Extending, Trimming and Fusing WordNet for Technical Documents

Piek Vossen
Sail Labs
Address
Weesp, The Netherlands, 1382-VV
Piek.Vossen@irion.nl

Abstract

This paper describes a tool for the automatic extension and trimming of a multilingual WordNet database for cross-lingual retrieval and multilingual ontology building in intranets and domain-specific document collections. Hierarchies, built from automatically extracted terms and combined with the WordNet relations, are trimmed with a disambiguation method based on the document salience of the words in the glosses. The disambiguation is tested in a cross-lingual retrieval task, showing considerable improvement (7%-11%). The condensed hierarchies can be used as browse-interfaces to the documents complementary to retrieval.

Introduction

WordNet is a database that contains a mapping from the vocabulary of a language to a fund of concepts (Fellbaum 1998). It seems obvious to use a WordNet for information retrieval, since document indexes and queries can be converted to concept vectors instead of word stem vectors. The use of WordNet for information retrieval is however still not without dispute. According to Voorhees (1999), retrieval with WordNet expansion scores considerably lower than the baseline retrieval in TREC (13% and less). The main problem is the lack of proper disambiguation. In this paper, we present a system for customizing a multilingual WordNet database, compatible with EuroWordNet (Vossen 1998), for technical document collections in a domain. The system automatically builds a hierarchy using terms extracted from documents, which is combined with the WordNet hierarchy. Next, the hierarchy is trimmed to the context and can be combined with a personal ontology or classification.

The resulting term hierarchy represents a condensed classification of the document set. These hierarchies allow for fast terminology development, efficient translation of the terms and development of specialized ontologies. Secondly, the specialized hierarchies lead to considerable improvements for cross-lingual retrieval in intranet document collections. Finally, they provide users with a browse-interface to the documents, which is complementary to a cross-lingual retrieval engine.

In section 1, we describe the automatic extraction of terms and the building of extended WordNet hierarchies. In section 2, we explain how hierarchies can be trimmed with a disambiguation method that combines frequency information of the terms in the documents with the overall frequency of content words in the glosses. The effectiveness of the trimming is measured in a monolingual and cross-lingual retrieval experiment. Finally, section 3 shows how trees are customized and condensed by fusing and clustering.

1 Extending

Extension of the WordNet database is done to improve retrieval and access to information on support sites. These sites usually contain 10,000-40,000 technical documents, about 300-700MB HTML. The extraction of the terminology is done by the following procedure:

1. Extract significant NPs from shallow-parsed text.
2. Extract all salient and lexicalised multiword

sequences from the NPs.

3. Decompose the multiword sequences into head-modifier structures.
4. Fill the database with concepts from WordNet.
5. Build a hierarchy that combines the decomposition information with the concept hierarchy from WordNet.

1.1 Extracting terms from documents

NPs are automatically extracted on the basis of the syntactic structure of the text and the general text make up. The NPs are normalized, the head is lemmatised and a part of speech is assigned to the head (Noun or Proper Noun). Determiners and quantifiers are omitted. The NPs are stored as so-called topics with a frequency count, the part of speech, and the lemmatised head string. The exact (inflected) form is stored as a variant. We deliberately use the word *topic* instead of *term* because many extracted topics would not be listed in a terminology list.

Topics are often larger phrases that may contain useful parts. From the topics, we therefore extract so-called subtopics: all embedded multiword sequences. We do not store the document frequency for subtopics but only the number of topics in which they occur as a subtopic. This is called the element-frequency. In addition, we store so-called topic-to-topic relations between each subtopic and the topics from which they have been extracted.

Following Justeson and Katz (1995), we restrict the multiwords to salient combinations. Salience is based on the document frequency, the element frequency and the number of elements. The element frequency is adjusted for subtopics derived from non-salient topics. Salience of topics and subtopics is stored in the database. No complex formula is used here. We can directly set the frequency thresholds interactively, either as absolute values or proportional.

The best values for salience selection depend on the technicality of the domain, the number of documents and the size of the documents. For the support sites, we extract terms in the amount of: 125,000 topics, 100,000 subtopics, and, after selection, 80,000 salient topics and subtopics. Topics are salient with document frequency higher than 10 and element frequencies higher than 15, and less than 6 elements.

Manual inspection shows that about 10% of the salient topics is wrongly selected with these settings. Nevertheless, there are also good terms are neglected. Increasing the thresholds will improve the quality of the extracted terms but will also remove many more good candidates.

