WIDIT in TREC-2005 HARD, Robust, and SPAM tracks

Kiduk Yang, Ning Yu, Nicholas George, Aaron Loehrlen, David McCaulay, Hui Zhang,
Shahrier Akram, Jue Mei, Ivan Record
School of Library and Information Science, Indiana University, Bloomington, Indiana 47405, U.S.A.
{kiyang, nyu, nlgeorge, aloehrle, smccaula, hz3, sakram, jumei, irecord}@indiana.edu

1. Introduction

Web Information Discovery Tool (WIDIT) Laboratory at the Indiana University School of Library and Information Science participated in the HARD, Robust, and SPAM tracks in TREC-2005. The basic approach of WIDIT is to combine multiple methods as well as to leverage multiple sources of evidence. Our main strategies for the tracks were: query expansion and fusion optimization for the HARD and Robust tracks; and combination of probabilistic, rule-based, pattern-based, and blacklist email filters for the SPAM track.

2. HARD track

In this year's HARD track, we explored the possibility of improving retrieval performance by taking a fusion approach that combines a variety of techniques as well as different data sources. In order to deal effectively with the "difficult" topics, we employed a variety of query expansion strategies, the results of which were combined via an automatic fusion optimization process.

We hypothesized that the "difficulty" of topics is often due to the lack of appropriate query terms and/or misguided emphasis on non-pivotal query terms by the system. Thus, our first-tier solution was to devise a wide range of query expansion methods that can not only enrich the query with useful term additions but also identify important query terms. Our automatic query expansion included such techniques as noun phrase extraction, acronym expansion, synonym identification, definition term extraction, keyword extraction by overlapping sliding window, and Web query expansion.

For synonym identification, we integrated a sense disambiguation module into WIDIT's synset identification module so that best synonym set can be selected according to the term context. To reduce the noise from word definitions, we applied the overlapping sliding window (OSW) method to multiple definitions harvested from web and extracted the overlapping terms. To extract noun phrase, we combined the results of multiple NLP taggers as well as applying the OSW method. OSW method was also applied to topic fields to identify important topic terms. The Web query expansion method was a slight modification of the PIRC approach.

To produce the optimum baseline results, we merged various combinations of query formulation results, which involved combinations of topic fields (title, description, narrative) and stemming (simple plural stemmer, combination stemmer), and the query expansion results using a weighted sum fusion formula. The fusion weights were determined using last year's Robust data to train the system via an automatic fusion optimization process. The fusion optimization involved iterations of fusion runs (i.e., result merging), where best performing systems in selected categories (e.g., short query, top 10 systems, etc.) were combined using average precision as fusion weights until the performance gain fell below a threshold.

We viewed the clarification form (CF) as both manual query expansion and relevance feedback mechanism. Our clarification form included query term synonyms, noun phrase, and best sentences from top documents of the baseline result. Since difficult topics tend to produce few relevant documents in top ranks, we clustered the results and selected the best sentence from each cluster to include in the CF. In addition to expanding the query with user-selected terms

from the clarification form, we also utilized the user's best sentence selection by boosting the rank of the documents in which selected sentences occurred.

2.1 WIDIT HARD System

WIDIT HARD system consists of five main modules: indexing, retrieval, fusion (i.e. result merging), reranking, and query expansion modules. The query expansion module employs a wide range of query expansion methods that can not only enrich the query with useful term additions but also identify important query terms. The retrieval module produces multiple result sets from using different query formulations. The fusion module, which is optimized via an automatic tuning process, combines result sets using weighted sum formula, and the reranking module boosts the ranks of documents with important phrases and floats to the top ranks documents selected as relevant in CF. The overview of WIDIT HARD system architecture is displayed in Figure 1.

