Indexing with WordNet synsets can improve text retrieval

Julio Gonzalo and **Felisa Verdejo** and **Irina Chugur** and **Juan Cigarrán**
UNED
Ciudad Universitaria, s.n.
28040 Madrid - Spain
{julio,felisa,irina,juanci}@ieec.uned.es

Abstract

The classical, vector space model for text retrieval is shown to give better results (up to 29% better in our experiments) if WordNet synsets are chosen as the indexing space, instead of word forms. This result is obtained for a manually disambiguated test collection (of queries and documents) derived from the Semcor semantic concordance. The sensitivity of retrieval performance to (automatic) disambiguation errors when indexing documents is also measured. Finally, it is observed that if queries are not disambiguated, indexing by synsets performs (at best) only as good as standard word indexing.

1 Introduction

Text retrieval deals with the problem of finding all the relevant documents in a text collection for a given user's query. A large-scale semantic database such as WordNet (Miller, 1990) seems to have a great potential for this task. There are, at least, two obvious reasons:

- It offers the possibility to discriminate word senses in documents and queries. This would prevent matching *spring* in its "metal device" sense with documents mentioning *spring* in the sense of *springtime*. And then retrieval accuracy could be improved.
- WordNet provides the chance of matching semantically related words. For instance, *spring*, *fountain*, *outflow*, *outpouring*, in the appropriate senses, can be identified as occurrences of the same concept, '*natural flow of ground water*'. And beyond synonymy, WordNet can be used to measure semantic distance between occurring terms to get more sophisticated ways of comparing documents and queries.

However, the general feeling within the information retrieval community is that dealing explicitly with semantic information does not improve significantly the performance of text retrieval systems. This impression is founded on the results of some experiments measuring the role of Word Sense Disambiguation (WSD) for text retrieval, on one hand, and some attempts to exploit the features of WordNet and other lexical databases, on the other hand.

In (Sanderson, 1994), word sense ambiguity is shown to produce only minor effects on retrieval accuracy, apparently confirming that query/document matching strategies already perform an implicit disambiguation. Sanderson also estimates that if explicit WSD is performed with less than 90% accuracy, the results are worse than non disambiguating at all. In his experimental setup, ambiguity is introduced artificially in the documents, substituting randomly chosen pairs of words (for instance, *banana* and *kalashnikov*) with artificially ambiguous terms (*banana/kalashnikov*). While his results are very interesting, it remains unclear, in our opinion, whether they would be corroborated with real occurrences of ambiguous words. There is also other minor weakness in Sanderson's experiments. When he "disambiguates" a term such as *spring/bank* to get, for instance, *bank*, he has done only a partial disambiguation, as *bank* can be used in more than one sense in the text collection.

Besides disambiguation, many attempts have been done to exploit WordNet for text retrieval purposes. Mainly two aspects have been addressed: the enrichment of queries with semantically-related terms, on one hand, and the comparison of queries and documents via conceptual distance measures, on the other.

Query expansion with WordNet has shown to be potentially relevant to enhance recall, as it permits matching relevant documents that could not contain any of the query terms (Smeaton et al., 1995). However, it has produced few successful experiments. For instance, (Voorhees, 1994) manually expanded 50 queries over a TREC-1 collection (Harman, 1993) using synonymy and other semantic relations from WordNet 1.3. Voorhees found that the expansion was useful with short, incomplete queries, and rather useless for complete topic statements -where other expansion techniques worked better-. For short queries, it remained the problem of selecting the expansions automatically; doing it badly could degrade retrieval performance rather than enhancing it. In

(Richardson and Smeaton, 1995), a combination of rather sophisticated techniques based on WordNet, including automatic disambiguation and measures of semantic relatedness between query/document concepts resulted in a drop of effectiveness. Unfortunately, the effects of WSD errors could not be discerned from the accuracy of the retrieval strategy. However, in (Smeaton and Quigley, 1996), retrieval on a small collection of image captions - that is, on very short documents - is reasonably improved using measures of conceptual distance between words based on WordNet 1.4. Previously, captions and queries had been manually disambiguated against WordNet. The reason for such success is that with very short documents (e.g. *boys playing in the sand*) the chance of finding the original terms of the query (e.g. of *children running on a beach*) are much lower than for average-size documents (that typically include many phrasings for the same concepts). These results are in agreement with (Voorhees, 1994), but it remains the question of whether the conceptual distance matching would scale up to longer documents and queries. In addition, the experiments in (Smeaton and Quigley, 1996) only consider nouns, while WordNet offers the chance to use all open-class words (nouns, verbs, adjectives and adverbs).

