CMST 25(1) 41–56 (2019) DOI:10.12921/cmst.2018.0000051

Paintball – Automated Wordnet Expansion Algorithm based on Distributional Semantics and Information Spreading

Maciej Piasecki

Faculty of Computer Science and Management, G4.19 Research Group
Wrocław University of Science and Technology
Wyb. Wyspiańskiego 27, 50-370 Wrocław, Poland

E-mail: maciej.piasecki@pwr.wroc.pl

Received: 07 November 2018; revised: 30 December 2018; accepted: 03 January 2019; published online: 29 January 2019

Abstract: plWordNet has been consequently built on the basis of the corpus-based wordnet development method. As plWordNet construction had started from scratch it was necessary to find a way to reduce the amount of work required, and not to reduce the quality. In the paper we discuss the gained experience in applying different tools based on Distributional Semantics methods to support the work of lexicographers. A special attention is given to the Paintball algorithm for semi-automated wordnet expansion and its application in the WordnetWeaver system.

Key words: wordnet, lexical semantic network, automated wordnet expansion, natural language engineering, linguistic knowledge extraction, WordNet, plWordNet

I. DREAMS FULFILLED

After more than 13 years of continuous development of plWordNet [1] we have achieved much more with the version 4.0 emo than we could have dreamt when starting its construction from scratch in the year 2005. plWordNet 4.0 emo provides a very comprehensive coverage of the Polish lexical system and Polish large corpora: 221,972 synsets, 191,393 lemmas and 287,880 lexical units[1] (henceforth LUs) described by about 650,000 relation links. From the very beginning we decided that we could not follow the *transfer approach* [2] and [3] to wordnet development, if we wanted to make plWordNet a faithful description of the Polish lexical system [4]. Due to the lack of any publicly available electronic dictionaries of Polish on that time[2], we also also were not able to apply a typical *merge approach* [3] based on the utilisation of data from the existing monolingual dictionaries. Thus, we proposed a *wordnet-based development method* [5] in which a large corpus is the main source of the linguistic knowledge. However, a wordnet must be very large to have a practical impact on the Natural Language Engineering (henceforth NLE) (also called Natural Language Processing) and other applications – its large size is almost a basic requirement. One must go towards a comprehensive coverage of the language data by a wordnet and in the same time make it a resource providing a proper distinction and characterisation of lexical meanings. Wordnets have to compete with statistical models (like often used word embeddings) that are relatively easily extracted from the very large corpora. Thus, a wordnet must be a trustworthy language resource of high quality, built manually to the possible largest extent and should include a thorough description of language

[1] A lexical unit can be technically defined as a triple: ⟨Part of Speech, lemma, sense identifier⟩, where a *lemma* is a basic morphological form arbitrarily selected to represent a set of word forms that differ only with respect to the values of the grammatical categories like number, gender, case etc., but they share the same meaning and, approximately, the morphological stem.

[2] That is still the case if we take into account semantic dictionaries built by lexicographic teams.

data, but also appropriately built abstractions over the language data.[3] We were aware that in order to fulfil these goals an enormous amount of work would be required. We had to decrease this amount as far as possible, while not reducing responsibility of lexicographers – wordnet editors – for every single element of the wordnet structure. Thus, we have planned, developed and applied a set of tools and systems for the extraction of linguistic knowledge from large corpora and support wordnet editors in their decisions. Distributional Semantics methods, e.g. [6], play an important role in this set, as a basis for semantic exploration of the corpora. The most sophisticated tool is *WordnetWeaver* – a system for semi-automated expansion of a wordnet that can utilise several different knowledge sources including Measures of Semantic Relatedness produced by the Distributional Semantics methods, e.g. word embeddings [7].

In the paper we present the corpus-based wordnet development process with its application to plWordNet and with the main focus given to the role of Distributional Semantics methods in this process. However, the central point is the discussion of *Paintball* – an algorithm for semi-automated wordnet expansion. Its evaluation on the two largest world wordnets illustrates the potential of the proposed solution. Finally, we report on the experience which has been collected for more than ten years on the application of a whole set of tools during the wordnet development process.

II. CORPUS-BASED WORDNET DEVELOPMENT

A manually built wordnet provides a description of relatively high quality for lemmas and their senses, i.e. LUs. Thus, the lager number of words is covered by a wordnet in a processed text, the better this wordnet is for NLE applications. However, adding new lemmas to a wordnet is costly. One needs to decide what is worth adding first. There are two overlapping criteria:

- natural limits of the general lexical system (i.e. the extent of the lexicalisation), and
- the language usage.

Concerning the former the main issue is the distinction between the *general* vs *specialist* language, as well the issue of Proper Names (PNs). Concerning the latter the key point is selection of a representative corpus and the decision concerning a frequency threshold for lemmas to be included into the given wordnet.

As Proper Names are an open and very dynamic class, and are tightly connected to knowledge representation, in the very beginning we decided to keep them out of plWordNet by default [4]. The only exception was made for PNs that are derivational bases for common nouns and adjectives that are frequent enough. Because plWordNet have been consequently constructed on the basis of the *corpus-based wordnet development method* proposed by us [8], lemma frequency in a large corpus is always the basic criterion for their selection. As we intended to build a wordnet of the general Polish language, we have been developing plWordNet Corpus (henceforth *plWNC*) to the largest possible size. For the development of plWordNet 3.1 emo we collected more than 4 billion words in the 10th version of plWNC[4].

Wordnet construction should follow a typical process of dictionary writing [13], cf [14, 15]. Lexicography distinguishes four phases: data collection, selection, analysis and presentation [13]. In the plWordNet project, language technologies support all four phases. Professional linguists under the supervision of senior coordinators work with *WordnetLoom* 1.0 [16] and 2.0 [17] – a distributed system of the client-server architecture for wordnet editing, which offers functionality of graph-based visual browsing and editing of wordnet relations concurrently by a group of lexicographers. Many semi-automatic tools have been integrated into it.

In the *data collection phase*, a large corpus is essential [18]. plWNC 1.0 included only 0.5 billion words. With the growing plWordNet size we have been gradually increasing the size of plWNC to beyond 4 billion words with version 10. Thus, instead of attacking the problem of the corpus representativeness, we try to diminish it by collecting a very large set of texts available for Polish, and in this way balancing its content. Moreover, as it is explained below, lexicographers are not confined in their editing decisions only to the material from the collected corpus. They can expand the lemma set, e.g. according to their language competence or available dictionaries.

In the *data selection phase*, the most frequent lemmas are chosen [19]. With the growing size of plWordNet we faced the problem of the lack of coverage by morphological analysers[5]. As a result the selection of the new lemmas to be included in the wordnet from the list of the most frequent tokens must be performed manually by linguists.

Next, a Measure of Semantic Relatedness (MSR) is extracted from the corpus by the *SuperMatrix* system [20] or, recently, on the basis of word embeddings, namely, constructed by the *word2vec* [21] and *fastText* [22] algorithms[6]. The constructed MSR is used as a basis for clustering the selected new lemmas into packages including around 100,

[3] A wordnet provides only a partial description of the lexical meaning by a couple of dozens of lexico-semantic relations that are abstractions over the continuity of the lexico-semantic similarities and differences.

[4] It consists of IPI PAN Corpus [9], the first annotated corpus of Polish, National Corpus of Polish [10], Polish Wikipedia (from 2016), *Rzeczpospolita* Corpus [11] – a corpus of electronic editions of a Polish newspaper from the years 1993-2003, supplemented with texts acquired from the Web – only texts with a small percentage of words unknown to a very comprehensive morphological analyser Morfeusz 2.0 [12] were included; duplicates were automatically eliminated from the merged corpus.

[5] However, also the problem of over-generation of morphological analysers, i.e. recognition of non-existing words.

[6] A presentation of the constructed word embeddings on the basis of plWNC and their evaluation is given in [23]. The embeddings are published in the CLARIN-PL Repository [24]. The repository includes also more models for Polish constructed on the basis of plWNC 10.0, e.g. [25] and [26].

up to 200 lemmas with the help of the CLUTO system [27]. They are intended to be work units assigned to individual lexicographers during wordnet expansion. A single package includes lemmas that are semantically related. However, the description provided by MSRs is not precise and complete, especially for lemmas occurring less than 200 times per 1 billion tokens. So, a typical package expresses 2-3 main semantic topics on average. Nevertheless, we discovered that such a way of semantically motivated work assignment to individual lexicographers is superior to semantically non-informed methods, e.g., on the basis of a simple alphabetic division. As a result, a lexicographer can concentrate on a limited number of semantic domains while working on lemmas from a package.

The primary task of lexicographers is to recognise lexico-semantic relations for a new lemma and to assign its LUs (senses) to synsets: new or already existing ones. Lexicographers are supported in these hard tasks by several software tools.

First, a word-sense induction algorithm LexCSD [28] is applied to selects up to $k = 10$ examples of word usage, that are intended to represent different meanings.[7] LexCSD works on the basis of simple co-occurrence statistics. We plan to explore more sophisticated representation of use example in a form of word embeddings or Deep Learning based sentence representations. Usage examples appeared to be especially important in the case of adjectives and verbs that seemed to be less thoroughly described in the existing dictionaries of Polish.

The editors can also browse plWNC using the Poliqarp interface [29] or recently KonText[8] [30].