Figure 1. WIDIT HARD System Architecture
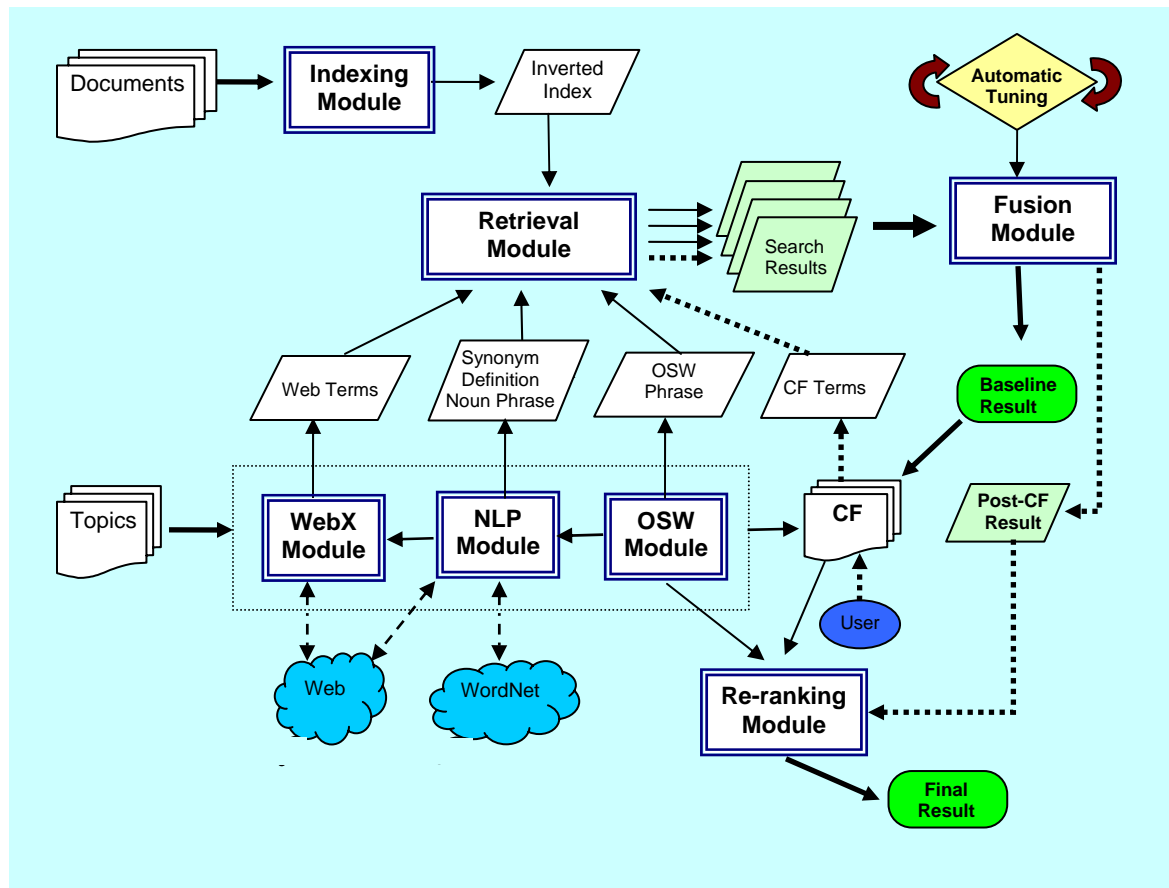
2.2 Query Expansion

The query expansion module consists of three submodules: Web expansion (WebX) module expands the query with terms from Google search results; NLP module finds synonyms and definitions of query terms and identifies nouns and noun phrases in the query; Overlapping Sliding Window (OSW) module extract key phrases from the query.

2.2.1 OSW Module

The main function of OSW module is to identify important phrases. OSW method, which is based on the assumption that phrases appearing in multiple fields/sources tend to be important, works as follows:

i.   Define window size and the number or maximum words allowed between words.
ii.   Slide window from the first word in the source field/source. For each of the phrase it catches, look for the same/similar phrase in the search fields/sources.
iii. Produce the OSW phrases
iv.   Change source field/source and repeat step 1 to 3 till all the fields/sources have been used.

OSW method was applied to topic descriptions and query term definitions to identify key phrases.

2.2.2 NLP Module

WIDIT's NLP module expands acronyms using a Web-harvested acronym list, identifies nouns and noun phrases using multiple taggers, and finds synonyms and definitions via querying the Web. Two main objectives of the NLP module refinement this year were to reduce noise in query expansion and to identify key (i.e. "important") phrases. For noise reduction, we integrated a sense disambiguation[1] into WIDIT's synset identification module so that best synonym set can be identified based on the term context, and refined the WIDIT's definition module by applying OSW to extract overlapping terms from the multiple Web-harvested definitions (*WordIQ*, *Dictionary.com*, *Google*, *Answers.com*). For key phrase identification, we used a combination of NLP tools as well as WordNet to identify 4 types of noun phrases: proper names, dictionary phrases, simple phrases, complex phrases. The noun phrase identification algorithm, which is based on the UIC approach (Liu, Sun, & Yu, 2004), is described below and depicted graphically in Figure 2:

   i.   Apply Minipar to identify proper names and term relations (AND, OR, NOT).
   ii.   Apply Brill's tagger to identify part of speech.
   iii. Apply Collins' parser to decompose noun phrases as base phrases that cannot be decomposed by Collin's parser.
   iv.   Query WordNet to identify proper nouns and dictionary phrases (Liu et al., 2004).
   v.   Categorize phrases which are not proper noun or dictionary phrases into complex or simple phrases.

