A Machine Learning Approach to Building Aligned Wordnets

Gerard de Melo
Max Planck Institute for Informatics
Campus E1 4
66123 Saarbrücken, Germany
demelo@mpi-inf.mpg.de

Gerhard Weikum
Max Planck Institute for Informatics
Campus E1 4
66123 Saarbrücken, Germany
weikum@mpi-inf.mpg.de

Abstract

WordNet is a lexical database describing English words and their senses. We propose a method for automatically producing similar resources for new languages by taking advantage of the original WordNet in conjunction with translation dictionaries. A small set of training mappings is used to learn a model for predicting associations between terms and senses. The associations are represented using a variety of scores that take into account structural properties as well as semantic relatedness and corpus frequency information. For evaluation, we created a German-language wordnet, and the data indicate a significantly better coverage and higher precision than previous heuristics. The resulting resources provide not only valuable information for monolingual NLP tasks but also enable a high degree of cross-lingual interoperability.

1 Introduction

Princeton WordNet (Fellbaum, 1998) is a well-known lexical resource that provides information about how words and word senses in the English language are linked. It lists the senses that a word can assume and provides structural information about how such senses are related, e.g. via the hypernymy relation that holds when one term is a generalization of another term, e.g. "`publication`" is a hyponym of "`journal`". The original WordNet for the English language later inspired endeavours to create similarly structured databases ("wordnets") for other languages, e.g. in the context of the EuroWordNet EU project (Vossen, 1998), the BalkaNet project, as well as under the auspices of the Global WordNet Association. Nevertheless, we contend that despite several decades of work on such resources, there is still a great need for additional research into more efficient means of producing them. Consider, for instance, that there are about 7,000 living languages, but only around 40 for which wordnet versions have been created, many indeed still in a preliminary stage with very low coverage, and less than a handful of languages with wordnet versions that are freely downloadable from the Internet. Furthermore, several existing wordnets unfortunately form completely independent networks that are not connected in any way to other wordnets.

In order to complement the existing manually compiled wordnets, we thus propose a new approach to building wordnets that trades off accuracy for a much faster compilation process, and hence frequently leads to more terms being covered than in existing wordnets. Our approach is based on learning classifications, and therefore is completely automatic once an initial set of training mappings is provided. Certainly, the resulting wordnets will not have the same level of accuracy as resources carefully constructed by expert lexicographers, however they can 1) serve as a valuable starting point for creating more accurate ones, and 2) be used immediately in many natural language processing tasks where coverage is more important than perfect accuracy. The fact that the wordnets are aligned with the Princeton WordNet greatly facilitates interoperability with existing wordnets (e.g. English-language glosses are available) as well as with additional resources such as topical domain labels (Bentivogli et al., 2004) or mappings to ontologies (Niles and Pease, 2003; Suchanek et al., 2007). This additional information is an enormous benefit in practical applications.

The remainder of this paper is organized as follows. Based on a brief introduction to classification learning in Section 2, we present our wordnet building approach in Sections 3 and 4. This approach is then evaluated in Section 5, while Section 6 compares it to other existing work in this field. Finally, concluding remarks are provided in Section 7.

2 Preliminaries

A *classification* is an assignment of class labels $y \in \mathcal{Y}$ to objects $x \in \mathcal{X}$ in the form of a function $\widehat{f} : \mathcal{X} \times \mathcal{Y} \longrightarrow \{\top, \bot\}$ where $\top$ indicates the object being assigned the class label, and $\bot$ indicates the contrary. We consider only binary problems, where $\mathcal{Y} = \{C, \overline{C}\}$ for some class C and its complement $\overline{C}$. Learning a classification then consists in finding a function f that approximates a true classification $\widehat{f}$ with low approximation error, given a set of correctly labelled training examples $(x, y) \in \mathcal{X} \times \mathcal{Y}$.

