sets are likely to be exactly the same, nor would they be exactly like the ones illustrated in Figure 2-1. The MFs of Figure 2-1 do not capture this uncertainty; quite the contrary, they eliminate it by mapping, for example, 10 degrees to exactly 0.8 “cold”. If one were to draw the MFs while taking into account the different interpretations of the words “cold”, “mild” and “hot” that each member of the sample group holds, it might look more like Figure 7-1. In that figure, the filled-in area inside the MF is the FOU, and it contains the continuum of MFs plotted by the sample group to represent the three target words.

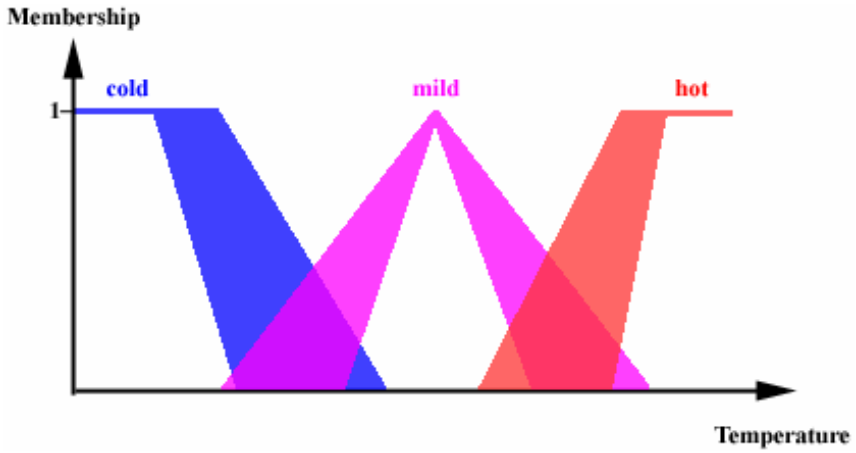


Figure 7-1: Type-2 fuzzy sets dividing a temperature scale.

In much the same way that the MFs in the above example were drawn using the input of several people in a sample group, the statistics in our research were computed from several sources, namely the many domains forming the training corpus. While each MF in that example represents the meaning of the three temperature words according to each person, each point in our probability/distance scatter diagrams represents the meaning of a noun-verb pair in a domain. We can thus see a parallel emerging between T2FS and our work. While we use scatter diagrams such as the ones of Figure 5-13 to compute possibility distributions, it is quite conceivable to use them to delimit the FOU of fuzzy sets in an application of T2FS.

This line of argument points to a completely new direction for our research albeit one beset by important intellectual problems that prevent its implementation at the present time. Of great importance is the fact that, while building T2FS based on groups of T1FSs is a straightforward matter, to the best of our knowledge no one has ever attempted to do so on the basis of scatters of points. A second problem comes from the need to introduce type-reduction and defuzzification methods into our method. In a typical implementation of a type-2 fuzzy logic system, the T2FSs are reduced to T1FSs and then defuzzified into the crisp values that constitute the output of the system [60]. However, the mathematical techniques used to perform the type-reduction and the defuzzification vary depending on each situation, and determining the best ones to use in the context of our method will require an in-depth study of the question. Finally, it may prove necessary to assign weights to the different regions of the FOU. Almost all current applications of T2FS assume a normal

weighting scheme; however, in our method, it could be preferable to assign a greater weight to the regions of the scatter diagram that have the greatest concentration of points so as to reduce the impact of outliers and of empty regions in the diagram. Unfortunately, this creates new problems, as at this time there exists no practical way to perform the calculations required to handle T2FSs with non-uniform weights [60].

In light of this discussion, this new direction of research falls far outside the scope of this thesis and leads into a territory that currently lacks the practical tools needed to implement and test the work. It will be up to future studies to explore the feasibility of refining and expanding upon our research.

7.4.2 Other Directions

A few avenues for future work follow from our research. In the overall context of this study where the emphasis is on finding triplets that represent the central actions described in a document, the triplet extraction method can be taken one step further by limiting its scope to key phrases in the text documents. With such a setup, the method would first recognise the key phrases in the text, and then extract the actions represented in those phrases only instead of extracting the actions from all phrases of the text. This modification can be implemented without losing the essence of our method by implementing a keyphrase extraction system that assumes no *a priori* knowledge about the text documents, as in that presented in [36]. This would improve the accuracy of the method by limiting it to relevant actions. However, this improvement will come at the expense of a loss of data with which to compute the semantic distances and the probabilities.

Another drawback of the triplet extraction process is related to the use of the noun categories introduced in Section 3.2.3. In that section, we proposed a partially-automated process to generate the categories. However, this task should be entirely automated to make the system more rigorous and to spare the user this long and tedious work. The semantic distance measure introduced in Section 5.2.4 could be used to this end. Indeed, we can conceive of a system that forms categories by grouping together nouns that are at a small semantic distance from each other. This approach would also aid us in addressing noun

polysemy by allowing these nouns to be close to the several groups that represent their many different meanings. Expanding on an idea mentioned in Section 6.3.2, we can see that to obtain the most general noun categories, the system can perform the clustering task on only the most general training corpus at its disposal. The system can then specialize the categories in the following way. First, it computes the semantic distance of the pairs of nouns within each category using a new training domain. Then, by using a data-partitioning algorithm [24] or a data-clustering algorithm [4], it detects cases where these distances cluster some of the nouns into isolated sub-groups. Finally, it splits these sub-groups into new categories.

Moreover, we should not overlook the fact that this research is carried out within the context of the LORNET project. As such, it is imperative that it should effectively handle not only natural text documents, but also metadata files, which are a major feature of this project. The implementation carried out in this study shows that this can be done by isolating natural-text fields of the metadata objects, such as “description” fields, and extracting triplets from them. Although our experimental results prove the feasibility of this approach, the analysis uncovers a shortcoming: discarding information by ignoring some important metadata fields, such as “title” or “keywords”, whose entries are not written in grammatical sentences but rather in lists of keywords. A possible solution to this problem would be to enrich the triplets extracted from text fields with information taken from keyword fields. For instance, based on the information of the keywords, some general noun categories used in the triplets could be replaced by more specific ones. One foreseeable challenge of applying this solution is that, while the present system considers all sentences as equivalent, different metadata fields can have a different meaning and a different level of importance, which must be taken into consideration.

