Information Spreading in Expanding Wordnet Hypernymy Structure

Maciej Piasecki
Institute of Informatics
Wrocław Univ. of Technology
maciej.piasecki@pwr.wroc.pl

Radosław Ramocki
Institute of Informatics
Wrocław Univ. of Technology
rramocki@gmail.com

Michał Kaliński
Institute of Informatics
Wrocław Univ. of Technology
168023@student.pwr.wroc.pl

Abstract

The paper presents a wordnet expansion algorithm, which is based on lexico-semantic relations extracted from large text corpora. We do not assume that the extracted relation instances (i.e. word pairs) are described by probabilities. Thus, results produced by any method, including pattern-based and Distributional Semantics approaches can be used. The algorithm is based on a general spreading activation model. Support for word-to-word semantic associations is first mapped on the existing wordnet structure. Next, the support is spread over the wordnet network in order to find attachment areas for a new word. Evaluation and comparison with other approaches in experiments on Princeton WordNet 3.0 is presented.

1 Introduction

Wordnets became important large scale language resources providing relational description of lexical meanings. e.g. WordNet (Fellbaum, 1998), GermaNet (Hamp and Feldweg, 1997) or plWordNet (Maziarz et al., 2012). The required large amount of work on wordnet construction can be lessened by supporting manual work with automated tools for the extraction of lexico-semantic relations and wordnet expansion. A scheme for lexico-semantic network extraction from corpus includes, e.g. (Yang and Callan, 2009; Navigli et al., 2011): *term extraction*, extraction of *term associations* and *taxonomy induction*. A taxonomy structure is mostly a subset of the whole wordnet hyper/hyponynymy structure. Thus, a more general task, for the last phase, is *extraction of lexico-semantic relations* (*sensu stricto*), called *relation formation* in (Yang and Callan, 2009). In our work, we focus on the automated expansion of a such wordnet hypernymy structure.

Upper levels of a wordnet hypernymy describe more general, often highly abstract lexical units (i.e. pairs: lemma and its sense). Such fine grained distinctions are hard to trace in a corpus, but mostly it is this part of a wordnet that is created first. Thus, we assumed that upper hypernymic levels are already built manually, and what is needed is to expand the wordnet structure towards the lower levels. Our goal is to develop a method of automated expansion of wordnet hypernymy structure based on both lexico-semantic associations extracted automatically from a large text corpus and the prior partial wordnet structure. We do not assume the existence of any kind of semantic annotation or document structure, to make the proposed method general.

Most taxonomy induction methods use only the existing hypernymy structure as a basis for the incremental wordnet expansion, e.g. (Snow et al., 2006; Piasecki et al., 2009b). We explore all different types of wordnet links to identify the appropriate location for a new lemma sense.

2 Related works

(Alfonseca and Manandhar, 2002) and (Witschel, 2005) treat wordnet hypernymy as a kind of decision tree applied to word meanings described by Distributional Semantics. (Widdows, 2003) attaches words on the basis of their *semantic neighbours* – k most similar words according to their co-occurrence with the most frequent words.

(Snow et al., 2006) proposed Probabilistic Wordnet Expansion (PWE) method, which is based on a probabilistic model of the taxonomy

Proceedings of Recent Advances in Natural Language Processing, pages 553–561,
Hissar, Bulgaria, 7-13 September 2013.

expressed in terms of taxonomic relations. For WordNet expansion Snow *et al.* consider two type of relations: (transitive) *hypernymy* and (m,n)-*cousin*. To prevent adding a new word to overly-specific hypernym λ coefficient was introduced penalized by: λ^{k-1} factor, where k is number of links between attachment synset and its hypernym. (m,n)-cousinhood occurs between two word senses i and j if their *least common subsumer* is exactly m links from i and n links from j in the WordNet graph. Hypernymy and (m,n)-cousinhood instances imply sets of other instances, e.g. a direct hypernym of one word sense implies all other indirect hypernyms. To add a new word to the taxonomy, the whole taxonomy must be (locally) searched for an attachment place that maximises probabilities of all the implied relations. The attachment of new elements transforms the structure **T** into a new **T'**. The appropriate **T'** maximises the probability of the change in relation to the evidence at hand. *Multiplicative change* computation is based on all added relation links, including the links *implied* by hyponymy. Multiplicative change depends on the inverse odds of the prior k which is a constant independent of words and taxonomy **T**. (Snow et al., 2006) have not provided any value of k.

