UNSUPERVISED INDUCTION OF SEMANTIC CLASSES USING SEMANTIC SIMILARITY METRICS

By
Elias M. Iosif

SUBMITTED IN PARTIAL FULFILLMENT OF THE
REQUIREMENTS FOR THE DEGREE OF
MASTER OF SCIENCE
AT
TECHNICAL UNIVERSITY OF CRETE
CHANIA, GREECE
JULY 2007

TECHNICAL UNIVERSITY OF CRETE
DEPARTMENT OF
ELECTRONICS AND COMPUTER ENGINEERING

The undersigned hereby certify that they have read and recommend to the Faculty of Graduate Studies for acceptance a thesis entitled "**Unsupervised Induction of Semantic Classes using Semantic Similarity Metrics**" by **Elias M. Iosif** in partial fulfillment of the requirements for the degree of **Master of Science**.

Dated: July 2007

Supervisor: _____
Assoc. Prof. Alexandros Potamianos

Readers: _____
Prof. Vasilis Digalakis

_____
Assoc. Prof. Evripidis Petrakis

TECHNICAL UNIVERSITY OF CRETE

Date: **July 2007**

Author: **Elias M. Iosif**

Title: **Unsupervised Induction of Semantic Classes using Semantic Similarity Metrics**

Department: **Electronics and Computer Engineering**

Degree: **M.Sc.** Convocation: **July** Year: **2007**

Signature of Author

To my mother, Maria

Table of Contents

List of Tables

List of Figures

Abstract

The unsupervised learning of semantics from text is the "holy grail" of Natural Language Processing. Many applications dealing with textual information require classification of words into semantic classes including dialogue systems, language modeling as well as semantic web.

In this thesis, we investigate several similarity metrics for semantic similarity computation between words and induction of semantic classes. All of the proposed algorithms are fully automatic, without using any manually-crafted knowledge source. Thus, they are unsupervised and language independent.

First, we propose unsupervised algorithms for combining semantic similarity metrics for the task of automatic class induction. The semantic similarity metrics that are evaluated and combined are based on narrow- and wide-context cosine similarity. The metrics are combined using linear weights that are computed 'on the fly' and are updated at each iteration of the class induction algorithm, forming a corpus-independent metric. The proposed algorithms are evaluated on two corpora: a semantically heterogeneous news domain (HR-Net) and an application-specific travel reservation corpus (ATIS). It is shown, that the (unsupervised) adaptive weighting scheme outperforms the (supervised) fixed weighting scheme. Up to 50% relative error reduction is achieved by the adaptive weighting scheme.

In our second approach, we propose a soft-decision, unsupervised clustering algorithm that generates semantic classes automatically using the probability of class membership for each word, rather than deterministically assigning a word to a semantic class. Semantic classes are induced using an unsupervised, automatic procedure that uses a context-based similarity distance to measure semantic similarity between words. The proposed soft-decision algorithm is compared with various "hard" clustering algorithms and it is shown to improve semantic class induction performance in terms of both precision and recall for a travel reservation corpus.

Lastly, we propose two novel web-based metrics for semantic similarity computation between words. The first metric considers only the page counts returned by a search engine. The second metric downloads a number of the top ranked documents and applies "wide-context" and "narrow-context" metrics. The proposed metrics work automatically, without consulting any external knowledge resource. The metrics are compared with WordNet-based methods. The metrics' performance is evaluated in terms of correlation with respect to the pairs of the commonly used Charles-Miller dataset. The proposed "wide-context" metric achieves 71% correlation, which is the highest score achieved among the fully unsupervised metrics in the literature up to

date.

Acknowledgements

I feel that I am incredibly fortunate for working under the supervision of professor Alexandros Potamianos for a second time! The first time was for my diploma thesis. I would like to express my sincere gratitude to my advisor for his valuable guidance. He provided with true generosity all the "inspiration resources" upon which this research effort is built.

I would also like to thank professors Vasilis Digalakis and Evripidis Petrakis for participating in the examining committee.

Furthermore, the fruitful discussions that I had with professor Evripidis Petrakis and Kelly Zervanou during the meetings of the Intelligent Systems Lab, helped me to become familiar with research fields that are related with my work with a broader sense.

The research described in this thesis was partially supported by the EU-IST-FP6 MUSCLE Network of Excellence.

I could not forget to thank my family, especially my mother: Maria this thesis is dedicated to you.

Introduction

The unsupervised learning of semantics from text is the "holy grail" of Natural Language Processing, contributing to the natural language understanding task. Many applications dealing with textual information require classification of words into semantic classes including dialogue systems, language modeling, speech understanding and machine translation, as well as web applications. For example, a natural language understanding module, embedded in dialogue system, requires knowledge of semantic classes in order to semantically parse the transcribed audio stream. In the field of information retrieval, semantic classes can be used for semantic document representation and query expansion. For example, during a web search the submitted query is often semantically reformed by the user for getting a new set of results in order to access more relevant documents than in the initial query. Usually the refined query contains words that are semantically close to the words of the initial query. In query expansion the addition of semantically related words can increase the relevant retrieved documents, which in other case may be missed. This introduces the idea of semantic representation and retrieval instead of simplistic lexical string matching. Generally, query expansion is able to increase the recall of the retrieved documents.

Statistical language modeling attempts to capture regularities of natural language using text processing. An example of such regularity is the frequent co-occurrence of two words. This phenomenon is well modeled through a statistical approach that gives a measurement to the event of co-occurrence. Furthermore, this embodies an abstract relationship about the two words. Their co-occurrence reflects a kind of association between them. From this point of view each word contains an amount of information about the other words of its lexical environment. Moving to a higher level, we can say that a word has some conceptual relationships with the words of its context.

The contextual computational model of semantics is referred as the word-space model. A model that measures the semantic relationship between words is defined with respect to the vocabulary which forms a high-dimensional space whereas each word can be considered as one dimension. The word-space model reflects a spatial representation of word meaning. The key idea of this model is that semantic similarity can be represented as proximity in n-dimensional space, where n is the cardinality of vocabulary set. Spatial proximity between words as a representation of their semantic similarity seems to be very intuitive and naturally derived with respect to the way that human conceptualize similarities. The distributional (contextual) hypothesis of meaning assumes that words with similar distributional properties have similar meaning. Statistical methods can learn the distributional properties of words which can be

employed into similarity measures in order to compute similarity scores between words and derive semantic classes.

Manual construction of semantic classes is a time-consuming task and often requires costly expert knowledge. Furthermore, semantics are sensitive to domain changes. The meaning of a word changes as the word is used in different domains with different ways. An automatic or semi-automatic algorithm for extracting semantic classes from text leads to the rapid development of many natural language processing systems, capturing the specific semantics for the domain of interest.

Many approaches have been proposed in the literature that perform text mining in order to capture semantic information and induce semantic classes. A number of methods are applied on text corpora, such as collections of news articles. More recent methods exploit the world wide web, which is an extremely large multilingual source of textual information with an increasing rate of growth. Both types of the above approaches can supervised or unsupervised. The notion of supervision has the sense that human knowledge is incorporated in the procedure of semantics' acquirement. An example of a supervised technique is the use of annotated corpora where the contained entries are tagged linguistic properties. Also, many methods apply human corrections that are given as feedback to the system. Furthermore, manually-crafted knowledge resources, such as dictionaries, thesauri and ontologies, are used by various methods.

In this thesis, we adopt the unsupervised approach in order to capture semantic information. In particular, the experimental corpora are unrestricted (raw text) without any kind of encoded linguistic information. Using this tactic, no knowledge resources are used. The proposed algorithms work automatically, thus, they are language independent.

The research effort of this thesis can be divided in three branches:

1. We use the assumption that similarity of context implies similarity of meaning. Using this hypothesis, we apply several statistical contextual similarity metrics in order to compute semantic similarity between words. These metrics embody different ranges of contextual scopes. "Narrow-context" and "wide-context" metrics are defined and linearly combined, using an adaptive weighting scheme, throughout an iterative procedure. The idea behind metric combination is that each individual metric serves a classifier with its own advantages, so the combination of different classifier is able to lead to more robust and qualitative results. In other words, our motivation for creating an adaptive weighting scheme, is that the relative performance of each metric varies from iteration to iteration. It is expected that by updating the weights of each metric at each iteration the combined metric performance can significantly improve. In addition, our goal is to create a fully unsupervised, corpus-independent metric combination algorithm that does not require experiment on held-out data to compute the weights. Semantic classes are generated according to the computed similarity scores. This work is discussed in *Chapter 3*.

2. Using the distributional hypothesis of meaning, as before, we propose a soft-clustering algorithm. Words are allowed to be assigned to more than one semantic class, via a probabilistic membership function. Furthermore, each categorized

word retains its lexical type. This is motivated by the idea that the presence of lexical and semantic information can contribute to the better estimation of probabilities, in order to automatically induce semantic classes. Also, this approach alleviates the drawbacks of hard-clustering where (a) multiple semantic categorizations for words are ignored, (b) erroneous word assignments are retained throughout the iterative procedure, (c) the lexical type of categorized words are not considered because the words are substituted by a class labels, and (d) generated classes containing nonterminals tend to form overgeneralized concepts. This work is presented in *Chapter 4*.

3. Lastly, we exploit the world wide web using search engines in order to compute semantic similarity scores. The core motivation of this approach is that the web does not suffer from data sparseness, so similarity between words can be better estimated. Similarity scores are computed using (i) hit counts for queries submitted to web search engines, and (ii) by downloading a number of top ranked documents and applying the above similarity metrics. *Chapter 5* deals with this work.

The rest of the thesis is organized as follows: Chapter 1 introduces some basic material on N-gram statistical language modeling, since the N-gram probabilities are used in the "narrow-context" metrics. Chapter 2 describes other metrics that estimation of semantic similarity between words. Some of them are used as "wide-context" metrics. The rest chapters deal with the research contribution of this thesis. Chapter 3 presents the automatic combination of similarity metrics. The soft-clustering algorithm is discussed in Chapter 4. Chapter 5 describes the web-based similarity metrics. Chapter 6 summarizes the research conclusions of this thesis and outlines interesting directions for future work.

Chapter 1

N-gram Statistical Language Modeling

1.1 Introduction

This chapter introduces some basic material on N-gram statistical language modeling. Statistical language modeling tries to acquire regularities of natural language using corpora processing [95]. Corpora are collections of text, e.g., news articles, scientific papers, transcribed dialogues; they can be considered as informative resources. The statistical approach of text processing is naturally being used by researchers due to the categorical character of language and the large vocabularies, in order to estimate numerous parameters [95].

It is generally accepted that words of natural languages have meaning and that the meaning of a sentence is formed by its words [64]. A word can be viewed as a corpus-related event which whenever it occurs, it transmits a message. One of the various interpretations for the lemma "word", given by Oxford Dictionary, is "intelligence, a message" [25]. The notion of "word" in linguistics is denoted by the term "lexeme" [64] and it is the minimal unit of language, which has one or more semantic interpretations. A word exists with other words and these units build more larger comprehensive units: phrases, sentences, paragraphs, etc. For example, "bank" is a lexeme, while "bank financial institution" and "bank-edge of a river" consist of lexemes. It is reasonable to claim that a word preserves a kind of conceptual relationship with its neighboring words, in some way. From this point of view each word contains an amount of information about the other words of its' lexical environment. We can say that the occurrence of a word is dependent to the context words . This assumption was advocated as "a word is characterized by the company it keeps" by J.R. Firth and it is known in linguistics as the *distributional hypothesis*. Thus, it is possible to utilize the surface statistics of language in order to proceed to a deeper level.

1.2 Language modeling

In few words, a language model gives the probability $P(s)$ of a sentence s. Very often the domain of statistical language modeling overlaps the domain of *computational*

linguistics. A way to discriminate the two fields in terms of language modeling is the following. Let S be a word sequence (i.e., surface form) and let M be some underlying structure (i.e., semantics) related with it. In general, statistical language modeling estimates $P(S)$, while computational linguistics deals with the estimation of $P(S \mid M)$ [95].

The first significant use of language modeling was made in 1980. Since then many techniques of language modeling have been proposed [43]. The majority of the language models decomposes the sentence probability, $P(s)$, into a product of conditional probabilities

$$P(s) = P(w_1 \dots w_n) = \prod_{i=1}^{N} P(w_i|h_i) \tag{1.2.1}$$

where w_i is the i^{th} word in the sentence and $h_i = \{w_1, w_2, \dots, w_{i-1}\}$ is the sequence of preceding words (the *history* of w_i).

During the last three decades many language modeling techniques have been proposed in the literature. In [95], many major established approaches are discussed, and the thorough work of Goodman [30] makes a comparative study of them.

In the next section we present the popular N-gram language model, being used in a variety of commercial and research systems.

1.3 N-gram language modeling

A word sequence is useful to be modeled for a variety of reasons. For example, in many cases one may want to predict the next word in a sequence. The prediction becomes applicable when it can be measured. A probabilistic language model treats words as events, distributing to them a probability mass. Furthermore, as we will see, it is possible to estimate the probability even for a completely unknown word; imagine a neologism that will be "invented" by the future generations!

The simplest language probabilistic model let any word to follow any other word with equal probability. For example, if the vocabulary of a certain natural language consists of 75.000 unique words, then the probability of any word following any other word equals to $\frac{1}{75.000}$. A more complex language model uses the frequency of occurrence of a word. For example, the previous paragraph has totally 81 words, in which the words "word" and "a" occur 3 and 5 times, respectively. According to the simple language model the words "word" and "a" have $\frac{3}{81}$ and $\frac{5}{81}$ probability, respectively, to follow any word. But for the sequence "to predict the next", the word "word" is more reasonable than "a" to follow "next". This intuitive observation considers the conditional probability of a word given the previous word, instead of using the relative word frequency.

1.3.1 General

The N-gram language model considers the language as a Markov process of order $N-1$.

$$P(w_i|h_i) = P(w_i \mid w_{i-N+1}, \dots, w_{i-1}) \approx P(w_i \mid w_{i-N+1}^{i-1}) \tag{1.3.1}$$

Equation 1.3.1 states that the probability of word w_i given all the previous words of the sentence can be approximated by the probability given only the previous $N-1$ words.

N-gram probabilities are computed by counting and normalizing the N-gram occurrences. For the bigram case the conditional probability of word w_{i-1} given that it is followed by word w_i is computed as

$$P(w_i|w_{i-1}) = \frac{C(w_{i-1}w_i)}{\sum_w C(w_{i-1}w)} = \frac{C(w_{i-1}w_i)}{C(w_{i-1})} \tag{1.3.2}$$

Equation 1.3.2 takes the count of $w_{i-1}w_i$ bigram and divides it by the sum of all bigrams that have w_{i-1} as first word. Note that the latter sum is equal to the count of w_{i-1} unigram. For the general case of N-gram model the above equation is written as

$$P(w_i|w_{i-N+1}^{i-1}) = \frac{C(w_{i-N+1}^{i-1}w_i)}{C(w_{i-N+1}^{i-1})} \tag{1.3.3}$$

Equations 1.3.2 and 1.3.3 use the frequency interpretation of probability [78], applying the technique of Maximum Likelihood Estimation (*MLE*). Even with large corpora many N-grams occur only once or they have low counts, so, the computation of N-gram probabilities remains a sparse estimation problem. This fact is prescient with Chomsky's observation that a model suffering from lack of data assigns low probability to a phrase regardless its sensical or grammatical correctness [59]. Thus, it is preferable not to apply *MLE* of N-gram probabilities in a straightforward way, based on counts. Instead, several smoothing approaches [14] can be used in order to smooth the *ML* estimates.

1.3.2 Smoothing

The N-gram models are trained from corpora. In practice, every training corpus is of finite size, so, naturally some acceptable N-grams are bound to be absent. This intrinsic characteristic of corpora leads to zero and low counts of N-grams. Using the *MLE* approach the absent N-grams are assigned zero probability, while the probabilities of low-count N-grams are underestimated.

Consider the sentence "put language back into language modeling" [1]. If the bigram "put language" has never occurred in the training corpus, then

$$P(language \mid put) = \frac{C(put\ language)}{\sum_w C(put\ w)} = \frac{0}{a}, \qquad a > 0 \tag{1.3.4}$$

The probability of sentence $P(put\ language \ldots modeling) = 0$. Clearly, this is an underestimate for the sentence probability, since in real life there is an "amount" of probability by which the sentence is likely to occur.

Smoothing battle the problem of data sparseness by re-evaluating the zero- and low-probabilities and assigning them non-zero values. The name of this strategy describes

[1] This statement belongs to F. Jelinek, suggesting that the statistical natural language techniques should also take into consideration the deep structure of language [44].

what is actually happens. Smoothing techniques make the probability distributions more uniform: adjust low probabilities upward and high probabilities downward [14]. Next, we briefly survey some of the most widely-used smoothing strategies in order to outline the underlying ideas.

Additive smoothing

This is a simplistic technique of smoothing, since it pretends that an N-gram occurs δ times more than it does, where $0 < \delta \leq 1$ [61, 47, 42]. For example, for the bigram case we have [14]

$$P_{Add}(w_i \mid w_{i-1}) = \frac{\delta + C(w_{i-1}\ w_i)}{\delta \mid V \mid + \sum_{w_i} C(w_{i-1}\ w_i)} \tag{1.3.5}$$

where set V is the vocabulary of the training corpus and $\mid V \mid$ denotes the cardinality of V. In general, the additive smoothing has poor performance [26, 27].

Good-Turing estimate

The key idea of Good-Turing smoothing is the exploration of N-grams of high counts in order to re-estimate the amount of probability mass that is to be given to N-grams with zero or low counts [29]. The Good-Turing estimate feigns that for any N-gram that occurs r times we can feign that it occurs $r^\star$ times:

$$r^\star = (r+1)\frac{k_{r+1}}{k_r} \tag{1.3.6}$$

where k_{r+1} and k_r is the number of N-grams that occur exactly $r+1$ and r times, respectively. For instance, if a bigram occurs r times, the corresponding probability is

$$P_{GT}(w_i \mid w_{i-1}) = \frac{r^\star}{\sum_{\infty}^{r=1} r k_r} \tag{1.3.7}$$

In particular, the Good-Turing estimate is applied as stand alone N-gram smoothing approach, because does not combine hign- and low-order models that obtain better performance [14].

Deleted interpolation

It is fruitful to interpolate higher-order N-gram models with lower-order N-gram models, because there are cases whereas there is no sufficient data to compute probabilities for the higher-order models [45]. So, the lower-order models is more trustworthy, providing supplementary useful information.

$$P_{DelInt}(w_i|w_{i-N+1}^{i-1}) = \lambda_1(w_{i-N+1}^{i-1})P_{ML}(w_i|w_{i-N+1}^{i-1}) + \lambda_2(w_{i-N+1}^{i-1})P_{DelInt}(w_i|w_{i-N+2}^{i-1}) \tag{1.3.8}$$

The smoothed model of Equation 1.3.8 use recursion as interpolates linearly an N^{th}-order model estimated with maximum likelihood and an N^{th-1}-order smoothed model [10]. Note that the λ weights sum to 1:

$$\sum_i \lambda_i = 1 \tag{1.3.9}$$

Each λ value is a function of the context. The optimal $\lambda_(w_{i-N+1}^{i-1})$ is different for different histories. For example, a context that have been occurred for many times should be given a high weight since its' distribution tends to be reliable. In contrast, for a history of low frequency, a lower λ weight will be reasonable. For the training of the λ parameters many approaches have been proposed in the literature [45, 2, 13, 14].

Witten-Bell smoothing

The Witten-Bell smoothing can be considered as an instance of deleted interpolation [110]. [2] As Equation 1.3.8, the N^{th}-order maximum likelihood model is linearly interpolated with the N^{th-1}-order smoothed model:

$$P_{WB}(w_i|w_{i-N+1}^{i-1}) = \lambda(w_{i-N+1}^{i-1})P_{ML}(w_i|w_{i-N+1}^{i-1}) + (1 - \lambda(w_{i-N+1}^{i-1}))P_{WB}(w_i|w_{i-N+2}^{i-1}) \tag{1.3.10}$$

The computation of $\lambda(w_{i-N+1}^{i-1})$ parameter requires the number of unique words that follow the history w_{i-N+1}^{i-1}. This number is denoted as $U_{1+}(w_{i-N+1}^{i-1}\bullet)$. Using a more formal notation we write [14]

$$U_{1+}(w_{i-N+1}^{i-1}\bullet) =\mid \{w_i : C(w_{i-N+1}^{i-1}w_i) > 0\} \mid \tag{1.3.11}$$

The number of words that occur one or more times (1+) are denoted by U_{1+}. The symbol $\bullet$ is used for a free variable (in our case, word) that is summed over. The parameter $\lambda(w_{i-N+1}^{i-1})$ is calculated as

$$\lambda(w_{i-N+1}^{i-1}) = 1 - \frac{U_{1+}(w_{i-N+1}^{i-1}\bullet)}{U_{1+}(w_{i-N+1}^{i-1}\bullet) + \sum_{w_i} C(w_{i-N+1}^{i})} \tag{1.3.12}$$

Substituting Equation 1.3.12 into Equation 1.3.10, we have

$$P_{WB}(w_i|w_{i-N+1}^{i-1}) = \frac{C(w_{i-N+1}^{i}) + U_{1+}(w_{i-N+1}^{i-1}\bullet)P_{WB}(w_i|w_{i-N+2}^{i-1})}{\sum_{w_i} C(w_{i-N+1}^{i}) + U_{1+}(w_{i-N+1}^{i-1}\bullet)} \tag{1.3.13}$$

According to Equation 1.3.10, we use the higher-order model with probability $\lambda(w_{i-N+1}^{i-1})$, while the lower-order model is used with $1 - \lambda(w_{i-N+1}^{i-1})$ probability. The probability mass that equals to the $1 - \lambda(w_{i-N+1}^{i-1})$ probability, is the probability of a word that is not observed immediately after the w_{i-N+1}^{i-1} history in the training data, but appears in any later position.

Absolute smoothing

In absolute smoothing [74] a higher-order model is interpolated with a lower-order model. However, instead of multiplying the higher-order distribution by a factor $\lambda(_{i-N+1}^{i-1})$, the higher-order distribution is derived by subtracting a fixed discount $D \leqslant 1$ from each non-zero count.

$$P_{Abs}(w_i|w_{i-N+1}^{i-1}) = \frac{max\{C(w_{i-N+1}^{i}) - D, 0\}}{\sum_{w_i} C(w_{i-N+1}^{i})} + (1 - \lambda(w_{i-N+1}^{i-1}))P_{Abs}(w_i|w_{i-N+2}^{i-1}) \tag{1.3.14}$$

[2] This method was introduced as *M*ethod C in [110].

In order to make the distribution sum to 1, the $\lambda(w_{i-N+1}^{i-1})$ factor is computed as

$$\lambda(w_{i-N+1}^{i-1}) = 1 - \frac{D}{\sum_{w_i} C(w_{i-N+1}^{i})} U_{1+}(w_{i-N+1}^{i-1}\bullet) \tag{1.3.15}$$

In [74], a suggested value for D is

$$D = \frac{n_1}{n_1 + 2n_2} \tag{1.3.16}$$

where n_1 and n_2 are the total number of N-grams of the higher-order distribution with exactly one and two counts, respectively.

Kneser-Ney smoothing

In Kneser-Ney smoothing [53] the higher-order model is interpolated with a lower-order model and the higher distribution is discounted as in absolute smoothing. The difference between absolute and Kneser-Ney smoothing is in the lower-order distribution. In Kneser-Ney method, the lower-order distribution is proportional to the number of different words that it follows. Consider for example a language model trained over a corpus about computer industry and the word "Packard". If the frequency of this word is high, then the *MLE* of the unigram probability will, also, be high. The idea of Kneser-Ney smoothing is that the unigram probability of word "Packard"must be low, since it folows only one different word, "Packard".

$$\begin{aligned} P_{KN}(w_i|w_{i-N+1}^{i-1}) = & \frac{max\{C(w_{i-N+1}^{i}) - D, 0\}}{\sum_{w_i} C(w_{i-N+1}^{i})} + \\ & + \frac{D}{\sum_{w_i} C(w_{i-N+1}^{i})} U_{1+}(w_{i-N+1}^{i-1}\bullet) P_{KN}(w_i|w_{i-N+2}^{i-1}) \end{aligned} \tag{1.3.17}$$

In order to make the distribution sum to 1, we take

$$P_{KN}(w_i|w_{i-N+2}^{i-1}) = \frac{U_{1+}(\bullet w_{i-N+2}^{i})}{U_{1+}(\bullet w_{i-N+2}^{i-1}\bullet)} = \frac{\mid \{w_{i-N+1} : C(w_{i-N+1}^{i}) > 0\} \mid}{\mid \{w_{i-N+1}, w_i : C(w_{i-N+1}^{i}) > 0\} \mid} \tag{1.3.18}$$

1.3.3 Backoff

One main contribution of the discussed smoothing methods is the solution of the problem caused by the zero-count N-grams. Moreover, there is another methodology that tackle this problem. Suppose that there are no occurrences of a particular trigram, $w_{i-2}\ w_{i-1}\ w_i$, in the training corpus. In this case we can estimate the trigram probability $P(w_i \mid w_{i-2}\ w_{i-1})$ using the bigram probability $P(w_i \mid w_{i-1})$. In the same manner, if there are no counts of the bigram $w_{i-1}\ w_i$, we can estimate $P(w_i \mid w_{i-1})$ using the unigram probability $P(w_i)$. This strategy is called *b*ackoff. According to the above description of backoff method, an amount of probability mass is taken away from the higher-order models and is distributed to the lower-order models [65, 48, 43]. Of course, the resulted probability estimation must remain valid, i.e., sums to one.

