

PAWEŁ KĘDZIA, MACIEJ PIASECKI, & MARLENA J. ORLIŃSKA

G4.19 Research Group, Wrocław University of Technology, Poland

{pawel.kedzia,maciej.piasecki,marlena.orlinska}@pwr.edu.pl

## WORD SENSE DISAMBIGUATION BASED ON LARGE SCALE POLISH CLARIN HETEROGENEOUS LEXICAL RESOURCES

### Abstract

Lexical resources can be applied in many different Natural Language Engineering tasks, but the most fundamental task is the recognition of word senses used in text contexts. The problem is difficult, not yet fully solved and different lexical resources provided varied support for it. Polish CLARIN lexical semantic resources are based on the plWordNet — a very large wordnet for Polish — as a central structure which is a basis for linking together several resources of different types. In this paper, several Word Sense Disambiguation (henceforth WSD) methods developed for Polish that utilise plWordNet are discussed. Textual sense descriptions in the traditional lexicon can be compared with text contexts using Lesk's algorithm in order to find best matching senses. In the case of a wordnet, lexico-semantic relations provide the main description of word senses. Thus, first, we adapted and applied to Polish a WSD method based on the Page Rank. According to it, text words are mapped on their senses in the plWordNet graph and Page Rank algorithm is run to find senses with the highest scores. The method presents results lower but comparable to those reported for English. The error analysis showed that the main problems are: fine grained sense distinctions in plWordNet and limited number of connections between words of different parts of speech. In the second approach plWordNet expanded with the mapping onto the SUMO ontology concepts was used. Two scenarios for WSD were investigated: two step disambiguation and disambiguation based on combined networks of plWordNet and SUMO. In the former scenario, words are first assigned SUMO concepts and next plWordNet senses are disambiguated. In latter, plWordNet and SUMO are combined in one large network used next for the disambiguation of senses. The additional knowledge sources used in WSD improved the performance. The obtained results and potential further lines of developments were discussed.

**Keywords:** word sense disambiguation; WSD; page rank; plWordNet; graphs; lexical resources; SUMO

## 1. Introduction

Words in any natural language can have more than one meaning. In the case of Polish, the word *zamek* is often cited as a canonical example of a word with several distinct meanings:

- a castle — a defence construction,
- a lock — a mechanism used for locking doors, drawer, etc.,
- a zipper — a kind of fastener for clothes,
- a breechblock — a part of a gun,
- $\approx$ a lock — a kind of situation in hockey during the power play period, in which the attacking team surrounds the defending team in their defensive zone,
- a lock — for accessing resources in a computer system.

*Zamek* is so overused example, as it is a homonym with clear distinctions between the different meanings. Genuinely polysemous words like *linia* ‘a line’ or *agent* ‘an agent’ have larger number of meanings with much more subtle differences among them. Every language evolves, words change meanings, some go out of use, others are added, e.g. on the basis of metonymy, especially in the case of polysemous words. Some words start to be used as terms in specialist domains. Meanings are not distinguished by word forms. The context of use clarifies, to some extent, the intended meaning with which the given word form was used in an utterance. That is obvious for a man, human addressee can recognise the meanings assigned by the speaker, but this is very difficult for computers.

*Word Sense Disambiguation* (henceforth WSD) (Yarowsky, 2010) is an automated assignment of the contextually appropriate *word senses* (i.e. *lexical meanings*) to words in text, in such a way that:

- the word senses come from a selected *sense inventory* and all possible senses are recorded in it,
- the assigned word senses match the semantic contexts of the word occurrences.

In the case of monosemous words, WSD retrieves only word sense symbols from the inventory, but in the case of the other words, the appropriate word sense must be chosen, i.e. disambiguated.

WSD is one of the oldest problems in Natural Language Engineering (NLE, Natural Language Processing in the past, NLP), because it is very fundamental. WSD is necessary for many different Natural Language Engineering tasks, as the recognition of word senses determines many following steps of processing. WSD problem has been formulated in the fifties (Yngve, 1955). However, practical solutions were not developed on these, days due to the limited computing power on that time. Later the WSD problem was not considered in separation to the issue of the semantic analysis. No earlier than in seventies, WSD started to be noticed as a separate problem and started receiving attention among researchers in the NLP/NLE field. Despite its long history and many methods proposed, WSD has not been solved till now.

In order to disambiguate a word occurrence, WSD methods try to the word sense that matches the best the given occurrence context. Thus WSD can be

casted as a classification problem. Several questions must be answered. What is the appropriate set of possible senses (e.g. its granularity)? How to represent it, i.e. how to organise and represent the sense inventory? Finally, how to describe the occurrence context and how to define the matching/classification procedure? The solution should be also adaptable to many if not all existing languages.

A closely related problem is grouping words occurrences according to the semantic features of their contexts in order to find uses of the same sense. If word senses are not known a priori, the process is called *word sense induction* — representations of word senses are induced from the established groups of occurrences. The idea is potentially very attractive as we need only a lot of text (for the statistical validity of the inference) and no a priori knowledge about the sense inventory. However, the main obstacle is that the identified word senses are not named, are very often not intuitive for humans and the influence of the statistical noise on the results is very hard to be estimated.

Many different methods have been proposed for WSD in the literature. Following (Yarowsky, 2010) we can divide them into three main groups:

- methods based on the supervised Machine Learning,
- methods utilising unsupervised Machine Learning (mostly word sense induction methods),
- weakly or remotely supervised methods based on a knowledge source.

### 1.1. Supervised WSD methods

As we noticed, WSD problem can be treated as a classification problem: each word occurrence must be assigned a word sense from the predefined set. Words senses are classes and their assignment to the occurrences depends on features describing the occurrences. The features can be derived from any aspect of the context, but we aim at making them related to the semantic information. First of all, the presence or the frequency of the other words in the context can be used, as well as the presence of collocations, morphosyntactic tags, some lexico-syntactic dependencies or constructions, etc.

Almost all Machine Learning algorithms were used to train and build classifiers for WSD. Some problem-specific extensions were also proposed. The first group of algorithms applied to WSD were: decision trees (Brown, Pietra, Pietra, & Mercer, 1991), decision lists (DL) (Yarowsky, 1994), naive Bayes classifier (NB) (Gale, Church, & Yarowsky, 1992) and k-nearest-neighbor algorithm (kNN) (Ng & Lee, 1996). Later, approaches based on the second generation of neural networks, AdaBoost (AB) (Schapire & Singer, 1999) or support vector machines (SVM) (Lee & Ng, 2002) were proposed. Wiriyathamabhum, Kijisirikul, Takamura, and Okumura (2012) used also neural networks of the third generation i.e. Deep Belief Networks (DBN).

Classifiers trained on well designed features and large training sets can achieve good accuracy. However, this promising approach to WSD have two significant drawbacks. Hundreds of manually disambiguated learning examples of word occurrences are needed for every word sense. Most words have very imbalanced distribution of

their senses. It is difficult to find examples of rare senses even in a very large corpora. In addition manual word sense annotation is very laborious and can consume many person-years for several thousand words. Large data sets of this type are not available for most languages. In a consequence, supervise WSD methods can be applied only to limited subsets of the vocabulary and their practical applications are difficult. They are more mostly used as to support method other types of WSD. A supervised WSD system for one language cannot be directly ported to another language without retraining on a new word sense annotated corpus for the given language.

However, supervised WSD methods achieve the best results (for the covered vocabulary), and they are popular in research. Yarowsky and Florian (2002) showed that the size of the training set and the selection of the features influence significantly the result. They suggested to concentrate on the exploration of the feature vector space and improvement of the training set.

## 1.2. Unsupervised WSD methods

Unsupervised methods do not utilise manually built resources such as: annotated training corpora, databases storing sense inventories or rules. A typical unsupervised WSD method is based on automated grouping of word occurrences.<sup>1</sup> In unsupervised WSD, it is assumed that similar occurrence contexts represent similar senses. Thus every obtained cluster should include occurrences representing the same sense.

Unsupervised WSD proposed in the literature can be divided into three groups (cf Pantel, 2003):

- hierarchical algorithms,
- partitional algorithms,
- and hybrid algorithms.

