Accessing GermaNet Data and Computing Semantic Relatedness

Iryna Gurevych and Hendrik Niederlich
EML Research gGmbH
Schloss-Wolfsbrunnenweg 33
69118 Heidelberg, Germany
http://www.eml-research.de/~gurevych

Abstract

We present an API developed to access GermaNet, a lexical semantic database for German represented in XML. The API provides a set of software functions for parsing and retrieving information from GermaNet. Then, we present a case study which builds upon the GermaNet API and implements an application for computing semantic relatedness according to five different metrics. The package can, again, serve as a software library to be deployed in natural language processing applications. A graphical user interface allows to interactively experiment with the system.

1 Motivation

The knowledge encoded in WordNet (Fellbaum, 1998) has proved valuable in many natural language processing (NLP) applications. One particular way to integrate semantic knowledge into applications is to compute semantic similarity of WordNet concepts. This can be used e.g. to perform word sense disambiguation (Patwardhan et al., 2003), to find predominant word senses in untagged text (McCarthy et al., 2004), to automatically generate spoken dialogue summaries (Gurevych & Strube, 2004), and to perform spelling correction (Hirst & Budanitsky, 2005).

Extensive research concerning the integration of semantic knowledge into NLP for the English language has been arguably fostered by the emergence of WordNet::Similarity package (Pedersen et al., 2004).[1] In its turn, the development of the WordNet based semantic similarity software has been facilitated by the availability of tools to easily retrieve data from WordNet, e.g. WordNet::QueryData,[2] jwnl.[3]

Research integrating semantic knowledge into NLP for languages other than English is scarce. On the one hand, there are fewer computational knowledge resources like dictionaries, broad enough in coverage to be integrated in robust NLP applications. On the other hand, there is little off-the-shelf software that allows to develop applications utilizing semantic knowledge from scratch. While WordNet counterparts do exist for many languages, e.g. GermaNet (Kunze & Lemnitzer, 2002) and EuroWordNet (Vossen, 1999), they differ from WordNet in certain design aspects. E.g. GermaNet features non-lexicalized, so called *artificial* concepts that are non-existent in WordNet. Also, the adjectives are structured hierarchically which is not the case in WordNet. These and other structural differences led to divergences in the data model. Therefore, WordNet based implementations are not applicable to GermaNet. Also, there is generally lack of experimental evidence concerning the portability of e.g. WordNet based semantic similarity metrics to other wordnets and their sensitivity to specific factors, such as network structure, language, etc. Thus, for a researcher who wants to build a semantic relatedness application for a language other than English, it is difficult to assess the effort and challenges involved in that.

Departing from that, we present an API which allows to parse and retrieve data from GermaNet. Though it was developed following the guidelines for creating WordNet, GermaNet features a couple of divergent design decisions, such as e.g. the use of non-lexicalized concepts, the association relation between synsets and the small number of textual definitions of word senses. Furthermore, we

[1] http://www.d.umn.edu/~tpederse/similarity.html

[2] http://search.cpan.org/dist/WordNet-QueryData

[3] http://sourceforge.net/projects/jwordnet

Proceedings of the ACL Interactive Poster and Demonstration Sessions,
pages 5–8, Ann Arbor, June 2005.

build an application accessing the knowledge in GermaNet and computing semantic relatedness of GermaNet word senses according to five different metrics. Three of these metrics have been adapted from experiments on English with WordNet, while the remaining two are based on automatically generated definitions of word senses and were developed in the context of work with GermaNet.

2 GermaNet API

The API for accessing GermaNet has to provide functions similar to the API developed for WordNet. We evaluated the C-library distributed together with GermaNet V4.0 and the XML encoded version of GermaNet (Lemnitzer & Kunze, 2002). As we wanted the code to be portable across platforms, we built upon the latter. The XML version of GermaNet is parsed with the help of the Apache Xerces parser, http://xml.apache.org/ to create a JAVA object representing GermaNet. For stemming the words, we use the functionality provided by the Porter stemmer for the German language, freely available from http://snowball.tartarus.org/german/stemmer.html. Thus, the GermaNet object exists in two versions, the original one, where the information can be accessed using words, and the stemmed one, where the information can be accessed using word stems.