Lists of the lemmas that are most semantically related to new lemmas, henceforth called k-nearest neighbour lists (k-NNL) appeared also to be a tool for a kind of semantic exploration. k-NNL with typically $k = 20$ can include instances of different types of lexico-semantic relations. We tried to tune MSR extraction methods in such a way that the ratio of relation instances on a k-NNL is maximised, e.g. [31]. Word embeddings seem to provide even better results, see [23].

Finally, editing is supported by *WordnetWeaver* [8], a system for semi-automated wordnet expansion that was implemented as an extension to the *WordnetLoom*. For a new lemma *WordnetWeaver* suggests up to several places that seem to match well its meanings and are suitable for connecting a given lemma to the lexico-semantic net. Hints generated by *WordnetWeaver* usually yield new distinguished senses. The suggestions are generated with the help of the *Paintball* algorithm, presented in Sec. III [32]. Every suggested LU (a new sense) is visually presented as a subgraph of the hypernymy structure. Each subgraph is intended to illustrate a given identified new sense, and is presented in *WordnetWeaver* in a way enabling its immediate editing. The lexicographer can freely introduce changes to the wordnet relation structure, e.g., inspired by the suggestion.

The corpus browser, usage examples and *WordnetWeaver* enable increasingly complex language processing: from simple queries in the plWordNet Corpus, through the presentation of a small list of disambiguated usage examples, to highly sophisticated lemma-placement suggestions.

Apart from primary sources and automated tools, the editors are encouraged to look up words and their descriptions in the available Polish dictionaries, thesauri, encyclopaedias, lexicons, and on the Web. In the end, the new lemma and all its LUs are integrated with plWordNet and displayed in *WordnetLoom*.

Intuition matters despite even the strictest definitions and tests, so one cannot expect two linguists to come up with the same wordnet structure. In corpus-building it is feasible to have two people annotate the same portion and adjudicate the effect, but wordnet development is a more complicated matter. That is why we have a three-step procedure:

1. wordnet editing by a linguist (described above),
2. wordnet verification by a coordinator (a senior linguist),
3. and (wordnet revision, again by a linguist supported by a diagnostic system [33].

Full verification would be too costly, so it is done on (large) samples of the editors' work. A coordinator corrects errors, adjust the wordnet editor's guidelines,[9] and initiates revision during which systematic errors are corrected and the wordnet undergoes synset-specific modification.

III. PAINTBALL ALGORITHM

III. 1. The idea of information spreading

A corpus is a very imprecise source of lexical semantics knowledge. Information about word senses is always partial in a corpus: not all senses occur, for most senses we cannot collect enough diversified use contexts. Stylistic factors, (e.g. avoiding word repetition), metaphor, lack of precision of, writers etc., cause that information extracted from corpora may include many errors (except well-defined domains) and suggest accidental semantic associations between words, e.g. accidental words associated by a classifier trained on Wikipedia for *feminism*: *proviso*, *antecedent*, *first half*, etc. As a result, knowledge describing lexico-semantic relations extracted from corpora by any method is always partial and prone to error (even in the case of the best performing extraction methods). If we cannot avoid errors, we can to try to compensate them by combining and confronting word associations suggested by several extraction methods. Due to information partiality, conclusions about the lack of associations between a pair of words cannot be drawn reliably.

[7] Usage examples, welcome by the editors, help them distinguish senses [15].
[8] https://github.com/czcorpus/kontext
[9] That is a 50-80-page documents per each Part of Speech.

In spite of the differences between the relation extraction algorithms, their results can be uniformly represented as sets of triples: $\langle x, y, w\rangle$, where y is a word included in the wordnet to be expanded, x is a 'new word', i.e. not yet described in the wordnet and $w \in R$ is a weight of real value. We call such a set a *knowledge source* (henceforth KS) that has been extracted by some method. A triple $\langle x, y, w\rangle$ included in a knowledge source K informs that x is semantically associated with y according to the method used to extract K and w describes the strength of this association. In many approaches, e.g. [34], weights are interpreted as probabilities. However, many relation extraction methods are not based on statistics, and word-pairs extracted by them cannot be described by probabilities, e.g. the majority of pattern-based methods extract word pairs on the basis of a several or even singular occurrences (but the recall is still low). Nevertheless, as we need to 'squeeze' all possible lexical information from the text, we have to try to utilise such non-probabilistic KSs, too. Thus, we assume that w is a value of *support* for the semantic association expressed by the given word pair. KSs are extracted by the vast majority of methods for words, not word senses. As there are no robust, large scale methods for extracting associations between word senses (especially new ones, not yet covered by a wordnet), we do not consider such possibility here.

A triple $\langle x, y, w\rangle$ from a KS K_i suggests linking x to synsets including y. If K_i represents synonymy or hypernymy, the triple defines a place for x in the wordnet hypernymy structure. However, there are two serious problems: x and y can have several senses each, and the triple can express some error, e.g. the link may not be a close one, but, instead, based on metonymy, metaphor, or even driven by situational associations. Concerning the first, the triple suggests linking x to different senses of y represented by the synsets including y – each synset describes a possible meaning of x, but we do not know which of them is valid, e.g. triples generated by PWE hypernymy classifier [34] ⟨*feminism*, *movement*, 1.0⟩, ⟨*feminism*, *theory*, 0.948⟩, ⟨*feminism*, *politics*, 0.867⟩, ⟨*feminism*, *feminist*, 0.201⟩, ⟨*feminism*, *liberalism*, 0.207⟩, ⟨*feminism*, *pacifism*, 0.208⟩, etc.

With respect to the directness of links suggested by KSs, apart from the clearly wrong, accidental triples, KSs very often include too general suggestions, e.g. y can be in fact an indirect hypernym of x or y can be associated with x by a kind of fuzzynymy instead of describing the appropriate location for an x sense in the wordnet structure. Combining information coming from several different triples describing x may solve both problems by identifying those parts of the wordnet hypernymy structures that are best supported by the evidence in KSs.

Paintball [32] is a wordnet expansion algorithm which is based on a general model of *spreading activation* [35–37]: the support from KS triples is the initial, direct activation which is next spread along the structure of the wordnet relations. The *Paintball* algorithm is based on a metaphor of semantic support for x resembling drops of liquid paint that initially fall on some wordnet graph nodes (i.e. synsets), following KSs, and next the paint starts spreading over the graph. Those regions that represent the highest amounts of paint after the spreading has been completed represent possible senses of x and include potential locations for x senses.

The spreading model is motivated by the nature of KSs. KSs are typically extracted to represent selected wordnet relations, e.g. synonymy and hyper/hyponymy, but in practice KS triples represent a whole variety of relations, e.g. indirect hypernymy, but also meronymy, co-hyponymy (cousin or coordinate) or just stronger semantic association. A KS element $\langle x, y\rangle$ can suggest linking an x sense directly to a y sense by synonymy, but also indirectly by some other relation, depending on the nature of the method applied to create a given KS. KSs based on Distributional Semantics do not specify this relation. Pattern-based KS are mostly focused on hypernymy but their precision is always limited. So, real attachment places for an x sense can be somewhere around y synsets assuming that they are semantically similar to y and the given KS does not include too serious errors or does not describe too fuzzy semantic associations.

On the basis of the assumption that semantic similarity between a synset S, which is a proper attachment place for x, and y (suggested by the KS) is correlated with the length of the shortest path in the wordnet graph linking S and a synset of y, we can expect that the proper attachment places for the senses of x are accessible from the synsets of y via relatively short paths in the wordnet graph. Such subgraphs describe expected types of errors included in the KSs and their shape should depend on the nature of a given KS. For instance, as it is easier to mismatch synonymy and hypernymy than hypernymy and antonymy, the subgraph is more likely to include hypo/hypernymic paths than paths including antonymy links, too. As we expect that KSs of some minimal accuracy include a large number of minor errors[10], we need to consider only subgraphs with limited length of paths corresponding to less serious errors. Thus, each KS triple marks whole wordnet subgraphs as potential attachment places for the senses of x.

Spreading the activation model follows a general scheme, e.g. [37]:

$$a'_i = \lambda u_i + \mu f(\sum w_{j,i} \times a_j)$$

where a_j is activation of the node j, a'_i – activation in the next step, u_i is the initial activation of i, $w_{i,j}$ – the weight for the link $\langle j, i\rangle$ (an input link to i), λ, μ are parameters representing the amount of *initial activation* and *activation decay*, respectively [38]; function f provides a possibility to define a non-linear dependency of the new activation value on the input values.

[10] In the sense of a semantic difference between the suggested place and the proper one.

In our approach we identify activation with semantic support for x and represent it by a real value. The initial activation is called *direct activation* while support coming from other nodes is called *indirect activation*. The λ parameter is set to 1, if the direct activation expresses the information included in KSs. Indirect activation is a result of our attempts to compensate errors of KSs and resolve the ambiguity of the lemma-based information delivered in them.

Most frequent wordnet relations link synsets, e.g. *hypo/hypernymy* or *mero/holonymy*, but in every wordnet there are also many relations linking directly LUs, e.g. *antonymy*. In order to use the whole wordnet graph structure, not only defined by synset relations, we treat LUs as nodes and synset relations are mapped to relations between all LUs from the linked synsets.

In a way typical for spreading activation models, the activation decay parameter $\mu \in [0, 1)$ decreases semantic support with every link on the path. However, due to the likeliness of KS error types, not all links should be treated in the same way. In *Paintball* the activation decay value depends also on the link types. Following [39], that part of the decay dependent on the link type is represented by two functions: *transmittance* and *impedance*.

