Adapting a synonym database to specific domains

Davide Turcato Fred Popowich Janine Toole
Dan Fass Devlan Nicholson Gordon Tisher
gavagai Technology Inc.
P.O. 374, 3495 Cambie Street, Vancouver, British Columbia, V5Z 4R3, Canada
and
Natural Language Laboratory, School of Computing Science, Simon Fraser University
8888 University Drive, Burnaby, British Columbia, V5A 1S6, Canada
{turk,popowich,toole,fass,devlan,gtisher}@{gavagai.net,cs.sfu.ca}

Abstract

This paper describes a method for adapting a general purpose synonym database, like WordNet, to a specific domain, where only a subset of the synonymy relations defined in the general database hold. The method adopts an eliminative approach, based on incrementally pruning the original database. The method is based on a preliminary manual pruning phase and an algorithm for automatically pruning the database. This method has been implemented and used for an Information Retrieval system in the aviation domain.

1 Introduction

Synonyms can be an important resource for Information Retrieval (IR) applications, and attempts have been made at using them to expand query terms (Voorhees, 1998). In expanding query terms, overgeneration is as much of a problem as incompleteness or lack of synonym resources. Precision can dramatically drop because of false hits due to incorrect synonymy relations. This problem is particularly felt when IR is applied to documents in specific technical domains. In such cases, the synonymy relations that hold in the specific domain are only a restricted portion of the synonymy relations holding for a given language at large. For instance, a set of synonyms like

(1) {*cocaine, cocain, coke, snow, C*}

valid for English, would be detrimental in a specific domain like weather reports, where both *snow* and *C* (for *Celsius*) occur very frequently, but never as synonyms of each other.

We describe a method for creating a domain specific synonym database from a general purpose one. We use WordNet (Fellbaum, 1998) as our initial database, and we draw evidence from a domain specific corpus about what synonymy relations hold in the domain.

Our task has obvious relations to word sense disambiguation (Sanderson, 1997) (Leacock et al., 1998), since both tasks are based on identifying senses of ambiguous words in a text. However, the two tasks are quite distinct. In word sense disambiguation, a set of candidate senses for a given word is checked against each occurrence of the relevant word in a text, and a single candidate sense is selected for each occurrence of the word. In our synonym specialization task a set of candidate senses for a given word is checked against an entire corpus, and a subset of candidate senses is selected. Although the latter task could be reduced to the former (by disambiguating all occurrences of a word in a test and taking the union of the selected senses), alternative approaches could also be used. In a specific domain, where words can be expected to be monosemous to a large extent, synonym pruning can be an effective alternative (or a complement) to word sense disambiguation.

From a different perspective, our task is also related to the task of assigning Subject Field Codes (SFC) to a terminological resource, as done by Magnini and Cavaglià (2000) for WordNet.

Assuming that a specific domain corresponds to a single SFC (or a restricted set of SFCs, at most), the difference between SFC assignment and our task is that the former assigns one of many possible values to a given synset (one of all possible SFCs), while the latter assigns one of two possible values (the words belongs or does not belong to the SFC representing the domain). In other words, SFC assignment is a classification task, while ours can be seen as either a filtering or ranking task.

Adopting a filtering/ranking perspective makes apparent that the synonym pruning task can also be seen as an *eliminative* process, and as such it can be performed *incrementally.* In the following section we will show how such characteristics have been exploited in performing the task.

In section 2 we describe the pruning methodology, while section 3 provides a practical example from a specific domain. Conclusions are offered in section 4.

2 Methodology

2.1 Outline

The synonym pruning task aims at improving both the accuracy and the speed of a synonym database. In order to set the terms of the problem, we find it useful to partition the set of synonymy relations defined in WordNet into three classes:

1. Relations *irrelevant* to the specific domain (e.g. relations involving words that seldom or never appear in the specific domain)

2. Relations that are *relevant* but *incorrect* in the specific domain (e.g. the synonymy of two words that do appear in the specific domain, but are only synonyms in a sense irrelevant to the specific domain);

3. Relations that are *relevant* and *correct* in the specific domain.

The creation of a domain specific database aims at removing relations in the first two classes (to improve speed and accuracy, respectively) and including only relations in the third class.

