Bar Ilan University

The Department of Computer Science

# Directional Distributional Similarity for Lexical Expansion

by

**Lili Kotlerman**

Submitted in partial fulfillment of the requirements for the Master's Degree in the Department of Computer Science, Bar-Ilan University

Ramat Gan, Israel April 2009, Nissan 5769

This work was carried out under the supervision of Dr. Ido Dagan

The Department of Computer Science

Bar-Ilan University

Israel

# TABLE OF CONTENTS

# ABSTRACT

One of the fundamental problems faced by automatic text understanding is the variability of semantic expression, where the same meaning may be expressed in many different ways. To address this problem, many Natural Language Processing (NLP) applications utilize *lexical expansion*, altering a given text by terms of similar meaning.

A widely used approach for automatic learning of semantically similar words is based on the *Distributional Similarity Hypothesis*, which suggests that words occurring within similar contexts are semantically similar (Harris, 1968). The main disadvantage of common state-of-the-art distributional similarity approaches is that they have substantially low precision with respect to the lexical expansion task. Additional imperfection of distributional similarity is its being a symmetric model, while often semantic expansion requires a directional approach.

Recent works on distributional similarity proposed a novel view called *Distributional Inclusion*, assuming that prominent semantic traits of an expanding word should co-occur with the expanded word as well. In this work we investigated the applicability of this rational for automatic corpus-based acquisition of directional similarity rules for lexical expansion. We analyzed the components and parameters of distributional similarity measures that capture distributional inclusion, formulated the desired properties of an inclusion-based similarity measure and defined novel directional measures of semantic similarity. We performed application-oriented evaluation of the defined inclusion measures within two different NLP tasks and showed considerable improvement as compared to earlier state-of-the-art measures.

## ACKNOWLEDGEMENTS

This thesis paper has been accomplished with the help of a number of people, and I wish to express my heartfelt thanks to them.

I am grateful to Dr. Ido Dagan at Bar-Ilan University for his supervision of the thesis and for his tutoring on how to observe, analyze and scientifically formalize. It has been a great pleasure to work with him on this subject and I have learned a lot.

I would like to thank Dr. Maayn Geffet at Bar-Ilan University for her guidance at the beginning of the way and helpful comments throughout the work.

I would like to thank our NLP group for their mutual support.

I would like to thank my husband and parents for understanding, supporting and encouraging, and my sons for simply loving me.

Finally, I would like to thank the NEGEV project (www.negev-initiative.org) for a partial support of this work.

# LIST OF TABLES AND FIGURES

## List of Tables

## List of Figures

# 1 INTRODUCTION

One of the fundamental problems faced by automatic text understanding is the variability of semantic expression, where the same meaning may be put into words in many different ways. To address this problem, many Natural Language Processing (NLP) applications utilize lexical expansion, in which a given text (usually a query) is altered by adding terms of similar meaning. Such systems need to recognize when the meaning of one word can be inferred from another word. For example, in Information Retrieval, the word *company* in the query might be substituted in the text by *firm*, *automaker*, etc.

A widely used approach for automatic learning of semantically similar words is based on the *Distributional Similarity Hypothesis*, which suggests that words occurring within similar contexts are semantically similar (Harris, 1968). Thus, the problem of comparing two word meanings comes to the problem of comparing the words' contexts. In the computational setting word contexts are represented by weighted feature vectors, where features typically correspond to other words that co-occur with the characterized word in the same context. Similarity measures then compare a pair of feature vectors that characterize two words.

As shown by Mirkin et al. (2009), the main disadvantage of the state-of-the-art distributional similarity measures is that, indeed yielding considerable recall increase, they have substantially low precision, which causes many applications to avoid the use of this lexical expansion resource. Additional imperfection of distributional similarity is its being a symmetric, non-directional model, while many semantic relations require a directional approach. For example, usually we can replace a general word (*hypernym*), e.g. *fruit*, with a more specific (*hyponym*) distributionally similar word, e.g. *lemon* (*fruits grow in a garden* implies *lemons grow in a garden*), but a reverse substitution cannot take place (*lemons are yellow and sour* does not allow to conclude that *fruits are yellow and sour*).

As a refinement of the distributional similarity scheme, Geffet and Dagan (2005) suggested two *Distributional Inclusion Hypotheses*. Their motivating assumption is that if a word *w* entails another word *v* (under some corresponding senses of the two words), then all the prominent features of *w* are expected to co-occur with *v* in a sufficiently large corpus. Their research showed that distributional inclusion indeed can be used to improve automatic acquisition of pairs of semantically similar words: their web-based *Inclusion Testing Algorithm (ITA)* managed to filter out approximately 85% of inappropriate word pairs in their initial test set.

Due to the data sparseness problem of a given corpus, it is impossible to observe *complete* inclusion of *all* the prominent features of one word in the feature vector of the other word. ITA uses the web to overcome this problem, which is an extremely time-consuming approach, implemented by Geffet and Dagan for a small test sample in order to prove the validity of their *Distributional Inclusion Hypotheses.* It is obvious that such a web-based algorithm is not applicable for a real-life broad-coverage setting. Therefore in this work we aimed to develop similar inclusion testing methods that would not go beyond the given corpus data.

In this work we analyzed the behavior of feature vectors from a distributional inclusion perspective, in order to apply distributional inclusion for corpus-based learning of lexical expansion rules. Since a simple binary test of *complete* inclusion employed by the ITA is not applicable for a corpus-based algorithm due to data sparseness, we investigated statistical tests of *partial inclusion,* aiming to develop a statistical model that would assign to each pair of words (*w*,*v*) an inclusion score, reflecting the level of the algorithm's confidence that distributional inclusion indeed holds for the two words.

We investigated the factors, which impact the distributional inclusion scheme, such as selection of the features that should be used for inclusion testing, influence of their relevance ranking in the feature vectors of both words etc., and formulated the desired properties of an inclusion-based measure of semantic similarity. In view of these properties we analyzed the existing, *Precision*-based, inclusion measures and defined our novel inclusion measures based on the *Average Precision* (*AP*) metric.

We performed application-oriented evaluations of the inclusion measures using two different NLP tasks. The results of our evaluation assessed the appropriateness of the defined properties and confirmed the validity of the *Average Precision*-based approach to inclusion testing. The modifications, proposed for the classical *AP* measure in order to satisfy the defined properties, proved to be valid. One of the novel measures, which includes all proposed modifications, showed the best performance within both tasks, considerably improving earlier state-of-the-art measures. In addition, this measure proved to be least sensitive to the major parameter of the distributional inclusion scheme.

In Section 2 we review the lexical expansion mechanism, state-of-the-art methods of automatic acquisition of semantically similar words, and related work on distributional inclusion. Section 3 describes our investigation of the distributional inclusion scheme. Section 4 summarizes the results of this analysis and defines our novel distributional similarity measures, based on the formulated desired properties of an inclusion measure. Finally, Section 5 presents the evaluation results and analysis, performed over two different NLP application datasets.

During this work we detected a few worthwhile directions for further research, which are described in Section 6, along with our conclusions.

# 2 BACKGROUND

## 2.1 Lexical Expansion

Lexical expansion is widely used in Natural Language Processing applications in order to overcome lexical variability, when the same concept may be expressed by, or inferred from, different texts. For example, Question Answering systems need to recognize corresponding parts in the question and in the answer passage, even though these parts may be phrased in different words. Summarization engines need to identify different phrases of equivalent meaning in order to reduce redundancy. Information Retrieval and Extraction systems aim to discover as many relevant texts as possible, without making the user reformulate the original query.

Lexical expansion means altering a given text (usually a query) by adding terms of similar meaning. We will call the relation between the original term, which the system aims to expand, and the altering term of similar meaning the *expansion relation*. Thus, *expansion rules* used for lexical expansion are rules of the type *expanding term* → *expanded term*, denoting that occurrence of the *expanding* term in a text is likely to imply a reference to the meaning of the *expanded* term.

### 2.1.1 Lexical Reference

In order to evaluate whether the meaning of a lexical item of one text is expressed also within the other text, Glickman et. al (2006) proposed the following definition: a word or a word compound *w* is *lexically referenced* by a text *t* if there is an explicit or implied reference in *t* to a possible meaning of *w*.

In terms of lexical expansion, this criterion allows us to define whether a text *t*, in which the original query term *w* was replaced with a semantically similar term, indeed references the meaning of the original query term.

### 2.1.2 Lexical Entailment

Geffet and Dagan (2004) proposed a direct criterion for semantic word similarity, termed *substitutable lexical entailment* (*lexical entailment*, in short). *Lexical entailment* is a new type of semantic relationship, which directionally holds under some common matching senses of a pair of given terms (*w*, *v*), where *w* entails *v*, if both of the following conditions are fulfilled:

1. *Word meaning entailment*: whether the meaning of *w* implies the meaning of *v*;
2. *Substitutability*: whether *w* can semantically substitute *v* in some natural contexts, such that the meaning of the modified context entails the meaning of the original one.

We note that *lexical entailment* can be seen as a somewhat stricter version of the *expansion* relation: words fulfilling the *lexical entailment* criterion are by definition useful for lexical expansion, but this criterion does not cover many additional words that can also be useful for this purpose. For example, word pairs like *breastfeeding→birth*, *divorce→marry*, *prosecutor→indictment*, *barrel→crude oil* don't correspond to the *lexical entailment* criterion, but conform to the criteria of the *expansion* relation.

## 2.2 Distributional Similarity

Much of the work on automatic acquisition of semantically similar words is based on the *Distributional Similarity Hypothesis*, which assumes that semantically similar terms appear in similar contexts. This hypothesis suggests a methodology of comparing words by comparing the contexts in which they occur. The process of calculating distributional word similarity involves two main phases. First, weighted context features are assigned for each word, and then the weighted contexts of different words are compared by some vector similarity measure.

Distributional Similarity has been an active research area for a couple of decades (Hindle (1990); Ruge (1992); Grefenstette (1994); Lee (1997); Lin (1998a); Dagan et al.

(1999); Weeds and Weir (2003)). Various context-feature weighing functions and vector similarity measures were suggested. Since our research doesn't aim to improve the current distributional similarity methods, we leave analysis and comparison of different distributional similarity schemes beyond our scope, focusing on the state-of-the-art techniques. As such, we have chosen the widely cited and competitive (as shown by Weeds and Weir (2003)) measure of Lin (1998a), from now on termed as *LIN*, as representing stare-of-the-art symmetric similarity measures.

### 2.2.1 The Distributional Similarity Measure of Lin

In the computational setting each word *w* is represented by a feature vector, where an entry in the vector corresponds to a feature *f*. Each feature represents another word (or term) which co-occurs with *w*, and also specifies the syntactic relation between the two words. The degree of statistical association between the feature and the corresponding word is reflected by a feature weighing function. The *LIN* measure uses (point-wise) *Mutual Information* (*MI*) to assign feature weights:

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

Features that constitute strong collocations with the target word are expected to yield high *MI* scores. A common practice is to filter out features by minimal frequency and weight thresholds.

Once feature vectors have been constructed, the similarity between two words is defined by a vector similarity metric, grounded on principles of Information Theory. It computes the ratio between the information shared by the features of both words and the sum over the features of each word:

$$sim_{LIN}(w,v) = \frac{\sum_{f \in F(w) \cap F(v)} (weight_{MI}(w,f) + weight_{MI}(v,f))}{\sum_{f \in F(w)} weight_{MI}(w,f) + \sum_{f \in F(v)} weight_{MI}(v,f)},$$

where *F(word)* is the set of features in the vector of the corresponding word.

### 2.2.2 Bootstrapping of Feature Weights

Geffet and Dagan (2004) proposed a new feature weighing function based on bootstrapping of feature weights (*BFW*), whose underlying idea is to promote features of a word *w* that characterize many words, which are known (initially) to be similar to it. The initial similarities are computed using the *LIN* method and then the bootstrapped feature weights are calculated. The method is supposed to concentrate characteristic word features at the top of the feature vectors, thus allowing aggressive feature reduction, after which the $sim_{LIN}$ function (defined in Section 2.2.1) is employed to re-compute the similarity lists using the new feature weights.

This approach was reported to yield more accurate distributional similarity lists than the state-of-the-art *LIN* measure and to produce feature vectors of better quality. However, applying inclusion testing to *BFW*-scored vectors did not yield better results as compared to using *MI*-scored vectors generated by the *LIN* algorithm. Therefore we do not discuss this bootstrapping approach in more detail.

### 2.2.3 Directional Similarity Measures vs. Symmetric Measures

A well-known imperfection of distributional similarity measures is their being a symmetric, non-directional model, while many semantic relations require a directional approach. Most applications, which make use of distributional similarity lists, would prefer to deal with directional word pairs instead of symmetric ones. In Information Retrieval, for instance, user looking for "*baby food*" will be contented to find documents about "*baby pap*", "*baby juice*" etc. (since *pap*→*food*, *juice*→*food*), but a person looking for "*frozen juice*" will not be satisfied by "*frozen food*" (since *food*→*juice*). In similar manner, many kinds of applications using lexical expansion seek directional word pairs, thus making directional similarity measures clearly preferable.

Notwithstanding this evident need of directional similarity measures, research in this perspective counts, to the best of our knowledge, only few works (Weeds and Weir

(2003), Weeds et al. (2004), Geffet and Dagan (2005), Szpektor and Dagan (2008)) and is utterly lacking. In Section 2.3 we describe the approaches presented in these works, which are extended in our research.

## 2.3 Distributional Inclusion

Weeds et al. (2004) attempted to refine the distributional similarity goal to predict whether one term is a generalization (or specification) of the other. They presented a distributional generality concept, expected to correlate with semantic generality. Their assumption is that the majority of the features of the more specific word are expected to be included in the set of features of the more general one. They proposed a directional measure for learning such relation between two words, which we describe in Section 2.3.2.

Extending this rationale, Geffet and Dagan (2005) suggested that if the meaning of a word *w* entails that of another word *v*, then it is expected that all the prominent semantic traits of *w* will be possessed also by *v*. Since in the distributional model semantic traits of words are represented by word contexts (features), the characteristic contexts of *w* are expected to be included within all *v*'s contexts (but not necessarily amongst the most characteristic ones for *v*). Their research showed that indeed there is a strong correlation between the complete inclusion of the prominent features and *lexical entailment*. Formalizing this observation, they defined two *Distributional Inclusion Hypotheses*, which correspond to the two directions of inference, relating distributional inclusion and lexical entailment.

Let $w_i$ and $v_j$ be two word senses of the words *w* and *v*, correspondingly, and let $w_i$ => $v_j$ denote the (directional) *lexical entailment* relation between these senses. Assume further that there exists a measure that determines the set of characteristic features for the meaning of each word sense.

Then the distributional inclusion hypotheses state:

***Hypothesis I:***

If $w_i$ => $v_j$ then all the characteristic features of $w_i$ are expected to appear with $v_j$ in a sufficiently large text corpus.

***Hypothesis II:***

If all the characteristic features of $w_i$ appear with $v_j$ then we expect that $w_i$ => $v_j$.

According to these hypotheses, given a pair of words (*w*,*v*) we can check whether all the characteristic features of *w* are included inside the vector of the word *v* and in this way define whether th *lexical entailment* relation holds between the two words. We will further refer to features of *w* used for the inclusion test as *tested* features, and to the features that were found inside the vector of *v* as *included* features.

### 2.3.1 Web-based Inclusion Testing Algorithm

The Web-based Inclusion Testing Algorithm (ITA) was developed by Geffet and Dagan (2005) to check the validity of the *Distributional Inclusion Hypotheses* described above. Their research showed that the ITA may be used for predicting lexical entailment, filtering out non-entailing word pairs produced by the distributional similarity model.

Although the distributional inclusion hypotheses are formulated for word senses, ITA works at the word level due to the lack of a robust sense tagging tool or large manually tagged resource. Given a test (directional) word pair (*w*,*v*) ITA performs three steps:

1) Corpus-based generation of tested features. Bootstrapping of feature weights (described in Section 2.2.2) is applied and the top-100 features are taken as the most characteristic for *w* (the tested features).
2) Corpus-based inclusion test (using the same corpus as for the creation of the set of tested features). The tested features of *w* that co-occur with *v* in the corpus are marked as included.

3) Complementary web-based inclusion test, performed to overcome data sparseness of the corpus. Features left unmarked in *w*'s set of tested features are checked for co-occurrence with the word *v* in the web. Co-occurring features are marked as included.

Inclusion in the given direction holds if all the tested features were marked as included. Word pairs that satisfy this inclusion criterion are predicted to be *lexically entailing* in the specified direction. We note that the test performed is a simple binary test, i.e. the words are judged as entailing only if *all* the tested features of one word co-occurred (in the corpus or in the web) with the other word.