Transmittance is a function of a general scheme:
lexico-semantic relation $\times R \to R$
and describes the ability of links to transmit support.

Link-to-link connection is characterised by the *impedance* function of the general scheme:

$$\textit{relation pair} \times R \to R.$$

The impedance describes how much indirect activation can be transferred through the given connection, e.g. the transmission of activation through *holonymy–meronymy* would mean that the direct activation assigned to the whole (a holonym) via a part (a meronym) could be attributed to another whole (its second holonym), e.g. *car–holonymy–windscreen–meronymy:substance–glass*: indirect activation could go from *car* to *glass* that is clearly too far. By an appropriate impedance function we can reduce the spreading or block it, i.e. we can shape the considered part of the wordnet graph.

In *Paintball* activation spreading is analysed in the graph of LUs, but the final results are mapped back to synsets. On the synset level subgraphs of synsets with significant activation are identified as descriptions of different x.

III. 2. Wordnet model

The plWordNet structure has been generally inspired by the structure of Princeton WordNet. However, it expresses several very significant differences when it comes to the underlying model. First of all, the plWordNet model is consequently based on LUs (lexical units) as the basic building blocks, cf [8, 40] and especially [41]. In short, synset groups LUs that share lexical-semantic relations (of selected types called constitutive relations). Thus, a relation link between two synsets (called *conceptual* in Princeton WordNet) can be considered as a notational abbreviation for the relation links between all respective pairs of LUs belonging to the two synsets. Each synset can be easily replaced in the plWordNet structure by its set of LUs and their relations without losing any information. As a result, the plWordNet structure can be presented as a graph whose nodes correspond to LUs and arcs represent instances of lexico-semantic relations.

The vast majority of the wordnet expansion algorithms proposed in literature are based on the hypernymy structure linking synsets and do not utilise other relations. By taking into account relations other than hypernymy and hyponymy in *Paintball*, we want to explore better the knowledge encoded in the wordnet structure as well as acquired from corpora in a form of the various knowledge sources. A typical description of a wordnet as a graph of synsets makes a description of an expansion algorithm difficult in the case it explores relations of different types. So, for the sake of further discussion we are going to model a wordnet as a set of lexico-semantic relations defined on the universe of LUs. Next, it will be used to discuss *Paintball* and its variants in detail.

A wordnet is :

$$WN = \langle J, \mathbf{S}, L, f_{Lem} \rangle \tag{1}$$

where:

- J is a set of lexical units,
- $\mathbf{S} \subseteq 2^{J^2}$ – a set of lexico-semantic relations[11] defined on J,
- L – a set of lemmas of a given natural language,
- $f_{Lem} : J \to L$ – a function[12] which assigns a lemma for every LU from J.

A graph in which the nodes are from J and the arcs from $\mathbf{S}$ will be called a *wordnet graph*.

In order to simplify the description of the algorithm we assume that all arcs of the wordnet graph are directed, i.e. the relations from $\mathbf{S}$ are antisymmetric. If a lexico-semantic relation is not directed on the level of the linguistic description, it can be represented in a modified wordnet graph by a pair of relations describing the respective directions.

The model (1) does not include synsets as wordnet elements, because synsets can be derived in plWordNet on the basis of the constitutive lexico-semantic relations cf [5]. However, the lexicographic practice showed that some synsets are defined directly by lexicographers with the help of the synonymy notion characterised as mutual hypernymy.

[11] For any set A, 2^A means a set of all subsets of A, and A^2 is a Cartesian product generated from A.
[12] Defined for domain J

This happens in cases in which the defined constitutive relations are not sufficient for differentiating particular lexical meanings. This practice can be reflected in the wordnet model by distinguishing one relation from **S** as the synonymy relation: S_{syn}. In addition, it is also worth expanding the model (1) with several helpful elements:

$$WN = \langle J, \mathbf{S}, L, f_{Lem}, S_{Syn}, f_{Snst} \rangle \quad (2)$$

where

- $S_{Syn} \in \mathbf{S}$ is a constant representing a distinguished relation of synonymy,
- and $f_{Snst} : J \rightarrow 2^J$, such that $f_{Snst}(j) = \{j' : \langle j, j' \rangle \in S_{Syn}\}$ is a function which for a LU j returns a synset (i.e. a subset of J) which it belongs to.[13]

III. 3. Wordnet graph

In order to simplify the description of *Paintball* we assume that a wordnet has been converted to a graph of relations between LUs, prior to the application of the algorithm, i.e. according to the model (2) in which synsets are represented by the synonymy relation. In the case of plWordNet such an operation is trivial as synset relations (and synsets) are only notational abbreviations. In the case of other wordnets, e.g. Princeton WordNet conceptual relations (i.e. synset relations), cf [42], must be mapped onto the level of LUs that requires changing their types, but as such conceptual relations somehow mimic linguistic lexico-semantic relations (even preserving names), then this operation is also straightforward.

For the conversion of a wordnet to the graph of LU relations we assume that:

1. it is a directed graph with LUs as nodes and all synset relations mapped onto LUs,
2. a synset relation is mapped on all respective pairs of LU (i.e. the $n : m$ scheme),
3. a synset is represented by a pair of relations:
   - *synonymy* and *synonymy bis* – being mutually reverse,
   - and, in order to avoid several cycles inside a synset, its LUs are linked into a chain traversed in two directions by the two relations; the order[14] reflects the one defined by linguists in the description of synsets.

During the work on tuning the *Paintball* parameters we discovered that multiple connections between LU groups resulting from synsets[15] cause a kind of local amplification of activation being transferred through these clusters of connections. In this way a size of a synset has a significant influence on the work of the algorithm and larger synsets have a tendency to collect activation that blurs the global perspective on activation spreading. In order to cope with this problems, step 2 of the above conversion was changed to:

- after a synset has been converted to a synonymy chain of LUs, all relation links of this synset are mapped onto its head LU only.

where *a head LU* is the first LU in sequential representation of a synset stored in the wordnet database and presented visually to the user.

Activation coming to the synset is spread further through connections of the head LU, but also along the synonymy links and further via individual relations of LUs (i.e. the originating relations of LUs). However, every traversed link in a graph normally causes the decrease of the activation. In the case of a slightly accidental order of the synonymy chain this would not be justified. This phenomenon can be controlled, or even completely removed, by the appropriate setting of *transmittance*.

III. 4. Definitions

Let

- WN – a wordnet compatible with the model (2),
- $\mathbf{K}$ – a set of knowledge sources, where every K is a set of triples of the type: $L \times L \times R^+$ (R^+ is a set of non-negative real numbers),
- $\mathbf{Q} : J \times R^+$ – a matrix including activation values for the LUs,
- $\mathbf{F} : S \times R^+$ – a matrix storing activation values for synsets calculated on the basis of values for their LUs,
- x – a new lemma to be added to the wordnet structure,
- $T \subseteq J$ – a list of LUs to be processed,
- $\sigma : J \times L \rightarrow R^+$ – provides for a LU j and a lemma x (i.e. typically a new lemma to be added to a wordnet) complex initial activation of j as being semantically associated with x on the basis of all knowledge sources.

The function σ can be defined in many ways, but if we assume that knowledge sources are independent, then σ can be defined in a natural way as the sum of weights from the knowledge sources: $\sigma(j, x) = \sum_{K \in \mathbf{K}} K(lemma(j), x)$ where where $lemma(j)$ returns a lemma of the LU j.

Parameters:

1. μ – a decay factor, typical for spreading the activation scheme, defines what portion of activity is spread to the next LU (i.e. the next node), is applied with each traversed link, typically set in the range $(0, 1)$ and aimed at stopping the spreading in some distances measured by the number of links traversed,

[13] In every wordnet, including plWordNet, one synset represents one lexical meaning. Thus, a LU must belong to exactly one synset.

[14] A synset is a set of LUs but in most if not all wordnets it is presented to the users as a sequence and in such a form it is stored in the database. It is hard to find guidelines concerning the order in which LUs should be added to a synset, but in the case of plWordNet results of works on Word Senses Disambiguation suggest that this order is somehow correlated with salience of different lexical meaning or even their frequency, i.e. most frequent LUs seem to be added as first to synsets.

[15] Every LU from one synset is connected with every LU from the other synset.

2. τ_0 – a minimal activity level threshold,
3. ϵ – a stop threshold defining the minimal activity level for sustaining further spreading,
   let $\epsilon = \frac{\tau_0}{4}$
4. tau_4 – a strong support threshold,
5. $f_T : J^2 \times R^+ \rightarrow R^+$ – a function defining *transmitance* characteristic of a link, i.e. how the activation value is changed when spread through a given link, typically *transmittance* is defined for the whole relations that is described below,
6. $f_I : J^2 \times J^2 \times R^+ \rightarrow R^+$ – a function defining *impedance* characteristic of a connection of two links, typically is defined for relation pairs, see below.

III. 5. Algorithm

The algorithm works in four main steps preceded by the preparatory Step 0. First, the initial activation for LUs is calculated on the basis of KSs. Next, the local activation is recursively replicated from LUs to their local subgraphs of connected LUs and added to their activations, i.e. the activation values of the connected LUs are increased (according to the algorithm), but the activation of the source LU is not decreased. After replication-based spreading on the graph of LUs is completed (and the resulting total activation of LUs has been calculated), the activation for synsets is computed on the basis of the activation of their LUs[16]. Finally, connected wordnet subgraphs such that each synset in a subgraph has some significant activation level are identified, cf [39]. Such subgraphs are called *activation areas*. Top several activation areas with the highest activation values are selected as *attachment areas* – that represent descriptions of potential senses of x. In each attachment area, the synset with highest activation is a potential place to locate x sense, i.e. according to the algorithm a new LU for x can be added to this synset or to a synset linked with it by a short path of wordnet relations. Attachment areas are meant to be presented to linguists as explanations of the suggested meanings of x. The new LU is described by a subgraph to reflect the intrinsic errors of the input KSs.