The overall goal of the described method is to inspect all synonymy relations in WordNet and classify each of them into one of the three aforementioned classes. We define a *synonymy relation* as a binary relation between two synonym terms (with respect to a particular sense). Therefore, a WordNet synset containing n terms defines $\sum_{k=1}^{n-1} k$ synonym relations. The assignment of a synonymy relation to a class is based on evidence drawn from a domain specific corpus. We use a tagged and lemmatized corpus for this purpose. Accordingly, all frequencies used in the rest of the paper are to be intended as frequencies of $\langle lemma, tag \rangle$ pairs.

The pruning process is carried out in three steps: (i) manual pruning; (ii) automatic pruning; (iii) optimization. The first two steps focus on incrementally eliminating incorrect synonyms, while the third step focuses on removing irrelevant synonyms. The three steps are described in the following sections.

2.2 Manual pruning

Different synonymy relations have a different impact on the behavior of the application in which they are used, depending on how frequently each synonymy relation is used. Relations involving words frequently appearing in either queries or corpora have a much higher impact (either positive or negative) than relations involving rarely occurring words. E.g. the synonymy between *snow* and *C* has a higher impact on the weather report domain (or the aviation domain, discussed in this paper) than the synonymy relation between *cocaine* and *coke.* Consequently, the precision of a synonym database obviously depends much more on frequently used relations than on rarely used ones. Another important consideration is that judging the correctness of a given synonymy relation in a given domain is often an elusive issue: besides clearcut cases, there is a large gray area where judgments may not be trivial even for humans evaluators. E.g. given the following three senses of the noun *approach*

(2) a. {*approach, approach path, glide path, glide slope*}
(the final path followed by an aircraft as it is landing)

b. {*approach, approach shot*}
(a relatively short golf shot intended to put the ball onto the putting green)

c. {*access, approach*}
(a way of entering or leaving)

it would be easy to judge the first and second senses respectively relevant and irrelevant to the aviation domain, but the evaluation of the third sense would be fuzzier.

The combination of the two remarks above induced us to consider a manual pruning phase for the terms of highest 'weight' as a good investment of human effort, in terms of rate between the achieved increase in precision and the amount of work involved. A second reason for performing an initial manual pruning is that its outcome can be used as a reliable test set against which automatic pruning algorithms can be tested.

Based on such considerations, we included a manual phase in the pruning process, consisting of two steps: (i) the ranking of synonymy relations in terms of their weight in the specific domain; (ii) the actual evaluation of the correctness of the top ranking synonymy relation, by human evaluators.

2.2.1 Ranking of synonymy relations

The goal of ranking synonymy relations is to associate them with a score that estimates how often a synonymy relation is likely to be used in the specific domain. The input database is sorted by the assigned scores, and the top ranking words are checked for manual pruning. Only terms appearing in the domain specific corpus are considered at this stage. In this way the benefit of manual pruning is maximized. Ranking is based on three sorting criteria, listed below in order of priority.

Criterion 1. Since a term that does appear in the domain corpus must have at least one valid sense in the specific domain, words with only one sense are not good candidates for pruning (under the assumption of completeness of the synonym database). Therefore *polysemous terms* are prioritized over monosemous terms.

Criterion 2. The second and third sorting criteria are similar, the only difference being that the second criterion assumes the existence of some inventory of relevant queries (a term list, a collection of previous queries, etc.). If such an inventory is not available, the second sorting criterion can be omitted. If the inventory is available, it is used to check which synonymy relations are actually to be used in queries to the domain corpus. Given a pair $\langle t_i, t_j \rangle$ of synonym terms, a score (which we name *scoreCQ*) is assigned to their synonymy relation, according to the following formula:

(3) $$scoreCQ_{i,j} = (fcorpus_i * fquery_j) + (fcorpus_j * fquery_i)$$

where $fcorpus_n$ and $fquery_n$ are, respectively, the frequencies of a term in the domain corpus and in the inventory of query terms. The above formula aims at estimating how often a given synonymy relation is likely to be actually used. In particular, each half of the formula estimates how often a given term in the corpus is likely to be matched as a synonym of a given term in a query. Consider, e.g., the following situation (taken form the aviation domain discussed in section 3.1):