In Table-1, you see the most frequent multiword terms with *technology* as a head (out of 208 *technology* terms in total). As you can see, there are very-specific terms only relevant to the particular client, but there are also very general terms: *core technology, internet technology.*

Table-1 : Automatically extracted multiword terms

Normalized Key	Doc	Elem
lto technology	82	16
mmx technology	47	3
hp inkjet technology	44	7
backweb technology	30	1
inkjet technology	27	14
jetsend technology	27	5
internet technology	25	1
pa-risc technology	25	1
tapealert technology	20	1
cd technology	19	1
zoomsmart scaling technology	19	7
thermal inkjet technology	17	6
cutting edge web technology	16	0
cis technology	15	1
sign technology	15	0
epic technology	14	2

1.2 Decomposing multiword terms

After determining the salient terms, we need to organize them as a hierarchy. Following Grefenstette (1997), Woods (1997) and Morin and Jacquemin (1999), we extract subsumption relations from the syntactic head of the multiword term (or compound). The stored head of each topic and subtopic can directly be used to build a first tree or semantic network. The above *technology* examples will then all be stored as children of *technology*. However, this may not be the correct chunking in levels. Multiwords that consist of 3 or more words (e.g. *thermal inkjet technology*) could be linked to other multiwords (e.g. *inkjet technology*) that are also linked to *technology*, thus creating natural sub-levels.

The chunking of a multiword can be ambiguous. The above example can be decomposed in two ways: [*thermal inkjet*] [*technology*] or [*thermal*] [*inkjet technology*]. We therefore developed a heuristic to extract the most likely chunking. The decomposer will first look for the most salient head and then try to decompose the remaining multiwords into modifiers. The procedure is then as follows. First, we check if there are any lexicalized multiwords embedded in the multiword phrase. If the lexicalized multiword segments overlap, we select the most 'salient' candidate. If there are no lexicalized multiwords embedded in the multiword phrase, we apply the same criteria to all multiword segments.

Salience of the multiword segment is determined as follows. We first select the unit with the highest probability (proportional document and element frequencies). In the case of equality, we select the unit with most elements. If still equal, we take the longest string.

We store the relation for each segment in the database as a specific topic-to-topic relation (head or modifier). We then remove the multiword segment from the whole multiword topic and apply the same procedure to the remaining phrase. This process is repeated until the remainder is a single word. It is possible that there are no salient or lexical multiword phrases. In that case, we take the smallest head string and split all remainder words in single-word modifiers. Every multiword topic thus has at least a head-relation and one or more modifier relations to other topics in the database. Below is an example of the chunking that will result for some of the above *technology* examples.

```
technology
  printing technology
     digital printing technology
     smart printing technology
  inkjet technology
     generation inkjet technology
       next generation inkjet technology
     inkjet technology through third parties
     leading inkjet technology
       world leading inkjet technology
     hp thermal inkjet technology
  color layering technology
     photoret iii color layering technology
  color technology
```

Instead of a hierarchy with 2 levels, we thus have created 4 levels. There is however still a flaw in this structure. The levels of *generation inkjet technology* and *leading inkjet technology* are odd. It only makes sense to decompose multiword terms into other multiword terms, if there are other coordinate multiwords that share the same head. Structures like these are called **Stairs**, which can cover several steps of unary-branching nodes.

Stairs are corrected by removing all newly extracted multiwords with one child. After removal, all its descendants are lifted. Because the procedure works bottom-up, longer multiword terms may climb up a Stairs with several steps, up to a level where there are coordinate terms or a single word head. We remove about 350 new terms from a tree with 20,000 new terms extracted from technical support documents.

1.3 Linking Topics to Concepts

The above tree is completely based on the decomposition of multiwords. The tree will consist of as many tops as there are single words in the term database. This can be a few thousand for the collections of documents that we process.

There are several reasons why we would like to augment these compositional trees with a semantic network as WordNet:

1. WordNet synonyms can be used to cluster or merge nodes and thus branches in this tree;
2. WordNet can be used to reduce the number of tops by adding classifications of tops and intermediate levels;
3. Via WordNet synsets we can extract translations from topics to other languages wiith the multilingual WordNet database;

To integrate the WordNet hierarchy with the term hierarchy, we import related WordNet synsets as separate concept records into the database together with their concept-to-concept relations. All the imported concepts are linked to the topics if there is a match between the topic variants and the concept variants. Topics will get a list of concept references and concepts a list of topic references. There will also be concepts without topic references and topics without concept references.