(Kozareva and Hovy, 2010) presented two step taxonomy induction. First, hyponym-hypernym pairs are extracted from Internet and ranked. The extraction mechanism is exclusively based on "doubly-anchored lexico-syntactic patterns" and a heuristic iterative algorithm. The process is weakly controlled by a *root* and *seed* lemmas. (Navigli et al., 2011) divided taxonomy induction into four steps. Three initial ones are devoted to the extraction of hypernymy instances. The process is focused on ontology learning, identification of overt definitions in text and the extraction of hypernymy instances from them. However, definitions are infrequent and occur only in specific text genres. The initial graph emerging from the extracted pairs is next weighted and pruned.

(Yang and Callan, 2009) proposed a *metric-based taxonomy induction framework* aimed at utilising different extraction methods: 15 methods in total were used. Each method produces a *feature function* a term pair $\rightarrow$ a real value or $\{0, 1\}$ value. The process starts with an initial partial taxonomy T^0, used also to estimate values of parameters, so it is a taxonomy expansion. The expansion is controlled by *Minimum Evolution Assumption* and *Abstractness Assumption* principles. The first results in minimising "the overall semantic distance among the terms" but also avoiding "dramatic changes" between the initial taxonomy and the expanded one. The total distance and change are characterised by the Information Function of a taxonomy T. Weights for different *feature functions* can be estimated in supervised training for each taxonomy level separately by approximating ontology metrics for term pairs:
$d(c_x, c_y) = \sum_{j \in features} w_j h_j(c_x, c_y)$
where $h_j()$ is a feature function and w_j its weight. The approximation was done by ridge regression, but it is not clear whether it was done separately for different taxonomy levels. Finally, Multi-Criterion Optimization Algorithm (MCOA) finds a place for each new term by joint application of both conditions, i.e. by minimising: the change in the taxonomy Information Function and the sum over the square error of the difference between new ontology metrics and their estimation based on the weighted feature functions. (Yang and Callan, 2009) performed evaluation on WordNet and an ontology. As far the first, 50 " hypernym taxonomies" were extracted from 12 topics (mostly concrete nouns) and 50 "meronymic taxonomies" from 15 topics (mostly concrete). The size of the test taxonomies and the way of their delimitation was not defined. Feature functions were built on the basis of a corpus including English Wikipedia and 1000 top documents per each term from Google. Precision and recall were calculated on the level of relation links. The number of the correctly attached terms is not known. MCOA achieved slightly better results in reconstructing of the hypernymic taxonomies than PWE.

(Piasecki et al., 2009a; Piasecki et al., 2011) proposed a heuristic wordnet expansion algorithm called *Area Attachment Algorithm* (AAA) which utilises different relation extraction methods. A modified version of AAA, was presented in (Piasecki et al., 2012). Our present work inherits several assumptions from AAA, but it based on a different model of spreading activation.