### 2.3.2 Statistical Inclusion Measures

In this section we describe earlier measures of distributional inclusion. In our work we extended these measures and performed empirical evaluation and comparison of their performance for the lexical expansion task.

Weeds and Weir (2003) proposed a directional measure for learning hyponymy between two words (*hyponym*→*hypernym*) by calculating the coverage of hyponym's features by the hypernym. The proposed coverage measure is termed *Precision* (we will further address it as *WeedsPrecision*) and is defined as follows:

$$WeedsPrecision(w \to v) = \frac{\sum_{f \in FV(w) \cap FV(v)} weight(f,w)}{\sum_{f \in FV(w)} weight(f,w)},$$

where *FV(word)* is the set of features in the vector of the corresponding word, and *weight(f,w)* is the weight[1] of the feature *f* in the vector of *w*. We see that *WeedsPrecision* is virtually a measure of distributional inclusion, calculating the ratio between the weighted sum of the included features and the weighted sum of the features of the expanding word.

[1] In their work they also used *Mutual Information* to assign feature weights.

Having applied the *WeedsPrecision* measure for learning entailment rules for unary lexical-syntactic templates (parse fragments including a single variable), Szpektor and Dagan (2008) observed that this measure tends to prefer directional rules in which the *entailing* left-hand-side template is infrequent, thus generating highly-scored incorrect rules. They proposed to penalize infrequent templates by means of the symmetric measure of *LIN* and called the resulting measure *Balanced-Inclusion (*we will further address it as *balPrecision*):

$$balPrecision(w \to v) = sim_{LIN}(w,v) \cdot WeedsPrecision(w \to v)$$

The *balPrecision* measure is reported to significantly outperform both the *LIN* and the *WeedsPrecision* methods.

# 3 ANALYSIS OF DISTRIBUTIONAL INCLUSION

The main goal of our work was to examine the possibility of using the *Distributional Inclusion Hypotheses* for corpus-based learning of directional expansion rules. Due to the data sparseness of the corpus, we cannot expect to observe inclusion of *all* the tested features of one word in the vector of the other word; therefore our approach involves evaluation of *partial inclusion* based on the proportion of included features.

In this section we analyze the factors, influencing inclusion reliability, in order to detect desired behaviors of a probabilistic or statistical measure of partial inclusion, which we aim to define (further *inclusion measure* for brevity).

## 3.1 Data Sources

With an intension to develop a good inclusion measure, we analyzed feature vectors of known similar and non-similar word pairs, aiming to observe some distinctive behavior of features in both cases. Do to the fact that *expansion* is not a well-defined relation there are no available resources of expanding/non-expanding word pairs. Therefore all the data analysis in current section is performed using the experimental dataset from Geffet and Dagan (2004), judged according to the *lexical entailment* criterion. The validity of this decision can be checked only empirically, but our manual investigation allowed us to conclude that this dataset complies with our analysis needs better than any other available dataset, since as explained in Section 2.1.2, *lexical entailment* is virtually a stricter version of the *expansion* relation.

The dataset consists of manually judged similarity lists, created for a sample of 30 frequent common nouns[1] by the *LIN* and *BFW* algorithms using an 18 million word subset of Reuters RCV1 corpus[2]. The dataset includes of 1067 entailing and 2703 non-

[1] The 30 nouns were sampled from the corpus nouns with frequency higher than 500 in their corpus.

[2] Reuters Corpus, Volume 1, English Language, 1996-08-20 to 1997-08-19

entailing word pairs. Examples of entailing and non-entailing word pairs from this dataset are presented below in Table 1.

| Entailing pairs | Non-entailing pairs |
|---|---|
| air force => warplane | abuse ≠> bribe |
| argument => reason | broker ≠> journalist |
| broker => trader | ceasefire ≠> federation |
| care => treatment | central bank ≠> army |
| chairman => chief executive | chairman ≠> founder |
| debt => loan | murder ≠> war |
| government => state | performance ≠> success |
| prison term => sentence | research ≠> management |
| town => city | town ≠> airport |
| war => aggression | vessel ≠> warplane |

Table 1 – Examples of entailing and non-entailing pairs used for the analysis of inclusion.

For vector creation in our work we used the Reuters RCV1 corpus, containing about 800,000 news stories, each from a few hundred to several thousand words in length. Throughout the work we used syntactic features, generated by parsing the corpus using the Minipar[1] dependency parser (Lin, 1993). A feature is defined as a triple <*word*, *syntactic relation*, *direction*>, where *word* represents word's lemma and its part-of-speech tag; *syntactic relation* specifies the relation between the feature and the word it characterizes; *direction* reflects the role of the feature in the syntactic relation – it can be the head of the relation, the word's modifier or a member of a symmetric relation.

The web-based ITA algorithm of Geffet and Dagan (2005), described in Section 2.3.1, used *BFW*-scored vectors for creating tested feature sets. The underlying idea for this decision was that the bootstrapping method leads to concentration of characteristic word features at the top of the feature vectors, thus making the use of the new vectors most natural for selection of tested features.

However, we decided to check the use of *MI*-scored vectors for inclusion testing as well. First, *MI*-based feature scoring, which is basically the common practice, generates vectors of quite good quality deserving further investigation. Second,

[1] http://www.cs.umanitoba.ca/~lindek/minipar.htm

generating *BFW*-scored vectors is rather time and space consuming, so we found it expedient to examine the potential of inclusion testing without involving *BFW*.

We also performed inclusion testing using the *BFW*-scored vectors. The results obtained for this source of feature vectors were noticeably lower than for the *MI*-scored vectors. Our analysis revealed several defects of the *BFW*-scored vectors, such as an inconsistency between feature ranking and their relevance to the word they represent. At the same time, the amounts of included features are on the whole much higher than for the *MI*-scored vectors (60-70% vs. 10-15%). We conclude that the bootstrapping algorithm has certain potential and is worse further research and revision. For now, in this thesis we concentrate on using the *MI*-scored vectors.

### 3.2 Selection of Tested Features

One of the factors that influences distributional inclusion is the amount of features chosen to be the tested features of the expanding word.

If we would perform a binary test of *complete* inclusion then we would find it more trustworthy if it were performed using a sufficiently large quantity of tested features. For instance, we would trust inclusion of 100 tested features more than inclusion of 10. On the other hand, selecting too many tested features we run the risk that *complete* inclusion will no longer take place.

In case of *partial inclusion* the reliability of the inclusion test would also be affected by inappropriate selection of the amount of tested features:

- Selecting a short set of tested features does not allow to consider the inclusion test statistically reliable. It can also leave the majority of relevant features out of this set, thus lowering the chance of a valid word pair to reach high levels of inclusion.

- A very large set may include many irrelevant tested features, which are not supposed to co-occur with the entailed word and thus can lower the score of a valid pair of words.
- A "tradeoff" amount of tested features may turn out either too short or too long for different word pairs, unevenly influencing their reliability scores.

Table 2 shows typical dependence of inclusion level on the amount of selected tested features. We see that choosing top-100 tested features hardly allows us to distinguish between a similar and a non-similar word pairs, while increasing the number of tested features to 200 displays a much clearer gap of scores. Such notable difference between the scores of similar and non-similar pairs remains for rather an ample range of 200-1000 tested features and comes to naught at top-5000 (which in this case virtually means using all the features in the vector as tested).

| Amount of tested features | Percent of included features | |
|---|---|---|
| | entailing pair<br>(election → vote ) | Non-entailing pair<br>(election → reform ) |
| top-100 | 0.04 | 0.03 |
| top-200 | 0.09 | 0.01 |
| top-500 | 0.12 | 0.02 |
| top-1000 | 0.13 | 0.04 |
| top-5000 | 0.22 | 0.19 |

Table 2 – Typical example of influence of tested features selection on the level of inclusion. The table shows the percent of included tested features for an entailing and a non-entailing word pair. The pair *election→vote* is an entailing pair ($sim_{Lin}$=0.15) and the pair *election→reform* is a non-entailing pair ($sim_{Lin}$=0.11).

Our analysis showed that a specific number of tested features, allowing evident distinction between similar and non-similar word pairs, differs from pair to pair in range of about 100-1500 tested features, but the tendency is common: selecting too short and too long sets of tested features makes the task almost impossible, while a good selection of tested features brings rather good results.

The *Distributional Inclusion Hypotheses* by definition imply availability of a measure, determining the set of characteristic features that should be used for inclusion testing, but there is no such state-of-the-art measure actually available and most applications content themselves with threshold tuning. Virtually, there are 3 possible ways to select some top features of a word to constitute its set of tested features:

- select top-*k* features of a vector;
- define a threshold of feature weight;
- select top-*k* percents of the vector's features.

Each of the approaches has its constraints, briefly discussed below.

**Using top-*k* features of a vector**. The web-based ITA algorithm involved a test of *complete* inclusion of the top-100 tested features for *BFW*-scored vectors. In our settings adopting this threshold is not appropriate, since *BFW*-scored vectors are considerably shorter than the *MI*-scored ones.

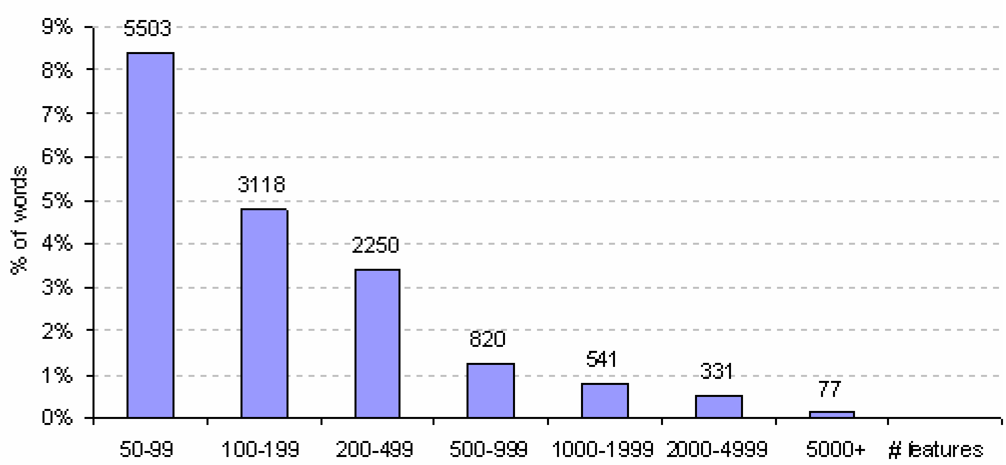


Figure 1 – Distribution of the *MI*-scored vector lengths. The diagram shows the number of feature vectors of different lengths out of 12,640 vectors that have more than 50 features each, which makes up 19.3% of all the vectors in our nouns database. Vectors shorter than 50 features constitute the other 80.7% (52,895 vectors).

As we can see from Figure 1, using top-100 features would not be suitable for the greater part of the words in our database, as it would imply using their full feature vector instead of some top characteristic features. Similarly, for long vectors it would imply using short and thus not indicative sets of tested features. We see that the amount of features in a vector varies from a couple of dozens to several thousands, thus making it almost impossible to tune a good uniform top-*k* threshold. An additional limitation is that a threshold, tuned for a definite scoring algorithm, may turn out unsuitable for another algorithm, as we see it in the case of the *LIN* and *BFW* measures, because different algorithms produce vectors of different lengths. The same problem takes place when applying a definite scoring method to different corpora, since the lengths of the vectors may vary considerably for different domains and corpus types.

**Using a threshold of feature weights**. This seems to be the most natural and precise way of selecting the tested features, because weights are supposed to reflect the relevance of features for the word they represent. Yet, having analyzed the feature vectors of words of different occurrence frequencies, we understood that identifying some common meaningful threshold is not possible, since the *Mutual Information* measure depends to a great extent on occurrence and co-occurrence frequencies of words and features and doesn't yield homogenous scores. Thus, the disadvantages of setting up some weight threshold are similar to those of using top-*k* tested features, since we have to tune the threshold separately for different scoring algorithms, which in their turn depend on corpus size, corpus type and word frequencies.

**Using top-*k* percents of the features.** This simple threshold solves the problem of different lengths of the vectors produced by different algorithms for words of different frequencies using corpora of different sizes and types. This threshold still involves deciding what should be the value of *k*, but this decision is easier and more universal.

We see thus, that selection of tested features is indeed a non-trivial factor influencing the inclusion trustworthiness. The solution that seems to us the most convenient and will be one of the goals of our research lies in developing a measure of inclusion reliability, which would be minimally vulnerable to lame selection of the tested features.

In further sections we aim to define additional ways to both ameliorate the distinguishing ability of inclusion measures and overcome the dependency on selection of the tested features.

### 3.3 Feature Relevance to the Expanding Word

One of the evident characteristics of the tested features, not used by the ITA algorithm, is that the features inside each vector constitute a ranked list according to their relevance to the corresponding word.

By definition, more relevant features are supposed to have higher weights and, correspondingly, to be placed at higher ranks in the vectors. We thus presume that features at higher ranks are more important in the view of inclusion reliability. For example, inclusion of 10 features from the first dozen of tested features is supposed to be of more importance as compared to inclusion of 10 features scattered along the tail of the list. In case of choosing some constant number of features from the top of each vector as its tested features, the influence of features' rank is likely to turn out minor, since all the features chosen are placed rather close to the top of the vector. Hence, in case of selecting top percent of features as tested or in case of involving a weight threshold, as well as in case of using all the available features for inclusion testing, the number of tested features can become sizable enough to attach significance to their rank.

Figure 2 shows a typical example of the distribution of included features according to their rank in the vector of tested features. Analyzing the data presented by Figure 2, we see that:

1. Irrespective of the amount of the tested features:
    - similar word pairs have more included features at higher ranks than non-similar ones;
    - non-similar pairs have more included features towards the tail of the list.

    This allows us to conclude that giving greater impact to higher-ranked features can improve the distinguishing ability of an inclusion measure.

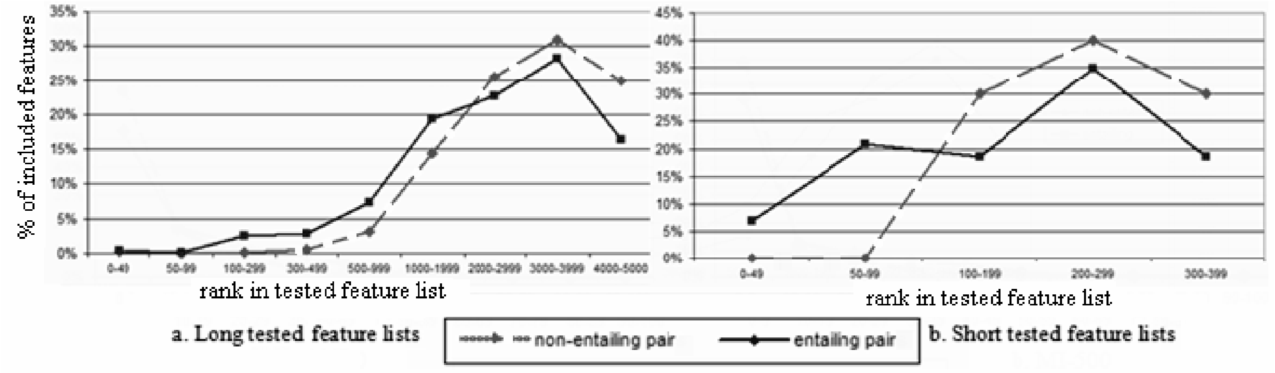


Figure 2 – Typical example of distribution of included features according to their rank in the vector of tested features. The x axis shows rank ranges inside the lists of tested features; the y axis shows the percent of included features at the corresponding rank. Each diagram shows the distribution for an entailing pair and a non-entailing word pair. Diagram *a* represents the distribution for the word pairs *election→vote* (entailing) and *election→reform* (non-entailing), having long vectors. Diagram *b* represents the distribution for the word pairs *sweet→candy* (entailing) and *sweet→brandy* (non-entailing), having short vectors.

