Using Information Extraction in Adaptive Filtering Relevance Feedback

Abstract

We present an evaluation of several different named-entity anchored feature sets for use in document-level and nugget-level adaptive filtering. We show encouraging improvements at the nugget level when using entity wildcards in context as an approach to generalizing from feedback. Specific entity mentions showed very little improvement at the document or nugget level, and in some cases degraded performance significantly.

1. Introduction

Adaptive Filtering (AF) is the task of online prediction of the relevance of articles from a temporal stream of documents. The user provides an initial query or an example document as an indication of his or her information need. The AF system is then supposed to monitor a stream of documents and select some of them for the user's attention. The user goes through these documents and marks some of them as relevant, which are then used by the system to update its query model and present better documents to the user in the future. Adaptive Filtering has been studied extensively and large scale evaluations have been conducted as part of TREC and TDT forums (Fiscus & Duddington, 1998; Fiscus & Wheatley, 2004; Robertson & Hull, 2001; Robertson & Soboroff, 2002). Various learning algorithms have been applied for incremental learning of user's information needs, and Logistic Regression is considered to be the state of the art due to its learning performance as well as computational efficiency in light of frequent updates to the query model (Yang et al., 2005).

In realistic settings, user feedback is often scarce, at least during the early iterations for a new query. Hence, how to make effective use of this feedback is very crucial. We explore this question in terms of two problems faced by AF systems in realistic settings:

Preliminary work. Under review by the International Conference on Machine Learning (ICML). Do not distribute.

1. Users are often interested in tracking specific aspects of a news event. For example, rather than being interested in all news articles about the Iraq War, users might be interested in specific topics like sectarian violence, American casualties, visits by US leaders, troop levels, etc. Given such specific queries, it is more likely that the user's information needs are met by relatively shorter passages rather than full news articles, which place an undue burden on the user to extract the relevant piece of information. However, passage retrieval is generally harder than document retrieval, since the passage itself might not contain terms from the query, and may use anaphora to refer to people and places. Hence, the "context" of a passage might not be clear to the AF system, thus making assessment of relevance much more difficult. Moreover, how to learn effectively from such a piece of feedback with limited context is an open research question.

2. Irrespective of the learning method used to update the query model, the basic idea is to favor terms that occurred in passages that the user liked, and downweight the terms that occured in the passages that the user did not like. However, such a scheme unnecessarily biases the query model towards specific passages seen by the user, and hence, towards specific terms that occurred in those passages. Therefore, the system would try to retrieve new passages that contain those same terms, and hence, the same information. As an example, consider the following passage:

   ```
   ...Parker reported that over
   120 people have died in various
   shootings and explosions in
   Baghdad and Diwaniyah.  36 of
   them...
   ```

   If the user marks this passage as relevant, a naive approach would be to add terms like "Parker", "120", "people", "died", "shooting", "explosions", "baghdad", "diwaniyah" and "36" into the query model. However, it is obvious that the user is neither interested in the person named

> "Parker", nor the specific numbers like "120" and "36". Instead, he or she is more likely to be interested in other acts of violence, reports of casualties and their counts. In fact, future articles might not even contain words like "died" or "explosion", and instead, use words like "killed" or "bombing". In this paper, we explore ways by which the system can infer the user's information needs more intelligently by learning to generalize from the specific words in the feedback.

In this paper, we approach these problems by leveraging recent advances in information extraction technology, specifically, fast and accurate extraction of named entities and event entities, and identifying co-referent entities.

The use of IE techniques in the field of information retrieval has had mixed success in the past. In areas such as question answering, predictive named entity annotations over the corpus have been crucial in moving forward the current state of research. However, in ad-hoc retrieval, information extraction has been of little general use. It is our hypothesis that our abovementioned challenges can be addressed to some degree by utilizing output from information extraction systems.

Enriching the bag-of-words textual representation with named entity annotations could make the task of passage retrieval easier – with more features to judge relevance by, a better estimate of relevance could be made. Moreover, identifying co-referent entities could provide more context than what is directly evident from a passage.