We implemented a range of JAVA based methods for querying the data. These methods are organized around the notions of word sense and synset. On the word sense (WS) level, we have the following methods: *getAntonyms()* retrieves all antonyms of a given WS; *getArtificial()* indicates whether a WS is an artificial concept; *getGrapheme()* gets a graphemic representation of a WS; *getParticipleOf()* retrieves the WS of the verb that the word sense is a participle of; *getPartOfSpeech()* gets the part of speech associated with a WS; *getPertonym()* gives the WS that the word sense is derived from; *getProperName()* indicates whether the WS is a proper name; *getSense()* yields the sense number of a WS in GermaNet; *getStyle()* indicates if the WS is stylistically marked; *getSynset()* returns the corresponding synset; *toString()* yields a string representing a WS.

On the synset level, the following information can be accessed: *getAssociations()* returns all associations; *getCausations()* gets the effects that a given synset is a cause of; *getEntailments()* yields synsets that entail a given synset; *getHolonyms()*, *getHyponyms()*, *getHypernyms()*, *getMeronyms()* return a list of holonyms, hyponyms, immediate hypernyms, and meronyms respectively; *getPartOfSpeech()* returns the part of speech associated with word senses of a synset; *getWordSenses()* returns all word senses constituting the synset; *toString()* yields a string representation of a synset.

The metrics of semantic relatedness are designed to employ this API. They are implemented as classes which use the API methods on an instance of the GermaNet object.

3 Semantic Relatedness Software

In GermaNet, nouns, verbs and adjectives are structured within hierarchies of *is-a* relations.[4] GermaNet also contains information on additional lexical and semantic relations, e.g. hypernymy, meronymy, antonymy, etc. (Kunze & Lemnitzer, 2002). A semantic relatedness metric specifies to what degree the meanings of two words are related to each other. E.g. the meanings of *Glas* (Engl. *glass*) and *Becher* (Engl. *cup*) will be typically classified as being closely related to each other, while the relation between *Glas* and *Juwel* (Engl. *gem*) is more distant. *RelatednessComparator* is a class which takes two words as input and returns a numeric value indicating semantic relatedness for the two words. Semantic relatedness metrics have been implemented as descendants of this class.

Three of the metrics for computing semantic relatedness are information content based (Resnik, 1995; Jiang & Conrath, 1997; Lin, 1998) and are also implemented in WordNet::Similarity package. However, some aspects in the normalization of their results and the task definition according to which the evaluation is conducted have been changed (Gurevych & Niederlich, 2005). The metrics are implemented as classes derived from *InformationBasedComparator*, which is in its turn derived from the class *PathBasedComparator*. They make use of both the GermaNet hierarchy and statistical corpus evidence, i.e. information content.

[4] As mentioned before, GermaNet abandoned the cluster-approach taken in WordNet to group adjectives. Instead a hierarchical structuring based on the work by Hundsnurscher & Splett (1982) applies, as is the case with nouns and verbs.

We implemented a set of utilities for computing information content of German word senses from German corpora according to the method by Resnik (1995). The TreeTagger (Schmid, 1997) is employed to compile a part-of-speech tagged word frequency list. The information content values of GermaNet synsets are saved in a text file called an information content map. We experimented with different configurations of the system, one of which involved stemming of corpora and the other did not involve any morphological processing. Contrary to our intuition, there was almost no difference in the information content maps arising from the both system configurations, with and without morphological processing. Therefore, the use of stemming in computing information content of German synsets seems to be unjustified.

The remaining two metrics of semantic relatedness are based on the Lesk algorithm (Lesk, 1986). The Lesk algorithm computes the number of overlaps in the definitions of words, which are sometimes extended with the definitions of words related to the given word senses (Patwardhan et al., 2003). This algorithm for computing semantic relatedness is very attractive. It is conceptually simple and does not require an additional effort of corpus analysis compared with information content based metrics.

