The Use of WordNet Sense Tagging in FAQFinder

Steven L. Lytinen, Noriko Tomuro and Tom Repede
DePaul University
School of Computer Science, Telecommunications and Information Systems
243 S. Wabash Ave.
Chicago, IL 60606
{lytinen, tomuro}@cs.depaul.edu

Abstract

FAQFinder is a Web-based, natural language question-answering system. It answers a user's question by searching the Usenet Frequently Asked Questions (FAQ) files for a similar FAQ question, and displaying its answer to the user. To find the most similar FAQ question, FAQFinder measures similarity in part by using WordNet (Miller, 1990). To increase the accuracy of the similarity metric, we have incorporated an automated WordNet sense tagger into the process. In this paper, we show that the use of this sense tagger improves FAQFinder's matching accuracy. We argue that WordNet sense tagging can also be used in more general Web search tasks.

Introduction

FAQFinder (Burke, *et al.*, 1997) is a Web-based, natural language question-answering system which uses Usenet Frequently Asked Questions (FAQ) files to answer users' questions. Since FAQ files are written in question-and-answer format, FAQFinder tries to answer a user's question by retrieving the answer of a similar FAQ question, if one exists. Currently, FAQFinder uses a library of over 600 FAQ files, allowing it to answer questions about a broad range of subjects. Figure 1 shows the initial FAQFinder screen where a user question is entered as natural language. FAQFinder can be found at `http://faqfinder.ics.uci.edu`.

Given a user question, FAQFinder matches it with a FAQ question in 2 stages. In the first stage, relevant FAQ files are identified using the SMART information retrieval system (Salton, 1971). FAQFinder displays the 5 highest-ranked FAQ files to the user, who selects the file which looks most promising. Figure 2 shows a screen where the top 5 files are presented to the user. In the second stage, FAQFinder calculates a similarity score for each question in the selected FAQ file as compared to the user question, using three metrics: term vector similarity, coverage, and *semantic similarity*. Semantic similarity is calculated using WordNet (Miller, 1990), and involves finding the minimum path length between WordNet concepts (called *synonym sets* or *synsets*) referred to by words in the user and FAQ questions. The system displays up to 5 best-matching FAQ questions (if their similarity measures pass a certain threshold) to the user, who can then view the answers to one or more of these questions. Figure 3 shows a screen where the top 5 FAQ questions are presented to the user.

In (Burke *et al.*, 1997), we reported the results of an empirical test of FAQFinder's performance in terms of recall and rejection.[1] With the system tuned for maximum recall, we tested the system on a random set of questions collected from the FAQFinder server logs. We found that, in the second stage of FAQFinder's processing, the system correctly identified a relevant FAQ question for about 67% of the test questions. We also reported that if we tuned the system for improved rejection by adjusting the similarity threshold upward, recall suffered more than we would like. We hypothesized that an improved trade-off between recall and rejection would require *deeper* semantic analysis of user and FAQ questions.

However, there was one problem in our previous test in measuring the semantic similarity: the method was rather naive and was not very accurate. In order to avoid computational complexity which would arise from disambiguating word senses, the system simply determined the semantic similarity of two words as the inverse of the minimum distance between all senses of both words. Then the semantic similarities of individual pairs of words were combined to produce an overall semantic similarity between a user question and a FAQ question. Obviously, in some cases this method produces an inaccurate measure. For example, the words "bug" and "termite" have a relatively high semantic similarity, because "bug" and "termite" both have an "insect" sense. However, if these words actually appeared in the questions "How do I check my house for telephone bugs?" and "How do I check my house for

[1] Rejection is a metric somewhat analogous to precision which, as we will explain in section 3, we feel is a better measure of performance in the FAQFinder task than precision.

