Control and Cybernetics

vol. **40** (2011) No. 2

Evaluating lexicographer controlled semi-automatic word sense disambiguation method in a large scale experiment*†

by

Bartosz Broda and Maciej Piasecki

Institute of Informatics, Wrocław University of Technology, Poland

Abstract: Word Sense Disambiguation in text remains a difficult problem as the best supervised methods require laborious and costly manual preparation of training data. On the other hand, the unsupervised methods yield significantly lower precision and produce results that are not satisfying for many applications. 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 volume of untagged text. A comparison showed that the approach is better than most frequent sense baseline in most cases.

Keywords: natural language processing, word sense disambiguation, semi-supervised machine learning.

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 than one meaning, but usually only one of them is active in a given context. For example, an electronic thesaurus called WordNet (Fellbaum et al., 1998) has 36 entries for *line*. WSD is a 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 a robust WSD system helps in choosing the correct translation across contexts. Also information retrieval, information extraction, text mining or computer-aided lexicography could benefit from a high quality WSD system (Agirre and Edmonds, 2006).

*Work financed by Innovative Economy Programme project POIG.01.01.02-14-013/09.
†Submitted: October 2010; Accepted: June 2011.

WSD is not an easy problem to solve, partly 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 means to enumerate all different word senses. In WSD a set of senses is called *a sense inventory*.

There are two main approaches to WSD based on machine learning: supervised and unsupervised (Agirre and Edmonds, 2006).[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 construction of the supervised WSD system. Some features are extracted from those text snippets (or contexts[2]) and classifiers are trained using these manually labeled data. Usually, supervised approaches are superior to unsupervised in terms of precision of automatic disambiguation when used on the same type of texts that the systems were trained on.

There are two main drawbacks of the supervised approaches to WSD: building manually annotated datasets for learning and domain adaptation. Building manually disambiguated corpora is a 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 the 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 Dumains, 1997), Hyperspace Analogue to Language (Lund and Burgess, 1996) or Clustering by Committee (Pantel, 2003) recently adapted to Polish (Broda et al., 2010b). On the other hand, word sense discrimination focuses on splitting text snippets with an ambiguous word into clusters, where each cluster contains text snippets with only one sense of the ambiguous word. Most well-known approaches include Context Group Discrimination (Schütze, 1998) and SenseCluster (Pedersen, 2010)[3].

[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).

[2] We will use term *context* to denote a passage of text containing ambiguous word.

[3] The latter can be also used for WSI.

Unsupervised approaches to WSD have some important drawbacks. Precision of those systems is usually lower in comparison 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, 2010). 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., 2010b).

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. LexCSD output is not hard to understand: the 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, LexCSD performance can be measured using methods used by supervised learning, i.e., on manually annotated corpus. In Broda and Piasecki (2009) LexCSD precision was comparable to the 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.

There are many approaches based on weak supervision that share some similarities with LexCSD and were applied to a problem of WSD. They can be divided roughly into two groups: semisupervised (Abney, 2008) and active learning (Settles, 2009). The first group of methods usually starts with small amounts of labeled training data and large amounts of unlabeled data. Next, the labeled data is iteratively extended. In contrast, LexCSD starts with unlabeled data and asks for labels of a few instances. The labelling step is shared with active learning approaches, which focus on iterative process of construction of training set that minimise labelling effort and maximise classification precision. LexCSD takes the active learning component to the extreme, i.e., it only asks for one label per sense.

LexCSD was tested solely on a small manually semantically disambiguated corpus of Polish, containing only 1348 text snippets for 13 ambiguous words. The 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 to evaluate LexCSD on a large scale using more robust version of the semantically annotated corpus. We also want to test the impact of different classification algorithms on LexCSD.

[4]Oracle is usually a user working with the algorithm. The user is not strictly required. For example, one can develop a way of mapping sense usage example found by LexCSD to dictionary entries.