Provided that the objects $x \in \mathcal{X}$ are represented in a suitable manner, usually as numerical *feature vectors* $\mathbf{x}$ in an m-dimensional Euclidean feature space $\mathbb{R}^m$, one of several learning algorithms can be employed to learn a classification. Support vector machines constitute a class of algorithms based on the idea of computing a decision hyperplane $\mathbf{w}^T\phi(\mathbf{x})+b = 0$ that maximizes the margin between positive and negative training instances in the feature space (Vapnik, 1995). Such maximum-margin hyperplanes tend to entail lower generalization errors than other separation surfaces, and the task of finding them leads to a quadratic optimization problem. Additional slack variables can be included for a soft margin solution that is able to deal with training data that cannot be separated cleanly (Cortes and Vapnik, 1995). The decision surface can then be computed using Lagrange multipliers and decomposing techniques such as sequential minimal optimization (Platt, 1999).

3 Building Wordnets by Learning Classifications

In order to build wordnets automatically, we suggest the following approach. Let L_{N} denote the language for which a wordnet is to be constructed, and L_0 denote the language of an existing wordnet that serves as a template for the new one, in our case the English language due to our choice of Princeton WordNet as the template. The existing wordnet immediately provides a sense inventory as well as information about the links between the senses, though certain relations need to be interpreted as generic relatedness links between senses (e.g. the derivation relation), or are completely excluded from being adopted (e.g. region domains). The most significant missing ingredient at this point are the links from terms in L_{N} to their respective senses. This is tackled by means of translation dictionaries, however with the constraint of relying on a minimal amount of information specific to L_{N} so that the approach remains generalizable to as many languages as possible. The dictionary is thus conceived as offering a simple $n : m$-mapping between terms in L_0 and terms in L_{N}, with optional part of speech information, as in the following excerpt:

```
{n}   Schulabbrecher   -  dropout
      ...                 ...
{n}   Schulklasse      -  class
{n}   Schulklasse      -  form
      ...                 ...
      schulmäßig       -  scholastic
{adv} schulmäßig       -  scholastically
```

Given a translation from a term t from L_{N} to a term e from L_0, one may assume that there is very likely some semantic overlap between t and e, so some sense of e is likely to also be a sense of t. We thus proceed as follows: for each term t from L_{N}, retrieve the set of translations $\phi(t)$. For each L_0-translation e in such a $\phi(t)$, retrieve the set of senses $\sigma(e)$ from our existing wordnet, e.g. for the German term "`Schulklasse`" the senses of the translations "`class`" and "`form`" would be considered. Our goal is now to determine for each sense $s \in \bigcup_{e \in \phi(t)} \sigma(e)$ whether s is also an appropriate sense of t. This is undoubtedly a very difficult task, as the dictionaries provide only limited information that could aid in determining which of the often many different senses listed by WordNet apply, e.g. 9 senses for the word "`class`" and 23 senses for "`form`". In our approach, the problem is construed as a binary classification problem. A real-valued feature vector $\mathbf{x}$ is created for each pair (t, s) of a term t from L_{N} and a relevant candidate sense s from the wordnet for L_0. For example, if t represents "`Schulklasse`", then s could be one of the senses of "`class`". In order to create the feature vectors, a variety of different fitness scores x_i are used as features and combined as components of numeric vectors $\mathbf{x} = (x_1, \ldots, x_m) \in \mathcal{X} = \mathbb{R}^m$.

Based on a small set of manually established labels for such (t, s)-pairs, we create the corresponding training set of feature vectors and derive a classification model that can be used to make predictions for any other (t, s)-pair. Assuming that the model provides confidence scores $c_{t,s}$ for (t, s)-pairs, we apply one of the following rules for every L_{N}-term t from the translation dictionary, combined with each of its possible candidate senses as defined above:

a) accept as a weighted connection with weight $c_{t,s}$ whenever $c_{t,s} > 0$, or
b) accept as an unweighted connection whenever $c_{t,s} \geq \alpha_1$ or $\forall s' \neq s : c_{t,s} > c_{t,s'}$ (for two pre-defined constants α_1 and $\alpha_2 \leq \alpha_1$).