Our essential retrieval strategy in the experiments reported here is to adapt a classical vector model based system, using WordNet synsets as indexing space instead of word forms. This approach combines two benefits for retrieval: one, that terms are fully disambiguated (this should improve precision); and two, that equivalent terms can be identified (this should improve recall). Note that query expansion does not satisfy the first condition, as the terms used to expand are words and, therefore, are in turn ambiguous. On the other hand, plain word sense disambiguation does not satisfy the second condition, as equivalent senses of two different words are not matched. Thus, indexing by synsets gets maximum matching and minimum spurious matching, seeming a good starting point to study text retrieval with WordNet.

Given this approach, our goal is to test two main issues which are not clearly answered -to our knowledge- by the experiments mentioned above:

- Abstracting from the problem of sense disambiguation, what potential does WordNet offer for text retrieval? In particular, we would like to extend experiments with manually disambiguated queries *and* documents to average-size texts.

- Once the potential of WordNet is known for a manually disambiguated collection, we want to test the sensitivity of retrieval performance to disambiguation errors introduced by automatic WSD.

This paper reports on our first results answering these questions. The next section describes the test collection that we have produced. The experiments are described in Section 3, and the last Section discusses the results obtained.

2 The test collection

The best-known publicly available corpus hand-tagged with WordNet senses is SEMCOR (Miller et al., 1993), a subset of the Brown Corpus of about 100 documents that occupies about 11 Mb. (including tags) The collection is rather heterogeneous, covering politics, sports, music, cinema, philosophy, excerpts from fiction novels, scientific texts... A new, bigger version has been made available recently (Landes et al., 1998), but we have not still adapted it for our collection.

We have adapted SEMCOR in order to build a test collection -that we call IR-SEMCOR- in four manual steps:

- We have split the documents to get coherent chunks of text for retrieval. We have obtained 171 fragments that constitute our text collection, with an average length of 1331 words per fragment.

- We have extended the original *TOPIC* tags of the Brown Corpus with a hierarchy of subtags, assigning a set of tags to each text in our collection. This is not used in the experiments reported here.

- We have written a summary for each of the fragments, with lengths varying between 4 and 50 words and an average of 22 words per summary. Each summary is a human explanation of the text contents, not a mere bag of related keywords. These summaries serve as queries on the text collection, and then there is exactly one relevant document per query.

- Finally, we have hand-tagged each of the summaries with WordNet 1.5 senses. When a word or term was not present in the database, it was left unchanged. In general, such terms correspond to groups (vg. *Fulton_County_Grand_Jury*), persons (*Cervantes*) or locations (*Fulton*).

We also generated a list of "stop-senses" and a list of "stop-synsets", automatically translating a standard list of stop words for English.

Such a test collection offers the chance to measure the adequacy of WordNet-based approaches to IR independently from the disambiguator being used, but also offers the chance to measure the role of automatic disambiguation by introducing different rates

Experiment	**% correct document retrieved in first place**
Indexing by synsets	62.0
Indexing by word senses	53.2
Indexing by words (basic SMART)	48.0
Indexing by synsets with a 5% errors ratio	62.0
Id. with 10% errors ratio	60.8
Id. with 20% errors ratio	56.1
Id. with 30% errors ratio	54.4
Indexing with all possible synsets (no disambiguation)	52.6
Id. with 60% errors ratio	49.1
Synset indexing with non-disambiguated queries	48.5
Word-Sense indexing with non-disambiguated queries	40.9

Table 1: Percentage of correct documents retrieved in first place

of “disambiguation errors” in the collection. The only disadvantage is the small size of the collection, which does not allow fine-grained distinctions in the results. However, it has proved large enough to give meaningful statistics for the experiments reported here.

Although designed for our concrete text retrieval testing purposes, the resulting database could also be useful for many other tasks. For instance, it could be used to evaluate automatic summarization systems (measuring the semantic relation between the manually written and hand-tagged summaries of IR-Semcor and the output of text summarization systems) and other related tasks.

3 The experiments

We have performed a number of experiments using a standard vector-model based text retrieval system, Smart (Salton, 1971), and three different indexing spaces: the original terms in the documents (for standard Smart runs), the word-senses corresponding to the document terms (in other words, a manually disambiguated version of the documents) and the WordNet synsets corresponding to the document terms (roughly equivalent to concepts occurring in the documents).