Table 1. Annotated corpus statistics and inter-annotator agreement information. There are 1344 annotated examples in corpus and the average agreement is $\kappa = 0.88$

Word	**No. of senses**	**Annotated senses**	**Examples**	κ
agent	6	1/9/3/47/10	70	0.80
automat	7	1/24/30/4/46	105	0.97
dziób	6	28/13/31/9	81	0.98
język	7	3/23/49	75	0.97
klasa	14	15/6/12/11/14/31/10/8/1/10/1	119	0.80
linia	14	13/3/2/2/4/2/11/13/4/3/1/2/21	81	0.72
pole	11	1/1/23/25/46	96	0.86
policja	3	17/25/22	64	0.73
powód	3	136/122	258	0.98
sztuka	10	12/10/2/11/41/19	95	0.84
zamek	6	18/19/36/19	92	1.00
zbiór	7	32/7/8/31/9	87	0.87
zespół	7	10/4/28/58/1/20	121	0.95

This paper is organised as follows: first, work on construction of the semantically annotated corpus is presented. Next, LexCSD is described in details. Section 4 discusses methodology of experimental evaluation and results of experiments. The paper is finished with conclusion and directions of further works.

2. Sense annotated reference corpus

Unsupervised approaches to WSD do not intrinsically depend on the existence of a corpus annotated with word senses. However, we still need a reference resource of the lexical semantics in order to evaluate the quality of the method and its characteristics. As the reference corpus (henceforth RefCorp) is not intended to be a training corpus, it is not necessary to make it large and we are free from the serious problem of time estimates presented by Mihalcea (2003). Nevertheless, a RefCorp must represent a good coverage of different problem types, i.e. lemmas with different numbers of senses and types of ambiguity. The latter we understood as levels of difficulty in distinguishing between particular senses of a lemma[5]. The simplest problem pose homonymous lemmas like *bank*, the worst seems to be lemmas with many senses delimited in a fuzzy way where among them many can represent a kind of metaphor, e.g. *line*. Thus, the corpus construction had to be rather started with selecting a test set of lemmas that are representative in relation to different types of ambiguity, than with collecting texts.

We selected 13 semantically ambiguous lemmas as a basis for the corpus construction. The selected lemmas represent the whole spectrum of different semantic ambiguity types. On the list one can find a prototypical Polish example

[5]By the notion of the *lemma sense* we refer to a sense represented by one of the word forms corresponding to this lemma while used in some context.

of a homonym, i.e. *zamek* – with 6 different meanings from a different semantic field each (see below) – but also highly polysemous *linia* or *agent* with many senses that are not sharply separable. Thus, we expected that the selected lemmas should represent a spectrum of less and more difficult problems for WSD. The initial list was defined in Baś et al. (2008) and later revised in Broda et al. (2010a). All senses were taken from plWordNet – a Polish wordnet (Piasecki et al., 2009). First, in Baś et al. (2008), we followed thoroughly the initial, draft version of plWordNet, later, the senses were updated to the version 1.0 of plWordNet and compared with several dictionaries and crowd sourced knowledge resources, e.g. Wikipedia, searching for specialist, colloquial or new senses in the last case. Finally, we corrected both: the sense list for the selected lemmas as well as senses of those lemmas that were described in the newer version of plWordNet, namely 1.1 (Broda et al., 2010a). All identified senses were described with definitions expressed in the natural language. The selected lemmas together with their short definitions are presented below (senses that are not covered by the collected RefCorp, due to the revision of the sense list, are marked by an asterisk):

- *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', 'bodyguard', coll. *'amazing guy, chem. *'agent, a particular kind of substance', *'agent in programming';
- *automat* [7]: 'automaton, machine', 'a coin-operated automatic machine', *'electric washing machine', 'automaton, a person who acts like a machine, without thinking or feeling', 'telephone in a telephone booth', 'submachine gun', *'automatic transmission car';
- *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';
- *klasa* [15]: 'category, type', 'class, rank', *'travel class', 'class (at school): teaching group', 'class: a period of time in which students are taught something', 'classroom', 'mathematical, logical category in set theory', 'savoir-vivre', 'class, a layer of social stratification', *'special subject at art school', *'class: a taxonomic rank in biology', plurale tantum *'hopscotch', 'class in programming', 'a division in football league system', klasa! 'excellent!';
- *linia* [14]: 'line, a long, straight real or imaginary curve on surface or in space', '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 oppose the rival team in sport com-

petition', '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 of a building', *'field, collection of similar information in a computer';
- *policja* [3]: 'police (organization)', 'police station', 'policeman, members of police';
- *powód* [3]: 'reason', 'plaintiff, claimant, complainant', *'strap (used in horse-riding)';
- *sztuka* [10]: 'art', 'craftsmanship', *'act of craftsmanship', 'item, piece of something', 'a beautiful girl', 'person, individual', 'dramatic play', 'theatrical performance of a play', *'an amount of fabric (for example wool), bale', *'a piece of meat';
- *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';
- *zespół* [7]: 'team', 'band, ensemble, performance group', 'group of machines', 'complex (of buildings)', 'syndrome', *'sport team', botanical 'association'.