The first rule results in a weighted statistical wordnet for L_{N}, whereas the second one yields a conventional unweighted wordnet. Finally, new senses may be introduced manually to cover terms for which no candidate senses were found.

This approach has several advantages compared to the previous work in this field (cf. Section 6). First of all, the previous automatic approaches were based on hard acceptance criteria - either a (t, s)-pair satisfies a criterion or not. Many attributes of word senses do not lend themselves easily to such an antagonistic view, e.g. sense relatedness measures produce numeric scores, and thus can be better accommodated in a model that uses real-valued feature vectors. Furthermore, while Atserias et al. (1997) investigate combinations of two heuristics to arrive at a greater accuracy, a classification learning approach can take into account suitable combinations of even more heuristics, indeed arbitrary linear (or even non-linear) combinations of feature values.

4 Feature Computation

Following the description of the overall procedure, we will now go into more detail on how the feature values x_i are computed before being combined to create the feature vector for a given (t, s)-pair.

4.1 Lexical Category Compatibility

Unlike previous work, our study considers all lexical categories (parts of speech) covered by the existing wordnet rather than just nouns. This immediately leads to the problem that the number of candidate senses greatly increases, and we need to come up with some means of preventing a noun from being mapped to a verb sense in WordNet, for instance.

Our solution rests on two pillars. Obviously, whenever the translation dictionary explicitly provides lexical category information, one can simply use hard-coded compatibility indicators, e.g. we give any German adjective a compatibility value of 0.0 with English noun senses, but 1.0 with English adjective as well as adverb senses.

In light of the fact that such information may not always be available, we resort to additional heuristics when such explicit information is not available, thereby ensuring that our approach remains applicable to a broad range of different scenarios. For each lexical category, a C4.5 decision tree is used to estimate the compatibility based on superficial attributes of the terms such as suffixes and capitalization. Growing the trees does not require any manually created training data, because we can leverage terms where all candidate senses share the same lexical category. The features employed are given in the following list. Note that since the terms in L_{N} can be multi-word expressions, much of this information is captured separately for the first and last word of any candidate expression.

- prefixes of the first and last word up to a length of 10, e.g. for the German verb "`einschulen`", "`e`", "`ei`", "`ein`", etc. would be considered
- suffixes of the first and last word up to a length of 10 (without case conversion), e.g. "`n`", "`en`", "`len`", etc. for "`einschulen`".

- capitalization of the first and last word (Boolean features for no capitalization, capitalized first character, and all characters capitalized)
- term length

The decision trees were pruned to have confidence levels of at least 0.25 with at least 2 instances per leaf. The confidence estimations from the leaves can then be used to determine a lexical category compatibility score as a feature in the feature vector. For languages where the predictions are too unreliable, we may instead use a constant value of 0.5.

4.2 Sense Weighting Functions

Several features that will be described later on depend on some kind of assessment of the importance of senses s with respect to the particular L_{N}-term t under consideration. We consider the following weighting functions $\gamma(t, s)$:

- $\gamma_1(t, s) = 1$ is used for unweighted features
- $\gamma_{\mathrm{lc}}(t, s)$ represents an estimation of the lexical category compatibility between t and s, as described earlier
- $\gamma_{\mathrm{r}}(t, s)$ considers the ranks of the senses as listed by WordNet for the translations of t, as these are indicators for the importance of a sense. It is computed as follows:

$$\gamma_{\mathrm{r}}(t, s) = \gamma_{\mathrm{lc}}(t, s) \left[\sum_{e \in \phi(t)} \frac{1}{r(e, s)} \right]$$

  where $r(e, s)$ yields 1 if s is the highest-ranked sense for e, 2 for the second sense, and so on.
- $\gamma_{\mathrm{f}}(t, s)$ considers the corpus frequency information provided with WordNet:

$$\gamma_{\mathrm{f}}(t, s) = \gamma_{\mathrm{lc}}(t, s) \left[\sum_{e \in \phi(t)} \frac{f(e, s)}{\sum\limits_{s' \in \sigma(e)} \lambda_{s,s'} f(e, s')} \right]$$

  where $f(e, s)$ returns the number of occurrences of term e with sense s in the corpus, and $\lambda_{s,s'}$ is 1 if the lexical category of s and s' match, and 0 otherwise.