Similarly, a rich set of entity and event types would allow us to generalize from specific terms that occur in text to generic entity types, thereby stepping away from a strict bag-of-words query and document model.

2. Entity Feedback in Document-level AF

In document-level adaptive filtering the task is to retrieve unseen future relevant documents in a stream given previously seen, identified relevant (or non-relevant) documents. Feedback in the form of terms or phrases from the previously identified relevant documents are used to enhance the internal representation of a topical information need. Future documents are then retrieved based on this evolving information need representation.

One potential pitfall of relevance-feedback based approaches to retrieval is that the internal representation of the information need can be overly biased towards those documents first retrieved. This could miss aspects of the original information need that were not present in the first batch of retrieved documents but may still be relevant to the end user. Various relevance feedback methods have been proposed to address queries with multiple aspects such as random-walk models (Collins-Thompson & Callan, 2005) or local context analysis (LCA) (Xu & Croft, 1996). These methods focus on the original query terms and their relationship in the document collection or external resources.

In the current adaptive filtering framework, we investigate the degree to which specific named entities are associated with aspects of a particular query. In order to do this we identify specific *named entities* highly relevant to the information need but not necessarily present in previously retrieved documents. It is our hypothesis that topically relevant named entities could sufficiently broaden the information need representation and enable retrieval of relevant documents outside of the original ranked list.

2.1. Named Entity Relevance

The first step in testing our hypothesis is to establish a method for retrieving named entities relevant to a given topical query. To do this, we follow the method described in (Raghavan et al., 2004) and build language models from the context surrounding each unique named entity in the corpus. These entity-anchored language models, or *entity models*, can be retrieved similar to documents in the language modeling approach to information retrieval, thereby ranking entities relevant to any given query. Our method is slightly different from the Raghavan et. al. approach in the following ways: (1) we are using all entity mentions – named, pronominal and nominal – whereas the previous approach only uses named mentions (2) we use the 3 sentences surrounding the named entity mention as opposed to a sliding term window and (3) we focus only on {`PERSON`} mentions in the corpus.

The process for building the entity models is simple: scan the corpus for all {`PERSON`} mentions, extract the surrounding sentences from each mention and add that text to a pseudo-documents for each unique entity. We will refer to this collection of entity pseudo-documents as an entity corpus. Because we will be evaluating these entity models in an adaptive filtering framework, we build several different entity corpora at each 5-day increment so that we can retrieve entities based only language models built from documents observed up to a given day. The entity model retrieval is done with

the Indri retrieval engine[1], a document retrieval engine combining aspects of the language-modeling and inference network approaches to retrieval (Metzler & Croft, 2004).

2.2. Experimental Setup

To evaluate the effectiveness of our entity relevance measure for feedback in adaptive filtering we simulated an AF system using TDT4 data. To do this, we split the corpus in 5-day increments, and for each time cutoff we formed a weighted bag-of-words (BOW) query using relevant documents before the time cutoff. Using that query, we ranked entities using our entity corpus for the corresponding time slice. Then, with a query created from combining the BOW query and top retrieved entities, we retrieved "unseen" documents from the next time increment. Figure 1 shows a graphical representation of our experimental setup.

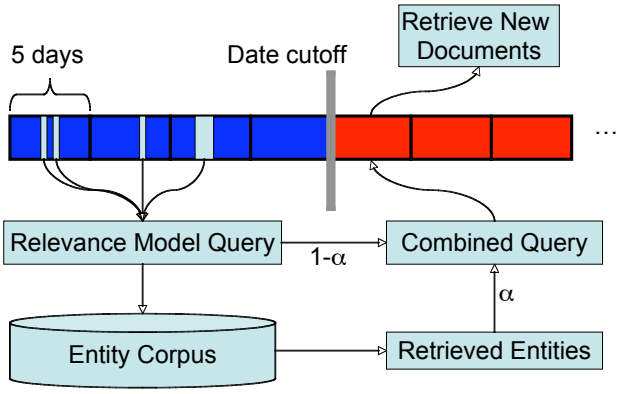

Figure 1. Experimental Setup of the Document-level Evaluation

In order to form the BOW query, we built a *relevance model* from the relevant documents before the time slice using Method 1 as described in (Lavrenko & Croft, 2001). This process poses a risk of generating a query that is unrealistically good because this process utilizes the full set of true relevant documents before the time slice. In order to mitigate against this risk, we attempted to degrade this query by truncating this relevance model at several different levels during testing: 2, 5, 10 and 20 terms.

