Web-based WSD using Adjective-Noun pairs

Davide Buscaldi[1], Manuel Montes y Gómez[2,3], and Paolo Rosso[2]

[1] Dipartimento di Informatica e Scienze dell'Informazione (DISI),
Università di Genova, Italy
buscaldi@disi.unige.it
[2] Dpto. de Sistemas Informáticos y Computación (DSIC),
Universidad Politecnica de Valencia, Spain
{mmontes, prosso}@dsic.upv.es
[3] Lab. de Tecnologías del Lenguaje
Instituto Nacional de Astrofsica, Optica y Electrónica, México
mmontesg@inaoep.mx

Abstract. This paper investigates the effectiveness of using the redundancy of the web for solving the Word Sense Disambiguation task. The web-based algorithm looks for the adjective-noun pairs in the web to disambiguate an english noun. Preliminary results show that a better precision than the baseline is obtained but with a low recall. Moreover, the web seems to be more effective than the WordNet Doamains when integrated rather than stand-alone.

1 Introduction

The problem of the resolution of the lexical ambiguity that appears when a given word in a context has several different meanings is commonly referred as Word Sense Disambiguation (WSD). Both supervised and not supervised paradigms for WSD seem to be stuck because of the knowledge acquisition bottleneck. Usually, samples size is too small and, therefore, it is worthwhile to investigate the possibility to use the Web as a resource for WSD. In this paper, we describe our first attempt to use the web to understand the sense of an english noun taking into account the adjective which goes before it. The probability of finding an adjective-noun pair increases with the redundancy of the web. Our work is inspired by [8] in which the use of the redundancy of the web [1] was employed to look for noun-verb relationships.

In the following sections we describe our web-based algorithm, the description of the formulae we studied during our preliminary experiments, a comparison of search engines and the discussion of the experimental results we obtained. Finally, conclusions and future work are discussed in Section 6.

2 Description of the Web-based Algorithm

The disambiguation of a noun (w) with $|w|$ senses is carried out by taking into account the adjective (a) referring to the noun itself, the n synonyms $\{s_{ik}, 0 \leq$

$i < n\}$ of the k-th sense (w_k) of the noun, and the m words in the direct hypernym synset of w_k, $\{h_{jk}, 0 \leq j < m\}$ (some of the formulae of the next section will include hyponym synsets as well). We name $f_S(x, y)$ the function returning the number of pages containing the pair "x y", obtained by searching the web with a search engine S, where x and y are strings. Moreover, we name $f_S(x)$ the function returning the number of pages containing the string x, by using a search engine S. In some cases we used also the direct hyponyms of the noun.

The algorithm is basically divided into the following steps:

1. Select the adjective a immediately before the noun w;
2. for each sense w_k of w, and for every synonym s_{ik} and direct hypernym h_{jk} of w_k, compute $f_S(a, s_{ik})$ and $f_S(a, h_{jk})$;
3. assign to each sense w_k a weight depending on a formula F which combines the results obtained in the step before;
4. select the sense having the resulting higher weight.

For example, consider the following sentence, extracted from the Senseval-3 All-Words task corpus: *A faint crease appeared between the man's eyebrows.* Suppose we are disambiguating the word *crease*, having three senses, according to WordNet 2.0: $crease_1 : \{fold, crease, plication, flexure, crimp, bend\}$, $crease_2 : \{wrinkle, furrow, crease, crinkle, seam, line\}$ and $crease_3 : \{kris, creese, crease\}$. The direct hyperonyms are, for each sense: h_1 ={*angular shape, angularity*}, h_2 ={*depression, impression, imprint*} and h_3 ={*dagger, sticker*}. Then we search the web for the following pairs: (*faint fold*), (*faint plication*), (*faint flexure*), (*faint crimp*), (*faint bend*), (*faint angular shape*), (*faint angularity*) for the first sense, (*faint wrinkle*), (*faint furrow*), etc. for the second sense and so on. The weights obtained for each sense depend on the formula and the search engine used.

3 Description of Formulae

We introduce the comprehensive list of formulae used during the Web-based WSD experiments. Each formula returns a weight for the k-th sense of the noun to disambiguate W.

