Direct Word Sense Matching for Lexical Substitution

Ido Dagan[1], **Oren Glickman**[1], **Alfio Gliozzo**[2], **Efrat Marmorshtein**[1], **Carlo Strapparava**[2]
[1]Department of Computer Science, Bar Ilan University, Ramat Gan, 52900, Israel
[2]ITC-Irst, via Sommarive, I-38050, Trento, Italy

Abstract

This paper investigates conceptually and empirically the novel *sense matching* task, which requires to recognize whether the senses of two synonymous words match in context. We suggest direct approaches to the problem, which avoid the intermediate step of explicit word sense disambiguation, and demonstrate their appealing advantages and stimulating potential for future research.

1 Introduction

In many language processing settings it is needed to recognize that a given word or term may be substituted by a synonymous one. In a typical information seeking scenario, an information need is specified by some given *source* words. When looking for texts that match the specified need the source words might be substituted with synonymous *target* words. For example, given the source word 'weapon' a system may substitute it with the target synonym 'arm'.

This scenario, which is generally referred here as *lexical substitution*, is a common technique for increasing recall in Natural Language Processing (NLP) applications. In Information Retrieval (IR) and Question Answering (QA) it is typically termed query/question expansion (Moldovan and Mihalcea, 2000; Negri, 2004). Lexical Substitution is also commonly applied to identify synonyms in text summarization, for paraphrasing in text generation, or is integrated into the features of supervised tasks such as Text Categorization and Information Extraction. Naturally, lexical substitution is a very common first step in textual entailment recognition, which models semantic inference between a pair of texts in a generalized application independent setting (Dagan et al., 2005).

To perform lexical substitution NLP applications typically utilize a knowledge source of synonymous word pairs. The most commonly used resource for lexical substitution is the manually constructed WordNet (Fellbaum, 1998). Another option is to use statistical word similarities, such as in the database constructed by Dekang Lin (Lin, 1998). We generically refer to such resources as *substitution lexicons*.

When using a substitution lexicon it is assumed that there are *some* contexts in which the given synonymous words share the same meaning. Yet, due to polysemy, it is needed to verify that the senses of the two words do indeed match in a given context. For example, there are contexts in which the source word 'weapon' may be substituted by the target word 'arm'; however one should recognize that 'arm' has a different sense than 'weapon' in sentences such as "repetitive movements could cause injuries to hands, wrists and arms."

A commonly proposed approach to address sense matching in lexical substitution is applying Word Sense Disambiguation (WSD) to identify the senses of the source and target words. Then, substitution is applied only if the words have the same sense (or synset, in WordNet terminology). In settings in which the source is given as a single term without context, sense disambiguation is performed only for the target word; substitution is then applied only if the target word's sense matches at least one of the possible senses of the source word.

One might observe that such application of WSD addresses the task at hand in a somewhat indirect manner. In fact, lexical substitution only requires knowing that the source and target senses

Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the ACL, pages 449–456,
Sydney, July 2006.

do match, but it does not require that the matching senses will be explicitly identified. Selecting explicitly the right sense in context, which is then followed by verifying the desired matching, might be solving a harder intermediate problem than required. Instead, we can define the *sense matching* problem directly as a binary classification task for a pair of synonymous source and target words. This task requires to decide whether the senses of the two words do or do not match in a given context (but it does not require to identify explicitly the identity of the matching senses).

A highly related task was proposed in (McCarthy, 2002). McCarthy's proposal was to ask systems to suggest possible "semantically similar replacements" of a target word in context, where alternative replacements should be grouped together. While this task is somewhat more complicated as an evaluation setting than our binary recognition task, it was motivated by similar observations and applied goals. From another perspective, sense matching may be viewed as a lexical sub-case of the general textual entailment recognition setting, where we need to recognize whether the meaning of the target word "entails" the meaning of the source word in a given context.

This paper provides a first investigation of the sense matching problem. To allow comparison with the classical WSD setting we derived an evaluation dataset for the new problem from the Senseval-3 English lexical sample dataset (Mihalcea and Edmonds, 2004). We then evaluated alternative supervised and unsupervised methods that perform sense matching either *indirectly* or *directly* (i.e. with or without the intermediate sense identification step). Our findings suggest that in the supervised setting the results of the direct and indirect approaches are comparable. However, addressing directly the binary classification task has practical advantages and can yield high precision values, as desired in precision-oriented applications such as IR and QA.

