# SINAI at Robust WSD Task @ CLEF 2008: When WSD is a good idea for Information Retrieval tasks?

Fernando Martínez-Santiago, José M. Perea-Ortega, Miguel A. García-Cumbreras
SINAI Research Group. Computer Science Department. University of Jaén
Campus Las Lagunillas, Ed. A3, E-23071, Jaén, Spain
{dofer,jmperea,magc}@ujaen.es

**Abstract**

This year we have participated in the first edition of Robust WSD task with the aim of investigating the performance of disambiguation tools applied to Information Retrieval (IR). The main interest of our experimentation is the characterization of queries where WSD is a useful tool. That is, which issues must be fulfilled by a query in order to apply an state-of-art WSD tool? After the interpretation of our experiments, we think that only queries with terms very polysemous and very high IDF value are improved by using WSD.

## Categories and Subject Descriptors

H.3 [**Information Storage and Retrieval**]: H.3.1 Content Analysis and Indexing; H.3.3 Information Search and Retrieval; H.3.4 Systems and Software; H.3.7 Digital Libraries

## General Terms

Disambiguation, Information Retrieval, Experimentation

## Keywords

Robust WSD

## 1 Introduction

Word Sense Disambiguation (WSD) is a traditional task into the discipline of Natural Language Processing (NLP). WSD is the identification process of sense of a word (having a number of different senses) used in a given sentence [1]. Information Retrieval (IR) is a task even older than WSD into the NPL community. IR is defined as the matching of some stated user query against a set of free-text records [2]. These records could be any type of mainly unstructured text, such as newspaper articles, real estate records or paragraphs in a manual. User queries can range from multi-sentence full descriptions of an information need to a few words.

Nowadays, the information unit managed by most IR models is the word. A theoretical good idea is the elaboration of IR systems based on concepts better than words or the lemmas of those words. We define a concept as a lexicographic-independent representation of an idea or object. Given a language, it does not care the vocabulary available in order to represent such a concept. Thus, a concept-based IR system translates words into concepts. The advantages of such theoretical system are very interesting:

- Given a word with two or more senses, the representation of such word is different for every sense and only documents relative to the right concept will be retrieved.

- In the same way, the vocabulary of the user and the vocabulary of a given relevant document could be different. No matter common words, only common concepts.

- Finally, if the representation of the concepts is language-independent, virtually the IR system is multilingual.

Obviously, if we want to make a concept-based IR system, we need at least three resources:

1. A thesaurus or terminological ontology. Given a word, which concept or concepts are represented by such word? In the same way: given a concept, which word or words are suitable for such concept?

2. A WSD tool. It maps a word in the correct concept according to the terminological ontology and the context of the word. Words with a sense only would be a trivial case.

3. WordNet is a semantic lexicon for the English language. It groups English words into sets of synonyms called *synsets*, provides short, general definitions, and records the various semantic relations between these synonyms sets [3].

However, in spite of the impressive amount of available resources, nowadays there is not any concept-based IR system that outperforms the best word-based IR systems. An usual reason given in the literature is that WordNet is excessively fine-grained. By example, "*house*" has 14 different senses, and "*give*" has more than 30 senses. In this way, Agirre and Lopez de Lacalle [4] depict a set of methods to cluster WordNet word senses. Respect of WSD and IR, Gonzalo et al. [5] claim that concepts-based indexes outperform words-based indexes only if the WSD tool that outperforms 90% of recall. State-of-art WSD tools obtain about 60% of precision/recall [6, 7] for "*fine-grained all words*" task[1]. Is this enough to improve an IR system? Which queries are improved and which queries are damaged? Which issues will determinate such sets of queries? After the interpretation of our experiments, we think that only queries with terms very polysemous and very high IDF value are improved by using WSD.

## 2 Experimental Framework

In the experiments carried out in this paper we have used the two disambiguated collections provided by the NUS [6] and UBC [7] teams and the default collection for Robust WSD task without WSD data. The default English collection for the Robust WSD task consists of 169,477 documents composed of stories from the British newspaper *Glasgow Herald* (1995) and the American newspaper *Los Angeles Times* (1994).

For each disambiguated collection we have generated four different indexes:

- **A index type**. This index stores each token and its *synset code* which has the highest score.

- **B index type**. This index stores only the *synset code* which has the highest score for each disambiguated token.

- **A2 index type**. It is the same as A index type but adding the two *token+synset* which have the highest score.

- **B2 index type**. It is the same as B index type but adding the two *synset codes* which have the highest score.