For a new lemma x to be added to the wordnet:

Step 0 Construction a LU graph on the basis of the synset graph.

Step 1 Setting up the initial activation:

1. $\forall_{j \in J}.\mathbf{Q}[j] := \sigma(j, x)$
2. `for each` $j \in J$ `if` $\mathbf{Q}[j]) > \tau_0$
   $T := append(T, j)$
3. $T := sort\_descendingly(T)$

Step 2 Activation replication across the LU graph:

1. $k := head(T)$, next $T := tail(T)$
2. $actReplication(k, x, \sigma(k, x))$ – the activation for x is replicated from k onto the connected nodes
3. `if not` $empty(T)$ `then goto` 1

Step 3 Synset activation calculation:

1. `for each` s `in` Syn
   $\mathbf{F}[s] := synsetAct(s, \mathbf{Q})$

Step 4 Identification of attachment areas

1. Recognition of connected subgraphs in WN, such that $G_m = \{s \in Syn : \mathbf{F}[s] > \tau_3\}$
2. `for each` G_m $score(G_m) := \mathbf{F}[j_m]$, where $j_m = max_{j \in G_m}(\mathbf{F}(j))$
3. Return G_m, such that $score(G_m) > \tau_4$ as activation areas.

In Step 1 only nodes that represent some meaningful value of the initial (local) activation (τ_0) are added to the queue as the starting points for the replication in Step 2. The value of τ_0 depends on the KSs, but it can be set to the smallest weight value that signals good triples in the KS of the biggest coverage. All threshold values can be also automatically optimised, e.g., as in [43].

Activation replication

In Step 2 activation replication is run for nodes stored in the queue and is described by the $actReplication$ function taking on the input:

- j is a LU to be processed,
- x – a new lemma to be added,
- M is the activation value to be replicated.

In its description below, two auxiliary function are used:

- $dsc(j)$ returns the set of outgoing relation links,
- and $p|_1$ returns the first element of a pair, in the case of the wordnet graph this is the target node of a relation link.

The start of activation replication from a node j is defined as follows:

$actReplication(j, x, M)$:
1. `if` $M < \epsilon$ `then return`
2. `for each` $p \in dsc(j)$
   $actRepTrans(p, x, f_T(p, \mu * M))$

For each outgoing link a portion of the j activation is replicated according to the transmittance of the given link. Depending on the link type this portion can be smaller of larger (even nullified), see the discussion on transmittance implementation below.

The activation replication for each following node along the path of spreading goes as follows (where p is the incoming link):

[16] This step can be omitted in the case of lexical semantic networks without synsets.

$actRepTrans(p, x, M)$:
1. `if` $M < \epsilon$ `then return`
2. `for each` $p' \in dsc(p|_1)$
   $actRepTrans(p', x, f_I(p, p', f_T(p', \mu * M)))$
3. $\mathbf{Q}[p|_1] := \mathbf{Q}[p|_1] + M$

Incoming activation is stored in the given node and a part of it is replicated further, first of all according to the decay factor μ (a global factor), but also according to the link-wise defined transmittance function f_T (a local factor). They both jointly contribute to the activation decay. The replication stops when the incoming activation goes down below ϵ, i.e. a global stop condition.

In addition, the impedance function f_I is meant to be a factor controlling the direction of spreading or even shaping the spreading graph. It can block some pre-defined junctions of links of selected types or at least decrease the amount of activation going through link junctions of certain types, e.g. $\langle holonymy, antonymy \rangle$ The value of ϵ was heuristically set to $\tau_0/4$, but it can be obtained during optimisation, as all other parameters, cf [43]. The parameters μ and ϵ together control the maximal distance on which the initial activation of a node can influence its local subgraph.

Synset activation

In Step 3, activation for synsets is calculated on the basis of the activation for LUs included in them. It can be done in many different ways, e.g. starting with a simple sum over support values of LUs. However, the best results were obtained by using a function proposed by us in [8]:
$synsetSup(S, \mathbf{Q})$:
1. `sum` $:= \sum_{j \in S} \mathbf{Q}[j]$
2. `if` $\delta(1, \mathtt{sum}, |S|) > 0$ `then return sum`
   `else return 0`

where $\delta(h, n, s) =$
1. `1 if` $(n \geq 1, 5 * h \wedge s \leq 2) \vee (n \geq 2 * h \wedge s > 2)$
2. `else 0`

The idea is to expect more activation for larger synsets, but this dependency is not linear, as a larger synset very often includes many less frequent and worse described LUs. In Step 3, we also filter out synsets that do not have any local activation in order to preserve only the most reliable data.

Activation areas

Finally, in Step 4, activation areas (subgraphs) are identified with the help of a subset of wordnet relations, which includes all relations defining the basic wordnet structure, e.g. in some wordnets a synset can be linked by a relation different from hyponymy as its only relation. An activation area as a whole expresses a single location found by the algorithm for lemma x. Nevertheless, we also need to single out a particular synset from an activation area as an attachment point for a LU of x. Thus, we look for local maxima of the activation values inside activation areas and use these values as semantic activation for the whole attachment areas. *Paintball* is focused on supporting linguists and its recall is important, so up to max_{att} activation areas (according to their activation values) are finally returned as suggested *attachment areas*.

Transmittance and impedance implementation

All knowledge sources include errors. However, it is more likely that a knowledge source links together two semantically related lemmas, e.g. a hypernym and its hyponym, then completely unrelated. *Transmittance* is a property of a link, describes its ability to transmit activation between nodes, typically depends on the relation expressed by the link, and can deemphasise, or sometimes, emphasise activation: f_T(*relation*, *activation*) returns *modified activation*.

The transmittance function should be tuned for a particular application of *Paintball*, but as a general rule the transmittance function should produce higher values for those relations that are more likely to be expected as erroneous connections in knowledge sources. Moreover, transmittance can be used to influence the directions of activation spreading, and, e.g., to let more activation being passed to hypernyms, i.e. allowing for some generalisation.

If we restrict the transmittance to the scheme:

$$f_T(r, v) = 1 * v \tag{3}$$

it can be specified by the value of the coefficient v which is mostly defined in a relation-wise way. During the experiments we assumed the following values of v:

- *hypernymy*:1 (in the direction from a hypernym to it hyponym),
- *hyponymy*: 0.7,
- *antonymy*: 0.4,
- *meronymy*: 0.6 (from a meronym to its holonym),
- *holonymy*: 0.6,
- *synonymy* and *synonymy bis*:1,
- *inter-register synonymy*:1,
- *converse*:1,
- *feminity*, *young being*, *augmentativity*:0,7 (relations between nouns based on derivations).

The lower value for *hyponymy* is meant to prevent too far going generalisation of the activation initially delivered by the knowledge sources. All derivationally motivated relations are similar to *hyponymy* and are assigned the same value of v.

In the case of a wordnet represented by a graph of LUs, resulting from the conversion described in Sec. III. 3., a synset is represented by a chain of LUs linked by *synonymy* and *synonymy bis*. For a synset, all synset relations are mapped to the LU which is its 'head', but other lexical relations are connected to different LUs of the chain. As a result, the activation must be passed through the chain in

order to be further spread via the synset relations, and is several times decreased by the decay factor μ. In order to avoid such a non-intuitive loss, the transmittance coefficient for both *synonymy* and *synonymy bis* can be set to $1/\mu$. Here, transmittance is used to emphasise the activation. It shows the versatility of the model. The values of the transmittance coefficient can be optimised in a way similar to the optimisation of the other parameters, cf [43].

Impedance describes how much indirect activation can be transferred through a given junction of two links, and it is mostly defined on the level of relations, i.e. link types:
f_I(*relation in*, *relation out*, *activation*) returns *modified activation*.

Activation spreading can also be perceived as a kind of reasoning. The impedance expresses intuition that some patterns of such 'reasoning' do not make sense, e.g. from a LU to hypernym via *hyponymy* and next to the antonym of that hypernym via *antonymy*, so the path *hyponymy*–*antonymy* should be excluded from the activation spreading and it can be done by setting impedance to returning 0 for all link junctions of the type: (*hyponymy*,*antonymy*). Taking another example, spreading activation through *holonymy*–*meronymy* would mean that the direct activation assigned to the whole (a holonym) via a part (a meronym) could be attributed to another whole – the LU at the end that can be a holonym to something etc., e.g. 'car' *holonym*–'windscreen' *meronym:substance* – 'glass' – with the indirect activation replicated from 'car' to 'glass' which is intuitively going in the wrong direction.

Thus, in the experiments presented in the next section, we use the impedance function f_I to block activation spreading via link junction of certain types by returning the zero as the output value for relation pairs:
(*hyponymy*, *antonymy*), (*hyponymy*, *meronymy*), (*hypernymy*, *hyponymy*), (*hypernymy*, *holonymy*), (*antonymy*, *antonymy*), (*antonymy*, *meronymy*), (*antonymy*, *holonymy*), (*meronymy*, *antonymy*), (*holonymy*, *antonymy*).