3 *Paintball* Algorithm

3.1 The idea of information spreading

A corpus is a very imprecise source of lexical semantics knowledge. Knowledge describing lexico-semantic relations that is extracted from

it is always partial (not all word senses occur, most senses are infrequent) and may suggest erroneously accidental semantic associations between words. We cannot avoid errors, but we can to try to compensate them by combining word associations suggested by several extraction methods. Relations extracted automatically can be represented as sets of triples: $\langle x, y, w \rangle$, where y is a word already included in the wordnet, x is a 'new' word not included yet, and $w \in \Re$ is a weight. We call such a set a *knowledge source* (henceforth KS) extracted by a method from a corpus. A triple $\langle x, y, w \rangle$ from a KS K informs that x is semantically associated with y and w describes the strength of this association. In many approaches, e.g. (Snow et al., 2006), 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 few occurrences. Nevertheless, as we need to 'squeeze' all available lexical knowledge out from the text, and we cannot loose any KS. We have to try to utilise those non-probabilistic KSs, too. Most if not all reliable extraction methods produce KSs for words, not word senses. Thus, we assume that w is a value of *support* for the given word pair x & y as semantically associated.

A triple $\langle x, y, w \rangle$ from a KS K_i suggests linking x to synsets including y. However, there are two problems: x and y can have several senses each, and the triple can express some error. In fact, the triple suggests linking x to different senses of y represented by synsets – each y synset describes a possible meaning of x. The triple does not disambiguate this, e.g. PWE hypernymy classifier generates $\langle$*feminism*, *movement*, 1.0$\rangle$, $\langle$*feminism*, *theory*, 0.948$\rangle$, $\langle$*feminism*, *politics*, 0.867$\rangle$, etc. As far the second, apart from 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. 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.

We proposed a wordnet expansion algorithm called *Paintball* which is based on a general model of *spreading activation* (Collins and Loftus, 1975; Salton and Buckley, 1988; Akim et al., 2011): the support from KS triples is the activation which is spread along the wordnet relations. *Paintball* algorithm is based on the metaphor of semantic support for x resembling drops of liquid paint that initially fall on some wordnet graph nodes (synsets) due to KSs and next the paint starts spreading over the graphs. Those regions that represent the highest amounts of paint after the spreading represent possible senses of x and include places for x.

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. KSs based on Distributional Semantics do not specify this relation, and pattern-based KS are mostly focused on hypernymy. So, a real attachment places for an x sense can be somewhere around the y synsets assuming that the given KS does not include too serious errors or too fuzzy semantic associations, e.g. triples generated by PWE hypernymy classifier: $\langle$*feminism*, *relationship*, 0.768$\rangle$, $\langle$*feminism*, *study*, 0.951$\rangle$, $\langle$*feminism*, *idea*, 0.951$\rangle$, etc. 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 a x sense is linked to y synset with relatively short path. For a KS triple we should consider a subgraph of potential synsets for x. Its shape should depend on the nature of a given KS. For instance, as it is easier to mismatch synonymy and hypernymy then 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[1], 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 activation model follows a general

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

scheme, e.g. (Akim et al., 2011), in which initial activation is set at the start and then the node activation depends on the previous value and the activation coming from the connected nodes. The spreading is controlled by parameters representing the amount of *initial activation* and *activation decay*, respectively (Troussov et al., 2008). We identify activation with semantic support for x, the initial activation is called *direct support* while support coming from other nodes is called *indirect support*. Indirect support is intended to compensate errors of KSs and resolve the ambiguity of lemma-based information delivered in them.

Most frequent wordnet relations link synsets, but in every wordnet there are also many relations linking directly *lexical units* (LUs) (i.e. pairs word–sense number, 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 Spreading Activation models, the activation decay parameter $\mu \in [0, 1)$ and have the same value for all links. In our approach the activation decay value depends on the link types due to different distribution of errors across KSs. Following (Piasecki et al., 2012), that part of the decay dependent on the link type is represented by two functions: *transmittance* and *impedance*. Transmittance is a function: *lexico-semantic relation* $\rightarrow \Re$ and describes the ability of links to transmit support. Link-to-link connection is characterised by the *impedance* function: *relation pair* $\rightarrow \Re$. The impedance describes how much indirect support can be transferred through the given connection, e.g. the transmission of support through holonymy–meronymy would mean that the direct support assigned to the whole (a holonym) via a part (a meronym) could be attributed to another whole (its second holonym), e.g. *car*–holo–*windscreen*–mero:substance–*glass*: indirect support could go from *car* to *glass* that is clearly too far. By an appropriate impedance function we can reduce the transmission or block it, i.e. we can shape the considered part of the wordnet graph.