Hierarchical algorithms can utilise agglomerative or divisive clustering. According to the first, smaller clusters are linked iteratively into the larger groups. The process ends with one top supercluster including all analysed occurrences. In divisive clustering the process starts with large incoherent clusters that are next iteratively divided into smaller ones in a way maximising some similarity measure among the occurrences belonging to one cluster. The similarity measure is based on the similarity of the contexts of occurrences. In both schemes, a dendrogram of clusters is produced: starting with top large clusters and finishing with the bottom tiny clusters. The main question is: which clusters do represent the genuine word senses?

Partitional algorithms do not generate a hierarchy of clusters but flat division into a predefined number of clusters. The clusters are defined on the basis of the optimisation of a criterion, starting points (e.g. selected occurrences) and the stop criterion. Examples of the algorithms of this group are k-means and its variants.

Hybrid algorithms merge together features of the two other groups, namely, hierarchical and partitional, e.g. buckshot algorithm and Clustering by Commit-

---

<sup>1</sup>It would be enough to focus only on semantically ambiguous words if one knew which words were ambiguous not having any sense inventory.

tee (CBC). The first algorithm is a combination of the hierarchical agglomerative clustering and the k-means clustering algorithm. It takes on the input the number of clusters expected and data. The method starts with hierarchical clustering on the data set, so at the bottom level of the cluster tree leaves are singleton clusters (including one occurrence each). In the following steps smaller clusters are merged into larger according to the similarity measure, until the total number of clusters is not larger from the expected number delivered as the input parameter. Next, centroids are calculated for clusters with the help of the k-means algorithm.

The CBC method, which was proposed in Pantel (2003), and further expanded in Broda, Piasecki, and Szpakowicz (2010) into the LexCSD WSD system, works in three phases. First, word-to-word similarity is computed by Distributional Semantics methods on the basis of the occurrence contexts of words. The similarity measure is next used to calculate for each word a list of the  $k$  most similar words to it. In the second phase, the lists are used as an input to the agglomerative clustering. During the last phase each word is assigned to the most probable clusters for it (on the basis of its similarity to the cluster).

Unsupervised methods do not require manually annotated training examples and this is their main advantage. They are not also based on a priori assumed, closed sets of words senses, as they induce the division between different senses. Thus, they are independent from a particular language or a domain. However, the main problem is the lack of intuitive descriptions of the induced word senses. One cannot analyse the results until the clusters are manually labeled, e.g. each cluster could be described with words characteristic for it, as it is done in CBC. This creates problems for the evaluation.

The unsupervised methods found applications in Information Retrieval (IR), where consistency in indexing documents is the key aspect. The automatically induced word senses define elements of the indexing vectors and help to group together documents about similar topics. However, for the need of the semantic analysis of the text, WSD methods mapping text on well known lexical meanings are required.

### 1.3. Weakly supervised WSD methods

In weakly supervised approaches the manual supervision is limited to delivering pre-prepared knowledge sources, e.g. databases describing possible word senses, i.e. sense inventories. In brief, there are four main types of sense inventories (cf Sec. 2):

- *dictionary-based inventories* — providing textual definitions for senses,
- *concept hierarchies* — formalised or semi-formalised lexico-semantic networks,
- *domain tags/subject codes* — classifying words into semantic domains (the implicit assumption is that a word has at most one sense per domain).
- *multilingual translation equivalents* — different translations corresponds to different senses.

Several weakly supervised methods were proposed, e.g. Lesk's algorithm (Lesk, 1986), methods based on Measures of Semantic Relatedness (MSR) (Patwardhan, Banerjee, & Pedersen, 2003) and Page Rank (Agirre, López de Lacalle, & Soroa, 2014).

Lesk's algorithm utilises a sense inventory that can be identical to the traditional monolingual dictionary, i.e. it is assumed that every sense is described by short textual definition. In the algorithm, for a word in text the definition of its senses are compared with the occurrence context. The sense with the largest overlapping is selected as the contextually appropriate for the given word occurrence. Different similarity measures can be applied. The main problems are limited dictionary definitions<sup>2</sup> and high computational complexity, as many word sets must be compared. Despite its simplicity Lesk's algorithm can express surprisingly good accuracy, depending on the properties of the sense inventory.

Measures of Semantic Relatedness that are based on the hierarchical lexico-semantic network, e.g. a wordnet, determine the degree of the relationship between two word senses taking into account the distance between them in the network hierarchy or their information content. The information content is a value that is assigned to each sense in the hierarchy derived from their occurrences in the corpus. MSR-based WSD works in two steps. First, similarity between all senses of a word to be disambiguated and all senses of words from the surrounding context is measured. In the second step scores associated with each combination of senses are summed. The sense of the target word with the highest score is selected. Five different MSRs based on the Princeton WordNet (Fellbaum, 1998) were tested in Patwardhan et al. (2003): Leacock-Chodorow, Resnik's measure, Jiang-Conrath measure, Lin's measure and Hirst-St. Onge measure.

Finally, in the method of Mihalcea, Tarau, & Figa (2004); Agirre & Soroa (2009); Stevenson, Agirre, & Soroa (2012); Agirre et al. (2014) the well know Page Rank algorithm was applied to compute the level of support for different words senses on the basis of the occurrence context. This method will be discussed in details in Sec. 3.

Weakly supervised methods do not require manually word sense disambiguated corpora for training. They use already existing word sense inventories, so they are cheap in application. Their properties and quality depends a lot on the type and quality of the sense inventory utilised. Such methods express usually lower precision than supervised methods and comparable to the unsupervised. However, they express much higher recall than the other two types. Their coverage is limited only by the size of the inventory. The inventory must be manually expanded if new words or new senses appear or are absent. This is the main drawback of such methods.

#### 1.4. Goal

So far there is no wide coverage WSD method for Polish. Several supervised methods were proposed (Baś, Broda, & Piasecki, 2008; Młodzki & Przepiórkowski, 2009) but they have experimental character and work for small word subsets. An unsupervised method was also proposed, namely LexCSD. However, LexCSD has also limited coverage and problems with learning and recognising less frequent word senses, as it clearly tends to focus on the most frequent word senses. Moreover, LexCSD needs manual labelling of the induced senses for many approaches. Such la-

---

<sup>2</sup>They include limited number of words and the words used in the definition may not occur in the context.

bellings would be time consuming, and worth to be constructed only if the method's coverage has been improved.

As there is a huge wordnet for Polish providing very extensive coverage of Polish word senses, we assumed that the easiest way to build a robust WSD method for Polish is to use one of the weakly supervised methods.

Thus, our goal was to develop a Word Sense Disambiguation method which can be applied to all words described in a huge wordnet for Polish, namely plWordNet.

As plWordNet includes textual glosses only for small portion of word senses and the existing glosses are short, Lesk's algorithm could not be applied. In a similar way the method proposed in Patwardhan et al. (2003) did not seem to be feasible. Instead we decided follow the general scheme of the Page Rank-based WSD method proposed by Agirre et al. (2014).

plWordNet differs slightly in its general model and several specific features from Princeton WordNet, e.g. the number of relations is bigger in plWordNet and the network is denser. Not all resources assumed by (Agirre et al., 2014) are available for Polish, e.g. limited number of glosses in plWordNet and the lack of word sense disambiguated corpus for Polish of a similar type to the one used by (Agirre et al., 2014) for increasing the number of the network links. Thus, our second goal is to investigate how useful is plWordNet for WSD, which of its properties should be improved or already suits well WSD and how to modify the Page Rank WSD algorithm to Polish language resources.

## 2. plWordNet as a Sense Inventory

Sense inventory is a database providing descriptions for word senses (i.e. lexical units). Its primary function is to enumerate all existing word senses and assign a unique identifier to each. Moreover, each word sense should be described in in some established and consistent format. However, the exact method of the description can vary among sense inventories.

Four main types of sense inventories can be distinguished: dictionary-based inventories (textual descriptions for senses), lexico-semantic networks (senses are described by relation links), domain tags (or subject codes, tags are the only explicit description) and multilingual translation equivalents (the equivalents are sense identifiers). Formalised semantic lexicon are only theoretically possible, as they are mostly too small to have practical importance and are too laborious in development. If a WSD tool is to have broader application, it is must be based on a free licence inventory. There is no large monolingual dictionary for Polish on an open licence. Domain tags provide too coarse grained information for weakly supervised methods. Multilingual translation equivalents (multilingual translation dictionaries) could be applied if we had some additional resources in one of the target languages or a very large bilingual corpus aligned at least on the sentence level. Fortunately, plWordNet — a huge wordnet for Polish — is the biggest world's wordnet and provides very comprehensive description of Polish lexical units.