4.3 Semantic Relatedness Measures

Apart from weighting functions, our approach is fundamentally based on measures of semantic relatedness between senses, e.g. the single sense of "`schoolhouse`" is related to the educational institution sense of "`school`", but not to the sense of "`school`" that refers to groups of fish. Before going into details of how semantic relatedness contributes to many of our fitness scores, we shall first introduce several relatedness estimation heuristics.

- $\mathrm{sim}_{\mathrm{id}}(s_1, s_2)$ is simply the trivial identity indicator function, i.e. yields 1 if $s_1 = s_2$, and 0 otherwise.

$$\mathrm{sim}_{\mathrm{id}}(s_1, s_2) = \begin{cases} 1 & s_1 = s_2 \\ 0 & \text{otherwise} \end{cases}$$

- $\mathrm{sim}_{\mathrm{f}}(s_1, s_2)$ considers not only whether two senses are identical but also takes into account senses that stand in a parent-child or sibling relationship in terms of the hypernym hierarchy.

$$\mathrm{sim}_{\mathrm{f}}(s_1, s_2) = \begin{cases} 1 & s_1 = s_2 \\ 0.8 & \text{hypernymy/hyponymy} \\ 0.7 & \text{siblings, no hypernymy} \\ 0 & \text{otherwise} \end{cases}$$

- $\mathrm{sim}_{\mathrm{n}}(s_1, s_2)$ considers the graph neighbourhood and acknowledges relations other than hypernymy/hyponymy as well as transitive connections (e.g. a holonym of a hypernym). For a given path in the graph, we may compute an inverse distance score multiplicatively from relation-specific edge weights (e.g. 0.8 for hypernymy, 0.7 for holonymy). The relatedness score is then defined as the maximum score for all paths between s_1 and s_2 if this maximum is above or equal a pre-defined threshold $\alpha_n = 0.35$, and 0 otherwise. It can be obtained efficiently using a Dijkstra-like algorithm (de Melo and Siersdorfer, 2007).
- $\mathrm{sim}_{\mathrm{c}}(s_1, s_2)$ uses the cosine similarity of context strings for senses, which are constructed by concatenating glosses and lexicalizations of the sense itself with those of senses directly related via hyponymy, holonymy, derivation, or instance relations, as well as with those of 2

levels of hypernyms. The terms are stemmed using Porter's stemmer, and feature vectors $\mathbf{v}_1$, $\mathbf{v}_2$ with TF-IDF values are created based on the bag-of-words vector space model. The score is then computed as the cosine of the angle between the vectors, i.e. as $\mathbf{v}_1^T\mathbf{v}_2(||\mathbf{v}_1||\,||\mathbf{v}_2||)^{-1}$.

- $\mathrm{sim_m}(s_1, s_2)$, finally, is simply defined as $\max\{\mathrm{sim_f}(s_1, s_2), \mathrm{sim_n}(s_1, s_2), \mathrm{sim_c}(s_1, s_2)\}$, and hence combines the power of $\mathrm{sim_f}$, $\mathrm{sim_n}$, and $\mathrm{sim_c}$, which is particularly valuable due to the fact that $\mathrm{sim_n}$ and $\mathrm{sim_c}$ are based on very different characteristics of the senses.

4.4 Semantic Overlap Features

One important way of making use of the semantic relatedness measures is to exploit that a mapping should more likely be accepted when a term t has multiple English translations e, and the candidate sense s under consideration is somewhat pertinent to multiple of them. For instance, the German "`Schulklasse`" has the terms "`class`" and "`form`" as translations. While "`form`" can not only refer to a body of students who are taught together but also e.g. to a tax form, only the former of these two senses overlaps semantically with the senses of "`class`".