The backoff model was introduced by Katz [50] and is similar to the deleted interpolation in the sense that the construction of an N-gram model is based on an N-1 model. The difference between backoff and deleted interpolation is that in backoff, for example, if there are non-zero frequency trigarm, we use only these counts without interpolating the bigram and unigram models [48]. The "back off" step downwards to a lower-order model is followed if there are zero counts for the higher-order model [3].

For a trigram language model, the backoff method is defined as [48]

$$P(w_i \mid w_{i-2}\ w_{i-1}) = \begin{cases} P_{ML}(w_i \mid w_{i-2}\ w_{i-1}), & \text{if} \quad C(w_{i-2}\ w_{i-1}\ w_i) > 0 \\ \alpha_1 P_{ML}(w_i \mid w_{i-1}), & \text{if} \quad C(w_{i-2}\ w_{i-1}\ w_i) = 0 \\ & \quad \text{and } C(w_{i-1}\ w_i) > 0 \\ \alpha_2 P_{ML}(w_i), & \text{otherwise} \end{cases} \tag{1.3.19}$$

Some smoothing techniques assume that the unseen N-grams are all equally probable and an amount of probability mass is distributed ti them according to an even scheme. A more neat and fair way is to combine smoothing with backoff for distributing the probability mass to the unseen events. The smoothing quantifies the total mass of probability that must be reserved for the unseen events and the backoff procedure defines how to assign the reserved probability.

Let's consider Equation 1.3.19. The presence of α parameters ensures that the computed probability is a valid probability. This can be explained as follows. If the frequency of the trigram of interest is non-zero, then the $P_{ML}(w_i \mid w_{i-2}\ w_{i-1})$ probability that is computed over relative frequencies is a true probability. Otherwise, we have to back off to a lower-order model, and, then, we will add extra probability mass, resulting to a non-true probability. So, the backoff model must be smoothed. Using these considerations, the $P_{ML}(\cdot)$ probabilities of Equation 1.3.19 must be substituted by smoothed probabilities $\tilde{P}(\cdot)$. The use of smoothing saves an amount of probability mass for the lower-order models. Moreover, the α parameters guarantee that the sum of the distributed (to the lower-order models) portions of probability mass is equal to the initially saved amount of probability [48]. In the general N-gram case, the probability mass that must be given form an N-gram to an N-1-gram is defined as follows [48].

$$\alpha(w_{i-N+1}^{i-1}) = \frac{1 - \sum_{w_i : C(w_{i-N+1}^{i}) > 0} \tilde{P}(w_i \mid w_{i-N+1}^{i-1})}{1 - \sum_{w_i : C(w_{i-N+1}^{i}) > 0} \tilde{P}(w_i \mid w_{i-N+2}^{i-1})} \tag{1.3.20}$$

Note that the α parameter is a function of the history w_{i-N+1}^{i-1}. Also, recall that the $\tilde{P}(\cdot)$ probabilities are estimated using smoothing. In final, Equation 1.3.19 is reformulated

[3] Some variants of the backoff method look to a lower-order model if the counts of interest are less than a predefined threshold [51].

as [48]

$$P_{Bo}(w_i \mid w_{i-2}\ w_{i-1}) = \begin{cases} \tilde{P}(w_i \mid w_{i-2}\ w_{i-1}), & \text{if} \quad C(w_{i-2}\ w_{i-1}\ w_i) > 0 \\ \alpha(w_{i-2}^{i-1})\tilde{P}(w_i \mid w_{i-1}), & \text{if} \quad C(w_{i-2}\ w_{i-1}\ w_i) = 0 \\ & \quad \text{and } C(w_{i-1}\ w_i) > 0 \\ \alpha(wi-1)\tilde{P}(w_i), & \text{otherwise} \end{cases} \tag{1.3.21}$$

Some approaches when apply the backoff method treat N-grams of only one occurrence as zero-frequency events.

1.3.4 Class-based n-gram language modeling

Some words are related with other words regarding their meaning and syntactic use, e.g., *Monday* and *Saturday*. Their vicinal words very often tend to have similar probability distribution. Of course their use is not identical. It is unlikely to hear a conscious researcher saying to his colleagues *It's Monday! Let's go clubbing!* The assignment of words into classes it is possible to provide more reasonable probability computation for the less frequent (or unknown) histories under the assumption that they are similar to other more frequent histories.

Let the vocabulary V be partitioned into C classes, whereas a word w_i is grouped to a class $c_i \in C$. The language model is defined to be $n-gram\ class\ model$ if it is an n-gram model and if $1 \leq i \leq n$. The sentence probability of Equation 1.2.1 becomes [11]

$$P(w_1 \ldots w_n) = \prod_{i=1}^{n} P(w_i|c_i)P(c_i|h_i^{'}) \tag{1.3.22}$$

where the history is $h_i^{'} = \{c_1, c_2, \ldots, c_{i-1}\}$. In general, an n-gram class language model has fewer independent parameters than an n-gram language model, as is presented in Table 1.1. This type of language modeling can be viewed as another way to tackle data

Model	Number of independent parameters
n-gram	$V^n - 1$
n-gram class	$V + C^n - C - 1$

Table 1.1: Number of independent parameters for n-gram and n-gram class models

sparseness. However, the quality of a class-based n-gram model is strongly depended on the clustering procedure. In semantically narrow domains, such as ATIS [86], sufficient results are obtained using human-built semantic classes [109]. Of course, improvement is not always guaranteed, especially for broader domains. Automatic techniques of clustering have been proposed, which resulted to sufficient improvements [11, 52, 24]. Also, an n-gram language model can be linearly interpolated with a class-based one [112].

1.4 N-gram language modeling toolkits

We list some widely-used toolkits for N-gram language modeling that are freely available:

- **CMU-Cambridge Statistical Language Modeling toolkit:** the first version was written by Roni Rosenfeld at Carnegie Mellon University [17]. The current version is available at *http://mi.eng.cam.ac.uk/∼prc14/toolkit.html*.
- **SRI Language Modeling Toolkit:** developed in the SRI Speech Technology and Research Laboratory [106]. In addition, the toolkit was enhanced during several summer workshops organized by Johns Hopkins University/CLSP. It can be downloaded from *http://www.speech.sri.com/projects/srilm/download.html*.
- **HTK toolkit:** originally developed at the Machine Intelligence Laboratory of the Cambridge University Engineering Department [113]. The toolkit is primarily used for building and manipulating HMMs for speech recognition, however it includes a component for N-gram language modeling. It is available for downloading at *http://htk.eng.cam.ac.uk/*.

1.5 Evaluation of language models

The field of information theory [102] provides some useful notions in order to measure the performance of a language model. *Entropy* and *perplexity* are used to evaluate a language model.

Natural language is a kind of information source and a natural language sentence can be considered as a emitted signal, being a sequence of words. The distribution of the next word is highly dependent to the previous words. There is a great deal of variability and uncertainty in natural language. Entropy is a measure of information. Alternatively, entropy can be considered as a measure of"uncertainty" of a random variable. Let W be a random variable that ranges over the corpus vocabulary V and has a probability function P_w. The entropy of the random variable is

$$H(W) = -\sum_{w \in V} P(w) \log_2 P(w) \tag{1.5.1}$$

If log base 2 is used, the resulting units called binary digits. If the base 10 is used, the resulting units are expressed in decimal digits. Intuitively, entropy can be interpreted as a lower bound of bits that are required in order to encode a chunk of information according to an optimal encoding [48].

Given that a language model uses all possible vocabulary words to predict the next words, it follows that the model embodies a *per-word entropy* (*entropy rate*). The per-word entropy of a language model, L, for all possible sequences of words $w_1, w_2, \ldots, w_m$ is as follows [113]

$$H(L) = -\lim_{m \to \infty} \frac{1}{m} \sum_{w_1, w_2, \ldots, w_m} P(w_1, w_2, \ldots, w_m) \log_2 P(w_1, w_2, \ldots, w_m) \tag{1.5.2}$$

If the language being modeled is ergodic [18], the summation in Equation 1.5.2 can be omitted and $H(L)$ becomes [113]

$$H(L) = -\lim_{m\to\infty} \frac{1}{m} \log_2 P(w_1, w_2, \ldots, w_m) \tag{1.5.3}$$

It is interesting to note that if we have a long enough sequence of words (given the ergodicity), then $H(L)$ can be approximated as [113]

$$\hat{H}(L) = -\frac{1}{m} \log_2 P(w_1, w_2, \ldots, w_m) \tag{1.5.4}$$

Equation 1.5.4 has a suitable form for measuring the quality of a language model in terms of per-word entropy. This measurement is achieved by the notion of perplexity as [113]

$$PP = 2^{\hat{H}(L)} \tag{1.5.5}$$

If we substitute Equation 1.5.4 into Equation 1.5.5, we have [113]

$$PP = \hat{P}(w_1, w_2, \ldots, w_m)^{-\frac{1}{m}} \tag{1.5.6}$$

that is the perplexity of a language model. $\hat{P}(w_1, w_2, \ldots, w_m)$ denotes the estimated probability that the language model, L, assigns to the sentence $w_1, w_2, \ldots, w_m$.

Perplexity can be seen as a measurement giving the average number of the most probable words ,which can follow any word, with equal probability. It follows that more qualitative language models have lower perplexities.

1.6 Other language models

Despite the popularity and the successful use of N-gram language models, many other approaches have been proposed and used. It is beyond the scope of this thesis the complete discussion of them, but it is worth to briefly present few of other techniques of language modeling.

- **Decision tree models:** The space of histories is arbitrarily partitioned according to arbitrary binary questions. The questions are asked at the internal nodes. The probability $P(w \mid h)$ is computed at each leaf [1]. This approach it is likely to outperform N-gram models but is computationally costly [95].

- **Maximum entropy models:** Many models of language modeling suffer by data fragmentation. For example, as the order of the model increases, the data for estimation of the new parameters decreases. The maximum entropy approach was introduced in [20]. The main idea of this method is that arbitrary knowledge sources can be incorporated [94, 93], avoiding the problem of data fragmentation. As knowledge source is meant any function of a word and its history.

- **Dimensionality reduction:** The motivation behind this method is that the "true" (in other words, practically useful) size of the vocabulary is lower than the original size. In [5], the approach of *l*atent semantic analysis [19] was applied in order to reduce the dimensionality of the vocabulary.

Of course, many other methods for language modeling have been proposed in the literature. The survey of Rosenfeld [95] gives a short, yet complete, description of the whole field. The extended study of Goodman [30] presents several experiments with informative evaluation results.

1.7 Summary

In this chapter we discussed N-gram statistical language modeling, trying to acquire regularities of natural language using text processing, by computing N-gram probabilities. Smoothing and backoff techniques were presented for the problem of data sparseness problem. Also, an evaluation measurement for the quality of language models was defined. In final, freely available toolkits for N-gram statistical language modeling, as well as some other language modeling methods were briefly presented.

Chapter 2

Semantic Similarity Metrics

2.1 Introduction

In the previous chapter we studied various modeling techniques of natural language by the perspective that the surface of language (i.e., statistical regularities) can lead to a deeper knowledge. As deeper knowledge we may regard the "correctness" of a word sequence, expressed with a probability value, calculated by a language model. Of course, this is only an initial, pre-mature achievement compared to the whole richness of language. However, is still a step towards the inner structure of language.

In this chapter we discuss some major metrics that attempt to provide a numerical estimation of semantic similarity between words. The first family of metrics explores the lexical environment of words, motivated by the assumption that there is a correlation between context and meaning. Also, we present a second family of metrics that exploits knowledge resources, like thesauri and ontologies, in order to compute semantic similarities.

2.2 Contextual similarity metrics

The computational model of semantics is refered as *word-space model* by Hinrich Schütze [99]. A model that measures the semantic relationship between words is defined with respect to the vocabulary which forms a high-dimensional space whereas each word can be considered as one dimension. The word-space model reflects a spatial representation of word meaning. The key idea of this model is that semantic similarity can be represented as proximity in n-dimensional space, where n is the cardinality of vocabulary set [98].

Spatial proximity between words as a representation of their semantic similarity seems to be very intuitive and naturally derived with respect to the way that human conceptualize similarities. This *geometric metaphor of meaning* has been pointed out by the work of Lackoff and Johnson [56, 57]. They state that metaphors form the raw base of abstract conceptualization. Also, they argue that these metaphors are used by human mind for reasoning about abstract and complex phenomena, such as natural language and semantics. This physical tendency of human mind places the conceptual locations of words with similar meaning to be "near" each other, while the

dissimilar words are placed "far apart. Of course, a sole word in a high-dimensional space gives no additional information for deeper understanding off the word. The space must be populated with other words in order to apply the proximity as an indicator of similarity. The geometric metaphor of meaning conceptualize the words as locations in a word-space and the similarity is considered as the proximity between the locations [98].

2.2.1 The distributional hypothesis of semantic similarity

The word-space model provides not only a spatial representation of meaning, but also naturally suggests a way to build the model. That is, only the words are used with no a priori knowledge or constraints about the underlying semantics. Statistical approaches are valuable tools in order to learn the distributional properties of words. This framework is suitable for measuring the proximity as is reflected by the distributional similarity. The next step is the semantic similarity estimation, which is motivated by the *distributional (contextual)* hypothesis of semantically similar words. This hypothesis assumes that words with similar distributional properties have similar meaning.

One of the first studies of the distributional hypothesis is the work of Rubenstein and Goodenough [96], stating that "words which are similar in meaning occur in similar contexts". Schütze and Pedersen [100], re-phrased this hypothesis, considering the data sparseness problem, as "words with similar meanings will occur with similar neighbors if enough text material is available". The linguist Zelling Harris [31, 32] initialy believed that it is possible to typologize the whole of linguistic phenomena using their distributional behavior without intrusion of other features. Later, he extended his distribution-based analysis, considering that in many cases the meaning goes beyond the formal linguistic theory, affected by many extralinguistic factors, such as social situations. Even in these cases, Harris was suggested that will always exist a distributional correlation between the extralinguistic factors and the influenced linguistic phenomena. The core idea of his work is that the differences of meaning are mediated by differences of distribution: "... if we consider words or morphemes A and B to be more different in meaning that A and C, then will often find that the distributions of A and B are more different than the distributions of A and C. In other words, difference of meaning correlates with difference of distribution".

The earliest, 1965, validation of the distributional hypothesis was conducted by Rubenstein and Goodenough [96]. They compared the contextual similarities of 65 noun pairs with synonymy scores assigned by students. It is worth to quote their conclusions; (a) "there is a positive relationship between the degree of synonymy (semantic similarity)existing between a pair of words and the degree to which their contexts are similar", and (b) "it may safely inferred that a pair of words is highly synonymous if their contexts show a relatively great amount of overlap. Inference of degree of synonymy from less amounts of overlap, however, is apparently uncertain since words of low or medium synonymy differ relatively little in overlap". Moreover, Rubenstein and Goodenough, state that the generalization of the above conclusions is dependent on factors like vocabulary size and homogeneity of content. Three decades later, 1991, Miller and Charles [70] repeated the experiment of Rubenstein and Goodenough using 30 of the 65 pairs and they reached similar results, supporting the distributional

(contextual) hypothesis.

However, the distributional assumption faces a criticism, which claims that a simple word-space is inefficient to represent the complete nature of semantic similarity between words. Hence, it does not give an exact discrimination between semantic relations, such as synonyms, antonyms, and hyponyms. This critique is valid if certain semantic relations are a priori well-defined, but from another point of view these relations are not axiomatic. Thus, the distributional hypothesis seems to formalize a useful tool, operating on the broad notion of semantic similarity [98].

2.2.2 Contextual metrics

In this section we briefly present some major techniques that rely solely in context in order to estimate the semantic similarity between words. From this point of view, these metrics are unsupervised; they require no external knowledge and, thus, they can be regarded as language independent.

Mutual Information

Mutual information is an information theoretic concept that measures an association norm between words. Word association is an important notion of *psycholinguistics*, especially in te field of lexical retrieval. Imagine the case when a subject thinks of a word after he/she experienced a previous one. The strength among the two words is referred as their association norm.

Mutual information between two words, w_1 and w_2, having probabilities $P(w_1)$ and $P(w_2)$, respectively, is defined to be [15]

$$I(w_1, w_2) = \frac{P(w_1, w_2)}{P(w_1)P(w_2)} \tag{2.2.1}$$

The physical interpretation of Equation 5.3.3 is that it compares the probability of seeing words w_1 and w_2 together with the probability of meeting the words independently [1]. If there is a strong association between w_1 and w_2, then the joint probability $P(w_1, w_2)$ will be greater than $P(w_1)P(w_2)$, giving $I(w_1, w_2) >> 0$. In the case that there is no remarkable relation between the two words, then $P(w_1, w_2) \approx P(w_1)P(w_2)$, resulting to $I(w_1, w_2) \approx 0$. Last, if words w_1 and w_2 have complementary distributional properties, then the joint probability wlll be less than the product of the unigram probabilities, leading to $I(w_1, w_2) << 0$ [15].

The enumerator of Equation 5.3.3 denotes the joint probability of two words that is computed over the times that word w_2 follows word w_1. This consideration can be extended by using a context window size parameter. This window allow us to search for the co-occurrence of both words within a length of contexts; in other words, to use different scopes. Usually small window sizes capture fixed expressions (like, duty free) and other relations of short range. Larger windows are able to provide a raw indication of relationships, formed over larger range.

[1] Note that we deal with unigram and bigram probabilities which can be estimated using any technique of language modeling, presented in the previous chapter. This probability estimation holds for every N-gram probability throughout this thesis, unless a different estimation is declared.

Kullback-Leibler metric

The *Kullback-Leibler (KL)* divergence is a measure of the difference between two probability distributions. Suppose two distributions, P and Q of a discrete random variable. Their KL divergence is computed as

$$D_{KL}(P\|Q) = \sum_{y \in Y} P(y) \log \frac{P(y)}{Q(y)} \tag{2.2.2}$$

over all values $y \in Y$. The KL metric is not a distance metric because it is not symmetric ($D_{KL}(P\|Q) \neq D_{KL}(Q\|P)$) and does not satisfy the triangle inequality ($D_{KL}(P\|Q) + D_{KL}(Q\|P) \not\geq D_{KL}(Q\|Z)$).

In fact, KL metric is a measure of dissimilarity [2], since equals to 0 when the two distributions are the same, and greater than zero otherwise. This intuitively explains the reason why the KL metric does not satisfy the triangle inequality: in [34], Hatzivassiloglou and McKeown observed that dissimilarity is not transitive.

One may wonder about the usefulness of KL divergence given that is not a valid distance metric. The answer to this question can be reached through the fields of statistics, information theory, and the maximum entropy principle. The proof goes beyond the scope of this thesis, so we will mention the concluded results. Given that the KL metric measures the dissimilarity between two distributions, the greater their divergence is, the easier (on average) their discrimination is [55, 58]. From another point of view, if the difference between distributions P and Q is large, then P and Q is dissimilar, so, it is inefficient (on average) to use Q instead of P [55, 58].

We can apply the KL metric to measure the semantic dissimilarity between two words, w_1 and w_2, based on the distributional hypothesis which assumes that similar words have similar contextual distribution. First, we have to define exactly which context we aim to consider. Assume that for each word we examine its' immediate context. In order to have a visualization of this consider a word, w, with its' immediate context in a sequence

$$\ldots \; v_{1,L} \; w \; v_{1,R} \; \ldots$$

where $v_{1,L}$ represents the first word appeared in the left of word w, and $v_{1,R}$ is the first word occurred in the right of w. For the moment, by immediate context let's mean the words that is possibly follow (right contexts) the words of interest $w_1 and w_2$. In particular, we measure the divergence between the bigram probability distributions of words w_1 and w_2, denoted by the corresponding capital letters W_1 and W_2, respectively. These distributions are defined over all possible contexts, that is the words of the corpus vocabulary V. So far, the dissimilarity of words w_1 and w_2 can be computed using the KL divergence of the corresponding right bigram conditional probability distributions W_1 and W_2, as

$$D^R_{KL}(W_1\|W_2) \equiv D^R_{KL}(w_1, w_2) = \sum_{v_{1,R} \in V} P(v_{1,R} \mid w_1) \log \frac{P(v_{1,R} \mid w_1)}{P(v_{1,R} \mid w_2)} \tag{2.2.3}$$

[2] Despite the fact that KL is a dissimilarity metric, is still useful to indicate semantic similarity, as dissimilarity is the other side of the coin of semantic relationships.

where $v_{1,R}$ denotes the first word that occurs in the right contexts of word w_i ($i = 1\ or\ 2$), $P(v_{1,R} \mid w_i)$ is the bigram conditional probability of the bigram $w_i\ v_{1,R}$ given that word w_i has occurred. Also, note that the two bigram distributions, W_1 and W_2, are compared over the whole vocabulary V [3]. It must be noted that the dissimilarity computed by Equation 2.2.3 is not complete with the sense that we should, also, compute the divergence of W_2 and W_1 for the right contexts:

$$D_{KL}^{R}(W_2 \| W_1) \equiv D_{KL}^{R}(w_2, w_1) = \sum_{v_{1,R} \in V} P(v_{1,R} \mid w_2) \log \frac{P(v_{1,R} \mid w_2)}{P(v_{1,R} \mid w_1)} \tag{2.2.4}$$

Moreover, in order to have a left context-dependent divergence we have to formulate Equations 2.2.3 and 2.2.4 for the left contexts, as

$$D_{KL}^{L}(W_1 \| W_2) \equiv D_{KL}^{L}(w_1, w_2) = \sum_{v_{1,L} \in V} P(v_{1,L} \mid w_1) \log \frac{P(v_{1,L} \mid w_1)}{P(v_{1,L} \mid w_2)} \tag{2.2.5}$$

and

$$D_{KL}^{L}(W_2 \| W_1) \equiv D_{KL}^{L}(w_2, w_1) = \sum_{v_{1,L} \in V} P(v_{1,L} \mid w_2) \log \frac{P(v_{1,L} \mid w_2)}{P(v_{1,L} \mid w_1)} \tag{2.2.6}$$

, respectively. The left-context bigram probabilities are calculated using the reversed word order. In final, the symmetric left and right contextual dissimilarity between words $w_1 and w_2$ is [79]

$$D_{KL}^{L,R}(w_1, w_2) = D_{KL}^{L}(w_1, w_2) + D_{KL}^{L}(w_2, w_1) + D_{KL}^{R}(w_1, w_2) + D_{KL}^{R}(w_2, w_1) \tag{2.2.7}$$

A drawback of the *KL* metric is its' unbounded character, since has ratios with denominators which are probable to approach zero value. Hence, few words can dominate the computation of the *KL* divergence. This is more likely to happen when poor training data are available with few counts for some N-grams. Instead, bounded metrics can be used, as we will see next.

Information-radious metric

The *Information-radious (IR)* metric is similar to the *KL* metric, with the difference that is bounded, since the denominator is the average of the probability distributions:

$$D_{IR}(P \| Q) = \sum_{y \in Y} P(y) \log \frac{P(y)}{\frac{1}{2}(P(y) + Q(y))} \tag{2.2.8}$$

By definition the denominator is greater than zero, thus no constaints are imposed on the training data and estimates that may approach zero can be used directly. As in

[3] It is possible that we will not have training data for some certain bigrams. In this case it is strongly advisable to use the backoff strategy during the language modeling construction, in order to use a lower-order model for the probability estimation of the unseen events.

KL metric, the divergences of bigram conditional probability distributions W_1 and W_2, and vice versa, are

$$D_{IR}^{R}(W_1 \| W_2) \equiv D_{IR}^{R}(w_1, w_2) = \sum_{v_{1,R} \in V} P(v_{1,R} \mid w_1) \log \frac{P(v_{1,R} \mid w_1)}{\frac{1}{2}(P(v_{1,R} \mid w_1) + P(v_{1,R} \mid w_2))} \tag{2.2.9}$$

and

$$D_{IR}^{R}(W_2 \| W_1) \equiv D_{IR}^{R}(w_2, w_1) = \sum_{v_{1,R} \in V} P(v_{1,R} \mid w_2) \log \frac{P(v_{1,R} \mid w_2)}{\frac{1}{2}(P(v_{1,R} \mid w_1) + P(v_{1,R} \mid w_2))} \tag{2.2.10}$$

for the right contexts, respectively. Similarly, for the left contexts we have

$$D_{IR}^{L}(W_1 \| W_2) \equiv D_{IR}^{L}(w_1, w_2) = \sum_{v_{1,L} \in V} P(v_{1,L} \mid w_1) \log \frac{P(v_{1,R} \mid w_1)}{\frac{1}{2}(P(v_{1,L} \mid w_1) + P(v_{1,L} \mid w_2))} \tag{2.2.11}$$

and

$$D_{IR}^{L}(W_2 \| W_1) \equiv D_{IR}^{L}(w_2, w_1) = \sum_{v_{1,L} \in V} P(v_{1,L} \mid w_2) \log \frac{P(v_{1,L} \mid w_2)}{\frac{1}{2}(P(v_{1,L} \mid w_1) + P(v_{1,L} \mid w_2))} \tag{2.2.12}$$

Finally, the symmetric left and right contextual dissimilarity among words w_1 and w_2 is computed as [79]

$$D_{IR}^{L,R}(w_1, w_2) = D_{IR}^{L}(w_1, w_2) + D_{IR}^{L}(w_2, w_1) + D_{IR}^{R}(w_1, w_2) + D_{IR}^{R}(w_2, w_1) \tag{2.2.13}$$

Each of the four terms of the above summation has an upper bound value equal to log(2), so the maximum score of absolute dissimilarity [4] is $4 \times \log(2)$.