3.2 Algorithm

The algorithm works in four main steps preceded by the preparatory Step 0. First, the initial local support for LUs is calculated on the basis of KSs. Next, the local support is recursively replicated from LUs to local subgraphs of connected LUs. Support for synsets is calculated on the basis of their LUs. Finally, following (Piasecki et al., 2012), connected wordnet subgraphs such that each synset in a subgraph has some significant support are identified. Such subgraphs are called *activation areas*. Top several activation areas with the highest support value are selected as *attachment areas* – descriptions of potential senses of x. In each attachment area, the synset with the highest support is a potential place to add x sense. Attachment areas are next presented to linguists to explain the suggested meanings of x.

Let x be a new word, J be a set of LUs, L – a set of lemmas, and $\mathbf{A} \subseteq 2^{J^2}$ – a set of lexico-semantic relations defined on J (including relations inherited from synsets like hypernymy and *lexical relations*). A knowledge source K is a set of triples of the type: $L \times L \times \Re$ where $\Re$ is a set of real numbers. Let $\mathbf{K}$ be a set of all KSs and $\sigma : J \times L \rightarrow \Re$; $\sigma(j, x) = \sum_{K \in \mathbf{K}} K(j, x)$ equals the sum of all weights assigned to the pair. The *transmitation* is represented by: $f_T : \mathbf{A} \times \Re \rightarrow \Re$ and the *impedance* is represented by: $f_I : \mathbf{A}^2 \times \Re \rightarrow \Re$.

Step 0 Constructing a graph of LUs on the basis of the graph of synsets

Step 1 Setting up the initial state

1. $\forall_{j \in J}.\mathbf{Q}[j] = \sigma(j, x)$
2. `for each` $j \in J$ `if` $\mathbf{Q}[j]) > \tau_0$ add j to the queue T

Step 2 Support replication across the LU graph

1. $k =$ take first node from T
2. $supReplication(k, x, \sigma(k, x))$ – support for *x* is replicated from k onto the connected nodes
3. `if not` $empty(T)$ `then goto` 1

Step 3 Synset support calculation: `for each` s `in` Syn
`if` s does not have any support in any KS for x `then` $\mathbf{F}[s] = 0$
`else` $\mathbf{F}[s] = synsetSup(s, \mathbf{Q})$

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 local support (τ_0) are added to the queue as 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 (Łukasz Kłyk et al., 2012).

In Step 2 support replication is run for nodes stored in the queue and is described by the following functions (where j is a LU to be processed and M support value to be replicated, $dsc(j)$ returns the set of outgoing relation links and $p|_1$ returns the first element – a relation link target node).
$supReplication(j, x, M)$:
1) `if` $M < \epsilon$ `then return`
2) `for each` $p \in dsc(j)$
$supRepTrans(p, x, f_T(p, \mu * M))$

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

Incoming support is stored in the given node and part of it is spread further on according to μ. The parameter μ together with the transmittance function f_T corresponds to activation decay. The spreading stops when the incoming support goes down below ϵ and is additionally blocked on connections of the predefined types by the impedance function f_I. The value of ϵ was heuristically set to $\tau_0/2$, but it can be obtained during optimisation. The parameters μ and ϵ control (together) the maximal distance of the support flow.