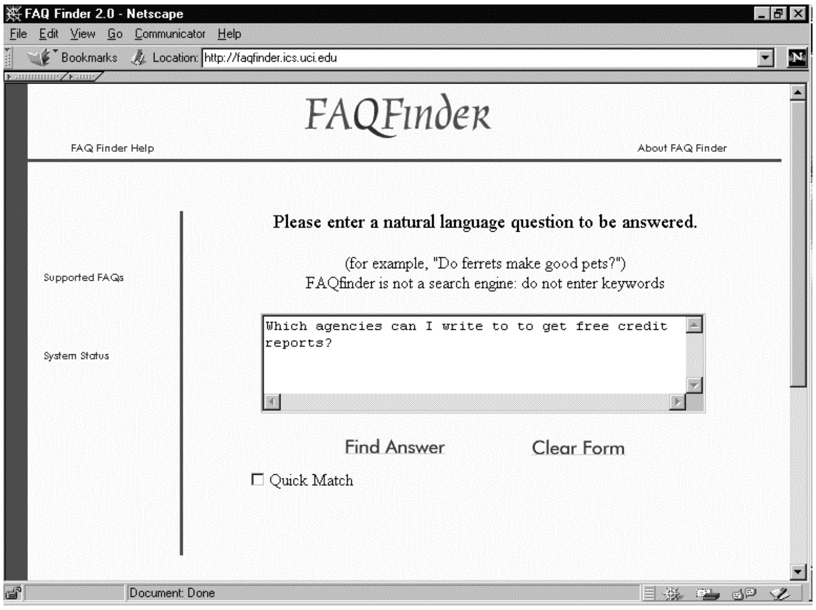

Figure 1: User question entered as a natural language query to FAQFinder

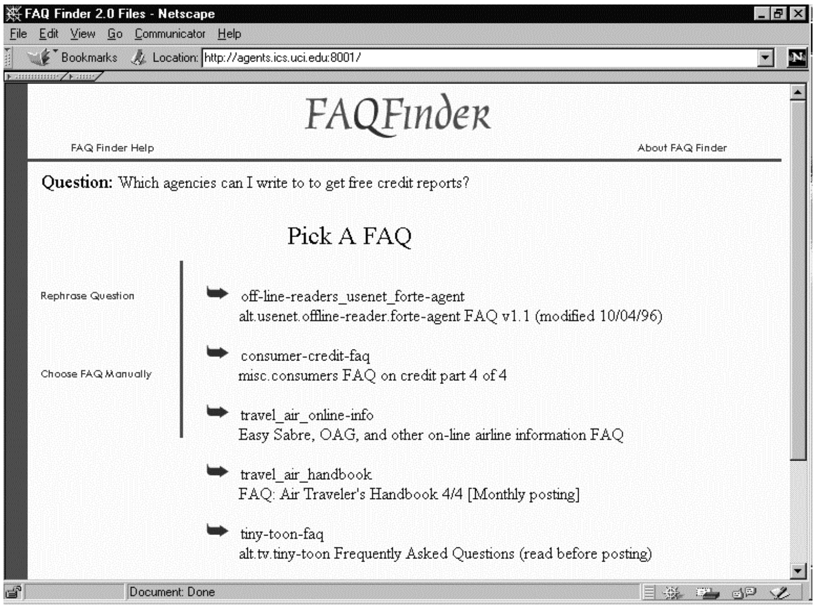

Figure 2: The 5 highest-ranked FAQ files

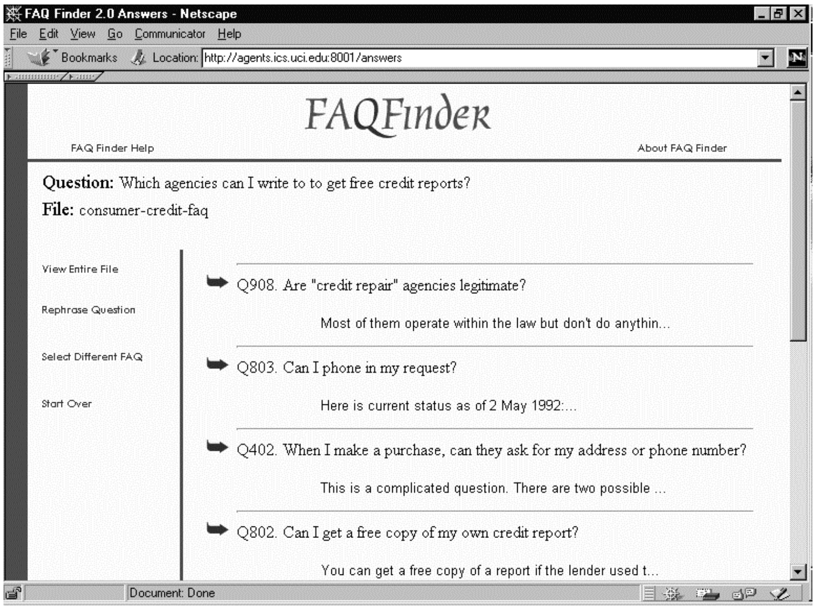

Figure 3: The 5 best-matching FAQ questions

termites?", then the sense of "bugs" on a telephone is not similar to "termites".