(4) $$fcorpus_{snow} = 3042$$
$$fquery_{snow} = 2$$
$$fcorpus_C = 9168$$
$$fquery_C = 0$$

It is estimated that C would be matched 18336 times as a synonym for *snow* (i.e 9168 * 2), while *snow* would never be matched as a synonym for C, because C never occurs as a query term. Therefore $scoreCQ_{snow,C}$ is 18336 (i.e. 18336 + 0).

Then, for each polysemous term i and synset s such that $i \in s$, the following score is computed:

Table 1: Frequencies of sample synset terms.

j	$fcorpus_j$	$fquery_j$
cocaine	1	0
cocain	0	0
coke	8	0
C	9168	0

(5) $scorePolyCQ_{i,s} = \sum\{scoreCQ_{i,j} | j \in s \wedge i \neq j\}$

E.g., if S is the synset in (1), then $scorePolyCQ_{snow,S}$ is the sum of $scoreCQ_{snow,cocaine}$, $scoreCQ_{snow,cocain}$, $scoreCQ_{snow,coke}$ and $scoreCQ_{snow,C}$. Given the data in Table 1 (taken again from our aviation domain) the following $scoreCQ$ would result:

(6)
$$scoreCQ_{snow,cocaine} = 2$$
$$scoreCQ_{snow,cocain} = 0$$
$$scoreCQ_{snow,coke} = 16$$
$$scoreCQ_{snow,C} = 18336$$

Therefore, $scorePolyCQ_{snow,S}$ would equal 18354.

The final score assigned to each polysemous term t_i is the highest $scorePolyCQ_{i,s}$. For *snow*, which has the following three senses

(7) a. {*cocaine, cocaine, coke, C, snow*}
(a narcotic (alkaloid) extracted from coca leaves)

b. {*snow*}
(a layer of snowflakes (white crystals of frozen water) covering the ground)

c. {*snow,snowfall*}
(precipitation falling from clouds in the form of ice crystals)

the highest score would be the one computed above.

Criterion 3. The third criterion assigns a score in terms of domain corpus frequency alone. It is used to further rank terms that do not occur in the query term inventory (or when no query term inventory is available). It is computed in the same way as the previous score, with the only difference that a value of 1 is conventionally assumed for $fquery$ (the frequency of a term in the inventory of query terms).

2.2.2 Correctness evaluation

All the synsets containing the top ranking terms, according to the hierarchy of criteria described above, are manually checked for pruning. For each term, all the synsets containing the term are clustered together and presented to a human operator, who examines each $\langle term, synset\rangle$ pair and answers the question: does the term belong to the synset in the specific domain? Evidence about the answer is drawn from relevant examples automatically extracted from the domain specific corpus. E.g., following up on our example in the previous section, the operator would be presented with the word *snow* associated with each of the synsets in (7) and would have to provide a yes/no answer for each of them. In the specific case, the answer would be likely to be 'no' for (7a) and 'yes' for (7b) and (7c).

The evaluator is presented with all the synsets involving a relevant term (even those that did not rank high in terms of *scorePolyCQ*) in order to apply a contrastive approach. It might well be the case that the correct sense for a given term is one for which the term has no synonyms at all (e.g. 7b in the example), therefore all synsets for a given term need to be presented to the evaluator in order to make an informed choice. The evaluator provides a yes/no answer for all the $\langle term, synset\rangle$ he/she is presented with (with some exceptions, as explained in section 3.1).