In Step 3, support for synsets is calculated on the basis of the support for LUs included in them. It can be done in many different ways, but the best results were obtained by using a function proposed in (Piasecki et al., 2009b):
$synsetSup(S, Q') =$
1) `sum` $= \sum_{s_i \in S} Q'[s_i]$
2) `if` $\delta(1,$`sum`$, |S|) > 0$ `then return sum else return 0`
where $\delta(h, n, s) =$ `1 if` $(n \geq 1,5 * h \wedge s \leq 2) \vee (n \geq 2 * h \wedge s > 2)$ `else 0`

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

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 then hyponymy as its only relation. The whole activation area expresses a location found by the algorithm for x: however, we also need one particular synset to attach a LU for x. Thus, we look for local maxima of the support value and use these values as the semantic support for the whole attachment areas. *Paintball* is focused on supporting linguists, recall is important, so up to max_{att} activation areas are finally returned as suggested *attachment areas*.

4 Evaluation

4.1 Methodology

The evaluation is based on wordnet reconstruction task proposed in (Broda et al., 2011): randomly selected words are removed from a wordnet and next the expansion algorithm is applied to reattach them. Removing of every word changes wordnet structure, so it is best to remove one word at a time, but due to the efficiency, small word samples are processed in one go. As the algorithm may produce multiple attachment suggestions for a word, they are sorted according to semantic support of the suggested attachments. A histogram of distances between a suggested attachment place and the original synset is built. We used two approaches to compute the distance between the proposed and original synsets. 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.

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 and an optional meronymic link must be final. In this approach we consider close cousins (co-hyponyms) as valuable suggestions for linguists.

The collected results are analysed according to three strategies. In the *closest path* strategy we analyse only one attachment suggestion per

lemma that is the closest to any of its original locations. In the *strongest*, only one suggestion with the highest support for a lemma is considered. In the *all* strategy all suggestions are evaluated.

A set of test words was selected randomly from wordnet words according to the following conditions. Only words of the minimal frequency corpus 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.

4.2 Experiment setup

For the sake of comparison with (Snow et al., 2006) and (Piasecki et al., 2012) two similar KSs were built: a *hypernym classifier* and a *cousin classifier*. The first (Snow et al., 2004) was trained on English Wikipedia corpus (1.4 billion words) parsed by *Minipar* (Lin, 1993). We extracted all patterns linking two nouns in dependency graphs and occurring at least five times and used them as features for logistic regression classifier from *LibLINEAR*. Word pairs classified as hyperonymic were described by probabilities of positive decisions. Following (Piasecki et al., 2012), the cousin classifier was based on distributional similarity instead of text clustering as the clustering method was not well specified in (Snow et al., 2006). The cousin classifier is meant to predict (m, n)-cousin relationship between words. The classifier was trained to recognize two classes: $0 \leq m, n \leq 3$ and the negative. The measure of Semantic Relatedness (MSR) was used to produce input features to the logistic regression classifier. MSR was calculated as a cosine similarity between distributional vectors: one vector per a word, each vector element corresponds to the frequency of co-occurrences with other words 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 (Israel, 1992). Trained classifiers were applied to every pair: a test word and a noun from WordNet.

As a *baseline* we used the well known and often cited algorithm PWE (Snow et al., 2006). 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: $k = 4$, cousins neighbourhood size: $(m, n) \leq (3, 3)$, maximum links in hypernym graph: 10, penalization factor: $\lambda = 0.95$.

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 support spreading in *Paintball*. The graph was formed by hyper/hyponymy (H/h), holo/meronymy (o/m), antonymy (a) 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. The parameter α was 1 for other selected relations and 0 for non-selected. Impedance allows for controlling the shape of the spreading 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: $\langle$h, a$\rangle$, $\langle$h, m.$\rangle$, $\langle$H, h$\rangle$, $\langle$H, o$\rangle$, $\langle$a, a$\rangle$, $\langle$a, m$\rangle$, $\langle$a, o$\rangle$, $\langle$m, a$\rangle$ and $\langle$o, a$\rangle$.