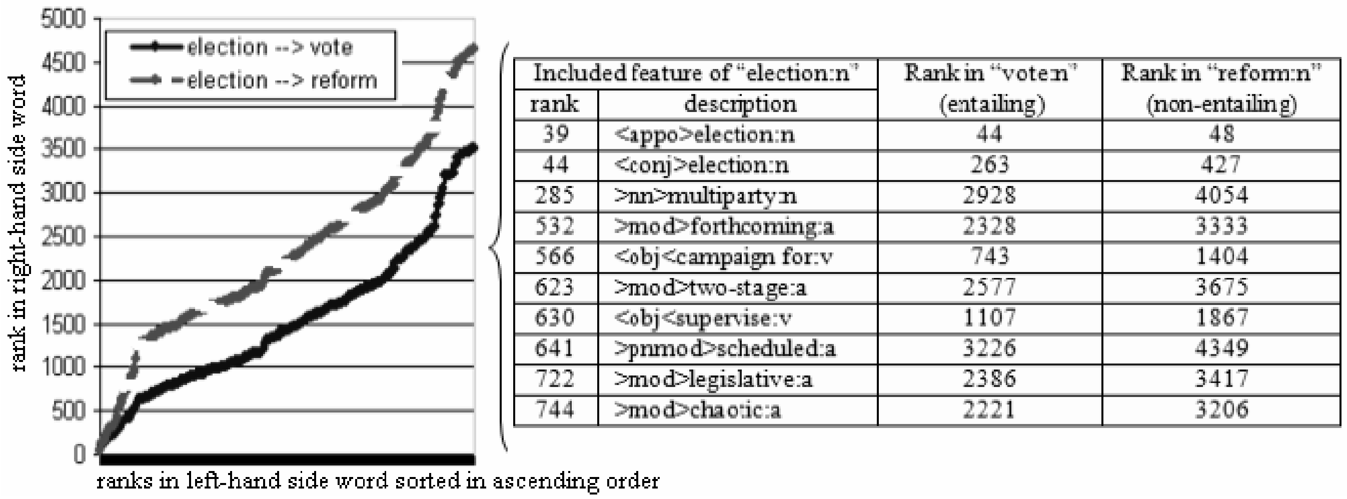
2. Both for similar and non-similar word pairs, the amounts of lower-ranked included features grow as compared to the amounts of included features at higher ranks. Such behavior equally holds for short and long lists of tested features. This allows us to admit that emphasizing the impact of higher-ranked features and diminishing the impact of the lower-ranked ones can also help to achieve greater robustness in respect of selecting the tested features.

We conclude thus, that reflecting the relevance of the included features to the expanding word is indeed expedient for an inclusion measure.

### 3.4 Feature Relevance to the Expanded Word

Having examined the influence of the rank of included features inside the list of the tested features, we found it instructive to study the influence of their rank in the vector of the expanded word. We hypothesized that finding tested features at top ranks of the vector of the expanding word may imply higher similarity between the two words than finding them at the tail of the vector. In other words, if tested features, which are by definition highly relevant to the expanding word, turn out highly relevant to the expanded word as well (and not simply co-occurring with it in some infrequent contexts), this may testify to the expansion relationship between the two words.

We investigated the influence of the rank of included features inside the vector of the expanded word, and indeed detected certain regularity.



| Included feature of "election:n" | | Rank in "vote:n" | Rank in "reform:n" |
|---|---|---|---|
| rank | description | (entailing) | (non-entailing) |
| 39 | <appo>election:n | 44 | 48 |
| 44 | <conj>election:n | 263 | 427 |
| 285 | >nn>multiparty:n | 2928 | 4054 |
| 532 | >mod>forthcoming:a | 2328 | 3333 |
| 566 | <obj<campaign for:v | 743 | 1404 |
| 623 | >mod>two-stage:a | 2577 | 3675 |
| 630 | <obj<supervise:v | 1107 | 1867 |
| 641 | >pnmod>scheduled:a | 3226 | 4349 |
| 722 | >mod>legislative:a | 2386 | 3417 |
| 744 | >mod>chaotic:a | 2221 | 3206 |

Figure 3 – Typical example of influence of the rank of included features in the vector of the expanded word. The diagram shows the ranks for word pairs *election→vote* (entailing) and *election→reform* (non-entailing), using the features that turned out included for both word pairs. The x axis shows the rank of the included features in the list of the tested features (sorted in ascending order); the y axis shows the rank of the corresponding features in the vector of the expanded word. The table on the right shows an example of the data represented by the diagram.

Figure 3 shows the typical behavior of included features in the perspective of relevance to the expanded word. We see that the ranks of the same included features in the expanded word's vector are consistently lower for entailing pairs than for non-entailing ones. This means that an inclusion measure can improve its distinguishing ability by emphasizing higher relevance of included features to the expanded word.

### 3.5 Vector Length of the Expanded Word

The last factor we would like to analyze is the length of the vector of the expanded word. At first sight, we are likely to find higher quantity of tested features inside a long vector than inside a short one, and hence would like to normalize our inclusion measure using the length of the vector in order not to demote valid expansion rules having expanded word with a short vector. However, our analysis showed that expanded words with short vectors are not quite reliable from our perspective.

The reason of such unreliability of short-vector expanded words is that the number of included features is virtually limited by the number of features in the vector of the expanded word. Consequently, some arbitrary expanding words with short feature vectors are likely to have high coverage and thus yield higher scores, while valid expanding words with longer vectors are demoted.

| | Entailing pairs | Non-entailing pairs |
|---|---|---|
| Less than 500 features | 7 % | 20 % |
| 500-2,000 features | 23 % | 26 % |
| 2,000-5,000 features | 51 % | 45 % |
| 5,000- 30,000 features | 19 % | 9 % |

Table 3 – Vector lengths of the expanded words for entailing and non-entailing word pairs.

From Table 3 we see that expanded words with less than 500 features participate in 20% of the non-entailing word pairs and only in 7% of the entailing ones. In other words, a decision not to rely on expanded words having less than 500 features would filter out a considerable amount of low-quality rules, not too much affecting the correct rules. We also see that expanded words with long feature vectors participate in incorrect rules much more rarely.

We conclude that expanded words with longer vectors are more trustworthy for automatic learning of expansion rules and thus should be promoted at the expense of short-vector expanded words.

# 4 DISTRIBUTIONAL INCLUSION MEASURES

In this section, summarizing the factors investigated in Section 3, we formulate the desired properties of an inclusion measure (Section 4.1). Then we will analyze the behavior of the two existing inclusion measures, *WeedsPrecision* and *balPrecision*, with respect to these properties and propose our novel inclusion measures. Since *balPrecision* is a product of *WeedsPrecision* and *LIN* similarity, we first analyze the *WeedsPrecision* measure (Section 4.2); then, taking into consideration the results of the analysis, we propose our novel inclusion measures (Section 4.3) and then discuss the balancing mechanism of the *balPrecision* measure and its application to the newly-defined inclusion measures (Section 4.4).

## 4.1 Desired Properties of an Inclusion Measure

In view of the factors analyzed above in Section 3, we can formulate the desired properties of an inclusion measure, which should quantify our confidence that distributional inclusion (and, consequently, the *expansion* relation) indeed holds for two given words. The scores, assigned by the inclusion measure, should:

*Property 1 (P1):* reflect the proportion (or amount) of included features;

*Property 2 (P2):* reflect the relevance of included features to the expanding word;

*Property 3 (P3):* reflect the relevance of included features to the expanded word.

In addition, the scores should correspond to the following observations:

*Property 4 (P4):* expanding words with short vectors (small amounts of tested features) are less reliable;

*Property 5 (P5):* expanded words with short feature vectors are less reliable.

That is, we want to promote word pairs having many included features that are highly relevant for both words; in addition we trust inclusion less if the number of tested features is small or in case that the expanded word has a short vector.

## 4.2 Analysis of the WeedsPrecision Measure

As stated in Section 2.3.2, *WeedsPrecision* measure is virtually a distributional inclusion measure.

$$WeedsPrecision(w \to v) = \frac{\sum_{f \in FV(w) \cap FV(v)} weight(f, w)}{\sum_{f \in FV(w)} weight(f, w)},$$

where *FV(word)* is the set of features in the vector of the corresponding word.

Indeed, we can see that *WeedsPrecision* measures the sum of weights of all the included features and normalizes it with the weighted sum of all the features in the vector of the expanding word.

We note that the *WeedsPrecision* measure was proposed two years before the *Distributional Inclusion Hypotheses* were formulated, and was originally applied to the full feature vectors of expanding words (with no selection of top characteristic features for inclusion testing, as proposed by the hypotheses). Our evaluation revealed that the *WeedsPrecision* measure performs similarly with simply measuring the proportion of included features.

Below we analyze this measure in the view of its compliance with the desired properties of an inclusion measure formulated above in Section 4.1.

***P1:*** *The score should reflect the proportion (or amount) of included features*. The *WeedsPrecision* measure seems to comply with this requirement – the more features of *w* will co-occur with *v*, the higher score the rule *w*→*v* will get. But on closer examination the behavior of the measure is not that simple:

- Because the measure is based on weight summation, inclusion of many low-weighed features may be equivalent to inclusion of few high-weighed features (this behavior is discussed below in connection with the rest of the properties).
- Due to the heterogeneity of Mutual Information weights, two different pairs of words having the same quantity of included features (even placed at the same

ranks) are likely to receive different scores. This might be a disadvantage when comparing reliability of several different expansion rules.

***P2:*** *The score should reflect the relevance of included features to the expanding word*. At first sight, this requirement is fulfilled by the *WeedsPrecision* measure – impact of each included feature is equal to its weight in the vector of the expanding word and thus reflects its relevance. However, there are several observations we would like to note:

- The sum of weights of a host of included features from the tail of the vector may turn out higher than the weigh of an included feature placed at a high rank.
- According to Figure 2 (Section 3.3), the majority of included features are placed at low ranks, so features of low relevance are very likely to have higher joint impact than several highly relevant included features.
- Figure 2 also shows that non-entailing pairs tend to have more lower-ranked included features as compared to entailing pairs, and thus giving high joint impact to features of low relevance will most probably decrease the distinguishing ability of the *WeedsPrecision* measure.

Taking all this into consideration, we presume that the *WeedsPrecision* measure will perform better for short sets of tested features, while applying it to longer feature sets will most likely decrease the results.

***P3:*** *The score should reflect the relevance of included features to the expanded word*. This requirement is not addressed by the *WeedsPrecision* measure, i.e. impact of included features does not depend on their different relevance to the expanded word.

***P4:*** *Expanding words with short vectors (small quantity of tested features) are less reliable*. The *WeedsPrecision* measure does not comply with this property due to the following reasons:

- We can expect that there will be smaller amounts of included features in short sets of tested features sand thus the sum of the weights in the nominator is likely to be lower for short-vector words; but since the measure is normalized by the weighted sum of all the tested features, the denominator will also be low, thus producing high-score ratio.
- Moreover, expanding words with very short vectors will most probably yield scores tending to the maximum score of 1, since it's very likely that the majority of their few tested features or even all of them will co-occur with the expanded word.

***P5:*** *Expanded words with short feature vectors are less reliable*. This requirement is most probably fulfilled by the measure, since in short vectors we are likely to find smaller amounts of tested features than in long ones. Still, it cannot be contended for certain.

Thus we detected three main disadvantages of the *WeedsPrecision* measure:

(i) Being based on the feature weights, it inherits their heterogeneity, thereby complicating the comparison of reliability scores, obtained for different word pairs.

(ii) Summarizing feature weights with no emphasis on their ranking most probably gives rise to undesirable noise, introduced by included features of low relevance;

(iii) Measuring the ratio between weighted sums of features apparently promotes unreliable similarity rules with expanding word having a short vector.

### 4.3 Using Average Precision as an Inclusion Measure

As concluded in Section 4.1, we would like an inclusion measure to promote word pairs having many included features that are highly relevant for both words; in addition we should trust inclusion less if the number of tested features is small or in case that expanded word has a short vector. Furthermore, taking into consideration the disadvantages of the *WeedsPrecision* measure detected in Section 4.2, we would prefer our inclusion measure to estimate relevance of features using their ranks instead of weights.

We suggest looking at this task in terms of another NLP challenge – Information Retrieval (IR) evaluation – with further adaptation of the state-of-the art IR evaluation method to our problem. In IR evaluation the task is to compare different retrieval systems: given a query, each system returns a list of documents, ranked according to their relevance to the query. A good system should:

- bring many relevant documents(i.e. have high recall);
- bring no or not many irrelevant documents (i.e. have high precision);
- concentrate the relevant documents at the top of the ranked list.

The common method for comparing IR systems involves calculating each system's *Average Precision (AP)* – a measure that combines precision, relevance ranking and overall recall (Voorhees and Harmann, 1999), and is defined as follows:

$$AP = \frac{\sum_{r=1}^{N}[P(r) \cdot rel(r)]}{number\ of\ relevant\ documents};$$

where *r* is the rank of a retrieved document, *rel(r)* is a binary function on the relevance of the document at a given rank *r*, and *P(r)* is precision at the given cut-off rank *r*.

In our settings, the feature vector of the expanded word will virtually represent the list of all the relevant documents, and the tested features of the expanding word will stand for the retrieved documents. Included features will thus represent retrieved documents that turned out relevant.

$$AP(w \to v) = \frac{\sum_{r=1}^{N}[P(r) \cdot rel(f_r)]}{|FV|}; \quad rel(f_r) = \begin{cases} 0, & if \quad f_r \notin FV \\ 1, & if \quad f_r \in FV \end{cases};$$

where *r* is the current rank inside the tested features list of the word *w*, $f_r$ is the feature placed at rank *r* inside the tested features list, *N* is the number of tested features, *FV* is the feature vector of the expanded word *v*, *rel(*$f_r$*)* is a function on the relevance of the feature $f_r$ to the expanded word, *rank(*$f_r$*,FV)* is the rank of the feature $f_r$ inside the vector *FV* and *P(r)* is precision at the given cut-off rank *r* defined as follows:

$$P(r) = \frac{|\, included \;\; features \;\; at \;\; ranks \;\; 1 \;\; to \;\; r\,|}{r}.$$

Such adaptation indeed turns the *Average Precision* measure into an inclusion reliability measure. Furthermore, evolving to *Average Precision* from the existing *Precision*-based measure appears to be a sensible step, since the *AP* measure corrects one of the main disadvantages we detected for the *WeedsPrecision* – deficient consideration of feature ranking.

The *Average Precision* measure emphasizes returning more relevant documents at higher ranks, which in terms of distributional inclusion means observing more included features placed higher in the list of the tested features, and thus complies with two of the desired properties (*P1* and *P2*) listed above in Section 4.1. In order to address the other properties, we introduce two refinements of the classical *AP* measure:

**I.** In the classical *AP* formula, *rel(r)* is a binary function returning 1 for each of the retrieved documents that was judged as relevant, and 0 for retrieved documents that turned out not relevant. In other words, all the relevant documents are considered relevant to the same extent. In our case relevant documents represent features in the vector of the expanded word and have different relevance, which according to our analysis should be reflected in the inclusion reliability score (property *P3*). We suggest reformulating *rel(r)* in the following way:

$$rel(f) = \begin{cases} 0, & if \quad f \notin FV \\ 1 - \dfrac{rank(f, FV)}{|FV| + 1}, & if \quad f \in FV \end{cases};$$

where *f* is a tested feature, *FV* is the feature vector of the expanded word, *rank(f, FV)* is the rank of the feature *f* in the feature vector *FV*.

Such reformulation of *rel(f)* gives us a real number in the range (0,1) for relevance estimation and also leads to some demotion of expanded words with short feature vectors, thus satisfying one more requirement we defined for an inclusion measure (property *P5*).

**II.** The classical AP measure is normalized using the number of relevant documents in order not to demote queries with small number of relevant documents. In our settings such normalization means giving preference to expanded words with shorter vectors, which is not desirable according to our analysis (property *P5*). Therefore we suggest using the number of tested features for normalization: the refined measure in this case will conform to the distributional inclusion hypotheses more closely, since it will virtually measure the ratio of the features shared by the two words and the features tested for the expanding word.