Entities are retrieved from the entity corpus corresponding to the time cutoff as described in section 2.1. Initial experimentation showed that we can on average retrieve around 2 entities that occur in unseen relevant documents, and in the following experiments we only considered adding 2 entities to the combined query.

The final document ranking function on new documents is given by:

$$P(Q_{comb}|\Theta_D, \alpha) = P(RM|\Theta_D)^{(1-\alpha)} P(ENT_{rel}|\Theta_D)^{\alpha}$$

Where $P(Q_{comb}|\Theta_D, \alpha)$ is the probability of relevance of the combined query given document D and mixing factor α, and Θ_D is the document language model. $P(RM|\Theta_D)$ is the probability of the relevance model to the document:

$$P(RM|\Theta_D) = \prod_{q_i \in RM} P(q_i|\Theta_D)^{w(q_i)}$$

and $P(ENT_{rel}|\Theta_D)$ is the probability of relevance to the top ranked relevant entities to the document:

$$P(ENT_{rel}|\Theta_D) = \prod_{e_i \in ENT_{rel}} P(e_i|\Theta_D)^{w(e_i)}$$

. $w(\cdot)$ is the normalized weight assigned to the tokens or entities, as computed in the relevance model generation or as assigned by the retrieval engine in our entity retrieval. For details on the estimation of the above probabilities and weights, the reader is referred to (Lavrenko & Croft, 2001) and (Metzler & Croft, 2004).

2.3. Results

Figure 2 shows the performance of our simulated AF system using this specific entity-level feedback.

It is clear from this analysis that selecting specific {PERSON} entities to use in document-level adaptive filtering in this way is not an effective approach, and in fact degrades performance as the weight, α is increased on the entity portion of the query. A very slight gain was seen at the lowest weight settings ($\alpha = 0.05$ to 0.10) when the relevance model query was drastically truncated to 2 words, but we do not believe this is a realistic situation and is overly-penalizing the bag-of-words relevance model query.

3. Feedback in Nugget-Level AF

Information presented by the system is useful only if it is previously unseen by the user. This fact is not taken into account by traditional AF systems, whic only focus on returning more relevant documents even if they contain no novel information. Moreover, almost none of the current adaptive filtering evaluation

[1]http://www.lemurproject.org/indri/

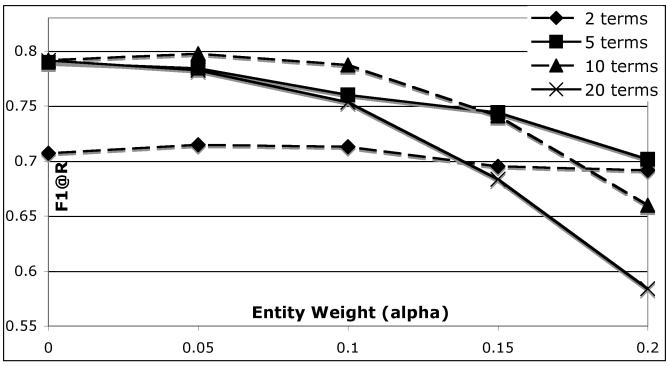

Figure 2. Performance of Specific Entity Feedback in Document-level Adaptive Filtering. F1@R as a function of α, different lines represent different relevance model size. R = the number of known relevant documents for a particular query.

schemes take novelty into account. We feel that this is unrealistic, and follow the adaptive filtering framework and evaluation scheme described in (Yang et al., 2007). Their AF system is based on passage retrieval, and is evaluated at the nugget level. Hence, a system-returned passage receives credit only if it contains a nugget previously unseen by the user.