Word senses are represented in plWordNet as *lexical units*. A lexical unit is a pair: lemma<sup>3</sup> plus sense identifier. Lexical units are the basic building blocks of

---

<sup>3</sup>A lemma is a basic morphological form representing a group of word forms that have the same meaning but differ in the values of the morphological categories.

plWordNet. Lexical units are described by lexico-semantic relations, e.g. hypernymy-hyponymy, holonymy-meronymy, antonymy, cause, inter register synonymy, etc. There are about 40 main types of lexico-semantic relations in plWordNet 2.2. Their detailed description can be found in (Piasecki, Szpakowicz, & Broda, 2009; Maziarz, Piasecki, & Szpakowicz, 2012; Maziarz, Piasecki, Rudnicka, & Szpakowicz, 2013; Piasecki, Szpakowicz, Fellbaum, & Pedersen, 2013). Some relations, e.g. holo/meronymy are further subdivided into subtypes for nouns into *part*, *time*, *place*, *collection element*, *material* and *taxonomic element*, where for verbs into *subsituation* (meronymy) and *accompanying situation* (meronymy). So, the total number of lexico-semantic relations in plWordNet 2.2 exceeds 90.

Lexical units that share a set of lexico-semantic relations are grouped together into sets called *synsets* and are considered to be *near synonyms*.

Not all lexico-semantic relations are shared (or are shared enough frequently and systematically), e.g. hyper/hyponymy is shared, but not antonymy. Those relations that can be shared are called *constitutive relations*, as they define synsets and the main structure of plWordNet. The shared relations are encoded as links between synsets,<sup>4</sup> other relations as direct links between lexical units.

plWordNet includes one-word and multi-word lexical units (i.e. lexical units with lemmas including more than one word). However only those multi-word expressions that can be treated as elements of the Polish lexical system are described by lexical units in plWordNet. Guidelines developed for the recognition of the multi-word lexical units are quite complicated and their presentation goes beyond the scope of this paper.

plWordNet provides also some additional means of the semantic description, namely: *stylistic registers*, *glosses* and *use examples*. Stylistic registers express pragmatic constraints on the use of lexical units. However such subtle differences have minor influence on WSD performance. Glosses are comments to the lexical units provided for human reader. Their purpose is to explain the motivation behind the given word sense and clarify its difference in relation to the other senses of the same lemma. Glosses are short descriptions and are similar in form to the dictionary definitions, but they have never meant to be properly formed lexicographic definitions. Glosses are secondary to the lexico-semantic relations that are the primary tool for the description of the lexical meanings in plWordNet, e.g. the genus information is expressed by the hypernymy and should not be provided in a gloss. Glosses are not enough elaborated to be used as a basis for Lesk's algorithm. In PWN glosses are provided for synsets, while in plWordNet they are attached to lexical units, as the basic building blocks. In addition to glosses, a lexical unit can be describe by one or more use examples. They are also focused on human readers, but they can be used in WSD as an additional source of information. Both glosses and use examples have not been word senses disambiguated yet.

The size of plWordNet is described in Table 1. We can notice that this is one of the biggest Polish dictionary and the largest world's wordnet:

---

<sup>4</sup>If two synsets are linked by, e.g., hypernymy, than all pairs of lexical units formed from the two synsets are linked by the hypernymy too. Synsets are a kind of notational abbreviation.

**Table 1:** Statistics for plWordNet from the 5th Nov., 2014 — used in the experiments in place of the older and smaller version 2.2 <http://plwordnet.pwr.wroc.pl>. The description covered about 156 000 lemmas.

Part of speech	Number of Lexical Units	Number of synsets
<b>Noun</b>	159 180	116 976
<b>Verb</b>	31 655	21 533
<b>Adjective</b>	27 936	23 773
<b>All</b>	218 798	162 282

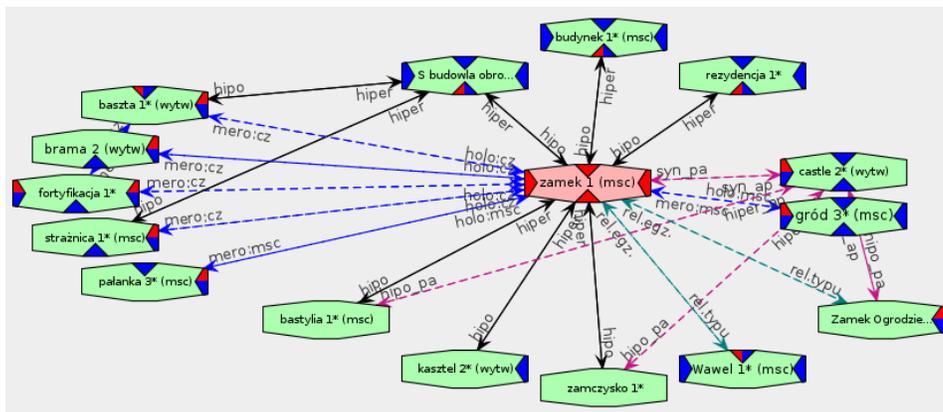
### 3. WSD based on the Page Rank algorithm

*Page Rank* algorithm has been developed as a method for ranking the search engine results for a user query (Brin & Page, 1998). The algorithm is based on iterative updating scores assigned to websites in a random browser model. Page Rank has been first adopted to WSD task by Mihalcea et al. (2004). A slightly modified version of Page Rank was proposed by Agirre and Soroa (2009); Stevenson et al., 2012; Agirre et al. (2014) in a form of a *Personalized Page Rank* and next *Personalized Page Rank word-to-word*.

Page Rank-based WSD explores a relation between the wordnet graph of relations and textual contexts of word occurrences. Wordnet is a graph of synsets linked by different types of lexico-semantic relations. Synsets include words that can occur in text. In Page Rank-based WSD we assume that word senses that are semantically related occur more likely together in text than non-related. So, if we map words senses from a text fragment on the wordnet graph we can expect that the ‘hits’ are located in short distances in the wordnet graph, i.e. they are linked by short paths of the lexico-semantic relation links. Moreover, many words are monosemous, so their mapping onto the wordnet graph are unambiguous and can point for find out the best matching of between wordnet graph areas and the given text document or fragment.

For instance, the word *zamek* is homonymous with 7 word senses. A part of the plWordNet graph including the lexical unit {*zamek* 1 ‘castle’ (msc)} is presented in Fig. 1. Only synsets linked directly to *zamek* 1 are presented. However, even in the case of this limited subgraph, we can notice many word senses that can be expected to appear in the text about {*zamek* 1 ‘castle’ (msc)}, not about *zamek* in the sense ‘zipper’.

Obviously, the problem is more complex, and the picture is not so clear on average. We need to take into account larger subgraphs and it is not enough to simply count number of hits as many words are polysemous and can have several potential hits in several subgraphs. Page Rank algorithm is used to propagate the initial network activation expressed by ‘hits’ from the text and spread it across larger wordnet graph areas. It has been assumed that the activation finally concentrates in the contextually most appropriate synsets representing word senses.



**Figure 1:** A screenshot from the WordnetLoom application with the centred synset of  $\{\text{zamek 1 'a castle'}\}$ . Only one lexical unit per synset is presented in this view.

A wordnet is used as the only knowledge source used in the Page Rank-based WSD and is represented as a graph  $G$ :

- the nodes represent the synsets,
- the edges — lexico-semantic relation links between synsets,
- if a pair of synsets  $S_i$  and  $S_j$  is linked by a directed relation  $R(S_i, S_j)$  in the wordnet,
- then the nodes  $v_i$  and  $v_j$  representing  $S_i$  and  $S_j$  are connected by the edge  $e(v_i, v_j, R)$  labeled as  $R$ .

Whole documents are used as the contexts for the disambiguation. The documents are represented as sequences of words, i.e. the syntactic and semantic structure of the document is not taken into account. It is assumed that only one word sense per lemma is used in the document.

### 3.1. Static Page Rank in WSD

Page Rank implements a model of the random walker and computes iteratively probability estimations for visiting a given node by a walker starting from some node in the graph. The model depends on the initial probabilities of randomly visiting the nodes and the graph structure.

