Evaluating LexCSD—a Weakly-Supervised Method on Improved Semantically Annotated Corpus in a Large Scale Experiment*

Bartosz Broda, Maciej Piasecki, and Marek Maziarz
Institute of Informatics, Wrocław University of Technology, Poland

Abstract

Word Sense Disambiguation in text is still a difficult problem as the best supervised methods require laborious and costly manual preparation of training data. On the other hand, the unsupervised methods express significantly lower accuracy and produce results that are not satisfying for many application. Recently, an algorithm based on weakly-supervised learning for WSD called Lexicographer-Controlled Semi-automatic Sense Disambiguation (LexCSD) was proposed. The method is based on clustering of text snippets including words in focus. For each cluster we find a core which is labelled with a word sense by a human and is used to produce a classifier. Classifiers, constructed for each word separately, are applied to text. The goal of this work is to evaluate LexCSD trained on large amount of untagged text. A performed comparison showed that the approach is better than most frequent sense baseline in most cases and in some cases beat the supervised equivalents. For the need of experiment semantically annotated corpus was improved in terms of coverage of sense inventory and annotations.

1 Introduction

The aim of Word Sense Disambiguation (WSD) is to choose the right sense (lexical meaning) for a word in a context. Many words have more then one sense, but usually only one of them is active in a given context. F.e., an electronic thesaurus called WordNet (Fellbaum et al., 1998) has 36 entries for line. WSD is difficult, but important problem for many applications in Natural Language Processing (NLP). The field of machine translation is an obvious example as the use of robust WSD system helps in choosing the correct translation in contexts. Also information retrieval, information extraction, text mining or computer-aided lexicography could benefit from high quality WSD system (Agirre and Edmonds, 2006).

WSD is not an easy problem to solve, partially because the definition of lexical meaning is not clear and the boundaries between different senses are not crisp and obvious (Kilgarriff, 2006). To overcome theoretical aspect of this problem dictionaries are used as a mean to enumerate all of the different word senses. In WSD a set of senses is called a sense inventory.

*This work was supported by fellowship co-financed by European Union within European Social Fund and co-financed by Innovative Economy Programme project POIG.01.01.02-14-013/09.
There are two main approaches to WSD based on machine learning: supervised and unsupervised (Agirre and Edmonds, 2006).\textsuperscript{1} Supervised learning focuses on the usage of manually disambiguated examples of text snippets containing ambiguous words. We need to choose an appropriate sense inventory in advance, at early stages of the construction of supervised WSD system. Some features are extracted from those text snippets (or contexts\textsuperscript{2}) and classifiers are trained using this manually labeled data. Usually, supervised approaches are superior to unsupervised in terms of accuracy of automatic disambiguation when used on the same type of texts that the systems were trained on.

There are two main drawbacks of supervised approaches to WSD: building manually annotated datasets for learning and domain adaptation. Building manually disambiguated corpora is very laborious and error-prone process. Construction of such a corpus for 20,000 ambiguous words would require 80 man-years of work according to Mihalcea (2003). Even more discouraging is the fact that every change in the domain of texts would require costly adaptation and extension of training corpus to incorporate domain-specific word senses and sense distribution.

Unsupervised approaches to WSD tend to use unlabeled data and automatically find sense distinctions. Usually those methods involve some form of clustering. Harris’ distributional hypothesis (Harris, 1968) can be used as a theoretical foundation for unsupervised methods of WSD. It states that “meaning of entities (...) is related to the restrictions on combinations of these entities relative to other entities.”. In this context entities can be understood as words or lemmas.