Given such a strict evaluation which does not favor redundancy, a system must learn to *generalize* from feedback to avoid repetition of information. New feature types might be necessary that go beyond the bag-of-words based feedback, which simply uses specific terms that occur in the passages marked by the user.

We consider four kinds of features – while the first two are based on generalizing from specific terms in text to generic entity types, which we call entity wildcards, the other two are based on the use of specific entities that have been co-referenced across documents.

3.0.1. Entity Wildcards

Lets come back to an example we introduced earlier – suppose the following passage is marked as relevant by the user

```
...Parker reported that over 120
people have died in various shootings
and explosions in Baghdad and
Diwaniyah.  36 of them...
```

A naive approach to using feedback would be to use terms like "parker", "120", "people", "shootings", "explosions", "baghdad", "diwaniyah", and "36" in the query model. However, some of these terms make the query very specific to the information contained in this particular passage. Another view of this passage can be obtained by replacing specific terms with their annotated entity types (e-types) –

```
...{PERSON} reported that over
{CARDINAL} people have died in
various {EVENTVIOLENCE} and
{EVENTVIOLENCE} in {LOCATION} and
{LOCATION}.  {CARDINAL} of them...
```

One can think of this view as a template that corresponds to the user's information need. The task of the AF system, then, is to return other documents containing information that can fill this (or a similar) template.

We can approximate this template-filling task by looking for documents that contain distribution of words and e-types that is similar to that of the query. To do so, we think of queries as well as documents as bags of words as well as e-types. The presence of e-types in queries can be thought of as entity wildcards since these will match any word or phrase that is annotated as that e-type in a future document.

The distribution of weights over e-types defines a distinct *e-type signature* of the query which will depend on the nature of the query. For instance, the signature of a query about casualties and troop levels will have higher weights for {CARDINAL} entities, while a query related to political event may give higher weights to {PERSON} and {EVENTDEMONSTRATION} entities.

Our hypothesis is that the distict e-type signature of a query will tend to favor documents related to the query, without giving undue preference to specific terms from previously marked passages. Hence, this approach would allow us to generalize beyond specific terms and find "related information" in future documents.

To investigate our hypothesis, we prepared a pilot data set in order to analyse the e-type signatures for documents associated with a set of diverse queries that simulated topics from the TDT4 corpus. Four news events that occured between March 11th,2007 to April 16th,2007 were chosen as queries. Two of these were topics related to world politics, while one each was related to entertainment and business. For each of the 4 queries, online news articles were harvested from the Wikinews published at 4 different time instants separated by atleast 2 weeks. At each time instant, up to 4 articles (1000 words) were collected. Thus, this document collection consists of relevant documents at 4 different points of time for a set of 4 queries.