Applying the two described refinements together and separately, in addition to using the classical *AP* measure itself, gives us four different measures of inclusion reliability:

1) The classical *AP* measure:

$$AP(w \to v) = \frac{\sum_{r=1}^{N} [P(r) \cdot rel(f_r)]}{|FV|}; \qquad rel(f_r) = \begin{cases} 0, & if \quad f_r \notin FV \\ 1, & if \quad f_r \in FV \end{cases}$$

2) The classical *AP* measure with refinement I applied:

$$AP^{I}(w \to v) = \frac{\sum_{r=1}^{N}[P(r) \cdot rel(f_r)]}{|FV|}; \quad rel(f_r) = \begin{cases} 0, & if \quad f_r \notin FV \\ 1 - \dfrac{rank(f_r, FV)}{|FV|+1}, & if \quad f_r \in FV \end{cases}$$

3) The classical *AP* measure with refinement II applied:

$$AP^{II}(w \to v) = \frac{\sum_{r=1}^{N}[P(r) \cdot rel(f_r)]}{N}; \quad rel(f_r) = \begin{cases} 0, & if \quad f_r \notin FV \\ 1, & if \quad f_r \in FV \end{cases}$$

4) The classical *AP* measure with both refinements I and II applied:

$$AP^{III}(w \to v) = \frac{\sum_{r=1}^{N}[P(r) \cdot rel(f_r)]}{N}; \quad rel(f_r) = \begin{cases} 0, & if \quad f_r \notin FV \\ 1 - \dfrac{rank(f_r, FV)}{|FV|+1}, & if \quad f_r \in FV \end{cases}$$

where *r* is the current rank inside the tested features list of the word *w*, *P(r)* is precision at the given cut-off rank *r*, $f_r$ is the feature placed at rank *r* inside the tested features list, *N* is the number of tested features, *FV* is the feature vector of the expanded word *v*, $rel(f_r)$ is a function on the relevance of the feature $f_r$ to the expanded word, $rank(f_r,FV)$ is the rank of the feature $f_r$ inside the vector *FV*.

We've shown that the defined measures comply, completely or partially, with the desired properties presented at the beginning of this chapter, except for demotion of expanding words with low amounts of tested features (property *P4*). We note that the *AP* and $AP^{I}$ measures do comply with this property to a certain extent, since the ratio they measure will not tend to be 1 for entailing words with short vectors, but quite the contrary. As for the $AP^{II}$ and $AP^{III}$ measures, we assume that the fulfillment of this requirement can be achieved by means of the *balancing* mechanism described below in Section 4.4.

### 4.4 The Balancing Mechanism

If the observation made by Szpektor and Dagan (2008) for unary templates is correct also in case of lexical expansion, the *WeedsPrecision* measure is supposed to give high scores to bizarre rules with infrequent expanding term. Such behavior indeed complies with our analysis:

- Infrequent words have short feature vectors. As argued above in Section 4.1, expanding words with short vectors most probably yield high *WeedsPrecision* scores.
- In addition to this theoretical reasoning, we manually checked similarity lists produced by the *WeedsPrecision* measure and the results of this examination confirmed these assumptions.

Szpektor and Dagan (2008) proposed to balance such undesirable behavior by means of the *LIN* measure[1], which tends to prefer rules in which elements are related, but do not necessarily participate in an entailment or equivalence relation. This solution was in fact consistent with the experiment of Geffet and Dagan (2005), which concluded that their inclusion testing algorithm (ITA) may be used for predicting lexical entailment, filtering out non-entailing word pairs produced by the *LIN* algorithm or another symmetric distributional similarity model. We decided to adopt this solution, which penalizes infrequent expanding terms and virtually allows bringing an inclusion measure into accord with property *P4* defined in Section 4.1: e*xpanding words with short vectors are less reliable*. As a result, we obtained 4 balanced measures *balAP*, $balAP^{I}$, $balAP^{II}$ and $balAP^{III}$ of the type $balAP^{(i)}(w \rightarrow v) = sim_{LIN}(w,v) \cdot AP^{(i)}(w \rightarrow v)$.

Having analyzed similarity lists generated by the *LIN* algorithm, we came to a conclusion that rules at the tails of the lists are absolutely unreliable, since the elements of these rules are lexically related extremely rarely. We propose to null the scores of

[1] They report significant improvement in comparison both with the *LIN* and the *WeedsPrecision* methods. We note that they applied their balanced measure to the full vector of expanding element and did not examine its application to some top characteristic features, as suggested by the *Distributional Inclusion Hypotheses*.

these low-grade rules[1], which results in improvement of the *MAP*[2] of up to 5% for different balanced inclusion measures.

### 4.5 Data examples

In Section 5 we perform evaluation and comparison of the performance of inclusion measures within two different NLP tasks. In current section, anticipating the quantitative evaluation, we present examples of expansion rules randomly selected from top-10 rules in similarity lists generated for our evaluation.

Table 4 presents the examples for the state-of-the-art symmetric *LIN* measure and the inclusion-based directional *balAP$^{III}$* measure, which involves all the extensions proposed for the classical AP measure.

| | *LIN* | *balAP$^{III}$* |
|---|---|---|
| Nouns | pornography ↔ writing<br>marriage ↔ birth<br>verdict ↔ indictment<br>living ↔ traveling<br>cellular telephone ↔ phone<br>bosnians ↔ elector | larry ellison[3] → founder<br>forum → meeting<br>bombardment → attack<br>novelist → writer<br>birthweight → birth<br>stoppage → demonstration |
| Verbs | live ↔ die<br>accuse ↔ meet<br>shoot ↔ fire<br>absorb ↔ borrow<br>kill ↔ arrest<br>meet with ↔ sue | detain → arrest<br>depose → elect<br>ship → transport<br>find out → hear<br>employ → hire<br>hospitalize → injure |

Table 4 – Examples of expansion rules generated by the LIN and the *balAP$^{III}$* measures.

In order to better illustrate the application of distributional inclusion and give the flavor of the nature of used syntactic features, in Table 5 we show an example of features that turned out included for expansion rule *bombardment → attack*.

[1] In our experiments we truncate the similarity lists of *LIN* after top-1000 rules, since after these rules we detected no adequate rules at all. As an example, the widely used online database of *LIN* similarities (available at http://www.cs.ualberta.ca/~lindek/downloads.htm) for each word lists up to 200 most similar words.

[2] For detailed description of experimental settings see Section 5.2.

[3] Larry Ellison is the co-founder and CEO of Oracle Corporation.

| *Included features* | | |
|---|---|---|
| <conj>artillery:n | <conj>separatist:n | <obj<witness:v |
| <obj<stop:v | <obj<apologise:v | <obj<launch:v |
| >mod>intensive:a | <conj>rocket:n | >nn>mortar:n |
| <conj>raid:n | <obj<suffer:v | >vrel>aim at:v |
| <subj<destroy:v | <subj<hit:v | <conj>fighting:n |
| >mod>massive:a | <obj<cease:v | >nn>army:n |
| >nn>wartime:n | <obj<survive:v | <subj<kill:v |

Table 5 – Examples of included features for expansion rule *bombardment→attack*.

### 4.6 Summary

In this chapter, we've formulated the desired properties of a good distributional inclusion measure, based on the investigation of components and parameters of the distributional inclusion scheme performed in Section 3.

In view of these properties, we've analyzed the existing, *Precision*-based, inclusion measures and defined our novel inclusion measures based on the *Average Precision* metric.

Further we present the evaluation of the inclusion measures and the analysis of the results. We note that two unbalanced measures – *AP* and $AP^{I}$ – and all the balanced measures comply to some extent with all the defined properties, and therefore we expect them to have higher performance. Unlike them, the rest of the unbalanced measures – *WeedsPrecision*, $AP^{II}$ and $AP^{III}$ – do not comply with all of the requirements, and consequently we expect them to yield worse results.

# 5 EVALUATION

## 5.1 Evaluation methodology and measures

As argued by Szpektor et al. (2007), an evaluation methodology for expansion rules should reflect the expected validity of their application within NLP systems. Following that line, a rule *'L → R'* should be "regarded as correct if in all (or at least most) relevant contexts in which the instantiated template *L* is inferred from the given text, the instantiated template *R* is also inferred from the text[1]". They ground the appropriateness of *instance-based* evaluation of expansion rules, when instead of directly evaluating rules in isolation the judges need to assess the rule's validity under the given context in each example. Following this rational, we performed automatic evaluation of expansion rules obtained from different similarity measures. We applied lexical expansion for two different NLP tasks – Automatic Content Extraction (ACE) and Unsupervised Keyword-based Text Categorization (TC) – and evaluated the results within the *instance-based* framework.

For that we used common measures from Information Retrieval and Statistical Classification. Two basic measures used for evaluation are *Precision* and *Recall*: *Precision* can be seen as a measure of exactness, whereas *Recall* is a measure of completeness.

$$Precision = \frac{|true - positives|}{|true - positives| + |false - positives|};$$

$$Recall = \frac{|true - positives|}{|true - positives| + |false - negatives|}.$$

Since often there is an inverse tradeoff relationship between *Precision* and *Recall*, where it is possible to increase one at the cost of reducing the other, usually

[1] This reasoning corresponds to the common definition of entailment in semantics, which specifies that a text *L* entails another text *R* if *R* is true in every circumstance (possible world) in which *L* is true (Chierchia and McConnell-Ginet, 2000).

*Precision* and *Recall* scores are not discussed in isolation. Instead both are combined into a single measure, such as the $F_1$*-measure*, which is the harmonic mean of *Precision* and *Recall*, evenly weighing both components: $F_1 = \frac{2 \cdot Precision \cdot Recall}{Precision + Recall}$.

In the case where the evaluated examples constitute a ranked list, it is customary to use the *Average Precision (AP)* measure, described in Section 4.3, which is the average of the precision values computed when truncating the ranked list after each positive example. In case of several lists to evaluate, *Mean Average Precision (MAP)*, which is the mean of the individual *AP* scores, is used.

Another common technique for evaluating ranked lists of examples, which was also used in our evaluation, is measuring *Precision* and *Recall* at different (fixed) top-*k* cut-offs.

For vector creation we used the Reuters RCV1 corpus, described in Section 3.1.

### 5.2 Evaluation within Automatic Content Extraction task

In order to perform application-oriented evaluation of our lexical expansion extraction methods, we decided to use the Automatic Content Extraction task (*ACE 2005 training corpus2*). The ACE corpus contains 15,724 sentences, and the annotation includes 33 types of events such as *Attack*, *Divorce*, and *Start-Position*. All the mentions of these events are annotated in a corpus collected from a variety of sources (newswire articles, transcription of phone calls, blogs, etc.), where the extent of an event mention is assumed to be a single sentence. For each event mention, ACE annotation includes a trigger, defined as “the word that most clearly expresses its occurrence”. The fact that each event mention can be related to an indicative trigger word justifies the lexical approach to event mention detection, which we use for our evaluation.

For each event we manually defined a set of representative *seed words* or *seeds*. The seed words were extracted from the ACE event definitions in the annotation guidelines; most of them are the annotated triggers in the example sentences given in the event definition. Some definitions also specify additional words as examples for the event, which we also added to the seed set. For example, given the guideline *"MEET: Events include talks, summits, conferences, meetings, visits"*, we added the nouns *talk, summit, conference, meeting* and *visit*. We extracted nouns, verbs and adjectives as expanded words. On average, we had 4 seed words per event.

Then we constructed a list of expansion rules *w*→*seed* using each of the similarity measures, which we aim to evaluate. We judge sentences that include *w* as containing a mention of the corresponding event, and then compare the obtained annotation to the gold-standard one. For example, for the *MEET* event we collected the word *summit* into the seed set; given a rule *Davos economic forum* → *summit*, we consider all the sentences containing *Davos economic forum* as sentences that mention the *MEET* event, and further check which of them indeed mention the *MEET* event according to the ACE annotation.

Since we are interested to evaluate the contribution of our expansion extraction methods, we decided to consider sentences that contain the seeds of a given event as trivial mentions of that event and therefore did not use these sentences for the evaluation of this event. As a result, we excluded 8 events having less than 10 non-trivial mentions in the corpus[1]. The number of sentences with event mentions ranges from 1181 (for *Attack*) to 12 (for *Sue*) with an average of 172.

According to the lexical reference criterion presented in Section 2.1.1, pairs like *divorce*→*marry* are supposed to be judged as useful for lexical expansion, because a mention of *divorce* in some text allows us to conclude that formerly an event *marry* occurred. We adapted the ACE annotation to comply with this principle. For example, all

[1] The 22 remaining events and their seed sets are listed in Appendix A.

the sentences annotated in ACE as mentioning the *Divorce* event were annotated as containing a mention of the *Marry* event as well[1].

Within the given settings, event detection can be viewed as a ranking task: using the list of expanding words *w*, rank all the sentences in the corpus according to the likelihood they contain an event mention. A perfect ranking would place all the sentences that indeed contain event mentions before the rest of the sentences. Different rankings can be compared using the *AP* and *MAP* measures described above. Note that expanding words in our lists are provided with scores reflecting their similarity to the corresponding expanded word, and so we can rank the corpus sentences according to these scores. A sentence containing an expanding term[2] *w* will be assigned *w*'s score; if a sentence contains several expanding terms $w_i$, its score will be the sum of $w_i$'s scores. Such scoring allows us to evaluate not only the quality of the expansion rules obtained, but also to evaluate their relative ranking inside the list.

Being a combination of *Precision*, *Recall* and relevance ranking, *Average Precision* and, consequently, the *MAP* measure do not give an estimate of the actual *Precision* and *Recall* of the compared systems. In order to obtain such an estimate, we measured *Precision* and *Recall* for using different top-*k* sentences graded by the evaluated measures.

### 5.2.1 Evaluation Results

Table 6 presents the results of using nouns for lexical expansion. The table shows the *MAP* values of all the events, for which noun-based expansion resulted in the *AP* value of no less than 0.1 for at least one of the inclusion measures. The 10 out of 22 events are: *Attack*, *Convict*, *Demonstrate*, *Die*, *Injure*, *Marry*, *Meet*, *Sentence*, *Sue, and*

[1] The pairs of events that we used for this additional annotation are*: Divorce→Marry, Execute→Sentence, Sentence→Convict, End-Position→Start-Position, Appeal→Trial-Hearing, Convict→Trial-Hearing, Execute→Trial-Hearing, Sentence→Trial-Hearing, Declare-Bankruptcy→Start-Org, End-Org→Start-Org, Merge-Org→Start-Org* and *Release-Parole→Arrest-Jail.*

[2] We used all words and word compounds from unigrams to 6-grams.

*Trial-Hearing.* In order to evaluate the use of different percentages of tested features for the symmetric measure of *LIN*, we calculated the *LIN* similarity using the corresponding percentage of features for both left-hand sides and right-hand sides of the similarity rules.

| | Top-10% | Top-25% | Top-50% | Top-75% | Top-100% | Standard deviation |
|---|---|---|---|---|---|---|
| *LIN* | 0.063 | 0.058 | 0.047 | 0.048 | **0.094** | *0.017* |
| *WeedsPrecision* | **0.129** | 0.049 | 0.030 | 0.030 | 0.029 | *0.039* |
| *AP* | 0.176 | **0.264** | 0.260 | 0.180 | 0.114 | *0.057* |
| *$AP^I$* | 0.184 | **0.272** | 0.270 | 0.230 | 0.157 | *0.046* |
| *$AP^{II}$* | **0.105** | 0.037 | 0.014 | 0.015 | 0.016 | *0.035* |
| *$AP^{III}$* | **0.110** | 0.053 | 0.041 | 0.040 | 0.038 | *0.027* |
| *balPrecision* | **0.300** | 0.299 | 0.270 | 0.258 | 0.246 | *0.022* |
| *balAP* | 0.226 | **0.271** | 0.262 | 0.256 | 0.232 | *0.017* |
| *$balAP^I$* | 0.236 | **0.272** | 0.268 | 0.264 | 0.254 | *0.013* |
| *$balAP^{II}$* | 0.265 | **0.299** | 0.281 | 0.273 | 0.268 | *0.012* |
| *$balAP^{III}$* | 0.271 | **0.304** | 0.287 | 0.284 | 0.284 | *0.011* |

Table 6 – Values of *MAP* for the ACE task, using nouns for expansion. The table shows *MAP* for different percentages of tested features. The highest *MAP* value of each measure is bolded. The last column shows the standard deviation of the MAP values from the average for each inclusion measure.

#### 5.2.1.1 Analysis of the desired properties of inclusion measures

The data presented in Table 6 allows us to evaluate the impact of the desired properties of inclusion measures that were formulated in Section 4.1, as well as the validity of the modifications we made to the classical *Average Precision* formula to make it correspond to these properties. Below we discuss these properties and modifications.

**The desired properties of an inclusion measure**. From Table 6 we see that two unbalanced measures – *AP* and *$AP^I$* – and all the balanced measures, which comply to some extent with all the defined properties, substantially improve the performance of the state-of-the-art measure of *LIN*. The other measures, which do not comply with all the requirements, do not manage to improve the state-of-the-art, but quite the contrary. This conforms to our expectations stated in Section 4.5 and allows us to assess the validity of the defined properties.

We now proceed to analyze the defined properties together with the mechanisms involved in their realization.