Let:

$G$ : Graph,

$N$ : the number of nodes of  $G$ ,

$V$ : a set of the nodes from graph  $G$ ,  $V = \{v_1, v_2, \dots, v_N\}$

$d_i$ : the node degree  $v_i$ ,  $i = 1, 2, \dots, N$ ,

$M_{N \times N}$ : the adjacency matrix,  $M_{N \times N} = [m_{ij}]_{i \in \{1, 2, \dots, N\}, j \in \{1, 2, \dots, N\}}$ ,

where

$$m_{ij} = \begin{cases} \frac{1}{d_i} & \text{if exists the edge from } v_i \text{ to } v_j \\ 0 & \text{otherwise} \end{cases} \quad (1)$$

In each iteration updated scores for nodes are calculated according to the following procedure run on node score vectors:

$$\mathbf{P}^{(\text{new})} = cM\mathbf{P}^{(\text{old})} + (1 - c)\mathbf{v}, \quad (2)$$

where:

$\mathbf{P}_{\mathbf{N} \times \mathbf{1}}$  — a vector, such that  $\mathbf{P}_{\mathbf{N} \times \mathbf{1}} = [p_i]_{i \in \{1, 2, \dots, N\}}$  — the updated score for the  $i$ -th node,

$c$ : the damping factor, defining the influence of the updates in relation to the initial score,

$\mathbf{v}_{\mathbf{N} \times \mathbf{1}}$ : a stochastic vector — the initial scores for node (a priori probability estimations),<sup>5</sup>

where  $\mathbf{v}_{\mathbf{N} \times \mathbf{1}} = [\alpha_i]_{i \in \{1, 2, \dots, N\}}$  — the initial score for the node  $i$

In each iteration step in (2) the old score of the node  $i$ , i.e.  $\mathbf{P}[i]$  is updated on the basis of its initial value  $\mathbf{v}[i]$  and the scores from the nodes linked to  $i$ . The matrix  $M$  defined in (1), determines the influence of the linked nodes on  $i$  and the spreading of scores in the whole graph. If the value of  $M_{ji}$  is different than 0 it means, that there is a link going from  $v_j$  to  $v_i$  and the score of  $v_j$  influences the  $v_i$  score in each update iteration. The second part (2) introduces the constant influence of the initial scores in  $\mathbf{v}$ . The strength of this influence is constrained by the dumping factor  $c \in \langle 0, 1 \rangle$  — mostly  $c$  is set to  $\langle 0.85, 0.95 \rangle$  (cf Mihalcea et al., 2004; Agirre et al., 2014). The algorithm stops if one of the two conditions is fulfilled:

1. the maximum number of iterations has been achieved,
2. the difference between  $\mathbf{P}^{(\text{old})}$  and  $\mathbf{P}^{(\text{new})}$  is smaller than the assumed threshold.

Several variants of Page Rank-based WSD can be distinguished that differ in a way of setting the initial values in  $\mathbf{v}$ :

- *Static Page Rank (SPR)* — discussed above, all  $\mathbf{v}$  elements are set to the same value and do not depend of the occurrence context;
- *Personalised Page Rank (PPR)* — the  $\mathbf{v}$  values depend on the context which includes the word being disambiguated ;
- *Personalised Page Rank Word-to-Word (PPR\_W2W)* —  $\mathbf{v}$  depends on the context, but the word being disambiguated is excluded from the context.

<sup>5</sup>In Agirre et al. (2014) it was renamed as a *personalised vector*  $v$ .

In *PPR* and *PPR\_W2W* only some nodes which are contextually supported are initially set to non-zero score values, so the processing is concentrated on some parts of the wordnet graph.

*SPR* and *PPR* disambiguate all ambiguous words from the analysed context at a time. *PPR\_W2W* can process only one ambiguous word at a time, the word being disambiguated is excluded from the context and only other words are used to set  $\mathbf{v}$ . The use of context in *SPR* is explained below, *PPR* and *PPR\_W2W* are described in the following sections, 3.2. and 3.3., respectively.

In *SPR* the graph  $G$  is built from the whole wordnet graph. *SPR* does not take into account the occurrence context, the  $\mathbf{v}$  values into vector are identical for all nodes and are set to:

$$\alpha_i = \frac{1}{N}, \quad i = 1, 2, \dots, N \quad (3)$$

where,  $N$  is the number of nodes into graph  $G$ . (Agirre et al., 2014) observed, that the ranking position is connected with the node degree, the higher node degree indicates the higher ranking position of the disambiguated word.

### 3.2. Personalised Page Rank

The graph  $G$  is created as in *SPR*, but  $\mathbf{v}$  are not uniform and depend on the occurrence of lemmas in the text:

$$\alpha_i = \frac{1}{\frac{CS}{NS(l)}}, \quad i = 1, 2, \dots, N \quad (4)$$

where  $CS$  is the number of different lemmas in the context,  $NS(l)$  is the number of synsets with a given lemma  $l$ . Words that are disambiguated are included in the context. Contrary to *SPR*, only synsets including lemmas from the context have non-zero values in  $\mathbf{v}$ . As the ambiguous words are included in the context, the whole context can be disambiguated at a time.

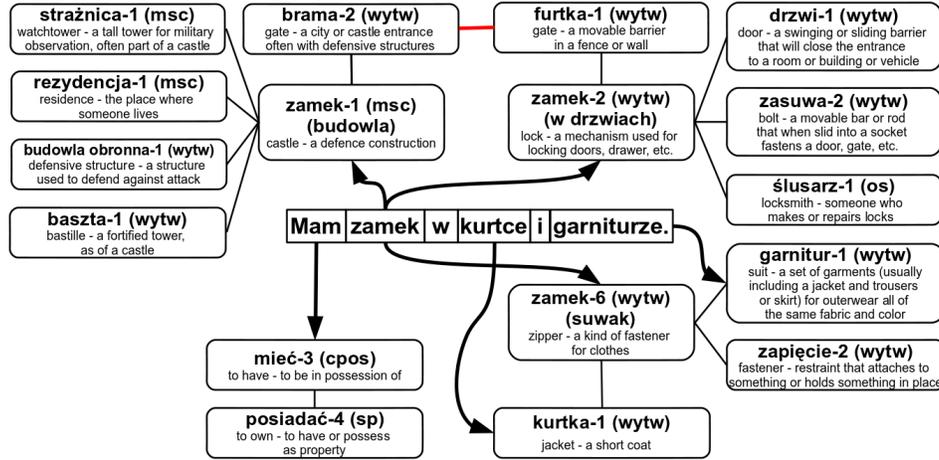
In PPR-based WSD, first, the graph  $G$  is built in the identical way to *SPR*. Next,  $\mathbf{v}$  is set according to Equation 4, i.e. all nodes not including lemmas from the context are assigned 0 in  $\mathbf{v}$ . After the initialisation, the iteration (2) is run for the predefined number of steps. The final score values of  $\mathbf{P}$  are used to rank word senses of the same lemmas. Following (Agirre et al., 2014), a ranking for a disambiguated word is normalised to the range  $\langle 0, 1 \rangle$ , and sorted in descending order. The top ranking position (i.e. a particular synset) is chosen as the meaning of the word occurrence.<sup>6</sup>

PPR performs significantly better than *SPR*, but PPR expresses one negative feature. If a word  $w$  to be disambiguated have at least two synsets that are close in  $G$ , then they reinforce each other. As a result those synsets dominate the  $w$  senses. This problem can be observed in PPR application to the disambiguation of the word *zamek* ‘castle, lock, zipper, ...’ in the context of a simple sentence which is illustrated in Figure (2):

„Mam zamek w kurtce i garniturze.” ‘I have zipper in the jacket and suit’.

Initially non-zero scores are assigned only to nodes (i.e. synsets) pointed to by arrows. This initial set includes also all synsets of *zamek* — not all of them are

<sup>6</sup>All occurrence of the same lemma in a context are assigned the same word sense in *PPR*



**Figure 2:** PPR disambiguation of the word *zamek* ‘castle, lock, zipper, ...’ in the context of: „Mam zamek w kurtce i garniturze.” ‘I have zipper in the jacket and suit.’.

presented in Figure (2). Because the synset „zamek-1” and „zamek-2” reinforce each other as being indirectly linked, they receive very similar scores and the wrong sense *zamek* 1 is finally selected instead of *zamek* 6, while the latter is clearly the most appropriate sense for the given context.