- F_1: This is the simplest formula, based on the average of weights:

$$F_1 = 1/2 * \left(\frac{\sum_{i=0}^{n} f_S(a, s_{ik})}{n} + \frac{\sum_{j=0}^{m} f_S(a, h_{jk})}{m} \right) \tag{1}$$

- F_2: This is the same as above, but it takes into account also the probabilities of each synonym s_{ik} and each hypernym h_{jk} of having the same sense of w_k. The probabilities $p(s_{ik}|w_k)$, which is the prior of each sense,and $p(x|w_k)$ were calculated over the SemCor corpus. Even if probabilities vary with domain, in this first approximation we assumed that they are the same over the SemCor and the web.

$$F_2 = 1/2 * \left(\frac{\sum_{i=0}^{n} f_S(a, s_{ik}) * p(s_{ik}|w_k)}{n} + \frac{\sum_{j=0}^{m} f_S(a, h_{jk}) * p(h_{jk}|w_k)}{m} \right) \quad (2)$$

The motivation of taking into account this probability is that some words can appear in the web with a different sense than the appropriate one, e.g. *air* as synonym of *melody* is rare, with a probability $p(\text{“}air\text{”}|6598312) = 0.0022$, where 6598312 is the WordNet 2.0 offset corresponding to the synset {*tune, melody, strain, air, line, melodic line, melodic phrase*}.

- F_3: This formula derives directly from F_1, taking into account also the hyponyms of the sense w_k of the noun to disambiguate. The hyponym weights are computed in exactly the same way of the synonyms and hypernyms in the formula above, the 1/2 is replaced by 1/3. The hyponyms can be seen as “use cases” of the sense of the word to disambiguate.
- F_4: This formula is the same of F_3 with the difference that it takes into account also the probabilities of synonyms, hypernyms and hyponyms of having the same sense of w_k.
- F_5: This formula uses the Conceptual Density (CD) [9] approach in order to select the roots of subhierarchies. The words in these root synsets are used in the same way as synonyms and hypernyms above to calculate weights. The roots of subhierachies are the points at which senses of the word start to differentiate. These weights are added to the formula F_4, where 1/3 is replaced by 1/4.
- F_6: The same as F_4, where hyponyms have been replaced by roots of subhierarchies.
- F_7: With this formula we calculate the maximum instead of the average.

$$F_7 = \max_{\substack{0 \leq i < n, \\ 0 \leq j < m}} (f_S(a, s_{ik}), f_S(a, h_{jk})) \quad (3)$$

- F_8: This is the F_7 formula with the probability weights:

$$F_8 = \max_{\substack{0 \leq i < n, \\ 0 \leq j < m}} (f_S(a, s_{ik}) * p(s_{ik}|w_k), f_S(a, h_{jk}) * p(h_{jk}|w_k)) \quad (4)$$

- F_9: This formula is the same of F_8, but it takes into account also the hyponyms.
- F_{10}: This formula, inspired by the F-measure, is obtained in the following way:

$$F_{10} = \frac{2 * \frac{\sum_{i=0}^{n} f_S(a, s_{ik})}{n} * \frac{\sum_{j=0}^{m} f_S(a, h_{jk})}{m}}{\frac{\sum_{i=0}^{n} f_S(a, s_{ik})}{n} + \frac{\sum_{j=0}^{m} f_S(a, h_{jk})}{m}} \quad (5)$$

- F_{11}: The *Mean Mutual Information*[10], or *Relative Entropy*, measures how much information is in the dependency of two successive words. It has been adapted to take into consideration information obtained both by synonyms and hypernyms:

$$F_{11} = \sum_{i=0}^{n} f_S(a, s_{ik}) \log \frac{f_S(a, s_{ik})}{f_S(a) * f_S(s_{ik})} + \sum_{j=0}^{m} f_S(a, h_{jk}) \log \frac{f_S(a, h_{jk})}{f_S(a) * f_S(h_{jk})} \tag{6}$$

  its main drawback is that the obtained weights are always negatives, due to the fact that the denominators are always much greater than the numerators.
- F_{12}: This formula is based on the *Dice coefficient*[5], which measures the similarity between two sets.