Manhattan-norm metric

The *Manhattan-norm (MN)* metric can be considered as a geometric distance (it is closely related with *Euclidean* distance) and is, also, a true distance metric as the term "norm" indicates. It is computed as

$$D_{MN}(P \| Q) = \sum_{y \in Y} \mid P(y) - Q(y) \mid \tag{2.2.14}$$

The *MN* metric computes the absolute difference between the bigram conditional probabilty distributions W_1 and W_2. Due to the absolute function the *MN* metric is symmetric:

$$D_{MN}(P \| Q) \equiv D_{MN}(Q \| P)$$

[4] Of course, the phrase "absolute dissimilarity" does not rely on any linguistic basis, since it is impossible to define a universal ground truth for dissimilarity. The quoted phrase is given with a technical sense. By the same sense, two words are computed to share semantically absolute similarity if they have identical contextual distributions.

The contextual distance between the bigram conditional probability distributions W_1 and W_2 is

$$D^R_{MN}(W_1 \| W_2) \equiv D^R_{MN}(w_1, w_2) = \sum_{v_{1,R} \in V} \mid P(v_{1,R} \mid w_1) - P(v_{1,R} \mid w_2) \mid \qquad (2.2.15)$$

for the right contexts. In similar manner, the distribution distance for the left context is defined as

$$D^L_{MN}(W_1 \| W_2) \equiv D^L_{MN}(w_1, w_2) = \sum_{v_{1,L} \in V} \mid P(v_{1,L} \mid w_1) - P(v_{1,L} \mid w_2) \mid \qquad (2.2.16)$$

Last, the left and right contextual distance (hence, dissimilarity) between words w_1 and w_2 is calculated as [79]

$$D^{L,R}_{MN}(w_1, w_2) = D^L_{MN}(w_1, w_2) + D^R_{MN}(w_1, w_2) \qquad (2.2.17)$$

Each of both terms of Equation 2.2.17 has a lower and upper bound of zero and two, respectively. Thus, two words of identical contextual distributions will have a zero value of *MN* distance, while a distance score equal to 4 indicates absolute dissimilarity.

Cosine similarity metric

The *Cosine (CS)* metric is a similarity metric in contrast to the above dissimilarity measures. It is computed as

$$D_{CS}(P \| Q) = \frac{\sum_{y \in Y} P(y) Q(y)}{\sqrt{\sum_{y \in Y} P(y)^2 \sum_{y \in Y} Q(y)^2}} \qquad (2.2.18)$$

Note that the *CS* metric is symmetric:

$$D_{CS}(P \| Q) \equiv D_{CS}(Q \| P)$$

For the distributions W_1 and W_2, the *CS* metric computes their vector product. Each vector is a sequence of bigram conditional probabilities. The contextual similarity between the bigram conditional probability distributions W_1 and W_2 is

$$D^R_{CS}(W_1 \| W_2) \equiv D^R_{CS}(w_1, w_2) = \frac{\sum_{v_{1,R} \in V} P(v_{1,R} \mid w_1) P(v_{1,R} \mid w_2)}{\sqrt{\sum_{v_{1,R} \in V} P(v_{1,R} \mid w_1) \sum_{v_{1,R} \in V} P(v_{1,R} \mid w_2)}} \qquad (2.2.19)$$

for the right contexts. Similarly, for the left context we have

$$D^L_{CS}(W_1 \| W_2) \equiv D^L_{CS}(w_1, w_2) = \frac{\sum_{v_{1,L} \in V} P(v_{1,L} \mid w_1) P(v_{1,L} \mid w_2)}{\sqrt{\sum_{v_{1,L} \in V} P(v_{1,L} \mid w_1) \sum_{v_{1,L} \in V} P(v_{1,L} \mid w_2)}} \qquad (2.2.20)$$

In final, the left and right contextual similarity between words w_1 and w_2 is computed as [79]

$$D^{L,R}_{CS}(w_1, w_2) = D^L_{CS}(w_1, w_2) + D^R_{CS}(w_1, w_2) \qquad (2.2.21)$$

Each of both terms of Equation 2.2.21 has a lower and upper bound of zero and one, respectively. Thus, two words of identical contextual distriutions will have *CS* similarity score equal to two.

Cosine similarity metric using context window

In the field of Information Retrieval documents and queries are often represented by vectors, in a space calles *vector space model*. This representation is widely used for similarity computation between documents and queries, document classification and clustering. Such a vector can be represented as

$$d = (t_1, t_2, \ldots, t_N)$$

where t_i is the i^{th} element (feature) of the vector. Various schemes are used for the definition of the feature values:

- **binary:** 1 if the feature occurs, else 0.
- **term frequency:** the frequency of feature.
- **term frequency-inverse document frequency (tf-idf):** greater importance is given to a feature that occurs frequently in a document, while appears rarely in the whole document collection.

Inspired by this approach, we can represent a word, instead of a vector, with respect to its contexts, in order to compute word similarity. This representation is called *bag-of-words model*. We rely on the distributional hypothesis of similarity, using the contexts of each word as features.

In "bag-of-words" [101, 41] models, for each word w in the vocabulary a context window size WS is selected. The right and left contexts of length WS in the corpus are considered for word w, e.g.,

$$[v_{WS,L} \quad ... \quad v_{2,L} \quad v_{1,L}] \quad w \quad [v_{1,R} \quad v_{2,R} \quad ... \quad v_{WS,R}]$$

where $v_{i,L}$ and $v_{i,R}$ represent the i^{th} word to the left and to the right of w respectively. The feature vector for every word w is defined as

$$T_{w,WS} = (t_{w,1}, t_{w,2}, ..., t_{w,N})$$

where $t_{w,i}$ is a non-negative integer and WS is the context window size. Note that the feature vector size is equal to the vocabulary size N, i.e., we have a feature for each word in the vocabulary V. The i^{th} feature value $t_{w,i}$ reflects the occurrences of vocabulary word v_i within the left or right context window WS. This feature value is set according to a Binary or a Term Frequency scheme. As we saw, Binary scheme assigns 1 if the word v_i appears within the left and right window context of size WS for the word w, while Term Frequency scheme assigns the number of occurrences of v_i in left and right WS. Both schemes assign a 0 value if v_i does not exist within WS. The "bag-of-words" model using Binary or Term Frequency Scheme, measures the similarity of two words, w_1 and w_2, as the product of their corresponding feature vectors, $T_{w_1,WS}$ and $T_{w_2,WS}$, like the cosine similarity metric [77, 41]:

$$CS_{BoW,WS}(w_1, w_2) = \frac{\sum_{i=1}^{N} t_{w_1,i} t_{w_2,i}}{\sqrt{\sum_{i=1}^{N} (t_{w_1,i})^2} \sqrt{\sum_{i=1}^{N} (t_{w_2,i})^2}} \tag{2.2.22}$$

given a context window of length WS.

2.3 Resource-based similarity metrics

In this section we present several methods that exploit knowledge resources in order to compute semantic similarity . So, these measures are supervised, since they rely on external knowledge. The usual used resources are *ontologies* that can be considered as structured models for knowledge representation. In general, an ontology describes the logical structure of a domain. It consists of *terms* that represent the domain concepts, *relationships* among the terms and *properties* for each term that describe several features and attribute of the term [5]. Concepts are usually organized in a hierarchical taxonomy. The most general concepts are located at the top of the taxonomy, while the more specific concepts are found in lower positions.

2.3.1 Ontology types

Ontologies can be categorized into: (a) *domain ontologies*, and (b) *generic ontologies*. Domain ontologies represent knowledge of a specific domain, while generic ontologies represent a generic (in other words, common sense) knowledge about the world.

Some representative examples of generic ontologies are [88]:
(a) WordNet [4, 69], that tries to model the lexical knowledge of native English speakers. It includes nouns, verbs, adjectives and adverbs, which are grouped into synonym sets, *synsets*. Each synonym set represents a concept.
(b) SENSUS [54], which it was constructed as an extension and re-organization of WordNet. Each node represents one concept and the concepts are linked with IS-A relations.
(c) Cyc Knowledge Base [76, 89], which includes terms and relations between them, combined with a large amount of human knowledge, expressed by facts, rules and heuristics, for reasoning.

Some major examples of domain ontologies are [88]:
(a) UMLS [73], MESH [72]. UMLS includes a vast amount of multilingual concepts about health and biomedicine. MESH is the acronym for Medical Subject Heading and is a thesaurus of medicine vocabulary, consisting of sets of terms, hierarchically structured.
(b) GO (Gene Ontology) [8, 37] deals with genomic data, like protein description.
(c) SDTS [105] includes data about topographic maps and hydrographic charts.

2.3.2 Ontology approaches

Ontologies can represent knowledge with different ways. So,different methods exist for comparing concepts within or across ontologies. Next, we present a brief description of the main approaches [88].

- **Single ontology approach**: All the knowledge sources are linked to only one, global ontology. There is a common vocabulary regarding the entity semantics. The object of each component source are related to the global model. This

[5] When we discuss about ontologies "terms" and "concepts" will be used interchangeably.

approach is suitable for cases where all the individual sources have almost the same domain view.

- **Hybrid approach**: Each knowledge source use its own ontology to describe the semantics. However, all ontologies are constructed over one common vocabulary. The common vocabulary includes basic terms (primitives), which the individual ontologies use in order to derive more complex terms. A complex term can be built by applying some operations, such as union or intersection, over the simple terms.
- **Multiple ontology approach**: Several knowledge sources are described by different ontologies. Also, there is no common vocabulary to be shared. The comparison of concepts is a difficult task and several methods have been proposed in the literature trying to address this interesting problem.

2.3.3 Ontology-based metrics

The data of knowledge sources are represented by properties and is hierarchically organized. The hierarchy has a taxonomic structure, containing superclasses and subclasses, and is constructed according to the ontology that corresponds to the knowledge source. The question is how the concepts are semantically compared upon this representation. In this section, we present several metrics that exploit this knowledge organization in order to measure the semantic similarity between concepts. In general, the ontology-based semantic similarity metrics can be distinguished to the following four categories: (a) a metric examine how close the two concepts are positioned in the taxonomy, (b) the shared information between the two concepts is considered, (c) a metric is based on the properties of the concepts, and, (d) the above approaches are combined [88]. For these categories we present some major methods for computing semantic similarity, based on the study of Raftopolou and Petrakis [88].

Let C be the set of concepts of an IS-A taxonomic structure. The desired measurement is the semantic similarity, sim, between two terms/concepts $c1, c2 \in C$.

Edge-counting metrics

This type of methods considers the proximity of two terms c_1 and c_2 in the taxonomy. In [87], a metric is proposed that is function of the *shortest path* between two terms, defined as

$$sim_{sp}(c_1, c_2) = 2maxP - L \tag{2.3.1}$$

where $maxP$ is the maximum path length between two terms, encountered in the whole taxonomy. The minimum number of links between c_1 and c_2 is denoted by L. This metric assigns greater similarity score to concepts that are located in close positions in the hierarchy. Li et al. [60] proposed a method that combines the shortest path between c_1 and c_2 and the taxonomic depth of the most specific common concept c. The most specific common concept c is defined to be the common parent of c_1 and c_2 with the less $IS - A$ links of c_1 and c_2.

$$sim_{Li}(c_1, c_2) = \exp^{-\alpha L} \frac{\exp^{\beta H} - \exp^{-\beta H}}{\exp^{\beta H} + \exp^{-\beta H}} \tag{2.3.2}$$

where H is the number of IS-A links from the most specific common concept c to the root of the taxonomy. The parameters $\alpha \geq 0$ and $\beta > 0$ scale the weight that is given to the shortest path length and depth, respectively.

Information content metrics

The measures of this category exploit the *information content* of each term. In practice, the information content of a concept c is related to the frequency of an instance of concept in some large corpus. The probability $P(c)$ is the probability of seeing an instance of concept c. This probability is defined as $P(c) = \frac{freq(c)}{N}$, where N denotes the number of terms in the taxonomy. Also, $freq(c) = \sum_{n \in words(c)} n$, where $words(c)$ is the number of terms subsumed by c (having c as parent). The probability of most informative subsumer of terms c_1 and c_2 is

$$P_{mis} = (c_1, c_2) = min_{c \in S(c_1,c_2)}\{P(c)\} \tag{2.3.3}$$

where $S(c_1, c_2)$ is the set of concepts that subsume c_1 and c_2. Lin [62], proposed the following metric for measuring the similarity between c_1 and c_2.

$$sim_{Lin}(c_1, c_2) = \frac{2 \ln P_{mis}(c_1, c_2)}{\ln P(c_1) + \ln P(c_2)} \tag{2.3.4}$$

Jiang and Conrath [46], suggested the following distance metric.

$$dist_{Jiang}(c_1, c_2) = -2 \ln P_{mis}(c_1, c_2) - (\ln P(c_1) + \ln P(c_2)) \tag{2.3.5}$$

In order to compute similarity between c_1 and c_2, we take $1 - dist_{Jiang}(c_1, c_2)$.

Feature-based metrics

The previous metrics do not consider the term features for similarity computation. The following metric proposed by Tversky [107] considers also the features of terms in order compute semantic similarity. Each term has a description set that includes a number of words, indicating its features.

$$sim_{Tversky}(c_1, c_2) = \frac{\mid C_1 \cap C_2 \mid}{\mid C_1 \cap C_2 \mid + \kappa \mid C_1 \setminus C_2 \mid + (\kappa - 1) \mid C_2 \setminus C_2 \mid} \tag{2.3.6}$$

C_1 and C_2 correspond to the description sets of c_1 and c_2, respectively. Parameter κ takes values between zero and one, defining the weight given to non-common features of c_1 and c_2.

Combinational metrics

The metrics of this category combine ideas from the above three categories. Rondriquez et al. [92] proposed a similarity metric that distinguish term features into three types: (a) functions, (b) parts, and (c) attributes. For example, for the term "university", the phrase "to educate" serves as a function. The words "wall" and "door" may be considered as parts, while "design properties" can be assumed as other attributes.

Then, the similarity between two terms c_1 and c_2 is calculated by a linear combination of three similarity factors. Each similarity factor is computed with respect to one of the three feature types.

$$sim_{Rondriquez}(c_1, c_2) = \lambda_f S_f(c_1, c_2) + \lambda_p S_p(c_1, c_2) + \lambda_a S_a(c_1, c_2) \tag{2.3.7}$$

The weights λ_f, λ_p, and λ_a, and the corresponding similarity scores (S_f, S_p, and S_a), stand for function, parts, and attributes term features, respectively. Note that $\lambda_f + \lambda_p + \lambda_a = 1$. Also, for S_f, S_p, and S_a computation the Tversky [107] metric is used.

2.4 Summary

In this chapter we introduced the distributional hypothesis of semantic similarity. That is, similarity in context implies similarity of meaning. Next, we presented fully contextual techniques for computing semantic similarity between words. These metrics are considered to be unsupervised, since they require no external knowledge. Last, we briefly described some major methods that exploit knowledge resources, such as ontologies, in order to compute semantic similarity.

Chapter 3

Unsupervised Combination of Metrics for Semantic Class Induction

3.1 Introduction

Many applications dealing with textual information require classification of words into semantic classes including dialogue systems, language modeling, speech understanding and machine translation. The use of knowledge sources, such as dictionaries and thesauri, for this task have some disadvantages compared to corpus-based, automatic methods [33]:

- The entries must be generated by hand that is a time-consuming effort.
- Possible changes to some entries may demand a revision of the knowledge source in order to preserve its consistency.
- Many entries are not present to some particular sources.
- In many cases, the entries give a general sense of the meaning, without being specific to any domain. This is contradictory to the fact that the meaning of a word changes as the word is used in different domains with different ways.
- The source construction is relatively subjective, since it is depended on human expertise and assumptions, while an automatic approach is based on examples extracted from an actual corpus.

The lack of data is very often for systems designed for new domains and requires significant manual effort. An automatic or semi-automatic algorithm for extracting semantic classes from text leads to the rapid development of many natural language processing systems, capturing the specific semantics for the domain of interest.

Semantic parsing has important role in speech recognition, since spontaneous spoken utterances it is possible to have disfluencies and ungrammatical parts [84]. In [84], a statistical framework for semantic parsing is described. Semantic classes are

used for writing rules of a semantic grammar about a travel information domain. The input is parsed and the appropriate substrings are mapped to the corresponding semantic classes. Using this representation, two information models are exponentially combined in order to resolve utterance ambiguity. The work of Fosler-Lussier and Kuo [24], presents an attempt for rapid dialogue system development in a domain with small amounts of training data. In [24], semantic classes are integrated into language models, for a natural language understanding module of an *ASR* dialogue system. Figure 3.1 shows how the original string (line 1) can be parsed using several levels of semantic parsing (lines 2-4). This approach can use reduced amount of required data for building

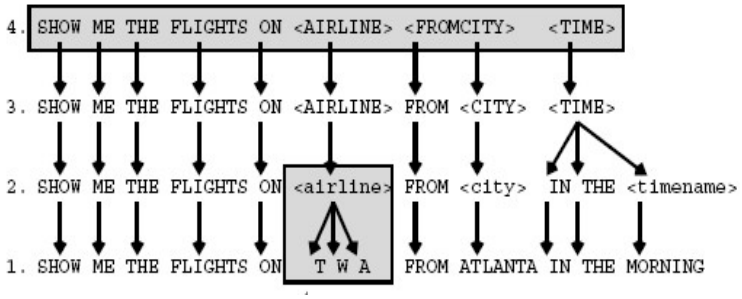

Figure 3.1: Utterance island parsing (Fosler-Lussier & Kuo, [24]).

language models. Also, a significant reduction of perplexity and *ASR* word error rate were reported, for such task with poor training data. In [49], a speech understanding system is described that answers questions about restaurants. A stochastic context-free grammar (*SCFG*) is used to smooth a bigram grammar and add structural constraints, as is shown in Figure 3.2. Suppose that the speech recognizer directs the sequence "I

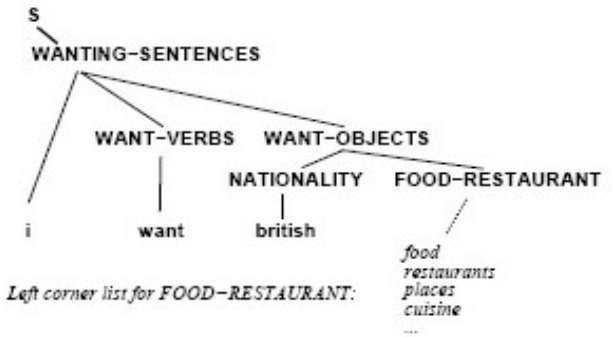

Figure 3.2: Semantic parsing in *SCFG* grammar (Jurafsky et al., [49]).

want British" to the parser. Then the parser will derive words "food", "restaurant", "places", "cuisine", etc, as following words. These words are members of the same

semantic class "FOOD-RESTAURANT", as is shown in Figure 3.2. The use of *SCFG* is reported to improve word error rate.

Semantic class construction and semantic similarity metrics are also important for web applications such as web search and document retrieval. For example, a user during web search often rephrases slightly the submitted query in order to get a new set of results, which it is likely to contain more relevant documents. In such cases the user includes in the reformed query words that are synonymous or semantically related to the words of the initial query. In query expansion the addition of semantically related words to the query it is likely to increase the relevant retrieved documents, which in other case may be missed [28]. This raises the idea of semantic representation and retrieval rather than lexical string matching [21]. In [108, 66, 23], the queries are expanded using related words, acquired from WordNet. In general, query expansion is shown to increase the recall of the retrieved documents. Gaugh et al. [28] added semantic similar words to queries according to a similarity measure. The similarity between query and candidate additional words was calculated using a corpus-based cosine similarity measure.

Our approach [41] [1] **:** we propose unsupervised algorithms for combining semantic similarity metrics for the problem of automatic class induction. The idea behind metric combination is that each individual metric serves a classifier with its own advantages, so the combination of different classifier is able to lead to more robust and qualitative results. The automatic class induction algorithm is based on the work of Pargellis et al. [79]. The semantic similarity metrics that combined are based on narrow- and wide-context cosine similarity. The metrics are combined using linear weights that are computed 'on the fly' and are updated at each iteration of the class induction algorithm, forming a corpus-independent metric. Specifically, the weight of each metric is selected to be inversely proportional to the inter-class similarity of the classes induced by that metric and for the current iteration of the algorithm. The automatic calculation of weights extends the work of Pangos et al. [77], where similarity metrics were combined according to fixed weights, estimated an held out data. The proposed algorithms are evaluated on two corpora: a semantically heterogeneous news domain, HR-Net [36], and an application-specific travel reservation corpus, ATIS [86]. Our approach is presented in Sections 3.3 - 3.6.

3.2 Related work

Numerous corpus-based techniques have been proposed for generating semantic classes. Some approaches consider the semantic relatedness by a relatively formal linguistic point of view and attempt to identify semantic relations such as *hypernymy/homonymy* and *meronymy*. Other methods rely on the distributional hypothesis of meaning, that is the contextual similarity implies similarity of meaning. The latter approach derives classes of words that their contextual use indicate that they share the same particular meaning.

Hearst [35] proposed a method for automatic acquisition of hyponymy relation. A

[1] **Joint work with Athanasios Tegos. Tegos is currently graduate student in the Department of Electronics and Computer Engineering, Technical University of Crete.**

noun B is a hyponym of a noun A, if B is a subtype or instance of A. For example "Nick Cave" is a hyponym of "artist" (and "artist" is a hypernym of "Nick Cave"). She selected by hand three lexico-syntactic patterns by text observation, that indicate hyponymy. These patterns were used in order to extract automatically another three patterns. Hearst tested this procedure for discovering meronymy relation, but without great success. Berland and Charniak [7] extended the work of Hearst by finding lexical patterns that indicate meronymy. Meronyny stands for part-whole relation. For example, "floor" is part of "building". For predefined pairs of words that are related with a part-whole relation, they collected sentences that include the pair words within close proximity. From these sentences they extracted by manual inspection five lexico-syntactic patterns as indicators for meronymy. An example sentence is "... the basement of the building". Berland and Charniak reported that the success of their method compared to the work of Hearst [35] for meromyny identification, is due to the use of a large corpus. More recently, Snow et al. [104], suggested a method for automatic classifier for hypernym/hyponym relation. They used examples of known hypernym pairs in order to automatically extract new examples of lexico-syntactic patterns. The acquired patterns were used for the training of the classifier.

Caraballo [12] used the Wall Street Journal to acquire conjunctions of noun phrases (e.g., "executive vice-president and treasurer") and appositives (e.g., "James H. Rosenfield, a former CBS executive). The motivation behind his work is that nouns in conjuctions or appositives tend to be semantically related. Then the head words of each noun phrase were stemmed, forming the nouns of interest. A vector was constructed for each noun, including as features the counts for how many times each other noun appears in a conjunction or a appositive with it. The semantic similarity between two nouns was computed by applying the cosine metric over the noun vectors. Caraballo used a bottom-up clustering method in order to create a tree of all of the nouns. First, each noun was put into its own node and the cosine similarity was computed between each pair of nodes. The two most similar nouns were combined by assigning them a common parent. Next, the new node's similarity was computed to each other node by taking a weighted average of the similarities between each of its children and the other node. As before, the two most similar nodes were allowed to be merged together under a common parent.

The work of Lin and Pantel [63] exploits dependency relationships between words in order to generate semantic classes. As a dependency relationship is defined an asymmetric binary relationship between a word called *head* and another word called *modifier*. The structure of a sentence can be represented by dependency relationships. Note that a sentence word is possible to have more than one modifiers, but each word can modify (if does) at most one word. The following figure gives an example of the dependency relationships for the sentence "John found a solution to the problem". The dependency relationships in Figure 3.3 are represented by arrows. An arrow is directed from the head to the modifier. Also, the depicted labels for each arrow denote several types of dependency relationships. Each word is represented by a vector which has as features the dependency relationships that were extracted from the corpus. The similarity between words is computed according to their feature vectors. Given this vector representation, a heuristic algorithm is used to cluster words into semantic

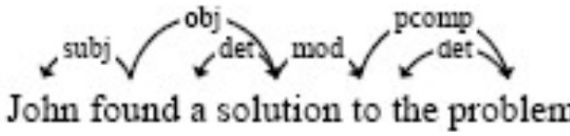

Figure 3.3: Example of dependency relationships within sentence (Lin & Pantel, [63]).

classes.

