Heterogeneous Knowledge Sources in Graph-Based Expansion of the Polish Wordnet

Maciej Piasecki, Roman Kurc, and Bartosz Broda

Institute of Informatics, Wrocław University of Technology, Poland
{maciej.piasecki,roman.kurc,bartosz.broda}@pwr.wroc.pl

Abstract. The paper presents an algorithm of automatic wordnet expansion on the basis of heterogeneous knowledge sources extracted from a large corpus. The algorithm is the reformulation of the algorithm used in the WordnetWeaver system in terms of the SOM model. Integration of knowledge sources is based on the weighted voting scheme. Several wordnet relations are explored to define attachment points for a new word. Influence of different knowledge sources on the algorithm performance was experimentally investigated. The new version presents better precision than the previous one.

1 Motivation and Related Works

We present a high-accuracy automatic method of identifying semantic similarity of lexical units (LUs)[1] on the basis of knowledge extracted from a corpus. If effective, such a method is directly applicable in semi-automatic wordnet expansion: it makes good suggestions for the linguist. We have tested our method on a new and growing wordnet for Polish – namely plWordNet.

Clustering algorithms applied to words described by semantic relatedness seem to be a natural way in automated wordnet construction. However, clustering usually produces a flat set of clusters. Changing such a set into a hierarchy poses two problems: how to identify the right shape of the tree and how to label higher levels of the cluster tree with the adequately general LUs. In any case, no automatic method can come up with a credible top portion of a wordnet hierarchy due to the highly abstract meanings of LUs occurring on these levels. Thus, we follow a semi-automatic wordnet expansion model: top levels of plWordNet's hypernymy hierarchy have been built manually and automatic methods produce useful suggestions of new LUs for inclusion in plWordNet (henceforth plWN).

Several projects have explored building an extended wordnet over an existing one. The advantage is the possibility of using the wordnet structure already in place, especially the hypernymy structure, as a knowledge source. [1] assigned a meaning representation extracted by Distributional Semantics methods to synsets, and treated the hypernymy structure so labelled as a kind of decision tree. [13] discusses a more radical decision-tree model with recursive upward propagation of meaning descriptions to the root. The description of the upper

[1] We will take a lexical unit, a little informally, to be a lexeme.

N.T. Nguyen, C.-G. Kim, and A. Janiak (Eds.): ACIIDS 2011, LNAI 6591, pp. 307–316, 2011.
© Springer-Verlag Berlin Heidelberg 2011

nodes represents the description of descendants. A synset's *semantic description* comprises LUs most similar to its LUs. Distributionally motivated similarity is based on co-occurrences of LUs in corpora. In evaluation on two subtrees from GermaNet, *Moebel* (*furniture*) (144 children) and *Bauwerk* (*building*) (902 children), the best accuracy of exact classification was 14% and 11%, comparable to that in [1].

[12] represents LU meaning by *semantic neighbours* – k most similar LUs. To attach a new LU is to find a site in the hypernymy hierarchy where its semantic neighbours are concentrated. To compute semantic similarity, each LU is first described by the co-occurrence, in a 15-word window, with 1000 most frequent one-word LUs. Evaluation was on the British National Corpus [2] and randomly selected common nouns, 200 each from three frequency ranges: > 1000, $[500, 1000]$ and < 500. Sites identified by the algorithm were compared with their exact hypernyms. The best accuracy of finding the direct hypernym, with no intervening nodes, producing exact reproduction of the wordnet (among 4 highest ranked labels) was 15% for $k = 3$ neighbours taken into account, but the overall classification (considering hypernyms located up to 10 links away from the suggested site in the wordnet structure) gave only 42.63%. The best accuracy of the overall classification was 82.06% for $k = 12$ neighbours considered but the accuracy of the exact placing (finding direct hypernyms) was reduced to 10.15%.