RefCorp has been collected on the basis of the initial sense list defined in Baś et al. (2008). Each sense from that list that was not marked with a star was assigned in the corpus at least one text snippet of about 100 surrounding words, in which it occurs (i.e. the corresponding lemma is used in the given sense). For each sense we tried to select examples that represent a textual context, which is typical for the given sense. We tended to obtain equal number of examples per each sense, but due to the significant differences in their distributions it was difficult. We tried to achieve a kind of balance in the number of examples, see Table 1, and for the majority of senses there are at least several examples. Nevertheless, the sense frequencies in RefCorp are not fully balanced. For some senses we could find only a few examples or none at all. Of all 101 senses only

72 were found. Among those 72 senses 8 occur in RefCorp only once, 13 with frequency 2–5, 27 with frequency 6–19 and 24 occur in the corpus at least 20 times. Average frequency of a sense is 18.7.

Initially, we wanted to construct RefCorp exclusively on the basis of the IPI PAN Corpus (a freely available large, general corpus of Polish) (Przepiórkowski, 2004), but it was not possible to find examples in this corpus for many of the selected senses. In these cases Internet was used as a supplementary source of examples. RefCorp consists of literature works, press articles and news, scientific works and legal texts. The special attention was paid to avoid taking all examples for a particular sense from the same source text and genre.

For the needs of the work presented in Broda et al. (2010a), RefCorp was re-annotated using the updated sense list presented above. As the updated list expresses for some lemmas more fine-grained sense distinctions, not all new senses could be found in RefCorp. The corpus was annotated independently by two annotators: a professional linguist and a computational linguist, see Broda et al. (2010). All differences in annotation decisions were discussed with a slight tendency to give the priority to the decisions made by the professional linguist.

Collocations and fuzziness of some meanings posed the biggest problems during annotation. Collocations, especially semantically non-compositional collocations, should be treated as separate lemmas with the senses of their own. However, only a limited number of multi-word lemmas were represented in plWordNet and there were limited means of recognising collocations for Polish, e.g., there is no large dictionary of collocations. Thus, we decided to describe the literal meaning of lemmas occurring in text as constituents of a collocation, see Broda et al. (2010a).

Fuzziness of some senses 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'.

Before the annotation process was started, the annotators had performed a trial annotation session and discussed potential problems. As a result, the achieved inter-annotator agreement, measured using Cohen's κ (Artstein and Poesio, 2008) was high on average. However, several lemmas with lower κ, e.g. *agent*, *policja* or *linia*, correlate with our expectations concerning lemmas that should be more difficult for the WSD method.

3. Lexicographer-controlled semi-automatic sense disambiguation

To overcome the knowledge acquisition bottleneck we have proposed (Broda and Piasecki, 2009) a 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 into 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 than in different clusters. The clusters are then analysed in search for common characteristics of examples in each cluster. The work is finished with the 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 the proposed algorithm. The method starts with gathering examples, which are clustered in the next step. The following step involves construction of classifiers — this step can be seen as computational way of analysing the clusters, especially if some rule-based classifier is used. Instead of formulating a definition, the sense-usage examples are given. After the training phase, LexCSD can be used to disambiguate previously unseen text.

3.1. Gathering word usage examples

The first step of the algorithm is relatively easily performed by the machine. 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 the machine learning algorithms.

There are many ways for performing this step, see 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 at a very narrow context, e.g., *zamek* in the meaning of *zipper* can be often disambiguated by the occurrence of *blyskawiczny*.[6] This follows a *one sense per collocation* heuristics proposed by Yarowsky (1993). For many other words only looking at a wider context, i.e., a whole sentence, whole paragraph or even whole document, is helpful for disambiguation. Unfortunately, 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 in advance how big a context should be.

[6] Zipper is usually translated as *zamek blyskawiczny*.

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 on only one type of features and one size of context. Namely, we will use only the simplest features, i.e., occurrence of a (lemma, flex class, frequency) triples in a context of ±20 segments (tokens).[7] There are a few reasons supporting this decision. The features are encoded as bag-of-words in (sparse) vectors in high dimensional feature space. First, having only one type of feature simplifies the reasoning on relative performance of different algorithms. Second, this type of encoding does not require using different encoding schemata 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 include them. Also worth noting are the words of Agirre and Stevenson (2006): "(...) co-occurrence vectors provide full coverage without scarfing that much precision."