Unsupervised approaches can be divided into two categories: Word Sense Induction (WSI) and word sense discrimination (Pedersen, 2006; Agirre and Soroa, 2007). WSI is concerned with automatic building of sense inventories, typically by clustering of words. There are many approaches to WSI, e.g., classical ones include Latent Semantic Analysis (Landauer and Dumais, 1997), Hyperspace Analogue to Language (Lund and Burgess, 1996) or Clustering by Committee (Pantel, 2003) recently adapted to Polish (Broda et al., to appear). On the other hand, word sense discrimination focuses on splitting text snippets with ambiguous word into clusters, where each cluster contains text snippets with only one sense of an ambiguous word. Most well-known approaches include Context Group Discrimination (Schütze, 1998) and SenseCluster (Pedersen, to appear).\textsuperscript{3}

Unsupervised approaches to WSD have some important drawbacks. Accuracy of those systems is usually lower compared to the systems based on supervised learning or even most frequent sense baseline (Agirre and Edmonds, 2006). Resulting clusters, for both sense induction and sense discrimination systems, are typically hard to categorize. There are no descriptive labels for automatically created groups of related words or groups of text snippets (Pedersen, to appear). Also, evaluation and comparison of such systems is not an easy task—manual evaluation is very hard and automatic methods are indirect and counter-intuitive (Pantel, 2003; Broda et al., to appear).

\textsuperscript{1}There is a plethora of other approaches to WSD, e.g., based on translational equivalence or hand-written rules. We omit those for brevity. For extensive overview of other methods see, e.g., Agirre and Edmonds (2006); Navigli (2009).\textsuperscript{2}We will use term context to denote a passage of text containing ambiguous word.\textsuperscript{3}The latter can be also used for WSI.
Recently, an algorithm based on weakly-supervised learning for WSD called Lexicographer-Controlled Semi-automatic Sense Disambiguation (LexCSD) was proposed (Broda and Piasecki, 2009). It has a potential of overcoming some of the problems of both supervised and unsupervised methods. It requires only a handful of manually disambiguated examples and training is performed on unlabelled data. Thus there is no need for creation of large semantically disambiguated corpus and domain adaptation is not an issue. Output of LexCSD is not hard to understand: algorithm provides sense-usage examples for each cluster and also remembers labels provided by Oracle\:^4\ which can be mapped to the resulting clusters. By providing this mapping, performance of LexCSD can be measured using methods used by supervised learning, i.e., on manually annotated corpora. In (Broda and Piasecki, 2009) LexCSD accuracy was comparable to supervised methods, but the drawback was coverage. Some infrequent senses were omitted by the algorithm, because LexCSD can abstain from making decisions when there is not enough evidence.

LexCSD was only tested on a small manually semantically disambiguated corpus of Polish, containing only 1348 text snippets for 13 ambiguous words. Sense inventory was taken from the early version of Polish wordnet (Piasecki et al., 2009) and there was only one annotator. Thus the aim of this work is twofold: we want to create more robust version of semantically annotated corpus for evaluation and using this corpus we want to evaluate LexCSD on large scale.

This paper is organised as follows. Next section describes a work on creation of semantically annotated corpus. Next, LexCSD is described in details. Section 4 discusses methodology of experimental evaluation and results of experiments. The paper is finished with conclusion and pointing direction of further works.

2 Development of Semantically Annotated Test Corpus

During our previous work (Baś et al., 2008) we have developed a small semantically annotated corpus, which was based on parts of IPI PAN Corpus (IPIC) (Przepiórkowski, 2004). This corpus (Test Corpus, TC) was annotated with senses from an early stage of development of the Polish wordnet (plWordNet) (Piasecki et al., 2009) by one annotator, who was not a professional linguist. We have chosen the same 13 different base forms, corresponding to several polysemous lexemes and homonyms, for creation of new sense inventory with wider coverage of meanings and annotated the corpus again. The annotation process was performed by two native speakers of Polish: a professional linguist and a computational linguist.

The number of senses increased compared to our previous work and varies now from 3 to 14. Senses were more precisely differentiated. The chosen words represent the variety of different problems for WSD; some of the senses have homonymous character, i.e., they represent separate homonyms of the same morphological base form. Homonyms are important as they express different meanings which can be problematic for statistical NLP applications, as it introduces large amount of noise. We tried to compose the set of words that would encompass intuitively less or more difficult problems for possible WSD algorithms.