[11] cast the extension of wordnet hypernymy structure in terms of a probabilistic model. Attachment of new elements transforms the former structure **T** into a new structure **T**'. The most appropriate **T**' maximises the probability of the change in relation to the evidence at hand. [11] applied two sources of evidence for extending Princeton WordNet (henceforth PWN) [3]: a classifier-based algorithm of extracting hypernymic LU pairs using lexico-syntactic relations, and a proposed algorithm of extraction of (m, n)-*cousins*. In a "fine-grained" evaluation, [11] manually evaluated randomly selected 100 samples from the first n up to 20000 automatically added hypernymic links. The applied uniform size of the samples equal to 100 for $n > 1000$ was too small to ascribe the results of the evaluation to the whole sets with sufficient statistical confidence. The evaluators were asked: "is X a Y?", where $\langle X, Y \rangle$ is an added link. It is not clear in [11] whether only direct hyponym/hypernyms counted as positive. For each pair of nouns, the algorithm finally selects only the best hit for a new lexeme. The achieved precision of 84% for $n = 10000$ is high, but may be hard to compare with other approaches, including ours (presented in a while), because it is given only for the best hit and the basic criterion cited above is not precise.

2 The Main Ideas

In [7] an algorithm of wordnet expansion was proposed. The algorithm combined several knowledge sources extracted from large corpora. It was successfully applied to plWN expansion, however, the algorithm depended in several steps on heuristic procedures. In this paper we want to reformulate it in terms of unifying basic model and explore its several possible extensions.

We applied heterogeneous knowledge sources for a potential relation of a new LU x with a LU already in the wordnet and thus a relation with some synset. The sources were produced by several methods of extracting lexical-semantic relations (LSR) for Polish. The results of all extraction methods were transformed to sets of LU pairs $\langle x, y \rangle$ such that x and y are semantically related according to the given method and the corpora analysed. Five methods were used:

- a Measure of Semantic Relatedness (MSR) based on the Rank Weight Function (MSR_{RWF}) developed for Polish nouns by [9] (while it extracts closely related LUs with high accuracy, the extracted LU pairs belong to more LSRs than just the typical wordnet relations);
  two sets were produced using MSR_{RWF} – the set $MSRset(y, k)$ of the k units most related to y, where $k = 20$ was used in all experiments, and that set restricted to *bidirectional relations*:
  $MSR_{BiDir}(y, k) = \{y' : y' \in MSRset(y, k) \wedge y \in MSRset(y', k)\}$;
- a C_H classifier [8] which we use for post-filtering LU pairs produced by MSR_{RWF}. C_H is described below in more details. We generated one set by the classifier C_H applied to filtering $MSRset(y, k)$ from LUs not in hypernymy, meronymy or synonymy with y;
- three manually constructed lexico-syntactic patterns in the style of [4]: ⟨NP, NP, … i inne (*and others*) NP⟩, ⟨NP jest (*is a*) NP⟩ and ⟨NP to (*is a*) NP⟩;
- six manually constructed patterns, similar to the above ones, but focused on mining Wikipedia.
- the *Estratto* method [6] in which lexico-morpho-syntactic patterns automatically extracted from corpus are used to extract LSR instances.

In general C_H was trained on LU pairs extracted from plWN: pairs of LUs associated by the synonymy, hyper/hyponymy (up to the distance of 2 links) and mero/holonymy relation — positive examples, and pairs of LUs which are not associated by any of the three relations — negative examples.

The accuracy of all methods in distinguishing *related LUs* (positive examples in C_H) is around 30%. This is in fact too low to support linguistic work effectively. But positively verified in [7] that by combining the results of different methods we can provide linguist with LU pairs of better accuracy.

Even though the combined method classifies LU pairs as related vs. non interesting well, it still cannot be used to distinguish among different wordnet relations. On the other hand when processing a new LU, all sites of its attachment to the wordnet structure are almost equally important. Thus, we proposed an automatic method of *activation-area attachment*: a new LU is attached to a small area in the hypernymy graph rather than to one synset, cf [10].

The method was inspired by the idea of learning in *Kohonen's networks* [5]. In a Kohonen network, a new learning example is used to modify not only the most similar neuron (the *winner*) but also neurons located close to it. The further the given neuron is from the winner, the smaller is its change caused by this learning example. The distance is measured by the number of links in the graph structure of the network. We aim at finding, for a new LU u, synsets for which we have the strongest evidence of LUs being in the close hypo/hypernym/synonym relation with u. Ideally that synsets should include near synonyms of the new LU. We assume, that the intrinsic errors in data preclude certainty about the exact attachment point for u. Even if synset t appears to fit, we must consider the evidence for t's close surroundings in the hypernymy structure – all synsets located no further than some upper-bound distance d. We treat the evidence from the surroundings as less reliable than that for LUs in the central synset t. The influence of the context evidence decreases with the distance.