[1] Fine-grained all words is the name of a usual WSD task. In this paper, we have used WSD in a very similar way.

| Experiment | WSD system | Index unit | AvgP |
|---|---|---|---|
| NUS-indexA-TD | NUS team | token+sysnset (type A) | 0.32 |
| NUS-indexB-TD | NUS team | synset only (type B) | **0.35** |
| NUS-indexA2-TD | NUS team | two first token+synsets (type A2) | 0.27 |
| NUS-indexB2-TD | NUS team | two first synsets (type B2) | 0.27 |
| Baseline case | none | stem of the word | 0.40 |

Table 1: The most outstanding results

In addition to these 8 indexes (4 for UBC team and 4 for NUS team), we have generated four common indexes (common-A, common-B, common-A2 and common-B2), merging a token from each disambiguated collection. Therefore, we have generated a total of 12 different indexes for the experiments with WSD data.

For the default collection without WSD data we have preprocessed it, making use of the *Porter stemmer* [8] and discarding the English *stop-words*.

# 3 Experiments and Results

We report only the results obtained by using the disambiguated collection by using the system developed by the NUS team. Anyway, when we have used the WSD system of UBC, we have obtained an unusual low *average precision* (AvgP), so we suppose we have some errors of implementation. We hope we will able to solve the errors for the revised version of this paper.

The selected set of experiments is depicted in the Table 1. As we expect, applying WSD in "*a blind way*" to improve IR does not work. The baseline case obtains better result than indexes based on disambiguated collections. We do not think that results get better by using other WSD tool, since the collections were disambiguated by using an state-of-art disambiguation software. On the other hand, the *synsets*-based indexes improve the indexes based on *term+synset*. We conclude that taking into account synonymous gets meaningful improvement.

Thus, state-of-art WSD systems must not be applied in the same way as other usual IR techniques such as pseudo-relevance feedback (PRF) or stemming, by example. The question is: Does exist any sort of queries where WSD should be apply? If so, how could we recognize such queries?

In order to carry out a more detailed analysis of results, we compared the *baseline* and "*NUS-indexB*" (disambiguation by using NUS WSD system) cases. *NUS-indexB* obtains better average precision than baseline case in 58 queries. It means improving 36.2% of queries by using disambiguated queries. If we count only the queries improved more than a 10%, a remarkable 28.6% (46 queries) is obtained. Thus, we aim to recognize a common set of properties in order to define these sets of queries in order to apply WSD properly for the IR task.

The first intuition that we want to evaluate is if "*very polysemous queries will be improved by WSD*". If we take into account the original 160 queries, the average number of senses per word is 2.39 (*stop-words* have been eliminated). If we take into account the 58 queries improved by using WSD, the average number of senses per word is 2.37. Finally, the average number of senses per word is 2.43 for not improved queries by using WSD (102 queries). These results are disappointing. A more detailed analysis reveals that non-empty words such as "*find*" or "*information*" are very common. In addition, these words are polysemous and they have very poor semantic weight.

Table 2 shows some queries where the difference between the baseline case and disambiguated index is noteworthy. Differences between both of them are huge so we think that the impact of WSD must be studied deeply. Next step is the evaluation at term level. In order to get an idea of the situation, we analyze some words. Results are depicted in the Table 3.

This is a very preliminary work, but there some interesting issues:

- There are words with very low IDF and very polysemous. By example, "*give*" is not a very interesting word for usual IR systems. Anyway, if the IR system uses an index based

| Query id | Query text (Title+Description) | AvgP using baseline | AvgP using *NUS-indexB* | Avg. of word senses |
|---|---|---|---|---|
| 10.2452/180-AH | Bankruptcy of Barings. What was the extent of the losses in the Barings bankruptcy case? | 0.025 | 0.765 | 3.71 |
| 10.2452/151-AH | Wonders of Ancient World. Look for information on the existence and/or the discovery of remains of the seven wonders of the ancient world | 0.061 | 0.571 | 3.54 |
| 10.2452/190-AH | Child Labor in Asia. Find documents that discuss child labor in Asia and proposals to eliminate it or to improve working conditions for children | 0.887 | 0.123 | 2.07 |
| 10.2452/252-AH | Pension Schemes in Europe. Find documents that give information about current pension systems and retirement benefits in any European country | 0.444 | 0.15 | 4 |