\(^4\)Oracle is usually a user working with the algorithm. The user is not strictly required. F.e., one can develop a way of mapping sense usage example found by LexCSD to dictionary entries.
Senses were distinguished in three steps: firstly, we constructed a list of words’ meanings using the best available Polish dictionaries (Dubisz, 2003; Doroszewski, 1958-69). Then we supplemented it with specialist or colloquial senses from Wiki- tionary.pl and Wikipedia.pl. Finally, we compared the list with meanings from the newest version of plWordNet to formulate accurate distinctions and definitions of contemporary Polish words. plWordNet has already reached the size of 27 000 lexical units and is publicly available for research (Piasecki et al., 2009).

TC consists of literature works, press articles and news, scientific works and legal texts. The special attention was paid to avoid taking all examples for particular sense from the same source text. Nevertheless, the sense frequencies in TC are still not fully balanced. For some senses we could find only a few examples in the whole IPIC. The examples for different words were chosen to represent optimum semantic contexts. The number of usages, however, tended to be balanced, so even senses which are rare in IPIC have their representation. It was pretty difficult to find enough occurrences, see Table 1.

As we used previously selected examples of usages, approximately a third part of senses did not have any representation in TC. All existing examples received the annotation tags on the newly constructed annotation editor\(^5\). The selected set of meanings includes:\(^6\)

- **agent [8]**: ‘a person who represents a company or firm’, ‘agent, a person who represents an actor, artist, writer or sportsman’, ‘secret police agent’, ‘intelligence agent, spy’, ‘bodygourd’, coll. *‘amazing guy’, chem. *‘agent, a particular kind of substance’, *‘agent in programming’;
- **dziób [6]**: ‘beak’, ‘hard pointed part of an object’, ‘bow, nose, front part of a boat, ship, plane, helicopter etc.’, informal ‘mouth, face (semantically marked)’, mus. *‘mouthpiece, woodwind’, *‘scar on face after disease, especially after smallpox’;
- **język [7]**: ‘tongue’, *(natural) language’, *‘means of non-verbal communication, e.g. body language’, fig. *‘a piece of land, wood, lake, natural landscape etc. that resembles a tongue’, *‘a piece of a device that resembles a tongue’, ‘source of information’, *‘artificial language’;
- **linia [14]**: ‘line, a long, straight real or imaginary curve on surface or in space’,

\(^5\)The editor and corpus browser are available online at \url{http://nlp.pwr.wroc.pl/webann}. Edition functionality is limited to registered users only.

\(^6\)Senses that did not appear in TC are marked with asterisks.
‘line, route’, ‘edge, imaginary line separating two areas’, ‘power line’, ‘assembly line’, ‘line, a connection to a telephone system’, *‘line, row’, ‘a row of positions used to fight with enemy in sport competition’, ‘line, approach to subject, a way of dealing with or thinking about something or someone’, *‘lineage’, *‘contour’, ‘figure, the shape of a person’, *‘ruler’, military ‘line, a row of positions used to defend against enemy attack’, ‘line, a range of similar things that are for sale’, *‘line, a row of characters as a unit of organization within text files’, *‘line, a straight curve in geometry’, *‘credit line’;

- pole [11]: ‘field (agricultural)’, ‘area, a particular part of a place, piece of land or country’, ‘playing field, an area, usually covered with grass, used for playing sport’, ‘area, part of a surface, surrounded by real or imaginary borders’, *(in medicine) a group of neural cells that constitutes a particular part of brain’, *‘physical field (e.g. electromagnetic)’, *‘area in geometry’, *‘semantic field (in linguistics)’, *‘field, an area of activity or interest’, regional ‘place outside a building’, *‘field, collection of similar information on a computer’;

- policja [3]: ‘police (organization)’, ‘police station’, ‘policemen’;

- powód [3]: ‘reason’, ‘plaintiff, claimant, complainant’, *‘strap (used in horse-riding)’;


- zamek [6]: ‘castle’, ‘lock’, ‘zipper’, ‘breechblock’, *‘trap in hockey’, *‘a part of machine or any device that stops its action’;

- zbiór [7]: ‘set, a group of similar things that belong together in some way’, ‘mathematical set’, ‘collection’ (usually in pl. zbiory), ‘harvest, the crops which are cut and collected’, ‘an act of harvesting’, *‘an exercise book’, *‘file’;