The process used to compute the possibility distributions in Chapter 5 could also be the subject of future research. For example, it would be interesting to study the effect of making the computation of the semantic distance domain-specific instead of computing the distances over the entire training corpus as is currently the case. It is difficult to anticipate what the impact of this change will be. On one hand, the semantic distance of pairs that include a polysemous word with two very different meanings in two domains should be more accurate,

which could lead to more accurate possibility distributions. On the other hand, as Chapter 5 illustrated, data sparseness is of significant concern in our computations, and the proposed change would compound this problem by computing the semantic distance of all pairs based on a much smaller corpus. This could lead to less accurate distance measures and thus less accurate possibility distributions. In the same vein, distinguishing between subject-verb pairs and object-verb pairs, instead of considering them both simply as noun-verb pairs, leads to a similar dilemma. Grouping the data from both pairs together would lead to more complete occurrence probabilities in Section 5.2.5, which would improve the possibility distributions. However, the meaning of a noun-verb pair can vary depending on whether the noun is the subject or object of the verb; eliminating this difference would lead to a loss of information which could hurt the overall system. Predicting which of these two solutions is more preferable would require further examination.

An interesting practical improvement that future work should seek to achieve consists in reaching the goal mentioned in Section 6.3.1, which is to expand the first training corpus so it includes the entire Brown Corpus. This would enable the domain classifier to classify documents covering all domains. Although the classifier's framework is complete, ample work remains before the entire Corpus can be integrated in it. To our knowledge, a possibilistic domain classifier on such a scale has never been devised. In addition to expanding the system horizontally by adding new domains, the possibility of expanding it vertically by adding sub-domains to the domains already implemented should also be explored. This would transform our classifier into a hierarchical classifier capable of making an initial classification and subsequently refining it. We already hinted to future work in this direction in our study, as the second training corpus in Section 0 is composed of sub-domains of one of the domains of the first training corpus. However, in this study, the domain classifiers trained with both corpora are not used sequentially. Instead, each one classifies its testing documents independently of the other. A hierarchical classifier, on the other hand, could, for example, begin by analyzing a testing document and classify it as belonging to the business domain. Then it would sub-classify the document as a text dealing with a corporate takeover, and further sub-classify it as describing either a friendly or a hostile takeover. The

implementation of this hierarchical structure would be quite different from the single-level system architecture presented in this study and would raise a number of new concerns. One such concern relates to applying the winner-takes-all approach of Section 6.3.3 to the smaller domains. It could be computed by considering the possibility of all child domains given a triplet with or without considering the possibility of the parent domain, or even by comparing the possibility of each child domain to that of the parent domain individually. Each scheme has its own advantages and drawbacks, and further efforts are needed to figure out the most appropriate scheme for our approach. Another concern has to do with how far down in the child domains the system should try to classify a testing document, and what its cut-off conditions to end the classification should be. Most importantly, the level of classification accuracy that our method can allow, and how precise the child domains can become, are also of concern.

* * * * *

The central theme of this thesis has been to develop and apply two new and robust methodological frameworks that can lend themselves to further application for the extraction and representation of semantic knowledge in texts. As such, the disciplined procedures developed in this work have the presumed potential to contribute to the scientific progress of this field of research for many years to come.

Appendix A

Parts-of-Speech of the Penn Treebank

For convenience, this table lists the 46 parts-of-speech of the Penn Treebank. It includes the symbolic tag that represents the part-of-speech in applications, the name of the part-of-speech, and either examples of the lexical item represented, a description thereof, or both. A complete and detailed presentation of this set of part-of-speech can be found in [74]. It is worth noting that the "blank" part-of-speech is not part of the Penn Treebank set, but an addition of our work.

Tag	Name	Description
	Blank	A missing word
#	Number sign	#
$	Dollar sign	Any word with the symbol $ in it
(	Left bracket	([{
)	Right bracket	)] }
:	Colon	: … - ;
,	Comma	,
.	Period	. ? !
"	Left quotes	"
``	Right quotes	``
"	Quotes	"
CC	Coordinating Conjunction	and, &, or, but, either, neither, plus, minus, v., thenceforth
CD	Cardinal Number	Any number, in digits or written out, and any word with a number in it
DT	Determiner	a, all, an, another, any, both, each, either, every, no, neither, some, that, the, these, this, those
EX	Existential there	there
FW	Foreign Word	non, etc, e.g., i.e. Any word not in the English lexicon
IN	Preposition / subordinating conjunction	about, above, across, after, against, along, although, among, amongst, anti, around, as, aside, at, because,

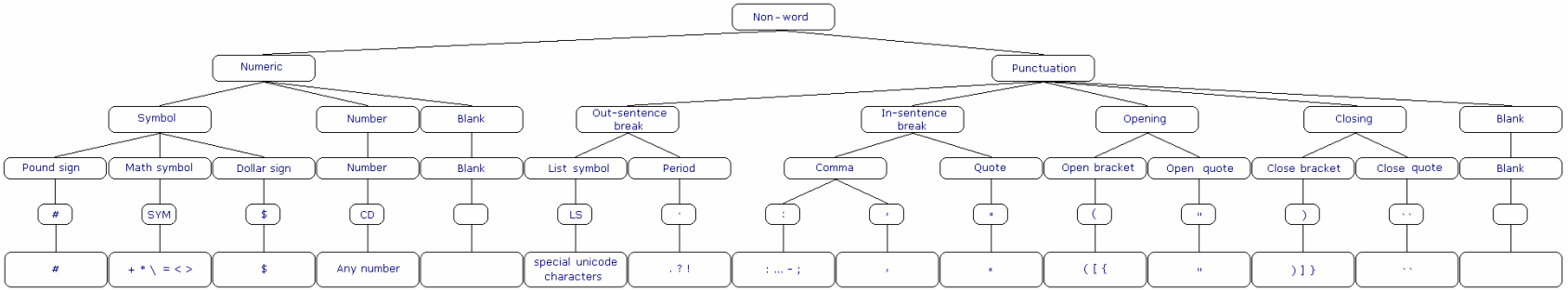
		before, behind, below, beside, between, beyond, bi, by, down, during, except, for, from, if, in, inside, into, lest, like, 'n, near, of, off, on, onto, out, outside, over, per, since, so, than, that, the, though, through, throughout, towards, 'till, under, unlike, until, up, upon, via, vs., whether, while, with, within, without
JJ	Adjective	Any adjective not JJR or JJS
JJR	Adjective, comparative	better, greater, larger, more, worse Any adjective finishing with –er
JJS	Adjective, superlative	best, greatest, largest, most, worst Any adjective finishing with -est
LS	List item marker	Special unicode characters
MD	Modal	can, cannot, could, couldn’t, may might, must, ought, shall, should, shouldn’t, will, would, 'll
NN	Noun, singular or mass	Any singular noun
NNP	Proper noun, singular	Any plural noun
NNS	Noun, plural	Any singular proper noun
NNPS	Proper noun, plural	Any plural proper noun
PDT	Predeterminer	all
POS	Possessive ending	' 's
PRP$	Possessive pronoun	her, his, its, my, our, their, your
PRP	Personal pronoun	he, him, her, himself, herself, itself, it, me, myself, oneself, our, ourselves, ownself, she, them, themselves, they, thou, thy, thyself, us, we, you, yours, yourself, yourselves
RB	Adverb	Any adverb not RBR, RBS or RP
RBR	Adverb, comparative	earlier, prouder, neater, closer, more Any adverb finishing with –er
RBS	Adverb, superlative	bluntest, most Any adverb finishing with –est
RP	Adverb, particle	out, down, up, away
SYM	Symbol	+ * \ = < >
TO	To	to
UH	Interjection	ah, alas, amen, bah, dammit, eh, fella, gee, gimme, goddammit, golly, hello, hey, hmfp, ho, howdy,