   The Reliable Information Access Workshop report (Harman & Buckley, 2004) listed 10 reasons why a retrieval system may fail. The failure analysis of 45 TREC topics showed that missing an important aspect or term in a query contributes to the largest number of problems. In order to address this problem, we created a relation identification module that identifies terms with three types of Boolean "AND" relations (BoolAND)—parallel, verb-noun, and modifier-noun—where BoolAND terms are query terms all of which must appear in relevant documents. To identify parallel relations, we used a simple rule that selects the terms or phrases linked by "and" in the title field. To identify verb-noun and modifier-noun relations, we used an NLP parser called Minipar (http://www.cs.ualberta.ca/~lindek/minipar.htm) on the topic description. BoolAND relations were included in the clarification form (CF) for the human experts for validation, and the CF feedback was used as input to the reranking module.

---

[1] WordNet word sense disambiguation software developed by the Natural Language Processing Group at the University of Minnesota, Duluth

Figure 2. Noun Phrase Identification Diagram
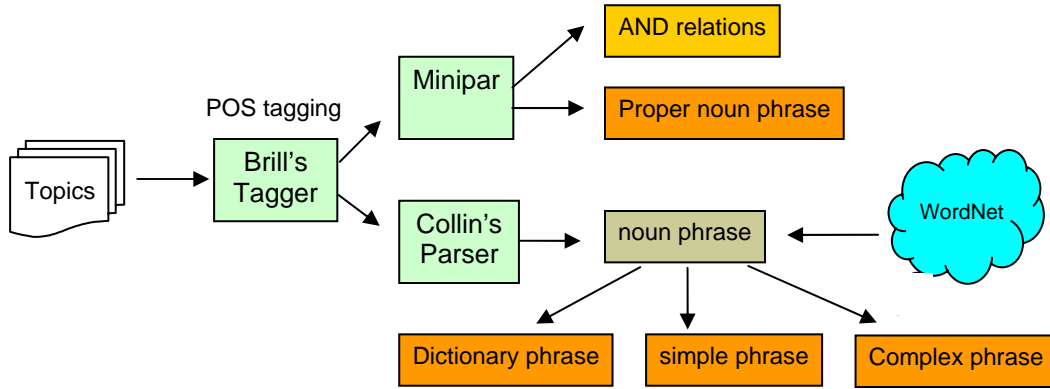
2.2.3 WebX Module

The PIRCS group has demonstrated that web expansion is an "effective" method for improving the performance of weak (i.e. difficult) topics (Grunfeld et al., 2004; Kwok et al., 2005). WebX module, which is based on the PIRC approach, expands the query with related terms harvested from Google search results. WebX module consists of Web query construction, Web search, search result parsing and term selection. Figure 3 illustrates the architecture of the WebX module.

Figure 3. WebX Module Architecture
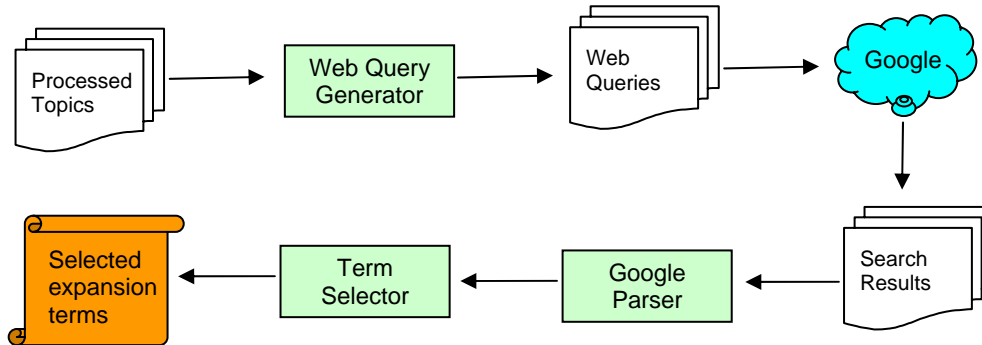
The Web query generator constructs Web queries by selecting up to 5 most salient terms from the processed HARD topics (i.e., stopped and stemmed text, nouns, phrases). The queries are then sent to Google, and subsequent search results (the snippets and the body texts) are parsed to extract up to 60 terms per topic to be used as query expansion terms.