More importantly, direct sense matching sets the ground for implicit unsupervised approaches that may utilize practically unlimited volumes of unlabeled training data. Furthermore, such approaches circumvent the sisyphean need for specifying explicitly a set of stipulated senses. We present an initial implementation of such an approach using a one-class classifier, which is trained on unlabeled occurrences of the source word and applied to occurrences of the target word. Our current results outperform the unsupervised baseline and put forth a whole new direction for future research.

2 WSD and Lexical Expansion

Despite certain initial skepticism about the usefulness of WSD in practical tasks (Voorhees, 1993; Sanderson, 1994), there is some evidence that WSD can improve performance in typical NLP tasks such as IR and QA. For example, (Shütze and Pederson, 1995) gives clear indication of the potential for WSD to improve the precision of an IR system. They tested the use of WSD on a standard IR test collection (TREC-1B), improving precision by more than 4%.

The use of WSD has produced successful experiments for query expansion techniques. In particular, some attempts exploited WordNet to enrich queries with semantically-related terms. For instance, (Voorhees, 1994) manually expanded 50 queries over the TREC-1 collection using synonymy and other WordNet relations. She found that the expansion was useful with short and incomplete queries, leaving the task of proper automatic expansion as an open problem.

(Gonzalo et al., 1998) demonstrates an increment in performance over an IR test collection using the sense data contained in SemCor over a purely term based model. In practice, they experimented searching SemCor with disambiguated and expanded queries. Their work shows that a WSD system, even if not performing perfectly, combined with synonymy enrichment increases retrieval performance.

(Moldovan and Mihalcea, 2000) introduces the idea of using WordNet to extend Web searches based on semantic similarity. Their results showed that WSD-based query expansion actually improves retrieval performance in a Web scenario. Recently (Negri, 2004) proposed a sense-based relevance feedback scheme for query enrichment in a QA scenario (TREC-2003 and ACQUAINT), demonstrating improvement in retrieval performance.

While all these works clearly show the potential usefulness of WSD in practical tasks, nonetheless they do not necessarily justify the efforts for refining fine-grained sense repositories and for building large sense-tagged corpora. We suggest that the sense matching task, as presented in the introduction, may relieve major drawbacks of applying WSD in practical scenarios.

3 Problem Setting and Dataset

To investigate the direct sense matching problem it is necessary to obtain an appropriate dataset of examples for this binary classification task, along with gold standard annotation. While there is no such standard (application independent) dataset available it is possible to derive it automatically from existing WSD evaluation datasets, as described below. This methodology also allows comparing direct approaches for sense matching with classical indirect approaches, which apply an intermediate step of identifying the most likely WordNet sense.

We derived our dataset from the Senseval-3 English lexical sample dataset (Mihalcea and Edmonds, 2004), taking all 25 nouns, adjectives and adverbs in this sample. Verbs were excluded since their sense annotation in Senseval-3 is not based on WordNet senses. The Senseval dataset includes a set of example occurrences in context for each word, split to training and test sets, where each example is manually annotated with the corresponding WordNet synset.

For the sense matching setting we need examples of pairs of *source-target* synonymous words, where at least one of these words should occur in a given context. Following an applicative motivation, we mimic an IR setting in which a single source word query is expanded (substituted) by a synonymous target word. Then, it is needed to identify contexts in which the target word appears in a sense that matches the source word. Accordingly, we considered each of the 25 words in the Senseval sample as a target word for the sense matching task. Next, we had to pick for each target word a corresponding synonym to play the role of the source word. This was done by creating a list of all WordNet synonyms of the target word, under all its possible senses, and picking randomly one of the synonyms as the source word. For example, the word 'disc' is one of the words in the Senseval lexical sample. For this target word the synonym 'record' was picked, which matches 'disc' in its musical sense. Overall, 59% of all possible synsets of our target words included an additional synonym, which could play the role of the source word (that is, 41% of the synsets consisted of the target word only). Similarly, 62% of the test examples of the target words were annotated by a synset that included an additional synonym.