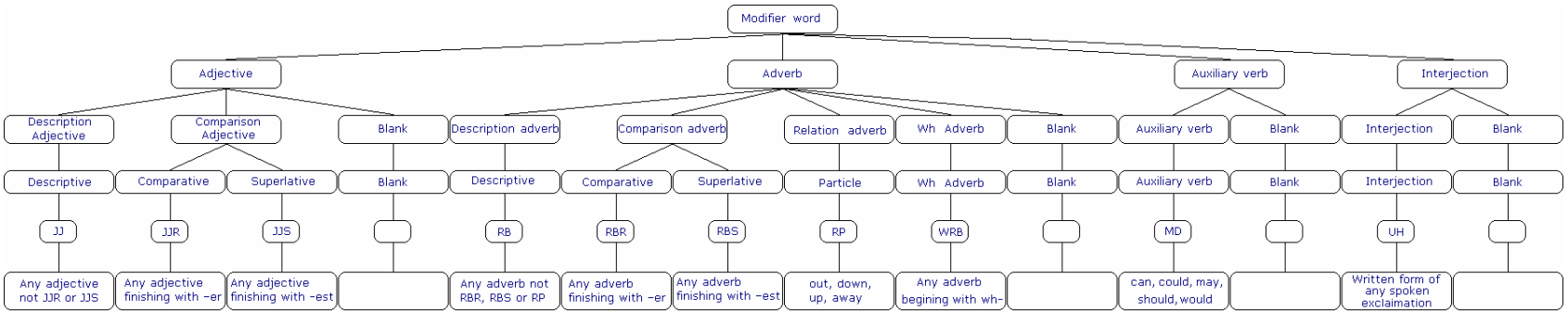
		hurrah, jeepers, mmm, ok, oops, shucks, tsk, uh-huh, wham, whoa, wow, yeah
VB	Verb, infinitive	Any infinitive verb (preceded by 'to')
VBD	Verb, past tense	was, were, lay, had, found, became Any verb finishing with –ed
VBG	Verb, gerund / present participle	Any verb finishing with –ing
VBN	Verb, past participle	been, given, found, put, read, held Any verb finishing with –ed
VBP	Verb, singular present non-3rd person	Any singular-present verb (infinitive verb not preceded by 'to')
VBZ	Verb, singular present 3rd person	Any verb finishing with –s
WDT	Wh– determiner	which, whichever, what, whatever, that
WP$	Wh– possessive pronoun	whose
WP	Wh– pronoun	who, whosoever, whoever, whom, what
WRB	Wh– adverb	how, when, whence, where, whereby, wherever, why