These are all the experiments considered here:

1. The original texts as documents and the summaries as queries. This is a classic Smart run, with the peculiarity that there is only one relevant document per query.

2. Both documents (texts) and queries (summaries) are indexed in terms of word-senses. That means that we disambiguate manually all terms. For instance “*debate*” might be substituted with “*debate%1:10:01::*”. The three numbers denote the part of speech, the WordNet lexicographer’s file and the sense number within the file. In this case, it is a noun belonging to the *noun.communication* file.

   With this collection we can see if plain disambiguation is helpful for retrieval, because word senses are distinguished but synonymous word senses are not identified.

3. In the previous collection, we substitute each word sense for a unique identifier of its associated synset. For instance, “*debate%1:10:01::*” is substituted with “*n*04616654”, which is an identifier for

   > “{*argument, debate1*}” (*a discussion in which reasons are advanced for and against some proposition or proposal; ”the argument over foreign aid goes on and on”*)

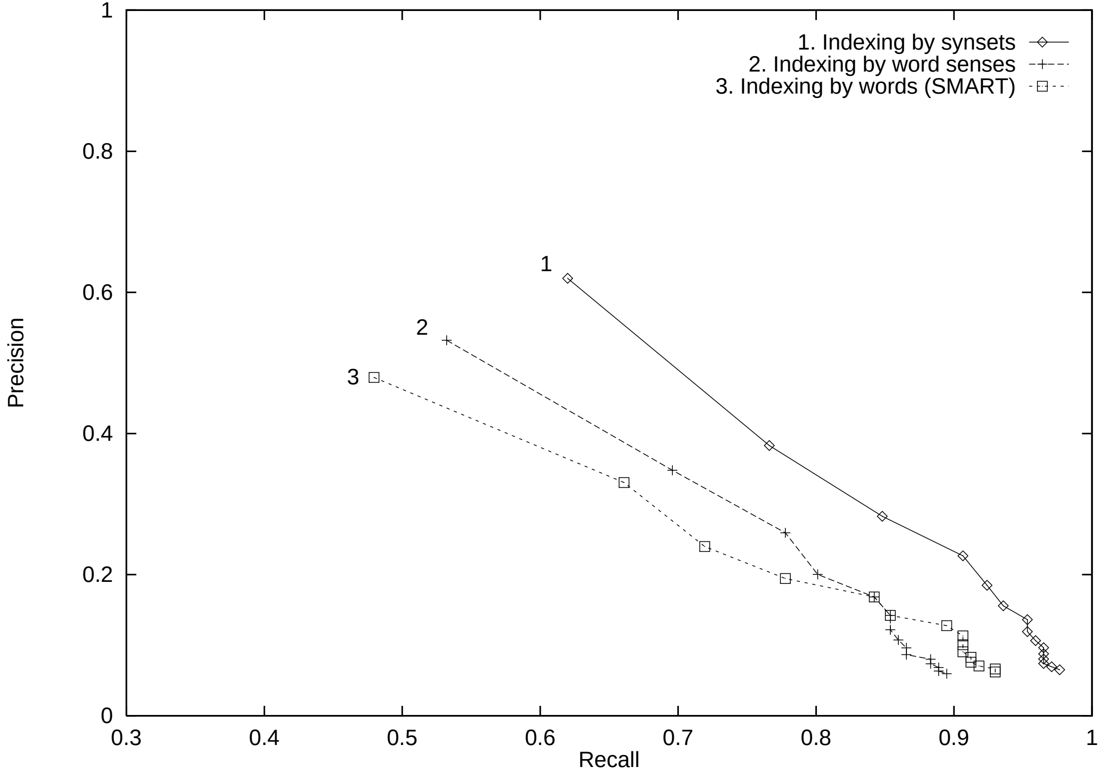
   This collection represents conceptual indexing, as equivalent word senses are represented with a unique identifier.

4. We produced different versions of the synset indexed collection, introducing fixed percentages of erroneous synsets. Thus we simulated a word-sense disambiguation process with 5%, 10%, 20%, 30% and 60% error rates. The errors were introduced randomly in the ambiguous words of each document. With this set of experiments we can measure the sensitivity of the retrieval process to disambiguation errors.