3.2. Clustering

The clustering step corresponds to the second step of lexicographer's work, i.e., splitting of word-usage examples into distinctive groups. This is a 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 a 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 the *cluster core* are used. Also, because of the statistical nature of the clustering algorithms we do not expect that all the senses will form their 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 the 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 number of senses in plWordNet, because clustering algorithms tend to find different patterns of usage for a few most frequent senses of a word and there are not enough examples of text snippets for infrequent senses to form clusters.

[7] A segment (token) is defined as word, words separators, but some words can be split into several segments. For discussion see Przepiórkowski (2006).

After forming the clusters we should label them with the appropriate plWordNet senses in order to test LexCSD on the manually annotated corpus.[8] 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 selecting a 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 cluster content — many unsupervised and weakly-supervised algorithms for WSD lack this property (Pedersen, 2010; Broda et al., 2010b).

3.3. Classification

Classification step roughly corresponds to the last step of the linguist's 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 enables 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.[9]

LexCSD is not tied to any specific classification scheme, but obviously this choice will affect 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.

The classification step ends with classifiers 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

As a first step in our experiments, we wanted to asses the performance of the supervised algorithms on RefCorp. We tested the following algorithms: Decision Tables (DT) (Kohavi, 1995), SVM with linear kernel (Vapnik, 1995), Random Forest (RF) (Breiman, 2001), AdaBoost (with simple decison stumps as weak classifiers) (Freund and Schapire, 1996), k Nearest Neighbours (k-NN) (Aha et al., 1991), decision trees (C4.5) (Quinlan, 1993), and Naive Bayes (NB). As mentioned earlier, we wanted to simplify the analysis by using only simple lexical features. For k-NN, we have tested different values for k (using leave one out cross-validation on the RefCorp) and the 1-NN approach achieved best results. Thus, in the following discussion we will use 1-NN. Table 2 reports results obtained in the leave-one-out cross validation in the supervised settings.

[8] Clusters can also be assigned to the *outliers* class or left unlabelled.

[9] This and other feedback loops (Broda and Piasecki, 2009) are not used in this article.

Table 2. Precision of disambiguation in supervised settings (in %). Last row contains weighted averages.

Word	**MFB**	**DT**	**SVM**	**RF**	**AB**	**1-NN**	**C4.5**	**NB**
agent	67.14	67.14	71.43	68.57	71.43	71.43	65.71	70
automat	43.81	71.43	79.05	69.52	60	55.24	70.48	87.62
dziób	38.27	59.26	88.89	55.56	46.91	40.74	72.84	83.95
język	65.33	77.33	77.33	69.33	65.33	74.67	70.67	76
klasa	26.05	42.86	63.87	60.5	14.29	43.7	52.1	68.91
linia	25.93	37.04	45.68	30.86	37.04	29.63	41.98	40.74
pole	47.92	75	68.75	63.54	68.75	36.46	73.96	69.79
policja	39.06	40.62	48.44	46.88	20.31	35.94	53.12	54.69
powód	52.71	88.37	89.53	86.05	77.13	81.4	81.4	86.05
sztuka	43.16	46.32	54.74	50.53	48.42	28.42	41.05	54.74
zamek	39.13	57.61	70.65	51.09	50	33.7	66.3	67.39
zbiór	36.78	57.47	74.71	63.22	55.17	40.23	48.28	73.56
zespół	47.93	65.29	77.69	67.77	47.93	70.25	69.42	75.21
w. avg.	44.57	64.06	72.92	63.99	53.79	53.5	64.66	72.4

The results are lower than presented by Baś et al. (2008) and a little bit lower than in the work of Młodzki and Przepiórkowski (2009). The most probable reason for this is the change in annotations and enlargement of the sense inventory. Another reason is the problem with overfitting in the feature selection scheme used in the first cited work. The most visible differences in comparison to Baś et al. (2008) were noted for the following words: *klasa*, *linia*, *pole*, *policja*. First two of them have now very fine grained sense distinctions, but sense distribution of the following two has changed significantly. Compared to Most Frequent sense Baseline (MFB, a heuristic classifier that chooses always the most frequent sense) the obtained results are satisfactory.[10] Descision Tables, Support Vector Machines, Random Forest and Naive Bayes are as good or better than MFB for all words. SVM and Naive Bayes achieve the best precision on average.