Given a term t and a candidate sense s, we integrate scores of the following form into the respective feature vector:

$$\sum_{e\in\phi(t)} \max_{s'\in\sigma(e)} \gamma(t, s')\,\mathrm{sim}(s, s') \tag{1}$$

$$\sum_{e\in\phi(t)} \frac{\sum\limits_{s'\in\sigma(e)} \gamma(t, s')\,\mathrm{sim}(s, s')}{\sum\limits_{s'\in\sigma(e)} \gamma(t, s')} \tag{2}$$

where $\mathrm{sim}(s_1, s_2)$ represents a semantic relatedness measure and the $\gamma(t, s)$ function provides weights as described earlier. The simple identity relatedness function $\mathrm{sim_{id}}$ and the constant weighting function $\gamma_1(t, s) = 1$ make Equation 1 yield a simple count of how many English terms are mapped to the sense, reminiscent e.g. of the equivalent word matching of Okumura and Hovy (1994) (cf. Section 6). By using the above formulae to produce a large number of feature values with all combinations of weighting functions and relatedness measures mentioned in Sections 4.2 and 4.3, we are able to account for cases where the terms are related but do not share senses.

4.5 Polysemy-Based Scores

Another set of features are based on the polysemy of the L_0-translations, i.e. on the idea that a mapping is more likely correct whenever there are few alternative senses to choose from. Akin to the monosemy heuristic of Okumura et al. (see Section 6), we can consider for instance the German "`Schulleiter`" with its translation "`headmaster`", which in turn only has one single sense listed in WordNet, so it is rather safe to accept this sense also for the German term. More generally, given a term t and a sense s, several scores can be computed as

$$\left(1 + \sum_{s'\in C} \gamma(t, s')(1 - \mathrm{sim}(s, s'))\right)^{-1} \tag{3}$$

where $\gamma(t, s)$ is a weighting function and $C = \bigcup\limits_{e\in\phi(t)} \sigma(e)$ stands for the complete candidate set.
Another set of scores is computed as

$$\sum_{e\in\phi(t)} \frac{\mathbf{1}_{\sigma(e)}(s)}{1 + \sum\limits_{s'\in\sigma(e)} \gamma(t, s')(1 - \mathrm{sim}(s, s'))} \tag{4}$$

where $\mathbf{1}_{\sigma(e)}(s)$ is the indicator function for $\sigma(e)$, and therefore yields 1 if $s \in \sigma(e)$ and 0 otherwise.

Again, we can use $\mathrm{sim_{id}}(s_1, s_2)$ and $\gamma_1(t, s)$ to illustrate the simplest case: Equation 3 then yields the reciprocal of the total number of candidate senses and in Equation 4 the denominator of each addend becomes 1 whenever the respective term e is monosemous according to WordNet. More advanced scores are computed by

- using Equations 3, 4 with $\gamma_1(t, s)$, combined with either $\mathrm{sim_f}$, $\mathrm{sim_c}$, $\mathrm{sim_n}$, or $\mathrm{sim_m}$, and
- using Equation 4 with $\mathrm{sim_{id}}(s_1, s_2)$ and one of the weighting functions $\gamma_{\mathrm{lc}}(t, s)$, $\gamma_{\mathrm{r}}(t, s)$, or $\gamma_{\mathrm{f}}(t, s)$.

4.6 Additional Features

We further consider a number of other, less essential features, including the following:

- scores based on the number of translations

$$\left(\sum_{e\in\phi(t)} \lambda(t,e)\right)^{-1}$$

as well as the ratio

$$\frac{\sum_{e\in\phi(t)} \lambda_{\mathrm{wn}}(t,e)}{\sum_{e\in\phi(t)} \lambda_{\mathrm{id}}(t,e)} = \frac{\sum_{e\in\phi(t)} \lambda_{\mathrm{wn}}(t,e)}{|\phi(t)|}$$

where $\lambda(t,e)$ is a translation weighting function that can be either $\lambda_{\mathrm{id}}(t,e) = 1$ or $\lambda_{\mathrm{wn}}(t,e)$, which is 1 if $\sigma(e) \neq \emptyset$, and 0 otherwise.
- a score based on back-translations

$$\sum_{e\in\phi(t)} \frac{\mathbf{1}_{\sigma(e)}(s)}{|\phi^{-1}(e)|}$$

where $\phi^{-1}(e)$ is defined as $\{t \mid e \in \phi(t)\}$.
- the number of lexicalizations of the candidate sense, i.e. $\left|\sigma^{-1}(s)\right|$, where $\sigma^{-1}(s)$ is defined as $\{e \mid s \in \sigma(e)\}$.
- the ratio of sense lexicalizations that are translations of t, i.e.

$$\frac{\sum_{e\in\sigma^{-1}(s)} \lambda_{\mathrm{tr}}(t,e)}{|\sigma^{-1}(s)|}$$

where $\sigma^{-1}(s)$ is defined as above, and $\lambda_{\mathrm{tr}}(t,e)$ yields 1 if $e \in \phi(t)$ and 0 otherwise.
- indicator values that express whether the candidate sense s is a noun, verb, adjective, or adverb sense, respectively.

5 Experimental Evaluation

While our approach is applicable to virtually any language, we evaluated it by generating a German wordnet based on the Ding German-English dictionary (Richter, 2007), a large and fairly reliable digital translation dictionary with around 216,000 entries, but not much additional information apart from (optional) part of speech tags. Princeton WordNet 3.0, which covers around 155,000 English terms and around 118,000 senses, served as the existing template for the new wordnet.

Table 1: Comparison with existing methods

	precision	recall
First Sense Heuristic	40.36%	67.46%
Rigau & Agirre	48.97%	63.58%
Atserias et al.[1]	75.00%	35.82%
Benítez et al.	73.14%	38.21%
Our Approach	81.11%	65.37%

[1]: excluding criteria based on additional background knowledge (see text)

We manually evaluated 1,834 candidate mappings for 350 randomly selected German terms from the dictionary for use as training data (407 mappings, i.e. 22%, were positive). To create a test set with both positive and negative examples, the same was repeated with another 1,624 candidate mappings for 350 further randomly selected terms. Based on this training data, the LIBSVM implementation (Chang and Lin, 2001) of support vector machine learning was used to derive a linear kernel model and additionally also estimate posterior class probabilities for the (t,s)-pairs using a variant of Platt's method (Lin et al., 2007). The thresholds $\alpha_1 = 0.5$ and $\alpha_2 = 0.45$ were applied on these estimates as described in Section 3 to generate a German wordnet.

Our technique is compared to four alternative approaches. We study the first sense heuristic, which involves simply accepting the first sense listed by WordNet for any English term, and is frequently cited as more successful than many other heuristics in word sense disambiguation tasks because the rank reflects the corpus frequency and importance of a sense. We also evaluate existing automatic approaches presented in Section 6. From the study by Atserias et al. (1997), we consider the monosemy 1-4, variant, as well as the combined brother and polysemy 1/2 criteria. The CD criteria and the field criterion were not applied because their implementation in the original study is mainly based on additional lexical information for the Spanish language apart from the list of translations. The standard classification evaluation measures of precision and recall were used. Given a test set, the precision is defined as $\frac{P_T}{P_T+P_F}$, and the recall is defined as $\frac{P_T}{P_T+N_F}$, where P_T, P_F, N_F are the sets of true positives, false positives, and false negatives, respectively. The results,

Table 2: Alternative confidence thresholds

α_1	α_2	precision	recall
0.90	0.80	94.21%	34.03%
0.90	0.60	91.50%	41.79%
0.70	0.60	87.50%	52.24%
0.60	0.50	83.90%	59.10%
0.50	0.45	81.11%	65.37%
0.40	0.35	73.64%	72.54%
0.35	0.25	70.53%	80.00%
0.30	0.25	67.32%	82.39%
0.20	0.15	55.93%	90.15%
0.10	0.05	40.41%	94.93%