Two linguistic difficulties emerged during our work: collocations and fuzziness of meaning. Collocations appeared to be a real problem. The phraseological units should have been treated as the multiword lexemes, but mostly they are not present in plWordNet. Since nouns in collocations sometimes do not loose their meanings (Vinogradov, 1977) we decided to annotate such word occurrences with their literal meaning. Thus język in ugrzyć się w język (literal ‘to bite one’s own tongue’, ‘quieten, silence oneself, stop saying at a right moment’) is connected with its literal sense ‘tongue’. Fuzziness was another problem. It was somehow difficult to distinguish between meanings such as agent ‘secret police agent’ and ‘intelligence agent, spy’, policja ‘police = institution’ and ‘police = policemen’, ‘police = institution’ and ‘police station’, sztuka ‘dramatic play’ and ‘theatrical performance of a play’, linia ‘contour’ and ‘figure, the shape of a person’, or klasa ‘class (at school): teaching group’, ‘class: a period of time in which students are taught something’.

Annotators discussed above difficulties during training session. After the annotation process we measured inter-annotator agreement using Cohen’s κ (Artstein
Table 1: Annotated corpus statistics and inter-annotator agreement information. There are 1344 annotated examples in corpus and the average agreement is $\kappa = 0.88$.

and Poesio, 2008). The agreement is surprisingly high, 0.88 for whole corpus. It means that definitions have been formulated unambiguously. In case of disagreements we used annotations provided by the more experienced linguist in our experiments and in Tab. 1. It is not a coincidence that agreement in cases of fuzzy senses (agent, linia, sztuka, policja) is relatively low. It is reasonable to suppose that systems in such situation would gain worse results. Also, such find grained sense distinctions (cf. Tab. 1) can have negative impact on the WSD algorithms. Nevertheless, we decided on that kind of sense inventory as merging of senses is usually easier then dividing.

3 Lexicographer-Controlled Semi-automatic Sense Disambiguation

To overcome the knowledge acquisition bottleneck we have proposed (Broda and Piasecki, 2009) semi-supervised method for WSD that was inspired by the method often used by lexicographers working on construction of dictionary entries. Corpus-based lexicographer work can be roughly divided onto four steps (Kilgarriff, 2006; Kilgarriff and Koeling, 2003). Linguists begin their work by gathering word usage examples from a corpus. Next, the examples are clustered on the basis of their usage, i.e., examples in one cluster have more in common then in different clusters. The clusters are then analysed in search for a common characteristics of examples in each cluster. The work is finished with formulation of dictionary definitions. According to limited research performed by Kilgarriff (1997) the last step is the hardest part of a linguist work. “The second hardest part is splitting” (Kilgarriff, 1997), i.e., the step of formulating clusters.

Those four steps were direct inspiration for proposed algorithm. The metod starts with gathering examples, which are clustered in the next step. Following step
involves construction of classifiers—this step can be seen as an computational way of analysing the clusters, especially if some rule-based classifier is used. Instead of formulation of definition the sense-usage examples are given. After the training phase, LexCSD can be used to disambiguate previously unseen text.

3.1 Gathering of word usage examples

The first step of the algorithm is relatively easy to perform by the machine. The occurrences of ambiguous words together with the surrounding contexts can be retrieved automatically from text corpora with little effort. Raw text snippets are not helpful, thus some kind of feature extraction has to be employed. We need to convert the text into vectors of numerical values, which can be used by machine learning algorithms.