2.3 Automatic pruning

The automatic pruning task is analogous to manual pruning in two respects: (i) its input is the set of synonymy relations involving WordNet polysemous words appearing in the domain specific corpus; (ii) it is performed by examining all $\langle term, synset\rangle$ input pairs and answering the question: does the term belong to the synset in the specific domain? However, while the manual pruning task was regarded as a filtering task, where a human evaluator assigns a boolean value to each pruning candidate, the automatic pruning task can be more conveniently regarded as a ranking task, where all the pruning candidates are assigned a score, measuring how appropriate a given sense is for a given word, in the domain at hand. The actual pruning is left as a subsequent step. Different pruning thresholds can be applied to the ranked list, based on different considerations (e.g. depending on whether a stronger emphasis is put on the precision or the recall of the resulting database). The score is based on the frequencies of both words in the synset (except the word under consideration) and words in the sense gloss. We also remove from the gloss all words belonging to a stoplist (a stoplist provided with WordNet was used for this purpose). The following scoring formula is used:

$$(8)\qquad \begin{aligned}&(average\_synset\_frequency/\\&synset\_cardinality^{k}) +\\&(average\_gloss\_frequency/\\&gloss\_cardinality^{k})\end{aligned}$$

Note that the synset cardinality does not include the word under consideration, reflecting the fact the word's frequency is not used in calculating the score. Therefore a synset only containing the word under consideration and no synonyms is assigned cardinality 0.

The goal is to identify $\langle term, sense\rangle$ pairs not pertaining to the domain. For this reason we tend to assign high scores to candidates for which we do not have enough evidence about their inappropriateness. This is why average frequencies are divided by some factor which is function of the number of averaged frequencies, in order to increase the scores based on little evidence (i.e. fewer averaged numbers). In the sample application described in section 3 the value of k was set to 2. For analogous reasons, we conventionally assign a very high score to candidates for which we have no evidence (i.e. no words in both the synset and the gloss). If either the synset or the gloss is empty, we conventionally double the score for the gloss or the synset, respectively. We note at this point that our final ranking list are sorted in reverse order with respect to the assigned scores, since we are focusing on removing incorrect items. At the top of the list are the items that receive the lowest score, i.e. that are more likely to be incorrect $\langle term, sense\rangle$ associations for our domain (thus being the best candidates to be pruned out).

Table 2 shows the ranking of the senses for the word C in the aviation domain. In the table, each term is followed by its corpus frequency, separated by a slash. From each synset the word C itself has been removed, as well as the gloss words found in the stop list. Therefore, the table only contains the words that contribute to the calculation of the sense's score. E.g. the score for the first sense in the list is obtained from the following expression:

$$(9)\qquad \begin{aligned}&((0+57)/2/2^2) +\\&((8{+}0{+}0{+}198{+}9559{+}0{+}1298)/7/7^2)\end{aligned}$$

The third sense in the list exemplifies the case of an empty synset (i.e. a synset originally containing only the word under consideration). In this case the score obtained from the gloss is doubled. Note that the obviously incorrect sense of C as a narcotic is in the middle of the list. This is due to a tagging problem, as the word *leaves* in the gloss was tagged as verb instead of noun. Therefore it was assigned a very high frequency, as the verb *leave*, unlike the noun *leaf*, is very common in the aviation domain. The last sense in the list also requires a brief explanation. The original word in the gloss was *10S*. However, the pre-processor that was used before tagging the glosses recognized *S* as an abbreviation for *South* and expanded the term accordingly. It so happens that both words *10* and *South* are very frequent in the aviation corpus we used, therefore the sense was assigned a high score.

2.4 Optimization

The aim of this phase is to improve the access speed to the synonym database, by removing all information that is not likely to be used. The main idea is to minimize the size of the