### 3.3. Personalized Page Rank Word-to-Word

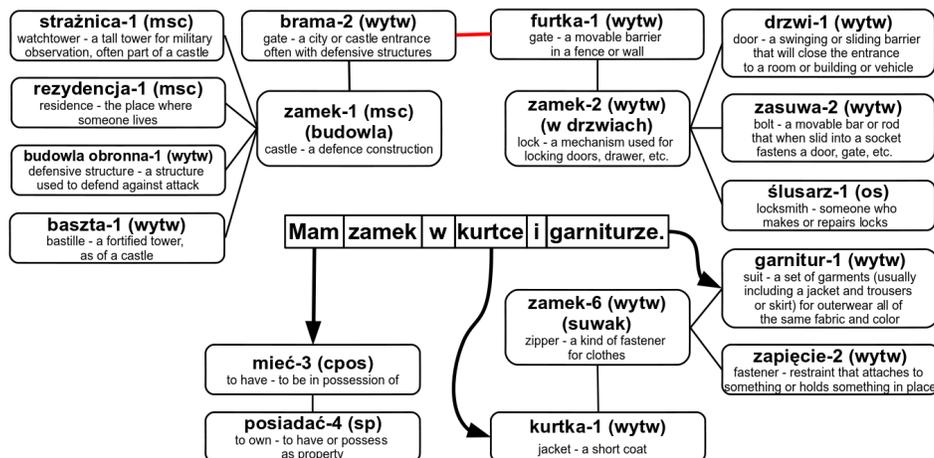
Personalised Page Rank Word-to-Word (PPR\_W2W) (Agirre & Soroa, 2009; Stevenson et al., 2012; Agirre et al., 2014) is a modification of PPR in which a word to be disambiguated is excluded from the occurrence contexts, i.e. all synsets of this word have initial scores in  $\mathbf{v}$  set to zero. Thus, PPR\_W2W cannot be run once for all ambiguous words in the context. The vector  $\mathbf{v}$  must be initialised individually for each ambiguous word in the context — this is a disadvantage of PPR\_W2W. A potential advantage is removing the effect of the mutual amplification of the closely connected senses of the word being disambiguated.

For each ambiguous word, the vector  $\mathbf{v}$  is set as follows:

$$\alpha_i = \begin{cases} \frac{1}{\frac{CS-1}{NS(l)}} & i = 1, 2, \dots, N \text{ if } v_i \text{ is the meaning of word from context,} \\ & \text{excluding disambiguated } w \\ 0 & \text{otherwise} \end{cases} \quad (5)$$

where  $CS$  is the number of different lemmas in context,  $NS(l)$  is the number of synsets with a given lemma  $l$  excluding disambiguated word.

Except the different initialisation of the vector  $\mathbf{v}$  and the repetition of this for every ambiguous word, PPR\_W2W works in exactly the same way as in PPR. A PPR\_W2W application to the same example as PPR, cf Sec. 3.2 is illustrated in Fig. 3. The algorithm is based only on the words from the context (marked



**Figure 3:** PPR\_W2W disambiguation of the word *zamek* ‘castle, lock, zipper, ...’ in the context of: „Mam zamek w kurtce i garniturze.” ‘I have a zipper in the jacket and the suit.’.

with black arrows), not on the word *zamek* ‘castle, lock, zipper, ...’, which is being disambiguated. As a result, *zamek* synsets have much lower influence on each other and the meaning *zamek* 6 ‘zipper’ is returned. The improvement is for the cost of larger complexity of PPR\_W2W and its lower efficiency. Table 2 presents processing time (expressed in seconds) for different Page Rank algorithms with different context types. The processing time for PPR\_W2W increase much more faster than PPR and Static Page Rank. This increase is caused by running Page Rank for each ambiguous word separately.

**Table 2:** Processing time in seconds for different genres of Page Rank algorithm and different types of context.

File size [B]	Number of tokens	PPR		PPR_W2W		STATIC	
		sentence	document	sentence	document	sentence	document
74 022	193	2.9030	1.7411	17.1580	17.2520	1.7364	1.7041
156 473	406	5.5898	1.7831	33.9999	34.0473	1.7930	1.7843
289 310	746	7.9633	1.8418	60.1102	60.3617	1.8453	1.8325
380 461	959	9.8775	1.8760	77.9755	77.5287	1.8840	1.8777
622 845	1 518	14.7760	1.9994	127.2792	127.5446	1.9924	1.9935

#### 4. Expanded Lexical Resources

Page Rank based WSD depends strongly on links between word senses. A denser lexico-semantic network can provide better descriptions for the individual senses. From the WSD point of view, the most interesting are links that express possible coincidence of word senses in different contexts. Among plWordNet lexico-semantic

relations, we can find two types that are of special usefulness for WSD:

- generalisation, e.g. hypernyms,
- co-occurrence of word senses, e.g. relations linking adjective lexical units with noun lexical units.

We aimed at expand plWordNet with resources enhancing word sense linking in terms of the network density and information expressed. First, we used the mapping of plWordNet to SUMO Ontology (Kędzia & Piasecki, 2014), see Sec. 4.1, as a source of potential generalising links. Merging plWordNet with SUMO results in an enhanced knowledge base, but also it provides an opportunity to apply a *two step disambiguation process*, that is discussed in Sec. 4.2.

#### 4.1. Mapping plWordNet onto SUMO Ontology

The *Suggested Upper Merged Ontology* (SUMO) is a formal upper and medium level ontology including definitions of concepts and selected instance organised into a network based on a few ontological relations (Pease, 2011). The mapping of plWordNet to SUMO follows and utilises a similar work done for the Princeton WordNet (PWN) and the aim was to provide formal semantic interpretations for the lexical meanings represented by the plWordNet synsets. The mapping procedure described in Kędzia and Piasecki (2014) has been based on the two other existing mappings: *interlingual relations* between plWordNet and PWN (Rudnicka, Maziarz, Piasecki, & Szpakowicz, 2012), and *relations between PWN synsets and SUMO concepts* (Niles & Pease, 2003; Pease & Fellbaum, 2010). Both mappings have been built manually. Connections between plWordNet synsets and SUMO concepts have been established semi-automatically on the basis of manually written rules operating on the both mappings. plWordNet synsets are linked to with SUMO concepts with one of the relations (in some cases a synset is connected to more than one concept):

- *subsumed* — plWordNet synset denotation is subsumed by SUMO concept (an analogue to linguistic hyponymy), e.g. {*brzdęk* 1 ‘twang’ (zj)} is subsumption of *RadiatingSound*;
- *instance of* — a synset denotation is an instance of a SUMO concept, e.g. {*Arystoteles* 1 ‘Aristotle’} is an instance of *Man*;
- *equivalent* — a synset is equivalent to the SUMO concept with respect to the synset’s denotation, e.g. {*sobota* 1 ‘Saturday’ (czas)} is equivalent to *Saturday*;

After mapping process, not each synset have been mapped to a SUMO concept. There is about 3% of noun and 5% verb synsets in which the rules abstained. Due to the created mapping SUMO with plWordNet graphs are merged into one heterogeneous graph.

#### 4.2. Two-step Word Sense Disambiguation

During the preprocessing, both graphs: plWordNet and SUMO, have been merged into one on the basis of the mapping plWordNet onto SUMO (Kędzia & Piasecki, 2014). The disambiguation process was divided into two steps with running Page Rank into both.

In the first step (called as *coarse grained disambiguation*), SUMO ontology is utilised as a network of interconnected concepts. For each word  $w$  from the context, we choose the set of concepts connected with  $w$  and initialise vector  $\mathbf{v}$  according to chosen disambiguation method. Then, the process of coarse disambiguation is run according to Equation (2). After the disambiguation, for one word  $w$  we choose only one concept. The second phase of disambiguation (*fine-grained disambiguation*) based on plWordNet graph and synset mappings onto SUMO ontology. The difference is in  $\mathbf{v}$  initialisation. Only these elements are initialised, which has connections (mappings) with concept chosen in previous step. For example, for word “zamek”, in the previous stage the *Lock* concept has been assigned, only two from six synsets are initialised: {*zamek* 2 ‘lock’ (wytw)} and {*zamek* 5 ‘semaphore’ (wytw)}. Next, the disambiguation process is run according to Equation (2) and as the result the ranking value for each node is returned. For each word, the synset with the highest ranking value is chosen as a meaning of words.