In this paper, we discuss an attempt to improve the semantic similarity measure used in FAQFinder. Our approach is to disambiguate word senses in both user and FAQ questions by using WordNet, before computing the semantic similarity of the two sentences. This way, the similarity measure becomes more accurate and the matching process becomes more efficient compared to the previous all-sense matching. With the addition of word sense disambiguation, we expected that FAQFinder would rank correct answers (i.e., FAQ questions which are semantically the closest to the user question) higher, thus the system would be able to maintain recall as the threshold was increased for better rejection. Indeed, the results we present in this paper show that FAQFinder's recall-rejection tradeoff benefits from the WordNet sense tagging. While sense tagging does not improve the system's performance when tuned for maximum recall, we found that as we tuned the system to gradually improve rejection, the degradation of recall was considerably lessened as compared to the performance of the system without sense tagging.

FAQFinder Without Sense Tagging

FAQFinder converts each FAQ question into a term vector and a *tagged term set*. The former is used to compute term vector similarity and coverage, while the latter is used to compute semantic similarity.

To compute the term vector for a question, stemming is performed on terms, using the WordNet morphing function, and a stop list is used to discard many closed-class terms. The weight w_i for each term t_i in the vector is computed using *tfidf* (Salton and McGill, 1983):

$$w_i = (1 + log(tf_i))\frac{logN}{df_i}$$

Here, a "document" is a single question; thus, N is the number of questions in the FAQ file, df_i is the number of questions in which t_i appears in the FAQ file, and tf_i is the number of times t_i appears in the question (usually 1).

To compute a question's tagged term set, each term in the question is tagged by part of speech using the Brill tagger (Brill, 1992).[2] Part of speech is used to constrain marker passing in WordNet. Terms are also stemmed and filtered using a stop list as before. The set of tagged terms which remain is stored.

On-line processing proceeds as follows: first, the user question is also converted to a term vector and a tagged term set in the same manner as the FAQ questions. To compute *tfidf*, the user question is considered to be one of the "documents"; thus N in the above equation is increased by 1, and all FAQ term vectors are adjusted to reflect the addition of the user question.

Next, the user question is compared with each FAQ question, and three metrics are computed. The first metric, term vector similarity, is computed as follows. Let $v_u = \langle w_{u1}, w_{u2}, ..., w_{un}\rangle$ be the term vector representing the user question, and let v_f =

[2] In the version of FAQFinder discussed in (Burke *et al.*, 1997), terms were tagged with their most commonly used syntactic category. We found that this did not significantly affect performance.

$\langle w_{f1}, w_{f2}, ..., w_{fn} \rangle$ be the term vector representing a FAQ question. Term vector similarity is computed using the cosine measure:

$$cos(v_u, v_f) = \frac{\sum w_{ui} w_{fi}}{\sqrt{\sum w_{ui}^2}\sqrt{\sum w_{fi}^2}}$$

The second metric, coverage, is the percentage of user question terms that appear in the FAQ question. It is obtained by finding the intersection of the (stemmed and stop list-filtered) terms in the term vectors of the two questions.

Finally, to compute the semantic similarity metric, we use the minimum distance between synsets in WordNet for pairs of terms, one term from the user question and the other from the FAQ question. In general, $\delta(t_1, t_2)$, the semantic distance between two (part-of-speech) tagged terms t_1 and t_2, each of which has n and m WordNet senses $S1 = \{s_1, .., s_n\}$ and $S2 = \{r_1, .., r_m\}$ respectively, is the minimum of all possible pair-wise semantic distances between S_1 and S_2, that is,

$$\delta(t_1, t_2) = \min_{si \in S1, rj \in S2} D(s_i, r_j)$$

where $D(s_i, r_j)$ is a path length between WordNet synsets s_i and r_j. For example, $\delta(bug, termite)$ is 2, because there is a hypernym (is-a) link between "bug" (noun sense 1) and "insect" (noun sense 1), and a hyponym (inverse is-a) link between "insect" (noun sense 1) and "termite" (noun sense 1). If there is no path between any of the synsets of t_1 and t_2, then $\delta(t_1, t_2) = \infty$.