As the knowledge sources overlap only partially, so we will use them all in extending the wordnet. We assume that the subsequent methods explore different pieces of partial information available in corpora. We assume, too, that the application of many different methods allows the use of as much lexical-semantic information as possible. Different sources are not equally reliable; this can be estimated, e.g., by manual evaluation of the accuracy of the extracted pairs. To trust the different sources to a different degree, we introduce weighted voting.

3 Algorithm of Activation-Area Attachment

The algorithm, henceforth AAA, is based on the idea of a *semantic fit*: between two lemmas, as representing two LUs linked by a LSR, and between a lemma and a synset, as defining a LU. The fit is computed from all evidence found in corpora. Next we group synsets that fit the input lemma into the *activation-areas* defining the attachment areas and LUs. AAA was presented in [7]. Here we aim at improving it in three aspects: calculation of lemma-to-lemma and lemma-to-synset *score and fit*, and the *range of relations* utilised. Score and fit functions were made more robust and are expressed now in terms of weighted voting.

In [10] the fit was meant to provide heuristic combination of knowledge sources, some of them sparse but highly accurate. The score was crafted to prioritize fit results and supplement description for pairs with fit equal 0. Score is based on MSR which is defined for every lemma pair, while other sources are partial. We found that approach excessively complicated and we propose an unified measure. Every source is assigned some weight of its reliability, e.g. on the basis of its precision manually evaluated. Next, weighting voting scheme is applied in order to combine support provided by different knowledge sources. Different schemes cab be used to calculate the final value of lemma-to-lemma fit or score. Two functions were used: sigmoid and normalization.

A lemma-to-lemma fit is a function $fit : \mathbf{L} \times \mathbf{L} \rightarrow \langle 0.0, 1.0 \rangle$, where $\mathbf{L}$ is a set of lemmas, which is calculated as following (each $k \in K$ is assigned its weight):
Let $norm = \sum_{k \in K} weight(k)$, then
$fit(x, x') = (\sum_{k \in K \wedge (x,x') \in pairs(k)} weight(k))/norm$
Where $weight$ maps a relation name to a weight (a real number).

AAA consists of two phases: *Phase I*, when we find all synsets that fit a new lemma and *Phase II* when we group the found synsets into connected subgraphs – *activation areas*. **Phase I** is inspired by Self Organizing Map. Input data in SOM is a vector of a finite size. In general SOM requires a similarity function and a distance function. During learning we use similarity function to choose neuron that is closest to some input. Then the-winner-takes-most strategy and the distance function are used to determine which neurons are adapted to new data. During application the similarity is used to locate data on the map.

In AAA the similarity is identified with the lemma-to-synset fit, which is based on the lemma-to-lemma fit and refers to the synset neighbourhood. Thus the whole surroundings is considered to be a vector describing the synset. The traversing order is bread first search. During **Phase I** we use the winner takes most strategy, as adding a new LU is equal to adapting synsets in close neighbourhood. Attachment of a new LU is similar to the SOM learning phase during which one piece of data is mapped to one SOM area. AAA seems to work in a opposite way: one lemma can be attached to a number of synsets. But, there is no multiple mapping by the same input data as one lemma can have many meanings. Therefore we perform disambiguation. There are several issues to be considered. First of all there is a question whether LUs in synsets should be treated as separate wordnet elements or only as parts of the synset. In the second case the size of the synset is important. The more elements from the synset are related to the new lemma, the better. Then question arises which elements of the neighbourhood should be taken into consideration, i.e., number of links, distance and relation types. Up to now only hypernymy/hyponymy was taken into consideration. However other relations are also valuable for linguists and covered by knowledge sources. Thus we need to take them into account. In **Phase II** the found synsets are grouped into connected subgraphs – activation areas. Let:

- x is a lemma, representing one or more LUs, to be added to the wordnet, S, S' – wordnet synsets, $|S|$ – synset size,
- $hMSR$ (set to 0.4) is the threshold defining highly reliable MSR values,
- $minMSR$ (set to 0.1) is the MSR value below which associations seem to be based on weak, accidental clues,
- $maxSens$ (set to 5) is the maximal number of presented attachment areas.
- $G\{relations\}$ wordnet projection with only links of the *relations* preserved,
- d is the upper-bound distance defining the hypernymic close surroundings,
- $bfs(S, G)$ returns synset and surrounding synsets in breadth first order.
- $dist(S, S', G)$ is the number of *relations* links between S and S',
- $dist\_modif(S, S')$ give value to modify fit of supporting synsets.

