PRACE FILOLOGICZNE

Warszawa 2012 TOM LXIII

MATEUSZ KOPEĆ, RAFAŁ MŁODZKI, ADAM PRZEPIÓRKOWSKI
Institute of Computer Science
Polish Academy of Sciences

WORD SENSE DISAMBIGUATION IN THE NATIONAL CORPUS OF POLISH

SŁOWA KLUCZOWE: język polski, korpus, ujednoznacznienie, sensy słów, homonimia, algorytm

KEYWORDS: corpus, Polish, disambiguation, word sense, homonymy, algorithm

Abstract

This paper describes the project[1] of annotating the National Corpus of Polish with word senses for a selection of ambiguous lexemes. The WSDDE (Word Sense Disambiguation Development Environment) tool is used for performing experiments with various supervised machine learning techniques and feature sets. The best algorithm is described in detail and evaluated.

1. Introduction

1.1. Word Sense Disambiguation

Word Sense Disambiguation (WSD) is the task of deciding automatically which sense of a given word is used in a given context. It can be illustrated with an example of the following two sentences:

1. I went fishing for some sea *bass*.
2. The *bass* line of the song is too weak.

[1] Research funded by the research and development grant R17 003 03 from the Polish Ministry of Science and Higher Education.

In the first sentence, the intended meaning is a type of fish, while in the second sentence the ambiguous word should be understood as denoting low frequency tones. The aim of a WSD system is straightforward: on the basis of the information from the context, annotate ambiguous words with their appropriate senses. Although this word sense assignment is mostly trivial for the humans, it is a difficult task for a computer.

1.2. Motivation

Word Sense Disambiguation could be useful in many other tasks of the Natural Language Processing, for example:

- machine translation;
- information retrieval;
- question answering.

A machine translation example is simple: it is impossible to translate the Polish word *piłka* into English without knowing the meaning in which it is used. It can be *a ball*, as well as *a small saw*. To perform any semantic processing of texts it is usually necessary to have knowledge about the precise meaning of each word.

1.3. National Corpus of Polish

The National Corpus of Polish[2] (NKJP; Pol. *Narodowy Korpus Języka Polskiego*; NKJP; http://nkjp.pl/) project was carried out between December 2007 and June 2011 and aimed at creating a large reference corpus of Polish. The resulting corpus is based on all previous Polish corpora and consists of over 1.5 billion words, with an approximate 1-million-word subcorpus annotated manually. Word senses are one of the linguistic levels of corpus annotation, apart from (word-level and sentence-level) segmentation, morphosyntax, syntax and named entities. However, not all the words are annotated with their senses, only a set of over 100 most frequent clearly homographic words.

2. Methodology

2.1. Task definition

The WSD task in the National Corpus of Polish was to disambiguate each corpus occurrence of any form of the 106 most frequent ambiguous lexemes with the best accuracy possible. The number of ambiguous words under consideration was limited to only about a hundred because of two main reasons. First, the

[2] More about the National Corpus of Polish in Przepiórkowski et al. 2010, 2011.

techniques which were chosen to disambiguate word senses require a great amount of manually annotated examples of each word's usage, and it is practically impossible to create such a resource for a large number of words. Secondly, an appropriate dictionary of all senses of disambiguated words must be available. Unfortunately, there is no dictionary of Polish which would distinguish word senses at the desirably coarse level.

Because of the fact that the best state-of-the-art WSD systems are based on the so-called supervised machine learning methodology, the obvious choice was to use supervised techniques also for the task at hand. Within the project, appropriate resources were created, i.e., a manually annotated **1 million word subcorpus**, to learn how to disambiguate senses, and a coarse-grained (with an average of **2.85 senses/lexeme**) dictionary of senses for all of the 106 ambiguous lexemes.

2.2. Subtasks

The WSD task in the NKJP was divided into several subtasks. First, there was a need for an Application Programming Interface (API) for the corpus, to facilitate programming access to XML-encoded annotated texts.

Second, a general (and, in principle, language- and corpus-independent) platform for conducting experiments with various machine learning methods was created, called WSD Development Environment (WSDDE; see below for details).

Finally, thousands of experiments were performed, leading to the choice of the best WSD methods for each of the words.

The main challenge was to achieve the best accuracy possible, but with efficiency in sight, as huge amounts of data needed to be processed in the course of annotating the whole corpus.