For the majority of words all tested supervised machine learning algorithms behave as expected, i.e., for highly polysemous words the results are lower. There are two exceptions to this observation. First, *policja* has only three senses, but the results are low. This is caused by the fact that different senses of *policja* are very related and sometimes it is very hard to contextually differentiate among them by humans. *Policja* exhibits one of the lowest agreements between annotators in RefCorp (only *linia* has lower agreement, but merely by 0.01). Second, *zamek* is difficult for many machine learning methods as opposed to humans, i.e., *zamek* has the highest inter-annotator agreement in RefCorp. The

[10]Note that MFB was calculated using whole RefCorp, which is quite small. On the other hand, MFB might suffer from overfitting, which elevated the baseline artificially.

problem with this word might be caused by two factor: simple lexical features are not useful for *zamek* and the size of the window might be too small in some cases.

Next series of experiments were aimed at assessing the performance of LexCSD trained on large unannotated corpora. 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 (RefCorp, Sec. 2). Parts of IPI PAN corpus are included in RefCorp, so we removed them from the training data. This can have potentially negative impact, because most occurrences of some infrequent senses were included in the corpus.

For the clustering step we have used the repeated bisection clustering algorithm using $e1$ criterion function (Karypis, 2002). We have used this approach to clustering as it was shown that in similar settings the algorithm exhibits very good quality of clustering (Broda and Mazur, 2009). We have used two cluster filtering schemata. The first is based on the assumption that text snippets which are closer to the cluster centroid are more informative then those located far away. Having too many data (even up to 10^5) of elements in clusters is undesirable for efficiency reasons, but also too few examples may prove problematic. As it was mentioned earlier, one of 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 many examples for every other class.[11]

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 the corpus. This can have negative impact on clustering and classification phases. Thus, we have also removed identical contexts from the training phase. During this step we also introduced simple heuristic rule which removes contexts containing parts of tables.

We will use the following measures for evaluation: precision of i-th sense P_i describing how many times the algorithm made a right choice, $P_i = \frac{h_i}{h_i+m_i}$, and 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 ith sense, m_i – the number of misses for the ith sense 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 the weighted average versions of precision $P_w = \frac{\sum_i P_i \cdot (h_i+m_i)}{\sum_i h_i+m_i}$ and coverage $C_w = \frac{\sum_i C_i \cdot (h_i+m_i+s_i)}{\sum_i (h_i+m_i+s_i)}$.

[11] This choice can have impact on the performance, but 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.

Table 3 presents precision of the selected classification algorithms in semi-supervised settings. Table 4 presents coverage. Those results are lower, but consistent with the work on LexCSD in the toy experiment of Broda and Piasecki (2009).

There are some problematic words that lower the 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 confirmed 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 four different meanings, the lexical features were not powerful enough for discrimination. This observation is further confirmed by relatively low precision for this word in the supervised settings. On the other hand, *zespół* has very different distribution of senses in RefCorp, 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 and contain mainly different usage patterns for those senses. Namely, general team is dominated by special (political) committees (as opposed to informal teams of peoples in RefCorp) and the *complex* sense is dominated by medical complexes (as opposed to school complexes in RefCorp). RefCorp also lacks examples for *sport team*, which are dominant in the joint corpus.

Interesting results were also obtained for *linia*. This word is usually cited in the context of WSD as one of the most difficult for disambiguation. Indeed, during the manual annotation process *linia* was one of the most problematic words. Also, the baseline and the inter-annotator agreement confirms this. Both supervised and weakly-supervised approaches reflect this observation in the results, which is not surprising. Interestingly, LexCSD using Decision Tables performed better than MFB and achieved also quite good coverage for *linia*. Inspection of the detailed output of the classifier revealed that the algorithm was perfect ($P = 100\%$) for *communication line*.

Usually, the F-measure is defined in terms of precision and recall. Because in our semi-supervised settings recall is not as important as coverage, we define F_1 measure as: $F_1 = \frac{2P \times C}{P+C}$. This allows for selection of the better performing algorithms using as an indicator both the values of precision and recall. Table 5 summarizes F_1 for all words. The three best performing algorithms are: Random Forest, Decision Tables and Support Vector Machines. It is worth noticing that the most precise algorithm – Naive Bayes – is very selective in terms of coverage. Namely, it selects only the easiest context for disambiguation and abstains from making difficult decisions. This property can be sometimes desirable, but from the practical point of view such an approach to disambiguation is not very useful. This observation clarifies the uncertainty in observations made previously during the work on Polish WSD (Broda and Piasecki, 2009; Broda et al., 2010a). Also, usage of Random Forest solves important problem of coverage mentioned in the previously cited works.

