JHU/APL at TREC 2002: Experiments in Filtering and Arabic Retrieval

Paul McNamee, Christine Piatko, and James Mayfield
Research and Technology Development Center
Johns Hopkins University Applied Physics Laboratory
11100 Johns Hopkins Road, Laurel, Maryland 20723-6099 USA
{mcnamee, piatko, mayfield}@jhuapl.edu

Overview

The Johns Hopkins University Applied Physics Laboratory (JHU/APL) participated in two tracks at this year's conference. We participated in the filtering track, again addressing the batch and routing subtasks, as well as the adaptive task for the first time. We also continued experiments in Arabic retrieval, emphasizing language-neutral approaches.

For ranked retrieval, we relied on a statistical language model to compute query/document similarity values. Hiemstra and de Vries describe such a linguistically motivated probabilistic model and explain how it relates to both the Boolean and vector space models [4]. The model has also been cast as a rudimentary Hidden Markov Model [13]. Although the model does not explicitly incorporate inverse document frequency, it does favor documents that contain more of the rare query terms. The similarity measure can be computed as

$$Sim(q,d) = \prod_{t \in q} \left(\alpha \cdot f(t,d) + (1-\alpha) \cdot df(t)\right)^{f(t,q)}$$

Equation 1. Similarity calculation.

where α is the probability that a query word is generated by a document-specific model, and *(1- α)* is the probability that it is generated by a generic language model. *df(t)* denotes the mean relative document frequency of term *t*. In our experiments an α of between 0.15 and 0.3 has worked well, but performance is fairly insensitive to the precise value used.

For text classification problems we used Support Vector Machines (SVMs), which efficiently perform binary classification tasks. We applied SVMs to this year's Filtering tasks; however, some of our routing runs were based on statistical language models instead.

Filtering Track

We participated in the routing, batch and adaptive tasks of the filtering track.

Filtering Approach Background

We continued to investigate the application of Support Vector Machines (SVMs) to filtering tasks. SVMs are used to create classifiers from a set of labeled training data, finding a hyperplane (possibly in a transformed space) to separate positive examples from negative examples. This hyperplane is chosen to maximize the margin (or distance) to the training points. The promise of large margin classification is that it does not overfit the training data and generalizes well to test data of similar distribution. See Hearst [3] for a general discussion of SVMs. We used the SVM-light package (version 3.50, by Thorsten Joachims [15]) to create classifiers based on the training data for classification of the test data, and wrote a JNI interface to SVM-light to support filtering with our HAIRCUT system. All runs used stem indices using a derivative version of the SMART stemmer.

We slightly reduced the term space to create test and training document vectors. Terms were selected using the top stems by document frequency in the training set. (Exact numbers of stems differed for different runs, noted per task in descriptions below.) Stopwords were not removed. We used tf/idf weighted vectors for each document. IDF values were based on training index statistics. Vectors were normalized to unit length. Given *n* positive training documents for a topic, we chose either all other training qrels documents as (presumed) negative examples, or randomly sampled (number of known positive examples) * *NegativeToPositiveRatio* presumed negative examples from the training index, throwing away any that were actually positive. (The use of all negatives or a particular ratio is noted below.) We trained linear SVMs, weighting positive and negative training examples equally *(-j 1* flag in SVM-light).

Filtering Training

Over the course of the year we had performed several experiments using the Reuters Corpus [18] and topics from TREC 2001. Based on track guidelines, we wanted to establish various parameters necessary for our system based on alternative data. We chose to reset these based on performance on Financial Times data that had been used in the TREC-8 Filtering Track. We did this in a straightforward way for the routing and batch training and test sets. However, no training documents had been used for the adaptive task in TREC-8, so for this training we randomly selected three relevant training documents from the batch FT training qrels for each topic.

Routing Task

We submitted two official runs for routing. We submitted an SVM based run *apl11Fsvm* and a rank-based merge run *apl11Frm*, the merge of the SVM run with an unofficial score-based run *apl11Frs*.