Table 2: Ranking of synsets containing the word C

Score		Frequencies
39.37	synset:	ATOMIC_NUMBER_6/0, CARBON/57
	gloss:	ABUNDANT/8, NONMETALLIC/0, TETRAVALENT/0, ELEMENT/198 OCCUR/9559, ALLOTROPIC/0, FORM/1298
62.75	synset:	AMPERE-SECOND/0, COULOMB/0
	gloss:	UNIT/3378, ELECTRICAL/2373, CHARGE/523, EQUAL/153 AMOUNT/1634, CHARGE/523, TRANSFER/480, CURRENT/242, 1/37106 AMPERE/4, 1/37106
224.28	synset:	∅
	gloss:	GENERAL-PURPOSE/0, PROGRAMING/0, LANGUAGE/445, CLOSELY/841 ASSOCIATE/543, UNIX/0, OPERATE/5726, SYSTEM/49863
241.69	synset:	COCAIN/0, COCAINE/1, COKE/8, SNOW/3042
	gloss:	NARCOTIC/1, ALKALOID/0, EXTRACT/31, COCA/1, LEAVE/24220
585.17	synset:	LIGHT_SPEED/1, SPEED_OF_LIGHT/0
	gloss:	SPEED/14665, LIGHT/22481, TRAVEL/105, VACUUM/192
743.28	synset:	DEGREE_CELSIUS/24, DEGREE_CENTIGRADE/28
	gloss:	DEGREE/43617, CENTIGRADE/34, SCALE/540, TEMPERATURE/2963
1053.43	synset:	100/0, CENTRED/0, CENTURY/31, HUNDRED/0, ONE_C/0
	gloss:	TEN/73, 10/16150, SOUTH/12213

database in such a way that the database behavior remains unchanged. Two operations are performed at the stage: (i) a simple *relevance test* to remove irrelevant terms (i.e. terms not pertaining to the domain at hand); (ii) a redundancy check, to remove information that, although perhaps relevant, does not affect the database behavior.

2.4.1 Relevance test

Terms not appearing in the domain corpus are considered not relevant to the specific domain and removed from the synonym database. The rationale underlying this step is to remove from the synonym database synonymy relations that are never going to be used in the specific domain. In this way the efficiency of the module can be increased, by reducing the size of the database and the number of searches performed (synonyms that are known to never appear are not searched for), without affecting the system's matching accuracy. E.g., the synset in (10a) would be reduced to the synset in (10b).

(10) a. AMPERE-SECOND/0, COULOMB/0, C/9168

b. C/9168

2.4.2 Redundancy check

The final step is the removal of redundant synsets, possibly as a consequence of the previous pruning steps. Specifically, the following synsets are removed:

- Synsets containing a single term (although the associated sense might be a valid one for that term, in the specific domain).
- Duplicate synsets, i.e. identical (in terms of synset elements) to some other synset not being removed (the choice of the only synset to be preserved is arbitrary).

E.g., the synset in (10b) would be finally removed at this stage.

3 Sample application

The described methodology was applied to the aviation domain. We used the Aviation

Safety Information System (ASRS) corpus (`http://asrs.arc.nasa.gov/`) as our aviation specific corpus. The resulting domain-specific database is being used in an IR application that retrieves documents relevant to user defined queries, expressed as phrase patterns, and identifies portions of text that are instances of the relevant phrase patterns. The application makes use of Natural Language Processing (NLP) techniques (tagging and partial parsing) to annotate documents. User defined queries are matched against such annotated corpora. Synonyms are used to expand occurrences of specific words in such queries. In the following two sections we describe how the pruning process was performed and provide some results.

3.1 Adapting Wordnet to the aviation domain

A vocabulary of relevant query terms was made available by a user of our IR application and was used in our ranking of synonymy relations. Manual pruning was performed on the 1000 top ranking terms, with which 6565 synsets were associated overall. The manual pruning task was split between two human evaluators. The evaluators were programmers members of our staff. They were English native speakers who had acquaintance with our IR application and with the goals of the manual pruning process, but no specific training or background on lexicographic or WordNet-related tasks. For each of the 1000 terms, the evaluators were provided with a sample of 100 (at most) sentences where the relevant word occurred in the ASRS corpus. 100 of the 1000 manually checked clusters (i.e. groups of synsets referring to the same head term) were submitted to both evaluators (576 synsets overall), in order to check the rate of agreement of their evaluations. The evaluators were allowed to leave synsets unanswered, when the synsets only contained the head term (and at least one other synset in the cluster had been deemed correct). Leaving out the cases when one or both evaluators skipped the answer, there remained 418 synsets for which both answered. There was agreement in 315 cases (75%) and disagreement in 103 cases (25%). A sample of senses on which the evaluators disagreed is shown in (11). In each case, the term being evaluated is the first in the synset.