In addition, impedance can be used to block some immediate loops of relations by returning zero for:
(*hypernymy*,*hyponymy*), (*hyponymy*,*hypernymy*), (*synonymy*, *synonymy bis*), (*synonymy bis*,*synonymy*), (*meronymy*,*holonymy*) and (*holonymy*,*meronymy*).

However, it should also be noted that by blocking *hyper/hyponymy* pairs we also block a possibility of changing the direction of activation spreading in the wordnet hypernymy tree resembling structure.

In all other cases the impedance function f_I returns 1, but there could also be some coefficient used.

Concluding, the exact specifications of both *transmittance* and *impedance* functions are in fact parameters of *Paintball*. By changing them we determine the way in which the wordnet graph is interpreted during the process of spreading activation.

IV. EVALUATION OF PAINTBALL

IV. 1. Methodology

The evaluation is based on the wordnet reconstruction task proposed in [44]:

- randomly selected lemmas are removed from a wordnet (i.e. all their LUs are removed), and next
- the expansion algorithm is applied to reattach them.

Removing even a single lemma from a wordnet changes its structure, and with the increasing number of lemmas removed, the changes are becoming more significant. So it would be best to remove one lemma at a time, but due to the efficiency issue small lemma samples are processed in one go.

In addition, we can face two types of problems when preparing the wordnet graph for evaluation:

1. artificial synsets that do not represent LUs, but are labelled by natural language expressions and are introduced into the hypernymy structure only to improve its readability for humans, cf [8],
2. empty synsets resulting from removing LUs for testing from singleton synsets.

A wordnet graph containing such synsets must be transformed before being converted to LU-based graph, as such elements are not natural for the semantic structure of the wordnet. During the transformation both artificial synsets and empty synsets should be removed and the links attached to them must be reconnected to other synsets. This is done in a way taking into account the link types (i.e. the semantic knowledge expressed by them).

- *hypernymy* and *hyponymy*: links are attached to the hypernym of the removed synsets while preserving the original directions of the links, if there is more than one hypernym, the original links are multiplied.
- *type* and *instance*: the same procedure as above.
- *inter-register synonymy*: as above.
- *holonymy* and *meronymy*: if there is a hypernym, the procedure is as above, but if not, the links are also removed,
- LU relations are attached to the head of the hypernym synset.

As the algorithm may produce multiple attachment suggestions for a lemma, they are sorted according to the activation of the suggested attachment areas. A histogram of distances between the suggested attachment places and the original synsets of the lemmas processed during an evaluation is built. We used two approaches to compute the distance between the proposed attachment synsets[17] and original synsets: *straight* and *folded*. According to the first, called *straight*, a proper path can include only hypernymy or hyponymy links (one direction only per path), and one optional final meronymic link. Only up to 6 links are considered, as longer paths are not useful suggestions for linguists.

[17]As an attachment synset a synset with the maximal activation in a given attachment area is taken.

In the second approach, called *folded*, shorter paths are considered, up to 4 links. Paths can include both hypernymy and hyponymy links, but only one change of direction is allowed and an optional meronymic link, if only present, there must be the final link in a path. In this approach we consider close cousins (co-hyponyms) as valuable suggestions for linguists.

The collected results are analysed according to three strategies:

1. *closest path* strategy we analyse only one attachment suggestion per lemma that is the closest one to any of its original locations,
2. *strongest* – only one suggestion with the highest support for a lemma is considered,
3. *all* – all suggestions are evaluated.

A set of test lemmas was selected randomly from wordnet lemmas according to the following conditions. Only words of the minimal frequency 200 were used due to the applied methods for relation extraction. Moreover, only words located further than 3 hyponymy links from the top were considered, as we assumed that the upper parts are constructed manually in most wordnets.

IV. 2. Knowledge sources and input data

For the sake of comparison with [34] and [39] two similar KSs were built: a *hypernym classifier* and a *cousin classifier*. A hypernym only classifier [45] was trained on English Wikipedia corpus (1.4 billion words) parsed by *Minipar* [46]. We extracted all patterns linking two nouns in dependency graphs and occurring at least five times and used them as features for the logistic regression classifier from *LIBLINEAR* [47]. The classifier was applied to lemma pairs to create a knowledge source: pairs classified as hypernymic were described by probabilities of positive decisions. During the wordnet development practice, a richer and more heterogeneous set of KSs is used with Paintball in WordnetWeaver, see the next two sections.

Following [39], the cousin classifier was based on distributional similarity instead of text clustering as it was originally in [34], because the clustering method used was not well specified in that paper. The cousin classifier is meant to predict (m, n)-cousin relationship between lemmas. The classifier was trained to recognise the following classes: $0 \leq m, n \leq 3$ and the negative class which indicates more distant or not linked lemmas at all. So, a Measure of Semantic Relatedness (MSR) was used to produce input features to the logistic regression classifier. The applied MSR was calculated as cosine similarity between two distributional vectors: one vector per a lemma, each vector element corresponds to the frequency of co-occurrences with other lemmas in the selected dependency relations. Co-occurrence frequencies were weighted by PMI.

A sample of 1064 test words was randomly selected from WordNet 3.0. It is large enough for the error margin 3% and 95% confidence level [48]. Trained classifiers were applied to every pair: a test word and a noun from WordNet. This yielded two lists containing evidence for both tested expansion algorithms.

IV. 3. Parameters of algorithms

As a *baseline* we used the well-known and often cited algorithm PWE [34]. Its performance strongly depends on values of predefined parameters. We tested several combinations of values and selected the following ones:

- minimal probability of evidence: 0.1,
- inverse odds of the prior: $k = 4$,
- maximum of the cousins neighbourhood size: $(m, n) \leq (3, 3)$,
- maximum links in hypernym graph: 10,
- penalization factor: $\lambda = 0.95$ of the hypernym probability.

In *Paintball* probability values produced by the classifiers were used as weights. The hypernym classifier produces values from the range $\langle 0, 1]$. Values from the cousin classifier were mapped to the same range by multiplying them by 4. Values of the parameters were set heuristically in relation to the weight values as follows: $\tau_0 = 0.4$, $\tau_3 = \tau_0$, $\tau_4 = 0.8$, $\epsilon = 0.14$ and $\mu = 0.65$.

Transmittance was used to define links for activation replication in *Paintball*. The graph was formed by hyper/hyponymy, holo/meronymy, antonymy and synonymy (represented by synsets). Transmittance is $f_T(r, v) = \alpha * v$, where alpha was: 0.7 for hypernymy, 0.6 for mero/holonymy and 0.4 for antonymy. Parameter α was 1 for other selected relations and 0 for non-selected. Transmittance can be tuned on the basis of the correlation of the activation values, e.g. a MSR observed on both ends of relation links cf [39].

Impedance allows for controlling the shape of the spreading activation graph. Here, the impedance function is defined as: $f_I(r_1, r_2, v) = \beta * v$, where $\beta \in \{0, 1\}$. We selected heuristically $\beta = 0$ for the following pairs (relation in, relation out):
⟨hyponymy, antonymy⟩: blocks antonym down to the hyponym, ⟨hyponymy, meronymy⟩, ⟨hypernymy, hyponymy⟩, ⟨hypernymy, holonymy⟩, ⟨antonymy, antonymy⟩, ⟨antonymy, meronymy⟩, ⟨antonymy, holonymy⟩, ⟨meronymy, antonymy⟩ and ⟨holonymy, antonymy⟩.

IV. 4. Results

Paintball and PWE algorithms were tested on the same lemma sample, and the results are presented in Tab. 1 and 2. Test lemmas were divided into two sub-samples: frequent words, >1000 occurrences (Freq in tables) and infrequent, ≤999 (Rare in tables), as we expected different precision and coverage of KSs for both subclasses. Statistically significant results were marked with a '*'. We rejected the null hypothesis of no difference between results at the significance level $\alpha = 0.05$. The paired t-test was used.

In the case of the straight paths and their maximal length up to 6 links PWE performs slightly better than *Paintball*.

Tab. 1. Straight path strategy: PWE and Paintball precision on WordNet 3.0

		Strategy	Hits distance [%]								
			0	1	2	3	4	5	6	[0, 2]	total
PWE	Rare	Closest	3.7	21.7	16.2	9.6	6.9	3.4	0.1	41.6	***61.5**
		Strongest	0.5	5.9	9.7	10.9	8.9	4.5	0.5	*16.1	**40.9**
		All	0.8	4.9	5.0	4.5	3.8	2.0	0.4	*10.7	**21.5**
	Freq	Closest	0.8	14.8	24.2	21.0	15.1	5.5	0.2	39.8	*81.6
		Strongest	0.1	2.7	9.4	16.1	15.7	13.2	0.8	*12.2	*58.0
		All	0.2	3.2	7.0	10.0	9.8	7.3	0.5	10.4	*38.0
Paintball	Rare	Closest	9.2	21.7	12.6	6.7	4.2	1.0	0.6	**43.5**	*56.1
		Strongest	4.8	13.1	10.0	6.5	3.4	1.2	0.4	***27.9**	39.4
		All	2.9	6.9	4.8	3.5	2.2	1.0	0.2	***14.6**	**21.5**
	Freq	Closest	6.3	20.5	15.0	11.9	6.7	2.6	0.5	**41.8**	*63.3
		Strongest	1.9	9.1	8.4	8.1	4.8	1.9	0.3	***19.4**	*34.7
		All	1.4	4.9	4.4	4.4	3.1	1.6	0.2	**10.7**	*20.0

Coverage for words and senses is also higher for PWE: 100% (frequent: 100%) 44.79% (43.93%) than for Paintball: 63.15% (freq.: 91.63%) and 24.66% (26.62%). However, a closer analysis reveals that PWE shows a tendency to find suggestions in larger distances from the proper place. If we take into account only suggestions located up to 3 links – the column [0,2] in Tab. 1, than the order is different: *Paintball* is significantly better than PWE. *Paintball* mostly suggests more specific synsets for new lemmas and abstains in the case of the lack of evidence. For instance, for $x =$ *feminism*, PWE suggests the following synset list:
{*abstraction*, *abstract entity*}, {*entity*}, {*communication*}, {*group*, *grouping*}, {*state*}.