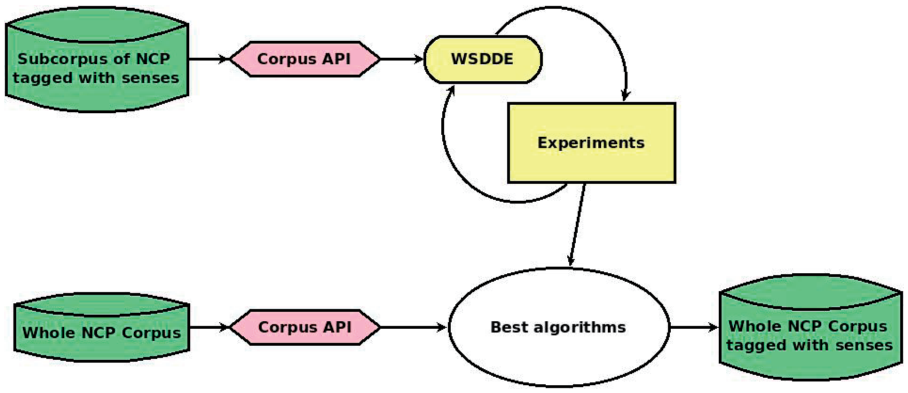
Subcorpus of NCP tagged with senses
Corpus API
WSDDE
Experiments
Whole NCP Corpus
Corpus API
Best algorithms
Whole NCP Corpus tagged with senses

Figure 1. *Data flow.*

The data flow is organized as depicted above: the manually annotated subcorpus of the National Corpus of Polish is accessed by WSDDE via the corpus API, leading to the choice of the best methods disambiguating the word senses. These methods are subsequently used to annotate the whole National Corpus of Polish with word senses, again with the use of the Application Programming Interface.

2.3. Word Sense Disambiguation Development Environment

The Word Sense Disambiguation Development Environment[3] (WSDDE) is a platform for conducting Word Sense Disambiguation experiments and testing the accuracy of various supervised machine learning algorithms.

The main idea of supervised machine learning algorithms is that they allow the computer to automatically learn – from already class-annotated examples – how to classify a new example, whose class is not given. A class in this case is a sense of a word, i.e., different classes are considered for different words. This means that each of the 106 ambiguous words must be treated separately so as to create 106 separate classifiers, each responsible for disambiguating the sense of a particular word in its various contexts.

WSDDE makes it possible to specify the training corpus (for learning) and the test corpus (for evaluation), and to choose which features of the context[4] should be taken into consideration. Also, it is possible to choose feature selection and machine learning algorithms, via mechanisms provided by the WEKA[5] toolkit.

In the task at hand, the appropriately split manually annotated subcorpus of the NKJP served as the training and test corpus.

Various context features were tried with the help of the 4 types of *feature generators* provided by WSDDE; these features can be grouped into:

- thematic features;
- structural features of type I;
- structural features of type II;
- keyword features.

The following subsections briefly describe each feature type.

2.4. Thematic features

Thematic features are intended to provide information about the general topic of the context. They simply indicate whether particular words occur in a window around the disambiguated word (henceforth called the *keyword*).

[3] See Młodzki, Przepiórkowski 2009, as well as http://nlp.ipipan.waw.pl/WSDDE/.
[4] The disambiguated word is considered a part of its context.
[5] See Witten, Frank 2005, as well as http://www.cs.waikato.ac.nz/ml/weka.

Various sizes of the window were tested (up to 25 segments[6]), as well as the possibility of lemmatising words in the context. The last variable was a decision whether to record simply the presence of the word in the context window, or its frequency (the **TF.IDF** measure to be specific).

To give an example of feature vectors generated by various contexts, let us consider the two sentences with the word *bass* given in the introduction of this paper. When we generate features for these two contexts, we get the following feature vectors:

	fishing	**song**	**weak**	**went**	**...**
Context 1	1	0	0	1	...
Context 2	0	1	1	0	...

Number 1 indicates that a particular word is present in the context, 0 – that it is not. Specifically, the first column means that the word *fishing* occurs only in the first context.

2.5. Structural features I

The first set of structural features gives information about the presence of a word, like the thematic features, but only on a given position close to the ambiguous word.

The size of the considered window is smaller (up to 5 segments), because it is unlikely for a long distance fixed positional relation to exist. Again, various window sizes were tested, with or without lemmatization, and recording the presence or the IDF measure associated with the context word. An example presenting structural feature vectors is given below.

	sea-1	**some-2**	**The-1**	**.+1**	**line+1**	**...**
Context 1	1	1	0	1	0	...
Context 2	0	0	1	0	1	...

Number 1 indicates that a particular word is present on a particular position with respect to the keyword, 0 – that it is not. For example, the *sea-1* column indicates that the word *sea* occurs right before the disambiguated word in the first context, but not in the second.