**The balancing mechanism**. From Table 6 we see that balanced measures consistently perform better than the unbalanced ones. We note that:

- *The WeedsPrecision*, $AP^{II}$ and $AP^{III}$ measures, normalized by the number of the tested features, yield high scores for expanding words with short vectors and do not comply with property *P4: expanding words with short vectors are less reliable*. These measures were balanced by means of the symmetric *LIN* measure, demoting such expanding words. We see that the balanced versions of these measures, *balPrecision*, $balAP^{II}$ and $balAP^{III}$, perform notably better than their unbalanced prototypes.
- In contrast to them, the *AP* and $AP^{I}$ measures (which are normalized by the vector length of the expanded word and thus conform to the aforesaid property), although performing somewhat worse than their balanced version, still yield competitive results even without balancing.

We conclude thus that compliance with property *P4* indeed improves the inclusion measures. We also conclude that balancing indeed brings the inclusion measures in accord with this property, successfully neutralizing the impact of unreliable entailing words.

**Reflecting different relevance of included features to the expanded word**. From Table 6 we see that both for balanced and unbalanced versions:

- $AP^{I}$ always performs better than *AP;*
- $AP^{III}$ always performs better than $AP^{II}$.

The $AP^{I}$ measure is a variant of *AP*, and the $AP^{III}$ measure is a variant of $AP^{II}$, while extending the relevance function $rel(f_r)$*.* The *AP* and $AP^{II}$ measures use a binary $rel(f_r)$

function, while in $AP^{I}$ and $AP^{III}$ this function estimates the relevance of the included features to the expanded word.

We see that reflecting different relevance of included features to the expanded word (property *P3*) indeed improves the inclusion measures. We also conclude that the extended relevance function, which we suggested in Section 4.3, indeed brings the inclusion measures in accord with this property.

**Alternative normalization using the number of tested features**. From Table 6 we see that the $balAP^{II}$ and $balAP^{III}$ measures always perform better than *balAP* and $balAP^{I}$. The difference between the pairs of measures lies in the normalization mechanism: the *balAP* and $balAP^{I}$ measures are normalized by the length of the expanded word's vector, while $balAP^{II}$ and $balAP^{III}$ are normalized using the number of tested features. We note that using the number of tested features for normalization was proposed to make the classical *AP* measure correspond with property *P5: expanded words with short vectors are less reliable*. This refinement also brings the classical *AP* measure to a more precise reflection of the notion of distributional inclusion, where inclusion is relative to the expanding word's features, and we conclude that it indeed leads to a considerable improvement in the results.

We've thus shown that all the suggested modifications improve the inclusion reliability measures.

#### 5.2.1.2 Performance for noun-based expansion

Below we analyze, according to Table 6, the performance of the $balAP^{III}$ measure, which involves all the proposed modifications:

1. With optimal selection of the parameter of the percentage of tested features (comparing the best values of *MAP* for each measure), we see that $balAP^{III}$ is the best-performing measure, having two competitors – the *balPrecision* and $balAP^{II}$ measures with almost the same results.

2. When the percentage of tested features is not optimal(comparing the worst values of *MAP* for each measure), we see that *balAP*$^{III}$ performs best with *MAP*=0.271, while *balPrecision* has substantially lower *MAP*=0.246.
3. From Table 6 we see that *balAP*$^{III}$ is less sensitive to poor selection of the percentage of tested features than the *balPrecision* measure. This fact is also confirmed by the *standard deviation* values. The lowest *standard deviation* is observed for the *balAP*$^{III}$ measure, while *balPrecision* has twice that value.

We conclude that the *balAP*$^{III}$ measure is preferable due to its high performance and lower sensitivity to the value of the parameter of the percentage of tested features.

#### 5.2.1.3 Verb-based expansion and expansion using both nouns and verbs

All the tendencies listed above for noun-based lexical expansion remained the same in experiments done for verbs and for joint use of verbs and nouns. In further investigation we concentrate on applying inclusion measures to the full vectors of expanding terms, since:

- avoiding the use of a parameter (percentage of tested features), which requires tuning, may be preferable;
- evaluating the measures in such settings is virtually a simulation of real-life conditions, when the parameter may not be optimally tuned.

Table 7 shows the values of *MAP* for noun-based expansion, verb-based expansion and expansion using both verbs and nouns[1]. We see that:

- the *balAP*$^{III}$ measure performs best both for the verb-based expansion and for the joint use of nouns and verbs;
- the *balAP*$^{III}$ measure benefits most from supplementing noun-based expansion with additional verb-based expansion rules, notably improving its results.

[1] We report the result for the same 10 events that were used for noun-based expansion.

| | Nouns | Verbs | Nouns & verbs |
|---|---|---|---|
| *LIN* | **0.094** | 0.033 | 0.068 |
| *WeedsPrecision* | 0.029 | 0.025 | **0.044** |
| *AP* | **0.114** | 0.068 | 0.089 |
| *$AP^{I}$* | **0.157** | 0.078 | 0.126 |
| *$AP^{II}$* | 0.016 | 0.026 | **0.028** |
| *$AP^{III}$* | 0.038 | 0.041 | **0.056** |
| *balPrecision* | **0.246** | 0.095 | 0.237 |
| *balAP* | **0.232** | 0.100 | 0.202 |
| *$balAP^{I}$* | 0.254 | 0.102 | **0.258** |
| *$balAP^{II}$* | 0.268 | 0.098 | **0.269** |
| *$balAP^{III}$* | 0.284 | 0.104 | **0.312** |

Table 7 – Values of *MAP* for the ACE task, using nouns, verbs and both verbs and nouns for expansion. The table shows the results for using full feature vectors of expanding words (top-100% of tested features). The column "Nouns" repeats the results shown in Table 4 for convenience of comparing the performance. The highest *MAP* value of each inclusion measure is bolded.

From Table 7 we see that verb-based expansion performs worse than noun-based expansion, still yielding noticeable improvement over the *LIN* baseline for *AP*, *$AP^{I}$* and all the balanced measures. Our analysis showed that one of the reasons of the deteriorated performance of the verb-based expansion is that there were many ACE events having verb seeds (i.e. expanded words) with short vectors, while for the noun-based expansion seeds with long vectors were more common. As argued in Section 3.5, expanded words with short feature vectors yield noisy similarity lists. In other words, for the noun-based expansion there were many events with reliable expanded words, while for the verb-based expansion there were mostly unreliable words to expand.

This is also one of the reasons for the behavior of inclusion measures in joint use of nouns and verbs, shown in Table 7: additional verb-based expansion rules lowered the performance of the *balAP*, *AP* and *$AP^{I}$* measures, which do not demote expanded words with short vectors as prescribed by property *P5*.

#### 5.2.1.4 Statistical significance

To assess the validity of the results we calculated statistical significance, comparing the best-performing *$balAP^{III}$* measure with other evaluated measures according to the Wilcoxon signed-rank test.

This verification confirmed statistical significance of the results achieved by the *balAP*$^{III}$ measure as compared to the common-practice symmetric measure of *LIN* with *p*=0.01.

Among the inclusion measures, the test proved statistical significance with *p*=0.01 for all the measures, except for the *balAP* and the *balAP*$^{I}$ measures, for which we observed *p*=0.05. This allows us to make the following conclusions:

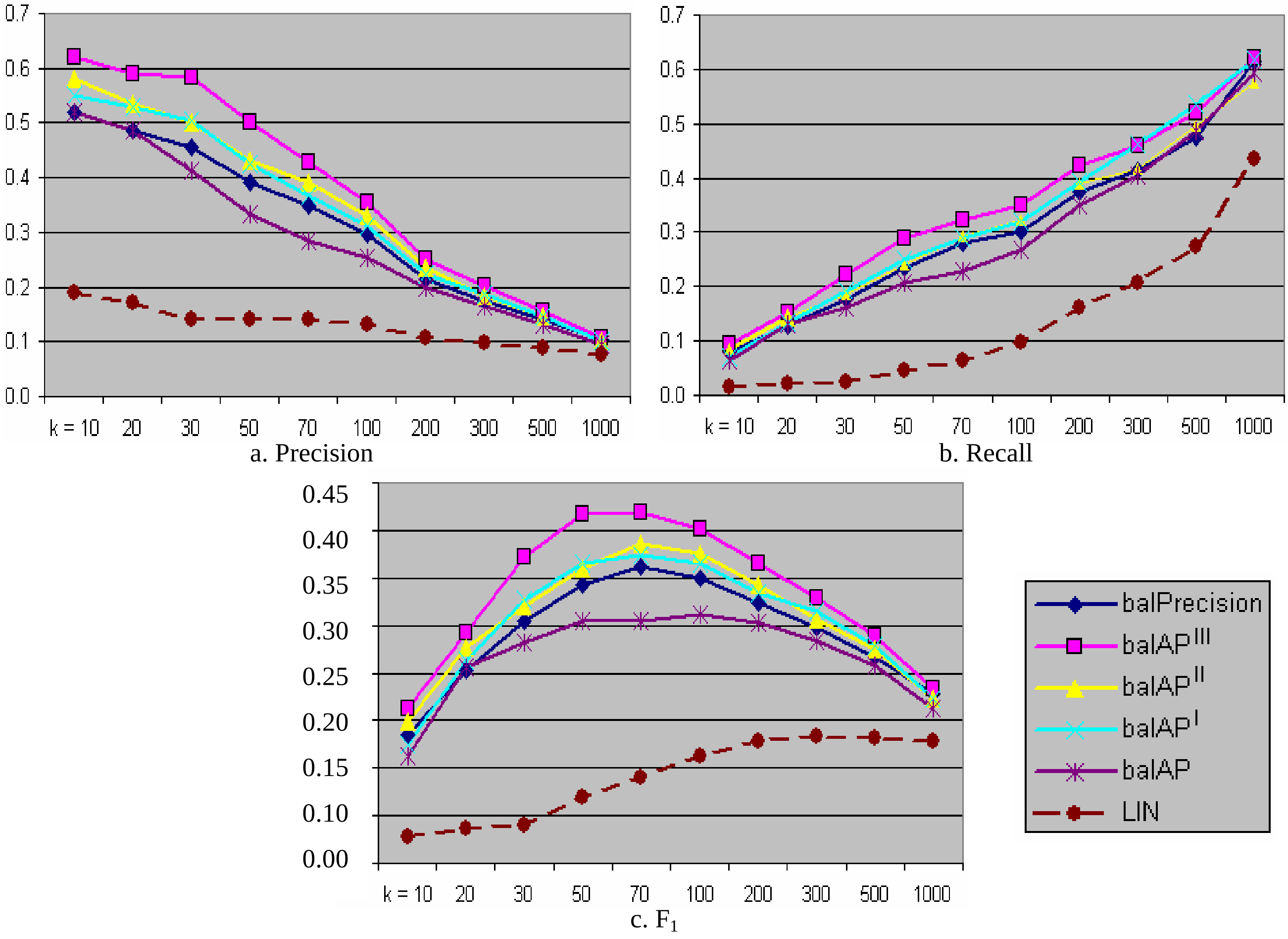
- The results assessing the validity of property *P4: expanding words with short vectors are less reliable*, and of the balancing mechanism involved to satisfy this property proved to be statistically significant with the highest level of confidence (*p*=0.01 shown for the *balAP*$^{III}$ measure when compared to all the unbalanced measures).
- The same level of statistical significance was observed in comparison with the *balAP*$^{II}$ measure, which differs from the *balAP*$^{III}$ measure by its relevance function. The the *balAP*$^{II}$ measure uses a binary $rel(f_r)$ function, while in the *balAP*$^{III}$ measure this function estimates the relevance of the included features to the expanded word We conclude thus that the results assessing the need of different relevance of included features to the expanded word (property *P3*) are not observed by coincidence.
- Comparison with *balAP* and the *balAP*$^{I}$ measures resulted in *p*=0.05. The *balAP*$^{III}$ measure differs from these measures by its alternative normalization involved to satisfy property *P5: expanded words with short vectors are less reliable*. We conclude thus that we are less confident that the results assessing the validity of this property are not observed as a result of a coincidence, and note that the value of *p*=0.05 still allows considering these results statistically significant.

#### 5.2.1.5 Precision, Recall and F$_1$

From Table 6 and Table 7 we see that the unbalanced measures proved to be noncompetitive for the current task, and thus we concentrate on the balanced measures for further analysis.

In Figure 4 we present *Precision*, *Recall* and $F_1$ of the balanced measures at different top-*k* sentences cut-off points for the ranked list of sentences.

From Figure 4 we see that the best values of *Precision*, *Recall* and $F_1$ are consistently achieved by the *balAP$^{III}$* measure, considerably outperforming the baseline measure of LIN as well as the *balPrecision* measure.



a. Precision

b. Recall

c. F$_1$

Figure 4 – *Precision*, *Recall* and $F_1$ of balanced inclusion measures at different top-*k* sentences cut-off points for the ACE task.

Table 8 shows the results of a 5-fold cross-validation, tuning the top-*k* sentences parameter for each of the balanced measures. In each fold we used 2 events for training (maximizing the value of $F_1$) and 8 events for testing. The pairs of events for the training were taken in the alphabetic order. This data allows us to evaluate the ability of each measure to tune the optimal cut-off point and shows that, again, the *balAP*$^{III}$ has the best ability to tune an appropriate cut-off point.

| | *Precision* | *Recall* | $F_1$ | *Top-k sentences* |
|---|---|---|---|---|
| *LIN* | 0.10 | 0.21 | 0.14 | *274* |
| *balPrecision* | 0.33 | 0.29 | 0.31 | *150* |
| *balAP* | 0.31 | 0.24 | 0.27 | *102* |
| *balAP*$^{I}$ | 0.39 | 0.27 | 0.32 | *80* |
| *balAP*$^{II}$ | 0.37 | 0.27 | 0.31 | *146* |
| *balAP*$^{III}$ | 0.44 | 0.27 | 0.34 | *82* |

Table 8– Tuning top-*k* sentences parameter for the ACE task. The table presents results of a 5-fold cross-validation – for each of the balanced measures it shows *Precision* and *Recall* values averaged over the folds, and the corresponding value of $F_1$. The last column shows the tuned top-*k* sentences value averaged over the folds.

### 5.2.2 Error Analysis

For further analysis we concentrate on the *balAP*$^{III}$ inclusion measure, which proved to be the best-performing measure for the current task. In order to perform error analysis we randomly sampled from the corpus 100 false-positive examples, choosing them from the top-500 sentences graded for each of the 22 ACE events. Thus we formed a test set with about 4-6 examples per event. The investigation of the sampled examples allowed us to identify several reasons of decision errors, whose distribution is presented in Table 9.

| Type of error | Percent |
|---|---|
| Context-related expanding word | 0.25 |
| Inappropriate context of entailing word | 0.25 |
| Ambiguous seed | 0.21 |
| Ambiguous expanding term | 0.16 |
| Strict gold-standard annotation | 0.07 |
| Invalid expanding word | 0.06 |
| Naïve matching mechanism | 0.01 |

Table 9 – Results of error analysis for the ACE task. Error sources are sorted according to the percent of the errors of the corresponding type.

Below we describe each of the detected error reasons.

1. **Seeds ambiguity**. The events of the ACE corpus have many ambiguous seeds. Words, which would serve as correct expansions for an alternative sense of a seed, lead to wrong decisions for the required sense.

| Event | Seed | Alternative sense of the seed | Sentence fragment |
|---|---|---|---|
| *Phone-Write* | n\|writing | the activity of creating pieces of written work, such as stories, poems or articles | …truly historic television and **journalism**… |
| *End-Position* | v\|fire | to cause a weapon to shoot bullets, arrows or missiles | …it was either **shelled** or **bombed**… |
| *Trial-Hearing* | n\|hearing | an official meeting that is held to gather the facts about an event or problem | ...at the Pentagon **briefing** today General Stanley McChrystal said… |
| *Convict* | n\|conviction | a strong opinion or belief;<br>a feeling of certainty about something | …I want to thank each of you for the **faith** you have shown in this country… |

Table 10 – Examples of wrong decisions made for the ACE task due to ambiguous seeds. The words, expanding the alternative sense of the seed, are bolded.

2. **Ambiguity of the expanding word**. When an expanding word has several senses, only one of them corresponds to the meaning of the *seed*. Occurrences of the rest of the senses cause false-positive decisions. Examples of this type of errors are presented below in Table 11.

| Event | Sentence fragment | Applied sense | Expected sense |
|---|---|---|---|
| *Charge-Indict* | …I **suspect** Rob will speak out after that date… | to think or believe something to be true or probable | to think that someone has committed a crime |
| *Meet* | …Everest is the highest **summit** on the planet… | the highest point of a mountain | an important formal meeting between leaders of governments |
| *Trans-port* | …The **communications** conditions once you're on Mars, just to talk to people on Earth… | the various methods of sending information between people and places | ways of moving between one place and another |

Table 11 – Examples of wrong decisions made for the ACE task due to ambiguous expanding words. The expanding words are bolded in the sentence fragments.