While creating source-target synonym pairs it was evident that many WordNet synonyms correspond to very infrequent senses or word usages, such as the WordNet synonyms *germ* and *source*. Such source synonyms are useless for evaluating sense matching with the target word since the senses of the two words would rarely match in perceivable contexts. In fact, considering our motivation for lexical substitution, it is usually desired to exclude such obscure synonym pairs from substitution lexicons in practical applications, since they would mostly introduce noise to the system. To avoid this problem the list of WordNet synonyms for each target word was filtered by a lexicographer, who excluded manually obscure synonyms that seemed worthless in practice. The source synonym for each target word was then picked randomly from the filtered list. Table 1 shows the 25 source-target pairs created for our experiments. In future work it may be possible to apply automatic methods for filtering infrequent sense correspondences in the dataset, by adopting algorithms such as in (McCarthy et al., 2004).

Having source-target synonym pairs, a classification instance for the sense matching task is created from each example occurrence of the target word in the Senseval dataset. A classification instance is thus defined by a pair of source and target words and a given occurrence of the target word in context. The instance should be classified as *positive* if the sense of the target word in the given context matches one of the possible senses of the source word, and as *negative* otherwise. Table 2 illustrates positive and negative example instances for the source-target synonym pair 'record-disc', where only occurrences of 'disc' in the musical sense are considered positive.

The gold standard annotation for the binary sense matching task can be derived automatically from the Senseval annotations and the corresponding WordNet synsets. An example occurrence of the target word is considered positive if the annotated synset for that example includes also the source word, and Negative otherwise. Notice that different positive examples might correspond to different senses of the target word. This happens when the source and target share several senses, and hence they appear together in several synsets. Finally, since in Senseval an example may be an-

source-target	*source-target*	*source-target*	*source-target*	*source-target*
statement-argument	subdivision-arm	atm-atmosphere	hearing-audience	camber-bank
level-degree	deviation-difference	dissimilar-different	trouble-difficulty	record-disc
raging-hot	ikon-image	crucial-important	sake-interest	bare-simple
opinion-judgment	arrangement-organization	newspaper-paper	company-party	substantial-solid
execution-performance	design-plan	protection-shelter	variety-sort	root-source

Table 1: Source and target pairs

sentence	*annotation*
This is anyway a stunning *disc*, thanks to the playing of the Moscow Virtuosi with Spivakov.	positive
He said computer networks would not be affected and copies of information should be made on floppy *discs*.	negative
Before the dead soldier was placed in the ditch his personal possessions were removed, leaving one *disc* on the body for identification purposes	negative

Table 2: positive and negative examples for the source-target synonym pair 'record-disc'

notated with more than one sense, it was considered positive if any of the annotated synsets for the target word includes the source word.

Using this procedure we derived gold standard annotations for all the examples in the Senseval-3 training section for our 25 target words. For the test set we took up to 40 test examples for each target word (some words had fewer test examples), yielding 913 test examples in total, out of which 239 were positive. This test set was used to evaluate the sense matching methods described in the next section.

4 Investigated Methods

As explained in the introduction, the sense matching task may be addressed by two general approaches. The traditional *indirect* approach would first disambiguate the target word relative to a predefined set of senses, using standard WSD methods, and would then verify that the selected sense matches the source word. On the other hand, a *direct* approach would address the binary sense matching task directly, without selecting explicitly a concrete sense for the target word. This section describes the alternative methods we investigated under supervised and unsupervised settings. The supervised methods utilize manual sense annotations for the given source and target words while unsupervised methods do not require any annotated sense examples. For the indirect approach we assume the standard WordNet sense repository and corresponding annotations of the target words with WordNet synsets.

4.1 Feature set and classifier

As a vehicle for investigating different classification approaches we implemented a "vanilla" state of the art architecture for WSD. Following common practice in feature extraction (e.g. (Yarowsky, 1994)), and using the mxpost[1] part of speech tagger and WordNet's lemmatization, the following feature set was used: bag of word lemmas for the context words in the preceding, current and following sentence; unigrams of lemmas and parts of speech in a window of +/- three words, where each position provides a distinct feature; and bigrams of lemmas in the same window. The SVM-Light (Joachims, 1999) classifier was used in the supervised settings with its default parameters. To obtain a multi-class classifier we used a standard one-vs-all approach of training a binary SVM for each possible sense and then selecting the highest scoring sense for a test example.