In addition to the previous tree that was built from topic-to-topic head relations, we can now also build trees based on the concept-to-concept hyperonym relations from WordNet. However, it is also possible to combine a tree of topics with a tree of concepts by including both the topic-to-topic and concept-to-concept relations. We follow the topic-to-topic relations up to a topic that has concept references. At that point we represent the topic as a concept and follow the concept-to-concept relations.

Likewise, we can extend the above *technology* tree with the concept relations from WordNet, or, vice versa, extend a WordNet hierarchy with new terms that are decomposed via topic-to-topic relations:

```
psychological feature 1
  cognition 1
    cognitive content 1
      knowledge base 1
        branch of knowledge 1
          technology 2
            printing technology
            inkjet technology...etc..
act 1
  activity 1
    employment 2
      application 3
        technology 1
          printing technology
          inkjet technology...etc...
```

There are two different classifications for technology because there are two different meanings in WordNet. By simply merging the tree of topic relations with the tree of concept relations, we will thus duplicate the topic subtrees at every meaning of every concept.

There will also be a reduction of branches in the tree because of the collapse of synonyms. Since *engineering* belongs to the same synsets as *technology*, all topics related to *engineering* will be linked to the same concept as the topics linked to *technology*. We can merge about 650 topics for the above tree of 26,000 nodes, which includes about 4,500 concepts.

2 Trimming

The term *technology* only has two meanings in WordNet, but others have many more. Especially if polysemy occurs at several levels, this leads to an explosion of terms in the hierarchy. A tangled hierarchy with a lot of duplication is not useful to provide access to information. Furthermore, the selection of synonyms and equivalents across languages is hampered by the polysemy, leading to diffusion in information retrieval.

To prevent such an explosion, we trim the trees by limiting the concepts to the particular context. For disambiguating the topics, we make use of their frequency information and the glosses in WordNet. Following Mihalcea and Moldovan (1999), the glosses in WordNet are used as a context definition. However, instead of comparing the words in the glosses with the context in the text, we weight the words in the gloss using their frequency in the documents set, compared to the frequency in all the glosses:

$$\rho(w) = \frac{df(w)+ef(w)}{gf(w)}$$

The weight r of a content word **w** in the gloss is obtained by cumulating its document frequency **df** with the element frequency **ef** and dividing the sum by the frequency of this word in all the glosses of WordNet: **gf**. A word has a high weight, if it occurs frequently in the document set compared to its overall frequency in the glosses. The weight of a concept (**C**) is then based on the sum of the weights of the content words (**w$_i$**), divided by the total number of content words (**N**) in that gloss:

$$\rho(C) = \frac{\sum_{w_i}^{N} \frac{df(w_i)+ef(w_i)}{gf(w_i)}}{N}$$

For each topic, we removed all concepts with a weight less than 75% of the highest weight. In total, 12,408 WordNet concepts could be associated with 4,446 extracted topics. Finally, 4,912 concepts have been selected.

In doing so, we assume that words are used in the same sense, throughout the document collection. This is stricter than the one-sense-per-discourse-hypothesis (Gale, Church, Yarowsky 1992). Since, we work in homogenous domains this may still be valid.

We tested the disambiguation in a mono and cross-lingual information retrieval experiment on the same document set from which the terminology was extracted. The document set consists of 26,260 English HTML documents. A set of 100 English queries was used. We manually looked for the best matching document. Note that it there may still be other matches. The queries were translated to Dutch and French by native speakers. We have carried out the following retrieval runs:

Literal:	Query terms are directly matched with the English index terms.
EN-EN:	English queries to English documents, expansion with synonyms;
NL-EN:	Dutch queries translated to English and expanded with synonyms;
FR-EN:	French queries translated to English and expanded with synonyms;

The results are in Table-2 below. The baseline is represented by the first row, where there is no query expansion and translation.

Table-2: Retrieval results with trimmed and general Multilingual wordnets

	EN	FR	NL
Literal Queries	89.6	39.9	43.6
Query expansion			
- All meanings	82.4	54.2	54.3
- Trimmed	86.4	65.4	61.7

As the baseline for the mono-lingual retrieval, we matched literal English queries to English documents. This resulted in 89.6% as an average score. Using synonyms for all meanings decreases the English monolingual retrieval to 82.4%. This is due to the fact that we introduce less precise synonyms, which tends to have a negative effect in specialized documents. Disambiguation improves this with 4%.

A stronger effect can be expected for the cross-lingual retrieval. Here the baseline is to match the French and Dutch queries directly with the English index. Obviously, this gives poor results: 39.9% and 43.6%. It still works a little bit because of the specific terminology that is the same in all 3 languages. Here taking translations for all meanings increases retrieval to 54% and disambiguation improves on that to 65.4 for French and 61.7% for Dutch.