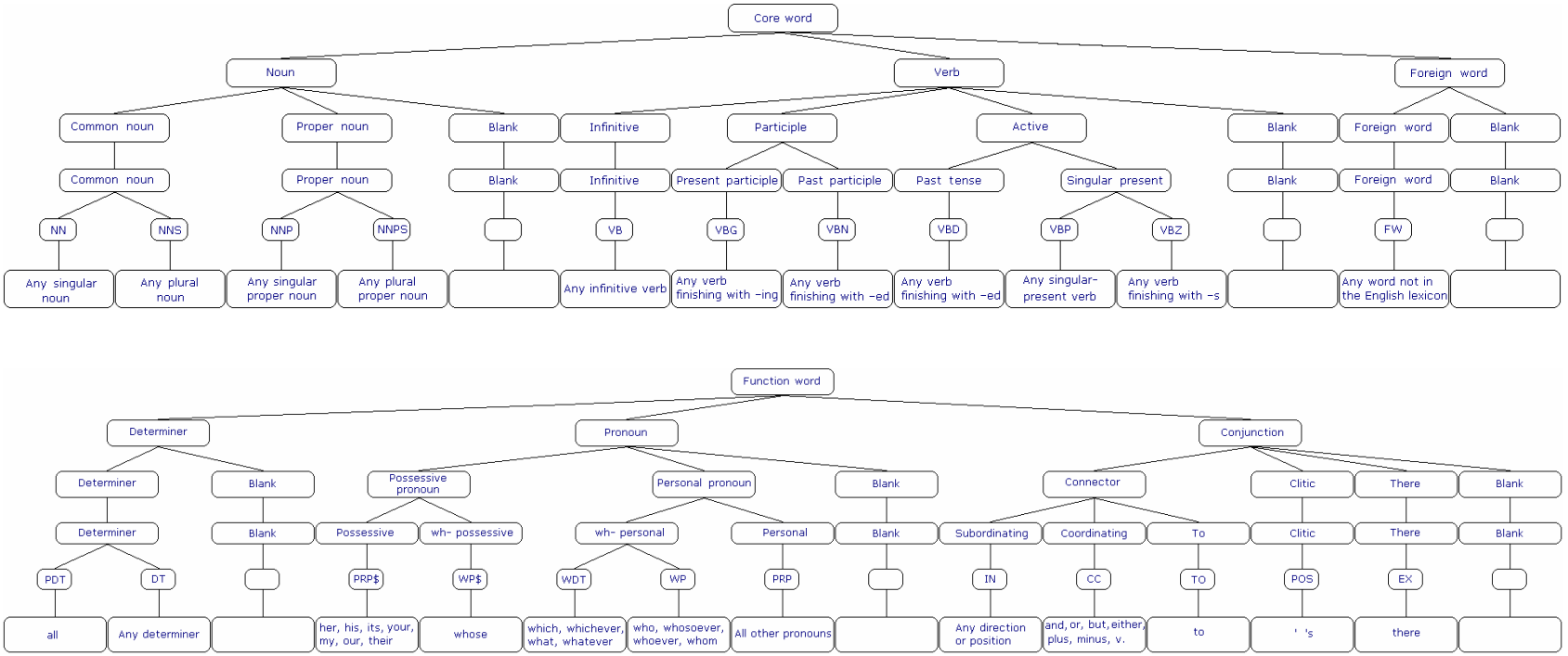
Appendix B

Graphical Representation of our Part-of-Speech Hierarchy

Due to space restriction, we show here each of the four super-lexical categories separately, along with all their sub-levels. The top level, in which these four super-lexical categories are grouped together in the single "Universe" category, is not shown here.






Core word
Noun
Verb
Foreign word
Common noun
Proper noun
Blank
Infinitive
Participle
Active
Blank
Foreign word
Blank
Common noun
Proper noun
Blank
Infinitive
Present participle
Past participle
Past tense
Singular present
Blank
Foreign word
Blank
NN
NNS
NNP
NNPS
VB
VBG
VBN
VBD
VBP
VBZ
FW
Any singular noun
Any plural noun
Any singular proper noun
Any plural proper noun
Any infinitive verb
Any verb finishing with -ing
Any verb finishing with -ed
Any verb finishing with -ed
Any singular-present verb
Any verb finishing with -s
Any word not in the English lexicon

Function word
Determiner
Pronoun
Conjunction
Determiner
Blank
Possessive pronoun
Personal pronoun
Blank
Connector
Clitic
There
Blank
Determiner
Blank
Possessive
wh- possessive
wh- personal
Personal
Blank
Subordinating
Coordinating
To
Clitic
There
Blank
PDT
DT
PRP$
WP$
WDT
WP
PRP
IN
CC
TO
POS
EX
all
Any determiner
her, his, its, your, my, our, their
whose
which, whichever, what, whatever
who, whosoever, whoever, whom
All other pronouns
Any direction or position
and, or, but, either, plus, minus, v.
to
' 's
there

Appendix C
Publications

The following publications are based on parts of the research and results presented in this thesis.

Journal Papers

- R. Khoury, F. Karray, Y. Sun, M. Kamel, and O. Basir, "Semantic Understanding of General Linguistic Items by Means of Fuzzy Set Theory", in *IEEE Transactions on Fuzzy Sets and Systems*, Volume 15, Issue 5, October 2007, pp. 757-771.
- R. Khoury, F. Karray, and M. Kamel, "A Novel Approach for Extracting and Representing Actions in English Documents", in *Computational Linguistics*. Submitted in April 2007, currently under review.
- R. Khoury, F. Karray, and M. Kamel, "An Improved NLP Method Based on a Part-Of-Speech Hierarchy", in *IEEE Intelligent Systems*. Submitted in June 2007, currently under review.

Conference Papers

- Y. Sun, R. Khoury, F. Karray, and O. Basir, "Semantic Context Classification by Means of Fuzzy Set Theory", in *Proceedings of the 2005 IEEE International Conference on Natural Language Processing and Knowledge Engineering (IEEE NLP-KE'05)*, Wuhan, China, 30 October – 1 November 2005, pp. 250-255.
- R. Khoury, F. Karray, and M. Kamel, "A Fuzzy Classifier for Natural Language Text using Automatically-Learned Fuzzy Rules", in *The second International Conference on Artificial & Computational Intelligence for Decision, Control and Automation - International Conference on Machine Intelligence (ACIDCA-ICMI'2005)*, Tozeur, Tunisia, 5-7 November 2005.

- R. Khoury, F. Karray, and M. Kamel, "A Method for Extracting and Representing Actions in Texts", in *Proceedings of the 2006 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE 2006)*, Vancouver, Canada, 16-21 July 2006, p.574-581.
- R. Khoury, F. Karray, and M. Kamel, "Extracting and Representing Actions in Text Using Possibility Theory", in *3rd Annual Scientific Conference of the LORNET Research Network (I^2LOR-06)*, Montreal, Canada, 9-11 November 2006.
- R. Khoury, F. Karray, and M. Kamel, "Keyword Extraction Rules Based on a Part-Of-Speech Hierarchy", in *4th Annual Scientific Conference of the LORNET Research Network (I^2LOR-07)*, Montreal, Canada, 4-7 November 2007.

Posters

- R. Khoury, Y. Sun, F. Karray, and M. Kamel, "Semantic Domain Classification by Means of Possibility Distributions", in *2nd Annual Scientific Conference of the LORNET Research Network (I^2LOR-05)*, Vancouver, Canada, 16-18 November 2005.

Bibliography

[1] K. Aas and L. Eikvil, "Text categorisation: a survey", Tech. Report No. 941, ISBN 82-539-0425-8, 1999.

[2] G. Akrivas and G. Stamou, "Fuzzy semantic association of multimedia document descriptions", in *Proceedings of International Workshop on Very Low Bitrate Video Coding (VLBV)*, October 2001.

[3] C. Apte, P. Damerau, S. Weiss, "Text mining with decision trees and decision rules", in *Proceedings of the Conference on Automated Learning and Discovery*, June 1998.

[4] H. Ayad and M. Kamel, "Finding natural clusters using multi-clusterer combiner based on shared nearest neighbours", in *Proceedings of the Fourth International Workshop on Multiple Classifier Systems MCS 2003*, Surrey, UK, 11-13 June 2003.