Phase I. Lemma-to-synset calculation

According to description of **Phase I** presented above functions of distance and similarity must be pre-defined.

1. $similarity(x, S) = \left(fit(x, S) + \sum_{S' \in bfs(S)} fit(x, S') * dist\_modif(S, S')\right)$
2. $strong\_fit(x, S) = \delta(1, similarity(x, S), |\mathrm{S}|)$
3. $weak\_fit(x, S) = \delta(hMSR, similarity(x, S), |\mathrm{S}|)$

where $\delta : \Re \times \Re \times \Re \to \{0, 1\}$, such that $\delta(h, n, s) = 1$
if and only if $(n \geq 1.5 * h$ and $s \leq 2)$ or $(n \geq 2 * h$ and $s > 2)$

Phase II. Identify lemma senses: areas and centres

1. $synAtt(x) = \{\mathbf{S} : \mathbf{S} = \{S : strong\_fit(x, S) \vee weak\_fit(x, S)\}$, and $\mathbf{S}$ is in G$\}$.

2. $maxScore(x, \mathbf{S}) = score\,(x, max_{S \in \mathbf{S}} score(x, S))$
3. Remove from $synAtt(y)$ all $\mathbf{S}$ such that $maxScore(x, \mathbf{S}) < minMSR$
4. Return the top $maxSens$ subgraphs from $synAtt(x)$ according to their $maxScore$ values; in each, mark the synset S with the highest $score(x, S)$.

The upper-bound distance d was set to 2 (**Phase I**), because we observed that no extraction method used can distinguish between direct hypernyms and close hypernyms. The δ function is a means of non-linear quantization from the strength of evidence to the decision. We require more *yes* votes for larger synsets, fewer votes for smaller synsets, but always more than one 'full vote' must be given – more than one synset member voting *yes*. The parameter h of δ relates the function to what is considered to be a 'full vote'. For weak fit, h is set to the value which signals a very high relatedness for the MSR used.

In **Phase II** we identify continuous areas (connected subgraphs) in the hypernymy graph, those which fit the new lemma x. For each area we find the local maximum of the score function for x. We keep all subgraphs with the synset of the maximum score based on the strong fit (the detail omitted above). From those based on the weak fit, we only keep the subgraphs above some heuristic threshold of the reliable MSR result. We also save for the linguists only a limited number of the best-scoring subgraphs ($maxSens = 5$ – it can be a parameter of the application). We do not want to clutter the screen with too many proposals. We do present all subgraphs with the top synset fit based on the strong fit.

4 Evaluation

Evaluation was based on the same automatic method as in [7,10]: AAA is first applied to the reconstruction of a wordnet from which a sample of LUs has been removed; the removed LUs are re-introduced with the help of AAA, and the precision of the process is attributed to AAA as its assessment. Laborious manual evaluation was not applied as its results seem to be correlated with those of the automatic one [7,10]. Two versions of plWN were used during evaluation: the so called plWN Core (plWNCore) version – built manually, the basis of evaluation in [7,10], and a more recent plWN version 1.0, cf [10]. We used the same sample of 1527 noun lemmas from lower parts of the hypernymy structure of plWNCore, as was applied in tests in [7]. In addition the second sample of 1658 noun lemmas was picked from plWN 1.0 in a slightly different way than the firs one: we selected LUs that were located in the lower parts of the hypernymy structure and were weakly connected with the whole structure of plWN 1.0. This sample represents less intensively developed parts of plWN. During one step of evaluation 10 noun lemmas are removed and re-inserted by AAA to the wordnet. If a LU was the only element of a synset, LU was removed, but the empty synset was left in the structure in order not to change the wordnet structure so significantly for the reconstruction step (a missing synset separates hypernymic subgraphs).