However, a straightforward adaptation of the Lesk metric to GermaNet turned out to be impossible. Textual definitions of word senses in GermaNet are fairly short and small in number. In cotrast to WordNet, GermaNet cannot be employed as a machine-readable dictionary, but is primarily a conceptual network. In order to deal with this, we developed a novel methodology which generates definitions of word senses automatically from GermaNet using the GermaNet API. Examples of such automatically generated definitions can be found in Gurevych & Niederlich (2005). The method is implemented in the class *PseudoGlossGenerator* of our software, which automatically generates glosses on the basis of the conceptual hierarchy.

Two metrics of semantic relatedness are, then, based on the application of the Lesk algorithm to definitions, generated automatically according to two system configurations. The generated definitions can be tailored to the task at hand according to a set of parameters defining which related concepts

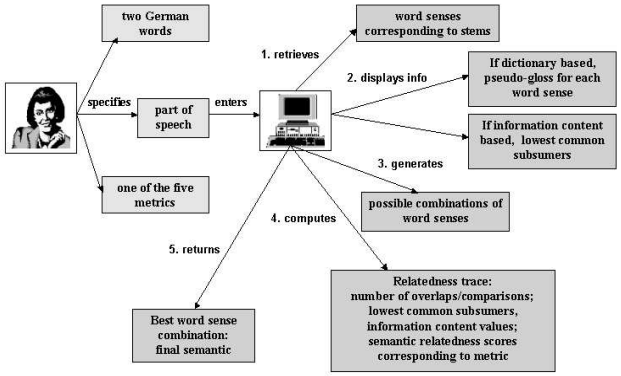

Figure 1: The concept of user-system interaction.

have to be included in the final definition. Experiments carried out to determine the most effective parameters for generating the definitions and employing those to compute semantic relatedness is described in Gurevych (2005). Gurevych & Niederlich (2005) present a description of the evaluation procedure for five implemented semantic relatedness metrics against a human *Gold Standard* and the evaluation results.

4 Graphical User Interface

We developed a graphical user interface to interactively experiment with the software for computing semantic relatedness. The system runs on a standard Linux or Windows machine. Upon initialization, we configured the system to load an information content map computed from the German *taz* corpus.[5] The information content values encoded therein are employed by the information content based metrics. For the Lesk based metrics, two best configurations for generating definitions of word senses are offered via the GUI: one including three hypernyms of a word sense, and the other one including all related synsets (two iterations) except hyponyms. The representation of synsets in a generated definition is constituted by one (the first) of their word senses.

The user of the GUI can enter two words together with their part-of-speech and specify one of the five metrics. Then, the system displays the corresponding word stems, possible word senses ac-

[5] www.taz.de

cording to GermaNet, definitions generated for these word senses and their information content values. Furthermore, possible combinations of word senses for the two words are created and returned together with various diagnostic information specific to each of the metrics. This may be e.g. word overlaps in definitions for the Lesk based metrics, or lowest common subsumers and their respective information content values, depending on what is appropriate. Finally, the best word sense combination for the two words is determined and this is compactly displayed together with a semantic relatedness score. The interface allows the user to add notes to the results by directly editing the data shown in the GUI and save the detailed analysis in a text file for off-line inspection. The process of user-system interaction is summarized in Figure 1.

5 Conclusions

We presented software implementing an API to GermaNet and a case study built with this API, a package to compute five semantic relatedness metrics. We revised the metrics and in some cases redesigned them for the German language and GermaNet, as the latter is different from WordNet in a number of respects. The set of software functions resulting from our work is implemented in a JAVA library and can be used to build NLP applications with GermaNet or integrate GermaNet based semantic relatedness metrics into NLP systems. Also, we provide a graphical user interface which allows to interactively experiment with the system and study the performance of different metrics.

Acknowledgments

This work has been funded by the Klaus Tschira Foundation. We thank Michael Strube for his valuable comments concerning this work.