Our statistical language model-based run *apl11Frs* used simulated routing (using a modified version of our HAIRCUT system to score indexed test documents using training index statistics). We formed queries using 60 terms per topic that were selected from the positive qrels training documents. Term selection was accomplished using mutual information based difference statistics with respect to the training documents.

For the SVM routing run *apl11Fsvm*, we used the top 12000 features ranked by document frequency and 10 times as many negative documents as known positive examples to train an SVM. We kept track of the top 1000 documents for a topic in a heap based on the SVM score.

	Avg. prec.	# terms	# bests	# ≥ median (50 topics)
apl11Frm	0.330	12000	4	43
apl11Fsvm	0.218	12000	1	24
apl11Frs	0.364	60	Score run	
apl11Frsvm2	0.364	40000	SVM run	
apl11Frm2	0.412	40000	Merge run	

Table 1. APL Routing Results, Assessor topics. Highlighted rows indicate unofficial runs.

	Avg. prec.	# bests	# ≥ median (50 topics)
apl11Frm	0.042	5	37
apl11Frsvm	0.043	4	35
apl11Frs	0.035	Score run	
apl11Frsvm2	0.041	SVM run	
apl11Frm2	0.045	Merge run	

Table 2. APL Routing Results, Intersection topics. Highlighted rows indicate unofficial runs.

In subsequent analysis of our results we realized that we submitted the wrong SVM-based routing run. We had intended to submit the SVM run *apl11Frsvm2* based on 40000 df terms (the best of numbers to use on FT data). This run does well, and makes an excellent run (*apl11Frm2)* when rank-merged with the score-based run.

Batch Filtering Task

We used the score of the test document from the topic-specific SVM to decide whether to return a document as possibly relevant. We used cross-validation for threshold selection. We applied *n*-fold cross-validation on training data to find the best threshold per topic for the given score function being optimized.

Lewis had applied exhaustive leave-one out to find optimal SVM *j* weights per topic last year [9], but this was computationally unrealistic for our implementation. Particular choices of *n* we used for cross-validation are noted below.

Batch Using Linear SVMs with TF/IDF Vectors

For the submitted *apl11FbF* run, we used the top 20000 df terms. We used all the presumed negative training examples from the training index. We used three-fold cross-validation on the training data to select the best topic-specific score thresholds for the T11F measure.

For the submitted run *apl11FbSU* run, we used the top 12000 df terms. We again used all the presumed negative training examples from the training index. We used five-fold cross-validation on the training data to select the best topic-specific thresholds for the T11SU measure.

	T11SU	T11F	SetPrec	SetRecall
apl11FbF	0.391	0.216	0.409	0.117
apl11FbSU	0.293	0.181	0.244	0.255

Table 3. APL Batch Results, Assessor topics

	T11SU	T11F	SetPrec	SetRecall
apl11FbF	0.338	0.026	0.068	0.013
apl11FbSU	0.035	0.028	0.027	0.275

Table 4. APL Batch Results, Intersection topics

Adaptive Filtering Task

We developed two heuristic approaches to using SVMs for adaptive filtering. While there is theory to explain how a static SVM generalizes to test data of similar distribution to training data, this theory has not yet been well developed for SVMs that are adapting over time based on feedback.

Our two approaches were similar to early filtering score-based buffer window approaches. An SVM was created based on training data for each topic. Three "buffers" of documents from the test document stream were maintained: a "good" buffer of documents correctly judged relevant; a "bad" buffer of those that had been incorrectly judged relevant; and a "presumed bad" (unjudged) buffer of those documents not retrieved (and presumed irrelevant).

All buffers were capped to a fixed size. Window sizes for the buffers were set somewhat arbitrarily based on limited experimentation as follows: 750 documents for the known positive documents, 750 for the known negative documents and 2000 or 50 (noted below) for the presumed negative documents (those not retrieved). We also used a heuristic parameter to guess occasionally if no documents had been retrieved for a long time.