5. To complement the previous experiment, we also prepared collections indexed with all possible meanings (in their word sense and synset versions) for each term. This represents a lower bound for automatic disambiguation: we should not disambiguate if performance is worse than considering all possible senses for every word form.

6. We produced also a non-disambiguated version of the queries (again, both in its word sense and

Figure 1: Different indexing approaches



synset variants). This set of queries was run against the manually disambiguated collection.

In all cases, we compared `atc` and `nnn` standard weighting schemes, and they produced very similar results. Thus we only report here on the results for `nnn` weighting scheme.

4 Discussion of results

4.1 Indexing approach

In Figure 1 we compare different indexing approaches: indexing by synsets, indexing by words (basic SMART) and indexing by word senses (experiments 1, 2 and 3). The leftmost point in each curve represents the percentage of documents that were successfully ranked as the most relevant for its summary/query. The next point represents the documents retrieved as the first or the second most relevant to its summary/query, and so on. Note that, as there is only one relevant document per query, the leftmost point is the most representative of each curve. Therefore, we have included this results separately in Table 1.

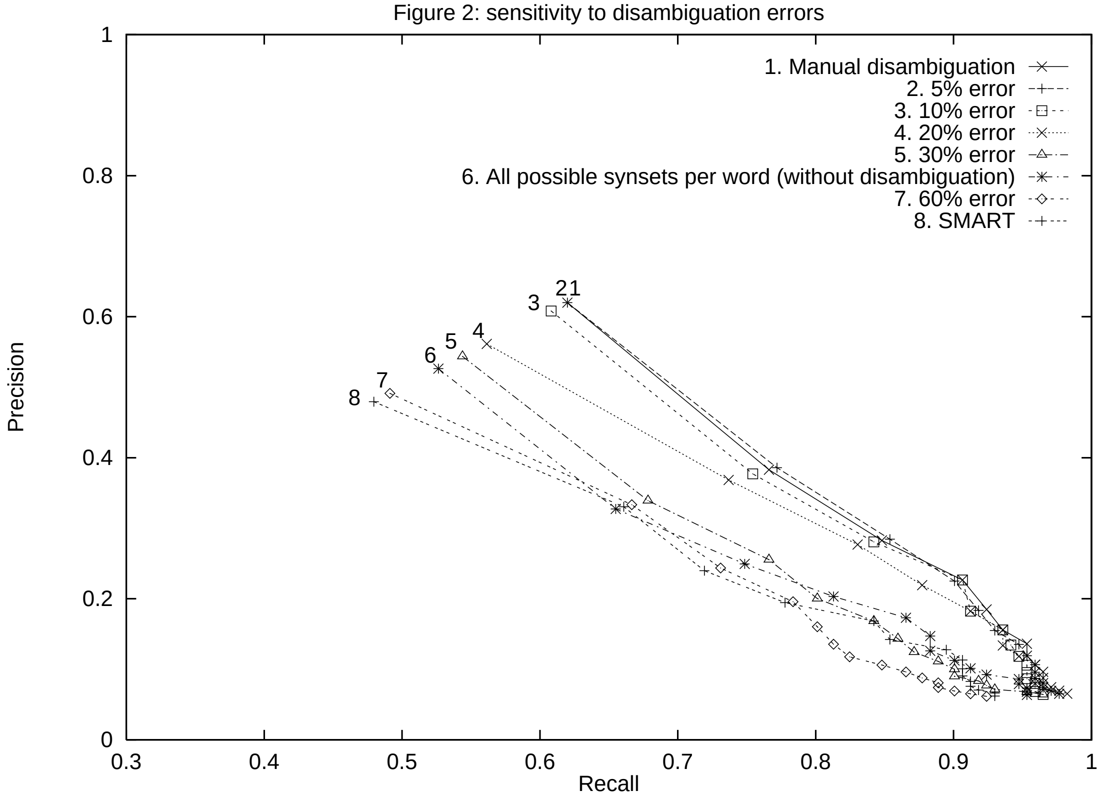
The results are encouraging:

- **Indexing by WordNet synsets** produces a remarkable improvement on our test collection. A 62% of the documents are retrieved in first place by its summary, against 48% of the basic SMART run. This represents 14% more documents, a 29% improvement with respect to SMART. This is an excellent result, although we should keep in mind that is obtained with manually disambiguated queries and documents. Nevertheless, it shows that WordNet can greatly enhance text retrieval: the problem resides in achieving accurate automatic Word Sense Disambiguation.