There are many ways for performing this step, cf. (Agirre and Edmonds, 2006). In the simplest form one can mark occurrences of a given word (or phrase) as a feature. A little bit more complex approaches involve morphosyntactic analysis and looking for, e.g., sequences of part-of-speech tags. One can look for even more complex dependencies within a given text snippet, e.g., complex morpho-syntactic relations between words. On the other hand, not only the feature type is important, but also the context size. Some words can be semantically disambiguated by looking only on a very narrow context, e.g., zamek in the meaning of zipper can be often disambiguated by the occurrence of błyskawiczny.\footnote{Zipper is usually translated as zamek błyskawiczny.} This follows a one sense per collocation heuristics proposed by Yarowsky (1993). For many others words only looking at a wider context, i.e., whole sentence, whole paragraph or even whole document, is helpful for disambiguation. The wider context captures also many unrelated linguistic phenomena for disambiguation task, which results in introduction of noise into machine learning algorithms. Usually there is no way of determining how big a context should be in advance.

In contrast to the previous work on LexCSD (Broda and Piasecki, 2009) and other work done recently for Polish (Młodzki and Przepiórkowski, 2009; Baś et al., 2008) we will focus only on one type of features and one size of a context. Namely, we will use only the simplest features, i.e., occurrence of a (lemma, flex class, frequency) triples in a context of \pm 20 segments (tokens).\footnote{A segment (token) is defined as word, words separators, but some words can be split onto several segments. For discussion see (Przepiórkowski, 2006).} There are a few reasons supporting this decision. The features are encoded as bag-of-words in a (sparse) vectors in high dimensional feature space. First, having only one type of feature simplifies the reasoning of relative performance of different algorithms. Second, this type of encoding does not require using different encoding schemes for different classification algorithms and the required binarization of feature vectors is trivial. Last but not least, this type of features is frequently used for building more complex feature spaces that includes them. Also worth noting are the words of Agirre and Stevenson (2006): “(...) co-occurrence vectors provide full coverage without scarifying that much precision.”
3.2 Clustering

The clustering step corresponds to the second step of lexicographer work, i.e., splitting of word-usage examples into distinctive groups. This is very important step, because labelled clusters will be used as input data for training the classifiers in later steps of the algorithm. Ideally, each cluster of text snippets will represent different usage pattern of a word, which will correspond to different meaning of a word. Obviously, this assumption is not strictly needed, as we can refine clustering results by filtering clusters so that only text snippets that are close to cluster core are used. Also, because of the statistical nature of the clustering algorithms we do not expect that all the senses will form its own clusters. Infrequent senses will usually be wrongly assigned to other clusters or treated as outliers.

Another problem is determining the number of clusters in an automatic way. A few approaches to this problem were proposed, e.g., based on gap statistic (Pedersen and Kulkarni, 2006). On the other hand, we can employ existing language resources (dictionaries, wordnets) for determining the number of different meanings of a word. Both approaches are supported by LexCSD, but in this work we will focus on the second one. This will enable fair comparison with supervised approach using manually annotated corpus with words taken from Polish WordNet called plWordNet (Piasecki et al., 2009). We set the number of clusters to two times the amount of senses in plWordNet, because clustering algorithms tends to find different patterns of usage for a few most frequent senses of a word and there is not enough examples of text snippets for infrequent senses to form clusters.

After forming the clusters we should label them with the appropriate plWordNet senses in order to test LexCSD on the manually annotated corpus. For reducing the workload each cluster is labelled with only one representative text snippet as an example. We will use a very simple approach for selection of the representative example, i.e., we will use the cluster centroid.

Aside from enabling straightforward evaluation procedure, the labels provide also a possibility of merging clusters that describe the same word sense in some step of LexCSD. They can also be used as short explanations of clusters’ content—many unsupervised and weakly-supervised algorithms for WSD lacks this property (Pedersen, to appear; Broda et al., to appear).

3.3 Classification

Classification step roughly corresponds to the last step of the linguist work: analysis of clusters. Every labeled cluster is treated as a collection of training examples for one class. In the previous step we have filtered some text snippets. Some clustering algorithms can also remove some text snippets as outliers. We treat all those rejected contexts as a distinct class of uncertain examples. This enable a classifier to abstain from making a decision in this step of an algorithm. The outlier class will be an input to another iteration of the algorithm.