[5] J. Bai, J.-Y. Nie, and G. Cao, "Integrating Compound Terms in Bayesian Text Classification", in *Proceedings of the 2005 IEEE/WIC/ACM International Conference on Web Intelligence*, 19-22 September 2005, pp. 598-601.

[6] C. F. Baker, C. J. Fillmore and J. B. Lowe, "The Berkeley FrameNet project", in *Proceedings of the COLING-ACL*, Montreal, Canada, 1998.

[7] S. Banerjee and T. Pedersen, "An adapted Lesk algorithm for word sense disambiguation using WordNet", in *Proceedings of the Third International Conference on Intelligent Text Processing and Computational Linguistics (CICLING-02)*, February 2002, pp. 17–22.

[8] J. Barwise and J. Perry, "Situations and attitudes", Cambridge, Massachusetts, 1983.

[9] T. Berners-Lee, J. Hendler and O. Lassila, "The semantic web", in *Scientific American*, vol. 284, pp. 34–43, 2001.

[10] D. Blaheta and E. Charniak, "Assigning function tags to parsed text", in *Proceedings of the 1st Annual Meeting of the North American Chapter of the ACL (NAACL)*, Seattle, Washington, vol. 4, 2000, pp. 234-240.

[11] H. C. Boas, "Bilingual FrameNet dictionaries for machine translation", in *Proceedings of the Third International Conference on Language Resources and Evaluation (LREC-2002)*, M. Gonzalez Rodriguez and C. Paz Suarez Araujo (Eds.), Spain 2002, vol. IV, pp. 1364-1371.

[12] G. E. P. Box and G. C. Tiao, "Bayesian inference in statistical analysis", Massachusetts: Addison-Wesley, 1973.

[13] G. Brassard and P. Bratley, "Fundamentals of Algorithmics", Prentice Hall, Englewood Cliffs, NJ, 1996.

[14] R. Braz, R. Girju, V. Punyakanok, D. Roth, and M. Sammons, "An Inference Model for Semantic Entailment in Natural Language", in *AAAI'05*, 2005, pp. 1678-1679.

[15] E. Brill, "Transformation-based error-driven learning and natural language processing: A case study in part-of-speech tagging", in *Computational Linguistics*, vol. 21, pp. 543-565, 1995.

[16] E. Brill, "Unsupervised learning of disambiguation rules for part of speech tagging", in *Proceedings of the Third Workshop on Very Large Corpora (WVLC 3)*, 30 June 1995, pp. 1-13.

[17] A. Budanitsky and G. Hirst, "Evaluating WordNet-based Measures of Lexical Semantic Relatedness", in *Computational Linguistics*, vol. 32, issue 1, March 2006, pp. 13-47.

[18] M. Cary, "Toward optimal ε-approximate nearest neighbor algorithms", in the *Journal of Algorithms*, vol. 41, issue 2, Nov. 2001, pp. 417-428.

[19] C. Chelba, "Portability of syntactic structure for language modeling", in *Proceedings of the 2001 IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP '01)*, 7-11 May 2001, vol. 1, pp. a-d.

[20] A. Choi and M. Hatala, "Towards browsing distant metadata with semantic signatures", in *4th International Semantic Web Conference ISWC2005*, Poster Track PID-86, 6-10 November 2005, pp.168-169.

[21] A. Choi and M. Hatala, "Towards browsing distant metadata using semantic signature" in *Workshop on Integrating Ontologies at K-CAP 2005*, 2 October 2005, pp.10-17.

[22] S. Chua and N. Kulathuramaiyer, "Semantic feature selection using WordNet", in *Proceedings of the IEEE/WIC/ACM International Conference on Web Intelligence (WI 2004)*, 20-24 Sept. 2004, pp. 166-172.

[23] S. Clark and D. Weir, "Class-based probability estimation using a semantic hierarchy", in *Proceedings of the 2nd Conference of the North American Chapter of the ACL*, Pittsburgh, PA, 2001.

[24] R. A. Dara, M. Makrehchi and M. Kamel, "Data partitioning evaluation measures for classifier ensembles", in the *6th International Workshop on Multiple Classifier Systems*, California, USA, 13-15 June 2005.

[25] R. Du, R. Safavi-Naini, and W. Susilo, "Web filtering using text classification", in the *11th IEEE International Conference on Networks (ICON2003)*, 28 Sept.-1 Oct. 2003, pp. 325-330.

[26] D. Dubois and H. Prade, “Fuzzy sets and probability: Misunderstandings, bridges, and gaps,” in *Proceedings of the 2nd IEEE International Conference on Fuzzy Systems* (FUZZ-IEEE’93), vol. 2, San Francisco, CA, 1993, pp. 1059–1068.

[27] D. Dubois, J. Lang and H. Prade, “Possibilistic logic”, in *Handbook of Logic in Artificial Intelligence and Logic Programming*, D. M. Gabbay, C. J. Hogger and J. A. Robinson (Eds.), Oxford Science Publications, 1994, vol. 3, pp. 439-514.

[28] D. Dubois, “Possibility theory and statistical reasoning”, in *Computational Statistics & Data Analysis*, vol. 51, no. 1, pp. 47-69, 2006.

[29] D. Dubois and H. Prade, “Possibility theory: An approach to computerized processing of uncertainty”, New York: Plenum, 1988.

[30] C. J. Fillmore and C. F. Baker, “Frame Semantics for Text Understanding”, in *Proceedings of WordNet and Other Lexical Resources Workshop (NAACL)*, Pittsburgh, June 2001.

[31] C. J. Fillmore, C. F. Baker and H. Sato, “Seeing arguments through transparent structures”, in *Proceedings of the Third International Conference on Language Resources and Evaluation (LREC-2002)*, pp. 787-791, Las Palmas, Spain, 2002.

[32] W. N. Francis and H. Kučera, “Manual of information to accompany a standard corpus of present-day edited American English, for use with digital computers”, Department of Linguistics, Brown University, Providence, Rhode Island, 1964.

[33] M. A. Garzone, “Automated classification of citations using linguistic semantic grammars”, M.Sc. Thesis, University of Western Ontario, 1997.

[34] D. Gildea and D. Jurafsky, “Automatic labelling of semantic roles”, *in Proceedings of the 38th Annual Meeting on Association for Computational Linguistics*, Hong Kong, 2000, pp. 512-520.

[35] I. J. Good, “The population frequencies of species and the estimation of population parameters”, in *Biometrika*, vol. 40, pp. 237-264, 1953.