2.3   Fusion Optimization

To produce the optimum baseline results, we merged various combinations of query formulation results, which involved combinations of topic fields (title, description, narrative) and stemming (simple plural stemmer, combination stemmer), and the query expansion results using a weighted sum fusion formula[2]. The conventional approaches to fusion weight determination ranges from

---

[2] Weighted Sum formula combines document scores weighted by the relative contribution of the systems.

$FS = (\sum w_i * S_i)$ where wi is the weight (estimated contribution) of system i and Si is the normalized retrieval score of a document by system i.

4

exhaustive search of weight combinations, which is computationally prohibitive when the number of parameters is large, to the use of past evidence to restrict the search space, which can be misleading at times. Last year, we devised a novel man-machine hybrid approach called the *Dynamic Tuning* as an alternative (Yang, Yu, & Lee, 2005; Yang & Yu, in press). This year, we devised another alternative fusion weight determination method called *Auto-Fusion Optimization*. The *Auto-Fusion Optimization* involves iterations of fusion runs (i.e., result merging), where best performing systems in selected categories (e.g., short query, top 10 systems, etc.) are combined using average precision as fusion weights until the performance gain fell below a threshold. Figure 4 illustrates the automatic fusion optimization process.

Figure 4. Auto-Fusion Optimization
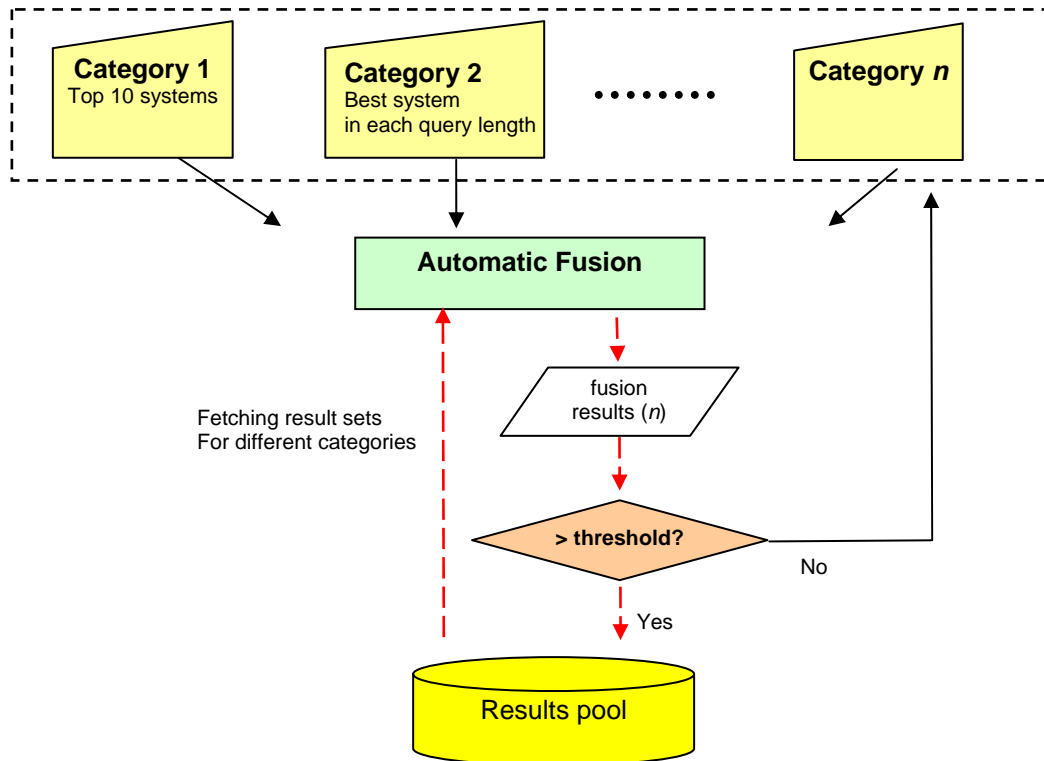
2.4   Clarification Form