LexCSD is not tied to any specific classification scheme, but obviously this

---

9 Clusters can also be assigned to the outliers class or left unlabelled.

10 This and other feedback loops (Broda and Piasecki, 2009) are not tested in this article for brevity.
choice will impact the performance of the whole system greatly. Note, that when choosing a classifier one should take into account what features were extracted from the corpora. On the basis of the performance examination of different approaches to the classification we have chosen algorithms based on two different paradigms of learning, namely probabilistic (Naive Bayes) and discriminative (C4.5).

The classification step ends with classifiers which are ready to be used. They can be used in the same way as trained classifiers form a supervised WSD algorithm to disambiguate unseen occurrences of ambiguous words.

4 Experiments

We used three corpora for training: IPI PAN Corpus (Przepiórkowski, 2004), electronic edition of Rzeczpospolita newspaper (Weiss, 2008) and a corpus of large documents collected from the Internet. The joint corpus contains roughly 570 million tokens. Trained LexCSD is evaluated on the basis of the manually disambiguated corpus (TC, Sec. 2). Parts of IPI PAN corpus are included in TC, so we removed them from the training data. This can have potentially negative impact, because most occurrences of some infrequent senses were included in TC.

For the clustering step we have used the graph partition algorithm (Karypis, 2002). We have tested two cluster filtering schemes. The first is based on the assumption that text snippets which are closer to the cluster centroid are more informative than those located far away. Having too much data (thousands or even hundred of thousands) of elements in clusters is undesirable for efficiency reasons, but also to few examples may prove problematic. As it was mentioned earlier, one of the groups contains only outliers. This group will be treated as a source of negative examples. The proportion of positive to negative examples is also important from the machine learning perspective. We decided that there should be at least 100 examples for the negative class (of outliers) and twice as much examples for every other class.\footnote{This choice can have impact on the performance, but as LexCSD has a potential to automatically tune its parameters via usage of the feedback loops. We leave this problem for further research. See (Broda and Piasecki, 2009) for more discussion.}

The second filtering scheme was introduced after the initial interaction with the system. Both the text snippets shown to the Oracle and manual inspection of the formed clusters showed that there are many identical text snippets present in corpus. This can have negative impact on clustering and classification phases. Thus we will also test removal of the identical contexts from the training phase.

We will use the following measures for evaluation, i.e., the precision of \(i\)-th sense \(P_i\) which describes how many times the algorithm made a right choice \(P_i = \frac{h_i}{h_i + m_i}\) and the coverage for \(i\)-th sense \(C_i = \frac{h_i + m_i}{h_i + m_i + s_i}\), where \(h_i\) is the number of hits for the \(i\)th sense, \(m_i\)—the number of misses for the \(i\)th senses and \(s_i\) is the number of times the algorithm abstained from making a choice. For measuring the performance on the whole set of senses we use weighted average version of the precision \(P_w = \frac{\sum_i P_i \cdot (h_i + m_i)}{\sum_i (h_i + m_i)}\) and coverage \(C_w = \frac{\sum_i C_i \cdot (h_i + m_i + s_i)}{\sum_i (h_i + m_i + s_i)}\).

As a first step in our experiments, we want to assess a performance of the supervised algorithms on newly annotated TC. We tested two algorithms—Naive
Bayes and C4.5 decision trees—with and without attribute selection. The attribute selection was performed on the basis of the best first search criterion (Hall, 1998). Table 2 reports result obtained in the leave-one-out cross validation in the supervised settings. Surprisingly, Naive Bayes works better without attribute selection (on average is 6.54% better) and C4.5, behave a little better with attribute selection turned on (0.85% better on average).

The results are much lower then the work of Baś et al. (2008) and a little bit lower then the work of Młodzki and Przepiórkowski (2009). The most probable reason for this is the change in annotations and enlargement of sense inventory. Another one is the problem with overfitting during feature selection scheme used in the first cited work. The most visible differences in comparison to Baś et al. (2008) are for the following words: *klasa*, *linia*, *pole*, *policja*. First two of those have now very fine grained sense distinction, but sense distribution of the following two has changed significantly (see Sec. 2 for more discussion). Compared to Most Frequent sense Baseline (MFB, a heuristic classifier that chooses always the most frequent sense) the obtained results are satisfactory. Naive Bayes without feature selection beats the MFB for all cases. Other algorithms had only trouble with one word: *sztuka* for C4.5 without attribute selection and *policja* for both algorithms with feature selection. It seems that attribute selection applied removes some very important features for disambiguation. Problems with those two words are also consistent with observations made in Sec. 2.