4.3 Results

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

Considering straight paths and their maximal length up to 6 links PWE performs slightly better than Paintball. Coverage for words and senses is also higher for PWE: 100% (freq.: 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 words and ab-

		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

Table 1: Straight path 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**

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

stains in the case of the lack of evidence, e.g., for x=*feminism*, PWE suggests the following synset list: {*abstraction*, *abstract entity*}, {*entity*}, {*communication*}, {*group*, *grouping*}, {*state*}
while suggestions of *Paintball*, still not perfect, are 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 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. *Paintball* outperforms PWE in the evaluation based on the folded paths. For more than half test words, the strongest proposal was in the right place or up to a couple of links from it. Suggestions were generated for 72.65% of lemmas and the sense recall was 24.63% that is comparable with other algorithms.

5 Conclusions

We presented a new wordnet expansion algorithm called *Paintball*. It is based on a spreading activation model applied to the wordnet and expanded with notions of transmittance and impedance. The model enables combining different heterogeneous and partial KSs extracted from corpora. Contrary to many approaches, e.g. PWE (Snow et al., 2006), *Paintball* can use any KS, as it does not assume the probabilistic character of KSs. *Paintball* includes several parameters (but the same is the case of PWE), but their values can be tuned on a wordnet sample. *Paintball* offers a simpler and less heuristic model than LAAA and is a general tool. There are almost no works on wordnet expansion by spreading activation, e.g. (Liu et al., 2005) presented rather an idea, not a solution, but this model was used for Word Sense Disambiguation, e.g. (Tsatsaronis et al., 2007). Contrary to (Yang and Callan, 2009) we do not assume any properties of the lexical semantic network, but we try to shape it according to the language data. We aim also at an unsupervised or very weakly supervised algorithm in which training is limited to finding only general properties of the wordnet relations. *Paintball* expressed significantly better results than well known PWE and LAAA algorithms in test on performed on on Princeton WordNet 3.0.

Acknowledgments

Co-financed by the European Union within European Innovative Economy Programme project POIG.01.01.02-14-013/09 and by by the Polish National Centre for Research and Development project SyNaT.

References

Nazihah Md. Akim, Alan Dix, Akrivi Katifori, Giorgos Lepouras, Nadeem Shabir, and Costas Vassilakis. 2011. Spreading activation for web scale reasoning: Promise and problems. In *Proceedings of WebSci '11, June 14-17, 2011, Koblenz, Germany*.

Enrique Alfonseca and Suresh Manandhar. 2002. Extending a lexical ontology by a combination of distributional semantics signatures. In *13th Int. Conf. Knowledge Eng. and Knowledge Management. Ontologies and the Semantic Web*, LNCS. Springer.

Bartosz Broda, Roman Kurc, Maciej Piasecki, and Radosław Ramocki. 2011. Evaluation method for automated wordnet expansion. In P. Bouvry, M. Kłopotek, F. Leprevost, M. Marciniak, A. Mykowiecka, and H. Rybiński, editors, *Security and Intelligent Information Systems*, LNCS. Springer.

Allan M. Collins and Elizabeth F. Loftus. 1975. A spreading-activation theory of semantic processing. *Psychological Review*, 82(6):407–428.

Christiane Fellbaum, editor. 1998. *WordNet — An Electronic Lexical Database*. The MIT Press.

Birgit Hamp and Helmut Feldweg. 1997. GermaNet – a Lexical-Semantic Net for German. In *Proceedings of the ACL Workshop on Automatic Information Extraction and Building of Lexical Semantic Resources for NLP Applications*, pages 9–15. Madrid.

G. Israel. 1992. Determining sample size. Tech. rep., University of Florida.

Zornitsa Kozareva and Eduard Hovy. 2010. A semi-supervised method to learn and construct taxonomies using the web. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing, MIT, Massachusetts, USA, 9-11 October 2010*, pages 1110–1118. ACL.

Dekang Lin. 1993. Principle-based parsing without overgeneration. In *Proc. ACL-93, Columbus, Ohio*.

Wei Liu, Albert Weichselbraun, Arno Scharl, and Elizabeth Chang. 2005. Semi-automatic ontology extension using spreading activation. *Journal of Universal Knowledge Management*, 0(1):50–58.