Table 3. Precision of LexCSD using different classification algorithms (in %). Last row contains weighted averages

Word	**MFB**	**DT**	**SVM**	**RF**	**AB**	**1-NN**	**C4.5**	**NB**
agent	67.14	65.71	55.17	43.08	67.14	18.57	30	58.54
automat	43.81	57.58	60.67	46.53	27.62	24.04	60	58.75
dziób	38.27	46.91	51.43	50	45.68	34.57	43.06	65.91
język	65.33	67.57	73.91	70.67	68	69.33	68.57	80
klasa	26.05	50.79	39.8	38.46	21.85	26.89	22.92	43.21
linia	25.93	29.63	18.06	16.22	2.47	2.47	4	9.09
pole	47.92	64.44	69.44	48.89	22.92	2.08	74.12	73.91
policja	39.06	0	35	28	26.56	37.5	0	0
powód	52.71	48.58	38.1	47.64	58.82	47.29	64.79	60
sztuka	43.16	58.82	48.75	40.66	43.16	11.58	48.65	48.72
zamek	39.13	36.47	52.73	38.89	20.65	35.87	29.23	40.62
zbiór	36.78	70.73	76.56	60.81	12.64	18.39	53.66	82.22
zespół	47.93	0	5.71	10.17	16.53	46.28	14.66	12.12
w. avg.	44.57	45.94	48.09	41.66	30.54	30.98	40.85	51.69

Table 4. Coverage of LexCSD using different classification algorithms (in %). Last row denotes weighted average

Word	**DT**	**SVM**	**RF**	**AB**	**1-NN**	**C4.5**	**NB**
agent	100	82.86	92.86	100	100	85.71	58.57
automat	94.29	84.76	96.19	100	99.05	85.71	76.19
dziób	100	86.42	96.3	100	100	88.89	54.32
język	98.67	61.33	100	100	100	46.67	40
klasa	52.94	82.35	98.32	100	100	80.67	68.07
linia	66.67	88.89	91.36	100	100	92.59	81.48
pole	46.88	75	93.75	100	100	88.54	71.88
policja	3.13	31.25	78.12	100	100	3.13	12.5
powód	95.74	16.28	98.45	13.18	100	27.52	21.32
sztuka	17.89	84.21	95.79	100	100	77.89	41.05
zamek	92.39	59.78	97.83	100	100	70.65	34.78
zbiór	47.13	73.56	85.06	100	100	94.25	51.72
zespół	98.35	57.85	97.52	100	100	95.87	27.27
w. avg.	74.18	62.2	95.01	83.33	99.93	68.68	46.35

Table 5. F_1 measure for weighted average precision and coverage for all words using different classification algorithms in LexCSD (in %)

	DT	**SVM**	**RF**	**AB**	**1-NN**	**C4.5**	**NB**
F_1	56.74	54.24	57.92	44.70	47.30	51.23	48.87

Detailed inspection of behaviour of different classifiers leads to interesting observations. Using AdaBoost yields perfect or almost perfect classifier for one sense (precision close to 100%), but very bad classifiers for other senses of a given words. Also, overall bad results of AdaBoost are surprising as the best method (Random Forest) uses similar approach to learning, i.e., using ensembles. Naive Bayes and SVM in most cases built classifiers that were good for two senses only. NB and SVM gave especially interesting results for *zbiór*. Both of them built good classifiers for four senses of the word, but only *zbiór* in the sense of *act of harvesting* was bad ($p = 0$, but with $c = 57.14$ for NB and $c = 42.86$ for SVM), as there were no examples of that sense discovered during clustering phase. Results of C4.5 are surprisingly bad, because induced decision trees where intuitively good. For example, C4.5 found some strong collocations indicating some senses (e.g., *'automat telefoniczny'*, *telephone*). On the other hand, C4.5 failed to notice subtle differences between related senses.