Our first approach used queues for all three of these buffers of documents, expiring the old documents as the buffers overfilled. The notion here is that older documents are less valuable than newer ones. (In this case we used the larger size 2000 buffer for negative documents.) Our second approach used heaps, based on the absolute value of the SVM score (smallest value on top), throwing away documents with larger scores as the buffers overfilled. The notion here is that documents closer to the margin of the current SVM are more useful as discriminative examples for training. (Here we used the smaller buffer size of 50 for the presumed negative documents.)

Our strategy was to update the topic-specific SVMs at data-driven intervals, using the documents in the current buffers. The intervals were based on sizes of the current buffers, as well as a "rate of change" heuristic.

Using the queue approach we had observed good (but statistically variable performance) based on the TREC 2001 data and topics (0.35 average T10SU, and a boxplot as good or better than the official boxplots from TREC-10 filtering). However, we did not do as well for this year's official adaptive task.

	T11SU	T11F	SetPrec	SetRecall
apl11Fah1	0.342	0.104	0.377	0.039
apl11Fah2	0.342	0.104	0.377	0.039
apl11Faq1	0.059	0.09	0.084	0.369
apl11Faq2	0.085	0.118	0.115	0.355

Table 5. APL Adaptive Results, Assessor topics

Clearly, the heap approach returned too few documents, whereas the queue approach returned too many. This is probably mainly due to the much lower amount of feedback. It was probably also adversely affected by our choice of "guess occasionally" parameter that guessed too often.

Filtering Results Discussion

In a low training/feedback situation, filtering seems to require more of a Statistical Language Model score-based approach. Based on the good performance possible in situations with lots of training and feedback (as in TREC-2001), there seems to be a continuum between score-based and classification approaches, depending on the amount of training and feedback available. We conjecture a hybrid approach will be useful to support this continuum.

Arabic Language Retrieval

The Cross-Language Retrieval task at TREC 2002 consisted of bilingual retrieval of Arabic newspaper articles given English topic statements. The document collection was the same as that used in the TREC 2001 CLIR Track. Monolingual submissions were also accepted using Arabic versions of the topics created by human translators. JHU/APL submitted five official runs; one monolingual and four bilingual runs that used only the <title> and <desc> topic fields. We continued to use the HAIRCUT retrieval engine for our experiments, again emphasizing language-neutral approaches to multilingual retrieval.

Tokenization

Over the past year several studies explored alternate representations for indexing Arabic text. Mayfield *et al.* [10] investigated the use of character n-grams for Arabic retrieval in TREC-2001 and found that n-grams of length 4 were most effective. Similarly, Darwish and Oard examined multiple tokenization strategies for retrieval of scanned Arabic documents and concluded that character n-grams of lengths 3 or 4 were the basis for the most successful approach [1]. Linguistic methods of combating Arabic morphology have also been fruitful. Xu et al. [14] investigated several problems unique to Arabic language text retrieval, specifically misspelled words, broken plurals, and infix morphology, and empirically evaluated techniques to overcome them. Larkey et al. [8] investigated methods for effectively stemming Arabic.

Given the successful reports of n-gram based retrieval for Arabic, we opted to continue using them this year. However, we decided to use a combination of tokenization methods in the same term space. We used n-grams of more than one length, and we included space-delimited words. We do perform one minor language-specific function, elimination or replacement of certain Arabic characters. Specifically, we map Alef Maksura to Yeh and Teh Marbuta to Teh, and we eliminate Hamza, Madda and any remaining Arabic letters or symbols that did not appear in a list of 28 letters that we had available.

Recent work in Asian language retrieval has shown that multiple length n-grams can be quite effective, and may result in a 10% relative improvement in mean average precision over the use of single length n-grams [12]. Accordingly, we examined multiple length n-grams. In particular, we construct the set of all 3-grams, 4-grams, and 5-grams that can be generated from a given input sequence.

We initially built several indexes to compare different methods for tokenization. Summary information about each is shown in Table 6.