Marek Maziarz, Maciej Piasecki, and Stan Szpakowicz. 2012. Approaching plWordNet 2.0. In Christiane Fellbaum and Piek Vossen, editors, *Proceedings of 6th International Global Wordnet Conference*, pages 189–196, Matsue, Japan, January. The Global WordNet Association. Book: `http://www.globalwordnet.org/gwa/proceedings/gwc2012.pdf`.

Roberto Navigli, Paola Velardi, and Stefano Faralli. 2011. A graph-based algorithm for inducing lexical taxonomies from scratch. In *Proceedings of IJCAI*.

Maciej Piasecki, Bartosz Broda, Maria Głąbska, Michał Marcińczuk, and Stan Szpakowicz. 2009a. Semi-automatic expansion of polish wordnet based on activation-area attachment. In *Recent Advances in Intelligent Information Systems*, pages 247–260. EXIT.

Maciej Piasecki, Stanisław Szpakowicz, and Bartosz Broda. 2009b. *A Wordnet from the Ground Up*. Oficyna Wydawnicza Politechniki Wrocławskiej, Wrocław.

Maciej Piasecki, Roman Kurc, and Bartosz Broda. 2011. Heterogeneous knowledge sources in graph-based expansion of the polish wordnet. In *Proc. of The 2nd Asian Conference on Int. Inf. and Database Systems*, number 6591 in LNAI. Springer.

Maciej Piasecki, Roman Kurc, Radosław Ramocki, and Bartosz Broda. 2012. Lexical activation area attachment algorithm for wordnet expansion. In Allan Ramsay and Gennady Agre, editors, *Proceedings of the 15th International Conference on Artificial Intelligence: Methodology, Systems, Applications*, volume 7557 of *Lecture Notes in Computer Science*, pages 23–31, Varna, Bulgaria. Springer.

G. Salton and C. Buckley. 1988. On the use of spreading activation methods in automatic Information Retrieval. In *Proceedings of ACM SIGIR*.

Rion Snow, Daniel Jurafsky, and Andrew Y. Ng. 2004. Learning syntactic patterns for automatic hypernym discovery. In *NIPS*.

Rion Snow, Dan Jurafsky, and Andrew Y. Ng. 2006. Semantic taxonomy induction from heterogenous evidence. pages 801–808. The Association for Computer Linguistics.

Alexander Troussov, Mikhail Sogrin, John Judge, and Dmitri Botvich. 2008. Mining socio-semantic networks using spreading activation technique. In *Proceedings of I-KNOW '08 and I-MEDIA '08 Graz, Austria, September 3-5, 2008*, pages 405–412.

George Tsatsaronis, Michalis Vazirgiannis, and Ion Androutsopoulos. 2007. Word sense disambiguation with spreading activation networks generated from thesauri. In *Proceedings of IJCAI-07*, pages 1725–1730.

Łukasz Kłyk, Paweł B. Myszkowski, Bartosz Broda, Maciej Piasecki, and David Urbansky. 2012. Metaheuristics for tuning model parameters in two natural language processing applications. In Allan Ramsay and Gennady Agre, editors, *Proceedings of the*

15th International Conference on Artificial Intelligence: Methodology, Systems, Applications, volume 7557 of *Lecture Notes in Computer Science*, pages 32–37, Varna, Bulgaria. Springer.

D. Widdows. 2003. Unsupervised methods for developing taxonomies by combining syntactic and statistical information. In *Proc. HLT of North American Chapter of the ACL*.

Hans Friedrich Witschel. 2005. Using decision trees and text mining techniques for extending taxonomies. In *Proc. of Learning and Extending Lexical Ontologies by using Machine Learning Methods, Workshop at ICML-05*.

Hui Yang and Jamie Callan. 2009. A metric-based framework for automatic taxonomy induction. In *Proceedings of the 47th Annual Meeting of the ACL and the 4th IJCNLP of the AFNLP*, pages 271–279. ACL.