Aggressive Morphology and Lexical Relations for Query Expansion

W. A. Woods Stephen Green Paul Martin
Ann Houston
Sun Microsystems Laboratories
1 Network Drive
Burlington, MA 01803
{William.Woods,Stephen.Green,Paul.Martin,Ann.Houston}@east.sun.com

1 Introduction

Our submission to TREC this year is based on a combination of systems. The first is the conceptual indexing and retrieval system that was developed at Sun Microsystems Laboratories (Woods et al., 2000a; Woods et al., 2000b). The second is the MultiText system developed at the University of Waterloo (Clarke et al., 2000; Cormack et al., 2000).

The conceptual indexing system was designed to help people find specific answers to specific questions in unrestricted text. It uses a combination of syntactic, semantic, and morphological knowledge, together with taxonomic subsumption techniques, to address differences in terminology between a user's queries and the material that may answer them. At indexing time, the system builds a conceptual taxonomy of all the words and phrases in the indexed material. This taxonomy is based on the morphological structure of words, the syntactic structure of phrases, and semantic relations between meanings of words that it knows in its lexicon.

It was not, however, designed as a question answering system. Our results from last year, while encouraging, showed that we needed more work in the area of question analysis (i.e., "What would constitute an answer to this question?") and answer determination (i.e., "Does this retrieved passage actually answer the question?") to support our relaxation ranking passage retrieval algorithm.

After conversations with the researchers at the University of Waterloo, we decided to submit a run where we would provide front-end processing consisting of query formulation and query expansion using our automatically derived taxonomy and Waterloo would provide the back-end processing via their MultiText passage retrieval system and their answer selection component. The result is a direct comparison of two question answering systems that differ only in the query formulation component.

2 The Conceptual Taxonomy

As we said earlier, Sun's conceptual indexing system builds a taxonomy of all the words (and possibly phrases) that are encountered during indexing. This taxonomy is built around the generality relationships between terms. More general terms are said to *subsume* more specific terms. There are three sources of knowledge that are used to build the taxonomy for a set of documents: syntactic structure of phrases, semantic subsumption relationships between words and word senses, and morphological structure and relationships between words. For this experiment, we used only the latter two sources of knowledge to expand terms in the input questions:

1. Semantic subsumption axioms. These are encoded in the lexicon used by the indexing system. The largest base lexicon currently used by the system system contains semantic subsumption information for something in excess of 15,000 words. This information consists of basic "kind of" and "instance of" information such as the fact that *book* is a kind of *document* and *washing* is a kind of *cleaning*.

2. Morphological rules. The current morphology component consists of approximately 1200 knowledge-based morphological rules. These rules cover prefixes, suffixes, lexical compounds, as well as some special cases (e.g., phone numbers).

The conceptual taxonomy for this experiment is constructed automatically as a byproduct of indexing the TREC material with the conceptual indexing system. As each word is encountered during the indexing process, it is looked up in the system's lexicon and if not found, it is given an entry whose content is determined by the morphological analysis component. This entry is used for any subsequent occurrences of the same term. The rules in the morphological analysis system can infer syntactic parts-of-speech for the word, morphological relationships to other words, and sometimes even semantic relationships to other words.

After the conceptual indexer has assured that the word has a lexical entry, the word is entered into the conceptual taxonomy if it is not already there. Then all of its root words and any more general concepts that are listed in this lexical entry are also entered into the conceptual taxonomy in the same way, and this word is recorded as being subsumed by those words. This process is carried on recursively so that a word's parents' parents are recorded and so on, until there are no more indirect parents that are not already in the taxonomy.

Thus the conceptual taxonomy includes all of the terms found in the indexed material, plus all of the more general terms that are known in its lexicon or inferred by morphological rules. Furthermore, the conceptual taxonomy contains only words that were induced by this process.

2.1 Aggressive Morphology

The morphological analysis component of this system makes use of a large set of morphological rules that can recognize and analyze words that are derived and inflected forms of known words, as well as words that appear to be derived or inflected forms of unknown words. They can also make plausible inferences about the syntactic categories of unknown words that do not appear to be derived from other words.