To verify that our implementation provides a reasonable replication of state of the art WSD we applied it to the standard Senseval-3 Lexical Sample WSD task. The obtained accuracy[2] was 66.7%, which compares reasonably with the mid-range of systems in the Senseval-3 benchmark (Mihalcea and Edmonds, 2004). This figure is just a few percent lower than the (quite complicated) best Senseval-3 system, which achieved about 73% accuracy, and it is much higher than the standard Senseval baselines. We thus regard our classifier as a fair vehicle for comparing the alternative approaches for sense matching on equal grounds.

[1] ftp://ftp.cis.upenn.edu/pub/adwait/jmx/jmx.tar.gz

[2] The standard classification accuracy measure equals precision and recall as defined in the Senseval terminology when the system classifies all examples, with no abstentions.

4.2 Supervised Methods

4.2.1 Indirect approach

The *indirect* approach for sense matching follows the traditional scheme of performing WSD for lexical substitution. First, the WSD classifier described above was trained for the target words of our dataset, using the Senseval-3 sense annotated training data for these words. Then, the classifier was applied to the test examples of the target words, selecting the most likely sense for each example. Finally, an example was classified as positive if the selected synset for the target word includes the source word, and as negative otherwise.

4.2.2 Direct approach

As explained above, the *direct* approach addresses the binary sense matching task directly, without selecting explicitly a sense for the target word. In the supervised setting it is easy to obtain such a binary classifier using the annotation scheme described in Section 3. Under this scheme an example was annotated as positive (for the binary sense matching task) if the source word is included in the Senseval gold standard synset of the target word. We trained the classifier using the set of Senseval-3 training examples for each target word, considering their derived binary annotations. Finally, the trained classifier was applied to the test examples of the target words, yielding directly a binary positive-negative classification.

4.3 Unsupervised Methods

It is well known that obtaining annotated training examples for WSD tasks is very expensive, and is often considered infeasible in unrestricted domains. Therefore, many researchers investigated unsupervised methods, which do not require annotated examples. Unsupervised approaches have usually been investigated within Senseval using the "All Words" dataset, which does not include training examples. In this paper we preferred using the same test set which was used for the supervised setting (created from the Senseval-3 "Lexical Sample" dataset, as described above), in order to enable comparison between the two settings. Naturally, in the unsupervised setting the sense labels in the training set were not utilized.

4.3.1 Indirect approach

State-of-the-art unsupervised WSD systems are quite complex and they are not easy to be replicated. Thus, we implemented the unsupervised version of the Lesk algorithm (Lesk, 1986) as a reference system, since it is considered a standard simple baseline for unsupervised approaches. The Lesk algorithm is one of the first algorithms developed for semantic disambiguation of all-words in unrestricted text. In its original unsupervised version, the only resource required by the algorithm is a machine readable dictionary with one definition for each possible word sense. The algorithm looks for words in the sense definitions that overlap with context words in the given sentence, and chooses the sense that yields maximal word overlap. We implemented a version of this algorithm using WordNet sense-definitions with context length of ±10 words before and after the target word.

4.3.2 The direct approach: one-class learning

The unsupervised settings for the direct method are more problematic because most of unsupervised WSD algorithms (such as the Lesk algorithm) rely on dictionary definitions. For this reason, standard unsupervised techniques cannot be applied in a direct approach for sense matching, in which the only external information is a substitution lexicon.

In this subsection we present a direct unsupervised method for sense matching. It is based on the assumption that typical contexts in which both the source and target words appear correspond to their matching senses. Unlabeled occurrences of the source word can then be used to provide evidence for lexical substitution because they allow us to recognize whether the sense of the target word matches that of the source. Our strategy is to represent in a learning model the typical contexts of the source word in unlabeled training data. Then, we exploit such model to match the contexts of the target word, providing a decision criterion for sense matching. In other words, we expect that under a matching sense the target word would occur in prototypical contexts of the source word.

To implement such approach we need a learning technique that does not rely on the availability of negative evidence, that is, a one-class learning algorithm. In general, the classification performance of one-class approaches is usually quite poor, if compared to supervised approaches for the same tasks. However, in many practical settings one-class learning is the only available solution.