Table 3: Coverage Statistics

	sense mappings	terms	lexicalized senses
nouns	53,146	35,089	28,007
verbs	13,875	5,908	6,304
adjectives	21,799	13,772	9,949
adverbs	4,243	2,992	2,593
total	93,063	55,522	46,853

presented in Table 1, demonstrate that our learning-based approach outperforms the existing approaches both in terms of precision as well as in terms of recall. While two previous heuristics arrive at similarly high levels of recall, this occurs at the expense of very low precision scores. By adjusting the α_1, α_2 confidence thresholds, our method can be made to produce recall scores well above 90% at such levels of precision. Table 2 provides a sample of results obtained using alternative thresholds.

In addition to the recall scores in Table 1, which are based on the test set, we also provide the absolute number of terms covered by the resulting German wordnet in Table 3. The figures are below the current size of GermaNet 5.0 (Kunze and Lemnitzer, 2002), but larger by an order of magnitude than many other manually compiled wordnets.

6 Related Work

Although no other studies have considered building new wordnets by classifying real-valued feature vectors, there has been prior work on heuristics for linking dictionaries to WordNet. Knight (1993) created an ontology for machine translation by linking entries in Longman's Dictionary of Contemporary English to WordNet, taking into account gloss definitions as well as the semantic hierarchy information present in the dictionary, though unfortunately not available in our setting. Okumura and Hovy (1994) used a Japanese-English dictionary to link a Japanese lexicon to this ontology, based on several heuristics, most importantly monosemy, i.e. considering when the ontology lists only one candidate concept for an English translation, and equivalent word matches, i.e. accepting the concepts shared by multiple translations of a word.

Rigau and Agirre (1995) presented a preliminary study on mapping Spanish nouns to WordNet senses by looking up the translations of the Spanish noun, and then checking whether the senses of those translations satisfy certain criteria. Atserias et al. (1997) proposed additional heuristics for generating a preliminary noun-only version of the Spanish WordNet that later were adapted for producing preliminary noun-only Catalan and Hungarian wordnets (Benitez et al., 1998; Miháltz and Prószéky, 2004).

Pianta et al. (2002) used similar techniques in conjunction with a cosine similarity-based heuristic to create rankings of the most likely candidate senses that were then presented to human lexicographers for selection. This methodology was used to create MultiWordNet Italian and later also the Hebrew WordNet. Sathapornrungkij and Pluempitiwiriyawej (2005) used criteria proposed by Atserias et al. (1997), and then performed a regression analysis in order to reduce the number of accepted mappings and thus increase the accuracy. Since they merely relied on 12 binary criteria rather than numeric scores, they were unable to obtain a higher recall by applying their model to other term-sense pairs not fulfilling one of the chosen criteria.

7 Conclusions

We have shown that wordnets can be built automatically if we are willing to accept a certain percentage of imprecise mappings. Our approach based on learning from a number of numeric scores leads to a better coverage than the hard criteria proposed in previous studies, while simultaneously also allowing for a higher level of accuracy. It is fair to assume that the method presented scales well to new languages, because care was taken to require just a minimal amount of information specific to L_{N}. The resulting resources greatly facilitate interoperability, as they are aligned to the original Princeton WordNet, and thus also to other resources that are similarly aligned.

In the future we would like to investigate techniques for extending the coverage of such automatically generated wordnets to senses not covered by the existing wordnet. It is well-known that for a variety of tasks one can benefit from the information stored in lexical resources, e.g. for word sense disambiguation, for query expansion in information retrieval, especially in image and multimedia retrieval, and for cross-lingual applications. We will soon provide a more detailed analysis of the quality of automatically generated wordnets, also studying in detail their suitability for use in monolingual as well as cross-lingual natural language processing tasks.

References

Jordi Atserias, Salvador Climent, Xavier Farreres, German Rigau, and Horacio Rodríguez. 1997. Combining multiple methods for the automatic construction of multilingual WordNets. In *Proc. Intl. Conf. on Recent Advances in NLP 1997*, pages 143–149.