Next series of experiments were aimed at the assessing performance of LexCSD trained on large unannotated corpora. Table 3 presents performance of the algorithm using Naive Bayes for classification and Table 4 presents performance using C4.5. Those results are lower, but consistent with work on LexCSD in the toy experiment of (Broda and Piasecki, 2009). In the current work weakly supervised

<table>
<thead>
<tr>
<th>word</th>
<th>MFB [%]</th>
<th>No Attribute Selection</th>
<th>Attribute Selection</th>
</tr>
</thead>
<tbody>
<tr>
<td>agent</td>
<td>67.14</td>
<td>70</td>
<td>71.43</td>
</tr>
<tr>
<td>automat</td>
<td>43.81</td>
<td>87.62</td>
<td>71.43</td>
</tr>
<tr>
<td>dziób</td>
<td>38.27</td>
<td>83.95</td>
<td>72.84</td>
</tr>
<tr>
<td>język</td>
<td>65.33</td>
<td>76</td>
<td>78.67</td>
</tr>
<tr>
<td>klasa</td>
<td>26.05</td>
<td>68.91</td>
<td>43.7</td>
</tr>
<tr>
<td>linia</td>
<td>25.93</td>
<td>40.74</td>
<td>37.04</td>
</tr>
<tr>
<td>pole</td>
<td>47.92</td>
<td>69.79</td>
<td>77.08</td>
</tr>
<tr>
<td>policja</td>
<td>39.06</td>
<td>54.69</td>
<td>37.5</td>
</tr>
<tr>
<td>powód</td>
<td>52.71</td>
<td>86.05</td>
<td>85.66</td>
</tr>
<tr>
<td>sztuka</td>
<td>43.16</td>
<td>54.74</td>
<td>52.63</td>
</tr>
<tr>
<td>zamek</td>
<td>39.13</td>
<td>67.39</td>
<td>70.65</td>
</tr>
<tr>
<td>zbiór</td>
<td>36.78</td>
<td>73.56</td>
<td>48.28</td>
</tr>
<tr>
<td>zespół</td>
<td>47.93</td>
<td>75.21</td>
<td>60.33</td>
</tr>
<tr>
<td>weighted average</td>
<td>72.42</td>
<td>64.66</td>
<td>65.88</td>
</tr>
</tbody>
</table>

Table 2: Precision of disambiguation in supervised settings.
algorithm beats the MFB for 9 of 13 words for Naive Bayes (8 for C4.5). Also, in a few cases the algorithm beats the supervised counterparts in terms of precision (marked in bold in Tab. 3 and Tab. 4). The semi-supervised nature of LexCSD has its cost in terms of coverage, which is quite limited for the majority of cases.

There are a few problematic words that lower overall performance of LexCSD,
namely: język, policja, powód, sztuka and zespół. The first three have one very
dominant sense in the joint corpus — the clustering phase of the algorithm finds
only this dominant sense. Manual inspection of the resulting clusters also con-
firmed those observations—it was very hard to find examples of other senses in the
data. Sztuka exposed limitations of using only co-occurrence features during the
experiments. Even if the clustering phase found 4 different meanings, the lexical
features were not powerful enough for discrimination. This observation is further
confirmed by relatively low accuracy for this word in the supervised settings. On
the other hand, zespół has very different distribution of senses in TC, especially
with respect to three meanings: team (excluding sports team and artistic team),
sport’s team and complex (of buildings). Clusters found for those meanings are
very pure, but contains mainly different usage patterns for those senses. Namely,
general team is dominated by special (political) committees (as opposed to in-
formal teams of peoples in TC) and the complex sense is dominated by medical
complexes (as opposed to school complexes in TC). TC also lacks examples for
sport team, which are dominant in the joint corpus.