- **Indexing by word senses** improves performance when considering up to four documents retrieved for each query/summary, although it is worse than indexing by synsets. This confirms our intuition that synset indexing has advantages over plain word sense disambiguation, because it permits matching semantically similar terms.

  Taking only the first document retrieved for each summary, the disambiguated collection gives a 53.2% success against a 48% of the plain SMART query, which represents a 11% improvement. For recall levels higher than 0.85, however, the disambiguated collection performs slightly worse. This may seem surprising, as word sense disambiguation should only increase our knowledge about queries and documents. But we should bear in mind that WordNet 1.5 is not the perfect database for text retrieval, and indexing by word senses prevents some matchings that can be useful for retrieval. For in-

Figure 2: sensitivity to disambiguation errors



stance, *design* is used as a noun repeatedly in one of the documents, while its summary uses *design* as a verb. WordNet 1.5 does not include cross-part-of-speech semantic relations, so this relation cannot be used with word senses, while term indexing simply (and successfully!) does not distinguish them. Other problems of WordNet for text retrieval include too much fine-grained sense-distinctions and lack of domain information; see (Gonzalo et al., In press) for a more detailed discussion on the adequacy of WordNet structure for text retrieval.

4.2 Sensitivity to disambiguation errors

Figure 2 shows the sensitivity of the synset indexing system to degradation of disambiguation accuracy (corresponding to the experiments 4 and 5 described above). From the plot, it can be seen that:

- Less than 10% disambiguating errors does not substantially affect performance. This is roughly in agreement with (Sanderson, 1994).
- For error ratios over 10%, the performance degrades quickly. This is also in agreement with (Sanderson, 1994).
- However, indexing by synsets remains better than the basic SMART run up to 30% disambiguation errors. From 30% to 60%, the data does not show significant differences with standard SMART word indexing. This prediction differs from (Sanderson, 1994) result (namely, that it is better not to disambiguate below a 90% accuracy). The main difference is that we are using concepts rather than word senses. But, in addition, it must be noted that Sanderson's setup used artificially created ambiguous pseudo words (such as *'bank/spring'*) which are not guaranteed to behave as real ambiguous words. Moreover, what he understands as disambiguating is selecting -in the example- *bank* or *spring* which remain to be ambiguous words themselves.
- If we do not disambiguate, the performance is slightly worse than disambiguating with 30% errors, but remains better than term indexing, although the results are not definitive. An interesting conclusion is that, if we can disambiguate reliably the queries, WordNet synset indexing could improve performance even without disambiguating the documents. This could be confirmed on much larger collections, as it does not involve manual disambiguation.

It is too soon to say if state-of-the-art WSD techniques can perform with less than 30% errors, because each technique is evaluated in fairly different settings. Some of the best results on a comparable setting (namely, disambiguating against WordNet, evaluating on a subset of the Brown Corpus, and treating the 191 most frequently occurring and

Explore Litigation Insights

Docket Alarm provides insights to develop a more informed litigation strategy and the peace of mind of knowing you're on top of things.

Real-Time Litigation Alerts

Keep your litigation team up-to-date with **real-time alerts** and advanced team management tools built for the enterprise, all while greatly reducing PACER spend.

Our comprehensive service means we can handle Federal, State, and Administrative courts across the country.

Advanced Docket Research

With over 230 million records, Docket Alarm's cloud-native docket research platform finds what other services can't. Coverage includes Federal, State, plus PTAB, TTAB, ITC and NLRB decisions, all in one place.

Identify arguments that have been successful in the past with full text, pinpoint searching. Link to case law cited within any court document via Fastcase.

Analytics At Your Fingertips

Learn what happened the last time a particular judge, opposing counsel or company faced cases similar to yours.

Advanced out-of-the-box PTAB and TTAB analytics are always at your fingertips.

API

Docket Alarm offers a powerful API (application programming interface) to developers that want to integrate case filings into their apps.

LAW FIRMS

Build custom dashboards for your attorneys and clients with live data direct from the court.

Automate many repetitive legal tasks like conflict checks, document management, and marketing.

FINANCIAL INSTITUTIONS

Litigation and bankruptcy checks for companies and debtors.

E-DISCOVERY AND LEGAL VENDORS

Sync your system to PACER to automate legal marketing.

WHAT WILL YOU BUILD? | **sales@docketalarm.com** | **1-866-77-FASTCASE**