$$F_{12} = \max_{\substack{0 \le i < n, \\ 0 \le j < m}} \left(\frac{2 * f_S(a, s_{ik})}{f_S(a) + f_S(s_{ik})}, \frac{2 * f_S(a, h_{jk})}{f_S(a) + f_S(h_{jk})} \right) \tag{7}$$

  In our case the two sets are the documents in the web containing the adjective a, and the set of documents containing the word s_{ik}, or h_{jk}. A high similarity between these sets means that the synonym words are often attributed by a. Therefore, this measure can be viewed as a measure of the adjective's relevance.
- F_{13}: This formula is derived from F_{11} and F_{12}, and resembles the *Similarity Theorem*[6]:

$$F_{13} = \max_{\substack{0 \le i < n, \\ 0 \le j < m}} \left(f_S(a, s_{ik}) * \frac{\log f_S(a, s_{ik})}{\log f_S(s_{ik})}, f_S(a, h_{jk}) * \frac{\log f_S(a, h_{jk})}{\log f_S(h_{jk})} \right) \tag{8}$$

  The formula reduces the problem of large denominators thanks to the use of logarithms.
- F_{14}: This formula is the same of F_{13} with the inclusion of hyponyms.
- F_{15}: This formula calculates the standard deviation between the synonyms and hyponyms weights (calculated with F_{13}) and the weight for the word we are disambiguating (i.e., $f_S(a, W) * \frac{\log f_S(a,W)}{\log f_S(W)}$). Therefore, the sense having the minimum standard deviation is selected.

4 Comparison of Search Engines

The search engines used were AltaVista[4] and MSN Search[5]. We used also the Lucene search engine[6], substituting the web with the TREC-8 collection[7], in

[4] http://www.altavista.com

[5] http://search.msn.com

[6] http://jakarta.apache.org/lucene

[7] http://trec.nist.gov

order to evaluate the effectiveness of the web with respect to a large document collection. We initially planned to use also Google[8], but we later abandoned it because of the limitations on the daily number of queries: only 1000 queries at a day are allowed, while our most query-demanding experiment needed about 14000 queries. In order to have an idea of how similar results could be when different search engines are used, we compared the precision, recall and coverage obtained over the Senseval-3 AWT corpus. These experiments were carried out using just the first seven formulae of those described in the previous section. The results obtained evidentiate that MSN and AltaVista are equivalent (even if we obtained slight differences in some experiments, on average results are almost the same for both engines). In the following experiments we preferred MSN due to a lower response time for the queries (e.g. the duration of the most demanding experiment was of 237 minutes with AltaVista, 171 minutes with MSN). A remarkable difference between the web-based search engines and the Lucene (which uses the TREC-8 collection instead of the web) is the 6% drop in coverage obtained when using the TREC-8 collection instead of the web, confirming that the huge redundancy of data in the web allows to disambiguate a greater number of words. Moreover, the precision obtained with Lucene is only 1% higher than the precision obtained when using the web. We expected an higher precision, due to the lower quality of the information in the web, however the improvement is not so evident to justify the use of a large text collection instead of the web.

5 Experimental Results

The aim of this first attempt to use the web to perform the noun sense disambiguation in english, was to study the adjective-noun pair pattern. Table 1 shows all the results of the preliminary experiments we carried out using the MSN search engine and all the formulae described previously. For some of the formulae was also included a *frequency correction* (fc) which indicates whether the resulting weight for a sense w_k was multiplied by $p(w|w_k)$, the probability for the word w of having sense w_k in the SemCor corpus, or not.

An interesting result is that the frequency-corrected formulae outperform those without the frequency factor. An average 4% gain is obtained in recall when frequency is taken into account. In one case (with F_{13}) we obtained better results than the Most Frequently Sense (MFS) baseline. We believe the better performance could depend on the fact that this formula has the advantage of not taking into account only the relevance of the adjective but also the number of co-occurrences of the adjective-noun pair. The web-based disambiguation provided better results even than the Conceptual Density - WordNet Domains (WND) approach [3][7], especially over the words not disambiguated by the standard CD method (16.6% with respect to the CD+WND formula).