On the other hand interesting results were obtained for linia. This word is
usually cited in the context of WSD as one of the most difficult for disambigu-
ation. Indeed, during the manual annotation process linia was one of the most
problematic word. Also, the baseline and the inter-annotator agreement confirms
this. LexCSD had no problem in beating MFB in majority of cases, and it even
beat the supervised algorithms in one case and achieved also quite good cover-
age. Inspection of the detailed output of the classifier and resulting decision tree
revealed that the algorithm was perfect ($P = 100\%$) for assembly line and route
senses and almost perfect ($P = 80\%$) for military sense.

Large scale experiment on the joint corpus provided a few observations regard-
ing the manual labelling phase of LexCSD. Selected contexts for labeling by the
algorithm were easy to disambiguate in most cases. Nevertheless, in a few cases
we stumbled upon two problems. First, for clusters of sztuka in item sense, the
text snippets corresponding to cluster centroid were part of tables in many cases.
Secondly, the algorithm created group for specific senses of a pole. A name of a per-
son (Marek Pol) was incorrectly morphosyntactically disambiguated and formed a
group. To mitigate the first problem, we can introduce some stylistic clues during
the selection of snippets for labelling. The second problem should be solved during
the preprocessing stage. Nonetheless, during semantic disambiguation we can use
additional language processing tools for this purpose, like named entity recognizer.
In current experiments we assigned those cluster to outliers class.

5 Conclusions and further works

We have presented an evaluation of a semi-supervised approach to Word Sense
Disambiguation called LexCSD. The design of the method was inspired by lex-
icographer’s work flow. The method creates training examples using unlabelled
data by the means of clustering. This data is then used for training classifiers.
Main method of operating LexCSD is performed in weakly supervised settings in
a spirit of Active Learning paradigm in which an Oracle is consulted to label the
extracted senses. We have chosen a very simple approach for selection of examples for labelling, but we are planning to use more elaborated approach like one proposed by Kilgarriff et al. (2008).

For the need of evaluation we improved manually disambiguated corpus. New sense inventory, containing more fine grained sense distinction was created. Whole corpus was annotated by two annotators. The inter-annotator agreement was very high for the whole corpus with Cohen’s $\kappa = 0.88$ on average.

Results obtained while training the method on large untagged corpora are promising. The method beat most frequent sense baseline in majority of cases, and even achieved better precision then supervised equivalents for a few words. There are few open problems which we need to overcome before LexCSD will be applicable on a practical scale. We need to improve coverage. This can be done by introduction of additional feedback loops as discussed by Broda and Piasecki (2009). We need to focus on finding infrequent senses during clustering. Investigation in a method of automatic finding the right number of cluster is needed.

Removing identical contexts during the filtering phase will not remove contexts that are very similar. Those contexts introduce unnecessary bias in sense distribution. In preliminary experiments performed for three words (dziób, linia and zamek) with removal of context based on the string distance measure the algorithm has achieved promising results - additional (infrequent) senses were found during the clustering step.

References


W. Doroszewski, editor (1958-69), Słownik języka polskiego, volume 1-11, PWN.

S. Dubisz, editor (2003), Uniwersalny słownik języka polskiego, volume 1-6, PWN.


G. Karypis (2002), CLUTO a clustering toolkit, Tech. report, Univ. of Minnesota.


R. Mihalcea (2003), The Role of Non-Ambiguous Words in Natural Language Disambiguation, in *Proceedings of the Fourth RANLP*.


Maciej Piasecki, Stan Szpakowicz, and Bartosz Broda (2009), *A wordnet from the ground up*, Oficyna wydawnicza Politechniki Wrocławskiej.

Adam Przepiórkowski (2004), *The IPI PAN corpus: Preliminary version*, Institute of Computer Science PAS.


V. Vinogradov (1977), Basic concepts of Russian phraseology as a linguistic subject, in V. Vinogradov, editor, *Selected Works: Lexicology and Lexicography*, Nauka.


David Yarowsky (1993), One sense per collocation, in *Proceedings of the workshop on Human Language Technology*, pp. 266–271, ACL.