In the same time, suggestions generated by *Paintball* are still not perfect, but they show to be more specific:
{*causal agent*, *cause*, *causal agency*}, {*change*}, {*political orientation*, *ideology*, *political theory*}, {*discipline*, *subject*, *subject area*, *subject field*, *field*, *field of study*, *study*, *bailiwick*}, {*topic*, *subject*, *issue*, *matter*}.

PWE very often suggests such abstract and high level synsets like: {entity}, {event}, {object}, {causal agent, cause, causal agency} etc. They dominate whole branches and are in a distance non-greater than 6 links to many synsets. Such general suggestions are not valuable support for lexicographers, in fact.

PWE achieved only slightly better results as measured for the straight paths than the baseline for the *strongest* and *all* evaluation strategies. This was caused by the fact that the baseline algorithm does not perform any sense disambiguation.

Paintball outperforms PWE in the evaluation based on the folded paths. For more than half test lemmas, the strongest proposal was in the right place or up to a couple of links from it. Suggestions were generated for 72.65% of lem-

Tab. 2. Folded path evaluation strategy: PWE and Paintball precision on WordNet 3.0

		Strategy	Hits distance [%]					
			0	1	2	3	4	total
PWE	Rare	Closest	3.7	21.7	18.4	11.8	2.5	*58.2
		Strongest	0.5	5.9	10.7	12.6	2.3	*32.0
		All	0.8	4.9	6.6	6.9	1.5	*20.7
	Freq	Closest	0.8	14.8	25.2	22.9	4.0	67.7
		Strongest	0.1	2.7	9.6	17.0	3.4	*32.8
		All	0.2	3.2	7.9	12.2	2.9	*26.4
Paintball	Rare	Closest	9.2	21.7	21.9	10.7	1.9	***65.5**
		Strongest	4.8	13.1	15.3	13.1	1.5	***47.9**
		All	2.9	6.9	14.7	13.2	1.7	***39.4**
	Freq	Closest	6.3	20.5	20.7	18.6	2.8	**68.8**
		Strongest	1.9	9.1	11.5	13.5	3.1	***39.2**
		All	1.4	4.9	8.4	11.6	2.3	***28.5**

mas and the sense recall was 24.63%. Both values are comparable with other algorithms. The folded path evaluation shows how many suggestions will be presented on the WordnetWeaver screen in a distance useful for lexicographers to spot the right place and quickly start editing.

V. APPLICATION OF PAINTBALL TO HETEROGENOUS KNOWLEDGE SOURCES

Paintball was also run on plWordNet 1.6 on the basis of five heterogenous knowledge sources acquired from plWordNet Corpus 6.0, including more than 1.8 billion words, with the help of several methods for the extraction of lexical-semantic relations:

- MSR based on the RWF transformation of the coincidence matrix (MSR_{RWF}), studied intensively for nouns [49]: MSR allows for the identification of lemmas that are strongly semantically associated, but it does not allow for discerning between different lexico-semantic relations;
  Two knowledge sources were produced with the help of MPZ_{RWF}:
  - a set of sets $MPZset(y,k)$ of the k ($k = 20$) most semantically related lemmas to y,
  - and a similar set but limited only to lemmas mutually occurring on such lists of the most semantically related ones
    $MPZ_{BiDir}(y,k) = \{y' : y' \in MPZset(y,k) \wedge y \in MPZset(y',k)\}$;
- a classifier C_H [50] used to filter out from MPZ_{RWF} lemma pairs that are not in a selected lexico-semantic relation. In this case the classifier was trained to recognise instances of the merged relations: *hypernymy* (also indirect up to 3 links), *meronymy* and *synonymy*;
- manually written lexico-morpho-syntactic patterns following the idea of [51] and presented in [8] of the general schemes:
  ⟨NP, NP, ... i inne 'and other' NP⟩, ⟨NP jest 'is' NP⟩ i ⟨NP to '≈is' NP⟩;
- the *Estratto* algorithm [52], in which extraction patterns are acquired automatically in an iterative, remotely supervised process.

As a result, the applied knowledge sources were identical with respect to their types with those applied in [8] for plWordNet 1.0. However, we applied new versions of language tools to extract them and a much bigger corpus [5]. For the evaluation, we randomly selected a sample of lemmas with the frequency ≥ 1000 in the corpus and located in a distance larger than three hyponymy links from any of the hypernymy root synsets in plWordNet 1.6.

The evaluation was based on the procedure described above and the results are presented in Tab. 3. In addition we tested different versions of evaluation with respect to the shape of a wordnet graph path considered as a proper connection between the suggestion and the original location, namely:

- *hyponymy only* path: a suggestion is a direct or indirect hyponym of the original LU (obviously, synonyms are also included in this and all other cases),
- *hypernymy only*: a suggestion is a direct or indirect hyponym of the original LU (i.e. the suggestion is more general),
- *hyper/hyponymy*: both direction: down and up the hypernymy structure are taken into account, but only one of them is considered at a time,
- *bidirectional* path: a path linking a suggestion and the original LU can consist of any sequence of hypernymy and hyponymy links, it can change directions several times,
- *shorten with one direction change* path: a path can consists of a combination of hypernymy and hyponymy links but only one change of the direction is acceptable (a cousin relation), the path must be no longer than three links; the last link can be any constitutive relation, not only hyper/hyponymy.

In Tab. 3, what is most important for linguists is the last configuration of the evaluation method in which only shorter paths are considered, as the cousin relation is a very useful suggestion for a new lemma. A narrow contest is delimited and linguists only need to make small corrections, if any. In this configuration more than 30% of the suggestions appeared to be correct even considering that the algorithm was tuned for achieving higher recall. We can also notice high accuracy of the top scored suggestions. However, most of them are not linked directly to the point but only attached to the appropriate subtree.

VI. PAINTBALL IN WORDNET DEVELOPMENT PRACTICE

For comparison with the PWE algorithm [34] we used a very limited set of KSs. However, in a way similar to the experiment presented in the previous section, for the needs of wordnet development, we tried to collect and utilise all possible KSs, namely:

1. lists of the k most related lemmas generated from MSRs built for different Parts of Speech on the basis of shallow and selective analysis of lexico-syntactic dependencies between word pairs [8],
2. lists of the k most related lemmas (k-MRL) filtered by the bidirectional relatedness, i.e. only words mutually present in both k-MRLs are preserved,
3. handcrafted patterns for the extraction of hyper/hyponymy relation [8],
4. automatically extracted shallow patterns for the recognition of hyper/hyponymy instances on the basis of a limited set of seed examples by *Estratto* algorithm [53],

Method	Distance of the suggestion							
	0	1	2	3	4	5	6	altogether
hyponymy only (R(meanings) = 14.84%)								
CLOSEST [%]	14.0	12.0	2.0	0.2	–	–	–	28.2
STRONGEST [%]	7.1	6.0	1.1	–	–	–	–	14.2
ALL [%]	3.0	2.8	0.8	0.3	0.1	–	–	7.0
hypernymy only (R(meanings) = 23.32%)								
CLOSEST [%]	14.0	19.4	9.1	1.9	0.6	0.1	0.1	45.2
STRONGEST [%]	7.1	10.0	6.5	1.0	0.2	0.1	0.1	25.0
ALL [%]	3.0	4.4	2.3	0.8	0.2	0.1	0.0	10.9
hyper/hyponymy (R(meanings) = 27,98%, R(lematy) =99.81%)								
CLOSEST [%]	14.0	25.3	10.6	1.9	0.6	0.1	0.1	52.6
STRONGEST [%]	7.1	13.3	7.5	0.9	0.2	0.1	0.1	29.2
ALL [%]	3.0	6.0	3.1	1.0	0.3	0.1	0.0	13.6
bidirectional (R(meanings) = 53.23%)								
CLOSEST [%]	14.0	25.3	27.8	9.3	7.2	4.8	0.8	89.3
STRONGEST [%]	7.1	13.3	18.1	9.6	8.6	7.7	1.7	66.0
ALL [%]	3.0	6.0	11.2	9.1	11.0	9.2	1.9	51.6
shorten with one direction change (R(meanings) = 43.61%)								
CLOSEST [%]	14.0	25.5	27.6	9.5	0.8	–	–	77.5
STRONGEST [%]	7.1	13.3	18.1	9.5	1.0	–	–	49.0
ALL [%]	3.0	6.1	11.3	8.9	1.7	–	–	31.0

Tab. 3. Precision of *Paintball* based on activation replication scheme in the wordnet reconstruction task performed on plWordNet 1.6

5. and a classifier recognising words linked by one of the interesting wordnet relations (direct and indirect) on the basis of several distributional features extracted from a corpus which was trained by Machine Learning [50].