3 Fusing

The trimmed hierarchy contains 3 types of words, represented in Figure-1 below with different colours and characters:

1. Newly derived document terms: yellow (y).
2. Document terms linked to WordNet concepts: green (g).
3. WordNet hyperonyms without document occurrences: red (r).

We get a division of 20,000 new terms (yellow), related to 4,000 document terms with WordNet concepts (green) and 500 WordNet hyperonyms without document occurrences (red).

For many domain-specific and client-specific applications, the upper-level (the red area) contains classifications and distinctions that are not relevant. At the left side of Figure-1, we see the classifications we will get using WordNet, where we mainly represented the red levels. For browsing through a hierarchy a user has to traverse quite a few levels before he sees the distinctions that apply to the document set: i.e. give access to information.

To be able to customize the upper classification, we developed a fusion function. This function takes a source tree and will fuse it with a target tree. The target tree can be any imported or hand-made top-ontology, as long as it also includes a so-called interface level for the fusion. The interface level should contain concepts or topics that occur in both the source and target tree.

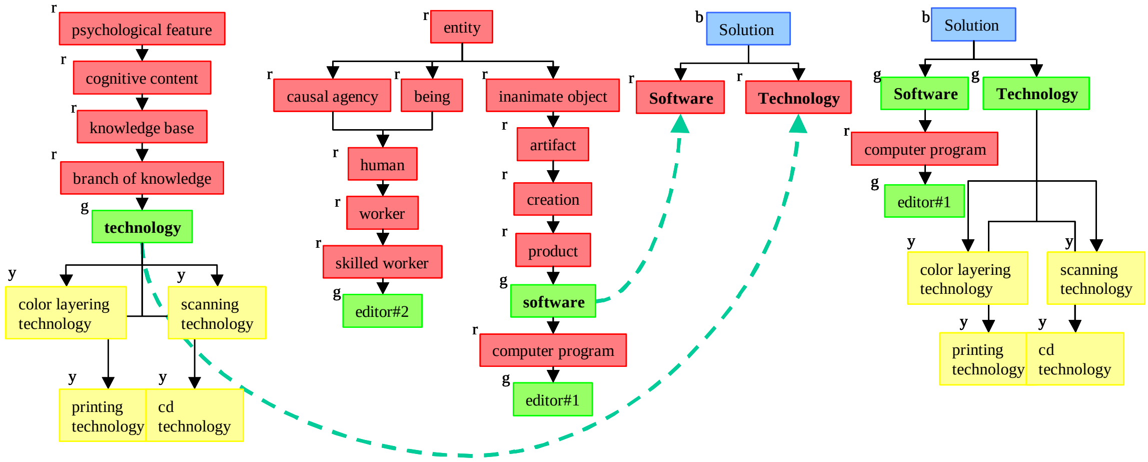

Figure-1: Fusion of an extended WordNet hierarchy with a private Top Ontology.

The fusion then works as follows. It will traverse the source tree bottom up or from right-to-left. Whenever it finds a matching node in the target tree, it will cut out the sub-tree from the source tree and place it below the matching node. If there is no match, it will go to the next node. The result of the fusion for this example is shown at the right side of Figure-1.

The hyperonym relations of the source tree are thus used to get at a level that matches an interface node in the target tree. This means that source trees are fused regardless of how deep and specific the terminology is. The interface nodes can be specified at any desired level of abstraction. In Figure-1, we see that *editor-2* will thus be related to the interface node *Software* via *computer program* (red), even though *computer program* itself does not occur in the documents. We also see that *editor-1* is not fused because there is no interface node for it. Tree fusion thus also filters concepts.

Furthermore, nodes can be re-classified. In this example, *Software* and *Technology* become coordinates of *Solution*, whereas they are totally unrelated in the WordNet hierarchy. In this way, the consistency of the private ontology classification is guaranteed.

If there is no match at any level, the complete branch will stay in the source Tree. In the end the source Tree is thus reduced to all branches for which there were no interface nodes. Inspecting the remaining source tree and extending the interface in the private top-tree can easily be done. You simply select the source nodes (and if necessary their descendant nodes) and drag them to the appropriate places in the private ontology. This can either be done for the fused end-results or just to improve the interface for the private ontology.