[36] K. Hammouda, D. Matute, and M. Kamel, “CorePhrase: Keyphrase extraction for document clustering”, in *the International Conference on Machine Learning and Data Mining in Pattern Recognition (MLDM 2005)*, the P. Perner and A. Imiya (Eds.), Springer Verlag, LNAI 3587, Leipzig, Germany, July 2005, pp. 265-274.

[37] X. He and C. DiMarco, “Using lexical chaining to rank protein-protein interactions in biomedical texts”, in *BioLink 2005: Workshop on Linking Biological Literature, Ontologies*

and Databases: Mining Biological Semantics, Conference of the Association for Computational Linguistics, Detroit, Michigan, June 2005.

[38] S. Hong-bo, S. Zhi-Hai, H. Hou-Kuanm, J. Li-Ping, "Text classification based on the TAN model", in *Proceedings of the IEEE Region 10 Conference on Computers, Communications, Control and Power Engineering*, vol. 1, 28-31 October 2002, pp 43-46.

[39] M. A. Hossain, Md. A. Rahman, A. El Saddik and P. Lévy, "Architecture for 3D navigation and authoring of distributed learning object repositories", in *Proceedings of the Third IEEE International Workshop on Haptic Virtual Environments and their Applications (HAVE2004)*, 2-3 October 2004.

[40] C. Hung; S. Wermter and P. Smith, "Hybrid neural document clustering using guided self-organization and WordNet", in *IEEE Intelligent Systems*, vol. 19, issue 2, pp. 68-77, 2004.

[41] Learning Technology Standards Committee of the IEEE, "Draft Standard for Learning Object Metadata", IEEE 1484.12.1-2002, New York, NY, 15 July 2002. http://ltsc.ieee.org/wg12/files/LOM_1484_12_1_v1_Final_Draft.pdf

[42] R. Iida, K. Inui, Y. Matsumoto "On the issue of combining anaphoricity determination and antecedent identification in anaphora resolution", in *Proceedings of 2005 IEEE International Conference on Natural Language Processing and Knowledge Engineering (IEEE NLP-KE '05)*, 30 October-1 November 2005, pp. 244-249.

[43] F. Jelinek and R. Mercer, "Probability distribution estimation from sparse data", in *IBM Technical Disclosure Bulletin*, vol. 28, pp. 2591-2594, 1985.

[44] G. Keswani, L. O. Hall, "Text classification with enhanced semi-supervised fuzzy clustering", in *Proceedings of the 2002 IEEE International Conference on Fuzzy Systems*, vol. 1, 12-17 May 2002, pp. 621-626.

[45] R. Khoury, F. Karray, Y. Sun, M. Kamel, and O. Basir, "Semantic understanding of general linguistic items by means of fuzzy set theory", in *IEEE Transactions on Fuzzy Sets and Systems*, to be published.

[46] P. Kingsbury and M. Palmer, "From Treebank to PropBank", in *Proceedings of the 3rd International Conference on Language Resources and Evaluation (LREC-2002)*, Las Palmas, Spain, 2002.

[47] R. Krovetz and W. B. Croft, "Lexical ambiguity and information retrieval", in *ACM Transactions on Information Systems (TOIS)*, vol. 10, no. 2, pp. 115–141, 1992.

[48] P. S. Laplace, "Philosophical essay on probabilities", Ney York: Springer-Verlag, 1995.

[49] W. Lehnert, “Symbolic/subsymbolic sentence analysis: Exploiting the best of two worlds”, in *Advances in Connectionist and Neural Computation Theory*, J. Barnden and J. Pollack (Eds.), New Jersey: Ablex Publishers, 1990, pp. 135–164.

[50] M. Lesk, “Automatic sense disambiguation using machine readable dictionaries: How to tell a pine cone from an ice cream cone”, in *Proceedings of SIGDOC*, pp. 24–26, 1986.

[51] G. J. Lidstone, “Note on the general case of the Bayes-Laplace formulae for inductive or *a posteriori* probabilities”, in *Transactions of the Faculty of Actuaries*, vol. 8, pp. 182-192, 1920.

[52] D. Lin, “Dependency-based Evaluation of MINIPAR”, *in Proceedings of The Evaluation of Parsing Systems: Workshop at the 1st International Conference on Language Resources and Evaluation (LREC’98)*, Granada, Spain, 28-30 May 1998.

[53] A. Lofti and A. C. Tsoi, “Importance of membership functions: A comparative study on different learning methods for fuzzy inference systems”, in *Proceedings of the Third IEEE Conference on Fuzzy Systems*, June 1994, vol. 3, pp. 1791–1796.

[54] R. E. Madsen, S. Sigurdsson, L. K. Hansen and J. Larsen, “Pruning the vocabulary for better context recognition”, in *Proceedings of the 2004 IEEE International Joint Conference on Neural Networks*, 25-29 July 2004, vol. 2, pp. 1439-1444.

[55] B. Mandelbrot, “Structure formelle des textes et communication”, *Word*, vol. 10, pp. 1-27, 1954.

[56] C. D. Manning and H. Schütze, “Foundations of Statistical Natural Language Processing”, MIT Press, Cambridge, MA, May 1999.

[57] M. Marcus, B. Santorini and M. A. Marcinkiewicz, “Building a large annotated corpus of English: the Penn Treebank”, in *Computational Linguistics*, vol. 19, no. 2, pp. 313-330, 1993.

[58] M. Marcus, “The Penn treebank: A revised corpus design for extracting predicate−argument structure”, in *Proceedings of the ARPA Human Language Technology Workshop*, Princeton, New Jersey, 1994.

[59] J. M. Mendel, “The perceptual computer: an architecture for computing with words”, in the *10th IEEE International Conference on Fuzzy Systems*, 2-5 Dec. 2001, vol. 1, pp. 35-38.

[60] J. M. Mendel, “Type-2 Fuzzy Sets and Systems: An Overview”, in *IEEE Computational Intelligence Magazine*, vol. 2, no. 1, 2007, pp. 20-29.

[61] R. Menon, S. S. Keerthi, H. T. Loh and A. C. Brombacher, “On the effectiveness of latent semantic analysis for the categorization of call centre records”, in *Proceedings of the 2004*

IEEE International Engineering Management Conference, 18-21 Oct. 2004, vol. 2, pp. 546-550.