Siu and Meng [103] proposed a semi-automatic method for clustering word into semantic classes, relying on the distributional hypothesis of meaning. They applied an iterative procedure for word clustering, resulting to a hierarchical class generation. Clustering was considered to be spatial and temporal. For spatial clustering the *KL* divergence was computed between the contextual probability distributions, in order to estimate the semantic similarity between words. The measure of Mutual Information was used for temporal clustering, estimating how much two consecutive words co-occur. An example of generated clusters is shown below.

SC_{14}: francisco, jose
TC_{15}: los angeles
TC_{16}: san SC_{14}

The clusters are labeled with SC_i or TC_i, denoting spatial and temporal clustering, respectively. Note that TC_{16} produce "san francisco" and "san jose" using the nonterminal SC_{14} which was generated in a previous iteration. Also, multiple merges were allowed for clusters of the same iteration. For example, (nashville, toronto), (nashville, tampa), and (detroit, nashville) can be merged into a single spatial cluster (nashville, toronto, tampa, detroit). Using this technique nonterminals with greater number of terminals can be quickly generated. However, this merging strategy was allowed to applied over a predefined, fixed number of clusters that were generated during the same iteration. Next, the clusters were post-processes by hand-editing, according to the following four operations: (a) some of the SC_i and TC_i tags were substituted by meaningful labels, such as month, etc., (b) some terminals were completed for some categories (for example the missing days in a cluster of week days were added), (c) clusters that belonged to the same semantic class were consolidated, and (d) irrelevant nonterminals and terminals were discarded. Thus, the nature of this approach is semi-automatic.

Pargellis et al. [79] extended the work of Siu and Meng [103] by proposing an unsupervised, iterative procedure for inducing semantic classes. The suggested system consists of three components: (a) a lexical phraser, (b) a semantic generalizer, and (c) a corpus parser. The lexical phraser identify frequently co-occurring words by applying a weighted point-wise mutual information measure. For example, consecutive words like "New York" are chunked into a single entry, e.g., "New_York". The second module, semantic generalizer, generates rules that map words (or previously induced classes) to semantic classes. The core idea of semantic generalizer is the distributional

hypothesis of meaning. The similarity between two words is computed according to their contextual distributions. Pargellis et al. experimented with four different metrics: (a) Kullback-Leibler, (b) Information-radious, (c) Manhattan-norm, and (d) Cosine similarity. A comparative study of these metrics about automatic induction can be, also, found in [80]. During a system iteration a metric outputs an ordered list of pairs that is ranked according to semantic similarity. Each pair is assumed to form a semantic class. A predefined, fixed number of top pairs are used for the induction of semantic classes during for every system iteration. After every generation of induction classes the module of corpus parser is run. All instances of the generated classes are substituted in the corpus with the corresponding semantic label. It should be noted that the class labels are artificial tags, without denoting the class concept. The generated by the lexical phraser chunks are retained if they were assigned to an induced class by the semantic generalizer. The whole procedure is repeated, as is depicted by Figure 3.4. The above system was applied in four different corpora. The first three were about

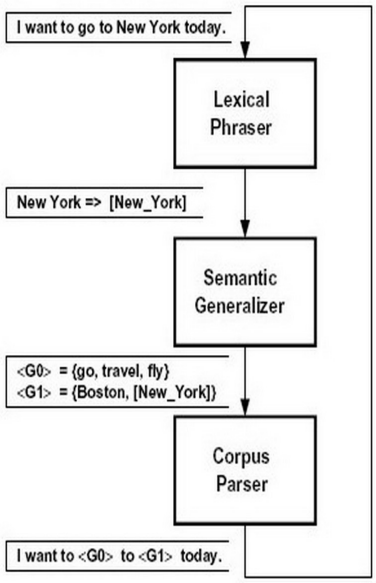

Figure 3.4: Auto-induced semantic classes system (Pargellis et al., [79]).

small, homogeneous domains dealing with movie information, travel information, and a children's computer game. The fourth corpus was about a heterogeneous domain, containing sentences from the Wall Street Journal. In addition, a longer-term goal of the work of Pargellis et al. is the study of similarities of different domains by identifying which semantic classes are related across different domains. A previous study of domain similarities can be found in [81].

More recently, Pangos et al. [77], proposed an algorithm for unsupervised semantic

class induction which is based on the hypothesis that similarity of context implies similarity of meaning. Two semantic similarity metrics that are variations of the cosine similarity distance were used in order to measure the semantic distance between words and to automatically generate semantic classes. The first metric computes "wide-context" similarity between words using a "bag-of-words" model, while the second metric computes "narrow-context" similarity using a bigram language model. A hybrid metric was proposed as the linear combination of the wide and narrow-context metrics, using fixed weights, estimated over held out data during an a priori experimental procedure. The semantic metrics were evaluated on two corpora: a semantically heterogeneous web news domain, HR-Net [36], and an application-specific travel reservation corpus, ATIS [86].

3.3 Proposed method

In this chapter, we focus on the problem of adaptive unsupervised weight estimation for combining multiple similarity metrics. The class induction system proposed in [77] works iteratively, updating a hierarchical structure of semantic classes in each iteration. Our motivation for creating an adaptive weighting scheme, is that the relative performance of each metric (in terms of precision and recall) varies from iteration to iteration. It is expected that by updating the weights of each metric at each iteration the combined metric performance can significantly improve. In addition, our goal is to create a fully unsupervised, corpus-independent metric combination algorithm that *does not* require experiment on held-out data to compute the weights. Next, we propose a fully unsupervised adaptive algorithm for combining semantic similarity metrics.

As in [77], we follow an iterative procedure for automatic induction of semantic classes, consisting of two main components: a *class generator* and a *corpus parser*. The *class generator*, explores the context information of every word, calculating the similarity between words; the semantic similarity metric combines two or more variations of the Cosine similarity metric. Semantically similar words or concepts are grouped together into classes. The *corpus parser*, re-parses the corpus using the class definitions generated by the *class generator*, i.e., substitutes all instances of each class member with the corresponding class label. The *class generator* and *corpus parser* are run sequentially and iteratively over the corpus.

3.3.1 Cosine Similarity Metrics

Our approach relies on distributional hypothesis of meaning: similarity of context implies similarity of meaning. We assume that words, which are similar in contextual distribution, have a close semantic relation [79, 96]. Both narrow- and wide-context is taken into account as described next.

Wide-context metric: Bag-of-words model

In "Bag-of-words" [101] models, for each word w in the vocabulary a context window size WS is selected. The right and left contexts of length WS in the corpus are considered for word w, e.g.,

$$[v_{WS,L} \quad ... \quad v_{2,L} \quad v_{1,L}] \quad w \quad [v_{1,R} \quad v_{2,R} \quad ... \quad v_{WS,R}]$$

where $v_{i,L}$ and $v_{i,R}$ represent the i^{th} word to the left and to the right of w respectively. The feature vector for every word w is defined as $T_{w,WS} = (t_{w,1}, t_{w,2}, ..., t_{w,N})$ where $t_{w,i}$ is a non-negative integer and WS is the context window size. Note that the feature vector size is equal to the vocabulary size N, i.e., we have a feature for each word in the vocabulary V. The i^{th} feature value $t_{w,i}$ reflects the occurrences of vocabulary word v_i within the left or right context window WS. This feature value is set according to a Binary (Bin.) or a Term Frequency (Freq.) Scheme.

This feature value is set according to a Binary (Bin.) or a Term Frequency (Freq.) Scheme:

Binary Scheme:

$$t_{w,i} = \begin{cases} 1, & \text{if} \quad v_i \in N_{WS,LR} \\ 0, & \text{if} \quad v_i \notin N_{WS,LR} \end{cases}$$

Term Frequency Scheme:

$$t_{w,i} = \begin{cases} f(v_i|WS), & \text{if} \quad v_i \in N_{WS,LR} \\ 0, & \text{if} \quad v_i \notin N_{WS,LR} \end{cases}$$

where $f(v_i|WS)$ is frequency of v_i within a left and right window context of size WS for the word w and for the whole corpus.

The "Bag-of-words" metric, $S_{A,WS}$, using Binary or Term Frequency Scheme, measures the similarity of two words, w_1 and w_2, as the cosine similarity of their corresponding feature vectors, $T_{w_1,WS}$ and $T_{w_2,WS}$ [77]:

$$S_{A,WS}(w_1, w_2) = \frac{\sum_{i=1}^{N} t_{w_1,i} t_{w_2,i}}{\sqrt{\sum_{i=1}^{N} (t_{w_1,i})^2} \sqrt{\sum_{i=1}^{N} (t_{w_2,i})^2}} \tag{3.3.1}$$

given a context window of length WS.

Narrow-context metric: N-gram language model

In an N-gram language model a word w is considered with its neighboring words $v_{1,L}$ and $v_{1,R}$ in the left and right contexts within a sequence. In order to calculate the similarity of two words, w_1 and w_2, we compute the cosine distance between two feature vectors; each feature vector of a word w measures the conditional probability of all possible contexts v_i given that word $p(v_i|w)$, i.e., each vector contains bigram language model probabilities for (context, word) pairs. Semantic similarity is defined as

$$S_B(w_1, w_2) = S_B^L(w_1, w_2) + S_B^R(w_1, w_2) \tag{3.3.2}$$

where the two terms of Eq. (3.3.2) are [79]:

$$S_B^L(w_1, w_2) = \frac{\sum_{i=1}^{N} p(v_{i,L}|w_1) p(v_{i,L}|w_2)}{\sqrt{\sum_{i=1}^{N} p(v_{i,L}|w_1)^2} \sqrt{\sum_{i=1}^{N} p(v_{i,L}|w_2)^2}} \tag{3.3.3}$$

$$S_B^R(w_1, w_2) = \frac{\sum_{i=1}^{N} p(v_{i,R}|w_1)p(v_{i,R}|w_2)}{\sqrt{\sum_{i=1}^{N} p(v_{i,R}|w_1)^2}\sqrt{\sum_{i=1}^{N} p(v_{i,R}|w_2)^2}} \tag{3.3.4}$$

where $V = (v_1, v_2, ..., v_N)$ is the vocabulary set, and $p(v_i|w)$ is the conditional probability of word v_i preceding w in the corpus given word w, i.e., the (v_i, w) bigram model probability.

3.3.2 Generating Word Classes

Similarity metric $S_{A,WS}$ or S_B output a list of pairs, ranked according to the semantic similarity of their members, from semantically similar to semantically dissimilar. Words and semantic classes (induced in previous system iterations) are valid members of such pairs. From this list that contains all possible word pairs in the corpus, one has to choose a fixed number of top ranking pairs in order to induce the next set of semantic classes. In [79], a new class label is created for each pair and the two members are assigned to the new class. However, there is no way to merge more than two lexical units at one step which may lead to a large number of hierarchically nested classes. In [103], a more insightful clustering algorithm is proposed generating multiple class merges. However, the number of class merges is predefined and remains constant for all system iterations.

We extend the latter algorithm by allowing varying number of class merges [38], described next. The algorithm examines multiple pairs and finds those pairs that have a common element. Provided that certain conditions are met, a new class label is created and the union of these pairs is assigned to this class. Assume that the pairs (one,two), (one,ten), (two,twenty) were ranked at the upper part of the list. According to the proposed algorithm, the class (one,two,ten,twenty) will be created. To avoid over-generalizations only pairs that are rank ordered close to each other are allowed to participate in this process. The parameter "Search Margin", SM, defines the maximum distance between two pairs (in the semantic distance rank ordered list) that are allowed to be merged in a single class. Consider the ranked pairs of Table 3.1. For $SM = 2$ the classes (one,two,ten) and (monday,friday,sunday) will be generated,

Position in List	1	2	3	4	5
Pairs	one two	two ten	monday friday	friday sunday	ten twenty

Table 3.1: Top ordered pairs, ranked according to semantic similarity.

while for $SM = 3$ the classes (one,two,ten,twenty) and (monday,friday,sunday) will be generated. By adding the search margin SM constraint it was observed that the semantic homogeneity of the created classes was better preserved.

3.3.3 Combined Similarity Metrics

Combining similarity metrics of various context lengths makes it possible to utilize multiple lexical scopes of the contextual information. In [77], two different metrics

were linearly combined using fixed weights: $S_{A,WS=50}$ with broad lexical scope and S_B that focuses on the immediate context of each word.

Our goal is to improve on the combined metric by adding multiple context lengths and, most importantly, to automatically and adaptively estimate the weights assigned to each contributing individual metric. We propose the following weighted linear combination:

$$C(w_1, w_2) = \sum_{m=1}^{M} \lambda_m S_m(w_1, w_2) \tag{3.3.5}$$

where S_m can be any of $S_{A,WS}(w_1, w_2)$ or $S_B(w_1, w_2)$ (for various context lengths WS). Note that λ_m varies from iteration to iteration and that

$$\sum_{m=1}^{M} \lambda_m = 1 \tag{3.3.6}$$

Unsupervised Weight Estimation

The hybrid metric C takes into account different metrics with different lexical scopes using a weighted linear combination. The weight λ_m estimation algorithm is motivated by the work in [85], where it is shown that the "optimal" weights for the combined classifier should be approximately inversely proportional to the classification error rate of each (stand-alone) classifier. In order to have an unsupervised weighting metric, however, we need some estimate of the individual classification error rates of the component metrics. Assuming equal priors (and variance normalized data) the classification error rate can be assumed to be approximately proportional to the inter-class similarity (or inversely proportional to the inter-class distance). Thus the optimal weights should be *inversely proportional to the inter-class similarity*. This agrees with our intuition that greater importance should be given to the individual metric S_m that achieves better class separability.

Consider that (at iteration I) the metric S_m generates a number of classes $c_{i,m}$ from the top NP ranking word pairs; where i is the class index for metric S_m. The quality of class induction for each metric at iteration I is measured by employing a criterion of inter class similarity. Specifically, the *inter class similarity* between two classes, c_i and c_j generated by S_m is computed as:

$$D_{i,j,m} = \frac{\sum_{w_k \in c_{i,m}} \sum_{w_l \in c_{j,m}} S_m(w_k, w_l)}{\mid c_{i,m} \mid \mid c_{j,m} \mid} \tag{3.3.7}$$

where $\mid . \mid$ denotes set cardinality. The *average inter class similarity* $D_{m,avg}$ for metric S_m is computed by averaging over all similarity scores between all possible pairs of classes (i, j):

$$D_{m,avg} = \langle D_{i,j,m} \rangle_{(i,j)}. \tag{3.3.8}$$

Finally, the combination weight λ_m assigned to S_m is equal to the inverse of the average inter class similarity:

$$\lambda_m = \frac{1}{\mu_m D_{m,avg}} \tag{3.3.9}$$

Note that μ_m is a smoothing factor and is a moving average of $D_{m,avg}$ over all past system iterations.

3.4 Experimental corpora and procedure

The first corpus we experimented with was the semantically heterogeneous HR-Net corpus that was downloaded from the Hellenic Resources Network [36]. Specifically, news in English from the Hellenic Radio (ERA) between 01/01/2005 and 05/11/2005 were gathered. HTML tags were removed from each document. The number of articles in the corpus is 2,082, the total number of words is 549,660, the size of the vocabulary is 22,904 words, the average number of words per document is 264, the maximum number of words found in a document is 1,495 and the minimum 41. The second corpus we experimented with was the domain specific ATIS corpus [86]. This corpus contains transcribed utterances dealing with travel information. We used an experimental corpus consisting of 1,705 utterances. The total number of words is 19,197 and the size of vocabulary is 575 words [77].

3.4.1 Corpora samples

For example purposes we present a political article from the HR-Net corpus:
"at least people were injured yesterday during violent clashes that erupted in a refugee camp south of the gaza strip between activists of the palestinian groups fatah and hamas the conflict took place in the camp al mugazi when fatah members who had gathered to celebrate their partys victory in the area in the recent municipal elections met with a group of hamas members who were also celebrating their groups victory in the gaza strip according to hospital sources five of the injured people had gunshot wounds meeting between sharon and abbas in the meantime israeli pm ariel sharon and palestinian president mahmoud abbas are expected to meet sometime in the next two weeks for the first time since abbas came to office as per sharons spokesperson david baker the meeting will take place during the second week of february and the talks would aim to make progress between both sides contingent on continued efforts by the palestinians to prevent terrorism to israel".

Also, some example utterances from the ATIS corpus are shown below.

do you have a flight from charlotte to atlanta on june first
do you have flights from saint petersburg to toronto on monday
how much does it cost to rent a car in tacoma

The HR-Net corpus is semantically most diverse, compared to the ATIS corpus, because contains news articles. In contrast, ATIS includes short utterances of limited length and vocabulary that are oriented to the needs of a specific task. For both corpora, punctuation and letter capitalization were not taken into consideration during the experiments.

3.4.2 Experimental steps and parameters

The following experimental steps are performed:

1. One or more variations of wide-context metric $S_{A,WS}$ is/are calculated.
2. Narrow-context metric S_B is calculated. For the S_B metric, the Bigram Language Model was built using the CMU Statistical Language Modeling toolkit, applying Witten-Bell discounting and using back-off weights to compute the probability of unseen bigrams.
3. The results of each of the above metrics are normalized using min-max normalization.
4. The λ weights are computed using Equation (3.3.9).
5. The hybrid C metric is calculated and semantic classes are induced according to the algorithm described in Section 3.3.2.
6. All occurrences of the derived class members in the corpus are substituted by the corresponding class label.
7. The above procedure is repeated until the specified number of iteration SI is reached.

The following experimental parameters must be defined:

1. The context window WS for $S_{A,WS}$ as well as the scheme used, Binary (Bin.) or Frequency (Freq.) as described in Section 3.3.1.
2. The total number of system iterations (SI).
3. The number of induced semantic classes per iteration (IC).
4. The size of Search Margin (SM) defined in Section 3.3.2.
5. The number of pairs NP considered for inter-class similarity and weight computation as discussed in Section 3.3.3.

3.5 Evaluation

3.5.1 Benchmark

In order to evaluate the induced semantic classes for the HR-Net corpus, we used as a benchmark a taxonomy of 43 semantic classes including 1,820 word-members, manually crafted by two researchers. Table 3.2 illustrates 5 representative handcrafted classes along with example members. Every word was assigned only to one hand-crafted class and our system was tested only for these 1,820 words.

For the evaluation procedure of the ATIS corpus, we used a manually crafted semantic taxonomy, consisting of 38 classes that include a total of 308 members. Every word was assigned only to one hand-crafted class. Table 3.3 shows 5 representative handcrafted classes along with some members. For experimental purposes, we generated manually characteristic "word chunks", e.g., T W A → T_W_A. Also, for the ATIS experiments, all 575 words in the vocabulary were used for similarity metric computation and evaluation.

Class Name	Members
Education	university, school, student...
Politics	Karamanlis, president, minister...
Law	prosecutor, judge, crime...
Health	hospital, surgery, pharmaceutical...
Sports	Olympiacos, UEFA, Rehhagel ...

Table 3.2: Example of semantic classes from HR-Net benchmark

Class Name	Members
Fairtype	one_way, round_trip, nonstop...
City	Los_Angeles, Boston, New_York...
Airline	Delta, Lufthansa, T_W_A...
Meal	meal, lunch, breakfast ...
Day	Monday, Friday, Sunday...

Table 3.3: Example of semantic classes from ATIS benchmark

3.5.2 Evaluation measurements

For both corpora the evaluation focused only on the terminal semantic classes (hierarchical class generation was not evaluated). Every induced class was evaluated with respect only to the corresponding handcrafted class without examining its relationships with other classes over the taxonomy. An induced class is assumed to correspond to a handcrafted class, if at least 50% of its members are included ("correct members") in the handcrafted class. Precision and recall are calculated as follows:

$$\text{precision} = \frac{\sum_{i=1}^{m} c_i}{\sum_{i=1}^{m} \alpha_i} \tag{3.5.1}$$

$$\text{recall} = \frac{\sum_{i=1}^{m} c_i}{\sum_{j=1}^{r} \beta_j} \tag{3.5.2}$$

where m is the total number of induced classes, r is the total number of handcrafted classes, c_i is the "correct members" of the i^{th} induced class, α_i is the total number of members of the i^{th} induced class and β_j is the total number of members of the j^{th} handcrafted class that occur in the corpus.

3.5.3 Experimental Results

In Figure 3.5(a), the cumulative precision achieved by the metrics $S_{A,WS}$, S_B and their combination $\lambda_1 S_{A,WS=3} + \lambda_2 S_B$, is shown for the ATIS corpus (solid line). The weights for the combined metric are computed adaptively at each iteration using Equation 3.3.9; the weights are shown in Figure 3.5(b). For this experiment we used the following

parameters: Freq. Scheme for $S_{A,WS=3}$, $SI = 20$, $IC = 10$, $SM = 5$, $NP = 50$. Note that the cumulative precision of the combined metric with fixed weights $\lambda_1 = 0.35$ and $\lambda_2 = 0.65$ is also shown in Figure 3.5(a) (dotted line). It is interesting to note that the adaptive weighting schemes significantly outperforms the fixed weighting scheme in this experiment. The precision achieved for the metric $S_{A,WS=3}$ is good in the first few iterations but quickly decreases well below the precision of the S_B metric. In general, the ATIS corpus favors the narrow-context metrics.

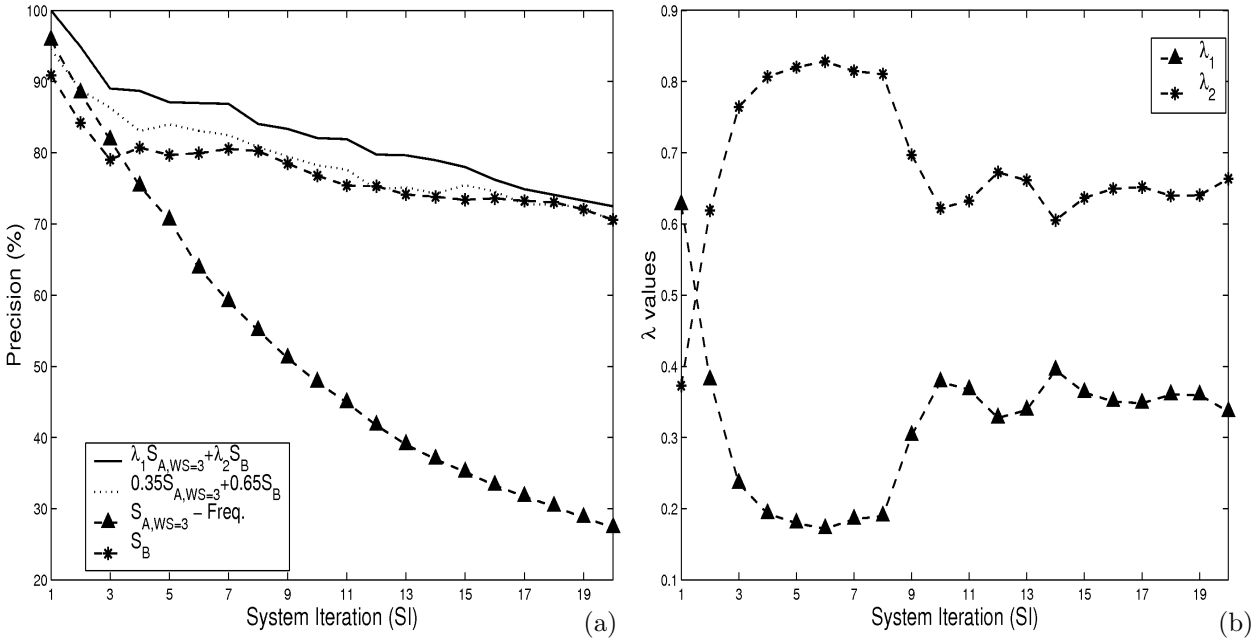

Figure 3.5: (a) Cumulative precision for the ATIS task for two individual metrics and two combined metrics (adaptive vs fixed weighting scheme), (b) Assigned weights for the ATIS task (adaptive weighting scheme).

The adaptive weights $\lambda_{1,2}$ in Figure 3.5(b) take reasonable values; as the precision of $S_{A,WS=3}$ relative to S_B decreases so does λ_1 relative to λ_2. The increase in the value of λ_1 after iteration 8 could be due to sparse data for weight estimation. Overall, at iteration $SI = 20$ the adaptively weighted combined metric generates 33 classes with 224 members achieving a recall of 72.7% (see Table 3.4).