Reconstruction results were assessed according to the *single highest-scoring attachment site* strategy (called *One* in [10]) We follow this strategy for the needs of comparison with other approaches. It is a little artificial in relation to the intended use WNW as tool, but it illustrates AAA behaviour when it works in a fully unsupervised way.

Two AAA variants was tested, as well, as different configurations of AAA parameters. As a baseline we used AAA version from [7], in which lemma-to-lemma fit was calculated according to a heuristic procedure. In all other experiments AAA version described in this paper, i.e., in which lemma-to-lemma fit is based on weighted voting discussed in Sec. 3, henceforth, it is called *weighted AAA*.

In addition to the basic AAA described in Sec. 3 we tested also a version in which not only hypernymic edges, but also edges of other relations were taken into account during calculation of the lemma-to-synset fit. The set of relations along which fit was collected in the synset context included: hypernymy/hyponymy, meronymy and antonymy. Each edge was assigned a weight depending on the lexico-semantic relation it represents. Weights were meant to influence fit values 'transmitted' through them. Hypernymy and hyponymy's factor was set to 1.0 while for the meronymy and holonymy it was experimentally set to 0.7. Factors were later used to modify the value of the score added to synset S by neighbour synset S'. The modifier for fit is calculated as follows: $mod(S, S') = \prod_{e' \in \{e : e \in shortest\_path(S,S')\}} relation\_weights(e')$, where e' is an edge, and $relation\_weights$ maps an edge to the factor.
The fits are then combined 3: $Sm = \sum_{(s,f) \in (fits, factors)} s * \frac{1}{2^n} * f$.

Experiments were performed on samples of plWN lemmas for AAA variants discussed above. Influence of the values of AAA parameters were tested. Firstly, basic algorithms: the *baseline* and *weighted* AAA were applied on: plWNCore and plWN 1.0. They differ in size and the richness of LU description. We wanted to analyse the impact of the wordnet development on the AAA performance. In addition, we wanted to evaluate how our new weighted model handles test data introduced in [10]. Secondly, we prepared a new test set of different characteristic, i.e., the set consists of weakly connected LUs. This set is meant to show how well the AAA manages deficiencies in the wordnet structure. Thirdly, the AAA variant sensitive to several relations in the context was tested. Next, we analyses how the number of knowledge sources applied impacts the weighted AAA performance especially when they do not overlap.

Finally, the influence of knowledge sources based on the measure of semantic relatedness was investigated. This experiment shows how different types of knowledge sources contribute to the overall AAA performance. Source were divided in two groups: SIM consisting of sources based on the semantic relatedness and PAT – including pattern-based sources.

plWN has been continuously expanded by adding new LUs and links, as well, as correcting errors. Table 1 shows that AAA achieves better precision when applied to a larger and more coherent wordnet. Especially the weighted AAA gains from the richer structure of WN. On one hand hand, weighted AAA is more selective. But on the other hand, bigger network size balances filtering characteristic of the weighted AAA and results in more attachment suggestions. We can observe this on both: old and new test data sets. Interesting results obtained

Table 1. Tab1. core WN vs. WN 1.0: baseline, weighted (one)

	wordnet core - old test sample						wordnet 1.0 - old test sample					
	baseline			weighted			baseline			weighted		
L	S	W	S+W	S	W	S+W	S	W	S+W	S	W	S+W
0&1	40,6%	13,2%	33,3%	13,3%	8,3%	8,6%	46,4%	12,5%	35,4%	64,5%	32,8%	39,1%
2	59,6%	28,4%	51,3%	23,3%	12,9%	13,7%	65,2%	31,5%	54,3%	81,0%	51,7%	57,5%
3	69,3%	35,2%	60,1%	44,4%	24,3%	25,7%	73,0%	39,3%	62,1%	87,9%	60,7%	66,1%
4	76,4%	43,5%	67,6%	55,6%	34,2%	35,6%	79,6%	47,1%	69,1%	92,1%	68,3%	73,0%
5	81,9%	51,6%	73,8%	72,2%	43,0%	45,0%	83,6%	53,9%	74,0%	93,1%	73,6%	77,5%
Hits:	1080	395	1475	90	1206	1296	987	473	1460	290	1171	1461