Then, the semantic similarity between the user question and a FAQ question is defined as follows. Let $T_u = \{u_1, .., u_n\}$ be the tagged term set representing the user question and $T_f = \{f_1, .., f_m\}$ represent a FAQ question. Then $sem(T_u, T_f)$, the semantic similarity between T_u and T_f, is defined as follows:

$$sem(T_u, T_f) = \frac{I(u, f) + I(f, u)}{|T_u| + |T_f|}$$

where

$$I(u, f) = \sum_{u \in T} \frac{1}{1 + \min_{f \in T} \delta(u, f)}$$

and

$$I(f, u) = \sum_{f \in T} \frac{1}{1 + \min_{u \in T} \delta(f, u)}$$

and $|T_u|, |T_f|$ denote the size of T_u and T_f. Thus, $sem(T_u, T_f)$ is essentially a metric which is the normalized sum of the inverse of pair-wise semantic distances between all words in T_u and T_f measured from both directions.

The overall similarity measure for a user and FAQ question is computed as a weighted sum of the term vector similarity, coverage, and semantic similarity between the two questions. FAQFinder then displays the 5 FAQ questions with the highest similarity measures, if they exceed a threshold, to the user. Adjusting the threshold results in a trade-off between system recall and rejection.

WordNet Sense Tagging

The algorithm for sense tagging terms which we used in the current work is based on a *marker passing* (Quillian, 1968) technique. The idea behind this algorithm is to find a set of WordNet senses, one for each term in the question, which are semantically closest together. Although it is well-known that marker passing will sometimes find a wrong path between 2 terms, particularly in the case of semantic garden-path sentences such as "The astronomer married the star", marker passing is a general technique for measuring the closeness of semantic concepts, and has been used in many previous AI and NLP tasks. We anticipated that the accuracy would be good enough to improve FAQFinder's performance.

Our tagging algorithm is described as follows. Let $T = \{t_1, t_2, ..., t_n\}$ be the (part-of-speech) tagged term set for a question, where each t_i $(1 \leq i \leq n)$ has senses $S_i = \{s_{i1}, s_{i2}, ..., s_{im}\}$, we wish to find a combination of n senses/synsets $S_{min} = \{s_1, s_2, ..., s_n\}$, where each $s_i \in S_i$, such that the summing of all pair-wise distances between two synsets s_i and s_j $(1 \leq j \leq n, i \neq j)$ is minimized. In other words, we would like to obtain a particular combination of synsets S_{min}, which is a member of the set of all combinations of term synsets $\mathcal{S}$, that minimizes the following measure $\Delta(S)$:

$$\Delta(S) = \sum_{si \in S} \min_{sj \in S, i \neq j} D_(s_i, s_j)$$

However, finding the optimal S_{min} requires exhaustively computing Δ for all possible $S \in \mathcal{S}$, and that requires exponential time. Instead, we use a greedy algorithm, which computes S_{min} incrementally. Given a term set $T = \{t_1, t_2, ..., t_n\}$ for a question sentence, we first select a pair of two terms t_i, t_j $(i \neq j)$ which has the shortest semantic distance $\delta(t_i, t_j)$ among all term pairs in T. By this, we assign senses s_i, s_j for t_i, t_j (where $s_i \in S_i, s_j \in S_j$) respectively, since s_i, s_j are the senses which yielded $\delta(t_i, t_j)$. Then we initialize S to be $\{s_i, s_j\}$. After that, we incrementally expand S by considering all remaining $t_k \in T, k \neq i \neq j$ and adding the synset $s_k \in S_k$ which has the shortest distance to any member in S selected so far. This process continues until one synset has been selected for each $t \in T$. If there is no s_i for term t_i such that the distance is finite (i.e., there are no paths in WordNet from s_i to any of the other synsets in S and D is ∞), then a default sense 0 is assigned to t_i.[3]

[3] A WordNet sense tag of 0 essentially means that the term is untagged; i.e., $D(s0, s2)$, where $s0$ is tagged as sense 0 is the minimum distance between all senses of the first term and sense $s2$.

Our greedy algorithm described above is quite efficient: it runs in time $O(n^2)$, where n is the number of words in a question (considering the time required to compute the semantic distance between two synsets is a constant). Then, we modify the computation of semantic similarity by only using the synsets in S_{min} in computing $\delta(u, f)$.