In Figure 3.6(a), the cumulative precision achieved by three variations of $S_{A,WS}$ (Binary scheme, window sizes 50, 10, and 2), and their combinations is shown for the HR-Net corpus. Two combined metrics are shown both using adaptively computed weights: $C = \lambda_1 S_{A,WS=50} + \lambda_2 S_{A,WS=10} + \lambda_3 S_{A,WS=2}$, and $C' = \lambda'_1 S_{A,WS=50} + \lambda'_2 S_{A,WS=10}$. Note that: $SI = 20$, $IC = 10$, $SM = 10$, $NP = 50$ for this experiment. The three metric combination C significantly outperforms the two metric combination C' in terms of precision. Also note that C outperforms the best of the $S_{A,WS}$ metrics and achieves up to 50% relative error rate reduction. The semantically heterogeneous nature of HR-Net corpus allows multiple metrics of different lexical scopes to be combined successfully.

In Figure 3.6(b), the assigned weights for the three-metric combination C are shown. During the very early iterations $S_{A,WS=10}$ is weighted most, but after the 6^{th} iteration

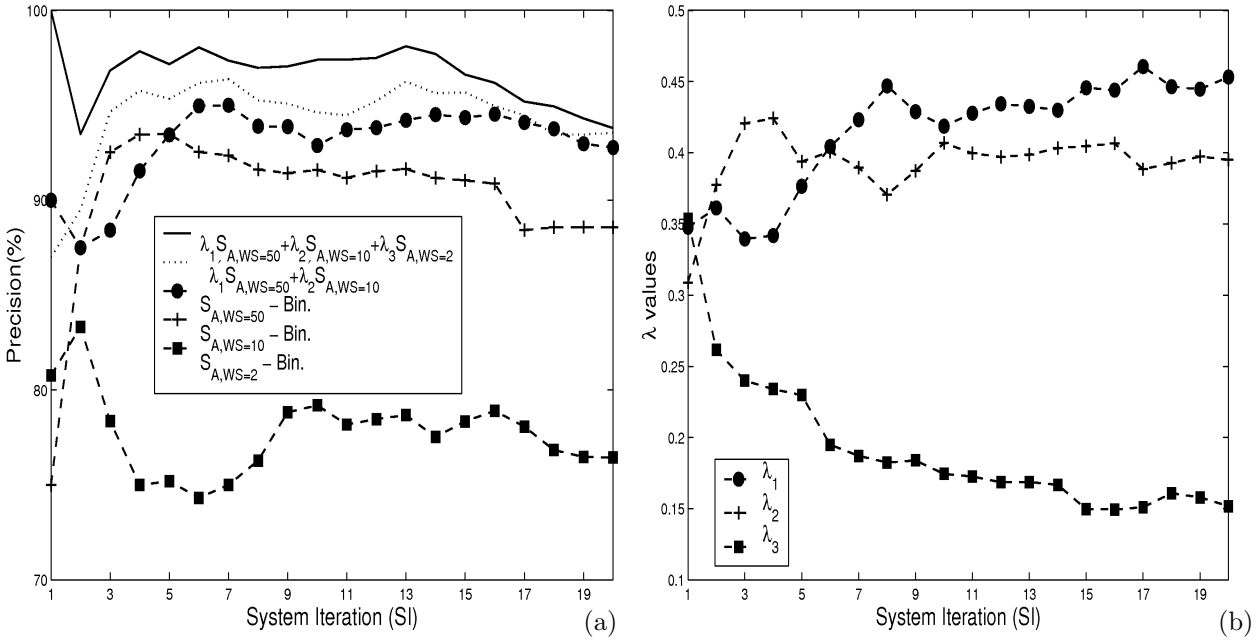

Figure 3.6: (a) Cumulative precision for the HR-Net task for three individual and two combined metrics (three-way and two-way adaptive combination), (b) Assigned weights for the three-way combination metric for the HR-Net task.

$S_{A,WS=50}$ is assigned the greatest weight. The metric with the smallest window size, $S_{A,WS=2}$, is given the lowest λ value.

This is consistent with the experiments in [77] showing that the wide-context metrics outperform the narrow-context ones for the semantically heterogeneous HR-Net corpus. It is interesting to note that when the proposed unsupervised weight computation algorithm is used there is no need to select metrics based on corpus characteristics. Instead a corpus independent combined metric can be used and automatically poor performing metrics will be weighted less in the combination. This is consistent with the corresponding precision curves and agrees with our intuition. Note that at iteration $SI = 20$ the combined metric C generates 20 classes with 304 members and achieves precision of 16.7%.

The following table shows the recall across different experiments. We can see that the adaptive weightings are relatively close to the fixed parameters in terms of recall, so the improvement in precision is not at the expense of recall.

3.6 Conclusions

We have presented an algorithm for unsupervised computation of weights to individual metrics of different lexical scopes. The metrics are linearly combined into a hybrid metric. The proposed algorithm monitors the efficiency of each individual metric and attempts to assign greater weight to the "best-performing" metric. Experiments on two different corpora showed that the adaptively computed weights outperform the

Recall at SI(%)	1	5	10	15	20
ATIS:adaptive $\lambda_{1,2}$	11	31.2	46.1	64	72.7
ATIS:fixed $\lambda_{1,2}$	12.7	33.1	49.7	65.3	75
ATIS:$S_{A,WS=3}$	14.9	34.4	44.8	53.2	63.3
ATIS:S_B	6.5	30.2	45.8	59.4	70.1
HR-Net:adaptive $\lambda_{1,2,3}$	1.3	5	9.8	13.9	16.7
HR-Net:adaptive $\lambda'_{1,2}$	0.8	5	9.6	13.3	16.6
HR-Net:$S_{A,WS=50}$	1	5.7	9.5	13.5	16.9

Table 3.4: Recall scores of adaptive and individual metrics (for ATIS and HR-Net corpus).

fixed weight computation scheme. Also three metric combination (wide-, mid- and narrow-context size) significantly outperformed each one of the individual metrics in terms of precision and recall of generated classes. The proposed unsupervised metric combination algorithm makes it possible to employ a *corpus-independent semantic similarity* metric for semantic class induction. Future work will investigate how to include estimation error variance is the weight estimation criterion as discussed in [85]. Combination of other types of semantic similarity measures it is, also, interesting to be investigated.

Chapter 4

A Soft-Clustering Algorithm for Automatic Induction of Semantic Classes

4.1 Introduction

In the previous chapter we described a method of word hard clustering, according to which a word is deterministically assigned to an induced semantic class. After a word assignment , the word is substituted by a class label and this representation is used during the next iterations. The hard clustering approach is used by almost all the related works [103, 80, 79, 77] appeared in the literature. However, the single class assignment suffers from some drawbacks:

- Other possible categorizations are ignored.
- Erroneous word assignments are retained throughout the iterative procedure of the system. So, induced classes of later iterations may include previous wrong assignments.
- The lexical type of a categorized word is not considered, since the word was substituted by a class label. In other words, a fraction of the corpus lexical information is eliminated.
- As the words are substituted by class labels, the semantic generalizer form new classes containing nonterminals and the system tends to overgeneralize classes.

Our approach [39]: in order to alleviate the disadvantages of the hard word clustering, we propose a soft-decision, unsupervised clustering algorithm that generates semantic classes automatically using the probability of class membership for each word, rather than deterministically assigning a word to a semantic class. Also, every word retains its lexical type. This is motivated by the idea that the presence of lexical and semantic information can contribute to the better estimation of probabilities, in order to automatically induce semantic classes. Semantic classes are induced using an unsupervised, automatic procedure that uses a context-based similarity distance to

measure semantic similarity between words. The proposed soft-decision algorithm is compared with various "hard" clustering algorithms, evaluated on a travel reservation corpus, ATIS [86]. Our approach is presented in Sections 4.3 - 4.6.

4.2 Related work

To our knowledge only one data-driven approach has been proposed in the literature that automatically clusters words into more than one class [82, 58]. Pereira et al. [82] proposed a method for hierarchical clustering of words according to their distribution in particular syntactic contexts. They considered transitive main verbs and the head nouns of their direct objects, working with articles form the Associated Press. The clustering procedure follows a deterministic annealing, in which a parameter β is continuously being increased. Initially, the parameter β is set to a very low value and a single cluster is formed whose centroid is the average of all noun distributions. Next, the value of β is slowly increased. For any β there is a number of leaf clusters that corresponds to a minimum of a so-called *free energy function*. This function incorporates the *KL* divergence between noun and centroid distributions and the cluster membership entropy. For some leaf clusters there is a critical, lowest β value for which the particular cluster an be split. In general, the splitting procedure can be repeated until a desired number of clusters or model cross-entropy is reached. Figure 4.1 shows

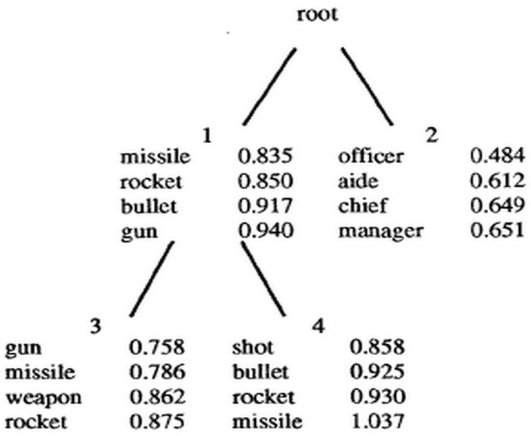

Figure 4.1: Soft clustering using free energy function (Pereira et al., [82]).

an experiment dealing with the 64 most frequent nouns appeared as heads of sirect objects for the verb "fire". For each cluster are shown the four most similar words to the cluster centroid. Cluster 1 corresponds to the weaponry sense of verb "fire", while cluster 2 reflects refers to the corresponding personnel. Regarding the split of cluster 1, note that words "missile" and "rocket" participate in cluster 3, as well as in cluster4. This clustering method was evaluated using two types of measurements: (a) relative model entropy, and (b) performance for the procedure of deciding which of two verbs

is more likely to have a noun as a direct object.

A remarkable portion of research efforts regarding word clustering adopts the perspective of class-based language modeling. The following reference to works about word clustering is, also, closely related to Chapter 3. However, in this chapter some major works are briefly described from a clustering-oriented point of view. This is because the main idea of the proposed method [39] of this chapter deals with word soft clustering. In contrast, Chapter 3 was focused to the unsupervised combination of similarity metrics [41], while the word hard clustering method was treated as an issue of minor research interest.

The first thorough study of class-based N-gram language modeling was conducted by Brown et al. [11]. Initially each word was assigned into a distinct class. Next, the average mutual information was computed among adjacent classes. All possible merges between pairs of classes were tested, and the merge that gave the least loss in average mutual information was retained. After generating a set of classes by successive merges, some words were moved from one class to another. For these word movings the mutual information was computed for the corresponding class. In final, the reassignment of a word terminates by finding a cluster with greater mutual information. This procedure results to a tree structure as is shown in Figure 4.2. The constructed class-based model

Figure 4.2: Word clustering using mutual information (Brown et al., [11]).

was interpolated with a word-based model. The perplexity of the interpolated model over the Brown corpus was reported to be equal to 236, while the perplexity of the word-based model was 244.

In [3], the approach of Brown et al. [11] was tested over a newspaper texts from Peole's Daily and compared with a method that used minimum discrimination information as a distance metric. This metric was used in the place of mutual information in order to find the least loss in average mutual information among pairs of classes. The suggested in [3] method was reported to give perplexity reduction comparable to the results of [11], while it was shown to be computationally cheaper.

The work of Bellegarda et al. [6] proposes a framework for word clustering formulated from the perspective of information retrieval, applying latent semantic analysis. While traditional approaches consider word co-occurrence within narrow contexts, Bellegarda et al. search for word co-occurrence at document level. A document is assumed as a semantically homogeneous set of sentences. A matrix of co-occurrences between words and articles from ARPA North American Business News corpus was constructed.

Singular value decomposition was applied, resulting to singular vector representation for words and documents. Next, a distance metric was defined over the generated vector space. In final, the singular vectors were clustered according to their distance. In particular, the K-means algorithm was used to partition the vocabulary into a small set of superclusters. Afterwards, bottom-up clustering was applied for each supercluster. However, no evaluation results were reported for work, but only some interesting observations about the semantic nature of the resulting clusters, were stated.

In [75], various class-based N-gram language models are interpolated with word-based models. Classes correspond to part-of-speech, as well as to automatically generated groupings. In general, the interpolated models had better performance regarding perplexity reduction and worrd error rate improvement.

4.3 Proposed method

Almost all of the above approaches for automatic induction of semantic classes assign deterministically a word into a particular induced class. In this chapter, we propose an iterative soft-decision, unsupervised clustering algorithm, which instead of deterministically assigning a word to a semantic class computes the probability of class membership in order to generate semantic classes. The proposed soft clustering algorithm is compared to the hard clustering algorithm, used in [103, 79, 41]. Various other "hard" clustering algorithms are also evaluated that use lexical-only, or lexical and (auto-induced) semantic information for deriving class estimates. Next, we briefly describe the "hard" clustering procedure, in order to be easily compared with the "soft" clustering algorithm that follows.

4.3.1 Hard clustering algorithm

We follow a fully unsupervised, iterative procedure for automatic induction of semantic classes, consisting of a *class generator* and a *corpus parser* [79]. The *class generator*, explores the immediate context of every word (or concept) calculating the similarity between pairs of words (or concepts) using KL metric. The most semantically similar words (or concepts) are grouped together generating a set of semantic classes. The *corpus parser*, re-parses the corpus and substitutes the members of each induced class with an artificial class label. These two components are run sequentially and iteratively over the corpus (the process is similar to the one shown in Figure 4.4 but with a hard decision in step II).

Class generator

Our approach relies on the idea that the similarity of context implies similarity of meaning [96]. We assume that words, which are similar in contextual distribution, have a close semantic relation [103, 79]. A word w is considered with its neighboring words in the left and right contexts within a sequence:

$$\ldots \quad w_1^L \quad w \quad w_1^R \quad \ldots$$

In order to calculate the distance of two words, w_x and w_y, we use a relative entropy measure, *Kullback-Leibler* (*KL*) distance, applied to their conditional probability distributions [103, 79, 82]. For example, the left *KL* distance is

$$KL^L(w_x, w_y) = \sum_{i=1}^{N} p(w_i^L|w_x) \log \frac{p(w_i^L|w_x)}{p(w_i^L|w_y)} \tag{4.3.1}$$

where $V = (w_1, w_2, ..., w_N)$ is the vocabulary set. The similarity of w_x and w_y is estimated as the sum of the left and right context-dependent symmetric KL distances:

$$\begin{aligned} KL(w_x, w_y) = KL^L(w_x, w_y) + KL^L(w_y, w_x) + \\ + KL^R(w_x, w_y) + KL^R(w_y, w_x) \end{aligned} \tag{4.3.2}$$

If w_x and w_y are lexically equivalent, then $KL(w_x, w_y) = 0$.

Using distance metric *KL* the system outputs a ranked list of pairs, from semantically similar to semantically dissimilar. The semantic classes are built as it was described in Section 3.3.2.

Corpus parser

The corpus is re-parsed after the class generation. All instances of each of the induced classes are replaced by a class label. Suppose that the words "Noon" and "LA" are categorized to the classes <time> and <city>, respectively.[1] The sentence fragment "Noon flights to LA" becomes "<time> flights to <city>". After the corpus re-parsing, the algorithm continues to the next system iteration. So, the lexical type of the clustered words is substituted by semantics and it is not present anymore in the corpus during the next iterations.

4.3.2 Soft clustering algorithm

The previously described hard clustering algorithm suffers from some drawbacks. First, a word is deterministically assigned to only one induced class. This isolates the word from additional candidate semantic classes. Furthermore, if the word categorization is false, then the erroneous induced class is propagated to the next class generation via the next iterations, leading to cumulative errors. Also, as the corpus is re-parsed, lexical information is being eliminated and substituted by the imported auto-induced semantic tags. This it is likely to produce fallacious semantic over-generalizations [79].

We propose a fully unsupervised, iterative *soft clustering algorithm* for automatic induction of semantic classes. The proposed algorithm follows a similar procedure as the hard clustering algorithm but alleviates the aforementioned disadvantages. A word is soft-clustered to more than one induced class according to a probabilistic scheme of membership computation thus reducing the impact of classification errors. In addition, the lexical nature of the corpus is preserved by equally weighting lexical and derived semantic information in the distance computation metric. Thus the soft clustering algorithm combines both lexical and induced semantic information as explained next.

[1] The algorithm has no concept of these classes and the above labels are used only for example reasons. In practice alphanumerics are used for each semantic class as it is created.

Soft class n-gram language model

Recall the example of Section 4.3.1 where the words "Noon" and "LA" are categorized to the classes <time> and <city>, respectively. The key idea of the proposed soft clustering algorithm allows words to belong to more than one induced semantic classes. In Figure 4.3 each word that belongs to multiple classes is represented by multiple

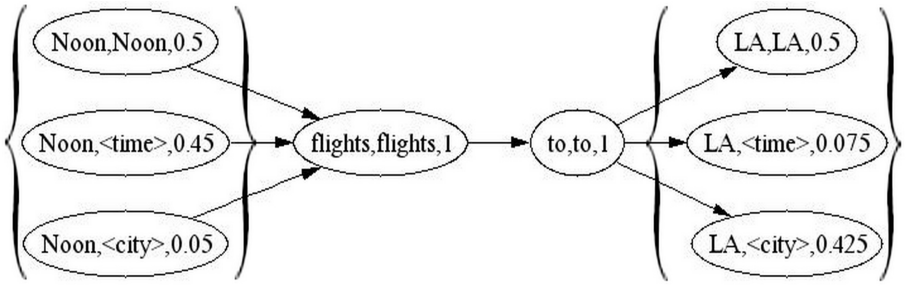

Figure 4.3: Sentence fragment with multiple semantic representations after the 1^{st} iteration of class induction.

triplets $(w_i, c_j, p(c_j|w_i))$, where w_i is the word itself, c_j is the label of an induced semantic class (concept) and $p(c_j|w_i)$ is the *probability of class membership*, that is the probability of word w_i being member of class c_j, defined next. This "soft" class assignment shown in Figure 4.3 is represented as (Noon,<time>,0.45), (Noon,<city>,0.05) and (LA,<time>,0.075), (LA,<city>,0.425). Note that it is not required that all words are assigned to classes; the multi-class soft-assignment criterion will be discussed next. Additionally, for all corpus words we retain the lexical form; for each word w_i there is an (additional) triplet $(w_i, w_i, p(w_i|w_i))$ with fixed probability $p(w_i|w_i) \equiv 0.5$, e.g., (Noon,Noon,0.5). By design the probability mass is equally split between the lexical and semantic information, i.e., for each word the sum of class membership probabilities over all classes is equal to 0.5 and equal to the probability of the word retaining its lexical form.[2]

A number of semantic classes are generated at every system iteration. We define the set S^n of induced classes generated up to the n^{th} iteration, the corpus vocabulary set V containing all words, and their union $C^n = S^n \bigcup V$. Using the above definitions we propose an n-gram language model for the class labels and words, elements of set C^n. The maximum likelihood (ML) unigram probability estimate for c_j is

$$\left(\hat{p}(c_j)\right)_{ML} = \frac{\sum_{\forall w_i \in V} p(c_j|w_i)}{\sum_{\substack{\forall w_i \in V \\ \forall c_j \in C^n}} p(c_j|w_i)} \tag{4.3.3}$$

i.e., the sum of class membership probabilities of every vocabulary word with respect to c_j. The corresponding maximum likelihood estimate for the bigram probability of a

[2] Note that, as shown in Figure 4.3, words that are not (yet) candidates for any semantic class have a lexical form probability equal to one, e.g., (flights, flights,1).

sequence c_j, c_{j+1} is

$$\Big(\hat{p}(c_{j+1}|c_j)\Big)_{ML} = \frac{\sum_{\forall w_i \in V} p(c_{j+1}|w_i)p(c_j|w_i)}{\sum_{\forall w_i \in V} p(c_j|w_i)} \tag{4.3.4}$$

In the case of unseen bigrams we use the back-off language modeling technique [43, 48] to estimate the bigram probability as:

$$\hat{p}(c_{j+1}|c_j) = \mathit{backoff}(c_j)\hat{p}(c_{j+1}) \tag{4.3.5}$$

The proposed soft class language model is built on both lexical and semantic context and differs somewhat from traditional class-based language models [52, 11].

Induction of semantic classes

The proposed system using the soft clustering algorithm works similarly to the hard clustering system, with the addition of the class membership calculation algorithm. The soft-clustering algorithm consists of three steps *class generator*, *membership calculator* and *corpus parser*, as shown in Figure 4.4. At the last iteration, words are (hard-)assigned to classes using the same rules for both the soft- and hard-clustering algorithms (thus, the main difference between the two algorithms is in the better estimation and smoothing of the contextual probabilities in the soft-clustering case). An example 1^{st} system iteration is also shown in this figure.

Class generator

First, each corpus word is transformed to the triplet format. Second, a soft class n-gram language model is built, as defined before. Then the *KL* distances between words are computed according to Equations 4.3.1 and 4.3.2. Note that the probabilities are computed using the generalized n-gram estimation Equations 4.3.3, 4.3.4 and 4.3.5. Next a set of semantic classes is generated using the pair merging strategy, described in Section 3.3.2. For each candidate class the class membership probability is computed using the *membership calculation* algorithm outlined next.

Membership calculator

Given the set of semantic classes S^n generated at the n^{th} system iteration, the probability of class membership between words and each class s_j of S^n is computed. This is not done for the entire corpus vocabulary, but only for the words that were assigned deterministically to the classes of S^n by the *class generator*. In other words, we relax the word-class hard assignment to word-classes soft assignment but otherwise keep the iterative process of word to class assignment as in the hard clustering algorithm. Let the words that are assigned to classes up to iteration n be members of a set $X^n \subset V$. Also, recall that each word member of X^n is retained (assigned to itself) with fixed probability equal to 0.5. The probability of class membership between a word, w_i, and a class (or itself), c_j, is given by the following equations:

$$p(c_j|w_i) \equiv 0.5, \tag{4.3.6a}$$

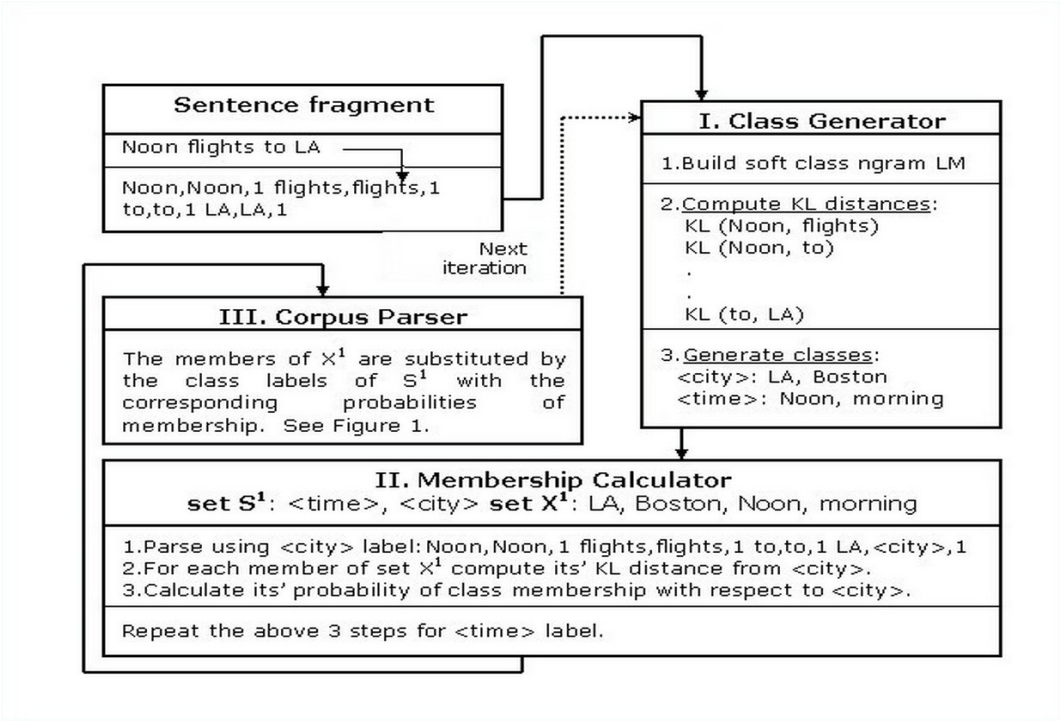

Figure 4.4: Soft clustering system architecture and example iteration.

if $c_j = w_i$ and $w_i \in V$, and

$$p(c_j|w_i) \equiv \frac{e^{-KL(s_j,w_i)}}{\sum_{\forall s_j \in S^n} e^{-KL(s_j,w_i)}}, \tag{4.3.6b}$$

where $c_j \in S^n$ and $w_i \in X^n \subset V$.

The KL distance between a word w_i and a class $c_j = s_j$ is computed as follows: (i) the corpus is parsed and each word in c_j (excluding w_i) is substituted by the appropriate class label, (ii) a bigram language model is built using Equations 4.3.3-4.3.5, and (iii) the KL distance is calculated using Equation 4.3.2. Then the equations above are applied to compute the probability of class membership.