The morphological analysis system considers both prefixes and suffixes and their interaction, and it also recognizes and analyzes lexical compounds formed from concatenating known words. For example, in the TREC-10 collection, "pointy," "repoint," "repointing," and "standpoint" were analyzed as forms of point."

The morphological analysis system makes use of different kinds of morphological rules, applied in a preferred order to words that are not already in the lexicon. Generally, the rules are ordered in decreasing order of specificity, confidence and likelihood. Very specific tests are applied in Step 1 to identify and deal with "words" that are not ordinary sequences of alphabetic characters. These include numbers, alphanumeric sequences, and expressions involving special characters. Failing this, an ordered sequence of suffix rules is applied in Step 2 in a first pass that will allow a match only if the proposed root word is "known." The same list of rules will be applied later in a second pass without this known-root condition if an earlier analysis does not succeed.

If no phase-one suffix rules apply, prefix rules are tried in Step 3 to see if an interpretation of this word as a prefix combined with some other "known" word is possible. Failing this, a set of lexical compound rules is tried, in Step 4, to see if the word is interpretable as a compound of two or more words, and failing that, lists of first and last names of people and names of cities are checked in Step 5. All of steps 3–5 are considered more reliable if they succeed than a phase-two pass of the suffix rules without any restriction to known roots that comes in Step 6. This ordering allows prefixes and compounding to be tried before less confident suffix analyses are attempted, and avoids applying weak suffix analyses to known names.

Lexical compound rules are called by a specialized interpreter that looks for places to divide a word into two pieces of sufficient size. The points of potential decomposition are searched from right to left, and the first such point that has an interpretation is taken, with the following exception: The morph compound analyzer checks for special cases where, for example, the first word is plural and ends in an *s*, but there is an alternative segmentation in which the singular of the first word is followed by a word starting with the *s*. In such cases, the decomposition using the singular first word is preferred over the one using the plural. For example, the word *minesweeper* will be analyzed as *mine+sweeper* rather than *mines+weeper*.

2.1.1 Interaction of rules with semantic axioms

It is useful to do full morphology on unknown words and to know morphological relationships for words in the lexicon in order to make connections between derived forms of words and semantic subsumption facts that may be known about their roots. For example, destruction may link morphologically to destroy, which then links semantically to damage. A simple stemming technique would not be able to find such connections (unless all the semantic axioms were similarly stemmed, a process that would result in many false subsumption paths in the taxonomy, due to the kinds of noise and errors that result from stemming algorithms).

3 Query Formulation and Term Expansion

Because the taxonomies that the conceptual indexing system builds are specific to a particular document collection, the first step of our query processing was to re-index the TREC Question Answering collection using the latest revision of the conceptual indexer. We then ran the queries through a modified version of the query formulation component of the system we used for TREC-9. The reformulated queries were then passed to the term expansion system.

The query formulation component is based on the pilot version of our conceptual indexing system. The query formulator interprets the question words and the format of the question to determine the desired answer type. It also either replaces the question word in the query with the desired answer type or simply removes it from the request. In addition, the query formulation component will generalize some terms (e.g., *high* for *tall*), substitute base forms for inflected forms (e.g., *principle* for *principles*), and drop some "noise" terms.

The query formulation component that we used this year differs from the one used for TREC-9 in small ways. First, we did not limit the number of terms in the reformulated query as we did last year. This limitation was due to a limitation of our passage retrieval system which is not a problem for the MultiText system. Second, we fixed a limitation to allow a query term to be expanded from each of its roots when there was more than one root for that term (e.g., saw from root see as well as root saw). Finally, we fixed a bug that generated incorrect answer types for certain question forms.