For our experiments we adopted the one-class SVM learning algorithm (Schölkopf et al., 2001) implemented in the LIBSVM package,[3] and represented the unlabeled training examples by adopting the feature set described in Subsection 4.1. Roughly speaking, a one-class SVM estimates the smallest hypersphere enclosing most of the training data. New test instances are then classified positively if they lie inside the sphere, while outliers are regarded as negatives. The ratio between the width of the enclosed region and the number of misclassified training examples can be varied by setting the parameter $\nu \in (0, 1)$. Smaller values of ν will produce larger positive regions, with the effect of increasing recall.

The appealing advantage of adopting one-class learning for sense matching is that it allows us to define a very elegant learning scenario, in which it is possible to train "off-line" a different classifier for each (source) word in the lexicon. Such a classifier can then be used to match the sense of any possible target word for the source which is given in the substitution lexicon. This is in contrast to the direct supervised method proposed in Subsection 4.2, where a different classifier for each pair of source - target words has to be defined.

5 Evaluation

5.1 Evaluation measures and baselines

In the lexical substitution (and expansion) setting, the standard WSD metrics (Mihalcea and Edmonds, 2004) are not suitable, because we are interested in the binary decision of whether the target word matches the sense of a given source word. In analogy to IR, we are more interested in positive assignments, while the opposite case (i.e. when the two words cannot be substituted) is less interesting. Accordingly, we utilize the standard definitions of precision, recall and F_1 typically used in IR benchmarks. In the rest of this section we will report micro averages for these measures on the test set described in Section 3.

Following the Senseval methodology, we evaluated two different baselines for unsupervised and supervised methods. The random baseline, used for the unsupervised algorithms, was obtained by choosing either the positive or the negative class at random resulting in $P = 0.262$, $R = 0.5$, $F_1 = 0.344$. The *Most Frequent* baseline has been used for the supervised algorithms and is obtained by assigning the positive class when the percentage of positive examples in the training set is above 50%, resulting in $P = 0.65$, $R = 0.41$, $F_1 = 0.51$.

5.2 Supervised Methods

Both the indirect and the direct supervised methods presented in Subsection 4.2 have been tested and compared to the most frequent baseline.

Indirect. For the indirect methodology we trained the supervised WSD system for each target word on the sense-tagged training sample. As described in Subsection 4.2, we implemented a simple SVM-based WSD system (see Section 4.2) and applied it to the sense-matching task. Results are reported in Table 3. The direct strategy surpasses the most frequent baseline F1 score, but the achieved precision is still below it. We note that in this multi-class setting it is less straightforward to tradeoff recall for precision, as all senses compete with each other.

Direct. In the direct supervised setting, sense matching is performed by training a binary classifier, as described in Subsection 4.2.

The advantage of adopting a binary classification strategy is that the precision/recall tradeoff can be tuned in a meaningful way. In SVM learning, such tuning is achieved by varying the parameter J, that allows us to modify the cost function of the SVM learning algorithm. If $J = 1$ (default), the weight for the positive examples is equal to the weight for the negatives. When $J > 1$, negative examples are penalized (increasing recall), while, whenever $0 < J < 1$, positive examples are penalized (increasing precision). Results obtained by varying this parameter are reported in Figure 1.

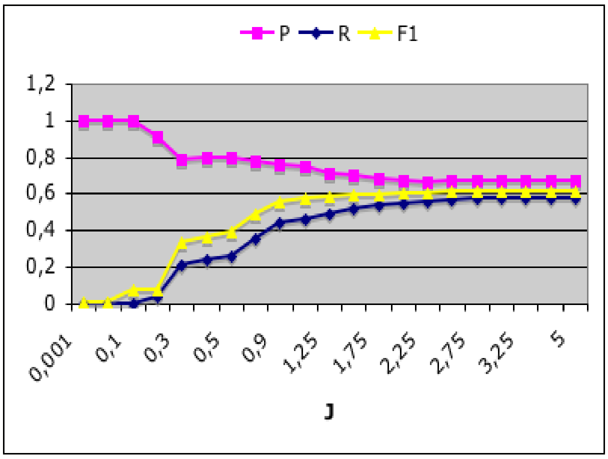

Figure 1: Direct supervised results varying J

[3] Freely available from www.csie.ntu.edu.tw/ /~cjlin/libsvm.