The motivation behind Equation 4.3.6b is that words that are semantically similar to a class are member candidates for this class. The enumerator of Equation 4.3.6b distributes exponentially less membership probability mass to the classes that have greater KL distance from the word w_i . The exponential form of Equation 4.3.6b has more drastic separability regarding strong and weak class candidates compared to a linear equation. Equation 4.3.6b is a slightly modified reverse-sigmoid membership function, commonly used in fuzzy logic. Note that the total probability of class membership for

every soft-clustered word $w_i \in X^n$ equals to 1, i.e.,

$$\sum_{\forall c_j \in C^n} p(c_j|w_i) = \overbrace{\sum_{\forall s_j \in S^n} p(c_j = s_j|w_i) + p(c_j = w_i|w_i)}^{0.5+0.5} = 1, \tag{4.3.7}$$

where $w_i \in X^n$. The equation implies a linear, fixed combination between lexical and semantic information, which are equally weighted [3]. Every word of X^n is allowed to participate into the generated classes of S^n with membership probabilities summing to 0.5, while it is also lexically retained with a fixed probability equal to 0.5.

Corpus parser

The *corpus parser* re-parses the corpus and substitutes the words in the middle field of the triplet with the appropriate class labels, assigning the corresponding probabilities of class membership to the third field. For example, given that the word "Noon" was assigned to the classes <time> and <city> with membership probabilities 0.45 and 0.05 respectively, it is parsed as (Noon,<time>,0.45) and (Noon,<city>,0.05), as is shown in Figure 4.3. Additionally, every corpus word is lexically retained with fixed probability equal to 0.5, if it was soft clustered (else 1). For example, the word "flights" was not grouped to any induced class by the *class generator*. The *corpus parser* keeps its' lexical form as (flights,flights,1). For the word "Noon" for instance, that was soft-clustered to the classes <time> and <city> the lexical probability is 0.5.

4.4 Experimental corpus and procedure

We experimented with the ATIS [86] corpus which consists of 1,705 transcribed utterances dealing with travel information. The total number of words is 19,197 and the size of vocabulary is 575 words.

We studied the performance of the proposed *soft-clustering* algorithm in terms of precision and recall. We compare the soft-clustering to the *hard clustering* algorithm where a word is assigned only to a single induced class [41]. Also, we conducted a hard-clustering experiment where the semantic classes are induced in a single iteration, henceforth referred as *lexical*. In the *lexical* experiment, no generated labels are imported to the corpus and only lexical information is exploited for class induction. Finally, we conducted a hard-clustering experiment where *semantic* and *lexical* information is combined using equal and fixed weights of 0.5, henceforth referred as the *hard+lexical* experiment. These additional experiments are included to help us better understand the cause of improvement of the proposed algorithm vs the one in [41]; specifically if the improvement is due to mixing lexical and semantic information, or using soft- instead of hard-clustering (or both).

The three components of the proposed soft-clustering algorithm are run sequentially and iteratively over the corpus, as depicted by Figure 4.4.

The following parameters must be defined:

[3] This weighting scheme is the optimal combination, estimated on held-out data.

1. The total number of system iterations (*SI*).

2. The number of induced semantic classes per iteration (*IC*).

3. The size of Search Margin (*SM*) defined in Section 4.3.1.

The same iterative procedure and parameters are also followed and defined for the hard-clustering algorithm, described in Section 4.3.1.

Regarding the lexical experiment, the class generator of Fig. 4.4 is run once (*SI*=1), generating the total required semantic classes for evaluation.

4.5 Evaluation

4.5.1 Benchmark and evaluation measurements

For the evaluation procedure of the ATIS corpus, we used a manually crafted semantic taxonomy, consisting of 38 classes that include a total of 308 members. Every word was assigned only to one hand-crafted class. For experimental reasons, we generated manually characteristic "word chunks", e.g., New York→ New_York. Also, for the ATIS experiments, all 575 words in the vocabulary were used for similarity metric computation and evaluation.

The evaluation focused only on the terminal semantic classes. Every induced class was evaluated with respect only to the corresponding handcrafted class without examining its relationships with other classes over the taxonomy (hierarchical class generation was not evaluated). An induced class is assumed to correspond to a handcrafted class, if at least 50% of its members are included ("correct members") in the handcrafted class. The induced semantic classes were evaluated in terms of precision and recall.

Representative examples of the benchmark of semantic classes, as well as definition of precision and recall can be found in Section 3.5.1 and Section 3.5.2, respectively.

4.5.2 Experimental Results

Figure 4.5 presents the achieved precision and recall for the *soft* and *hard* clustering algorithms, and also for the *lexical* and *hard+lexical* ones. Precision and recall were computed for 80 induced semantic classes, using *SM*=10.

The proposed *soft* algorithm generated 80 classes at the 3^{rd} iteration. During the previous two iterations we calculated the *probability of class membership* over 15 induced classes (5 and 10 classes at 1^{st} and 2^{nd} iteration). The *hard* and *hard+lexical* algorithm was run for 3 iterations, generating 5 deterministic classes at 1^{st} iteration, 10 at 2^{nd} and the rest 65 classes at the 3^{rd} iteration. During the *lexical* experiment 80 classes were generated at 1^{st} iteration.

The proposed *soft* algorithm (•) outperforms the other approaches (*hard* (△), *lexical* and their combination *hard+lexical*), especially for the first 40 induced classes, in terms of precision. It is also interesting that the *lexical* algorithm outperforms the *hard* clustering algorithm (△). The poor performance of the *hard* algorithm (△) is caused by a number of erroneous induced classes which are deterministic and they are propagated from iteration to iteration. Regarding recall scores, the *soft* algorithm

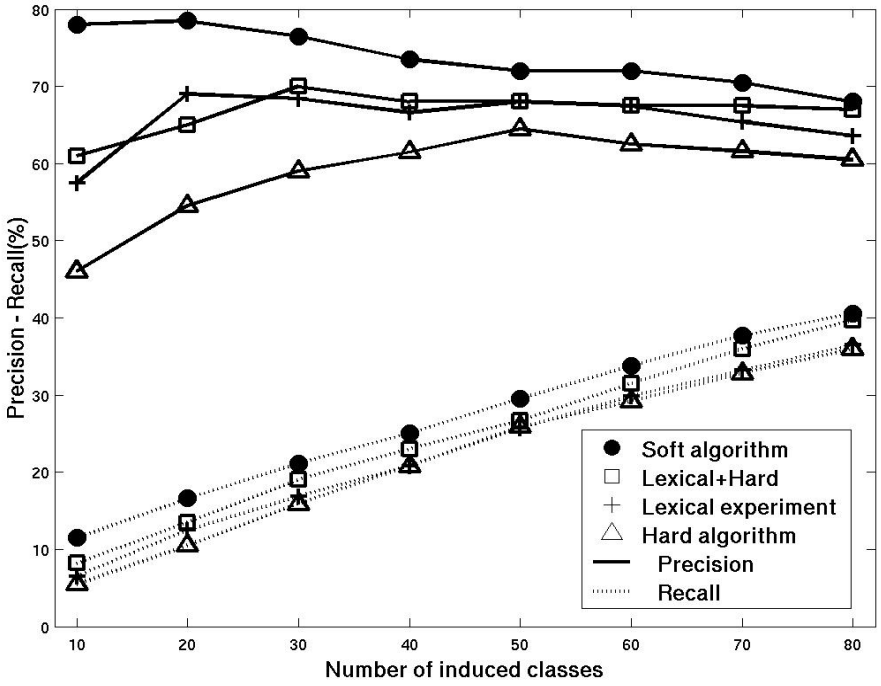

Figure 4.5: Precision and recall of *soft*, *hard*, *lexical* and *hard+lexical* algorithms on the ATIS corpus.

(•) is shown to achieve consistently higher results than the other approaches. Also the fixed combination *hard+lexical* performs slightly better than the other two "hard" algorithms indicating that the combination between lexical and semantic information does provide some performance advantage.

4.6 Conclusions

In this chapter, a soft-clustering algorithm for auto-inducing semantic classes was proposed that combines lexical and semantic information. It was shown, that the proposed algorithm outperforms state-of-the-art hard-clustering algorithms such as [41]. It was also shown that most of the improvement is due to the introduction of soft-clustering (via a probabilistic class-membership function) and less so to the combination of lexical and semantic information for class induction. The soft-clustering algorithm contributes to the better estimation and smoothing of the contextual probabilities, compared to the hard-clustering approach. Future work may investigate the effectiveness of the soft-clustering algorithm for various application domains, as well as computational complexity issues (compared with hard-clustering). Also, the optimal combination of

various metrics of lexical and semantic information in the semantic similarity distance can be incorporated in the soft-clustering method.

Chapter 5

Unsupervised Semantic Similarity Computation using Web Search Engines

5.1 Introduction

The world wide web is an extremely large multilingual source of textual information with an everyday increasing rate of growth. It provides a cheap and easy opportunity of publishing that is accessible even to the less-equipped person or organization, in contrast to other conventional means of publishing. The web can be considered as a living space which exists in parallel with the real world. Using this analogy, the web projects real life's knowledge to an unrestricted information canvas that includes billions of hypertext documents. This a space of loose coexistence, rather than a strictly organized community, of information units. The nature of published data ranges from informal information, like personal pages and blogs, to more formal types, such as newswire and government sites. The thematic character of the available data spans from generic easy readings, e.g., commercial advertisements and magazine-style articles, to deep thoughts, like philosophical essays and scientific studies. The web as a heterogeneous source of information is natural to embody various styles of expressive writing, since it hosts different authors in different areas of interest. For example, a textual advertisement consists of few words in order to be easily and quickly understood by the consumers. On the other hand, a specialized document includes expert terminology in order to preserve its integrity and it is intended for specific groups of people.

New words, neologisms and hapax legomena, are added frequently to the web. Thus it is the obvious place for mining semantic relationships for unseen words. Most of the text-based approaches to semantic similarity employ hand-crafted filtering rules and language resources to obtain and process the text corpus. As a result these methods are not of much use for applications where human and language resources are sparse. Recently there has been much research interest in developing text-based approaches for estimating probabilities and semantic similarity for unseen words; most of these methods use the web and search engines for text corpus mining.

Zhu and Rosenfeld [114] proposed a method for exploring the world wide web to acquire trigram estimates for statistical language modeling. N-grams were submitted as phrase queries to web search engines. The estimated web probability for a trigram $w_1w_2w_3$ is $\hat{P}_{web}(w_3 \mid w_1w_2) = \frac{C(w_1w_2w_3)}{C(w_1w_2)}$, where $C(.)$ is the N-gram count or the web page count, returned by the search engine. The web probability estimates were compared to a baseline corpus containing 103 millions words. The comparison showed that the two sources (web and corpus) were in almost complete agreement. Three different search engines were used for query submission: AltaVista, Lycos, and FAST. Each web-based language model was interpolated with an existing N-gram language model, using linear and geometric interpolation. The interpolated models were evaluated over a speech recognition task, using utterances from the TREC-7 Spoken Document Retrieval track data. The interpolated models showed significant improvement of word error rate, compared to the traditional baseline. Another interesting observation was that the choice of particular search engine or interpolation scheme did not have a significant impact on the performance of the interpolated model.

Semantic similarity metrics are also used in Semantic Web applications, like automatic annotation of web pages [16] and social networks construction [68, 71]. Cimiano et al. [16] used linguistic patterns to identify certain ontological relations using the web as a large corpus to overcome the data sparseness problem. This approach applied linguistic regular expressions to discover instance-concept relations in the text. For an instance a concept is suggested, with regard to an existing ontology, according to the maximal evidence derived from web statistics. As is shown in Figure 5.1, "South

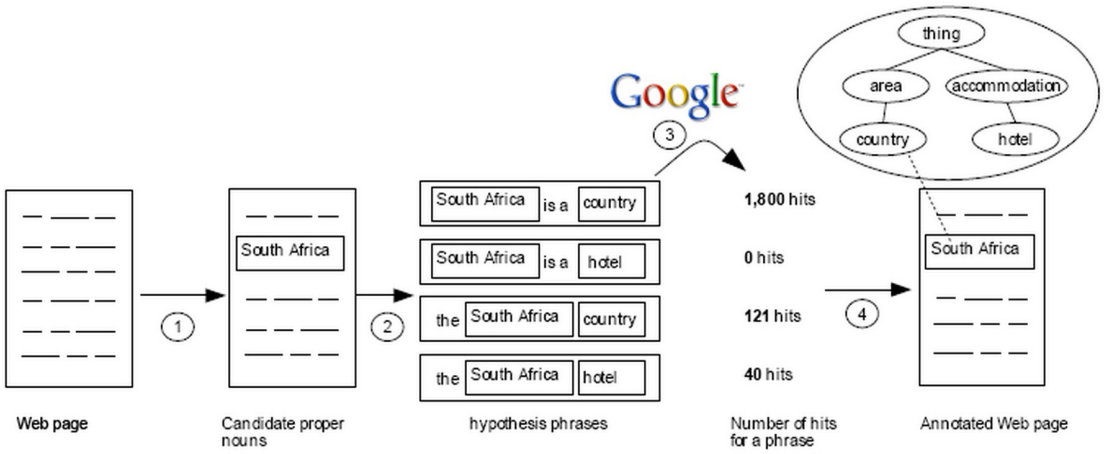

Figure 5.1: Automatic annotation of a web page with regard to an ontology (Cimiano et al., [16]).

Africa" is the instance of interest, while the candidate concepts are "country" and "hotel". The used linguistic pattern is the "is a" expression, denoting an $IS-A$ relation. The greatest hits were retrieved for the "South Africa is a country". Thus, "South Africa" was mapped to the "country" concept in the ontology. This procedure results

to an annotating of a web page using a reference ontology. Mori et al. [71], applied a similarity metric to calculate the context similarity between entity pairs in order to cluster them according to their similarity. After the creation of clusters, a number of terms were extracted from each cluster and used as labels in order to describe the relations among the entity pairs. The final goal this work was the social networks construction. This method is illustrated by Figure 5.2. Another important exploitation of

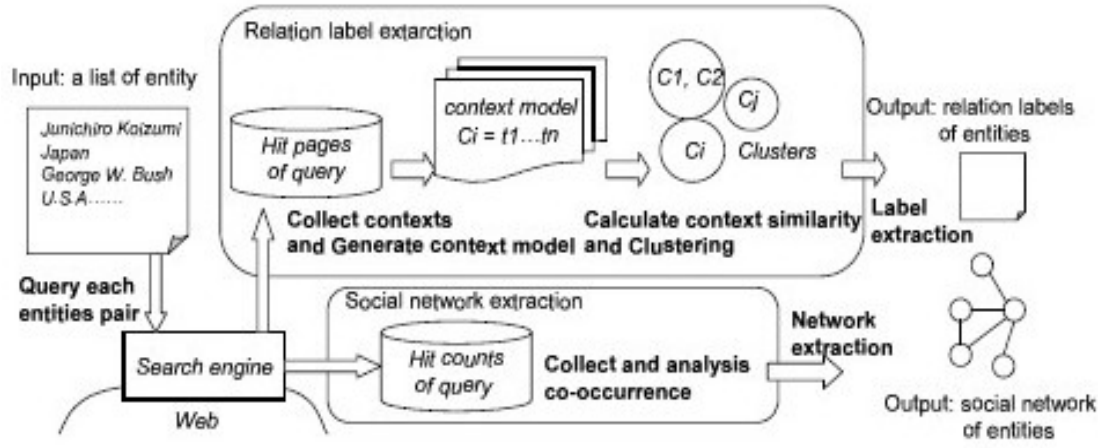

Figure 5.2: Social networks construction using web search engines and similarity metrics (Mori et al., [71]).

world wide web using search engines is for word sense disambiguation. In [67], a web search engine was used in order to disambiguate nouns, verbs, adverbs and adjectives, according to their senses found in WordNet. First, the words of interest are paired and the disambiguation of one word is tried with regard to the context of the other word. For this task several queries were sent to a search engine consisting of different senses (found in WordNet) of one word, while the other word was kept fixed. The sense were ranked according to the number of the returned hits. For computational purposes only a number of the top ranked senses were selected. Next, the order of retained senses was refined using the WordNet hierarchy. The evaluation shown 80% and 91% average accuracy for the first and first two ranked senses, respectively.

Our approach [40]: we focus on the problem of fully unsupervised semantic similarity computation, no hand-crafted rules or resources are employed. Web search engines are used for text corpus mining and context-based similarity distances are automatically computed on this corpus. The core motivation of this approach is that the web does not suffer from data sparseness, so similarity between words can be better estimated. Unsupervised semantic similarity estimation algorithms are important because they require no expert knowledge and no language resources; for many languages and applications this is the only realistic choice. In addition to their practical interest, automatically acquiring semantic similarity from text can also help us better understand the human language acquisition process, which is also (at the semantic

level) mostly unsupervised. The metrics' performance is evaluated in terms of correlation with respect to the pairs of the commonly used Charles-Miller dataset [70]. Also, the proposed metrics are compared with WordNet-based methods. Our approach is presented in Sections 5.3 - 5.6.

5.2 Related work

The metrics that measure semantic similarity between words or terms can be divided into three main categories regarding the use of knowledge resource or not:

- Resource-based metrics, consulting only human-built knowledge-bases, such as WordNet.
- Metrics that perform text mining, relying on knowledge resources.
- Unsupervised metrics that are fully text-based, exploring only the raw textual information. The metrics of this category are fully automatic and do not use any knowledge resource.

Several methods of the first category have been proposed in the literature regarding the use of WordNet and other knowledge resources for semantic similarity computation [1].

Rada et al. [87] proposed a method for exploiting WordNet [69] in order to compute semantic similarity between words. A fragment of WordNet hierarchy is shown in Figure 5.3, where words are linked with $IS-A$ relationships. One direct approach for

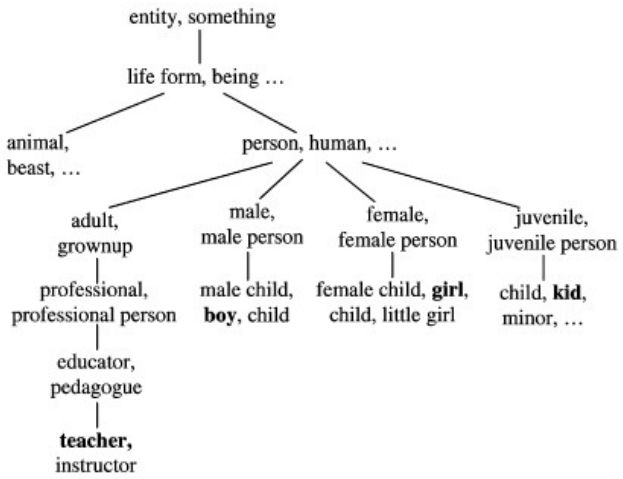

Figure 5.3: A fragment of WordNet hierarchy (Li et al., [60]).

similarity computation between two words is to identify the minimum length of path

[1] A brief description of resource-based similarity metrics can be, also, found in Section 2.3.3

that connects the two concepts containing the words of interest. Thus, this metric is an edge-based method. For example, in Figure 5.3 the shortest path between "boy" and "girl" is "boy-male-person-female-girl", resulting to minimum length of path equal to 4. The concept "person" is called subsumer for the words "boy" and "girl". In similar fashion, the minimum length path between "girl" and "teacher" is 6. So, one could claim that "boy" is more similar to "girl" than 'teacher" to "girl". In the case of a polysemous word there are multiple paths between two words. Then, only the shortest path is selected. Rada et al. reported that this approach has satisfactory performance on constraint medical semantic nets. When this method is applied over broader semantic hierarchy, such as WordNet, the results may be not so accurate [60]. For example, the minimum path from "girl" to "animal" equals to 4, which is less than the minimum path from "girl" to "teacher". Of course, it is not reasonable to claim that "girl" is more similar to "animal" than to "teacher". In the work of Petrakis et al. [83] the method proposed by Rada et al. [87] is characterized as *edge counting method*. Also in [83], the correlation performance of Rada et al. metric is reported to be 0.59, for the 28 noun pairs of the Charles-Miller [70] data set.

Jiang and Conrath [46] suggested a metric that combines edge-based method with information content as a decision weight. In particular, the information content (IC) is computed in order to determine the strength of an edge which links a parent and a child node. The information content of a concept c is defined as $IC(c) = log^{-1}P(c)$. The probability of encountering an instance of concept c is denoted by $P(c)$. Several approaches have been proposed for the estimation of $P(c)$ using a corpus [90, 91, 46]. In the work of Jiang and Conrath the Brown Corpus was used. The semantic similarity between two words was defined to be the summation of edge weights along the shortest path that links these two words. The Jiang and Conrath method is considered as an *information content method* by Petrakis et al. [83]. The obtained correlation score of Jiang and Conrath method , with respect to the 28 noun pairs of Charles-Miller data set, equals to 0.83, as is noted in [83].

Li et al. [60] extended the above methods by considering more information from the semantic hierarchy. They relied on the hypothesis that concepts at upper layers of the hierarchy are semantically more general with less similarity between them. On the other hand, they assumed that concepts which are located at lower layers embody more concrete semantics with stronger similarity. In addition, the local semantic density of words was taken into account by Li et al. Local semantic density is the information content of a concept. It was estimated from the Brown Corpus. Using these considerations, Li et al. proposed a similarity metric between two words that is a function of shortest path length, depth of subsumer in the hierarchy, and local semantic density. This method was evaluated with regard to 28 noun pairs from Charles-Miller data set, obtaining 0.82 correlation. The Li et al. metric is defined as an *edge counting method* in the work of Petrakis et al. [83].

The presented methods, edge counting [87, 60], and information content [46], are more capable for comparing words from the same ontology. Different ontologies have different structure and information content, so there are not directly comparable. The similarity computation between words of different ontologies is a difficult task and often employ *feature-based* and *hybrid* methods. Feature-based methods compute similarity

between two words according to word properties (e.g., definitions) or word relations with other words in the hierarchy. Hybrid methods combine edge counting, information content, and feature-based methods [83]. Recntly, Petrakis et al. [83] proposed a semantic similarity hybrid method, *X-Similarity*, for computing similarity between words from different ontologies. Two ontologies were used, WordNet and Mesh. It should be noted that the X-Similarity metric can be, also, applied for semantic similarity computation over a single ontology. The performance of X-Similarity method (using only WordNet) over 28 noun pairs of Charles-Miller data set was reported to give 0.74 correlation.

The work of Bollegala et al. [9] belongs to the second category of metrics, that is a method that perform text mining using a knowledge resource. For each pair of words numerous queries were submitted to Google web search engine. The computation of similarity incorporated two information sources: (a) the returned page counts, and (b) several lexico-syntactic patterns extracted from responded snippets. A snippet is a chunk of text of limited length, provided by the search engine that gives information about the local context of the query. The lexico-syntactic pattern extraction was based on 5.000 pairs of synonymous nouns, taken from WordNet. The proposed by Bollegala et al. metric, *SemSim*, achieved 0.83 correlation, evaluated in the 28 noun pairs of Charles-Miller data set.

Regarding the third category of metrics, there is little work for the computation of semantic similarity between words only by querying web search engines (with use of any knowledge source). In [9], three co-occurrence measures were tested, which use only the returned page counts for the sumitted queries: Jaccard, Overlap, and Dice coefficient, and point wise mutual information. With respect to the referred noun pairs of Charles-Miller data set , Jaccard and Dice coefficients resulted to 0.259 and 0.267 correlation, respectively. Overlap coefficient and point wise mutual information achieved 0.382 and 0.548 correlation, respectively. These co-occurrence metrics are simplistic, since it is likely two words to appear in a page without being semantically closely related. For example, “orange” is a fruit, but also is the name of a company. Sahami et al. [97], measured the similarity between short text snippets by using web search engine results to get greater context for the examined snippets. The implementation and evaluation of Sahami method by Bollegala et al. [9] was reported to give 0.579 correlation for the above benchmark of noun pairs.

5.3 Proposed method

We propose two novel web-based metrics for semantic similarity computation between words. Both metrics use a web search engine in order to exploit the retrieved information for the words of interest. The first metric considers only the page counts returned by a search engine, based on the work of [9]. The second is fully text-based; downloads a number of the top ranked documents and applies “wide-context” and “narrow-context” metrics (see Chapter 3). The proposed metrics work automatically, without consulting any external knowledge resource.