## 5. Experimental setting

Evaluation was based on applying the analysed algorithms to a corpus with manually disambiguated word senses (a subset of them) and next measuring the performance. We used the precision measure (see Equation (6)) as a main criterion for evaluation:

$$Pr = \frac{t}{t + f} \quad (6)$$

where:

- $t$ : the number of correctly disambiguated instances,
- $f$ : the number of incorrectly disambiguated instances.

The precision was measured against manual annotations in the *KPWr Corpus*, see Sec. 5.1. In each experiment we used the whole document as an occurrence context for the disambiguated words. plWordNet (the version from 5th November, 2014) and SUMO ontology were used as knowledge bases (SUMO was merged with plWordNet in some experiments). Both knowledge sources were treated as undirected graphs. The damping factor ( $c$ ) for the Page Rank algorithm has been set to 0.85 according to Mihalcea et al. (2004); Agirre et al. (2014). However, the number of iterations for each experiment was varied starting from 5 iterations, increasing by 5 and ending at 30 iterations.

### 5.1. Polish Corpus of the Wrocław University of Technology

The *Polish Corpus of the Wrocław University of Technology*, known as *KPWr*<sup>7</sup> (Broda, Marcińczuk, Maziarz, Radziszewski, & Wardyński, 2012) contains 1127 documents ( $\approx 250\,000$  tokens) divided into 11 thematic categories presented in Table 3. *KPWr* has been manually annotated and disambiguated on many levels of the natural language analysis, such as morpho-syntactic, syntactic relations, semantic relations, Named Entities, selected meta-informations and also lexical meanings. Concerning the lexical level, all occurrences of 45 different nouns and 29 verbs were manually mapped on the plWordNet synsets. Documents are balanced

<sup>7</sup>Due to its name in Polish: *Korpus Języka Polskiego Politechniki Wrocławskiej*. The corpus is available under the Creative Commons license: <http://nlp.pwr.edu.pl/kpwr>.

across different genres, namely written and spoken, universal and technical. Each document is structured into: paragraphs, sentences and tokens.

**Table 3:** Distribution of different categories in KPWr. KP

Domain	[%]
Blogs	8.5%
Stenographic recordings	7.0%
Dialogues	2.5%
Past prose	7.5%
Law	6.5%
Long press articles	8.5%
Short press articles	17.5%
Popular science and textbooks	5.5%
Wikipedia	32.0%
Official texts	3.5%
Technical texts	1.0%

Table 4 includes the statistics for word sense annotation. As selected words were only annotated, the result set can be divided into independent parts for nouns and verbs. The reason for this division was that the various parts of speech represent various problems during the Word Sense Disambiguation process.

**Table 4:** Statistic of WSD annotations in KPWr.

	Total	Nouns	Verbs
Number of tagged words	74	45	29
Number of tagged instances	5 148	3 219	1 929

Table 5 presents further statistics about the manual word senses annotations in KPWr. For 3219 annotated nouns and 1929 verbs, the average number of word senses per word are 6.67 and 8.42 respectively. The standard deviation is 4.01 for nouns and 4.08 for verbs. The median of number of senses for nouns is 6, where 534 nouns has the number of senses equal to the median. 1217 nouns has a number of senses greater than the median, and 1468 less. The median of number of senses for verbs is 6, where 359 verbs has a number of senses equal to the median. 622 verbs has a number of senses less than the median, and 948 more. Thus, the annotated words are quite diversified and representative.

**Table 5:** Statistic information about the number of manually word senses annotations in KPWr.

	Total Nouns Verbs		
Average number of senses	7.34	6.67	8.42
Standard deviation of number of senses	4.12	4.01	4.08
Median of number of senses	6	6	6
Number of senses equal to median	892	534	359
Number of senses less than median	2090	1468	622
Number of senses greater than median	2165	1217	948

## 5.2. Results

Our goal was to compare the quality of different WSD approaches, discussed so far. The Page Rank algorithm was used in three different versions of the lexical knowledge base. In the first approach, only nodes belonging to plWordNet (the version from the 5th November, 2014) were initialised and used in disambiguation process. In the second experiment, plWordNet (the version from the 5th November, 2014) and SUMO were merged into one big network and all nodes were used. Two-step disambiguation process was tested in the third experiment, where in the first step only nodes from SUMO ontology were initialised. After SUMO concepts were selected, nodes from plWordNet (the version from the 5th November, 2014) were initialised in the second step to find the word senses, cf 4.2.

All results, which was obtained for approaches described above, were presented in Tables 6, 7, 8, 9, 10: **STATIC** corresponds to the Static Page Rank, **PPR** is a Personalised Page Rank and **PPR\_W2W** — Personalised Page Rank Word-to-Word. Results for nouns (**N**) and verbs (**V**) were separated, while the column **All** contains precision for both parts of speech together. The best result in column are bolded.

First, the Page Rank algorithm was used with the smallest lexical knowledge base, i.e. only plWordNet and we treated this approach as a baseline for two others settings. Results are given in Table 6. The best result was achieved for *PPR\_W2W* for both parts of speech. In addition, we can also notice that the performance is much higher for nouns than for verbs. It is caused by the fact that nouns are better described in plWordNet than verbs. In addition verbs are more ambiguous than nouns.

The disambiguation precision for the second setting, i.e. for plWordNet merged with SUMO ontology as a lexical knowledge base, is given in Table 7. In this setting, nodes belonging to the plWordNet and the SUMO ontology were initialised. The highest disambiguation quality for both parts of speech was achieved for *PPR\_W2W* also. The result of *PPR\_W2W* for nouns is higher than that observed in the baseline setting, cf Table 6.

**Table 6:** Precision of disambiguation process based on the plWordNet from the 5th November, 2014.

Iterations	PPR			PPR_W2W			STATIC		
	V	N	All	V	N	All	V	N	All
5	33.90	52.94	45.80	37.58	57.50	50.04	<b>34.11</b>	<b>46.16</b>	<b>41.65</b>
10	<b>34.06</b>	<b>53.06</b>	<b>45.94</b>	<b>37.69</b>	57.63	<b>50.16</b>	<b>34.11</b>	<b>46.16</b>	<b>41.65</b>
15	<b>34.06</b>	53.03	45.92	37.64	<b>57.66</b>	<b>50.16</b>	<b>34.11</b>	<b>46.16</b>	<b>41.65</b>
20	<b>34.06</b>	53.03	45.92	37.64	<b>57.66</b>	<b>50.16</b>	<b>34.11</b>	<b>46.16</b>	<b>41.65</b>
25	<b>34.06</b>	53.03	45.92	37.64	<b>57.66</b>	<b>50.16</b>	<b>34.11</b>	<b>46.16</b>	<b>41.65</b>
30	<b>34.06</b>	53.03	45.92	37.64	<b>57.66</b>	<b>50.16</b>	<b>34.11</b>	<b>46.16</b>	<b>41.65</b>

**Table 7:** Precision of disambiguation process based on the plWordNet from 5th November, 2014 merged with SUMO ontology. Nodes belonging to the plWordNet and the SUMO ontology were initialised.

Iterations	PPR			PPR_W2W			STATIC		
	V	N	All	V	N	All	V	N	All
5	32.56	51.57	44.44	<b>36.39</b>	58.00	49.90	<b>32.19</b>	<b>46.63</b>	<b>41.22</b>
10	32.50	<b>51.63</b>	44.46	<b>36.39</b>	<b>58.09</b>	<b>49.96</b>	<b>32.19</b>	<b>46.63</b>	<b>41.22</b>
15	<b>32.61</b>	<b>51.63</b>	<b>44.50</b>	36.18	<b>58.09</b>	49.88	<b>32.19</b>	<b>46.63</b>	<b>41.22</b>
20	<b>32.61</b>	<b>51.63</b>	<b>44.50</b>	36.18	<b>58.09</b>	49.88	<b>32.19</b>	<b>46.63</b>	<b>41.22</b>
25	<b>32.61</b>	<b>51.63</b>	<b>44.50</b>	36.18	<b>58.09</b>	49.88	<b>32.19</b>	<b>46.63</b>	<b>41.22</b>
30	<b>32.61</b>	<b>51.63</b>	<b>44.50</b>	36.18	<b>58.09</b>	49.88	<b>32.19</b>	<b>46.63</b>	<b>41.22</b>

Table 8 presents the precision for the second setting also, in which plWordNet was merged with SUMO ontology, but in this case, only nodes belonging to the plWordNet were initialised. All results, which was obtained for nouns for *PPR* and *PPR\_W2W* have a higher disambiguation precision than in the baseline setting, cf Table 6.