Most HARD participants have discovered that the clarification form (CF) is an effective mechanism for improving the baseline result with user feedback. The main objective of our CF design strategy this year was to obtain more accurate and diverse feedback. We extended last year's CF approach (Yang et al., 2005), which served as manual query expansion and relevance feedback mechanism, to include the validation of BoolAnd relations. In addition to displaying important phrases and best sentences from top 200 retrieved documents[3], our CF (Figure 5) included synonym sets, definition terms, and query term relations with the use of JavaScript to make the interaction more friendly and efficient. The CF terms selected by the user was used to create a CF-expanded query. Phrases, BoolAnd terms and relevant documents identified in CF

---

[3] Since weak topics tend to retrieve non-relevant documents at top ranks, we clustered the top 200 documents and selected the best sentence from each cluster.

were used by the reranking module to boosts the ranks of documents with important phrases and relevant documents identified by the user.

Figure 5. Clarification Form

2.5 Reranking

The objective of reranking is to float low ranking relevant documents to the top ranks based on post-retrieval analysis of reranking factors. After identifying reranking factors such as *OSW terms*, *CF terms*, and *CF-reldocs*, which are relevant documents identified in CF form, we computed the reranking factor scores (*rf_sc*) for top *k* documents and boosted the ranks of documents with *rf_sc* above a threshold score above a fixed rank R using the following formula:

$$doc\_score = rf\_sc + doc\_score@rankR \qquad (1)$$

Although reranking does not retrieve any new relevant documents (i.e. no recall improvement), it can produce high precision improvement via post-retrieval compensation (e.g. phrase matching) or force rank-boosting to accommodate trusted information (e.g. *CF-reldocs*).

2.6 Results

Web query expansion (*WebX*) was the most effective method of all the query expansion methods. *WebX* method showed most gain in performance for short queries (i.e. title) but had an adverse effect for longer queries (i.e. description) except when using the rotating window approach (Figure 6). Among non-*WebX* query expansion methods, *Proper Noun Phrases*, *Overlapping Sliding Window* (*OSW*), and *CF Terms* helped retrieval performance for longer queries. The effect of query expansion is influenced by the query length. Without query expansion, longer queries usually outperform the shorter queries (Figure 7). With query expansion, however, query

length has opposite effect on *WebX* and non-*WebX* methods. The composite effects of query expansion and query length suggest that *WebX* should be applied to short queries, which contain less noise that can be exaggerated by Web expansion, and non- *WebX* should be applied to longer queries, which contain more information that query expansion methods can leverage.