3. **Valid entailing words applied in inappropriate contexts.** Many times a word, in definite contexts yielding true-positive decisions, results in incorrect decisions when applied in the same sense in different contexts. We note that it is difficult to judge the rules as definitely valid or invalid, because the *lexical entailment* criterion is too strict – rules, useful for lexical expansion, do not always correspond to this criterion.

Therefore we decided to divide useful expanding words into two different categories – entailing and context-related. The examples of application of valid *entailing* rules in inappropriate contexts are given below in Table 12.

| Event | Seed | Sentence fragment |
|---|---|---|
| *Injure* | n\|injury | …Of all the **wounds** time doesn't heal, the ones that fester deep in the soul… |
| *Marry* | n\|marriage | …Call Carnival **Wedding** Dept. at 1 800 933-4968… |
| *Phone-Write* | n\|phone | …Baghdad is a city where drinking water and working **telephones** are hot commodities… |
| *Elect* | v\|elect | …I thought you **voted for** it… |
| *Convict* | v\|convict | …I think the media made him **guilty**… |
| *Elect* | n\|election | …the United States still doesn't have the nine **votes** needed to win approval of the resolution… |

Table 12 – Examples of wrong decisions made for the ACE task due to application of valid entailing words in inappropriate contexts. The entailing words are bolded in the sentence fragments.

4. **Valid context-related expanding words**. These expanding words are related to the contexts, in which the meaning of the seed is likely to be mentioned, but do not comply with the lexical entailment criterion. Examples of wrong decisions caused by this kind of expanding words are presented below in Table 13.

| Event | Seed | Sentence fragment |
|---|---|---|
| *Demonstrate* | n\|demonstrator | ...the Israeli army had established roadblocks ... to prevent Palestinians **militants** to leave the area... |
| *Injure* | n\|injury | ...they'd replaced his chest tube, given him some **pain** medication and he was doing much better... |
| *Arrest-Jail* | v\|arrest | ...international community must confront Saddam Hussein's refusal to **disarm**... |
| *Transport* | v\|travel | ...involved other folks hitting my **parked** car... |

Table 13 – Examples wrong decisions made for the ACE task due to application of valid context-related words in inappropriate contexts. The applied words are bolded in the sentence fragments.

We assume that such words can be useful for lexical expansion, both allowing to detect new mentions of the expanded word's meaning (i.e. increase recall) and supporting decisions, made with participation of additional expanding words (i.e. improve relevance ranking). Table 14 presents two examples of true-positive decisions, made using this type of words.

| Event | Sentence fragment |
|---|---|
| *Die* | …in Pennsylvania a student fatally **shot**… |
| *Demonstrate* | …Jordanian lawyers staged a **sit-in** at the main court house after being forcibly blocked by **riot police** from **marching** towards the Iraqi **embassy** to show their **solidarity**… |

Table 14 – Examples of positive influence of context-related expanding words on decisions made for the ACE task. The expanding words are bolded in the sentence fragments.

   1. For the fist example given in Table 14 we note that the word "*fatally*" did not influence the decision, made relying on nouns and verbs only. In a system, using all parts of speech for expansion, the word "*shot*" would also be useful, helping to assign higher scores to sentences, in which a person was "*fatally shot*" and not to sentences, in which, for example, a study was "*fatally flawed*"
   2. For the second example, we see that there are three entailing words "*sit-in*", "*riot*" and "*marching*" (the words "*riot*" and "*police*" were applied separately) and three context-related words "*police*", "*embassy*" and "*solidarity*". In this case context-related words allowed assigning to the sentence a higher score, reflecting the level of certainty that it indeed contains the event mention. But, actually, we can imagine a sentence, containing only the listed context-related words – "*police*", "*embassy*" and "*solidarity*" – which should be scored higher than a sentence, for example, containing only one entailing word "*marching*".

We assume that a good scoring measure should yield lower scores for such rules, than for the *entailing* rules, reflecting the level of their reliability for lexical expansion.

5. **Invalid expanding words.** These are expanding words, which are most likely to cause only incorrect decisions. For example *n|abortion→n|marriage (Marry), n|extradition→n|birth (Be-born), v|irritate→v|sue (Sue), v|scream→ v|marry (Marry), n|broadcasting→n|travel (Transport)* etc. For these words we didn't manage to define any connection to the contexts that we would like to retrieve for their seed.

6. **Strict gold-standard annotation.** In several cases we would consider the retrieved sentences to be true-positive, i.e. indeed containing a mention of the corresponding event, while the gold-standard annotation judged them differently. The examples of such sentences are given below in Table 15.

| Event | Sentence fragment |
|---|---|
| *Attack* | …economic sanctions placed on Iraq in 1990 after its **invasion** of Kuwait… |
| *Trial-Hearing* | …the **appeal** judges found Hartzenberg's refusal did not relate to an error of law… |
| *Arrest-Jail* | …withhold the names of people **detained** as part of the 911 investigation… |
| *Injure* | …there are late reports that the **wounded** include the brother and a son of a Kurdish political leader… |

Table 15 – Examples of strictness of the gold-standard annotation of ACE. The expanding words are bolded.

7. **Naïve matching mechanism.** In our evaluation we simply match the lemma and the part-of-speech tag of the expanding words with the words in the sentences. Such naïve matching turned out an additional source of decision errors. The examples of such errors are presented below.

| Event | Sentence fragment |
|---|---|
| *Attack* | …soldiers discovered what they described as a purpose built **shooting** gallery… |
| *Demonstrate* | …the troops will come **marching** in[1]… |

Table 16 – Examples of errors introduced by the naïve matching used for the ACE task. The expanding words are bolded.

In Table 17 we present average scores of the sampled valid and invalid expansion rules. We note that scores assigned by the *balAP*$^{III}$ measure indeed belong to the conventional interval [0, 1], but are actually much lower than 1. We manually checked several pairs of synonyms (like *suit* → *lawsuit*) and obtained average score of *0.012*. This brief analysis shows that the score, obtained by less reliable entailing rules (which caused our sample errors) is much lower than the score of highly reliable rules such as for synonyms. We also see that the scores of the *entailing* rules are considerably higher than the scores of the context-related ones, which is definitely a beneficial effect.

[1] "*march*" – to participate in an event in which a large number of people walk through a public place to express their support for something, or their disagreement with or disapproval of something; "*march in*" (about *troops* etc.) – to send military forces to a particular place in order to attack it.

The scores of the invalid expansion rules are much lower than those of context-related and *entailing* rules. This means that the scores, assigned by the *balAP*$^{III}$ measure, tend to reflect the level of reliability of the expansion rules.

| Type of error | Average score |
|---|---|
| Inappropriate context of entailing word | 0.004 |
| Context-related expanding word | 0.001 |
| Invalid expanding word | 0.0002 |

Table 17 – Average scores of the sampled valid and invalid expansion rules.

### 5.2.3 Conclusions

The results of our evaluation assess the appropriateness of the desired properties defined for an inclusion reliability measure, and confirm the validity of the *Average Precision*-based approach to inclusion testing. The extensions, proposed for the classical *AP* measure in order to make it correspond with the defined properties, proved to be valid.

The *balAP*$^{III}$ measure, which involves all the extensions:

- Showed the best performance for both noun-based and verb-based expansions, as well as for joint use of nouns and verbs. The results achieved by this measure proved to be statistically significant.
- Proved to be least sensitive to the parameter of tested features selection as compared to all the other measures.
- The *MAP* value achieved by this measure is almost 0.25 points higher than the result of the state-of-the-art measure of *LIN* and 0.075 points higher than the *MAP* value achieved by the recently proposed *balPrecision* measure.
- This measure proved to consistently outperform all the other measures both in *Precision* and *Recall*.

Error analysis, performed for the *balAP*$^{III}$ measure, revealed that:

- More than a fourth of the errors are caused by semantically ambiguous expanded and expanding words.
- *Entailing* words, applied in inappropriate contexts and context-related rules yield the same quantity of errors (25% each).
- Only 6% of errors originate from using invalid expansion rules.

## 5.3 Evaluation within Keyword-based Text Categorization task

In order to obtain a broader evaluation, we evaluated our measures using an NLP application of another kind, in addition to the evaluation within the ACE task. To that end we used an available Keyword-based Text Categorization (TC) system[1].

Keyword-based text categorization methods aim at topical categorization of documents based on sets of terms, without requiring a supervised training set of labeled documents (McCallum and Nigam (1999); Ko and Seo (2004); Liu et al. (2004)). Generally speaking, such systems operate in two phases:

- setup phase, in which a set of characteristic terms for the category is assembled, constituting the category’s feature vector;
- classification phase, in which the term-based feature vector of a classified document is compared with the feature vectors of all categories.

For our evaluation we used the Reuters-10 corpus, which is constructed of the 10 most frequent categories of the Reuters-215788 collection[2]. The corpus contains 9296 documents, divided unevenly over the 10 categories, where most of the documents belong to the *Acquisition* and *Earn* categories. The collection's gold-standard is multi-class classified; hence each document is classified to one or more categories. Following the gold-standard classification approach, we will use the multi-class method in our

[1] Developed by Libby Barak.

[2] Available at http://kdd.ics.uci.edu/databases/reuters21578/reuters21578.html

evaluation. During classification, cosine similarity is measured between the feature vector of the classified document and the vectors of all categories. Document features consist of POS-tagged lemmas of single words and bigrams, limited to nouns, verbs, adverbs and adjectives, with term frequency as the feature value.

Taking the lexical expansion perspective, we assume that the characteristic terms in a category's vector should expand the term (or terms) denoting the category name. For each of the 10 categories we manually assigned such initial *seed terms*[1]. Accordingly, we construct the category's feature vector by taking first the category name itself, and then expanding it with all left-hand sides of lexical expansion rules, whose right-hand side is identical to the category *seed*. For example, the category *Trade* is expanded by rules such as *buying* → *trade*.

### 5.3.1 Evaluation Results

In our evaluation we will concentrate on the measures, which comply to some extent with all the desired properties of an inclusion measure that were defined in Section 4.1. These are two unbalanced measures – *AP* and $AP^{I}$ – and all the balanced measures. The rest of the inclusion measures – *WeedsPrecision*, $AP^{II}$ and $AP^{III}$ – do not comply with all of the requirements and, as expected, promote bizarre rules with low-frequency expanding terms. Below we present exemplary *expanding words* returned by these measures.

It turned out that our TC system is unable to evaluate these measures appropriately:

- Lists of expanding words produced by the *WeedsPrecision* and $AP^{II}$ measures are so noisy, that none of the words co-occur with the document vectors. Thus the results, returned by the system, are based on the category seeds only and are definitely meaningless for the evaluation of lexical expansion.

[1] The 10 categories are: *Acquisition* (n|acquisition), *Corn* (n|corn), *Crude* (n|crude), *Earn* (n|earnings), *Grain* (n|grain), *Interest* (n|interest), *Money-fx* (n|money, n|foreign exchange), *Ship* (n|ship), *Trade* (n|trade) and *Wheat* (n|wheat). In parenthesis we show the initial *entailed words* for each category in "part-of-speech| lemma" format.

- The $AP^{III}$ measure performs a bit better – some of its words (usually starting from the second hundred and below) are indeed found inside the documents – but still the produced results do not reflect the quality of its expansion rules.

| Category seed | *WeedsPrecision* | $AP^{II}$ | $AP^{III}$ |
|---|---|---|---|
| n\|corn | n\|zelenskaja (1.0)<br>n\|william baird (1.0)<br>n\|with (1.0)<br>n\|wicor (1.0)<br>n\|whipsaw (1.0) | n\|zelenskaja (1.0)<br>n\|william baird (1.0)<br>n\|with (1.0)<br>n\|wicor (1.0)<br>n\|whipsaw (1.0) | n\|soybean tender – taiwan (0.98)<br>n\|usda cattle (0.98)<br>n\|sale – a (0.97)<br>n\|dextrose (0.96)<br>n\|corn (0.95) |
| n\|money | n\|zakat (1.0)<br>n\| year-to (1.0)<br>n\|wti/cushing (1.0)<br>n\|wieandt (1.0)<br>n\|widows (1.0) | n\|zakat (1.0)<br>n\| year-to (1.0)<br>n\|wti/cushing (1.0)<br>n\|wieandt (1.0)<br>n\|widows (1.0) | n\|elysee (0.99)<br>n\|parekh (0.99)<br>n\|july-oct (0.98)<br>n\|mityukov (0.98)<br>n\|boatman (0.98) |
| n\|ship | n\|wto secretariat (1.0)<br>n\|vessels mannan (1.0)<br>n\|valentin khutorsk (1.0)<br>n\|uni-order (1.0)<br>n\|u.s.new (1.0) | n\|wto secretariat (1.0)<br>n\|vessels mannan (1.0)<br>n\|valentin khutorsk (1.0)<br>n\|uni-order (1.0)<br>n\|u.s.new (1.0) | n\|ship/name (0.86)<br>n\|floriana (0.65)<br>n\|flag nicolas (0.61)<br>n\|aqra (0.60)<br>n\|mv blue sapphire (0.58) |

Table 18 – Examples of *expanding words* generated by *WeedsPrecision*, $AP^{II}$ and $AP^{III}$ measures for the TC task. The similarity scores of the expansion rules are given in parentheses.

For two remaining unbalanced measures, *AP* and $AP^{I}$, we observed similar evaluation deficiency: when using top-75% of the features as tested, for 8 out of 10 categories all the expanding words co-occurred only with documents containing a seed term, thus not allowing adequate evaluation. We report no results for this setting.

Table 19 shows the results of lexical expansion for the remaining measures, using different percentages of tested features for inclusion testing. In order to obtain these results, we performed a 5-fold cross-validation, tuning the number of similarity rules, which should be used for the expansion. In each fold we used 2 categories for training (maximizing the value of $F_1$) and 8 categories for testing. The pairs of categories for the training were taken in the alphabetic order: (*Acquisition*, *Corn*), (*Crude*, *Earn*), (*Grain*, *Interest*), (*Money-fx*, *Ship*), (*Trade*, *Wheat*).

To evaluate the use of different percentages of tested features for the symmetric measure of *LIN*, we calculated the *LIN* similarity using the corresponding percentage of features for both left-hand sides and right-hand sides of the similarity rules.

| | Top-10% | Top-25% | Top-50% | Top-75% | Top-100% | Standard deviation |
|---|---|---|---|---|---|---|
| *LIN* | 0.246 *(20)* | 0.240 *(20)* | 0.243 *(20)* | 0.243 *(20)* | **0.561** *(20)* | *0.127* |
| *AP* | 0.589 *(58)* | **0.618** *(46)* | 0.598 *(26)* | - | 0.573 *(20)* | *0.016* |
| *AP$^I$* | 0.586 *(108)* | **0.624** *(36)* | 0.594 *(26)* | - | 0.572 *(20)* | *0.019* |
| *balPrecision* | **0.600** *(42)* | 0.579 *(58)* | 0.586 *(52)* | 0.596 *(32)* | 0.583 *(52)* | *0.008* |
| *balAP* | 0.596 *(26)* | **0.597** *(26)* | 0.592 *(20)* | 0.571 *(20)* | 0.560 *(20)* | *0.015* |
| *balAP$^I$* | **0.602** *(26)* | 0.589 *(26)* | **0.602** *(20)* | 0.583 *(20)* | 0.562 *(20)* | *0.015* |
| *balAP$^{II}$* | 0.579 *(68)* | 0.588 *(92)* | 0.592 *(42)* | **0.594** *(42)* | 0.595 *(42)* | *0.006* |
| *balAP$^{III}$* | 0.593 *(68)* | **0.604** *(42)* | **0.604** *(32)* | 0.601 *(32)* | 0.600 *(52)* | *0.004* |

Table 19 – Influence of tested features selection on the performance of inclusion measures for the TC task. The table shows values of $F_1$ averaged over the folds of 5-fold cross-validation. The highest result of each measure is bolded. The tuned top-*k* rules value averaged over the folds is given in parenthesis.

#### 5.3.1.1 Analysis of the desired properties of inclusion measures

We note that only the measures that comply with all the desired properties improve the result of the state-of-the-art measure of *LIN*, thus confirming the validity of the properties. Below we analyze the defined properties together with mechanisms involved in their realization.

**The balancing mechanism.** From Table 19 we see that behavior of the measures in this perspective differs from that within the ACE-based evaluation, where balanced measures consistently performed better than the unbalanced ones.