Evaluation of FAQFinder

Recall vs. Rejection

To test the effect of our new similarity measure, we selected a new set of 153 test questions gathered from the FAQFinder server logs, and compared FAQFinder's performance with and without WordNet sense tagging. To do this, we prepared an "answer key" for the test set, by manually inspecting the FAQ files to find the best answer to each test question. The FAQ question corresponding to the best answer was recorded in the answer key as the correct match for that test question. For some test questions, several FAQ questions were judged to be correct matches, because their answers provided the same information. If no FAQ question's answer was suitable, then we recorded in the answer key that the test question should not match any question. Of the 153 questions, we judged 91 to have at least one correct matching FAQ question, and 62 to have no correct match.

We evaluated FAQFinder's performance[4] based on recall and *rejection*, a metric which we feel represents performance in FAQFinder's task better than precision does.

Recall is computed as the percentage of test questions with at least one correct match for which FAQFinder displays one of these correct matches to the user. Although this is not the standard way to compute recall (note that FAQFinder can achieve 100% recall by displaying only one, rather than all, correct matches for each question), we feel it is better for the FAQFinder task, because if several FAQ questions provide the same answer, the user does not care if FAQFinder finds all matching questions or just one.

Rejection is computed as the percentage of test questions with no correct match for which FAQFinder displays no matches to the user. While FAQFinder generally displays up to 5 questions from a FAQ file, a question is only displayed if its similarity metric exceeds a threshold; thus, if no FAQ question's similarity metric exceeds the threshold, then no questions are displayed. We feel that rejection is a more suitable metric for FAQFinder's task than precision, because it is a better measure of the system's response to unanswerable questions, and the answers to many questions typed in by users are not contained in FAQ files.[5]

Figure 4 shows FAQFinder's performance on the test set with and without WordNet sense tagging. Maximum recall is 62% both with and without WordNet sense tagging. Notice that FAQFinder with sense tagging retains recall as the threshold (and therefore rejection) is increased. With sense tagging, recall stays above 60% until rejection is around 50% and stays above 50% until rejection is around 75%. This means the system correctly returns no answer half of the time when there is indeed no answer, while maintaining near-maximum recall when there is an answer. Without sense tagging, recall quickly drops down to 50% when rejection is only 37%.

Sense Tagging

As for sense tagging, our tagging algorithm assigned (non-default) tags to 50% of the terms in all FAQ and test questions. The accuracy was around 60%. This means that FAQFinder achieved significantly improved rejection by disambiguating 30% of the terms. This is a quite encouraging result, because our current tagging method is very simple and can be improved with more sophisticated techniques.

Conclusions and Future Work

We have shown that a sense tagging algorithm based on spreading activation improves performance of the FAQFinder system, especially its rejection rate. Our result agrees with other work which shows improved document retrieval performance by disambiguating word senses (Sussna, 1993) or by using WordNet synsets (Gonzalo *et al.*, 1998).

There is much work on WordNet sense tagging which is relevant to our own work. Several groups have incorporated the use of spreading activation over a hierarchically organized lexicon/concepts with other disambiguation techniques, such as Bayesian networks (Wiebe, 1998) and learning predicate-argument structure (Resnik, 1997). Our spreading activation technique, while more straightforward than these other approaches, has the advantage that it does not require training, and is applicable to unrestricted domains. In future work, we plan to investigate the incorporation of additional information to see if the accuracy of sense tagging in FAQFinder can be improved. In particular, FAQFinder users generally type grammatical (though short) questions, and we would like to exploit syntactic structure if possible to improve sense disambiguation. The work of Resnik is particularly relevant here.

We also plan to investigate the use of sense disambiguation in more general retrieval tasks such as Web

[4] The performance evaluations presented in this paper only evaluate FAQFinder's second stage of processing. For test questions with no correct match, the second stage was run on the FAQ file ranked highest by SMART in response to the test question, to see if false matches would be returned.

[5] Although precision is affected by system performance on unanswerable questions, performance on these questions may be overshadowed by the fact that precision also penalizes the system for displaying 5 matches for answerable questions instead of just the correct match.