5.3.1 Page-count-based similarity metrics

The basic idea under this approach is that the word co-occurrence is likely to indicate some kind of semantic relationship between words. A quick approximation of word co-occurrence can be estimated exploring the web. However, the number of documents in which a certain word pair co-occurs, does not express a direct semantic similarity. In addition, it is reasonable to, also, take into account the number of documents that include the each pair component individually for normalization purposes. [2] In other words, for a word pair, we need to know the information that the two words share, normalized by the degree of their independence. We define the following [22]:

$\{D\}$: a set containing the whole document collection that are indexed and accessible by a web search engine
$|D|$: the number of documents in collection $\{D\}$
w_i: a word or term
$\{D|w_i\}$: a subset of $\{D\}$, documents indexed by w_i
$\{D|w_i, w_j\}$: a subset of $\{D\}$, documents indexed by w_i and w_j
$f(D|w_i)$: the fraction of documents in $\{D\}$ indexed with w_i
$f(D|w_i, w_j)$: the fraction of documents in $\{D\}$ indexed with w_i and w_j

We use three co-occurrence measures in this work, Jaccard coefficient, Dice coefficient and Mutual Information, to compute semantic similarity between word pairs as in [9]. The Jaccard coefficient is a measurement calculating the similarity (or diversity) between sets. We use a variation of the Jaccard coefficient defined as:

$$Jaccard(w_i, w_j) = \frac{f(D|w_i, w_j)}{f(D|w_i) + f(D|w_j) - f(D|w_i, w_j)} \tag{5.3.1}$$

In probabilistic terms, Equation 5.3.1 finds the maximum likelihood estimate of the ratio of the probability of finding a document where words w_i and w_j co-occur over the probability of finding a document where either w_i or w_j occurs[3]. If w_i and w_j are the same word then the Jaccard coefficient is equal to 1 (absolute semantic similarity). If two words never co-occur in a document then the Jaccard coefficient is 0.

The Dice coefficient is related to the Jaccard coefficient and is computed as:

$$Dice(w_i, w_j) = \frac{2f(D|w_i, w_j)}{f(D|w_i) + f(D|w_j)} \tag{5.3.2}$$

Again, the Dice coefficient equals to 1 if w_i and w_j are identical, and 0 if two words never co-occur.

[2] It is interesting to note that web-based co-occurrence metrics often outperform more elaborate corpus-based metrics. This shows that overcoming the data sparseness problem is sometimes more important than building an accurate estimator. For example an improved n-gram language probability estimation using web n-gram occurrence can be found in the literature [114].

[3] Normalization by the total number of documents $|D|$ is the same for the nominator and denominator, and can be ignored.

If we consider the occurrence of words w_i and w_j as random variables X and Y, respectively, then the pointwise mutual information (MI) among X and Y measures the mutual dependence between the appearance of words w_i and w_j [15]. The maximum likelihood estimate of MI is

$$MI(X,Y) = \log \frac{\frac{f(D|w_i,w_j)}{|D|}}{\frac{f(D|w_i)}{|D|}\frac{f(D|w_j)}{|D|}} \tag{5.3.3}$$

Mutual information measures the information that variables X and Y share. It quantifies how the knowledge of one variable reduces the uncertainty about the other. For instance, if X and Y are independent, then knowing X does not give any information about Y and the mutual information is 0. For $X = Y$, the knowledge of X gives the value of Y without uncertainty and the mutual information is 1. Note that the fractions of documents are normalized by the number of documents indexed by the search engine, $|D|$, giving a maximum likelihood estimate of the probability of finding a document in the web that contains this word.

5.3.2 Fully text-based similarity metrics

Two semantic similarity metrics that are variations of the cosine similarity metric are used in order to measure the semantic distance between words and to automatically generate semantic classes. The first metric, CS^W_{WS}, computes "wide-context" similarity between words using a "bag-of-words" model, while the second metric, CS^N, computes "narrow-context" similarity using a bigram language model [4].

These metrics rely on the idea that similarity of context implies similarity of meaning. We hypothesize that words, which appear in similar lexical environment (left and right contexts), have a close semantic relation [96, 103, 79].

In "bag-of-words" [101, 41] models, for each word w in the vocabulary a context window size WS is selected. The right and left contexts of length WS in the corpus are considered for word w, e.g., $[v_{WS,L} \ ... \ v_{2,L} \ v_{1,L}] \ w \ [v_{1,R} \ v_{2,R} \ ... \ v_{WS,R}]$, where $v_{i,L}$ and $v_{i,R}$ represent the i^{th} word to the left and to the right of w respectively. The feature vector for every word w is defined as $T_{w,WS} = (t_{w,1}, t_{w,2}, ..., t_{w,N})$ where $t_{w,i}$ is a non-negative integer and WS is the context window size. Note that the feature vector size is equal to the vocabulary size N, i.e., we have a feature for each word in the vocabulary V. The i^{th} feature value $t_{w,i}$ reflects the occurrences of vocabulary word v_i within the left or right context window WS. This feature value is set according to a Binary (Bin.) or a Term Frequency (Freq.) Scheme. The binary Scheme assigns 1 if the word v_i appears within the left and right window context of size WS for the word w, while the term frequency scheme assigns the number of occurrences of v_i in left and right WS. Both schemes assign a 0 value if v_i does not exist within WS. The "bag-of-words" metric, CS^W_{WS}, using binary or term frequency scheme, measures the similarity of two words, w_1 and w_2, as the cosine distance of their corresponding feature

[4] These metrics are, also, defined in Chapter 3. We repeat their definition here in order to make Chapter 5 self-contained.

vectors, $T_{w_1,WS}$ and $T_{w_2,WS}$ [77, 41]:

$$CS^{W}_{WS}(w_1, w_2) = \frac{\sum_{i=1}^{N} t_{w_1,i} t_{w_2,i}}{\sqrt{\sum_{i=1}^{N}(t_{w_1,i})^2}\sqrt{\sum_{i=1}^{N}(t_{w_2,i})^2}} \tag{5.3.4}$$

given a context window of length WS.

In an N-gram language model a word w is considered with its neighboring words $v_{1,L}$ and $v_{1,R}$ in the left and right contexts within a sequence. In order to calculate the similarity of two words, w_1 and w_2, we compute the cosine similarity between two feature vectors; each feature vector of a word w measures the conditional probability of all possible contexts v_i given that word $p(v_i|w)$, i.e., each vector contains bigram language model probabilities for (context, word) pairs. Semantic similarity is defined as

$$CS^{N}(w_1, w_2) = CS^{N}_{L}(w_1, w_2) + CS^{N}_{R}(w_1, w_2) \tag{5.3.5}$$

where the two terms of Equation 5.3.5 are [79, 41]:

$$CS^{N}_{L}(w_1, w_2) = \frac{\sum_{i=1}^{N} p(v_{i,L}|w_1)p(v_{i,L}|w_2)}{\sqrt{\sum_{i=1}^{N} p(v_{i,L}|w_1)^2}\ \sqrt{\sum_{i=1}^{N} p(v_{i,L}|w_2)^2}} \tag{5.3.6}$$

$$CS^{N}_{R}(w_1, w_2) = \frac{\sum_{i=1}^{N} p(v_{i,R}|w_1)p(v_{i,R}|w_2)}{\sqrt{\sum_{i=1}^{N} p(v_{i,R}|w_1)^2}\ \sqrt{\sum_{i=1}^{N} p(v_{i,R}|w_2)^2}} \tag{5.3.7}$$

where $V = (v_1, v_2, ..., v_N)$ is the vocabulary set, and $p(v_i|w)$ is the conditional probability of word v_i preceding w in the corpus given word w, i.e., the (v_i, w) bigram model probability.

The bigram language model was built using the CMU Statistical Language Modeling Toolkit [17]. The "wide-context" and "narrow-context" metrics assign 0 similarity score to completely dissimilar words, and 1 in the case of identical words.

5.4 Experimental dataset and procedure

We experimented with (i) page-count-based, and (ii) contextual similarity metrics, described in Section 5.3.1 and Section 5.3.2, respectively.

5.4.1 Experimental dataset

As a benchmark we used the commonly used Miller-Charles dataset [70]. This dataset consists of noun pairs that were rated according to their semantic similarity by 38 human subjects. The assigned similarity scores range from 0 (not similar) to 4 (perfect synonymy). The selection of this dataset was motivated by its wide use. This fact enabled us to compare our work with other approaches of different nature that were, also, evaluated on this dataset.

Type 1	"w_1 AND w_2" (e.g., "boy AND lad")
Type 2	"w_1", "w_2" (e.g., "boy", "lad")

Table 5.1: Query types

5.4.2 Downloading procedure

For the contextual similarity metrics, for each pair, "w_1 w_2", of Miller-Charles dataset, we downloaded the 100 top ranked documents for the following query types: The *URLs* for the top ranked documents were retrieved using the Yahoo search engine via the Yahoo Search API which is freely available [111]. In Table 5.1 the first query type is a single query, which retrieves documents containing both words. The second query type consists of two distinct queries; the first one requires documents that contain, at least, w_1, while the second query is satisfied by documents in which w_2, at least, appears. For the first query type up to 100 documents were downloaded for each word pair (100 documents per word were downloaded for the second query type).

The motivation behind using the two different query types is that the type of documents retrieved by each type of query is semantically different. The Type 1 "and" query is expected to retrieve documents that are semantically homogeneous, while the Type 2 query will produce documents that are semantically more diverse. As a result using corpora generated from "and" type queries, i.e., for a particular word pair, the corresponding feature vectors are built on the basis of a relatively coherent lexical environment, since the component words co-occur in each retrieved document. In contrast, the use of Type 2 query type results to more diverse feature vectors. In this case the feature vector for each pair word is constructed using different documents of, probably, different expressive style and semantic content. In previous work [41], we have seen better results for semantic similarity computation in semantically homogeneous corpora. Also the optimum context length (window size) varies significantly depending on the semantic homogeneity of the corpus (smaller for semantically homogeneous corpora).

5.5 Evaluation

In this section, we present a comparative evaluation of the referred similarity metrics, in terms of correlation, with respect to the human rating of Miller-Charles pairs. First, the page-count-based similarity metrics defined in Section 5.3.1 are evaluated. Next, we evaluate the proposed fully text-based similarity metrics defined in Section 5.3.2. The proposed metrics are also compared with metrics that use WordNet as a knowledge source.

5.5.1 Evaluation of page-count-based metrics

The correlation scores between the page-count-based semantic similarity metrics of Equations 5.3.1 - 5.3.3 and human ratings of Miller-Charles pairs are presented in Table 5.2. The similarity metrics based on the Jaccard and Dice coefficients achieve

Metric	Jaccard	Dice	Mutual Information
Correlation	0.32	0.33	0.43

Table 5.2: Correlation of page-count metrics.

similar correlation. This is reasonable given the similarity of the two metrics. The Mutual Information metric achieves significantly better performance. This is due to the use of the logarithm non-linearity in the score computation and the different normalization used in this metric. Overall, all the page-count-based metrics obtain poor correlation results. This is expected considering that no textual information is included in this metrics, only the page counts information.

5.5.2 Evaluation of fully text-based metrics

Next we evaluate the "wide-context" metric, CS^W_{WS}, which computes similarity between words using a "bag-of-words" model and the "narrow-context" metric, CS^N, which is based on a bigram language model. For the "wide-context" metric we studied the impact of the feature vector weighting scheme (binary or term frequency), as well as how the window size (WS) affects the semantic similarity computation.

Figure 5.5.2 illustrates the correlation between the "wide-context" similarity metric and the human ratings of 28 Miller-Charles pairs. The correlation is shown for both binary and term frequency weighting schemes and various window sizes WS. The similarity metric was computed on the top 100 documents retrieved by the search engine for web queries of Type 1 ("and") for each of the 28 word pairs. The correlation performance of the "narrow-context" metric is also shown for the same query type as a dotted line.

We observe that the correlation decreases as the value of window size, WS, increases; best results are obtained for window size two or three. This suggests that the immediate context (preceding and following two words) is sufficient to compute semantic similarity using "and" query type 1. Furthermore, the "wide-context" metric using the binary scheme outperforms the term frequency weighting scheme. This is probably due to the fact that the term frequency scheme gives more weight to the commonly occurring context words that are often "non-content" words (aka stop-words). These non-content words dominate the similarity calculation between the two term frequency vectors, while the binary weighting scheme is more robust to the presence of such words. The bigram "narrow-context" metric achieves almost equal performance with the "wide-context" metric of the term frequency scheme and $WS = 2$, since both metrics use contextual frequency weighting schemes. Overall, the highest correlation is obtained by the "wide-context" metric using the binary scheme and $WS = 2$, and it is equal to 0.71.

In Table 5.3 the correlation for the "wide-context" metric using the binary scheme and $WS = 2$ is shown as a function of the number of Type 1 retrieved documents. As expected performance degrades as the number of downloaded documents per word pair decreases. Note that good correlation (0.69) is achieved even when the top 50

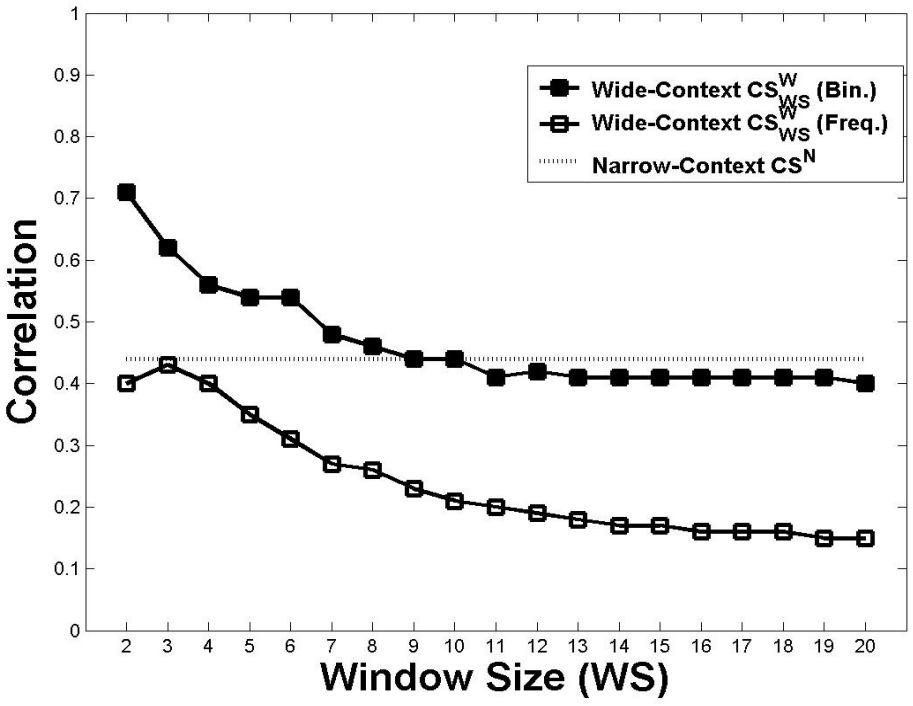

Figure 5.4: Correlation of fully text-based metrics for queries of Type 1

downloaded documents are used in the similarity computation.

No. of Docs	5	10	25	50	75	100
Correlation	0.41	0.50	0.66	0.69	0.69	0.71

Table 5.3: Correlation vs. number of docs.

The use of queries of Type 2 resulted to unexpectedly poor correlation scores. This is probably due to the very different types of document retrieved with Type 2 queries, i.e., lack of semantic homogeneity in the downloaded corpus.

5.5.3 Unsupervised vs supervised metrics

In this section we compare the performance of the proposed unsupervised metrics with other supervised and unsupervised metrics. All of them were evaluated with respect to the 28 pairs of Charles-Miller dataset in terms of correlation. A metric is considered to be unsupervised if does not consult any knowledge resource. The main characteristics of each metric are summarized in Table 5.4. The detailed semantic similarity scores and

Metric	Use of (√: yes, X: no)						Need of external knowledge	Correlation
	WWW Search engine	Page counts	Snippets	Lexico-Syntactic patterns	WordNet	Download documents		
Jaccard	√	√	X	X	X	X	X	0.32
Dice	√	√	X	X	X	X	X	0.33
MI	√	√	X	X	X	X	X	0.43
Sahami	√	X	√	X	X	X	X	0.58
SemSim	√	√	√	√	√	X	√	0.83
Li	X	X	X	X	√	X	√	0.82
Jiang	X	X	X	X	√	X	√	0.83
X-Similarity	X	X	X	X	√	X	√	0.75
Proposed $CS^{W}_{WS=2}$ (Binary)	√	X	X	X	X	√	X	0.71

Table 5.4: Characteristics of several similarity metrics

the overall correlation with human scores are presented in Table 5.5 for each metric.

The Li [60], Jiang [46], and X-Similarity [83] metrics exploit the semantic hierarchical structure of WordNet, to compute semantic similarity as described in Section 5.2. All three metrics achieve higher correlation, but at the cost of using additional information that was not derived from text in an automatic unsupervised manner.

The performance of the web-based metrics is summarized as follows. The resource-based SemSim metric, proposed in [9], achieves a correlation score that is similar to the ontology-based methods above. The fully unsupervised Sahami [97] metric is shown to have a moderate correlation (results are reproduced from the implementation and evaluation in [9]). The lowest correlation scores are achieved by the metrics that consider only the returned page counts for a query: Jaccard, Dice and mutual information (MI) metrics. The proposed unsupervised, "wide-context" metric $CS^{W}_{WS=2}$ with the binary weighting scheme achieves the highest correlation (0.71) among the unsupervised metrics.

5.6 Conclusions

We presented two types of unsupervised, web-based metrics for semantic similarity computation between words. Both types use a web search engine in order to exploit the retrieved information for the words of interest. The first type considers only the page counts returned by the search engine. The second type is fully text-based and needs a number of the top ranked documents to be downloaded. We applied two "wide-context" metrics and a "narrow-context" metric to the downloaded documents. The proposed metrics do not consult any external knowledge resource. The metric performance was evaluated on the commonly used Charles-Miller word pair dataset.

The page-count-based metrics produced low to mid correlation with human scores. Good correlation scores were obtained with the fully text-based metric using a binary weighting scheme, especially for small context windows. The semantic distance

Word Pair	Miller-Charles	Jaccard	Dice	MI	Sahami	SemSim	Li	Jiang	X-Similarity	Binary $CS^{W}_{WS=2}$
chord-smile	0.13	0.14	0.14	0.78	0.09	0	0.13	0.35	0.2	0.4
rooster-voyage	0.08	0.03	0.03	0.76	0.2	0.02	0	0.08	0	0
noon-string	0.08	0.2	0.21	0.79	0.08	0.02	0	0.18	0	0.16
glass-magician	0.11	0	0	0.82	0.14	0.18	0.14	0.68	0.14	0.18
monk-slave	0.55	0.23	0.24	0	0.1	0.38	0.45	0.39	0.32	0.19
coast-forest	0.42	0.61	0.63	0.81	0.25	0.41	0.25	0.29	0.18	0.76
monk-oracle	1.1	0.07	0.07	0.77	0.05	0.33	0.25	0.34	0.28	0.47
lad-wizard	0.42	0.09	0.1	0.82	0.15	0.22	0.45	0.32	0.36	0.37
forest-graveyard	0.84	0.13	0.13	0.85	0	0.55	0.09	0.19	0.07	0.11
food-rooster	0.89	0.03	0.03	0.76	0.8	0.6	0.04	0.4	0.05	0.35
coast-hill	0.87	0.99	0.99	0.82	0.29	0.87	0.44	0.71	0.34	0.18
car-journey	1.16	0.33	0.35	0.75	0.19	0.29	0	0.33	0.03	0.52
crane-implement	1.68	0.08	0.09	0.78	0.15	0.13	0.44	0.59	0.29	0.1
brother-lad	1.66	0.22	0.23	0.88	0.24	0.34	0.45	0.28	0.45	0.58
bird-crane	2.97	0.28	0.29	0.86	0.22	0.88	0.55	0.73	0.37	0.59
bird-cock	3.05	0.16	0.17	0.79	0.06	0.6	0.82	0.73	0.45	0.44
food-fruit	3.08	0.85	0.86	0.83	0.18	0.99	0.13	0.63	0.09	0.79
brother-monk	2.82	0.33	0.35	0.88	0.27	0.38	0.82	0.91	0.44	0.63
asylum-madhouse	3.61	0.07	0.08	0.95	0.21	0.77	0.82	0.97	0.44	0.51
furnace-stove	3.11	0.67	0.68	1	0.31	0.89	0.23	0.39	0.47	1
magician-wizard	3.5	0	0	0.92	0.23	1	1	1	0.1	0.59
journey-voyage	3.84	0.37	0.39	0.84	0.52	0.99	0.82	0.88	0.52	0.75
coast-shore	3.7	0.8	0.82	0.87	0.38	0.95	0.81	0.99	0.66	0.5
implement-tool	2.95	1	1	0.87	0.42	0.68	0.81	0.97	0.48	0.8
boy-lad	3.76	0.22	0.23	0.87	0.47	0.97	0.82	0.88	0.44	0.67
automobile-car	3.92	0.61	0.63	0.82	1	0.98	1	1	1	0.76
midday-noon	3.42	0.14	0.15	0.88	0.29	0.82	1	1	1	0.74
gem-jewel	3.84	0.44	0.45	0.91	0.21	0.69	1	1	1	0.53
Correlation	1	0.32	0.33	0.43	0.58	0.83	0.82	0.83	0.75	0.71

Table 5.5: Correlation for several types of similarity metrics

was computed for each word pair on a corpus of 50-100 retrieved documents where the words co-occurred. The best performance achieved for this metric was 0.71, which is the highest correlation score among the fully unsupervised metrics in the literature. Furthermore, we compared the proposed metrics with various state-of-art resource-based metrics that use ontologies (e.g., WordNet) to compute semantic similarity. Overall, the performance of the proposed method is satisfactory given that the method is language-independence, fully automatic, requires little computation-power and small amounts of web text.

Future work deals with investigating a variety of criteria for improving the semantic similarity method including better document selection (as opposed to web search engine ranking) and better context word feature extraction (including unsupervised algorithms for stemming and part of speech tagging). Further research is needed to better understand the limited performance for Type 2 queries and for the frequency weighting scheme.

Chapter 6

Conclusions and Future Work

In this chapter we summarize the research contribution of this thesis and we provide useful directions for future work. This thesis proposed several fully unsupervised similarity metrics for computing semantic similarity between words and concepts. We investigated metrics with a range of contextual scope: from narrow to wide context. We experimented with the adaptive linear combination of metrics (*see Chapter 3*). Also, we used the computed similarity scores in order to cluster words into semantic classes. For the clustering procedure we used both hard and soft approaches (*see Chapter 4*). Furthermore, we considered the world wide web as the largest corpus and we used search engines to acquire textual data for computing semantic similarity (*see Chapter 5*). Next, we summarize the conclusions for the above research areas, giving also some hints for future work.

6.1 Unsupervised combination of metrics for semantic class induction

We experimented with two types of corpora: a semantically heterogeneous news domain, and an application-specific travel reservation corpus. An individual "narrow-context" metric performed better than individual "wide-context" metrics with regard to the homogeneous, application-specific corpus. This is because the utterances have short length, so the immediate context is capable for meaning identification. On the other hand, for the semantically heterogeneous corpus, individual "wide-context" metrics obtained better results, compared to the individual "narrow-context" metric. This happened because the semantic diversity of the latter domain requires a broader contextual scope in order to estimate semantic similarity. The proposed adaptive weighting scheme of metric was observed to outperform the individual metrics and the fixed combination of them, for both types of corpora. The proposed algorithm achieves to monitor the efficiency of each individual metric and assigns automatically greater weight to the "best-performing" metric. Thus, there is no need to select metrics based on corpus characteristics. Instead a corpus independent combined metric can be used and automatically poor performing metrics will be weighted less in the combination.

Future work will investigate how to include estimation error variance is the weight estimation criterion. Combination of other types of semantic similarity measures it is,

also, interesting to be investigated.

6.2 Soft-clustering algorithm for automatic induction of semantic classes

We proposed a soft-clustering algorithm for auto-inducing semantic classes that combines lexical and semantic information. It was shown, that the proposed algorithm outperforms state-of-the-art hard-clustering algorithms. The poor performance of the hard-clustering algorithms is because a number of erroneous deterministic classes are generated and propagated from iteration to iteration. So, the useful lexical information substituted by wrong labels. In contrast, the proposed a soft-clustering avoids this phenomenon by combining lexical and semantic information. It was also shown that most of the improvement is due to the introduction of soft-clustering (via a probabilistic class-membership function) and less so to the combination of lexical and semantic information for class induction. The soft-clustering algorithm contributes to the better estimation and smoothing of the contextual probabilities, compared to the hard-clustering approach.

From a technical point of view, the proposed soft-clustering algorithm has a significant computational complexity, so more work is needed towards this direction. It is interesting to test the proposed algorithm on various application domains. Also, the optimal combination of several metrics of lexical and semantic information in the semantic similarity metric can be incorporated in the soft-clustering method.

6.3 Unsupervised semantic similarity computation using web search engines

We presented two types of unsupervised (no external knowledge resource, such as WordNet, are used), web-based metrics for semantic similarity computation between words. Both types use a web search engine in order to exploit the retrieved information for the words of interest. The first type considers only the page counts returned by the search engine. The second type is fully text-based and needs a number of the top ranked documents to be downloaded. We applied two "wide-context" metrics and a "narrow-context" metric to the downloaded documents. The metric performance was evaluated on the commonly used Charles-Miller word pair data set. The page-count-based metrics produced low to mid correlation with human scores. This shows that the co-occurrence of two words in a document can not form a strong implication of their semantic relation. Good correlation scores were obtained with the fully text-based metric using a binary weighting scheme, especially for small context windows. The semantic distance was computed for each word pair on a corpus of retrieved documents where the words co-occurred. The best performance achieved for this metric was 0.71, which is the highest correlation score among the fully unsupervised metrics in the literature. So, the similarity estimation via contextual metrics is more reliable than the simplistic page-count-based approach. The performance of the proposed method is satisfactory given that the method is language independent, fully automatic, requires

little computation-power and small amounts of web text.