Another interesting result is that the hyponym information could be helpful, since the formulae which take into account hyponyms (e.g. F_3 and F_4) have

[8] http://www.google.com

F	fc	$Precision$	$Recall$	$Coverage$	P_1	P_2
MFS		0.689	0.689	100%	0.623	0.629
CD		0.734	0.518	70.5%	0.625	0.000
$CD + WND$		0.653	0.584	89.3%	0.583	0.500
F_1	no	0.636	0.274	43.1%	0.337	0.348
F_2	no	0.627	0.271	43.3%	0.318	0.328
F_2	*yes*	0.718	0.311	43.3%	0.511	0.478
F_3	no	0.658	0.285	43.3%	0.387	0.338
F_4	no	0.661	0.286	43.3%	0.392	0.367
F_4	*yes*	0.759	0.329	43.3%	0.596	0.507
F_5	no	0.658	0.285	43.3%	0.381	0.348
F_6	no	0.643	0.278	43.3%	0.349	0.333
F_7	no	0.645	0.269	41.7%	0.333	0.357
F_8	no	0.639	0.268	41.9%	0.323	0.339
F_8	*yes*	0.709	0.306	43.1%	0.489	0.456
F_9	no	0.660	0.278	42.0%	0.373	0.333
F_9	*yes*	0.755	0.326	43.1%	0.586	0.500
F_{10}	no	0.661	0.272	41.7%	0.361	0.327
F_{10}	*yes*	0.696	0.286	41.2%	0.439	0.426
F_{11}	no	0.579	0.239	41.2%	0.179	0.152
F_{11}	*yes*	0.720	0.259	42.1%	0.532	0.565
F_{12}	yes	0.662	0.275	41.7%	0.424	0.444
F_{13}	*yes*	**0.777**	0.337	43.3%	**0.634**	**0.666**
F_{14}	*yes*	0.764	0.331	43.3%	0.607	0.623
F_{15}	*yes*	0.756	0.327	43.2%	0.589	0.594

Table 1. Performance over nouns. F: formula; fc: frequency correction; $Prec.$: overall precision; *Rec.*: overall recall; *Cov.*: overall Coverage; P_1: Precision over the disambiguated nouns (i.e., nouns such that "adjective noun"); P_2: Precision over the nouns not disambiguated by the CD method [2]; F_1-F_6: formulae based on weight average; F_7-F_{10}: formulae based on weight maximum; F_{11}-F_{15}: formulae based on similarity measures.

usually better performance than the related formulae which use only synonyms and direct hypernyms (e.g. F_1 and F_2). A further investigation is needed in order to understand better this issue, because for F_{14} hyponyms hurt rather than help. The error analysis allowed us to better understand each formula. For instance, in F_{12} it seems that heavier weights are given to proper nouns constituted by an adjective-noun pair (e.g. *national insurance*, which is an insurance company). In F_5, instead, roots of subhierarchies did not prove to be particularly effective, because of the problems of multi-word expressions and shallowness of roots of subhierarchies In fact, we observed that these roots are often too shallow in the subhierarchies (e.g. *entity*, or too deep (in this case they correspond to the synsets of the word to disambiguate). The resulting effect on disambiguation is to add respectively noise or no information. Moreover, in many cases the roots can contain words of uncommon use, like *psychological feature*, having very few occurrences in the Web (e.g. *psychological feature* appears only in 195 pages using MSN Search). Again, the result is that no useful information can be found in these cases. We investigated also the importance of polysemy of adjectives in the disambiguation of words. We calculated the averaged polysemy of the adjective when the referred word was disambiguated correctly and when the word was assigned the wrong sense.

F	fc	$Right$	$Wrong$	F	fc	$Right$	$Wrong$
MFS		4.26	4.3				
F_1	no	3.35	4.69	F_4	no	3.28	4.86
F_2	no	3.65	4.55	F_4	yes	4.17	4.40
F_2	yes	4.18	4.40	F_5	no	3.34	4.82
F_3	no	3.33	4.81	F_7	no	3.21	4.70
F_8	no	3.43	4.61	F_9	yes	4.17	4.41
F_{11}	no	3.9	4.36	F_{13}	yes	4.87	5.54

Table 2. Polysemy of adjectives. F: formula; fc: frequency correction; $Right$: average polisemy of adjectives for correctly disambiguated nouns; $Wrong$: average polisemy of adjectives for incorrectly disambiguated nouns.