(11) a. {*about, around*}
(in the area or vicinity)

b. {*accept, admit, take, take on*}
(admit into a group or community)

c. {*accept, consent, go for*}
(give an affirmative reply to)

d. {*accept, swallow*}
(tolerate or accommodate oneself to)

e. {*accept, take*}
(be designed to hold or take)

f. {*accomplished, effected, established*}
(settled securely and unconditionally)

g. {*acknowledge, know, recognize*}
(discern)

h. {*act, cognitive operation, cognitive process, operation, process*}
(the performance of some composite cognitive activity)

i. {*act, act as, play*}
(pretend to have certain qualities or state of mind)

j. {*action, activeness, activity*}
(the state of being active)

k. {*action, activity, natural action, natural process*}
(a process existing in or produced by nature (rather than by the intent of human beings))

It should be noted that the 'yes' and 'no' answers were not evenly distributed between the evaluators. In 80% of the cases of disagreement, it was evaluator A answering 'yes' and evaluator B answering 'no'. This seems to suggest than one of the reasons for disagreement was a different degree of strictness in evaluating. Since the evaluators matched a sense against an entire corpus (represented by a sample of occurrences), one common situation may have been that a sense did occur, but very rarely. Therefore, the evaluators may have applied different criteria in judging how many occurrences were needed to deem a sense correct. This discrepancy, of course, may compound with the fact that the differences among WordNet senses can sometimes be very subtle.

Automatic pruning was performed on the entire WordNet database, regardless of whether candidates had already been manually checked or not. This was done for testing purposes, in order to check the results of automatic pruning against the test set obtained from manual pruning. Besides associating ASRS frequencies with all words in synsets and glosses, we also computed frequencies for collocations (i.e. multi-word terms) appearing in synsets. The input to automatic pruning was constituted by 10352 polysemous terms appearing at least once in ASRS the corpus. Such terms correspond to 37494 $\langle term, synset \rangle$ pairs. Therefore, the latter was the actual number of pruning candidates that were ranked.

The check of WordNet senses against ASRS senses was only done unidirectionally, i.e. we only checked whether WordNet senses were attested in ASRS. Although it would be interesting to see how often the appropriate, domain-specific senses were absent from WordNet, no check of this kind was done. We took the simplifying assumption that WordNet be complete, thus aiming at assigning at least one WordNet sense to each term that appeared in both WordNet and ASRS.

3.2 Results

In order to test the automatic pruning performance, we ran the ranking procedure on a test set taken from the manually checked files. This file had been set apart and had not been used in the preliminary tests on the automatic pruning algorithm. The test set included 350 clusters, comprising 2300 candidates. 1643 candidates were actually assigned an evaluation during manual pruning. These were used for the test. We extracted the 1643 relevant items from our ranking list, then we incrementally computed precision and recall in terms of the items that had been manually checked by our human evaluators. The results are shown in figure 1. As an example of how this figure can be interpreted, taking into consideration the top 20% of the ranking list (along the X axis), an 80% precision (Y axis) means that 80% of the items encountered so far had been removed in manual pruning; a 27% recall (Y axis) means that 27% of the overall manually removed items have been encountered so far.

The automatic pruning task was intentionally framed as a ranking problem, in order to leave open the issue of what pruning threshold would be optimal. This same approach was taken in the IR application in which the pruning procedure was embedded. Users are given the option to set their own pruning threshold (depending on whether they focus more on precision or recall), by setting a value specifying what precision they require. Pruning is performed on the top section of the ranking list that guarantees the required precision, according to the correlation between precision and amount of pruning shown in figure 1.

A second test was designed to check whether there is a correlation between the levels of confidence of automatic and manual pruning. For this purpose we used the file that had been manually checked by both human evaluators. We took into account the candidates that had been removed by at least one evaluator: the candidates that were removed by both evaluators were deemed to have a high level of confidence, while those removed by only one evaluator were deemed to have a lower level of confidence. Then we checked whether the two classes were equally distributed in the automatic pruning ranking list, or whether higher confidence candidates tended to be ranked higher than lower confidence ones. The results are shown in figure 2, where the automatic pruning recall for each class is shown. For any given portion of the ranking list higher confidence candidates (solid lines) have a significantly higher recall than lower confidence candidates (dot-

Table 3: WordNet optimization results.