During a typical query run with the relaxation ranking passage retrieval system, a query term is expanded to include all terms that are subsumed by that term. In some cases the expansion can be quite dramatic. For example, in the AP sub-collection of the TREC QA collection, the query term *person* expands to more than 17,000 other terms. These terms include morphological variations such as *people* and *persons* as well as semantic variations such as *blacksmith* and *lawyer*. This set of terms reaches 25 levels deep into the conceptual taxonomy.

Although such a wide-ranging expansion may seem counter-intuitive, there doesn't seem to be any *a priori* way to determine a reasonable cutoff level. We can decide on using a small integer, say 2, but this may preclude useful expansions such as *counterrevolutionary*, which may be crucial in finding just the right document.

In the end, we decided that we would cut the expansion off after the first level of expansion, since this would get most of the morphological expansions for the term and many of the semantic expansions. Even with this stringent criterion, the query term *person* still expands to more than 8,000 terms.

We originally intended to integrate the answer types from our query formulation stage with the answer types from the MultiText system, but that proved not to be possible in the time available, so we ended up providing only the selected query terms and their expansions to the MultiText back end.

The reformulated, expanded queries were passed to the MultiText system as a conjunction of disjunctions of each of the expanded terms.

4 Results

The results for our two submitted runs as well as the corresponding MultiText runs are shown in table 1. The MultiText system seems to have done better on its own than with the Sun query formulation and expansion engine as the front end. In part, this may be due to the lack of full integration between our front end and the MultiText back end, and in part is may be due to the selection of query terms from the original question that was done by the query formulation stage. It is interesting to note that there are a significant number of questions that each system answered that the other didn't. It is interesting to look at the cases where the combined Sun/MultiText system found answers to questions that were not found by the MultiText system alone, and vice versa.

	NIST Judgment	
Run	Strict	Lenient
Sun baseline (mtsuna1)	0.307	0.322
Sun with Web reinforcement (mtsuna0)	0.405	0.418
MultiText baseline (uwmta2)	0.346	0.365
MultiText with Web reinforcement (uwmta1)	0.434	0.457

Table 1: Results for the main QA task.

Run	Sun only	MultiText only	Sun and MultiText	Neither
Baseline	38	61	193	200
Web reinforcement	47	55	225	165
Intersection	26	32		

Table 2: Differences in answers found

Table 2 shows the number of questions for which answers were found by only one of the systems, the number of questions for which answers were found by both systems, and the number of questions for which neither system found an answer. The row labeled "Intersection" shows the number of questions that were found in both of the Sun runs and neither of the MultiText runs, and vice versa. For the rest of the discussion, we will focus on the 26 questions that were found in both of the Sun runs, and neither of the MultiText runs.

There are two ways that the query expansions can affect whether an answer is found or not. First, the expanded query may give the passage retrieval component enough information to retrieve a passage that would not be found with the unexpanded query. Second, the expanded query may give the answer selection component better information about what words would make up a useful answer to the query.

A variation of the second type appears to have occurred for question 910, "What metal has the highest melting point?" In both runs, the top passage retrieved by the MultiText system contained the correct answer, but it appears that the answer selection component determined only for the Sun runs that this document held an answer.

In this case, the Sun query formulation stage transformed the original question to the sequence of terms: (METAL HIGHEST MELT POINT), which expanded to:

```
(metal aluminium bimetal chrome copper coppers dimetal gunmetal
immetal intermetal lead leads lithium metal's metaled metaler metalic
metalist metalled metallic metallist metallize metallized metallizing
metallurgy metalor metals metalware nickel nickeled nickeling nickels
nonmetal nonmetals palladium polymetal rust rusts silver silvers
sliver slivered slivers tin tins)
(highest top)
(melt melted melter melting melts melty molten re-melt remelt smelt
smelted smelting smelts)
(point barb barbs breakpoint breakpoints checkpoint checkpoints
crosspoint cusp cusps depoint endpoint endpoints gunpoint interpoint
isopoint middle middles midpoint midpoints multipoint nib nibs
nonpoint outpoint outpointing outpoints point's pointed pointer
pointers pointeur pointful pointing pointless points pointy repoint
repointing standpoint standpoints subpoint tip tips viewpoint
viewpoints wellpoint)
```

This example also illustrates a potential negative interaction between our expansions and the MultiText answer selection strategy, since in this case the answer could well be one of the query expansion terms, and could then be rejected by the heuristic of looking for answers that are non-query terms.