Future work deals with investigating a variety of criteria for improving the semantic similarity method including better document selection using the provided search options. We can also process the downloaded documents using stemming and part of speech tagging. Furthermore, we can apply the proposed web-based similarity computation framework for tasks such as word sense disambiguation. Longer-term goals of our work in this area will attempt to: (a) capture the time evolution of a relation between two entities by using the timestamps of the retrieved documents, (b) discover several subjective considerations for a specific event, (c) interpret the nature of a similarity score (negative or positive relation). For example, it is challenging to identify automatically whether a political event implies cooperation or conflict between the participated parties.

6.4 Looking through the window: an epilogue of self-criticism

A look through the lab window reveals the existence of a big picture: a tremendous semantic interchange is lying on the backbone of every social system. It forms an active multilingual process of information dissemination, ranging from personal communications to news broadcast intended for population masses. In this thesis we proposed several unsupervised metrics for the automatic computation of similarity and induction of semantic classes. One may wonder about the practical significance of the presented achievements. We believe that this question is the application-oriented part of a broader and deeper crucial issue under discussion. How the semantics are encoded in the use of language (spoken and written) with regard to different aspects of communication?

We saw that the word-space model provides a sufficient spatial representation of meaning. According to this model the contextual similarity metrics are conditioned to the distributional properties of the working domain. Thus, these methods are naturally biased by the applied model and limited by the training data. The used word-space model in combination with similarity metrics and class induction process attempted a migration from lexical to semantic space. This try was triggered on the basis of particular examples of language use, encountered in the training data. Any set of training data remains finite, while the creative power of language is not always predictable and the embodied semantics are also influenced by extralinguistic factors. So, we can not claim that the extracted semantics of this work are axiomatic and correspond to a universal ground truth. However, this research highlighted an applicable model for spatial representation of semantics and similarity measures, resulting to reasonable results with regard to certain thematic domains.

The majority of dialogue systems are designed for well-specified applications, having certain tasks to accomplish. In other words, the commercial dialogue systems do not deal with general purpose natural language understanding, in contrast to a Turing-style intelligent system. In such systems the automatic induction of semantic classes gives a quick view to the system designer about the expected concepts, helping by this way to the rapid system development. Moreover, these classes contribute to the semantic analysis of the recognized spoken utterance. A recognized sentence can be in

partial semantically parsed or assigned to a structure under a rule set of a context-free grammar. Also, the knowledge of semantics is a step towards the pragmatic analysis. Additionally, semantic classes can be employed in class-based language modeling. It was reported that the interpolation of word- and class-based language models obtains reduced perplexity compared to the word-based models, for example, 236 *vs* 244, respectively. Obviously, a dialogue system preserves on average its operational capability even with a modest subsystem like a language modeling component. However, a more sophisticated subsystem is still useful, especially for the cases that are beyond of a typical use. In practice, such cases may be rare, but their cost it is possible to be inversely proportional to their frequency. For example, imagine a dialoque system not dealing with the industry of enjoyment (e.g., purchase movie tickets), but managing calls for medical incidents which can be ordinary or urgent.

Another domain for applying similarity metrics and semantic classes is the Semantic Web and Information Retrieval. In general, this domain is semantically more diverse compared to the field of dialogue systems. This happens because a huge number of information sources are manipulated along with a large number of users with different needs. The use of semantics suggests the semantic representation and retrieval, rather than the simplistic lexical matching. For example, a query submitted to a search engine can be extended by including to it semantically similar terms to the originally keywords. Furthermore, the returned documents can be clustered on a basis of semantic distance. Several studies showed that this the semantic approach contributes to greater recall scores. Clearly, such systems can have reasonable performance without incorporating semantic information. In such case, their response will be focused on the lexical types by which they were questioned. Of course the above statement is an exaggeration, but emphasize how myopic tactic is to use only lexical features, while the language is enriched by a plethora of semantic properties. Very often it is useful to provide a semantically broader collection of information (greater recall). The key idea for this task is the use of semantics. At least, this approach may be optional for the users that have no time constraints to browse the results of a search.

The practical gain by the knowledge of semantics through an unsupervised approach is significant by the sense that a bit of progress is achieved towards the automatic natural language understanding. An issue of first priority is to understand how the semantics are encoded and used in the natural language. The problem of the practical use of semantics with regard to a specific application has lower priority.

At this point we can revisit the following statement of Jelinek: “put language back into language modeling”. The language modeling techniques and their applications are well-studied. The big goal is to study further how the deep structure of language is organized and used, and by which ways we can employ this knowledge into the methods of natural language processing.

Bibliography

[1] Bahl, R.L., Brown, F.P., de Souza V.P., Mercer, L.R., "*A Tree-based Statistical Language Model for Natural Language Speech Recognition.*" IEEE Transactions on Acoustics, Speech and Signal Processing, Vol.37, 1989.

[2] Bahl, R.L., Jelinek, F., Mercer, L.R., "*A Maximum Likelihood Approach to Continuous Speech Recognition.*" IEEE Transactions on Pattern Analysis and Machine Intelligence, 1983.

[3] Bai, S., Li, H., Lin, Z., Yuan, B., "*Building Class-based Language Models with Contextual Statistics.*" In: Proc. ICASSP, 1998.

[4] Beckwith, R., Miller, A.G., "*Implementing a Lexical Network.*" Technical Report 43, Princeton University, 1992.

[5] Bellegarda, R.J., "*A Multi-span Language Modeling Framework for Large Vocabulary Speech Recognition.*" IEEE Transactions on Speech and Audio Processing, Vol.6, 1998.

[6] Bellegrarda, J.R., Butzberger, J.W., Chow, Y.L., Coccaro, N.B., Naik, D., "*A Novel Clustering Algorithm based on Latent Semantic Analyis.*" In: Proc. ICASSP, 1996.

[7] Berland, M., Charniak, E., "*Finding Parts in Very Large Corpora.*" In: Proc. 37th Annual Meeting of the Association for Computational Linguistics, 1999.

[8] Blake, A.J., Harris, M., "*The Gene Ontology Project: Structured Vocabularies for Molecular Biology and their Application to Genome and Expression Analysis.*" Current Protocols in Bioinformatics, Willey and Sons, 2003.

[9] Bollegala, D., Matsuo, Y., Ishizuka, M., "*Measuring Semantic Similarity between Words using Web Search Engines.*" In: Proc. Int. WWW2007 Conf., 2007.

[10] Brown, F.P., Della Pietra, A.S., Della Pietra, J.V., Lai, C.J., Mercer, L.R., "*An Estimate of an Upper Bound for the Entropy of English.*" Computational Linguistics. Vol.18, 1992.

[11] Brown, F.P., Della Pietra, J.V., deSouza, V.P., Lai, C.J., Mercer, L.R., "*Class-Based n-gram Models of Natural Language.*" Computational Linguistics, Vol.18, 1992.

[12] Caraballo, S., "*Automatic Acquisition of a Hypernym-labeled Noun Hierarchy from Text.*" In: Proc. 37th Annual Meeting of the Association for Computational Linguistics, 1999.

[13] Chen, F.S, "*Probabilistic Models for Natural Language.*" Ph.D. thesis, Harvard University, 1996.

[14] Chen, F.S., Goodman, J., "*An Empirical Study of Smoothing Techniques for Language Modeling.*" In: Proc. Thirty-Fourth Annual Meeting of the Association for Computational Linguistics, 1996.

[15] Church, W.K., Hanks, P., "*Word Association Norms, Mutual Information, and Lexicography.*" Computational Linguistics, Vol. 16, 1990.

[16] Cimano, P., Handschuh, S., Staab, S., "*Towards the Self-annotating Web.*" In: Proc. Int. WWW2004 Conf., 2004.

[17] Clarkson, P.R., Rosenfeld, R., "*Statistical Language Modeling Using the CMU-Cambridge Toolkit.*" In: Proc. 5th European Conference on Speech Communication and Technology, 1997.

[18] Cover, M.T., Thomas, J.A., "*Elements of Information Theory.*" John Wiley, 1991.

[19] Deerwester, S., Dumais, T.S., Furnas, W.G., Landauer, K.T., Harshman, R., "*Indexing by Latent Semantic Analysis.*" Journal of American Society for Information Science, Vol.41, 1990.

[20] Della Pietra, S., Della Pietra, V., Mercer, L.R., Roukos, S., "*Adaptive Language Modeling using Minimum Discriminant Estimation.*" In: Proc. DARPA Workshop of Speech and Natural Language, 1992.

[21] Drymonas, E., Zervanou, K., Petrakis, G.M.E. , "*Exploiting MultiWord Similarity for Information Retrieval: the TSRM Approach.*" ACM 16th Conference on Information and Knowledge Management (to appear), 2007.

[22] Feldman, R., Dagan, I., "*Mining Text using Keywords Distributions.*" Journal of Intelligent Information Systems, 1998.

[23] Flank, S., "*A Layered Approach to NLP-based Information Retrieval.*" In: Proc. 17th Int. Conf. on Computational linguistics, 1998.

[24] Fosler-Lussier, E., Kuo, H.-K. J., "*Using Semantic Class Information for Rapid Development of Language Models Within ASR Dialogue Systems.*" In: Proc. ICASSP, 2001.

[25] Fowler, H.W., Fowler, F.G. (Ed.), "*The Concise Oxford Dictionary.*" Book Club Associates, London, 1974.

[26] Gale, W.A., Church K.W., "*Estimation Procedures for Language Context: Poor Estimates are Worse Than None.*" In: Proc. 9th Symposium in Computational Statistics, 1990.

[27] Gale, W.A., Church K.W., "*What's Wrong with Adding One?.*" Corpus-Based Research into Language, Rodolpi, 1994.

[28] Gauch, S., Wang, J., "*A Corpus Analysis Approach for Automatic Query Expansion.*" In: Proc. 6th Int. Conf. on Information and Knowledge Management, 1997.

[29] Good, I.J., "*The Population Frequencies of Species and the Estimation of Population Parameters.*" Biometrika, 1953.

[30] Goodman, J., "*A Bit of Progress in Language Modeling - Extended Version.*" Microsoft Research, 2001.

[31] Harris, Z., "*Mathematical Structures of Language.*" Interscience Publishers, 1968.

[32] Harris, Z., "*Distributional Structure.*" Papers in Structural and Transformational Linguistics, 1970.

[33] Hatzivassiloglou, V., "*Do We Need Linguistics When We Have Statistics? A Comparative Analysis of the Contributions of Linguistic Cues to a Statistical Word Grouping System.*" The Balancing Act, The MIT Press, 1996.

[34] Hatzivassiloglou, V., McKeown, K., "*Towards the Automatic Identfication of Adjectival Scales: Clustering of Adjectives According to Meaning.*" 31st Annual Meeting of the ACL, 1993.

[35] Hearst, A.M., "*Automatic Acquisition of Hyponyms from Large Text Corpora.*" In: Proc. 14th Conference on Coputational Linguistics, 1992.

[36] Hellenic Resources Network (HRNet). http://www.hri.org.

[37] Hill, D.P., Blake, A.J., Richardson, E.J., Ringwald, M., "*Extension and Integration of the Gene Ontology (GO): Combining GO Vocabularies with External Vocabularies.*" Gene Res, 2002.

[38] Iosif, E., "*Automatic Derivation of Ontologies from Text.*" Diploma Thesis, Dept. of Electronics and Computer Engineering, Technical University of Crete, 2004.

[39] Iosif, E., Potamianos, A., "*A Soft-Clustering Algorithm for Automatic Induction of Semantic Classes.*" In: Proc. Interspeech International Conference, (to appear), August, 2007.

[40] Iosif, E., Potamianos, A., "*Unsupervised Semantic Similarity Computation using Web Search Engines.*" In: Proc. IEEE/WIC/ACM International Conference on Web Intelligence, (to appear), November, 2007.

[41] Iosif, E., Tegos, A., Pangos, A., Fosler-Lussier, E., Potamianos, A., "*Unsupervised Combination of Metrics for Semantic Class Induction.*" In: Proc. IEEE/ACL Spoken Language Technology Workshop, 2006.

[42] Jeffreys, H., "*Theory of Probability.*" Clarendon Press, 1948.

[43] Jelinek, F., "*Methods for Speech Recognition.*" MIT Press, 1997.

[44] Jelinek, F., "*Closing Remarks.*" presented at the Language Modeling Summer Workshop, Baltimore, 1995.

[45] Jelinek, F., Mercer, L.R., "*Interpolated Estimation of Markov Source Parameters from Sparse Data.*" In: Proc. Workshop on Pattern Recognition in Practice, 1980.

[46] Jiang, J.J., Conrath, D.W., "*Semantic Similarity Based on Corpus Statistics and Lexical Taxonomy.*" In: Proc. Int. Conf. in Computational Linguistics, 1998.

[47] Johnson, W.E., "*Probability: Deductive and Inductive Problems.*" Mind, 1932.

[48] Jurafsky, D., Martin, J.H., "*Speech and Language Processing.*" Prentice Hall, 2000.

[49] Jurafsky, D., Wooters, C., Segal, J., Stolcke, A., Fosler-Lussier, E., Tajchman, G., N. Morgan, N., "*Using a Stochastic Context-Free Grammar as a Language Model for Speech Recognition.*" In: Proc. ICASSP, 1995.

[50] Katz, M.S., "*Estimation of Probabilties from Sparse Data for the Language Model Component of a Speech Recognizer.*" IEEE Transactions on Acoustics, Speech and Signal Processing, Vol.35, Iss.3, 1987.

[51] Kirchoff, K., Bilmes, J., Duh, K., "*Factored Language Models Tutorial.*" Technical Report, University of Washington, 2007.

[52] Kneser, R., Ney, H., "*Forming Word Classes by Statistical Clustering for Statistical Language Modeling.*" Contributions to Quantitative Linguistics, Kluwer, 1993.

[53] Kneser, R., Ney, H., "*Improved Backing-off for N-gram Language Modeling.*" In: Proc. International Conference on Acoustics, Speech and Signal Processing, 1995.

[54] Knight, K., Luk, S., "*Building a Large-Scale Knowledge Base for Machine Translation.*" In: Proc. National Conference of Artificial Intelligence (AAAI), 1994.

[55] Kullback, S., "*Information Theory and Statistics.*" John Wiley, 1959.

[56] Lackoff, G., Johnson, M., "*Metaphors we live by.*" University of Chigago Press, 1980.

[57] Lackoff, G., Johnson, M., "*Philosophy in the Flesh: The Embodied Mind and its Challenge to Western Thought.*" Basic Books, 1997.

[58] Lee, L., "*Similarity-based Approaches to Natural Language Processing.*" PhD Thesis, Harvard University, 1997.

[59] Lee, L., "*I'm sorry Dave, I'm afraid I can't do that": Linguistics, Statistics and Natural Language Processing circa 2001.*" Computer Science: Reflections on the Field, Reflections from the Field, 2004.

[60] Li, Y., Bandar, Z.A., McLean, D., "*An Approach for Measuring Semantic Similarity between Words using Multiple Information Sources.*" IEEE Trans. on Knowledge and Data Engineering, 2003.

[61] Lidstone, G.J., "*Note on the General Case of the Bayes-Laplace Formula for Inductive or a posteriori Probabilities.*" Transactions of the Faculty of Actuaries, 1920.

[62] Lin, D., "*Principle-based Parsing without Overgeneration.*" In: Proc. 31st Annual Meeting of the Association for Computational Linguistics, 1993.

[63] Lin, D., Pantel, P., "*Induction of Semantic Classes from Natural Language Text.*" In: Proc. SIGKDD, 2001.

[64] Lyons, J., "*Linguistic Semantics, An Introduction.*" Cambridge University Press, 1995.

[65] Manning, C.D., Schutze, H., "*Foundations of Statistical Natural Language Processing.*" The MIT Press, 2000.

[66] Mihalcea, R., Dan Moldovan, D., "*Semantic Indexing using WordNet Senses.*" In: Proc. ACL-2000 Workshop on Recent Advances in Natural Language Processing and Information Retrieval, 2000.

[67] Mihalcea, R.., Moldovan, I.D., "*A Method for Word Sense Disambiguation of Unrestricted Text.*" In: Proc. 37th Annual Meeting of Association for Computational Linguistics, 1999.

[68] Mika, P.,, "*Ontologies are Us: A Unified Model of Social Networks and Semantics.*" In: Proc. ISWC2005, 2005.

[69] Miller, A.G., Beckwith, R., Felbaum, C., Gross, D., Miller, K., "*Introduction to WordNet; an On-Line Lexical Database.*" International Journal of lexicography, 1990.

[70] Miller, G., Charles, W., "*Contextual Correlates of Semantic Similarity.*" Language and Cognitive Processes, Vol.6, 1991.

[71] Mori, J., Tsujishita, T., Matsuo, Y., Ishizuka, M., "*Extracting Relations in Social Networks from the Web Using Similarity Between Collective Contexts.*" In: Proc. 20th IJCAI, 2006.

[72] Nelson, J.S., Johnston, D., Humphreys, L.B., "*Relationships in the Organization of Headings.*" Relationships in the Organization of Knowledge, Kluwer Academic Publishers, 2001.

[73] Nelson, J.S., Powell, T., Humphreys, L.B., "*The Unified Medical Language System (UMLS) Project.*" Encyclopedia of Library and Information Science, Marcel Dekker,, 2002.

[74] Ney, H., Essen, U., Kneser, R., "*On Structuring Probabilistic Dependencies in Stochastic Language Modeling.*" Computer Speech and Language, 1994.

[75] Niesler, T.R., Whittaker, E.W.D., Woodland, P.C., "*Comparison of Part-of-Speech and Automatically Derived Category-based Language Models for Speech Recognition.*" In: Proc: ICASSP, 1998.

[76] O'Hara, T., Salay, N., Witbrock, M., Schneider, D., Aldag, B., Bertolo, S., Panton, K., Lehmann, F., et al., "*Inducing Criteria for Mass Lexical Mappings using the Cyc KB, and its Extension to WordNet.*" In: Proc. 5th International Workshop on Computational Semantics (IWCS-5), 2003.

[77] Pangos, A., Iosif, E., Potamianos, A., Fosler-Lussier, E., "Combining Statistical Similarity Measures for Automatic Induction of Semantic Classes." In: Proc. IEEE Automatic Speech Recognition and Understanding Workshop, 2005.

[78] Papoulis, A., Pillai, U.S., "*Probability, Random Variables and Stohastic Processes.*" McGraw-Hill, 2002.

[79] Pargellis, A., Fosler-Lussier, E., Lee, C, Potamianos, A., Tsai, A., "*Auto-Induced Semantic Classes.*" Speech Communication, Vol. 43, 2004.

[80] Pargellis, A., Fosler-Lussier, E., Potamianos, A., Lee, C., "*A Comparison of Four Metrics for Auto-inducing Semantic Classes.*" In: Proc. ASRU, 2001.

[81] Pargellis, A., Fosler-Lussier, E., Potamianos, A., Lee, C., "*Metrics for Measuring Domain Independence of Semantic Classes.*" In: Proc. 7th European Conf. on Speech Communication and Technology, Aalborg, Denmark, 2001.

[82] Pereira, F., Naftali, T., Lee, L.,, "*Distributional Clustering of English Words.*" In: Proc. 31st Annual Meeting of ACL, 1993.

[83] Petrakis, G.M.E., Varelas, G., Hliaoutakis, A., Raftopoulou, P., "*X-Similarity: Computing Semantic Similarity between Concepts from Different Ontologies.*" Journal of Digital Information Management, 2006.

[84] Potamianos, A., Kuo, H.-K. J., "*Statistical Recursive Finite State Machine Parsing for Speech Understanding.*" In: Proc. ICSLP, 2000.

[85] Potamianos, A., Sanchez-Soto, E., Daoudi, K., "*Stream Weight Computation for Multi-Stream Classifiers.*" In: Proc. ICASSP, 2006.

[86] Price, J.P., "*Evaluation of Spoken Language Systems: the ATIS domain.*" In: Proc. DARPA Speech and Natural Language Workshop, 1990.

[87] Rada, R., Mili, H., Bichnell, E., Blettner, M., "*Development and Application of a Metric on Semantic Nets.*" IEEE Transactions on Systems, Man, and Cybernetics, Vol. 9, 1989.

[88] Raftopoulou, P., Petrakis, E., "*Semantic Similarity Measures: a Comparison Study.*" Technical Report, Intelligent Systems Laboratory, Dept. of Electronics and Computer Engineering, Technical University of Crete, 2005.

[89] Reed, S., Lenat, D., "*Mapping Ontologies into Cyc.*" In: Proc. AAAI'02 Conference Workshop on Ontologies for the Sematic Web, 2002.

[90] Resnik, P., "*Using Information Content to Evaluate Semantic Similarity in a Taxonomy.*" In: Proc. 14th International Joint Conference on Artificial Intelligence, 1995.

[91] Richardson, R., Smeaton, F.A., "*Using WordNet in a Knowledge-based Approach to Information Retieval.*" Working Paper, CA-0395, School of Computer Applications, Dublin City University, 1995.

[92] Rondriquez, A.M., Egenhofer, J.M., "*Determinig Semantic Similarity Among Entity Classes from Different Ontologies.*" IEEE Transactions on Knowledge and Data Engineering, Vol.15, 2003.

[93] Rosenfeld, R., "*Adaptive Statistical Language Modeling: A Maximum Entropy Approach.*" PhD dissertation, Computer Science Department, Carnegie-Mellon University, 1994.

[94] Rosenfeld, R., "*A Maximum Entropy Approach to Adaptive Statistical Language Modeling.*" Computer Speech and Language, Vol.10, 1996.

[95] Rosenfeld, R., "*Two Decades of Statistical Language Modeling: where do we go from here?.*" Proceedings of the IEEE, Vol.88, Iss.8, 2000.

[96] Rubenstein H., Goodenough, B.J., "*Contextual Correlates of Synonymy.*" Communications of the ACM, Vol.8, 1965.

[97] Sahami, M., Heilman, T., "*A Web-based Kernel Function for Measuring the Similarity of Short Text Snippets.*" In: Proc. Int. WWW2006 Conf., 2006.

[98] Sahlgren, M., "*The Word-Space Model: using distributional analysis to represent syntagmatic and paradigmatic relations between words in high-dimensional vector spaces.*" Phd Thesis, Department of Linguistics-Computational Linguistics, Stockholm University, 2006.

[99] Schutze, H., "*Word Space.*" In: Proc. Conference on Advances in Neural Information Processing Systems (NIPS), 1993.

[100] Schutze, H., Pedersen, J., "*Information Retrieval Based on Word Senses.*" In: Proc. 4th Annual Symposium on Document Analysis and Information Retrieval, 1995.

[101] Sebastiani, F., "Machine Learning in Automated Text Categorization." ACM Computing Surveys, Vol. 34, 2002.

[102] Shannon, C.E., "*A Mathematical Theory of Communication.*" The Bell System Technical Journal, Vol. 27, 1948.

[103] Siu, K.-C., Meng, H.M., "*Semi-automatic Acquisition of Domain-specific Semantic Structures.*" In: Proc. EUROSPEECH, 1999.

[104] Snow, R., Jurafsky, D., Ng, Y. A., "*Learning Syntactic Patterns for Automatic Hypernym Discovery.*" Advances in Neural Information Processing Systems, Vol. 17, 2004.

[105] STDS url:. http://mcmcweb.er.usgs.gov/stds/.

[106] Stolcke, A., "*SRILM - An Extensible Language Modeling Toolkit.*" In: Proc. International Conference on Spoken Language Processing, 2002.

[107] Tversky, "*Features of Similarity.*" Psycological Review, Vol.84, 1977.

[108] Voorhees, E., "*Query Expansion using Lexical-Semantic Relations.*" In: Proc. 17th Int. ACM SIGIR Conf. on Research and Development in Information Retrieval, 1994.

[109] Ward, H.W., "*The CMU Air Travel Information Service: Understanding Spontaneous Speech.*" In: Proc. DARPA Speech and Natural Language Workshop, 1990.

[110] Witten, H.I., Bell, C.T., "*The Zero-Frequency Problem: Estimating the Probabilities of Novel Events in Adaptive Text Compression.*" IEEE Transactions on Information Theory, 1991.

[111] YahooSearchAPI. http://developer.yahoo.com/search/.

[112] Yokoyama, T., Shinozaki, T., Iwano, K., Furui, S., "*Unsupervised Language Model Adaptation using Word Classes for Spontaneous Speech Recognition.*" In: Proc. Spontaneous Speech Processing and Recognition Workshop, 2003.

[113] Young, S., Evermann, G., Kershaw, D., Moore, G., Odell, J., Olasson, D., Povey, D., Valtchev, V., Woodland, P., "*The HTK Book (for HTK Version 3.2).*" Cambridge University Engineering Department, 2002.

[114] Zhu, X., Rosenfeld, R., "*Improving Trigram Language Modeling with the World Wide Web.*" In: Proc. International Conference on Acoustics, Speech, and Signal Processing, 2001.