Figure 6. Web Query Expansion Effect

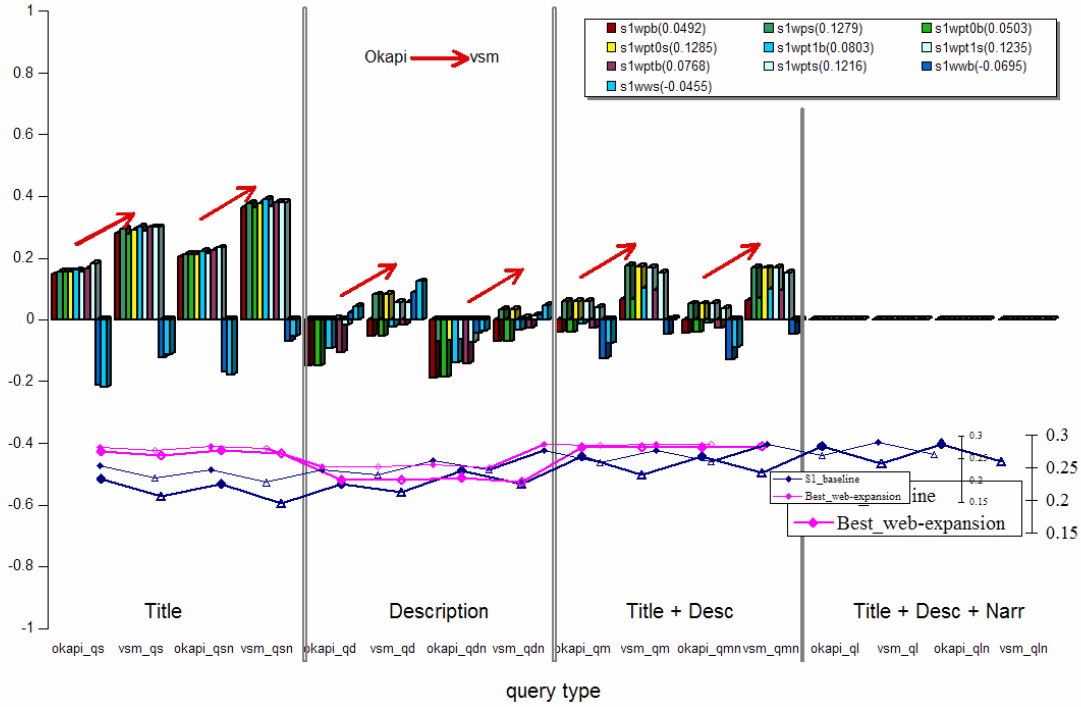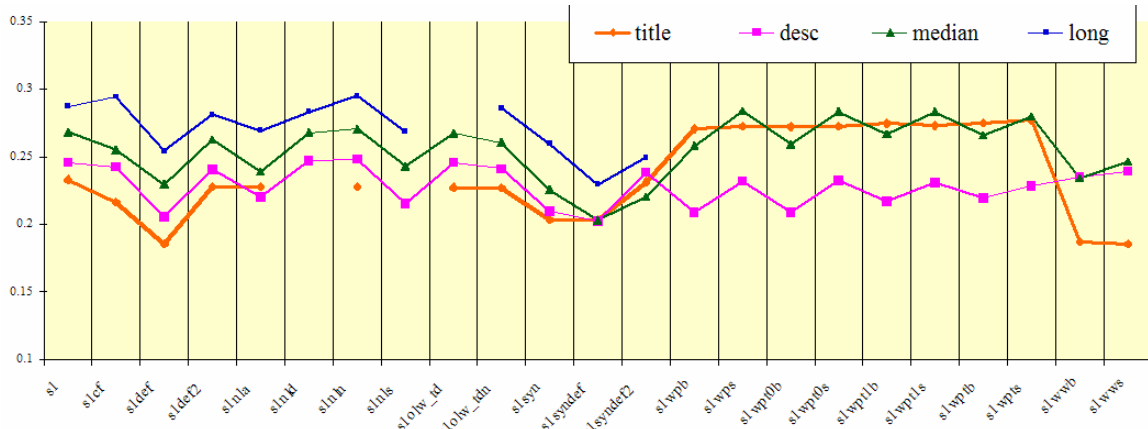**Figure 7. Query Length Effect**

As was the case last year, fusion (i.e. result merging) improved the retrieval performance across the board, showing that fusion optimization is a viable method to streamline the process of combining numerous result sets in an efficient manner (Table 1). As for reranking factors, CF-relevant documents had the most positive effect, followed by *OSW* and *CF Terms*.

7

Table 1. Fusion Effect

| Best MRP	Title Only	Desc. Only	Title+Desc+Narr
Non-fusion runs	0.2416	0.2395	0.2965
Fusion runs	0.3020 (+25%)	0.2625 (+10%)	0.3451 (+16%)

3. Robust Track

Our Robust track approach was essentially the same as our HARD track approach except for the lack of CF component and the addition of difficulty prediction component. Our difficulty prediction methods were: *average inverse collection term frequency*, which assumes that performance of a query can be predicted by the average quality of its composing terms, and *standard deviation of inverse document frequency* (*idf*), which assumes that composing terms of a poorly-performing query tend to have similar *idf* values. However, neither method proved to be effective. We suspect that this is due to the fact that our submitted results were from fusion runs that combine the results of various query formulations.

4. SPAM Track

For SPAM track, we created a fusion filter that integrates four individual SPAM filters. The individual filters make use of an adaptive learning process that learns the features of the test corpus in real time during the filtering process. The filters consist of a naïve Bayesian (NB) filter, a pattern-based (PB) filter, a rule-based (RB) filter, and a blacklist (BL) filter.

The results of four filters are taken into consideration to generate the final result by a fusion module, which implements both rule-based and weighted sum methods. The fusion module computes a set of weighted sum scores as well as rule-based scores that determine the contribution of each filter based on factors such as training run performance, filter overlap, and filter characteristics (e.g., BL filter has no false positives). The tuning of the weighted sum formulas proved difficult due to uneven performance across collections. Furthermore, rule-based fusion method showed much superior overall performance in training runs. Figure 6 describes the WIDIT SPAM system architecture.