Question 922, "Where is John Wayne airport", shows both effects. The sets of passages that were retrieved are completely disjoint. This query was only slightly expanded, and even though the answer was in passages in both sets, for the MultiText runs, the answer selection component seems to have focussed on other proper names than those in the query. We suspect that this is an effect of the removal of all query words before candidate selection. The query formulation stage transforms this to (JOHN WAYNE AIRPORT), which expands to:

```
(john dejohn demijohn john's johner johni johnie johny saint-john)

(wayne dewayne wayne's wayner waynes)

(airport airport's airporter airports multiairport)
```

With respect to questions that MultiText gets answers for that the joint Sun/MultiText system doesn't, a comparison of the two sets suggests that the joint system does better on questions with more complex descriptions (e.g., 1231 "What fruit is Melba sauce made from?") while the MultiText system does better alone on short direct questions (e.g., 1266 "What are pathogens?"). Another noticable pattern is that many of the questions for which MultiText does better alone contain plurals that the Sun question formulation stage generalizes to singulars (e.g., "Great Lakes" and "x-rays") where the plural is probably a better retrieval clue. These comparisons suggest that there is still a lot of tuning to be done to match the query formulation stage to the retrieval stage and answer selection stage, and that a switch between the two techniques based on the question type would do better than either one alone.

5 Conclusion and Future Work

We've only done a preliminary analysis of the results at this point. The results show that for some queries, the morphological and semantic expansions help, while for others (unfortunately somewhat more others) they degrade the results. This is typical of the impact of this kind of expansion on many retrieval techniques, with the notable exception of the penalty-based passage retrieval technique used in Sun's conceptual indexing system.

We haven't yet developed a full picture of how this aggressive expansion might be degrading the retrieved passages of the MultiText system. We have some hope that the MultiText system's passage retrieval method is fairly resistant to degradation, but we will have to further analyze the data to find out if that is the case, and if not, what can be done about it.

So far, the results above suggest a possible refinement to our front-end/back-end strategy. Since it seems clear that there are instances where the expanded queries aided more in the answer selection than in the passage selection, it would be interesting to try a run where the unexpanded queries are used for passage retrieval and the expanded queries are used during answer selection. Another obvious thing to try is a conditional system that uses the expansion technique for the longer more complex question types and avoids expansion for direct short questions, especially those that look like they are asking for the definition of a term.

References

(Clarke et al., 2000) C.L.A. Clarke, G.V. Cormack, D.I.E. Kisman, and T.R. Lynam. Question Answering by Passage Selection (MultiText Experiments for TREC-9). In *Proceedings of The Ninth Text REtrieval Conference (TREC 9)*. National Institute of Standards and Technology, 2000.

(Cormack et al., 2000) G. V. Cormack, C. L. A. Clarke, C. R. Palmer, and D. I. E. Kisman. Fast Automatic Passage Ranking (MultiText Experiments for TREC-8). In *Proceedings of The Eighth Text REtrieval Conference (TREC 8)*. National Institute of Standards and Technology, 2000.

(Woods et al., 2000a) W.A. Woods, Stephen Green, Paul Martin, and Ann Houston. Halfway to Question Answering. In *Proceedings of The Ninth Text REtrieval Conference (TREC 9)*. National Institute of Standards and Technology, 2000.

(Woods et al., 2000b) William A. Woods, Lawrence A. Bookman, Ann Houston, Robert J. Kuhns, Paul Martin, and Stephen Green. Linguistic Knowledge can Improve Information Retrieval. In *Sixth Annual Applied Natural Language Processing Conference*, pages 262–267. Association for Computational Linguistics, 2000.