	# terms	index size
words	539979	254MB
3-grams	27016	441MB
4-grams	225218	766MB
5-grams	1478593	1157MB
6-grams	6081618	1691MB
words + 3/4/5-grams	2876187	2422MB
words + 3/4/5/6-grams	9714673	4038MB

Table 6. Index statistics for the 869 MB, 384K article TREC-2002 Arabic collection.

Using the TREC-2001 CLIR test collection (*i.e.*, Arabic topics 1-25) we compared several knowledge-light methods for indexing Arabic text (see the chart in Figure 1). These experiments used only the <title> and <desc> portions of the topic statements and made use of pseudo-relevance feedback. Plain 4-grams did quite well, but slightly superior performance was found when a hybrid indexing scheme was used. Based on these training experiments, we selected this strategy for TREC-2002.

Thus, our official runs used both words and 3-, 4-, 5-grams to represent text in a single term-space. It should be noted that this tripled the disk space consumed by the index data structures compared to the use of solitary 4-grams; the use of 4-grams alone is probably justified when storage limitations are a concern.

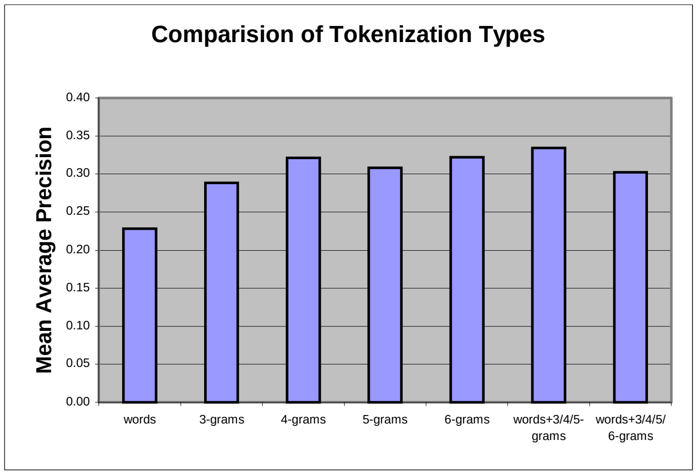

Figure 1. Comparison of tokenization methods using the TREC-2001 CLIR test suite. Mean average precision is plotted. The combination of words plus 3-, 4-, and 5-grams was the best performing approach.

Translation

Although we recently explored efficient methods for translating document representations at the CLEF-2002 evaluation [11], we focused on query translation for our work in the CLIR Track at TREC. We are convinced that the caliber of translation resources has a great effect on bilingual retrieval performance, so we were glad to see the track guidelines stipulate a standard set of resources. However, in several ways the formatting of these resources prevented us from using them in an optimal fashion. In particular, we had hoped to use the English / Arabic parallel texts from the United Nations. We were grateful for the statistical lexicon that was made available by BNN; however, it was of limited use to our system since we do not routinely stem English or Arabic.

Most of our bilingual runs simply relied on machine translation software. However, in an attempt to make use of the BBN statistical lexicon, we derived a surrogate dictionary. We first ran a Porter stemmer to create a set *E* of English words that could produce a given English stem; we also created a set of Arabic words *A*, that created the stems in BBN's lexicon using Kareem Darwish's Al-Stem stemmer. Then, we created an unweighted translation dictionary with entries between each English word in *E* and every word in *A* to which that word *might* be mapped. Queries were translated by substituting all possible translations for a given source language query term, preserving the original query term frequency. Lastly, we performed n-gram processing over the translated queries using only within-word n-grams.

Each of our official submissions used only the <title> and <desc> fields, augmented by pseudo-relevance feedback. For our monolingual Arabic run, *apl11ca1*, we used

- word plus 3-, 4-, and 5-gram indexing
- relevance feedback using queries expanded to 300 terms

Apl11ce1 was our first bilingual run using the English topics. We used the same approach as *apl11ca1*, but used the Almisbar web-based service to create translated queries. We also created a run using the (standard) Ajeeb translator, *apl11ce3*. Mappings derived from the statistical lexicon provided by BBN were used for *apl11ce4*. Finally, hoping that a combination of resources would maximize lexical coverage, and thus retrieval performance, we submitted a run based on merging scores from our two MT-based runs, *apl11ce2*. This run was not our best official run; use of only the standard MT-resource, the Ajeeb translator, was best.