Figure 6. WIDIT SPAM System
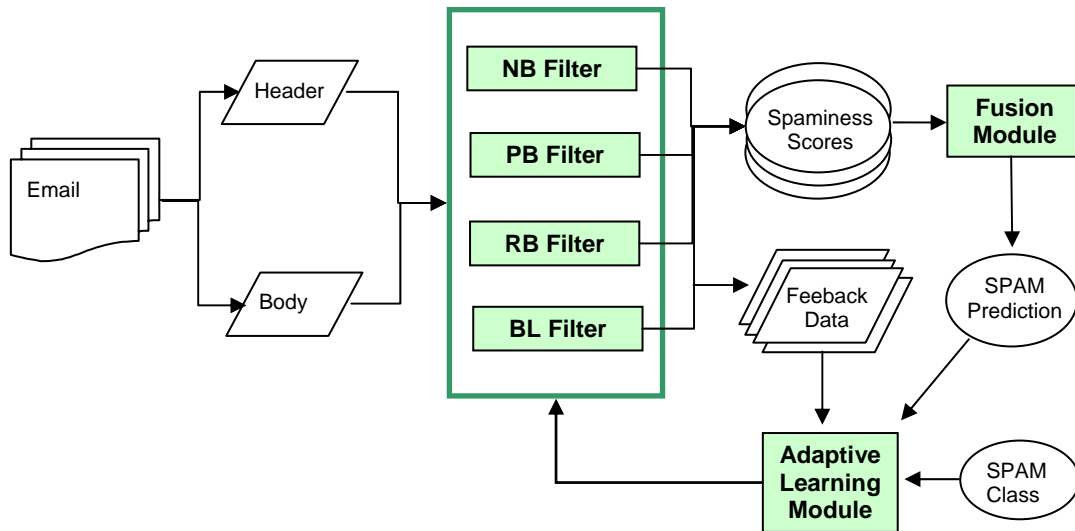
4.1 Naïve Bayes Filter

The WIDIT naïve Bayesian filter is a modified version of Graham's bspam filter (Graham, 2002; Graham, 2003). NB filter calculates the probability of the given email being spam by considering the current ratio of hams to spams, the number of hams that contain the term, and the number of spams that contain the term. Since NB filter dynamically determines the spaminess of terms based on their occurrences in spam email, it has an advantage over rule-based filters that can quickly become outdated when spammers devise new ways to fool spam filters.

NB filter performance was very strong (95-plus % accuracy) in preliminary tests, so minimal filter optimization was performed. The two optimization parameters were whether to train initially on a large number of emails (approx 250) or a small number of emails (approx 20) and whether to subsequently train the filter for all incoming emails or just the ones that the filter classified incorrectly. Based on the analysis of test runs, we chose to use the larger initial training data and to train on all incoming emails. NB was the most dominant filter that achieved best performance in training runs using the SpamAssassin data.

4.2 Pattern-based Filter

The WIDIT pattern-based (PB) filter uses a sliding window to identify sub-strings within emails that may identify it as a spam or a ham. PB filter is somewhat similar to NB filter in that it makes the spam/ham determination based on the spaminess of tokens in an email. The tokenization of email by PB filter, however, is string-based rather than word-based, which can identify spam patterns that span word boundary as well as those that are embedded inside a word. Furthermore, spam classification by PB filter is not probabilistic, but uses a simple pattern matching algorithm based on "guilt by association". We used IBM's Teiresias pattern discovery tool[4], which uses a sliding window that crosses word boundaries to extract patterns. Teiresias, originally used for genome and protein sequence research, was utilized by Rigoutsos and Huynh (2004) for SPAM filtering in the Chung-Kwei system as a part of the SpamGuru project.

The initial formulation of PB filter had poor performance, which were optimized after prototype submission by reducing the window size. The optimization involved removal of wild cards in pattern extraction (i.e., exact pattern matching only), and the shrinking of the sliding window to only three characters. The resulting PB filter produced the second best performance next to NB filter in training runs.