[62] R.E. Mercer and C. DiMarco, "A design method for a biomedical literature indexing tool using the rhetoric of science", in the *2004 Joint Conference on Human Language Technology/North American Association for Computational Linguistics (HLT-NAACL)* Boston, USA, May 2004.

[63] I. Michael, "English grammatical categories and the tradition to 1800", Cambridge University Press, 1970.

[64] I. Navas-Delgado, N. Moreno-Vergara, A. C. Gomez-Lora, M. del Mar Roldan-Garcia, I. Ruiz-Mostazo and J. F. Aldana-Montes, "Embedding semantic annotations into dynamic Web contents", in *Proceedings of the 15th International Workshop on Database and Expert Systems Applications*, 30 Aug.-3 Sept. 2004, pp. 231-235.

[65] P. A. Pantel, "Clustering by Committee", Ph.D. thesis, Department of Computing Science, University of Alberta, 2003.

[66] J. Pearl, "Probabilistic reasoning in intelligent systems: Networks of plausible inference", Morgan Kaufmann, San Mateo, CA, 1988.

[67] Md. A. Rahman, M. A. Hossain, and A. El Saddik, "LORNAV: A demo of a virtual reality tool for navigation and authoring of learning object repositories", in *Proceedings of the 8th IEEE International Symposium on Distributed Simulation and Real Time Applications (DS-RT 2004)*, 21-23 October 2004, pp. 240-243.

[68] B. B. Rieger, "Semiotic cognitive information processing: Learning to understand discourse. A systemic model of meaning constitution" in *Perspectives on Adaptivity and Learning*, R. Kühn, R. Menzel, W. Menzel, U. Ratsch, M. M. Richter and I. O. Stamatescu (Eds.), Heidelberg/Berlin/New York: Springer, 2003, pp. 347-403,.

[69] B. B. Rieger, "On understanding understanding. Perception-based processing of NL texts in SCIP systems, or meaning constitution as visualized learning", in *IEEE Transactions on Systems, Man and Cybernetics, Part C*, vol. 34, issue 4, pp. 425-438, 2004.

[70] B. B. Rieger, "Distributed semantic representation of word meanings", *Parallelism, Learning, Evolution*, J. D Becker, I. Eisele and F. W. Mündemann (Eds.), UK: Springer-Verlag, 1991, pp. 243-273.

[71] J. Rocchio, “Relevance feedback information retrieval”, in *The Smart Retrieval System – Experiments in Automated Document Processing*, Gerald Salton, Ed. New Jersey: Prentice-Hall, 1971, pp. 313-323.

[72] S. Russel, P. Norvig, “Artificial intelligence: a modern approach”, Prentice Hall Series in Artificial Intelligence. Englewood Cliffs, New Jersey, 1995.

[73] J. J. Saade, H. B. Diab, “Defuzzification techniques for fuzzy controllers”, in *IEEE Transactions on Systems, Man and Cybernetics Part B*, vol. 30, issue 1, pp. 223-229, 2000.

[74] B. Santorini, “Part-of-speech tagging guidelines for the Penn Treebank Project”, Technical report MS-CIS-90-47, Department of Computer and Information Science, University of Pennsylvania.

[75] Canada's SchoolNet (2006) web site, [Online]. Available: http://www.schoolnet.ca/

[76] F. Sebastiani, “Machine learning in automated text categorization”, in *ACM Computing Surveys*, vol. 34, no. 1, 2002, pp. 1–47.

[77] H. H. Shahri and A. A. Z. Barforush, “Data mining for removing fuzzy duplicates using fuzzy inference”, in the *IEEE Annual Meeting of the North American Fuzzy Information Processing Society (NAFIPS '04)*, Alberta, Canada, 27-30 June 2004, vol. 1, pp. 419-424.

[78] M. Shamsfard and A. A. Barforoush, “The state of the art in ontology learning: a framework for comparison”, in *The Knowledge Engineering Review*, vol. 18 no. 4, 2003, pp. 293-316.

[79] S. Soderland, D. Fisher, J. Aseltine, and W. Lehnert, “CRYSTAL: Inducing a conceptual dictionary”, in *Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence*, Chris Mellish, Ed. San Francisco: Morgan Kaufmann, 1995, pp. 1314–1319.

[80] S. G. Soderland, “Learning text analysis rules for domain-specific natural language processing”, Tech. Rep. UM-CS-1996-087, Department of Computer Science, University of Massachusetts, 1996.

[81] V.-W. Soo, C.-Y. Lee, C.-C. Li, S.L. Chen and C.-C. Chen, “Automated semantic annotation and retrieval based on sharable ontology and case-based learning techniques”, in *Proceedings of the 2003 Joint Conference on Digital Libraries*, 27-31 May 2003, pp. 61-72.

[82] V.-W. Soo, S.-Y. Yang, S.-L. Chen and Y.-T. Fu, “Ontology acquisition and semantic retrieval from semantic annotated Chinese poetry”, in *Proceedings of the 2004 Joint ACM/IEEE Conference on Digital Libraries*, 7-11 June 2004, pp. 345-346.

[83] P. Subasic and A. Huettner, “Affect analysis of text using fuzzy semantic typing”, in *IEEE Transactions on Fuzzy Systems, Special Issue*, August 2001.

[84] Y. Sun, R. Khoury, F. Karray, and O. Basir, "Semantic context classification by means of fuzzy set theory", in *the 2005 IEEE International Conference on Natural Language Processing and Knowledge Engineering (IEEE NLP-KE 2005)*, Wuhan, China, 30 October – 1 November 2005, pp. 250-255.

[85] Y. Sun, "Fuzzy Methodology for Enhancement of Context Understanding", Ph.D. Thesis, University of Waterloo, 2005.

[86] P. D. Turney, "Learning algorithms for keyphrase extraction", in *Information Retrieval*, vol. 2, 2000, pp. 303-336.

[87] P. Turney, "Word Sense Disambiguation by Web Mining", *in Proceedings of the Third International Workshop on the Evaluation of Systems for the Semantic Analysis of Text (SENSEVAL-3)*, Barcelona, Spain, 25-26 July 2004, pp. 239-242.