Official results

An overview of APL's five official runs for the Arabic track are shown in Table 7 below.

	Trans Res.	MAP	Recall (5909)	# best	# ≥ median	% mono
apl11ca1	NA	0.3410	4977	3	29	100.0
apl11ce1	Almisbar	0.2427	4396	0	20	71.2
apl11ce2	Almisbar & Ajeeb	0.2571	4488	2	18	75.4
apl11ce3	Ajeeb	0.2658	4444	0	21	77.9
apl11cf1	Stat. Lexicon	0.1777	3645	1	11	52.1

Table 7. Official results for Arabic runs (50 topics). The highlighted rows indicate bilingual runs that used only standard translation resources.

Figure 2 (below) compares our monolingual run against the median of 18 monolingual runs.

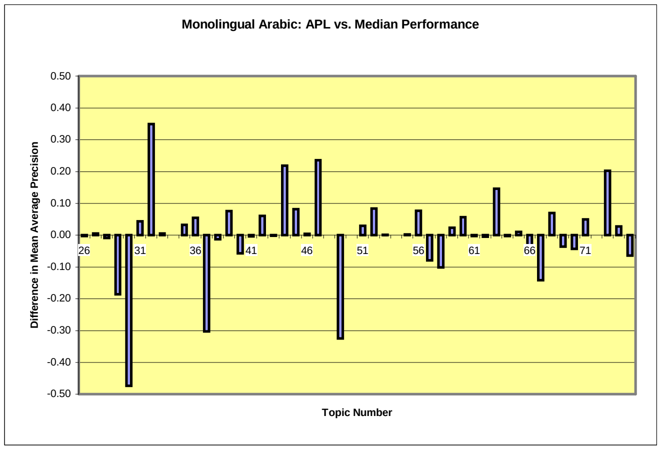

Figure 2. Comb chart for *apl11ca1*

The MT-based runs obtained performance between 71% and 78% of a monolingual baseline in terms of mean average precision; a relative recall at 1000 documents of 88% was found. A precision-recall graph comparing these results is plotted in Figure 3.

Conclusions

This year we participated in two tracks: filtering, and Arabic.

We continued our investigation of using Support Vector Machines (SVMs) to tackle text filtering challenges. We found promise for the use of SVMs for relevance feedback for routing. We plan to further investigate related SVM pseudo-relevance feedback effects on ad hoc retrieval. Our batch results appeared to be about median, and we had many "zero returns," so more remains to be done to tune this approach for low training situations. Perhaps Financial Times data was not similar enough to the evaluation data for use in parameter selection.

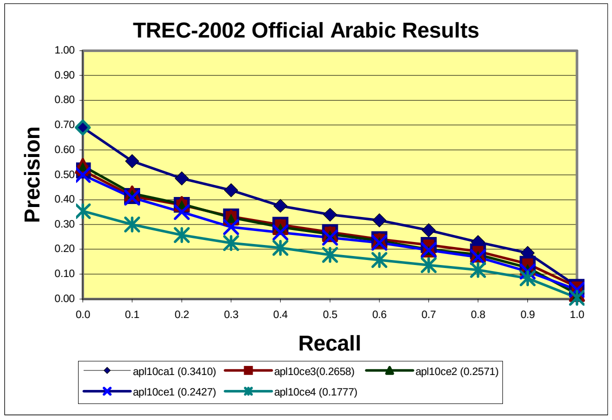

Figure 3. Recall-precision graph for APL's official Arabic track automatic submissions

Our adaptive filtering results were disappointing compared to what we had observed on TREC 2001 adaptive topics, although somewhat expected based on Financial Times parameter-setting experiments. Again, this is related to the much smaller amount of feedback in the track this year. It is possible to make better use of unlabeled (unjudged) data for SVM training, and we hope to revisit this in future experiments.