Laura Benitez, Sergi Cervell, Gerard Escudero, Monica Lopez, German Rigau, and Mariona Taulé. 1998. Methods and tools for building the Catalan WordNet. In *Proc. ELRA Workshop on Language Res. for Europ. Minority Lang., 1st Intl. Conf. on Language Resources & Evaluation*.

Luisa Bentivogli, Pamela Forner, Bernardo Magnini, and Emanuele Pianta. 2004. Revising the WordNet domains hierarchy. In *COLING 2004 Multiling. Ling. Resources*, pages 94–101, Geneva, Switzerland.

Chih-Chung Chang and Chih-Jen Lin, 2001. *LIBSVM: a library for support vector machines*.

Corinna Cortes and Vladimir Vapnik. 1995. Support-vector networks. *Machine Learning*, 20(3):273–297.

Gerard de Melo and Stefan Siersdorfer. 2007. Multilingual text classification using ontologies. In Gianni Amati, editor, *Proc. 29th European Conference on Information Retrieval (ECIR 2007)*, volume 4425 of *Lecture Notes in Computer Science*, Rome, Italy. Springer.

Christiane Fellbaum, editor. 1998. *WordNet: An Electronic Lexical Database (Language, Speech, and Communication)*. The MIT Press.

Kevin Knight. 1993. Building a large ontology for machine translation. In *Proc. Workshop Human Language Technology*, pages 185–190.

Claudia Kunze and Lothar Lemnitzer. 2002. GermaNet - representation, visualization, application. In *Proc. LREC 2002*, pages 1485–1491.

Hsuan-Tien Lin, Chih-Jen Lin, and Ruby C. Weng. 2007. A note on platt's probabilistic outputs for support vector machines. *Machine Learning*, 68(3):267–276.

Márton Miháltz and Gábor Prószéky. 2004. Results and evaluation of Hungarian Nominal WordNet v1.0. In *Proceedings of the Second Global WordNet Conference*, Brno, Czech Republic. Masaryk University.

Ian Niles and Adam Pease. 2003. Linking lexicons and ontologies: Mapping WordNet to the suggested upper merged ontology. In *Proc. 2003 Intl. Conf. Information and Knowledge Engineering, Las Vegas, NV, USA*.

Akitoshi Okumura and Eduard Hovy. 1994. Building Japanese-English dictionary based on ontology for machine translation. In *Proc. Workshop on Human Language Technology*, pages 141–146.

Emanuele Pianta, Luisa Bentivogli, and Christian Girardi. 2002. MultiWordNet: Developing an aligned multilingual database. In *Proc. 1st Intl. Global WordNet Conference, Mysore, India*, pages 293–302.

John C. Platt, 1999. *Fast training of support vector machines using sequential minimal optimization*, pages 185–208. MIT Press, Cambridge, MA, USA.

Frank Richter, 2007. *Ding Version 1.5*. `http://www-user.tu-chemnitz.de/~fri/ding/`.

German Rigau and Eneko Agirre. 1995. Disambiguating bilingual nominal entries against WordNet. In *Proc. Workshop 'The Computational Lexicon' at European Summer School Logic, Language & Information*.

Patanakul Sathapornrungkij and Charnyote Pluempitiwiriyawej. 2005. Construction of Thai WordNet lexical database from machine readable dictionaries. In *Proc. 10th Machine Translation Summit, Phuket, Thailand*.

Fabian M. Suchanek, Gjergji Kasneci, and Gerhard Weikum. 2007. Yago: A Core of Semantic Knowledge. In *16th Intl. World Wide Web Conference (WWW 2007)*. ACM Press.

Vladimir N. Vapnik. 1995. *The Nature of Statistical Learning Theory*. Springer-Verlag New York, Inc., New York, NY, USA.

Piek Vossen, editor. 1998. *EuroWordNet: A Multilingual Database with Lexical Semantic Networks*. Springer.