Supervised		P	R	F_1	Unsupervised		P	R	F_1
Most Frequent	Baseline	0.65	0.41	0.51	Random	Baseline	0.26	0.50	0.34
Multiclass SVM	Indirect	0.59	0.63	0.61	Lesk	Indirect	0.24	0.19	0.21
Binary SVM ($J = 0.5$)	Direct	0.80	0.26	0.39	One-Class $\nu = 0.3$	Direct	0.26	0.72	0.39
Binary SVM ($J = 1$)	Direct	0.76	0.46	0.57	One-Class $\nu = 0.5$	Direct	0.29	0.56	0.38
Binary SVM ($J = 2$)	Direct	0.68	0.53	0.60	One-Class $\nu = 0.7$	Direct	0.28	0.36	0.32
Binary SVM ($J = 3$)	Direct	0.69	0.55	0.61	One-Class $\nu = 0.9$	Direct	0.23	0.10	0.14

Table 3: Classification results on the sense matching task

Adopting the standard parameter settings (i.e. $J = 1$, see Table 3), the F_1 of the system is slightly lower than for the indirect approach, while it reaches the indirect figures when J increases. More importantly, reducing J allows us to boost precision towards 100%. This feature is of great interest for lexical substitution, particularly in precision oriented applications like IR and QA, for filtering irrelevant candidate answers or documents.

5.3 Unsupervised methods

Indirect. To evaluate the indirect unsupervised settings we implemented the Lesk algorithm, described in Subsection 4.3.1, and evaluated it on the sense matching task. The obtained figures, reported in Table 3, are clearly below the baseline, suggesting that simple unsupervised indirect strategies cannot be used for this task. In fact, the error of the first step, due to low WSD accuracy of the unsupervised technique, is propagated in the second step, producing poor sense matching. Unfortunately, state-of-the-art unsupervised systems are actually not much better than Lesk on all-words task (Mihalcea and Edmonds, 2004), discouraging the use of unsupervised indirect methods for the sense matching task.

Direct. Conceptually, the most appealing solution for the sense matching task is the one-class approach proposed for the direct method (Section 4.3.2). To perform our experiments, we trained a different one-class SVM for each source word, using a sample of its unlabeled occurrences in the BNC corpus as training set. To avoid huge training sets and to speed up the learning process, we fixed the maximum number of training examples to 10000 occurrences per word, collecting on average about 6500 occurrences per word.

For each target word in the test sample, we applied the classifier of the corresponding source word. Results for different values of ν are reported in Figure 2 and summarized in Table 3.

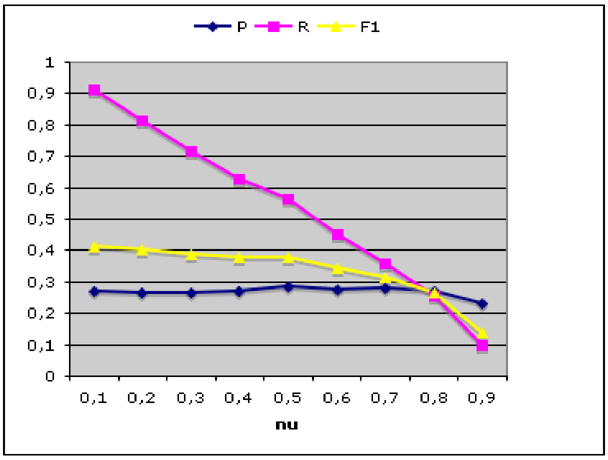

Figure 2: One-class evaluation varying ν

While the results are somewhat above the baseline, just small improvements in precision are reported, and recall is higher than the baseline for $\nu < 0.6$. Such small improvements may suggest that we are following a relevant direction, even though they may not be useful yet for an applied sense-matching setting.

Further analysis of the classification results for each word revealed that optimal F_1 values are obtained by adopting different values of ν for different words. In the optimal (in retrospect) parameter settings for each word, performance for the test set is noticeably boosted, achieving $P = 0.40$, $R = 0.85$ and $F_1 = 0.54$. Finding a principled unsupervised way to automatically tune the ν parameter is thus a promising direction for future work.