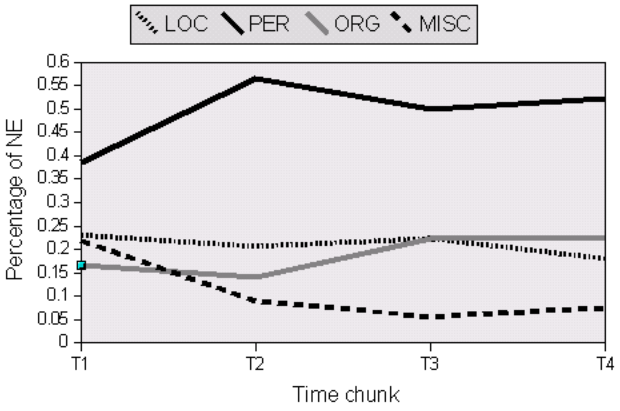

Figure 3. E-type signature for political news story

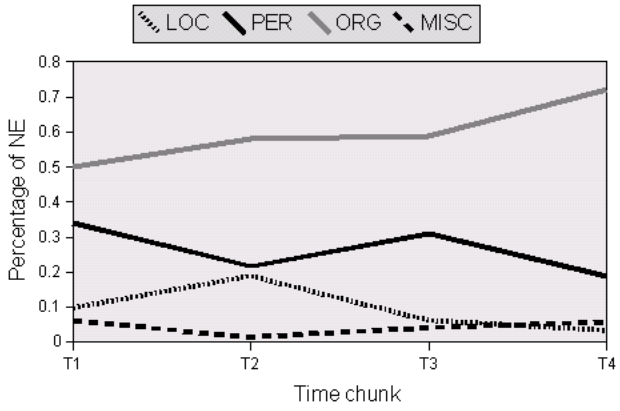

Figure 4. E-type signature for business news story

We used the SNoW Named Entity Tagger to tag the above data for 4 different entity types - a) {PERSON} b) {ORGANIZATION} c) {LOCATION} and d) {MISCELLANEOUS}. We can now view the data along two different dimensions. Firstly, for each query we observe the e-type signatures at different time instants. This allows us to investigate whether a relevant documents are indeed dominated by a particular entity type and if so, how the entity type distribution changes over time. This information would be useful while determining the entities to be used in query expansion. Alternately, we can also view for each of the entity type, the change in its relative frequency for different queries. This further boosts the argument that entity type occurences do not change rapidly with time for a given query.

Figure 3 and 4 show the e-type signatures for 2 events - French Presidential elections and Doubleclick takeover respectively. As one might expect, Fig 3 shows a dominance of {PERSON} entities while fig 4 shows the {ORGANIZATION} entity occuring most frequently. Also, the time series corresponding each of the entity types have limited dynamic range and hence their relative importance (if importance is measured in proportion to occurence) within the query does not change signficantly over time.

3.0.2. Entities in Context

Entity wildcards described in the previous section may not be robust on their own, since they might spuriously match against entities in documents unrelated to the query. For example, a query about number of casualties that gives a high weight to {CARDINAL} entities would also match against the numerous mentions of scores in sports documents. In other words, merely looking for similar distribution of words and e-types might not be enough to approximiate the template-filling task.

We address this problem by associating a context c_i with each e-type e_i added to the query. For every passage marked as relevant by the user, we extract all entities as well as words that appear in a k-word window around the entity. Thus, ‘‘...explosions in Baghdad and Diwaniyah...’’, will lead to the creation of (e_i, c_i) pairs, two of which would be: (EVENTVIOLENCE, {baghdad, diwaniyah}), and (LOCATION, {explosions}), thus matching against future passages that contain mentions of violence near the word “baghdad” or “diwaniyah”, or mentions of other locations near the word “explosions”.

Thus, passages no longer receive a high relevance score for simply containing the same e-types, but according to how well the contexts of e-types match with the corresponding contexts of the same e-types in the query. The context associated with each e-type is represented as a TF-IDF (Term Frequency-Inverse Document Frequency) vector, and the degree of match between two context vectors is determined by their cosine similarity.

3.1. Specific Relevant Entities

3.1.1. Approach I: Information Theoretic Criteria

An additional set of features added to the passage-level queries are specific cross-document co-referenced named {PERSON} entities present in an identified relevant document and also topically relevant to the original query. To identify potential named entities to add into the query, we score the named entities on three different dimensions, listed below. In the following, P_{rel} refers to the identified relevant passages, D_{rel} refers to

the documents containing the relevant passages, and e_i refers to an entity being scored and e_* refers to any entity.

1. **Document-level PMI**: The pointwise mutual information between observing any {PERSON} entity in the relevant documents and observing the entity being scored.

$$dPMI(e_i) = \log \frac{P(e_i \in D_{rel})}{P(e_i)P(e_* \in D_{rel})}$$

2. **Passage-level PMI**: The pointwise mutual information between observing any {PERSON} entity in the relevant passages and observing the entity being scored.

$$pPMI(e_i) = \log \frac{P(e_i \in P_{rel})}{P(e_i)P(e_* \in P_{rel})}$$

3. **Entity Relevance Score**: The score returned by Indri when scoring e_i against the original topical query. Note that the scores returned by Indri are in log-probability space.

All of the above measures are normalized to lie within the $[0, 1]$ interval by exponentiating the score and then applying the sigmoid function. These scores were then averaged to form a measure of each entity's association with previously seen relevant documents and passages as well as its topical relevance to the original query.