4.3 Rule-based Filter

The WIDIT rule-based (RB) filter evaluates an email according to a set of static rules constructed manually. The basic idea underlying RB filter was to manually identify email patterns not discovered by the other filters. The advantage of RB filter is its ability to leverage patterns and rules that go beyond tokens (e.g., an email is likely to be spam if its "To:" line is blank). RB filter assumes that spammers use similar techniques in their messages and relies on human intelligence, especially pattern-recognition ability, to identify patterns across all parts of email (i.e., header, subject, body) and construct a set of rules to make use of those patterns in filtering out spam emails.

Unfortunately, RB filter performance fell much below our initial expectations despite extensive data analysis to optimize the RB filter. RB filter appeared to perform unevenly across collections and it was difficult to find a balancing point between false positive and false negative optimizations.

---

[4] Teiresias implements the pattern extraction algorithm developed by IBM Bioinformatics and Pattern Discovery Group circa 1998 (http://cbcsrv.watson.ibm.com/Tspd.html).

4.4   Blacklist Filter

The WIDIT blacklist (BL) filter uses four blacklists (domain name, email address, IP address, URL), which are initially built by mining the lists of known spammers posted on SPAM resource websites (Goodman, 2004). The blacklists are updated during the adaptive learning process based on the occurrences of domain names, email and IP addresses in spam email headers and URLs in spam email body texts. A "stop list", which contains domains, such as hotmail.com, that are known to be used by spammers and non-spammers alike, is built in the same fashion to avoid possible over tuning. Although BL filter can identify only a small subset of spam emails, its spam accuracy is very high. The BL filter was optimized to produce no false positives at the cost of lower true positive identification.

4.5   Fusion Module

The results of four filters are taken into consideration to generate the final result by a fusion module, which implements both rule-based and weighted sum methods. The fusion module computes a set of weighted sum scores as well as rule-based scores that determine the contribution of each filter based on factors such as training run performance, filter overlap, and filter characteristics (e.g., BL filter has no false positives).

   Individual spam filters have their own strengths and vulnerabilities. By combining multiple filters, we hope to create a filter that will identify a wide variety of spamming ploys with the combined strength of all the filters.

References

Graham, P. (2002). A Plan for Spam. *Retrieved from* http://www.paulgraham.com/spam.html, May 1, 2005.

Graham, P. (2003). Better Bayesian Filtering. *Retrieved from* http://www.paulgraham.com/spam.html, April 30, 2005.

Liu, S., Liu, F., Yu, C., & Meng, W. (2004). An effective approach to document retrieval via utilizing WordNet and recognizing phrases. *Proceedings of the 27th Annual International ACM SIGIR Conference*.

Liu, S., Sun, C., & Yu, C. (2005). UIC at TREC-2004: Robust Track. *Proceedings of the 13th Text Retrieval Conference.*

Goodman, J. (2004). IP Addresses in Email Clients. *Conference on Email and Anti-Spam 2004*.

Grunfeld, L., Kwok, K.L., Dinstl, N., & Deng, P. (2004). TREC 2003 Robust, HARD, and QA track experiments using PIRCS. *Proceedings of the 12th Text Retrieval Conference*, 510-521.

Rigoutsos, I., Huynh, T. (2004). Chung-Kwei: a Pattern-discovery-based System for the Automatic Ientification of Unsolicited E-mail Messages. *Proceedings of the First Conference on E-Mail and Anti-Spam*.

Harman, D. & Buckley, C. (2004). The NRRC Reliable Information Access (RIA) workshop. *Proceedings of the 27th Annual International ACM SIGIR Conference*, 528-529.

Kwok, K. L., Grunfeld, L., Sun, H. L., & Deng, P. (2005). TREC2004 robust track experiments using PIRCS. *Proceedings of the 13th Text REtrieval Conference*.

Yang, K., & Yu, N. (2005). WIDIT: Fusion-based Approach to Web Search Optimization. *Asian Information Retrieval Symposium 2005*.

Yang, K, Yu, N., & Lee, Y (2005). Dynamic Tuning for Fusion: Harnessing Human Intelligence to Optimize System Performance. *Proceedings of the 9th World Multi-Conference on Systemics, Cybernetics and Informatics*.

Yang, K., Yu, N., Wead, A., La Rowe, G., Li, Y. H., French, C., & Lee, Y (2005). WIDIT in TREC2004 Genomics, HARD, Robust, and Web tracks. *Proceedings of the 13th Text Retrieval Conference (TREC2004)*.