	wordnet 1.0 - new test sample					
	baseline			weighted		
L	S	W	S+W	S	W	S+W
0&1	26,4%	8,8%	19,0%	37,0%	16,7%	19,2%
2	32,3%	12,1%	23,8%	46,3%	21,3%	24,4%
3	35,7%	14,0%	26,6%	50,9%	23,8%	27,2%
4	36,9%	14,8%	27,6%	51,9%	24,7%	28,1%
5	38,3%	15,4%	28,7%	54,6%	25,9%	29,5%
Hits	504	364	868	108	760	868

Table 2. Tab2 WN1.0(one): baseline, weighted, relations(mero, hipo, anto)

	baseline			weighted			relations		
L	S	W	S+W	S	W	S+W	S	W	S+W
0&1	26,39%	8,79%	19,01%	37,04%	16,71%	19,24%	44,64%	17,02%	20,60%
2	32,34%	12,09%	23,85%	46,30%	21,32%	24,42%	51,79%	21,41%	25,35%
3	35,71%	14,01%	26,61%	50,93%	23,82%	27,19%	55,36%	24,07%	28,13%
4	36,90%	14,84%	27,65%	51,85%	24,74%	28,11%		25,13%	29,05%
5	38,29%	15,38%	28,69%	54,63%	25,92%	29,49%	59,82%	26,06%	30,44%

on the new test data proved that weighted AAA can handle well described areas as well as sparse ones.

For a larger wordnet weighted AAA is better than baseline, cf Table 2. It is, however, restricted by available types of links between synsets. In the second phase (lemma to synset), when neighbourhood is important, additional links improve support for a target synset. This can be observed in the case of the last method presented in Table 2. Reliability of different relation links for AAA must be still investigated. However, as far as hypernymy is very informative, meronymy or antonymy may be used only to the limited extent.

Table 3 shows that increasing overlap between sources of evidence results in better precision without the loss in the number of suggestions. However when overlap is lower, i.e., baseline AAA resorts to the minority voting (weak fit), weighted AAA turns out to be much more selective.

Last set of experiments, cf Table 4, shows how different types of knowledge sources influence results. In general sources based on similarity guarantees large number of suggestions. At the same time other sources helps only in case of the

Table 3. Influence of evidence number[basic, wiki op syntactic, wiki op semantic, head] baseline, weighted

	weighted			w+op_synt			w+syn_sem		
L	S	W	S+W	S	W	S+W	S	W	S+W
0&1	31,4%	16,1%	20,5%	37,0%	16,7%	19,2%	45,5%	24,6%	25,0%
2	40,0%	20,2%	25,8%	46,3%	21,3%	24,4%	54,5%	32,0%	32,4%
3	43,9%	23,0%	29,0%	50,9%	23,8%	27,2%		35,1%	35,5%
4	46,7%	24,3%	30,6%	51,9%	24,7%	28,1%	63,6%	37,5%	38,0%
5	51,0%	26,0%	33,1%	54,6%	25,9%	29,5%	72,7%	40,8%	41,4%
hits total no:	255	639	894	108	760	868	11	578	589

Table 4. Distributional evidence vs. pattern based evidence

	baseline			sim only			no sim		
L	S	W	S+W	S	W	S+W	S	W	S+W
0&1	37,0%	16,7%	19,2%	31,0%	12,1%	19,1%	40,0%	5,7%	7,9%
2	46,3%	21,3%	24,4%	39,0%	15,2%	24,0%	50,0%	7,7%	10,4%
3	50,9%	23,8%	27,2%	42,7%	17,2%	26,6%	56,7%	9,3%	12,3%
4	51,9%	24,7%	28,1%	44,3%	18,5%	28,0%	63,3%	10,4%	13,8%
5	54,6%	25,9%	29,5%	45,8%	19,4%	29,1%	70,0%	12,0%	15,7%
hits total no:	108	760	868	323	552	875	30	441	471

strong match. They cannot improve weak match because of their low quality and weak match represents situations when minority of sources votes.