Both methods based on patterns appeared to be useful only for nouns. It was difficult to manually construct patterns for other Parts of Speech[18], and automatically extracted patters had too low accuracy. As the classification-based approach also utilises the MSR values as one of the features, an MSR is the main KS. Bi-directional k-MRLs expressed much higher accuracy than k-MRL alone, because they express a kind of rank-based filtering parameterised by k.

VII. CONCLUSIONS

An MSR is a helpful tool in wordnet development as it expresses semantic knowledge condensed from thousands of occurrences. According to the wordnet-based experimental evaluation word embeddings produce slightly but significantly better k-MRLs in terms of the number of wordnet relation instances covered by a single list of the k-most related lemmas to a given one. An MSR is always useful for semantic clustering of words into packages as work assignment units. Usefulness of the description of individual words, e.g., as k-MRLs, depends very much on the frequency of the given word in the corpus. Words with the frequency $\geq$ 1000 per 1 billion word corpus obtain very good description. Words $\geq$ 200 per 1 billion word corpus are mostly well described, in the case of words with the frequency between 200 and 100, there is a good chance of obtaining a useful description, between 30–100 one must be lucky, below 30, we can mostly see only noise. However, the experience of the expansion of plWordNet from version 2.0 to 3.0 showed that there are quite many normal common words that are less frequent than 30 in a quite well selected 1 billion word corpus. So, even a good MSR cannot replace a manual description in a large and comprehensive wordnet. Moreover, a good description is produced by an MSR only for the most salient senses. Usually a couple of senses dominate in an MSR, e.g. only they are visible in k-MRLs.

WordnetWeaver, and Paintball included in it, were very useful for frequent words which are well described. Especially the combination of MSR-based KSs and pattern-based KSs was very informative. The best situation was for nouns, where we could also use the elaborated hypernymy structure of plWordNet in Paintball. In the case of other Parts of Speech, the usefulness of WordnetWeaver was much decreased. For well described words, WordnetWeaver could draw lexicographers' attention to less obvious senses or senses not present in the traditional dictionaries but learned

[18] Especially because, e.g., two verbs associated by a lexico-semantic relation very rarely co-occur in the same sentence.

from corpora. Unfortunately, more frequent words are first to be described in a wordnet, so the importance of WordnetWeaver had been decreasing, and finally it became only a tool for visualising new lemma packages assigned to lexicographers. In the future, we plan to revive WordnetWeaver as a diagnostic tool for semantic evaluation of potential faults in the wordnet structure. With the growing density of the plWordNet relation graph its applicability should be increasing, because more information can flow to different nodes. It would also be very interesting to combine it with word embeddings built for word senses.

Automated extraction of the semantically representative use examples (i.e. a kind of word sense induction) appeared to be a tool which is the most appreciated by lexicographers. We plan to work on its version which goes beyond word co-occurrences and utilises semantic information, e.g. coming from word embeddings.

A combination of a wordnet and word embeddings as complementary lexical knowledge sources is an interesting challenge, but the latter will not help us with respect to the most laborious part of the work, i.e. the description of the less frequent words and senses.

Acknowledgement

The paper is the result of works carried out within the project funded by the National Science Centre (Narodowe Centrum Nauki), Poland under the grant agreement No UMO-2015/18/M/HS2/00100.

References

[1] M. Maziarz, M. Piasecki, E. Rudnicka, S. Szpakowicz, P. Kędzia, *plwordnet 3.0 – a comprehensive lexical-semantic resource,* [in:] *COLING 2016, 26th International Conference on Computational Linguistics, Proceedings of the Conference: Technical Papers*, December 11–16, 2016, Osaka, Japan (N. Calzolari, Y. Matsumoto, R. Prasad, eds.), pp. 2259–2268, ACL, ACL, 2016.

[2] P. Vossen, ed., *EuroWordNet. A multilingual Database with Lexical Semantic Networks*. Kluwer Academic Publishers, 1998.

[3] P. Vossen, *EuroWordNet General Document Version 3,* tech. rep., Univ. of Amsterdam, 2002.

[4] M. Derwojedowa, M. Piasecki, S. Szpakowicz, M. Zawisławska, B. Broda, *Words, Concepts and Relations in the Construction of Polish WordNet,* in *Proc. Fourth Global WordNet Conf.* (A. Tanács, D. Csendes, V. Vincze, C. Fellbaum, P. Vossen, eds.), pp. 162–177, 2008.

[5] M. Maziarz, M. Piasecki, E. Rudnicka, S. Szpakowicz, *Beyond the transfer-and-merge wordnet construction: plWordNet and a comparison with WordNet,* [in:] *Proc. International Conference Recent Advances in Natural Language Processing RANLP 2013*, pp. 443–452, INCOMA Ltd. Shoumen, BULGARIA, 2013.

[6] D. Widdows, *Geometry and Meaning*. CSLI Publications, 2004.

[7] T. Mikolov, I. Sutskever, K. Chen, G. Corrado, J. Dean, *Distributed representations of words and phrases and their compositionality, CoRR*, vol. abs/1310.4546, 2013.

[8] M. Piasecki, S. Szpakowicz, B. Broda, *A Wordnet from the Ground Up*. Wrocław: Oficyna Wydawnicza Politechniki Wrocławskiej, 2009.

[9] A. Przepiórkowski, *The IPI PAN Corpus: Preliminary version*. Institute of Computer Science, Polish Academy of Sciences, 2004.

[10] A. Przepiórkowski, M. Bańko, R.L. Górski, B. Lewandowska-Tomaszczyk, eds., *Narodowy Korpus Języka Polskiego [in Polish]*. Wydawnictwo Naukowe PWN, 2012. http://nkjp.pl/settings/papers/NKJP_ksiazka.pdf.

[11] D. Weiss, *Korpus Rzeczpospolitej [Corpus of text from the online edition of "Rzeczpospolita"]*, http://www.cs.put.poznan.pl/dweiss/rzeczpospolita, 2008.

[12] M. Woliński, *Morfeusz reloaded,* [in:] *Proceedings of the Ninth International Conference on Language Resources and Evaluation, LREC 2014*, pp. 1106–1111, ELRA, 2014.

[13] B. Svensén, *A Handbook of Lexicography. The Theory and Practice of Dictionary-Making*. Cambridge University Press, 2009.

[14] C. Fellbaum, *A Semantic Network of English: The Mother of All WordNets, Computers and the Humanities*, vol. 32, pp. 209–220, 1998.

[15] B. Broda, M. Maziarz, M. Piasecki, *Tools for plWordNet Development. Presentation and Perspectives,* [in:] *Proceedings of the Eight International Conference on Language Resources and Evaluation (LREC'12)*, pp. 3647–3652, European Language Resources Association (ELRA), May 2012.

[16] M. Piasecki, M. Marcińczuk, R. Ramocki, M. Maziarz, *WordNetLoom: a WordNet development system integrating form-based and graph-based perspectives, International Journal of Data Mining, Modelling and Management*, vol. 5, no. 3, pp. 210–232, 2013.

[17] T. Naskręt, A. Dziob, M. Piasecki, C. Saedi, A. Branco, *WordnetLoom – a multilingual wordnet editing system focused on graph-based presentation,* [in:] *Proceedings of the 9th Global Wordnet Conference, Singapore, 8–12 January 2018* (F. Bond, C. Fellbaum, P. Vossen, eds.), Global Wordnet Association, 2018.

[18] M. Wynne, ed., *Developing Linguistic Corpora: a Guide to Good Practice*. Oxford: Oxbow Books, 2005.

[19] plWordNet, *Frequency List from plWorNet Corpus,* 2012. www.nlp.pwr.wroc.pl/pl/narzedzia-i-zasoby/lista-frekwencyjna.

[20] B. Broda and M. Piasecki, *Parallel, massive processing in SuperMatrix – a general tool for distributional semantic analysis of corpora, International Journal of Data Mining, Modelling and Management*, vol. 5, no. 1, pp. 1–19, 2013.

[21] T. Mikolov, K. Chen, G. Corrado, J. Dean, *Efficient estimation of word representations in vector space, CoRR*, vol. abs/1301.3781, 2013.

[22] P. Bojanowski, E. Grave, A. Joulin, T. Mikolov, *Enriching word vectors with subword information, arXiv preprint arXiv:1607.04606*, 2016.

[23] M. Piasecki, G. Czachor, A. Janz, D. Kaszewski, P. Kędzia, *Wordnet-based evaluation of large distributional models for polish,* in *Proceedings of the 9th Global Wordnet Conference, Singapore, 8-12 January 2018* (F. Bond, C. Fellbaum, P. Vossen, eds.), Global WordNet Association, 2018.

[24] M. Piasecki, A. Janz, D. Kaszewski, G. Czachor, *Word embeddings for polish,* 2017. CLARIN-PL digital repository.

[25] J. Kocoń, *KGR10 FastText polish word embeddings,* 2018. CLARIN-PL digital repository.

[26] J. Kocoń and M. Marcińczuk, *Word embeddings for polish (KGR10, fasttext binary) kgr10_fasttext_bin_v1,* 2018. CLARIN-PL digital repository.

[27] G. Karypis, *CLUTO a clustering toolkit,* Technical Report 02-017, Department of Computer Science, University of Minnesota, 2002.