References

Fellbaum, Christiane (Ed.) (1998). *WordNet: An Electronic Lexical Database*. Cambridge, Mass.: MIT Press.

Gurevych, Iryna (2005). *Using the Structure of a Conceptual Network in Computing Semantic Relatedness*. Submitted.

Gurevych, Iryna & Hendrik Niederlich (2005). Computing semantic relatedness of GermaNet concepts. In Bernhard Fisseni, Hans-Christian Schmitz, Bernhard Schröder & Petra Wagner (Eds.), *Sprachtechnologie, mobile Kommunikation und linguistische Ressourcen: Proceedings of Workshop "Applications of GermaNet II" at GLDV'2005*, pp. 462–474. Peter Lang.

Gurevych, Iryna & Michael Strube (2004). Semantic similarity applied to spoken dialogue summarization. In *Proceedings of the 20th International Conference on Computational Linguistics*, Geneva, Switzerland, 23 – 27 August 2004, pp. 764–770.

Hirst, Graeme & Alexander Budanitsky (2005). Correcting real-word spelling errors by restoring lexical cohesion. *Natural Language Engineering*, 11(1):87–111.

Hundsnurscher, F. & J. Splett (1982). *Semantik der Adjektive im Deutschen: Analyse der semantischen Relationen*. Westdeutscher Verlag.

Jiang, Jay J. & David W. Conrath (1997). Semantic similarity based on corpus statistics and lexical taxonomy. In *Proceedings of the 10th International Conference on Research in Computational Linguistics (ROCLING)*. Tapei, Taiwan.

Kunze, Claudia & Lothar Lemnitzer (2002). GermaNet - representation, visualization, application. In *Proceedings of the International Conference on Language Resources and Evaluation (LREC)*, Las Palmas, Canary Islands, Spain, 29 - 31 May, pp. 1485–1491.

Lemnitzer, Lothar & Claudia Kunze (2002). Adapting GermaNet for the Web. In *Proceedings of the first Global WordNet Conference, Central Institute of Indian Languages. Mysore, India*, pp. 174–181.

Lesk, Michael (1986). Automatic sense disambiguation using machine readable dictionaries: How to tell a pine cone from an ice cream cone. In *Proceedings of the 5th Annual International Conference on Systems Documentation*, Toronto, Ontario, Canada, June, pp. 24–26.

Lin, Dekang (1998). An information-theoretic definition of similarity. In *Proceedings of the 15th International Conference on Machine Learning*, San Francisco, Cal., pp. 296–304.

McCarthy, Diana, Rob Koeling, Julie Weeds & John Carroll (2004). Finding predominant senses in untagged text. In *Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics*, Barcelona, Spain, 21–26 July 2004, pp. 280 – 287.

Patwardhan, Siddharth, Satanjeev Banerjee & Ted Pedersen (2003). Using measures of semantic relatedness for word sense disambiguation. In *Proceedings of the Fourth International Conference on Intelligent Text Processing and Computational Linguistics*, Mexico City, Mexico, pp. 241–257.

Pedersen, Ted, Siddharth Patwardhan & Jason Michelizzi (2004). WordNet::Similarity – Measuring the relatedness of concepts. In *Demonstrations of the Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics*, Boston, Mass., 2–7 May 2004, pp. 267–270.

Resnik, Phil (1995). Using information content to evaluate semantic similarity in a taxonomy. In *Proceedings of the 14th International Joint Conference on Artificial Intelligence*, Montréal, Canada, 20–25 August 1995, Vol. 1, pp. 448–453.

Schmid, Helmut (1997). Probabilistic part-of-speech tagging using decision trees. In Daniel Jones & Harold Somers (Eds.), *New Methods in Language Processing*, Studies in Computational Linguistics, pp. 154–164. London, UK: UCL Press.

Vossen, Piek (1999). *EuroWordNet: a mutlilingual database with lexical-semantic networks*. Dordrecht: Kluwer Academic Publishers.