The Most Frequent sense Baseline (MFB) is usually very hard to be beaten by algorithms without supervision (Agirre and Edmonds, 2006). Proposed semi-supervised approach using Random Forest beats the MFB in 6 of 13 cases (8 of 13 when using Decision Tables or SVM). The most precise algorithm, Naive Bayes, beat the MFB in 9 out of 13 cases. Among the top three classification algorithms employed, the sets of words, for which those algorithms beat the MFB are not fully overlapping. This suggests that there might not be a single globally best classification algorithm and other approaches can be more useful. Namely, using ensembles of strong classifiers or introduction of the feedback loops, which were mentioned earlier, can further improve the performance of LexCSD.

A large scale experiment on the joint corpus provided some observations concerning 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. The algorithm created group for specific senses of *pole*. A name of a person (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. During semantic disambiguation we can use additional language processing tools for this purpose, like named entity recognizer. In current experiments we assigned those clusters to the *outliers* class. On the other hand, some specific senses were not found in the clustering phases (e.g., *zamek* in the *zipper* sense) and also in rare cases the clustering phase found some senses that were underrepresented in the RefCorp (e.g., *sports team* sense of a *zespół*).

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 lexicographer's work flow. The method creates training examples using unlabelled data by means of clustering. These data are then used for training classifiers. The main way of operating with LexCSD is ralised in weakly supervised settings in the spirit of the Active Learning paradigm, in which an Oracle is consulted to label the extracted senses. We have chosen a very simple approach to select the examples for labelling, but we plan to use a more elaborate approach like the one proposed by Kilgarriff et al. (2008).

For the needs of evaluation we used an improved version of the manually disambiguated corpus. A new sense inventory, containing more fine grained sense distinctions was created. The 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.

We have tested several classification algorithms, which represent various approaches to machine learning. Support Vector Machines and Naive Bayes are the most precise algorithms in supervised settings. Results obtained during training LexCSD on large untagged corpora are promising. The method beats the most frequent sense baseline in the majority of cases for best classifiers. Among the tested classification algorithms, the Random Forest brings the best balance between precision and coverage, followed by Decision Tables nad Support Vector Machines. On the other hand, Naive Bayes is the most precise algorithm, but it also abstains from making a decision in the majority of cases. By employing Random Forest we have improved the coverage two times in comparison to Naive Bayes.

In the future we will focus on automatic selection of the best classification algorithm for a given word. We will try to accomplish this by introduction of feedback loops as discussed by Broda and Piasecki (2009). On the other hand, we need to focus on finding infrequent senses during clustering. Investigation on a method for automatic finding the right number of cluster is also needed.

References

Abney, S. (2008) *Semisupervised Learning for Computational Linguistics*. Chapman & Hall/CRC.

Agirre, E. and Edmonds, P., eds. (2006) *Word Sense Disambiguation: Algorithms and Applications*. Springer.

Agirre, E. and Soroa, A. (2007) Evaluating word sense induction and discrimination systems. In: *Proc. of the 4th International Workshop on Semantic Evaluations (SemEval-2007)*, Association for Computational Linguistics, 7–12.

Agirre, E. and Stevenson, M. (2006) Knowledge Sources for Word Sense Disambiguation. In: *Word Sense Disambiguation: Algorithms and Applications*. Springer.

Aha, D.W., Kible, D.R. and Albert, M.K. (1991) Instance-based learning algorithms. *Machine Learning*, **6**(1), 37–66.

Artstein, R. and Poesio, M. (2008) Inter-coder agreement for computational linguistics. *Computational Linguistics*, **34**(4), 555–596.

Baś, D., Broda, B. and Piasecki, M. (2008) Towards Word Sense Disambiguation of Polish. In: *3rd Int. Symp. Advances in AI and Applications*. IMCSIT, 73–78.

Breiman, L. (2001) Random forests. *Machine Learning*, **45**(1), 5–32.

Broda, B. and Mazur, W. (2009) Evaluation of Clustering Algorithms for Polish Word Sense Disambiguation. In: *5th Int. Symp. Adv. in AI and Applications*. IEEE, 25–32.

Broda, B. and Piasecki, M. (2009) Semi-supervised Word Sense Disambiguation Based on Weakly Controlled Sense Induction. In: *4rd Int. Symp. Adv. in AI and Applications*. IEEE, 17–24.

Broda, B., Piasecki, M. and Maziarz, M. (2010a) Evaluating LexCSD – a Weakly-Supervised Method on Improved Semantically Annotated Corpus in a Large Scale Experiment. In: *Intelligent Information Systems*. Wydawnictwo Akademii Podlaskiej, Siedlce, 63–76.