After fusing the source tree with a customized target top-tree, we apply node clustering. Following Peters et al. (1998) and Vossen et al. (1999), we cluster nodes when different senses of a word share the same hyperonym (sisters) or have a hyperonym-relation with each other (auto-hyponymy). In addition, we cluster:

1. **Compositional synonyms** the same hyperonym (co-hyponyms) and different modifiers that are synonyms: *e-mail application & e-mail software application, hewlett packard pc & hp desktop pc.*
2. **Product variants** topics with the syntax [company]+[brand]+version+[class], are synonyms of product names: *DeskJet 400, DeskJet 400 Series, HP DeskJet 400 printer.*
3. **Short cuts**: consisting of a modifier and a head, and, there is a co-hyponym that exactly matches just the modifier: *preview & preview image, database & database application, viewer & viewer application.*

For a hierarchy of 25,000 nodes in a technical domain, we find the following clusters: 689 product variants, 31 sisters, 17 auto-hyponymy, 343 compositional synonyms, and 32 short cuts. Note that each cluster involves at least two and possibly more nodes. Clustering removes several thousands of nodes. Finally, note that tree fusion will have an effect on the clusters that are derived because it affects the co-hyponymy relations.

By fusing and clustering, we can derive a condensed tree that has maximum coverage due to the extension, but only contains distinctions and classifications that are relevant and desired. Such a tree can be used as a monothetic classification interface to documents (Sanderson and Croft, 1999). In such an interface, you can browse through classified terms and access documents at each node. In addition, we provide the possibility to launch a query for the documents only related to the nodes in the tree. Because the hierarchy is trimmed and condensed but also extended, it gives the user a good feel for the content of the document collection.

Conclusion

In this paper, we described a tool for automatically extracting a document-specific hierarchy from text that is combined with the WordNet hierarchy. By trimming this hierarchy to the salient meanings and by fusing and clustering the trimmed tree to the relevant distinctions only, we showed that such a hierarchy can be made useful for providing users access to technical document collections. The trimming improved cross-lingual retrieval with 7%-11%, and the trees can be used to provide classification-browse access to documents. A pre-disambiguated tree no longer requires disambiguation for information retrieval. Furthermore, the trees make it easier to translate specific terminology and develop domain-specific ontologies.

Acknowledgements

I want to thank Michael Louw for implementing the database and Frank Platteau and Walter Lievens for testing the trimmed hierarchies.

References

Fellbaum, C. (ed.) (1998). *WordNet: An Electronic Lexical Database*. Cambridge, MA: MIT Press. 423 p.

Gale, W., K. Church, and D. Yarowsky (1992), *One Sense Per Discourse*. Proceedings of the 4th DARPA Speech and Natural Language Workshop. pp.233-237.

Grefenstette G. (1997), *SQLET: Short Query Linguistic Expansion Techniques, Palliating One-Word Queries by Providing Intermediate Structure to Text*. Proceedings of RIAO, 1997, pp. 500-509.

Justeson J.S. and S.M. Katz (1995) *Technical terminology: some linguistic properties and an algortihm for identification in text*. Natural Language Engineering, Volume 1, Part 1, March 1995, pp. 9-27.

Mihalcea R. and D.I. Moldovan (1999), *A Method for Word Sense Disambiguation of Unrestricted Text*, Proceedings of the 37th Annual Meeting of the ACL, University of Maryland, Maryland, pp. 152-158.

Morin E. and C. Jacquemin (1999), *Projecting Corpus-Based Semantic Links on a Thesaurus*, Proceedings of the 37th Annual Meeting of the ACL, University of Maryland, Maryland, pp. 389-396.

Peters W., I. Peters and P. Vossen (1998), *Automatic Sense Clustering in EuroWordNet*, Proceedings of LREC, 1998, Granada, pp. 409-416.

Sanderson M. and B. Croft (1999), *Deriving Concept Hierarchied from Text*, Proceedings of the 22nd ACM SIGIR Conference, pp. 206-213.

Vossen, P. (ed.) (1998) *EuroWordNet: A Multilingual Database with Lexical Semantic Networks*, Kluwer Academic Publishers, Dordrecht, 1998. 251 p.

Vossen P., W. Peters and J. Gonzalo (1999) *Towards a Universal Index of Meaning*. In Proceedings of the ACL-99 Siglex workshop, University of Maryland, June 21-22, 1999, pp. 81-90.

Voorhees E.M. (1999) *Natural Language Processing and Information Retrieval*. In "Information Extraction: Towards Scalable, Adaptable Systems", M. T. Pazienza, ed., Springer, Germany, pp. 32-48.

Woods W.A. (1997), *Conceptual indexing: A Better Way To Organize Knowledge*, a Sun Labs Technical Report: TR-97-61. Technical Reports, 901, Palo Alto, California 94303, USA.