One thing we have observed in our CLIR work is that it is difficult to define standard translation resources. For example, it has proved difficult this year to separate specific stemming algorithms (and implementations) from some of the standard resources. We also wonder whether cross-system comparisons would be facilitated if participants submitted runs that used only a single translation resource. For the TREC-2001 CLIR guidelines, systems could use any of the three options (dictionary, statistical lexicon, or MT system), thus giving 7 ways to use 'standard' resources.

References

[1] K. Darwish and D. W. Oard, 'Term Selection for Searching Printed Arabic'. In the Proceedings of the25th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR-2002), Tampere, Finland, pp. 261-268, 2002.

[2] S. Dumais, J. Platt, D. Heckerman, M. Sahami, "Inductive Learning Algorithms and Representations for Text Categorization," in Proceedings of the 7th International Conference on Information and Knowledge Management (CIKM 98) (1998).

[3] Marti A. Hearst. Trends and controversies: Support vector machines. IEEE Intelligent Systems, 13(4):18-28, 1998.

[4] D. Hiemstra and A. de Vries, 'Relating the new language models of information retrieval to the traditional retrieval models.' CTIT Technical Report TR-CTIT-00-09, May 2000.

[5] T. Joachims, Making large-Scale SVM Learning Practical Advances in Kernel Methods - Support Vector Learning, B. Schölkopf and C. Burges and A. Smola (ed.), MIT Press, 1999.

[6] T. Joachims, "Text categorization with support vector machines: learning with many relevant features," in *Proc. 10th European Conference on Machine Learning ECML-98* (1998).

[7] W. Kraaij, 'TNO at CLEF-2001'. In Results of the CLEF-2001 Cross-Language System Evaluation Campaign (Working Notes), Darmstadt, Germany, pp. 29-40, 2001.

[8] L. S. Larkey, L. Ballesteros, and M. E. Connell, 'Improving Stemming for Arabic Information Retrieval: Light Stemming and Co-Occurance Analysis'. In the Proceedings of the25th Annu al International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR-2002), Tampere, Finland, pp. 275-282, 2002.

[9] Dave Lewis, personal communication, TREC 2001 Batch Filtering Task Experiments.

[10] James Mayfield, Paul McNamee, Cash Costello, Christine Piatko, and Amit Banerjee, JHU/APL at TREC 2001: Experiments in Filtering and in Arabic, Video, and Web Retrieval. In E. Voorhees and D. Harman (eds.), *Proceedings of the Tenth Text REtrieval Conference (TREC 2001)*, Gaithersburg, Maryland, July 2002.

[11] Paul McNamee and James Mayfield, 'Scalable Multilingual Information Access', Draft version in the *Proceedings of the CLEF-2002 Workshop*, Rome, Italy, 19-20 September 2002

[12] P. McNamee, "Knowledge -light Asian Language Text Retrieval at the NTCIR-3 Workshop," *Working Notes of the 3rd NTCIR Workshop*, 2002.

[13] D. R. H. Miller, T. Leek, and R. M. Schwartz, 'A Hidden Markov Model Information Retrieval System.' In the Proceedin gs of the 22nd International Conference on Research and Development in Information Retrieval (SIGIR-99), pp. 214-221, August 1999.

[14] J. Xu, A. Fraser, and R. Weischedel, 'Emprical Studies in Strategies for Arabic Retrieval'. In the Proceedings of the25th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR-2002), Tampere, Finland, pp. 269-274, 2002.

[15] http://ais.gmd.de/~thorsten/svm_light/

[16] http:/english.ajeeb.com/

[17] http://www.almisbar.com/salam_trans.html

[18] Reuters Corpus, volume 1: English Language, 1996-08-20 to 1997-08-19. We gratefully acknowledge the provision of the research corpus by Reuters Limited; without it, our filtering experiments would not have been possible.