Broda, B., Piasecki, M. and Szpakowicz, S. (2010b) Extraction of Polish Noun Senses from Large Corpora by Means of Clustering. *Control and Cybernetics*, **39** (2), 401–420.

Fellbaum, C. et al. (1998) *WordNet: An electronic lexical database*. MIT Press, Cambridge, MA.

Freund, Y. and Schapire, R.E. (1996) Experiments with a New Boosting Algorithm. In: *ICML*, 148–156.

Harris, Z.S. (1968) *Mathematical Structures of Language*. Interscience Publishers, New York.

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

Kilgarriff, A. (1997) The hard parts of lexicography. *International Journal of Lexicography*, **11** (1), 51–54.

Kilgarriff, A. (2006) Word Senses. In: *Word Sense Disambiguation: Algorithms and Applications*. Springer.

Kilgarriff, A., Husák, M., McAdam, K., Rundell, M. and Rychlý, P. (2008) GDEX: Automatically finding good dictionary examples in a corpus. In: *Proceedings of EURALEX*. Universitat Pompeu Fabra, 425–32.

Kilgarriff, A. and Koeling, R. (2003) An Evaluation of a Lexicographer's Workbench Incorporating Word Sense Disambiguation. In: Gelbukh A.F., ed., *CICLing*. **LNCS 2588**, Springer, 225–240.

Kohavi, R. (1995) The power of decision tables. *Machine Learning: ECML-95*, **LNCS 912**, 174–189.

Landauer, T.K. and Dumais, S.T. (1997) A Solution to Plato's Problem: The Latent Semantic Analysis Theory of Acquisition. *Psychological Review*, **104**(2), 211-240.

Lund, K. and Burgess, C. (1996) Producing high-dimensional semantic spaces from lexical co-occurrence. *Behavior Research Methods Instruments and Computers*, **28**(2), 203-208.

Mihalcea, R. (2003) The Role of Non-Ambiguous Words in Natural Language Disambiguation. In: *Proceedings of the Fourth RANLP*. John Benjamins, 357–366.

Mlodzki, R. and Przepiórkowski A. (2009) The WSDDevelopment Environment. In: Vetulani, Z., ed., *Proc. 4rd Language and Technology Conference, Poznań, Poland*. Wydawnictwo Poznańskie, Poznań, 245–250.

Navigli, R. (2009) Word sense disambiguation: A survey. *ACM Comput. Surv.*, **41**(2), 1–69.

Pantel, P. (2003) Clustering by committee. Ph.D. thesis, Edmonton, Alta., Canada, Canada.

Pedersen, T. (2006) Unsupervised Corpus Based Methods for WSD. In: *Word Sense Disambiguation: Algorithms and Applications*. Springer, 133–166.

Pedersen, T. (2010) Computational Approaches to Measuring the Similarity of Short Contexts: A Review of Applications and Methods. *The Computing Research Repository*, abs/0.806.3787.

Pedersen, T. and Kulkarni, A. (2006) Automatic cluster stopping with criterion functions and the Gap Statistic. In: *Proceedings of the Demo Session of NAACL*. ACL, 276–279.

Piasecki, M., Szpakowicz, S. and Broda, B. (2009) *A WordNet from the Ground Up*. Oficyna Wydawnicza Politechniki Wroclawskiej.

Przepiórkowski, A. (2004) *The IPI PAN Corpus: Preliminary version*. Institute of Computer Science PAS, Warsaw.

Przepiórkowski, A. (2006) The Potential of the IPI PAN Corpus. *Poznań Studies in Contemporary Linguistics*, **41**, 31–48.

Quinlan, J.R. (1993) *C4. 5: programs for machine learning*. Morgan Kaufmann.

Schütze, H. (1998) Automatic word sense discrimination. *Computational Linguistics*, **24** (1), 97–123.

Settles, B. (2009) Active Learning Literature Survey. Computer Sciences Technical Report 1648, University of Wisconsin–Madison.

Vapnik, V.N. (1995) *The Nature of Statistical Learning Theory*. Springer Verlag.

Weiss, D. (2008) Korpus Rzeczpospolitej [on-line] http://www.cs.put.poznan.pl/dweiss/rzeczpospolita, corpus of texts from the online edition of Rzeczpospolita.

Yarowsky, D. (1993) One sense per collocation. In: *Proceedings of the workshop on Human Language Technology*. ACL, 266–271.