[88] H. Uejima, T. Miura and I. Shioya, "Improving text categorization by resolving semantic ambiguity" in *2003 IEEE Pacific Rim Conference on Communications, Computers and signal Processing (PACRIM)*, 28-30 Aug. 2003, vol. 2, pp. 796-799.

[89] P. Wang, B.-W. Xu, J.-J. Lu, D.-Z. Kang and Y.-H. Li, "A novel approach to semantic annotation based on multi-ontologies", in *Proceedings of the 2004 International Conference on Machine Learning and Cybernetics*, 26-29 Aug. 2004, vol. 3, pp. 1452-1457.

[90] F. Wang, "Parsing "grammatically incomplete" natural language queries to spatial databases", in the *Joint 9th IFSA World Congress and 20th NAFIPS International Conference*, Vancouver, Canada, 25-28 July 2001, vol. 4, pp. 2400-2404.

[91] B. Wang and S. Zhang, "A Novel Text Classification Algorithm Based on Naïve Bayes and KL-Divergence", in the *Proceedings of the Sixth International Conference on Parallel and Distributed Computing, Applications and Technologies (PDCAT 2005)*, 05-08 December 2005, pp. 913-915.

[92] L.-X. Wang, J.-M. Han, Z. Wei, and G.-C. Zhou, "Application of Layered Clustering and Plane Partition in Web Page Classification", in *Proceedings of 2005 International Conference on Machine Learning and Cybernetics*, 18-21 August 2005, vol. 4, pp. 2325-2330.

[93] Y. Wang and X.-J. Wang, "A New Approach to Feature Selection in Text Classification", in *Proceedings of 2005 International Conference on Machine Learning and Cybernetics*, 18-21 August 2005, vol. 6, pp. 3814-3819.

[94] D. H. Widyantoro, J. Yen, "A fuzzy similarity approach in text classification task", in *The Ninth IEEE International Conference on Fuzzy Systems*, vol.2, 7-10 May 2000, pp. 653-658.

[95] G. J. Wilms, “Using an on-line dictionary to extract a list of sense-disambiguated synonyms”, in *Proceedings of the 30th annual Southeast regional conference*, ACM Press, pp. 15–22, 1992.

[96] L. Wittgenstein, “The blue and brown books”, R. Rhees, Ed., Oxford: Harper Perennial, 1958.

[97] O. Wu and W. Hu, “Web sensitive text filtering by combining semantics and statistics”, in *Proceedings of 2005 IEEE International Conference on Natural Language Processing and Knowledge Engineering (IEEE NLP-KE '05)*, 30 Oct.-1 Nov. 2005, pp. 663-667.

[98] Z. Xiaohui, W. Huayong, L. Ying, C. Guiran and Z. Hong, “Dynamic vector-space model for internet textual information categorization”, in the *2002 IEEE International Conference on Systems, Man and Cybernetics*, 6-9 Oct. 2002, vol. 2, pp. 449-454.

[99] T. Yoshioka, Y. Takata, M. Ito and S. Ishii, “A neural visualization method for WWW document clusters”, in *Proceedings of the International Joint Conference on Neural Networks (IJCNN '01)*, 15-19 July 2001, vol. 3, pp. 2270-2275.

[100] F. Yuan, L. Yang, and G. Yu, “Improving the k-NN and applying it to Chinese text classification”, in *Proceedings of the 2005 International Conference on Machine Learning and Cybernetics*, 18-21 August 2005, vol. 3, pp. 1547-1553.

[101] L. A. Zadeh, “Toward a restructuring of the foundations of fuzzy logic (FL)” in *The 1998 IEEE International Conference on Fuzzy Systems Proceedings*, 4-9 May 1998, vol. 2, pp. 1676-1677.

[102] L. A. Zadeh, “The roles of soft computing and fuzzy logic in the conception, design and deployment of intelligent system”, in *IEEE Asia Pacific Conference on Circuits and Systems*, 18-21 November 1996, pp. 3-4.

[103] L. A. Zadeh, “A fuzzy-set-theoretic interpretation of linguistic hedges”, in *Journal of Cybernetics*, vol. 2, pp. 4–34, 1972.

[104] L. A. Zadeh, “Quantitative fuzzy semantics”, in *Information Sciences*, vol. 3, pp. 159–176, 1971.

[105] L. A. Zadeh, “Pruf and its application to inference from fuzzy propositions”, in *Proceedings of the IEEE Conference on Decision Control*, 1977, pp. 1359–1360.

[106] L. A. Zadeh, “Pruf–a meaning representation language for natural language”, in *Fuzzy Reasoning and its Applications*, pp.1–66, 1981.

[107] L. A. Zadeh, “Toward a theory of fuzzy information granulation and its centrality in human reasoning and fuzzy logic”, in *Fuzzy Sets and Systems*, Volume 90 Issue 2, pp. 111-127, 1997.

[108] L. A. Zadeh, “Outline of a computational approach to meaning and knowledge representation based on a concept of a generalized assignment statement”, in *Proceedings of the Interational Seminar on Artificial Intelligence and Man-Machine Systems*, M. Thoma and A. Wyner (Eds.), Springer Heidelberg, 1986, pp. 198-211.

[109] L. A. Zadeh, “Precisiated natural language (PNL) - Toward an enlargement of the role of natural languages in scientific theories”, in *Proceedings of the 2004 IEEE International Conference on Fuzzy Systems*, 25-29 July 2004, vol. 1, pp. 1-2.

[110] L. A. Zadeh. “Fuzzy sets as a basis for a theory of possibility”, in *Fuzzy Sets and System*, vol. 1, pp. 3-28, 1978. Reprinted in *Fuzzy Sets and Systems*, vol. 100, supplement 1, pp. 9-34, 1999.

[111] S. Zhou, T. W. Ling, J. Guan, J. Hu and A. Zhou, “Fast text classification: a training-corpus pruning based approach”, in *Proceedings of the Eight International Conference on Database Systems for Advanced Applications*, 26-28 March 2003, pp. 127-136.

[112] T.-D. Zhu, X.-X. Zhao, and Y.-S. Liu, “A new text classification model based on the sentence space”, in *Proceedings of 2005 International Conference on Machine Learning and Cybernetics*, 18-21 August 2005, vol. 3, pp. 1774-1777.

[113] G. K. Zipf, “Human behaviour and the principle of least effort”, Massachusetts: Addison-Wesley, 1949.