Table 2: Some queries where the difference between the baseline case and disambiguated index is noteworthy

| Term | Query id | IDF | *synset* (NUS) | NUS Confidence | Correct sense? |
|---|---|---|---|---|---|
| bankruptcy | 10.2452/180-AH | 4.76 | 0386165-n | 0.52 | Yes |
| ancient | 10.2452/151-AH | 4.42 | 01665065-a | 0.43 | Yes |
| world | 10.2452/151-AH | 1.64 | 06753779-n | 0.13 | No |
| child | 10.2452/190-AH | 2.94 | 07153837-n | 0.79 | No |
| give | 10.2452/252-AH | 1.597 | 01529684-v | 0.37 | No |

Table 3: Some terms and data about these terms

on *synsets*, then the IDF of each word increases because of polysemy: obviously, in an index based on *synsets*, every sense of each word will obtain an IDF rather higher than the corresponding word in an index based on stems or lemmas.

- On the other hand, there are words like "*bankruptcy*" or "*ancient*" in which the IDF is high, so the IDF of the corresponding disambiguated *synset* will be high, too. If the WSD software has a high confidence in order to assign the correct sense, then we think this is a good candidate of word to be disambiguated.

In order to obtain reliable conclusions we need a very elaborate list of words and a lot of information about how the word is being disambiguated in the query and in the document collection, and how the correct/erroneous disambiguation of the word affects to the final score of the document. Anyway, we would go so far as to say a first approximation: words with low IDF and a high number of senses must be not disambiguated; too much risk and too few benefit. On the other hand, words with high IDF and high disambiguation confidence must be disambiguated. Of course, this heuristic must be refined, studied and evaluated, but we think that the idea is correct: the selective application of WSD in the IR.

# 4 Conclusions and Future Work

State-of-art WSD is not an useful tool for every query, for every term of every query, but we think that some queries could be improved by using WSD. In this paper we investigate queries where WSD gets better results. We find that there are situations where WSD must be used, but these scenarios are very specific. Since some queries are improved by WSD and some queries not at all, if we want to apply WSD in a good way we have to manage two indexes per collection. In addition, the IR system will have to carry out a bit of additional analysis of the user query in order to take a decision about which of both indexes seem more suitable for each user query.

As future work, we think that there are promising ways to improve the obtained results. We want to explore a selective and fragmented evaluation of queries. We think that, given a user query, some words should be disambiguated and others do not. Thus, some words should be evaluated by using a index (the disambiguated one), and some words should be evaluated by using other index (the non-disambiguated one). We think that this line of investigation is promising, but some questions arise: which words should be disambiguated and which queries should not? This question is partially investigated in this text but a more in-depth analysis of results at word level is required. In this way, since we will have to manage simultaneously two indexes, how to calculate the score of each document for a given query? Finally, we think that the combination of this "*fragmented evaluation of queries*" and the application of clustering of senses such as is depicted in [4] will improve this future model proposed.

# 5 Acknowledgments

This work has been supported by a grant from the Spanish Government, project TIMOM (TIN2006-15265-C06-03), and the RFC/PP2006/Id_514 granted by the University of Jaén.

# References

[1] Y. Wilks, B. Slator, and L. Guthrie. *Electric words: dictionaries, computers and meanings.* Cambridge, MA: MIT Press, 1996.

[2] G. Salton and M. J. McGill. *Introduction to Modern Information Retrieval.* McGraw-Hill Book Company, London, U.K., 1998.

[3] C. Fellbaum. *WordNet: an electronic lexical database. Language, speech, and communication.* Cambridge, Mass: MIT Press, 1998.

[4] Eneko Agirre and Oier Lopez de Lacalle. Clustering wordnet word senses. In *Recent Advances on Natural Language (RANLP), Borovets, Bulgary*, 2003.

[5] Julio Gonzalo, Felisa Verdejo, and Irina Chugur. Indexing with wordnet synsets can improve text retrieval. pages 38–44, 1998.

[6] Jun Fu Cai, Wee Sun Lee, and Yee Whye Teh. Nus-ml:improving word sense disambiguation using topic features. In *Proceedings of the Fourth International Workshop on Semantic Evaluations (SemEval-2007), Prague, Czech Republic*, pages 249–252, 2007.

[7] Eneko Agirre and Oier Lopez de Lacalle. Ubc-alm: Combining k-nn with svd for wsd. In *Proceedings of the Fourth International Workshop on Semantic Evaluations (SemEval-2007), Prague, Czech Republic*, pages 342–345, 2007.

[8] M.F. Porter. An algorithm for suffix stripping. In *Program 14*, pages 130–137, 1980.