- Within the TC task the *balAP* and *balAP$^I$* measures turned out less competitive than their unbalanced prototypes – *AP* and *AP$^I$*. We note that both *AP* and *AP$^I$* measures comply with all the properties defined for an inclusion measure. In particular, they don't promote infrequent expanding words having short feature vectors, whose additional demotion is realized by means of the balancing mechanism.
- Having analyzed the similarity lists produced by the *AP* and *AP$^I$* measures, we conclude that their expanding words are indeed less frequent than those produced by their balanced versions. Their better performance as compared to the ACE task can be explained by the following reasons:
  - o In our TC evaluation we use the Reuters-10 collection, which is highly domain-specific. We note that in this work we used the Reuters RCV1 corpus for

creation of the vectors and, following the common practice, filtered out words with less than 10 occurrences. Thus, all the expanding words are actually taken from the same domain as the Reuters-10 categories. In such conditions less frequent expanding words are likely to be useful for the TC task, while being of no use within the ACE task.

- In the TC task we deal with whole documents, while in the ACE task decisions are made on the basis of a single sentence or a sentence fragment. Within such settings, less frequent expanding words have more chances to increase the *Recall* for the TC task than for the ACE task. Moreover, negative influence of a single wrong hit is less damaging in longer texts, thus increasing the possibility to achieve higher *Precision* within the TC task.

The same reasons, and also the fact that Reuters-10 categories are rather distinct and have quite indicative seeds, on the whole explain high results achieved by the measure of *LIN*, which was much less productive within the ACE task.

We note that the *AP* and $AP^I$ measures show competitive results and in optimal settings notably outperform all the balanced measures. We conclude that for similar tasks, in case when feature vectors can be created within the same domain, the *AP* and $AP^I$ measures can be successfully employed even without balancing.

Below we continue analyzing the data presented in Table 19 for the balanced inclusion measures.

**Reflecting different relevance of included features to the entailed word**. The $balAP^I$ measure performs better than *balAP*, and the $balAP^{III}$ measure performs better than $balAP^{II}$, thus showing that different relevance of features rank inside the vector of the expanded word indeed improves the inclusion measures.

**Alternative normalization using the number of tested features**. The *balAP*$^{II}$ and *balAP*$^{III}$ measures perform better than *balAP* and *balAP*$^{I}$, meaning that normalization using the number of tested features indeed improves the results of the inclusion measures.

We've thus shown that all the all the modifications, suggested for the classical *AP* formula in order to satisfy the defined properties, improve the inclusion reliability measures.

#### 5.3.1.2 Analysis of the performance

Below we analyze, according to Table 19, the performance of the *balAP*$^{III}$ measure, which involves all the proposed modifications:

Performance of the *balAP*$^{III}$ measure, involving all the suggested refinements:

1. With optimal selection of the parameter of the percentage of tested features (comparing the best $F_1$ values for each measure), we see that *balAP*$^{III}$ is the best-performing balanced measure, having two competitors – the *balPrecision* and *balAP*$^{I}$ measures with almost the same results.
2. When the percentage of tested features is not optimal (comparing the worst $F_1$ values for each measure), we see that *balAP*$^{III}$ performs best with $F_1$=0.593, while *balPrecision* has much lower $F_1$=0.579.
3. We see that *balAP*$^{III}$ is less sensitive to poor selection of the percentage of tested features than all the other measures. This fact is also confirmed by the *standard deviation* values. The lowest *standard deviation* is observed for the *balAP*$^{III}$ measure, while *balPrecision* has twice that value.

We conclude that the *balAP*$^{III}$ measure is preferable due to its high performance and lower sensitivity to the value of the parameter of the percentage of tested features.

For further analysis we will concentrate on using the whole vectors of the expanding words for inclusion testing, like it was done for the ACE-based evaluation.

Table 20 shows the results, obtained using the 5-fold cross-validation for the TC task, when taking top-100% of the features as tested.

| | *Precision* | *Recall* | $F_1$ |
|---|---|---|---|
| *LIN* | 0.425 | 0.826 | 0.561 |
| *AP* | 0.461 | 0.757 | 0.573 |
| $AP^I$ | 0.451 | 0.782 | 0.572 |
| *balPrecision* | 0.482 | 0.737 | 0.583 |
| *balAP* | 0.424 | 0.825 | 0.560 |
| $balAP^I$ | 0.427 | 0.822 | 0.562 |
| $balAP^{II}$ | 0.542 | 0.660 | 0.595 |
| $balAP^{III}$ | 0.544 | 0.668 | 0.600 |

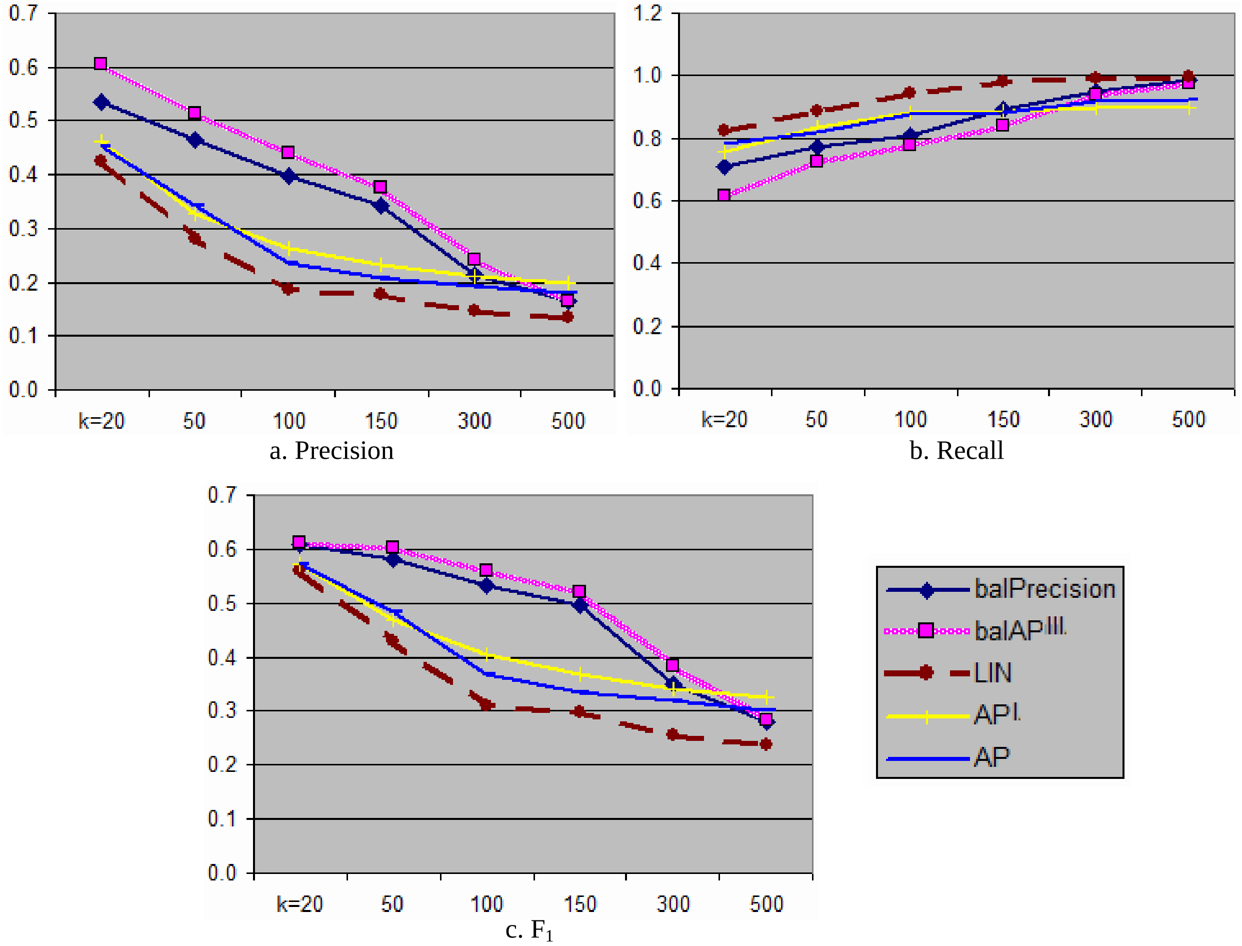
Table 20 – Results of 5-fold cross-validation for the TC task, using 100% of features as tested. For each measure the table shows *Precision* and *Recall* values averaged over the folds, and the corresponding value of $F_1$.

To assess the validity of the results we calculated statistical significance, comparing the best-performing $balAP^{III}$ measure with other evaluated measures according to the Wilcoxon signed-rank test. The test showed statistical significance of the results as compared to the common-practice symmetric measure of *LIN* with $p$=0.01. As compared to the inclusion measures, the involved test did not reveal statistical significance of the results, except for the *balAP* measure ($p$=0.01) and the $balAP^I$ measure ($p$=0.05). This can be explained by the nature of our TC evaluation, being domain-specific and dealing with long documents, and thus less sensitive to the differences in the applied inclusion-based measures.

#### 5.3.1.3 Sensitivity to the top-k rules parameter

Figure 5 shows *Precision*, *Recall* and $F_1$ at different cut-off points of top-*k* expanding words.

In order not to overburden the diagrams, we excluded the $balAP^{II}$ measure, performing slightly worse than $balAP^{III}$, and the *balAP* and $balAP^I$ measures, which perform almost the same as the measure of *LIN*.



a. Precision

b. Recall

c. $F_1$

Figure 5 – *Precision*, *Recall* and $F_1$ of different inclusion measures for using various cut-offs of top-*k* expanding words for the TC task.

From Figure 5 we see that the *balAP*$^{III}$ measure constituently outperforms all its competitors in *Precision* and in $F_1$. We also see that the improvement, introduced by the *balAP*$^{III}$ measure, is much more explicit as compared to the state-of-the-art *LIN* measure than it may be concluded from the cross-validation results presented in Table 20. The reason of such a difference is that performance of the *LIN* measure drops sharply when using more expanding words. The performance of the *balAP*$^{III}$ measure is much less sensitive to the quantity of the used expanding words, which witnesses to a much higher quality of its similarity lists and lower sensitivity to the length of the similarity list used.

### 5.3.2 Error Analysis

For further analysis we will concentrate on the *balAP*$^{III}$ inclusion measure, which proved to be the best-performing measure for the categorization task. In order to perform error analysis we sampled 50 false-positive examples of documents, choosing the first five errors for each category[1]. Thus we formed a test set of 97 wrongly applied expansion rules. The investigation of the sampled examples defined the same types of errors as the ones described in Section 5.2.2. Their distribution is shown below in Table 21.

| Type of error | Percent |
|---|---|
| Context-related expanding word | 0.35 |
| Inappropriate context of entailing word | 0.34 |
| Invalid expanding word | 0.18 |
| Ambiguous expanding term | 0.07 |
| Ambiguous seed | 0.06 |

Table 21 – Results of error analysis for the TC task. Error sources are sorted according to the percent of the errors of the corresponding type.

We see that invalid expanding rules like *opportunity* → *interest*, *result* → *earn* etc. caused 18% of errors, which is a higher quantity than observed for the ACE task. We note that almost all the invalid words and a part of the context-related words caused errors when being the only hit in the categorized document.

Like within the ACE task, valid context-related and *entailing* words yield almost the same quantity of errors. Examples of these errors are presented below in Table 22.

| Category | Sentence fragment |
|---|---|
| *Crude*[2] | …better economic performance, helped by a steadier **oil price** of around 18 dlrs a **barrel**… |
| *Trade* | …The Ministry of International **Trade** and Industry (MITI) will revise its long-term energy **supply**/**demand** outlook by August… |
| *Earn* | …Taiwan had a **trade surplus**[3] of 15.6 billion dlrs last yesr… |

Table 22 – Examples of wrong decisions originated from using valid entailing and context-related words for the TC task. The applied words are bolded in the sentence fragments.

[1] The Reuters-10 corpus is split in advance into train and test. We sampled the examples from the test part.

[2] *Crude* – oil in a natural state that has not yet been treated

[3] *Trade surplus* – a positive balance of trade, i.e. exports exceed imports (opposite of "*trade deficit*")

From Table 21 we see that, despite rather indicative seeds and domain-specific corpus, sense-related errors constitute a considerable amount of 13%. Examples of such errors are presented below in Table 23.

| | Category (seed) | Expected sense | Alternative sense | Sentence fragment |
|---|---|---|---|---|
| ambiguous seed | *Interest* | money that you earn from keeping your money in an account in a bank or other financial organization | the feeling of wanting to give your attention to something or of wanting to be involved with and to discover more about something | …The Australian government is awaiting the outcome of trade talks… with interest and **concern**… |
| | *Ship* | a vessel that carries passengers or freight | the crew of a vessel (for ex. *the ship was paid off*) | …the Federal Reserve Board has forbidden them to exchange **personnel**, or increase the business they do with each other… |
| ambiguous expanding word | *Money-fx (money)* | *(funds)* money needed or available to spend on something | a source of supply;<br>a stock | …creation of a development **fund** for further exploration… |
| | *Ship* | *(tonnage)* the size of a ship | weight measured in tons | …two underground mines in Masbate had been accelerated and the ore **tonnage** had increased… |

Table 23 – Examples of wrong decisions made for the TC task due to ambiguous seeds and expanding words. The expanding words are bolded in the sentence fragments.

### 5.3.3 Conclusions

The results of the evaluation confirmed the validity of the *Average Precision*-based approach to inclusion testing, as well as appropriateness of the modifications, proposed for the classical *AP* measure.

The *balAP*$^{III}$ measure, which accumulates all the modifications:

- Achieved the best values of *Precision* and $F_1$.
- *Precision* achieved by this measure is 12% higher than the result of the state-of-the-art measure of *LIN* and 6% higher than the result achieved by the recently proposed *balPrecision* similarity measure.

- The value of $F_1$ shown by the $balAP^{III}$ measure is 4% higher than the result of the state-of-the-art *LIN* measure, which is a considerable improvement. The result proved to be statistically significant.
- This measure proved to be least sensitive to the parameter of tested features selection and showed the best ability to tune the optimal cut-off point in the list of expansion rules.

Error analysis, performed for this measure, showed that:

- The majority (almost 70%) of errors originate from using valid expanding words in inappropriate contexts.
- Invalid expanding words lead to less than 20% of errors.
- Semantic ambiguity of seeds and expanding words caused 13% of errors.

# 6 CONCLUSIONS AND FUTURE WORK

## 6.1 Conclusions

In this work we performed the following research steps:

1. We investigated the components and parameters of the distributional inclusion scheme and defined the factors that influence distributional inclusion.
2. We formulated the desired properties of an inclusion-based measure of semantic similarity.
3. In view of these properties we analyzed the existing, *Precision*-based, inclusion measures and defined our novel inclusion measures based on extensions of the *Average Precision* metric.
4. Finally, we performed application-oriented evaluations of the inclusion measures using two different Natural Language Processing application datasets – Automatic Content Extraction (ACE) and Unsupervised Keyword-based Text Categorization (TC).

The results of our evaluation assessed the appropriateness of the defined properties and confirmed the validity of the *Average Precision*-based approach to inclusion testing:

- The extensions, proposed for the classical *AP* measure in order satisfy the defined properties, proved to be helpful.
- All of the proposed measures improved the results achieved by earlier state-of-the-art.
- One of the proposed measures ($balAP^{III}$), which includes all the extensions, showed the best performance within both NLP tasks.
- The results achieved by this measure proved to be statistically significant.

- In addition, this measure proved to be least sensitive to the major parameter of the inclusion scheme (amount of tested features).

Error analysis for this measure showed that within the majority of errors originated from using valid expanding words in inappropriate contexts, while invalid words caused much smaller amounts of errors. Considerable amounts of errors were caused by semantic ambiguity of the expanded and expanding words.

## 6.2 Future work

During this work we detected a few worthwhile directions for further research, which are described in current section.