2.6. Structural features II

The second set of structural features works like the first, but instead of looking at word forms, it is concerned with their morphosyntactic interpretations.

[6] *Segments* are roughly words, but, e.g., punctuation marks are treated as separate segments.

Again, various parameters were tested: the size of the window (up to 5 segments), the possibility to use the full ICS PAS Tagset (see, e.g., Przepiórkowski 2004) or only 5 coarse parts of speech, and whether to record the presence or the frequency (the IDF measure) of the words. The following example illustrates this feature type.

	noun-1	**det-1**	**punc+1**	**noun+1**	**...**
Context 1	1	0	1	0	...
Context 2	0	1	0	1	...

The *noun-1* column indicates that a noun occurs immediately before the disambiguated word in the first context, but not in the second.

2.7. Keyword features

The last set of features considered here are the features of keyword, i.e., of the disambiguated word, including its part of speech. Another variable was whether to use additional morphosyntactic levels of annotation, e.g., case, number or gender, as well as whether to consider the information about the capitalisation of the first letter of the word. The example below shows keyword feature vectors for the same contexts as in the previous examples.

	noun	**verb**	**singular**	**capitalized**	**...**
Context 1	1	0	1	0	...
Context 2	1	0	1	0	...

2.8. Feature selection and machine learning

It is easy to imagine that an enormous number of features of the types presented above can be generated. Therefore, a feature selection algorithm must be used to eliminate those features which do not contain much information about the sense of the disambiguated word.

Two algorithms were tested for that purpose, both available in WEKA: InfoGain and CfsSubsetEval (both with various parameters). Once the most important and useful features were chosen, a machine learning algorithm learned from the annotated corpus how to disambiguate the senses of a particular lexeme. Again, various machine learning algorithms made available in WEKE were tested:

- NaiveBayes;
- J48graft;
- VFI;
- Kstar;
- BayesNet;

- DecisionTable;
- RandomForest;
- AdaBoostM1.

3. Results and conclusion

3.1. Statistics

The sense-annotated subcorpus used for training and testing consisted of **1,217,822 segments** in **3,889 texts**. There were **34,186 sense annotations** of 106 different ambiguous lexemes. The *most frequent sense* baseline[7] for this corpus was **78.3%**.

215 experiments were performed for each of the 106 lexemes, which equals 22,790 experiments. These experiments tested various features, feature selection algorithms and machine learning algorithms, separately for each lexeme. The evaluation technique used was the **10-fold cross-validation** method. The result of the best methods at that stage was about **91.48%**, which significantly exceeds the baseline.

Subsequently, 5 best methods for each ambiguous lexeme were selected and evaluated with so-called *leave-one-out cross-validation*, achieving a similar result of **91.46%** and getting more precise information about the best methods. After that step the best single method for each lexeme was chosen, according to the accuracy in that step.

3.2. Best methods

This section describes the most frequent machine learning and feature selection algorithms found among the best methods for particular lexemes, and also the features most frequently selected in these best methods.

3.3. Machine learning

The most popular machine learning algorithm among the best methods is the Naive Bayes classifier, occurring in about **50%** of all best methods. The precise results are:

Machine learning	**Usage count**
NaiveBayes	51
BayesNet	32

[7] The *most frequent sense* baseline gives the accuracy of WSD on the assumption that each ambiguous word is always annotated with the sense which is most frequent in the training corpus.

Machine learning	**Usage count**
KStar	10
J48graft	7
DecisionTable	2
RandomForest	1
Trivial	3
Sum	**106**

What is interesting is that for three lexemes a deterministic classifier – always assigning the same sense – works just fine, as this is the only sense that occurred in the corpus (hence, *Trivial* as the name of this deterministic classifier). These three trivial lexemes are removed from the results below.

3.4. Feature selection

The same kind of statistics can be calculated for feature selection algorithms:

Feature selection	**Usage count**
InfoGain+CfsSubsetEval	60
InfoGain	43
Sum	**103**

The combined method of filtering the most informative features with InfoGain, and then processing subsets of them with CfsSubsetEval, works best. Feature generators generated 2,270 features per word (on average), and filtering via feature selection resulted in only 117 features per word (again, on average). The table below presents the most common 15 features remaining after the filtering step, i.e., used by the best algorithms to disambiguate senses.