The results showed in Table 2 demonstrate that the less polysemic is the adjective, the higher will be the probability of selecting the right sense. However, frequency-corrected formulae tend to be less subject to the polysemy of the adjective, obtaining values for the polysemy of the adjective closer to the values obtained with the MFS heuristics.

Finally, we investigated the possibility of using the same approach to perform the disambiguation of adjectives (i.e., using the web to look for $f_S(a_{ik}, W)$, where a_{ik} is the i-th synonym of the k-th sense of adjective a). We used an equivalent formula to F_{13} (the best formula for the disambiguation of nouns), but we obtained poor results (21.3% precision).

6 Conclusions and Further Work

In the preliminary experiments with the web-based method based on adjective-noun pairs, for some of the formulae we obtained a better precision than the baseline but a low recall. We believe that the main reason why the results are not so good, as we expected, is that the majority of pairs is still ambiguous (i.e., the adjective is not enough to understand the meaning of the noun and a bigger context should be taken into account). Our study over the importance of the polisemy of the adjective in the disambiguation seems to confirm our intuition. For example, the pair *cold fire* is ambiguous, since it can be assigned both the sense corresponding to *cold passion* or the sense corresponding to *cold fire.*

The result analysis allowed us to understand that it should be better to use the web in order to integrate existing systems rather than use it stand-alone. Moreover, the unsupevised method based on conceptual density provided better results if the web was also used as lexical resource instead of the WordNet Domains. Finally, we detected some problems in the use over the web of WordNet synonyms and hypernyms, since they are composed of multi-word expressions rarely found in the web. Our further investigation directions will be to investigate the use of shallow parsers in order to determine an unambiguous context for the word to disambiguate, to use another ontology (maybe SUMO) in order to overcome the multi-word expression issue, and finally to do some experiments over adjectives and verbs. At the moment, we are also investigating how to acquire extra lexical information using machine-readable dictionaries available on the Web in order to enrich WordNet glosses [3].

Acknowledgments

We would like to thank R2D2 CICYT (TIC2003-07158-C04-03), ICT EU-India (ALA/95/23/2003/077-054) and CONACyT-Mexico (43990A-1) projects, as well as the Secretaría de Estado de Educación y Universidades de España, for partially supporting this work.

References

1. Brill, E.: Processing Natural Language Processing without Natural Language Processing. Lecture Notes in Computer Science, Vol. 2588. Springer-Verlag (2003) 360–369.
2. Buscaldi, D., Rosso, P., Masulli, F.: The upv-unige-CIAOSENSO WSD System. Senseval-3 Workshop, Association for Computational Linguistics (ACL-04). Barcelona, Spain (2004).
3. Buscaldi, D., Rosso, P., Masulli, F.: Integrating Conceptual Density with WordNet Domains and CALD Glosses for Noun Sense Disambiguation. Lecture Notes in Artificial Intelligence, Vol. 3230. Springer-Verlag (2004) 183–194.
4. Gonzalo, J., Verdejo, F., Chugar, I.: The Web as a Resource for WSD. In: 1st MEANING Workshop, Spain (2003).
5. Frakes, W., Baeza-Yates, R.: Information Retrieval, Data Structures and Algorithms. Prentice Hall (1992).
6. Lin, D.: An Information-Theoretic Definition of Similarity. In: Proc. of the 15th Int. Conf. on Machine Learning. Toronto, Canada (2003)
7. Magnini, B. and Cavaglià, G.: Integrating Subject Field Codes into WordNet. In: Proc. of LREC-2000, 2nd Int. Conf. on Language Resources and Evaluation. (2000) 1413–1418.
8. Mihalcea, R., Moldovan, D.I.: A Method for Word Sense Disambiguation of Unrestricted Text. In: Proc. of the 37th Annual Meeting of the Association for Computational Linguistics (ACL-99). Maryland, NY, U.S.A. (1999)
9. Rosso, P., Masulli, F., Buscaldi, D., Pla, F., Molina, A.: Automatic Noun Disambiguation. Lecture Notes in Computer Science, Vol. 2588. Springer-Verlag (2003) 273–276.
10. Wackerbauer, R., Witt, A., Atmanspacher, H., Kurths, J., Scheingraber, H.: A Comparative Classification of Complexity Measures. Chaos, Solitons and Fractals, 4: 133-173 (1994).