5 Conclusions and Further Research

We presented a SOM inspired reformulation of the AAA. Secondly, several profitable AAA extensions were discussed, including the treatment of the increasing number of knowledge sources and the utilisation of several lexico-semantic relations present in the wordnet.

In general larger number of knowledge sources improves AAA precision. On the other hand if sources do not overlap then the AAA produces less suggestions due to the lack of synsets supported by the strong fit. Similarly, if we use larger number of relations, i.e., not only hyper/hyponymy links but also meronymy and antonymy ones, AAA comes up with better attachment suggestions for new lemmas. Moreover, we showed that expansion strategies should be adjusted accordingly to the maturity level of the wordnet. Problems of the insufficient description of particular synsets and different importance of relations for AAA must be further investigated. A comparison to other works on automatic wordnet expansion can be misleading. Our primary goal was to construct a tool that facilitates and streamlines the linguists' work. Still, even if we compare our automatic evaluation with the results in [12] during comparable tests on the Princeton WordNet, our results seem to be better. For example, we had 19.2% to 25%(new test data) and 39.1%(old test data) for the highest-scored proposal

(Table 1), while [12] reports a 15% best accuracy for a "correct classifications in the top 4 places" (among the top 4 highest proposals). Our similar result for the top 5 proposals is even higher, 21% up to 41% (new test data) and 77,5% for the old test data. The best results in [1,13], also at the level of about 15%, were achieved in tests on a much smaller scale. [13] also performed tests only in two selected domains. The algorithm of [11], contrary to ours, can only be applied to probabilistic evidence.

Acknowledgements. Work financed by the Polish Ministry of Education and Science, Project N N516 068637.

References

1. Alfonseca, E., Manandhar, S.: Extending a lexical ontology by a combination of distributional semantics signatures. In: Gómez-Pérez, A., Benjamins, V.R. (eds.) EKAW 2002. LNCS (LNAI), vol. 2473, pp. 1–7. Springer, Heidelberg (2002)
2. BNC. The British National Corpus, version 2 (BNC World) distributed by Oxford University Computing Services on behalf of the BNC Consortium (2001), http://www.natcorp.ox.ac.uk/
3. Fellbaum, C. (ed.): WordNet: An Electronic Lexical Database. MIT Press, Cambridge (1998)
4. Hearst, M.A.: Automated Discovery of WordNet Relations. In: Fellbaum [3], pp. 131–153
5. Kohonen, T.: Self-organized formation of topologically correct feature maps. Biological Cybernetics 43(1), 59–69 (1982)
6. Kurc, R., Piasecki, M.: Automatic acquisition of wordnet relations by the morpho-syntactic patterns extracted from the corpora in polish. In: 3rd International Symposium Advances in Artificial Intelligence and Applications (2008)
7. Piasecki, M., Broda, B., Głąbska, M., Marcińczuk, M., Szpakowicz, S.: Semi-automatic expansion of polish wordnet based on activation-area attachment. In: Recent Advances in Intelligent Information Systems, pp. 247–260. EXIT (2009)
8. Piasecki, M., Szpakowicz, S., Marcinczuk, M., Broda, B.: Classification-based filtering of semantic relatedness in hypernymy extraction. In: Nordström, B., Ranta, A. (eds.) GoTAL 2008. LNCS (LNAI), vol. 5221, pp. 393–404. Springer, Heidelberg (2008)
9. Piasecki, M., Szpakowicz, S., Broda, B.: Automatic selection of heterogeneous syntactic features in semantic similarity of Polish nouns. In: Matoušek, V., Mautner, P. (eds.) TSD 2007. LNCS (LNAI), vol. 4629, pp. 99–106. Springer, Heidelberg (2007)
10. Piasecki, M., Szpakowicz, S., Broda, B.: A Wordnet from the Ground Up. Oficyna Wydawnicza Politechniki Wrocławskiej, Wrocław (2009)
11. Snow, R., Jurafsky, D., Ng, A.Y.: Semantic taxonomy induction from heterogenous evidence. In: COLING 2006 (2006)
12. Widdows, D.: Unsupervised methods for developing taxonomies by combining syntactic and statistical information. In: Proc. of NAACL-HLT, pp. 197–204 (2003)
13. Witschel, H.F.: 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 2005 (2005)