Investigating further the results per word, we found that the correlation coefficient between the optimal ν values and the degree of polysemy of the corresponding source words is 0.35. More interestingly, we noticed a negative correlation (r = -0.30) between the achieved F_1 and the degree of polysemy of the word, suggesting that polysemous source words provide poor training models for sense matching. This can be explained by observing that polysemous source words can be substituted with the target words only for a strict subset of their senses. On the other hand, our one-class algorithm was trained on *all* the examples of the source word, which include irrelevant examples that yield noisy training sets. A possible solution may be obtained using clustering-based word sense discrimination methods (Pedersen and Bruce, 1997; Schütze, 1998), in order to train different one-class models from different sense clusters. Overall, the analysis suggests that future research may obtain better binary classifiers based just on unlabeled examples of the source word.

6 Conclusion

This paper investigated the *sense matching* task, which captures directly the polysemy problem in lexical substitution. We proposed a direct approach for the task, suggesting the advantages of natural control of precision/recall tradeoff, avoiding the need in an explicitly defined sense repository, and, most appealing, the potential for novel completely unsupervised learning schemes. We speculate that there is a great potential for such approaches, and suggest that sense matching may become an appealing problem and possible track in lexical semantic evaluations.

Acknowledgments

This work was partly developed under the collaboration ITC-irst/University of Haifa.

References

Ido Dagan, Oren Glickman, and Bernardo Magnini. 2005. The pascal recognising textual entailment challenge. Proceedings of the PASCAL Challenges Workshop on Recognising Textual Entailment.

C. Fellbaum. 1998. *WordNet. An Electronic Lexical Database*. MIT Press.

J. Gonzalo, F. Verdejo, I. Chugur, and J. Cigarran. 1998. Indexing with wordnet synsets can improve text retrieval. In *ACL*, Montreal, Canada.

T. Joachims. 1999. Making large-scale SVM learning practical. In B. Schölkopf, C. Burges, and A. Smola, editors, *Advances in kernel methods: support vector learning*, chapter 11, pages 169 – 184. MIT Press.

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. 1998. Automatic retrieval and clustering of similar words. In *Proceedings of the 17th international conference on Computational linguistics*, pages 768–774, Morristown, NJ, USA. Association for Computational Linguistics.

Diana McCarthy, Rob Koeling, Julie Weeds, and John Carroll. 2004. Automatic identification of infrequent word senses. In *Proceedings of COLING*, pages 1220–1226.

Diana McCarthy. 2002. Lexical substitution as a task for wsd evaluation. In *Proceedings of the ACL-02 workshop on Word sense disambiguation*, pages 109–115, Morristown, NJ, USA. Association for Computational Linguistics.

R. Mihalcea and P. Edmonds, editors. 2004. *Proceedings of SENSEVAL-3: Third International Workshop on the Evaluation of Systems for the Semantic Analysis of Text*, Barcelona, Spain, July.

D. Moldovan and R. Mihalcea. 2000. Using wordnet and lexical operators to improve internet searches. *IEEE Internet Computing*, 4(1):34–43, January.

M. Negri. 2004. Sense-based blind relevance feedback for question answering. In *SIGIR-2004 Workshop on Information Retrieval For Question Answering (IR4QA)*, Sheffield, UK, July.

T. Pedersen and R. Bruce. 1997. Distinguishing word sense in untagged text. In *EMNLP*, Providence, August.

M. Sanderson. 1994. Word sense disambiguation and information retrieval. In *SIGIR*, Dublin, Ireland, June.

B. Schölkopf, J. Platt, J. Shawe-Taylor, A. J. Smola, and R. C. Williamson. 2001. Estimating the support of a high-dimensional distribution. *Neural Computation*, 13:1443–1471.

H. Schütze. 1998. Automatic word sense discrimination. *Computational Linguistics*, 24(1).

H. Shütze and J. Pederson. 1995. Information retrieval based on word senses. In *Proceedings of the 4^{th} Annual Symposium on Document Analysis and Information Retrieval*, Las Vegas.

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

E. Voorhees. 1994. Query expansion using lexical-semantic relations. In *Proceedings of the 17^{th} ACM SIGIR Conference*, Dublin, Ireland, June.

D. Yarowsky. 1994. Decision lists for lexical ambiguity resolution: Application to accent restoration in spanish and french. In *ACL*, pages 88–95, Las Cruces, New Mexico.