The performance of a two-step disambiguation process, cf Sec. 4.2 is given in Table 9. The precision is lower than in both one-step settings. The decrease was caused by the fact that errors were propagated and multiplied from the first step to the second.

The last Table 10 contains summary of all results. It can be noticed that the highest results for nouns were obtained for PLWN + SUMO approach. In contrast the best results for verbs were obtained for the baseline i.e. when plWordNet was used only.

**Table 8:** Precision of disambiguation process based on the plWordNet from the 5th November, 2014 merged with the graph built from SUMO ontology. Nodes belonging to the plWordNet only were initialised.

Iterations	PPR			PPR_W2W			STATIC		
	V	N	All	V	N	All	V	N	All
5	32.40	<b>55.55</b>	<b>46.87</b>	36.91	58.37	50.33	<b>32.19</b>	<b>46.63</b>	<b>41.22</b>
10	<b>32.71</b>	55.27	46.81	37.01	<b>58.44</b>	<b>50.41</b>	<b>32.19</b>	<b>46.63</b>	<b>41.22</b>
15	<b>32.71</b>	55.27	46.81	<b>37.07</b>	58.40	<b>50.41</b>	<b>32.19</b>	<b>46.63</b>	<b>41.22</b>
20	<b>32.71</b>	55.27	46.81	<b>37.07</b>	58.40	<b>50.41</b>	<b>32.19</b>	<b>46.63</b>	<b>41.22</b>
25	<b>32.71</b>	55.27	46.81	<b>37.07</b>	58.40	<b>50.41</b>	<b>32.19</b>	<b>46.63</b>	<b>41.22</b>
30	<b>32.71</b>	55.27	46.81	<b>37.07</b>	58.40	<b>50.41</b>	<b>32.19</b>	<b>46.63</b>	<b>41.22</b>

**Table 9:** Precision of a two-step disambiguation process, described in Sec. 4.2.

Iterations	PPR			PPR_W2W			STATIC		
	V	N	All	V	N	All	V	N	All
5	32.40	<b>35.29</b>	34.21	35.20	36.07	35.74	<b>32.19</b>	<b>46.63</b>	<b>41.22</b>
10	<b>32.97</b>	<b>35.29</b>	<b>34.42</b>	35.25	<b>36.13</b>	<b>35.80</b>	<b>32.19</b>	<b>46.63</b>	<b>41.22</b>
15	<b>32.97</b>	35.23	34.38	<b>35.30</b>	36.10	<b>35.80</b>	<b>32.19</b>	<b>46.63</b>	<b>41.22</b>
20	<b>32.97</b>	35.23	32.38	<b>35.30</b>	36.10	<b>35.80</b>	<b>32.19</b>	<b>46.63</b>	<b>41.22</b>
25	<b>32.97</b>	35.23	32.38	<b>35.30</b>	36.10	<b>35.80</b>	<b>32.19</b>	<b>46.63</b>	<b>41.22</b>
30	<b>32.97</b>	35.23	32.38	<b>35.30</b>	36.10	<b>35.80</b>	<b>32.19</b>	<b>46.63</b>	<b>41.22</b>

**Table 10:** Summary of all results. Resources used in the initial vector are underlined

Approach	PPR			PPR_W2W			STATIC		
	V	N	All	V	N	All	V	N	All
PLWN	<b>34.06</b>	53.06	45.94	<b>37.69</b>	57.66	50.16	<b>34.11</b>	46.16	<b>41.65</b>
<u>PLWN</u> + SUMO	32.61	51.63	44.50	36.39	58.09	49.96	32.19	<b>46.63</b>	41.22
<u>PLWN</u> + SUMO	32.71	<b>55.55</b>	<b>46.87</b>	37.07	<b>58.44</b>	<b>50.41</b>	32.19	<b>46.63</b>	41.22
Two-step WSD	32.97	35.29	34.42	35.30	36.13	35.80	32.19	<b>46.63</b>	41.22

## 6. Conclusions and future work

We analysed three different variants of a weakly supervised WSD algorithm, based on the Page Rank in three settings of the lexical knowledge base. For the construc-

tion of the lexical knowledge base we used plWordNet (from 5th Nov., 2014) and the SUMO ontology, see Sec. 4.1.

In the first setting and as a baseline, we used only plWordNet. In the second setting both resources were merged using the existing mapping of plWordNet onto SUMO ontology. In this approach, firstly nodes belonging to plWordNet and the SUMO ontology were initialised, where secondly only nodes belonging to plWordNet were initialised. In the last setting, a two-step method for WSD was applied. This approach was described in Sec. 4.2.

The comparison between the baseline and the second setting shows that the combination of the two lexical knowledge bases, namely plWordNet and SUMO ontology, improve results for nouns. We can also noticed that a slightly higher results were obtain for PLWN + SUMO than for PLWN + SUMO approach. However, disambiguation precision do not increased for verbs. This situation was due to the fact that only nouns from plWordNet had links into the SUMO ontology.

Unfortunately, in the two-step WSD method, the results from the first phase heavily influenced on the performance in the fine-grained disambiguation step. As a consequence, the precision of the second step decreased. The coarse grained disambiguation do not work according to our expectations, as the number of SUMO concepts is quite large (close to the number of plWordNet senses!) and the SUMO network is small. The small network of connections do not provide good coverage for the occurrence context. One of the possible solution in future is to replace SUMO ontology by the WordNet Domains. It is due to the fact that WordNet Domains contains only 168 concepts/domains. Another possible direction is using directly SUMO to make decision about the domain of the disambiguated text. The disambiguation process may be carried out within the recognised domains.

We can conclude that Page Rank based WSD algorithms are promising solution for a language with as large lexical semantic resource as plWordNet. The algorithms could be easily adapted to Polish. Their biggest advantage is their ability to resolving ambiguity for as many lemmas as they are described in the lexico-semantic network.

Comparison of different types of Page Rank based WSD, we found out that Personalized Word-to-Word approach gives the best results, but for the price of the much longer processing time. This problem can be solved by parallelization of the disambiguation process for different ambiguous words from the context.

Combining lexical knowledge bases has improved results. In the future it should be considered whether the introduction of weights for nodes coming from different sources would improve the quality of disambiguation process also. Moreover, weights can be assigned not only to nodes, but also to the relation links, i.e. the graph edges. In the future, we plan also to exchange synsets as node by lexical units as the graph nodes.