### 6.2.1 Word Sense Disambiguation

Error analysis performed within both evaluated NLP tasks showed that many errors were caused by using ambiguous expanding and expanded words. During the work on this thesis we detected a new direction of word sense disambiguation (WSD), which seems to be promising. One of the common WSD methods is to split vectors of words into senses' vectors, in order to use vectors of senses instead of words' vectors for the similarity computation. We suggest performing such splitting using known semantically similar word pairs. Following the *Distributional Inclusion Hypotheses*, we expect all of the significant features of the sense $s_i$ of a given word $w$ to co-occur with words semantically similar to $s_i$. If a feature was not seen with any of these words, we assume that it belongs to another sense of the word[1] and thus can be filtered out of the vector of $s_i$. We plan to use the WordNet lexical database (Fellbaum, 1998), which for a given word $w$ provides lists of semantically related words separately for each of its senses and has a broad coverage both of words and of word senses. We suppose that

[1] Even if a feature belongs to the current sense, the fact that it did not co-occur with any semantically similar words shows that it will hardly be useful for automatic extraction of expanding words of the current sense.

using the WordNet database to perform vector splitting we will be able to create high-quality similarity lists separately for different senses of words. Our initial experiments in this direction showed that this approach indeed has a certain potential. Below we present an example of similarity lists obtained using this approach for a noun *suit*.

| Before splitting | After splitting using *clothing, wear, business suit, garment* | After splitting using *lawsuit, cause, causa, case, countersuit, proceeding, legal proceeding, proceedings* |
|---|---|---|
| lawsuit | scarf | lawsuit |
| shirt | attire | counterclaim |
| jacket | robe | writ |
| complaint | hat | legal proceeding |
| dress | headband | litigation |
| claim | pant | legal action |
| litigation | shirt | infringement |
| uniform | coat | complaint |
| coat | underwear | affidavit |

Table 24 – Example of word sense disambiguation based on distributional inclusion.

### 6.2.2 Alternative balancing

In this work we adopted the balancing mechanism proposed by Szpektor and Dagan (2008), which is based on multiplying the similarity scores of *LIN* and the scores assigned by inclusion measure. Virtually, this balancing mechanism allows bringing an inclusion measure into accord with one of the requirements defined in Section 4.1: *"expanding words with short vectors (small quantity of tested features) are less reliable"*. We consider it worth exploring to define an alternative balancing mechanism that would discriminate expanding words with short vectors by considering vector length directly, without involving the *LIN* measure.

### 6.2.3 Additional investigation of the bootstrapping algorithm

In our work we performed inclusion testing also using *BFW*-scored vectors, but the evaluation results were noticeably lower than the results of using *MI*-scored vectors, which were reported in our evaluation. Moreover, the results of the *LIN* similarity measure, calculated using the *BFW*-scored vectors, were lower than the *LIN* similarity for the *MI*-scored vectors. Our analysis revealed several defects of the *BFW*-scored vectors, such as an inconsistency between features ranking and their relevance to the word they represent. At the same time, the quantities of included features are on the whole much higher than for the *MI*-scored vectors (60-70% vs. 10-15%). We conclude that the bootstrapping algorithm has certain potential and is worth further research and modifications.

### 6.2.4 Manual evaluation

As argued by Glickman and Dagan (2005) suitable judgments for a given pair of words can be provided by human experts, since *"people attribute meanings to texts and can perform inferences over them"*. We think that conducting manual evaluation of the similarity lists produced by different inclusion measures would be worthwhile and would allow their additional comparison and analysis.

## REFERENCES

Roy Bar-Haim, Ido Dagan, Iddo Greental, and Eyal Shnarch. Semantic inference at the lexical-syntactic level. In Proceedings of the Twenty-Second AAAI Conference on Artificial Intelligence (AAAI‘07), pages 871–870, Vancouver, British Columbia, Canada, 2007.

Ido Dagan. Contextual Word Similarity, chapter 19, pages 459–476. Handbook of Natural Language Processing. Marcel Dekker Inc, 2000.

Ido Dagan, Oren Glickman, and Bernardo Magnini. The PASCAL Recognising Textual Entailment Challenge, volume 3944 of Lecture Notes in Computer Science. Springer, 2005.

Ido Dagan, Oren Glickman, Alfio Gliozzo, Efrat Marmorshtein, and Carlo Strapparava. Direct word sense matching for lexical substitution. In Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics, pages 449–456, Sydney, Australia, July 2006. Association for Computational Linguistics.

Ido Dagan, Lillian Lee and Fernando Pereira. 1999. Similarity-based models of cooccurrence probabilities. Machine Learning, Vol. 34(1-3), special issue on Natural Language Learning, pp. 43-69.

Ido Dagan, Shaul Marcus and Shaul Markovitch. Contextual word similarity and estimation from sparse data, Computer, Speech and Language, 1995, Vol. 9, pp. 123-152

Christiane Fellbaum, editor. WordNet: An Electronic Lexical Database. MIT Press, Cambridge, Massachusetts, 1998.

Maayan Geffet and Ido Dagan. Feature vector quality and distributional similarity. In Proceedings of the 20th international conference on Computational Linguistics (COLING ’04), pages 247–253, Geneva, Switzerland, 2004. Association for Computational Linguistics.

Maayan Geffet and Ido Dagan. The distributional inclusion hypotheses and lexical entailment. In Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL’05), pages 107–114, Ann Arbor, Michigan, June 2005. Association for Computational Linguistics.

Oren Glickman, Eyal Shnarch, and Ido Dagan. Lexical reference: a semantic matching subtask. In Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing, pages 172–179, Sydney, Australia, July 2006. Association for Computational Linguistics.

Grefenstette Gregory. 1994. Exploration in Automatic Thesaurus Discovery. Kluwer Academic Publishers. Boston.

Ralph Grishman, Lynette Hirschman, and Ngo Thanh Nhan. Discovery procedures for sublanguage selectional patterns: Initial experiments. Computational Linguistics, 12(3):205–215, 1986.

Sanda M. Harabagiu, Dan Moldovan, Marius Pa¸sca, Rada Mihalcea, Mihai Surdeanu, Razvan C. Bunescu, Roxana Gîrji, Vasile Rus, and Paul Morârescu. Falcon: Boosting knowledge for answer engines. In Proceedings of the ninth text retrieval conference (TREC-9), Gaithersburg, Maryland, 2000. NIST.

Zelig S. Harris. Mathematical Structures of Language. New York, 1968.

Marti A. Hearst. Automatic acquisition of hyponyms from large text corpora. In Proceedings of the 14th conference on Computational linguistics, pages 539–545, Nantes, France, 1992. Association for Computational Linguistics.

Youngjoong Ko and Jungyun Seo. Learning with unlabeled data for text categorization using a bootstrapping and a feature projection technique. In Proceedings of the 42nd Meeting of the Association for Computational Linguistics (ACL'04), Main Volume, pages 255–262, Barcelona, Spain, July 2004.

Lillian Lee. 1999. Measures of Distributional Similarity. In Proceedings of ACL-99. Maryland, USA.

Lillian Lee. 1997. Similarity-Based Approaches to Natural Language Processing. Ph.D. thesis, Harvard University, Cambridge, MA.

M. Lesk. 1986. Automatic sense disambiguation using machine readable dictionaries: How to tell a pine cone from an ice cream cone. In Proceedings of the ACM-SIGDOC Conference, Toronto, Canada.

Dekang Lin. Dependency-based evaluation of minipar. In Proceedings of the Workshop on Evaluation of Parsing Systems at LREC, 1998.

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

Bing Liu, Xiaoli Li, Wee Sun Lee, and Philip S. Yu. Text classification by labeling words. In Proceedings of the American Conference of Artificial Intelligence, pages 425–430, 2004.

Andrew McCallum and Kamal Nigam. Text classification by bootstrapping with keywords, em and shrinkage. In Proceedings of the ACL Workshop for unsupervised Learning in Natural Language Processing, pages 52–58, 1999.

George A. Miller, Richard Beckwith, Christiane Fellbaum, Derek Gross, and Katherine J. Miller. Introduction to wordnet: An on-line lexical database. Journal of Lexicography, 3(4):235–244, 1990.

Shachar Mirkin, Ido Dagan, and Maayan Geffet. Integrating pattern-based and distributional similarity methods for lexical entailment acquisition. In Proceedings of the COLING/ACL 2006 Main Conference Poster Sessions, pages 579–586, Sydney, Australia, 2006. Association for Computational Linguistics.

Shachar Mirkin, Ido Dagan, Eyal Shnarch. Evaluating the Inferential Utility of Lexical-Semantic Resources. 2009. The 12th Conference of the European Chapter of the Association for Computational Linguistics (EACL-09) [forthcoming].

Patrick Pantel and Dekang Lin. Discovering word senses from text. In Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining, pages 613–619, 2002.

Patrick Pantel and Deepak Ravichandran. Automatically labeling semantic classes. In Proceedings of HLT-NAACL, pages 321–328, Boston, Massachusetts, USA, 2004. Association for Computational Linguistics.

Deepak Ravichandran and Eduard Hovy. Learning surface text patterns for a question answering system. In Proceedings of 40th Annual Meeting of the Association for Computational Linguistics, pages 41–47, Philadelphia, Pennsylvania, USA, 2002. Association for Computational Linguistics.

Ellen Riloff and Rosie Jones. Learning dictionaries for information extraction by multi-level bootstrapping. In Proceedings of the Sixteenth National Conference on Artificial Intelligence (AAAI'99, pages 474–479, Orlando, Florida, United States, 1999. American Association for Artificial Intelligence.

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

Gerda Ruge. 1992. Experiments on linguistically-based term associations. Information Processing & Management, 28(3), pp. 317–332.

Sam Scott and Stan Matwin. Text classification using WordNet hypernyms. In Proceedings of the Workshop on Usage of WordNet in Natural Language Processing Systems (Coling-ACL'98), pages 45–51, Montreal, Canada, 1998. Association for Computational Linguistic.

Rion Snow, Daniel Jurafsky, and Andrew Y. Ng. Learning syntactic patterns for automatic hypernym discovery. In Lawrence K. Saul, Yair Weiss, and Léon Bottou, editors, Advances in Neural Information Processing Systems 17, pages 1297–1304. MIT Press, Cambridge, MA, 2005.

Idan Szpektor, Eyal Shnarch, and Ido Dagan. Instance-based evaluation of entailment rule acquisition. In Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics, pages 456–463, Prague, Czech Republic, June 2007. Association for Computational Linguistics.

Idan Szpektor, Ido Dagan, Roy Bar-Haim, and Jacob Goldberger. Contextual preferences. In Proceedings of ACL-08: HLT, pages 683–691, Columbus, Ohio, June 2008. Association for Computational Linguistics.

Ellen M. Voorhees. Query expansion using lexical-semantic relations. In Proceedings of the 17th Annual International ACM-SIGIR Conference on Research and Development in Information Retrieval, pages 61–69, Dublin, Ireland, 1994.

Ellen M. Voorhees. 1993. Using WordNet to disambiguate word sense for text retrieval. In SIGIR, Pittsburgh, PA.

Ellen M. Voorhees and D. Harmann, editors. 1999. Proceedings of the Seventh Text Retrieval Conference (TREC-7), Gaithersburg, MD, USA, July. NIST Special Publication.

Julie Weeds and David Weir. 2003. A General Framework for Distributional Similarity. In Proceedings of EMNLP 2003. Barcelona, Spain.

Julie Weeds, David Weir, and Diana McCarthy. Characterising measures of lexical distributional similarity. In Proceedings of the 20th international conference on Computational Linguistics (COLING ’04), pages 1015–1021, Geneva, Switzerland, 2004. Association for Computational Linguistics.

Yorick A. Wilks, Brian M. Slator, and Louise M. Guthrie. Electric words: dictionaries, computers, and meanings. MIT Press, Cambridge, MA, USA, 1996.

# APPENDIX A – LIST OF EVENTS AND SEEDS OF THE AUTOMATIC CONTENT EXTRACTION TASK

The events are given in alphabetic order. In brackets we show the list of the *seed words* for each event in “part-of-speech | lemma” format with “a” for adjective, “n” for noun and “v” for verb.

*Arrest-Jail* (v|arrest, n|jail, n|arrest, n|arrester, n|jailer, a|jailed, v|jail, n|jailor)

*Attack* (v|attack, n|attack, n|attacker)

*Be-Born* (n|birth, a|born)

*Charge-Indict* (v|charge, n|charger, v|indict, n|charge, n|indictment)

*Convict* (a|guilty, n|guilt, n|convict, n|conviction, n|guiltiness, v|convict)

*Demonstrate* (n|demonstration, n|demonstrator, v|demonstrate, a|demonstrative)

*Die* (n|death, v|die, n|dying, n|die, a|deathly, n|death, a|dead)

*Elect* (a|eligible, n|elector, n|elect, n|electorate, a|electoral, a|elective, v|elect, a|elected, n|election)

*End-Org* (n|fold, n|folding, v|fold, n|folder)

*End-Position* (n|fire, v|fire, n|firing, n|layoff)

*Injure* (v|injure, a|injured, n|injury)

*Marry* (a|married, n|marriage, n|married, v|marry)

*Meet* (n|meeter, n|meet, n|meeting, v|meet)

*Phone-Write* (n|writing, n|call, n|writer, v|send, n|sending, v|phone, n|phone, n|sender, a|callable, n|phoner, n|caller, n|sendee, n|calling, n|call, v|call, v|write)

*Sentence* (n|sentence, v|sentence)

*Start-Org* (v|found, v|launch, n|founder, n|launch, n|foundation, n|launcher, n|founding)

*Start-Position* (v|hire, n|hirer)

*Sue* (n|suit, n|lawsuit, v|sue, n|suer)

*Transfer-Money* (n|borrower, n|extortion, n|donation, v|borrow, v|donate, v|extort, n|extortionist)

*Transfer-Ownership* (a|acquirable, v|sell, n|acquisition, n|seller, n|buyer, n|purchase, n|acquiring, n|acquirer, n|selling, a|acquisitive, v|buy, n|purchasing, n|purchaser, n|sell, v|acquire, n|buy, v|purchase, n|buying)

*Transport* (n|travelling, n|traveler, n|transporter, v|transport, n|transport, n|traveling, n|transportation, n|travel, v|travel, n|traveler)

*Trial-Hearing* (n|trial, v|try, v|hear, n|trier, n|try, n|hearing).

# ABSTRACT (HEBREW)

אחת מבעיות היסוד העומדות בפני יישומים ל״הבנת טקסטים״ הוא שניתן לבטא את אותה משמעות בדרכים רבות. על מנת להתמודד עם בעיה זו, מערכות רבות לעיבוד שפות טבעיות משתמשות ב-*הרחבה לקסיקלית* (*lexical expansion*), כלומר שימוש במילים או ביטויים בעלי משמעות קרובה למילות הטקסט הנתון.

הגישה הרווחת ללמידה אוטומטית של הרחבות מסוג זה מבוססת על ״הנחת הדמיון ההתפלגותי״ (*Distributional Similarity Hypothesis*) , על פיה מלים המופיעות בהקשרים דומים צפויות להיות בעלות משמעות קרובה (Harris, 1968). החיסרון המרכזי של השיטות המקובלות במסגרת הנחה זו הוא הדיוק הנמוך שלהן ביחס למשימת ההרחבה הלקסיקלית. חיסרון נוסף הוא שהדמיון ההתפלגותי הינו מודל סימטרי, אשר אינו מבחין בין המילה ׳המורחבת׳ (*expanded word*) ל׳מרחיבה׳ (*expanding word*), ואילו במקרים רבים המשימה דורשת גישה מכוונת.

מחקרים אחרונים בתחום הדמיון ההתפלגותי מציעים גישה חדשה בשם ׳הכלה התפלגותית׳ (*Distributional Inclusion*), המניחה שהמאפיינים הסמנטיים העיקריים של המילה ׳המרחיבה׳ אמורים להופיע גם עם המילה ׳המורחבת׳. מחקר זה בוחן את השימושיות של גישה זו ללמידה אוטומטית מבוססת-קורפוס של חוקי דמיון א-סימטריים לצורך הרחבה לקסיקלית. בעבודה זו ניתחנו מרכיבים ופרמטרים של מדדי דמיון המתייחסים להכלה התפלגותית, ניסחנו את המאפיינים הרצויים של מדד דמיון מבוסס הכלה והגדרנו מדדי דמיון חדשים. כמו כן ביצענו הערכה אמפירית של המדדים הנ״ל במסגרת שתי משימות שונות של עיבוד שפה טבעית והראנו שיפור משמעותי ביחס למדדים הקיימים עד כה.

עבודה זו נעשתה בהדרכתו של דר' עידו דגן

מן הפקולטה למדעי המחשב

של אוניברסיטת בר-אילן

אוניברסיטת בר-אילן
המחלקה למדעי המחשב

# דמיון התפלגותי א-סימטרי להרחבה לקסיקלית

## לילי קוטלרמן

עבודה זו מוגשת כחלק מהדרישות לשם קבלת תואר מוסמך
בפקולטה למדעי המחשב של אוניברסיטת בר-אילן

רמת-גן, ישראל        אפריל 2009, ניסן תשס"ט