DB	Synsets	Word-senses
Full WN	99,642	174,008
Reduced WN	9,441	23,368

ted line).

Finally, table 3 shows the result of applying the described optimization techniques alone, i.e. without any prior pruning, with respect to the ASRS corpus. The table shows how many synsets and how many word-senses are contained in the full Wordnet database and in its optimized version. Note that such reduction does not involve any loss of accuracy.

4 Conclusions

There is a need for automatically or semi-automatically adapting NLP components to specific domain, if such components are to be effectively used in IR applications without involving labor-intensive manual adaptation. A key part of adapting NLP components to specific domains is the adaptation of their lexical and terminological resources. It may often be the case that a consistent section of a general purpose terminological resource is irrelevant to a specific domain, thus involving an unnecessary amount of ambiguity that affects both the accuracy and efficiency of the overall NLP component. In this paper we have proposed a method for adapting a general purpose synonym database to a specific domain.

Evaluating the performance of the proposed pruning method is not a straightforward task, since there are no other results available on a similar task, to the best of our knowledge. However, a comparison between the results of manual and automatic pruning provides some useful hints. In particular:

- The discrepancy between the evaluation of human operators shows that the task is elusive even for humans (the value of the agreement evaluation statistic κ for our human evaluators was 0.5);
- however, the correlation between the level of confidence of human evaluations and scores assigned by the automatic pruning procedure shows that the automatic pruning algorithm captures some significant aspect of the problem.

Although there is probably room for improving the automatic pruning performance, the preliminary results show that the current approach is pointing in the right direction.

References

Christiane Fellbaum, editor. 1998. *Wordnet: An Electronic Lexical Database.* MIT Press Books.

Claudia Leacock, Martin Chodorow, and George A. Miller. 1998. Using corpus statistics and WordNet relations for sense identification. *Computational Linguistics*, 24(1):147–165.

Bernardo Magnini and Gabriela Cavaglià. 2000. Integrating Subject Field Codes into WordNet. In Maria Gavrilidou, George Carayannis, Stella Markantonatou, Stelios Piperidis, and Gregory Stainhaouer, editors, *Proceedings of the Second International Conference on Language Resources and Evaluation (LREC-2000)*, pages 1413–1418, Athens, Greece.

Mark Sanderson. 1997. *Word Sense Disambiguation and Information Retrieval.* Ph.D. thesis, Department of Computing Science at the University of Glasgow, Glasgow G12. Technical Report (TR-1997-7).

Ellen M. Voorhees. 1998. Using WordNet for text retrieval. In Fellbaum (Fellbaum, 1998), chapter 12, pages 285–303.