## References

- Agirre, E., López de Lacalle, O., & Soroa, A. (2014). Random walks for knowledge-based word sense disambiguation. *Computational Linguistics*, 40(1), 57–84. [http://doi.org/10.1162/COLI\\_a\\_00164](http://doi.org/10.1162/COLI_a_00164)
- Agirre, E., & Soroa, A. (2009). Personalizing PageRank for word sense disambiguation. In *Proceedings of the 12th Conference of the European Chapter of the Association for Computational Linguistics, EACL '09* (pp. 33–41). Stroudsburg, PA: Association for Computational Linguistics. <http://doi.org/10.3115/1609067.1609070>
- Baś, D., Broda, B., & Piasecki, M. (2008). Towards word sense disambiguation of Polish. In *Proceedings of the International Multiconference on Computer Science and Information Technology — 3rd International Symposium Advances in Artificial Intelligence and Applications (AAIA '08)* (pp. 65–71).
- Brin, S., & Page, L. (1998). The anatomy of a large-scale hypertextual web search engine. *Computer Networks and ISDN Systems*, 30(1–7), 107–117. [http://doi.org/10.1016/S0169-7552\(98\)00110-X](http://doi.org/10.1016/S0169-7552(98)00110-X)
- Broda, B., Marcińczuk, M., Maziarz, M., Radziszewski, A., & Wardyński, A. (2012). KPWr: Towards a free corpus of Polish. In N. Calzolari, K. Choukri, T. Declerck, M. U. Doğan, B. Maegaard, J. Mariani, ... S. Piperidis (Eds.), *Proceedings of the Eight International Conference on Language Resources and Evaluation (LREC'12)*. Istanbul: European Language Resources Association (ELRA).
- Broda, B., Piasecki, M., & Szpakowicz, S. (2010). Extraction of Polish noun senses from large corpora by means of clustering. *Control and Cybernetics*, 39, 401–420.
- Brown, P. F., Pietra, S. A. D., Pietra, V. J. D., & Mercer, R. L. (1991). Word-sense disambiguation using statistical methods. In *Proceedings of the 29th Annual Meeting on Association for Computational Linguistics* (pp. 264–270). Stroudsburg, PA: Association for Computational Linguistics. <http://doi.org/10.3115/981344.981378>
- Fellbaum, C. (Ed.). (1998). *WordNet: An electronic lexical database (language, speech, and communication)*. Cambridge: MIT Press.
- Gale, W. A., Church, K. W., & Yarowsky, D. (1992). A method for disambiguating word senses in a large corpus. *Computers and the Humanities*, 26(5–6): 415–439. <http://doi.org/10.1007/BF00136984>
- Kędzia, P., & Piasecki, M. (2014). Ruled-based, interlingual motivated mapping of plwordnet onto sumo ontology. In N. Calzolari, K. Choukri, T. Declerck, H. Loftsson, B. Maegaard, J. Mariani, ... S. Piperidis (Eds.), *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, Reykjavik, Iceland. European Language Resources Association (ELRA).
- Lee, Y. K., & Ng, H. T. (2002). An empirical evaluation of knowledge sources and learning algorithms for word sense disambiguation. In *Proceedings of the ACL-02 Conference on Empirical Methods in Natural Language Processing* (Vol. 10, pp. 41–48). Stroudsburg, PA: Association for Computational Linguistics. <http://doi.org/10.3115/1118693.1118699>
- Lesk, M. (1986). Automatic sense disambiguation using machine readable dictionaries: How to tell a pine cone from an ice cream cone. In *ACM, Proceeding SIGDOC '86: Proceedings of the 5th Annual International Conference on Systems Documentation* (pp. 24–26). ACM Press. <http://doi.org/10.1145/318723.318728>
- Maziarz, M., Piasecki, M., Rudnicka, E., & Szpakowicz, S. (2013). Beyond the transfer-and-merge wordnet construction: PlWordNet and a comparison with Wordnet. In *Recent Advances in Natural Language Processing, RANLP 2013, 9–11 September, 2013, Hissar, Bulgaria* (pp. 443–452).

- Maziarz, M., Piasecki, M., & Szpakowicz, S. (2012). Approaching plWordNet 2.0. In *Proceedings of the 6th Global Wordnet Conference, Matsue, Japan*.
- Mihalcea, R., Tarau, P., & Figa, E. (2004). PageRank on semantic networks, with application to word sense disambiguation. In *Proceedings of the 20th International Conference on Computational Linguistics, COLING '04*. Stroudsburg, PA: Association for Computational Linguistics. <http://doi.org/10.3115/1220355.1220517>
- Młodzki, R., & Przepiórkowski, A. (2009). The WSD development environment. In Z. Vetulani (Ed.), *LTC* (pp. 224–233). Berlin: Springer. (Lecture Notes in Computer Science, 6562)
- Ng, H. T., & Lee, H. B. (1996). Integrating multiple knowledge sources to disambiguate word sense: an exemplar-based approach. In *Proceedings of the 34th Annual Meeting on Association for Computational Linguistics* (pp. 40–47). Stroudsburg, PA: Association for Computational Linguistics. <http://doi.org/10.3115/981863.981869>
- Niles, I., & Pease, A. (2003). Linking lexicons and ontologies: Mapping WordNet to the suggested upper merged ontology. In *Proceedings of the 2003 International Conference on Information and Knowledge Engineering (IKE 03) Las Vegas* (pp. 412–416).
- Pantel, P. A. (2003). *Clustering by Committee* (Unpublished PhD thesis). University of Alberta Edmonton, Alta., Canada.
- Patwardhan, S., Banerjee, S., & Pedersen, T. (2003). Using measures of semantic relatedness for word sense disambiguation. In A. Gelbukh (Ed.), *Computational linguistics and intelligent text processing* (pp. 241–257). Berlin: Springer. (Lecture Notes in Computer Science, 2588)
- Pease, A., & Fellbaum, C. (2010). Formal ontology as interlingua: The SUMO and WordNet linking project and global WordNet. In C.-R. Huang, N. Calzolari, A. Gangemi, A. Oltramari, & L. Prévot (Eds.), *Ontology and the lexicon: A natural language processing perspective*. Cambridge: Cambridge University Press. (Studies in Natural Language Processing).
- Pease, A. (2011). *Ontology: A practical guide*. Angwin: Articulate Software Press.
- Piasecki, M., Szpakowicz, S., & Broda, B. (2009). *A wordnet from the ground up*. Wrocław: Oficyna Wydawnicza Politechniki Wrocławskiej.
- Piasecki, M., Szpakowicz, S., Fellbaum, C., & Pedersen, B. (2013). Introduction to the special issue: On wordnets and relations. *Language Resources and Evaluation*, 47(3), 757–767. <http://doi.org/10.1007/s10579-013-9247-y>
- Rudnicka, E., Maziarz, M., Piasecki, M., & Szpakowicz, S. (2012). A strategy of mapping Polish Wordnet onto Princeton Wordnet. In *Proceedings of COLING 2012: Posters* (pp. 1039–1048). Mumbai, India: The COLING 2012 Organizing Committee.
- Schapire, R. E., & Singer, Y. (1999). Improved boosting algorithms using confidence-rated predictions. *Machine Learning*, 37(3), 297–336.
- Stevenson, M., Agirre, E., & Soroa, A. (2012). Exploiting domain information for word sense disambiguation of medical documents. *Journal of the American Medical Informatics Association*, 19(2), 235–240. <http://doi.org/10.1136/amiajnl-2011-000415>
- Wiriathamabhum, P., Kijirikul, B., Takamura, H., & Okumura, M. (2012). *Applying deep belief networks to word sense disambiguation*. arXiv preprint arXiv:1207.0396.
- Yarowsky, D. (1994). Decision lists for lexical ambiguity resolution: Application to accent restoration in Spanish and French. In *Proceedings of the 32nd annual meeting on Association for Computational Linguistics* (pp. 88–95). Stroudsburg, PA: Association for Computational Linguistics. <http://doi.org/10.3115/981732.981745>
- Yarowsky, D. (2010). Word sense disambiguation. In R. Herbrich & T. Graepel (Eds.), *Handbook of natural language processing* (pp. 315–338). CRC Press.

Yarowsky, D., & Florian, R. (2002). Evaluating sense disambiguation across diverse parameter spaces. *Natural Language Engineering*, 8(4), 293–310. <http://doi.org/10.1017/S135132490200298X>

Yngve, V. H. (1955). Syntax and the problem of multiple meaning. In W. N. Locke (Ed.), *Machine translation of languages* (pp. 208–226). London: The Technology Press of The Massachusetts Institute of Technology and John Wiley & Sons, Inc.

---

### Acknowledgment

Work financed by the Polish Ministry of Science and Higher Education, a program in support of scientific units involved in the development of a European research infrastructure for the humanities and social sciences in the scope of the consortia CLARIN ERIC and ESS-ERIC, 2015–2016.

The authors declare that they have no competing interests.

The authors' contribution was as follows: concept of the study: Paweł Kędzia, Maciej Piasecki, Marlena Orlińska; data analyses: Paweł Kędzia, Maciej Piasecki, Marlena Orlińska; the writing: Paweł Kędzia, Maciej Piasecki, Marlena Orlińska.

This is an Open Access article distributed under the terms of the Creative Commons Attribution 3.0 PL License (<http://creativecommons.org/licenses/by/3.0/pl/>), which permits redistribution, commercial and non-commercial, provided that the article is properly cited.

© The Authors 2015

Publisher: Institute of Slavic Studies, PAS, University of Silesia & The Slavic Foundation