[28] B. Broda, M. Maziarz, M. Piasecki, *Evaluating LexCSD – a Weakly-Supervised Method on Improved Semantically Annotated Corpus in a Large Scale Experiment,* [in:] *Proceedings of a Conference on Intelligent Information Systems* (M.A. Kłopotek, A. Przepiórkowski and K. Trojanowski, eds.), 2010.

[29] D. Janus and A. Przepiórkowski, *Poliqarp 1.0: Some technical aspects of a linguistic search engine for large corpora,* [in:] *The proceedings of Practical Applications of Linguistic Corpora*, 2005.

[30] T. Machalek, *KonText – a modern, customizable corpus query interface,* in *Book of Abstracts of the Corpus Linguistics 2017 Conference, 25-28 July 2017*, (Birmingham), University of Birmingham, 2017.

[31] M. Piasecki and M. Wendelberger, *Partial measure of semantic relatedness based on the local feature selection,* [in:] *Text, Speech and Dialogue – 17th International Conference, TSD 2014, Brno, Czech Republic, September 8-12, 2014. Proceedings* (P. Sojka, A. Horák, I. Kopecek, K. Pala, eds.), vol. 8655 of *Lecture Notes in Computer Science*, pp. 336–343, Springer, 2014.

[32] M. Piasecki, R. Ramocki, M. Kaliński, *Information spreading in expanding wordnet hypernymy structure,* [in:] *Proc. International Conference Recent Advances in Natural Language Processing RANLP 2013*, pp. 553–561, INCOMA Ltd. Shoumen, BULGARIA, 2013.

[33] M. Piasecki, Ł. Burdka, M. Maziarz, M. Kaliński, *Diagnostic tools in plwordnet development process,* [in:] *Human Language Technology. Challenges for Computer Science and Linguistics* (Z. Vetulani, H. Uszkoreit, and M. Kubis, eds.), vol. 9561 of *LNCS*, pp. 255–273, Springer, 2016.

[34] R. Snow, D. Jurafsky, A.Y. Ng., *Semantic taxonomy induction from heterogenous evidence.,* pp. 801–808, The Association for Computer Linguistics, 2006.

[35] A.M. Collins and E.F. Loftus, *A spreading-activation theory of semantic processing, Psychological Review*, vol. 82, no. 6, pp. 407–428, 1975.

[36] G. Salton and C. Buckley, *On the use of spreading activation methods in automatic Information Retrieval,* [in:] *Proceedings of ACM SIGIR*, 1988.

[37] N.M. Akim, A. Dix, A. Katifori, G. Lepouras, N. Shabir, C. Vassilakis, *Spreading activation for web scale reasoning: Promise and problems,* in *Proceedings of WebSci* '11, June 14-17, 2011, Koblenz, Germany, 2011.

[38] A. Troussov, M. Sogrin, J. Judge, D. Botvich, *Mining socio-semantic networks using spreading activation technique,* [in:] *Proceedings of I-KNOW '08 and I-MEDIA '08 Graz, Austria, September 3–5, 2008*, pp. 405–412, 2008.

[39] M. Piasecki, R. Kurc, R. Ramocki, B. Broda, *Lexical Activation Area Attachment Algorithm for Wordnet Expansion,* [in:] *Proc. 15th International Conference on Artificial Intelligence: Methodology, Systems, Applications* (A. Ramsay and G. Agre, eds.), vol. 7557 of *Lecture Notes in Computer Science*, pp. 23–31, Springer, 2012.

[40] M. Derwojedowa, S. Szpakowicz, M. Zawisławska, M. Piasecki, *Lexical Units as the Centrepiece of a Wordnet,* [in:] *Proc. 16th Int. Conf. on Intelligent Information Systems* (M.A. Kłopotek, A. Przepiórkowski, S.T. Wierzchoń, K. Trojanowski, eds.), pp. 351–358, 2008.

[41] M. Maziarz, M. Piasecki, S. Szpakowicz, *The chicken-and-egg problem in wordnet design: synonymy, synsets and constitutive relations, Language Resources and Evaluation*, vol. 47, no. 3, pp. 769–796, 2013.

[42] C. Fellbaum, ed., *WordNet – An Electronic Lexical Database*. The MIT Press, 1998.

[43] Ł. Kłyk, P. Myszkowski, B. Broda, M. Piasecki, D. Urbansky, *Metaheuristics for tuning model parameters in two natural language processing applications,* [in:] *Proceedings of the 15th International Conference on Artificial Intelligence: Methodology, Systems, Applications* (A. Ramsay and G. Agre, eds.), vol. 7557 of *Lecture Notes in Computer Science*, (Varna, Bulgaria), pp. 32–37, Springer, 2012.

[44] B. Broda, R. Kurc, M. Piasecki, R. Ramocki, *Evaluation method for automated wordnet expansion,* [in:] *Security and Intelligent Information Systems* (P. Bouvry, M. Kłopotek, F. Leprevost, M. Marciniak, A. Mykowiecka, H. Rybiński, eds.), LNCS, Springer, 2011.

[45] R. Snow, D. Jurafsky, A.Y. Ng, *Learning syntactic patterns for automatic hypernym discovery,* [in:] *NIPS*, 2004.

[46] D. Lin, *Principle-based parsing without overgeneration,* [in:] *Proc. ACL-93, Columbus, Ohio*, 1993.

[47] R.-E. Fan, K.-W. Chang, C.-J. Hsieh, X.-R. Wang, C.-J. Lin, *LIBLINEAR: A library for large linear classification, Journal of Machine Learning Research*, vol. 9, pp. 1871–1874, 2008.

[48] G. Israel, *Determining sample size,* tech. rep., University of Florida, 1992.

[49] M. Piasecki, S. Szpakowicz, B. Broda, *Automatic selection of heterogeneous syntactic features in semantic similarity of Polish nouns,* [in:] *Text, Speech and Dialogue, 10th International Conference, TSD 2007, Pilsen, Czech Republic, September 3-7, 2007, Proceedings* (V. Matousek and P. Mautner, eds.), vol. 4629 of *LNCS*, pp. 99–106, Springer, 2007.

[50] M. Piasecki, M.M. annd Stanisław Szpakowicz, B. Broda, *Classification-based filtering of semantic relatedness in hypernymy extraction,* [in:] *Advances in Natural Language Processing, 6th International Conference, GoTAL 2008, Gothenburg, Sweden, August 25-27, 2008, Proceedings* (B. Nordström and A. Ranta, eds.), vol. 5221 of *LNCS*, pp. 393–404, Springer, 2008.

[51] M.A. Hearst, *Automated Discovery of WordNet Relations*, ch. 5, pp. 131–151. Vol. 1 of Fellbaum [42], 1998.

[52] R. Kurc and M. Piasecki, *Automatic acquisition of wordnet relations by the morpho-syntactic patterns extracted from the corpora in polish,* [in:] *Proceedings of the International Multiconference on Computer Science and Information Technology – 3nd International Symposium Advances in Artificial Intelligence and Applications (AAIA'08)*, pp. 181–188, 2008.

[53] R. Kurc, M. Piasecki, S. Szpakowicz, *Automatic acquisition of wordnet relations by distributionally supported morphological patterns extracted from polish corpora,* [in:] *Text, Speech and Dialogue, 13th International Conference, TSD 2010, Brno, Czech Republic, September 6-10, 2010. Proceedings* (P. Sojka, A. Horák, I. Kopecek, K. Pala, eds.), vol. 6231 of *Lecture Notes in Computer Science*, pp. 133–141, 2010.

Maciej Piasecki Personal Data: born 28 January 1970, Wrocław, Poland; Assistant Professor at the Faculty of Computer Science and Management, Wrocław University of Science and Technology, the Polish National Coordinator of CLARIN (clarin.eu) (European language technology research infrastructure), since 04.2018 the Chair of CLARIN ERIC National Coordinators Forum, the coordinator of CLARIN-PL (clarin-pl.eu) (national project, around 6.9M Euro in years 2013-2018), is a leader of G4.19 Research Group: Computational Linguistics and Language Technology (nlp.pwr.edu.pl) in the Department of Computational Intelligence, (vice-head of the department). Dr Piasecki has been or is a coordinator of 14 large projects or their work packages (national and funded from EU structural funds, including 3 projects in cooperation with companies) on language technology and its different applications (more than 12M Euro of the total budget in the years 2008-till now). He is also a member of the DARIAH-PL Board and Global WordNet Association Board. The main mission of G4.19 is development of open robust language technology for Polish, both in monolingual and bilingual setting. His main research areas are: Computational Linguistics, Natural Language Engineering and Human Language Technology and the main research topics are: automated extraction of the lexico-semantic knowledge from text, semi-automated wordnet expansion, Distributional Semantics, relational lexical semantics and shallow semantic processing of text. He has been also working on morpho-syntactic processing of Polish (a co-author of the first publicly available morpho-syntactic tagger of Polish, with many applications), Information Extraction, Question Answering, formal semantics and Machine Translation. He has been a leader of the Polish wordnet project: plWordNet (`plwordnet.pwr.edu.pl`) – the largest language resource of this type in the world. Dr Piasecki is a PC member of several world major scientific conferences in Computational Linguistics and served as a reviewer in several calls of the Polish Ministry of Science and Higher Education, as well as European FP7 and H2020. Dr Piasecki has published 195 peer-reviewed research papers (one book, journal articles and conference papers). In Scopus his h-index = 9. 94 indexed works, 360 citations; in Google Scholar, h-index = 18, 662 citations after manual filtering out self-citations.