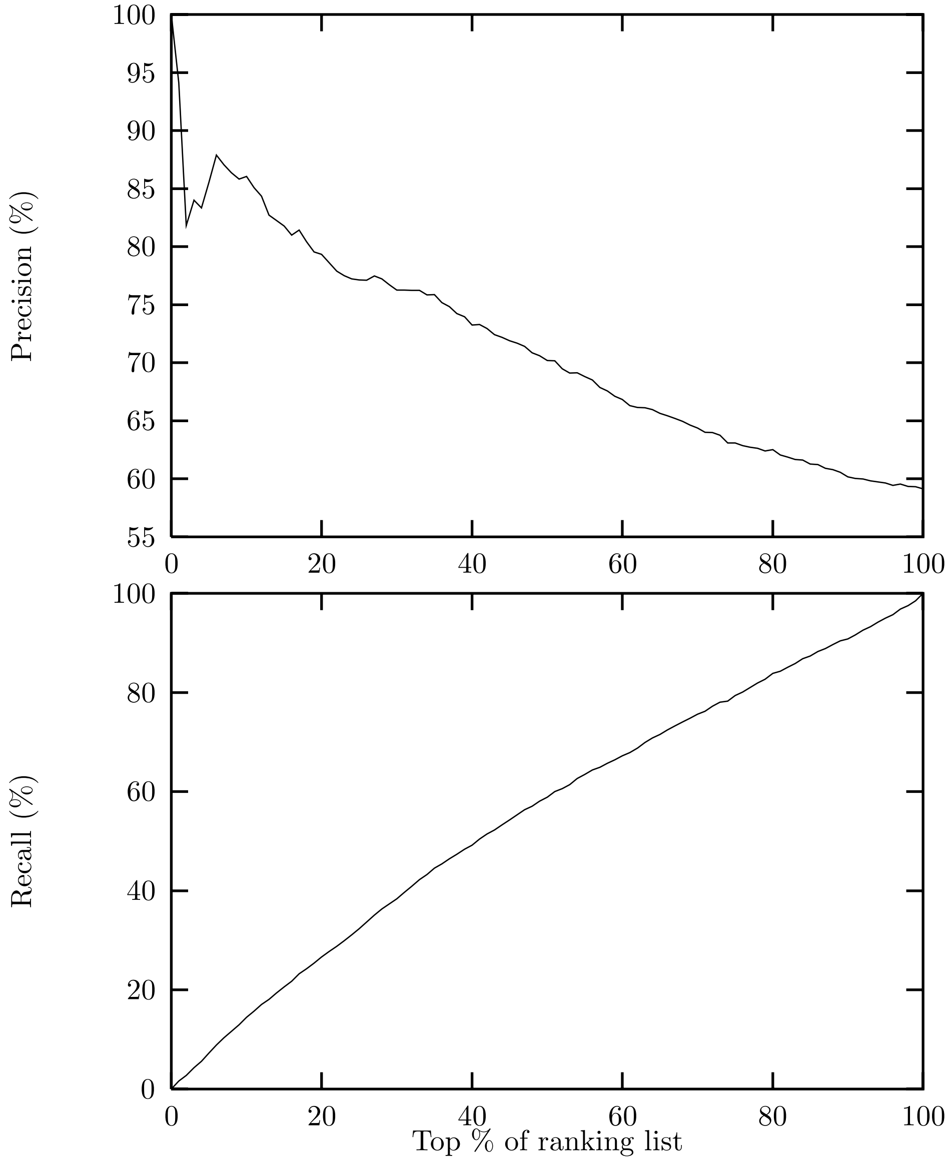

Figure 1: Precision and recall of automatic pruning

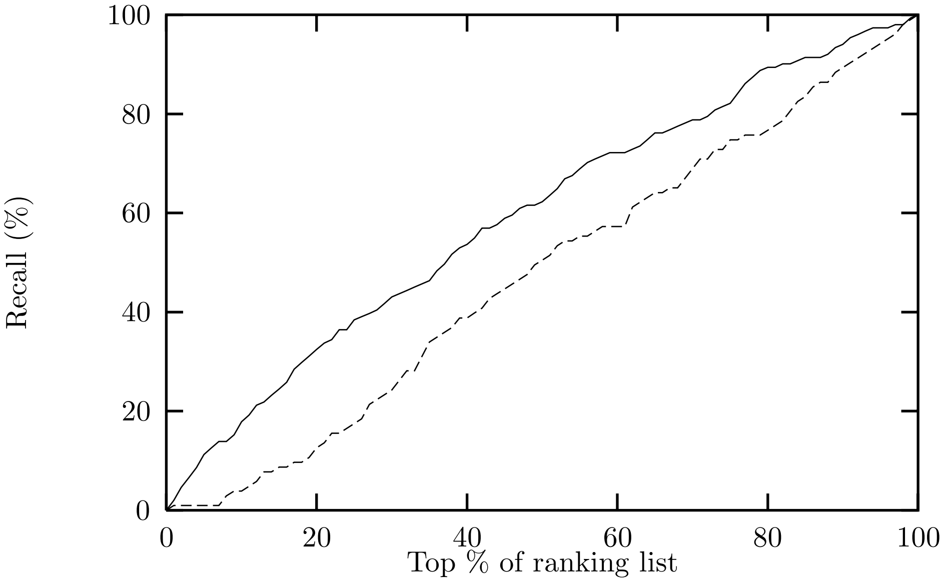

Figure 2: A recall comparison for different confidence rates