Usage count	**Feature**	**Feature type**
31	m3+1	SF2
31	w	TF
31	interp+1	SF2
31	capitalized	KF
32	sg+1	SF2
33	pl+1	SF2
33	prep+1	SF2
33	gen	KF

Usage count	Feature	Feature type
34	się	TF
38	gen+1	SF2
40	noun+1	SF2
40	sg	KF
40	pl	KF
40	prep-1	SF2

After calculating how many times features of 4 types occur, we get the following table:

Feature type	Features chosen	Percentage
TF	7436	60 %
SF1	2026	16 %
SF2	2163	17 %
KF	795	7 %

Thematic features acquire about 60% of the sum, both types of structural features get about 17% each, and key features are responsible for the remaining 7%. It does not necessarily mean that thematic features are the most informative ones. It rather means that each generator is useful, and the number of features chosen by filtering reflects more or less the number of features generated by each one before filtering.

3.5. Example – the word '*wysokość*'

To illustrate the statistical considerations above, here is an example for a particular word: *wysokość* (*height* or *highness* in English). The senses of *wysokość* in the NKJP dictionary are:

1. *wielkość mierzona od dołu do góry; odległość; częstotliwość drgań w dźwięku* (height),
2. *członkowie rodziny królewskiej lub książęcej* (highness).

Feature selection reduced the number of generated features from 1,489 to only 11. These are the features used by the best classifier for this word:

Feature	Feature type
Wysokość	KF
Lech	TF

Feature	Feature type
przecież	TF
,	TF
w	TF
Jego-1	SF1
Książę+1	SF1
m1-1	SF2
m1+1	SF2
akc-1	SF2
sg+1	SF2

4. Conclusion

The single best method for each of the 106 disambiguated words was used for the annotation of the National Corpus of Polish.

It is worthwhile considering the final accuracy of annotation. The *most frequent sense* baseline was considerably exceeded, but what is the ceiling of WSD? Research shows that the inter-annotator agreement for English can reach at most **95%** when there are only 2 very distinct senses of a noun to choose from (see Kilgariff 1998), and about **65–70%** when there are more senses (Chklovski, Mihalcea 2003). Due to the fact that the ambiguity of sense inventory was closer to 3 senses per lexeme, more than **91%** accuracy seems to be an excellent result.

Future research should show to what extent this result can be carried over to other ambiguous lexemes in Polish.

References

Witten I.H., Frank E., 2005, *Data Mining: Practical machine learning tools and techniques,* San Francisco.

Młodzki R., Przepiórkowski A., 2011, *The WSD development environment.* [in:] *Human Language Technology. Challenges for Computer Science and Linguistics: 4th Language and Technology Conference, LTC 2009, Poznań, Poland, November 6–8, 2009, Revised Selected Papers,* ed. Z. Vetulani, volume 6562 of *Lecture Notes in Artificial Intelligence*, pp. 224–233, Berlin.

Przepiórkowski A., 2004, *The IPI PAN Corpus: Preliminary Version*,Warsaw.

Przepiórkowski A., Górski R.L., Łaziński M., Pęzik P., 2010, *Recent Developments in the National Corpus of Polish*, [in:] *Proceedings of the Seventh International Conference on Language Resources and Evaluation, LREC 2010*, Valletta, Malta, 2010. ELRA., http://www.lrec-conf.org/proceedings/lrec2010.

Narodowy Korpus Języka Polskiego, eds. Przepiórkowski A., Bańko M., Górski R.L., Lewandowska-Tomaszczyk B. , 2012, Warszawa.

Chklovski, T., Mihalcea, R., 2003, *Exploiting Agreement and Disagreement of Human Annotators for Word Sense Disambiguation*, [in:] *Proceedings of RANLP 2003*, Borovetz, Bulgaria, September 2003, http://www.cse.unt.edu/~rada/papers/chklovski.ranlp03.pdf.

Kilgariff, A., 1998, *Inter-tagger agreement*, [in:] *Advanced Papers of the SENSEVAL Workshop*, Sussex, UK, http://www.nltg.brighton.ac.uk/events/senseval/ARCHIVE/PROCEEDINGS.

Ujednoznacznianie sensów słów w Narodowym Korpusie Języka Polskiego

S t r e s z c z e n i e

W artykule opisano procedurę ujednoznaczniania sensów słów w Narodowym Korpusie Języka Polskiego. Procedurze poddano 106 najczęstszych leksemów homonimicznych reprezentujących przeciętnie po prawie 3 różne znaczenia. Dla celów projektu sensy opisane w słownikach ogólnych zredukowano poprzez łączenie bardziej szczegółowych znaczeń w grupy znaczeniowo podobne.

Do wyróżnienia i identyfikacji sensów słów na podstawie kontekstu użyto narzędzia Word Sense Disambiguation Development Environment. Ekstrakcja cech z kontekstu dokonywana była za pomocą 4 rodzajów generatorów cech, wydobywających cechy tematyczne, dwa rodzaje cech strukturalnych oraz cechy słowa kluczowego.