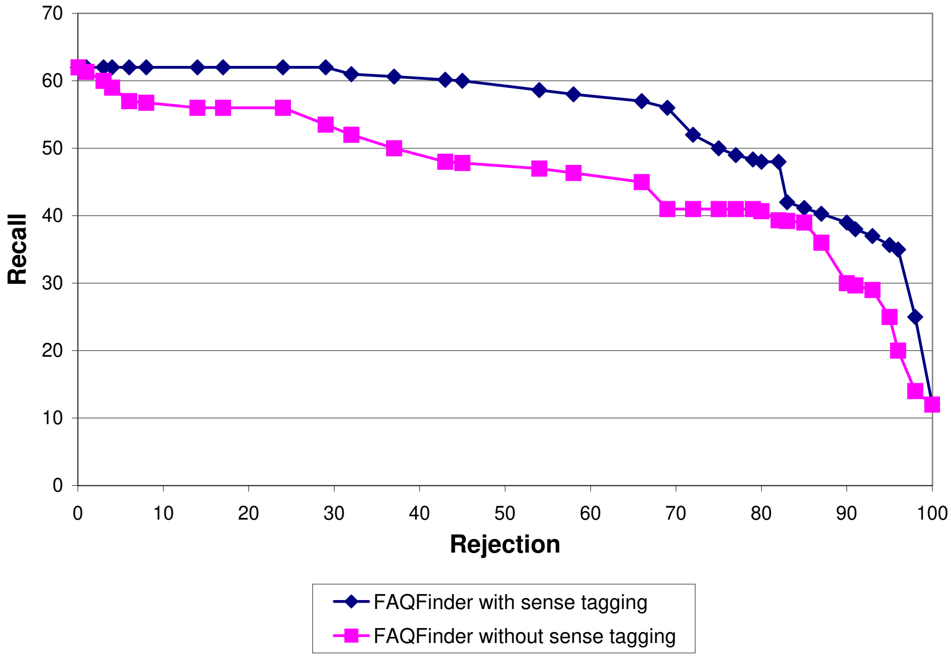

Figure 4: Recall vs. Rejection for FAQFinder with and without WordNet Sense Tagging

search. In FAQFinder, sense tagging and calculation of semantic similarity are much more computationally intensive than term vector processing. However, since FAQFinder matches single questions rather than entire documents, the computational complexity has not been an issue. Extending our approach to matching of HTML documents would probably only be feasible if we extracted key portions of documents, such as titles or portions of the body marked with certain HTML tags.

References

Brill, E. (1992). A Simple Rule-based Part of Speech Tagger. In *Proceedings of the Third Conference on Applied Natural Language Processing* (ANLP-92).

Burke, R., Hammond, K., Kulyukin, V., Lytinen, S., Tomuro, N. and Schoenberg, S. (1997). Question Answering from Frequently Asked Question Files: Experiences with the FAQFinder System. *AI Magazine*, Summer, **18** (2), pp. 57-66.

Gonzalo, J., Verdejo, F. Chugur, I. and Cigarran, J. (1998). Indexing with WordNet synsets can improve text retrieval. In *Proceedings of the Workshop on Usage of WordNet in Natural Language Processing Systems* at the 36th Annual Meeting of the Association for Computational Linguistics (ACL-98), Montreal, Canada.

Miller, G. (eds.) (1990). WORDNET: An Online Lexical Database. *International Journal of Lexicography*, **3** (4).

Quillian, R. (1968). Semantic Memory. In *Semantic Information Processing*, Marvin Minsky (ed), MIT Press: Cambridge, MA, p. 216-270.

Resnik, P. (1997). Selectional Preference and Sense Disambiguation, In *Proceedings of the ACL SIGLEX Workshop on Tagging Text with Lexical Semantics*, pp. 52-57.

Salton, G. (1971). *The SMART Retrieval System: Experiments in Automatic Document Processing.* Prentice-Hall, Englewood Cliffs, NJ.

Salton, G. and McGill, M. (1983). *Introduction to Modern Information Retrieval.* McGraw-Hill, New York.

Sussna, M. (1993). Word Sense Disambiguation for Free-text Indexing Using a Massive Semantic Network. In *Proceedings of the 2nd International Conference on Information and Knowledge Management (CIKM-93)*, Arlington, Virginia.

Wiebe, J., O'Hara, T. and Bruce, R. (1998). Constructing Bayesian networks from WordNet for word-sense disambiguation: representational and processing issues. In *Proceedings of the Workshop on Usage of WordNet in Natural Language Processing Systems* at the 36th Annual Meeting of the Association for Computational Linguistics (ACL-98), Montreal, Canada.