3.1.2. Approach - II: Exploiting E-type Signatures

In deciding the entities that must be used for query expansion, we use a combination of three features. Two of these features are similar to term representations in the vector space retrieval model in ad-hoc retrieval systems. However, in the context of adaptive filtering, computation of these features will have to take into account time information as well. The following 3 features were used.

a) Instantaneous Document Frequency (INDF)

Let N be the number of relevant documents returned by the system at current time t and m be the number of relevant documents that contain the Named Entity e. Then the $INDF(e)$ is defined as

$$INDF(e) = m/N \tag{1}$$

As opposed to conventional ad-hoc systems where terms that appear in large number of documents (such as stop-words) are weighed low, in this case the entities that appear in greater number of documents in a given chunk get higher weight. This is because the very indication that the term in question is a named entity almost precludes the possibility of it being a stop word.

b) Average normalized frequency (ANTF)

We define the entity frequency as the ratio of number of occurences of e in a relevant document to the total count of all entities in the document. We then normalize this with respect to the maximum entity frequency for the document to obtain the normalized entity frequency. Finally, the $ANTF(e)$ is obtained by averaging the normalized entity frequency over all the relevant documents at the current time instant. This feature adds to the previous feature by not only taking into account the number of documents that an entity appears in but also the relative frequency of an entity with respect to other entities. The normalizations serve the same purpose as in ad-hoc retrieval systems (to compensate higher term frequencies in longer documents).

c) E-type Signature based weights

In the previous section, we argued that varying e-type signatures indicate that certain entity types are more dominant than others within a query with the exact distribution depending on the nature of the query. In deciding the entities to be used for query expansion, we are faced with the possibility that the use of rare named entity types can reduce precision. It would thus be useful to assign a higher weight to dominant entity types compared to rare entity types.

To compute the weight for each named entity, we first compute the entity type distribution within the relevant documents at each of the past instants. These distributions are averaged to obtain mean entity type distribution. We then weigh each entity in proportion to its entity type dominance as given by the mean entity type distribution, thus ensuring that entities are weighed by their relative importance in the current query. Having computed the above features, the final score $score(e)$ for entity e is computed as

$$score(e) = (w1 * INDF(e) + w2 * ANTF(e)) * w_t(e) \tag{2}$$

where $INDF(e)$ is the Instantaneous Document Frequency of the entity, $ANTF(e)$ is the Averaged Normalized Frequency, $w1$ and $w2$ are the weights asscciated with the two features and $w_t(e)$ is the weight associated with the entity type computed from the mean entity type distribution. In the interest of time and computational resources, for the purpose of this study,

we choose $w1 = w2$. (they may alternately be estimated on a held-out query set)

3.1.3. Choosing the number of entities for query expansion

Having obtained a ranked list of candidate entities, one must decide the number of entities to be used to expand the query. This can be done either by using a threshold on the scores or by using a pre-set number of entities. It would be even more desirable to adjust the number of entities depending on how confident the system is of its entity ranking and/or the quality of the feedback. However, the difficulty in interpreting entity scores also complicates the process of updating their thresholdsand thus it might be more desirable to decide the actual number of entities to be used at each time instant.

For each query, we adapt the threshold based on the number of relevant documents returned by the system. Apart from indicating that the system is performing well, large number of relevant documents returned also provide a greater number of potentially useful named entities to update the query. If r is the number of relevant documents returned, then the number of entities k used in subsequent expansion is given by

$$k = 10 * (1 - 2 * (-0.3^r)) \quad (3)$$

This above functional form which ensures strict upper and lower bounds (10 and 2 respectively) on the number of entities returned also causes a steady increase in k with increase in r. Other than the above intuition, no other criterion informed the choice of this functional form.

3.2. Technical Cores

We use a logistic regression classifier which is considered to be the state of the art algorithm for incremental learning in adaptive filtering (Yang et al., 2005). Each query is considered to be a class that corresponds to the degree of relevance of a passage with respect to the query. The initial query as well as the positive feedback given by the user are used as positive instances, while negative feedback is taken as negative instances for training the classifier. The logisitic regression model is regularized with a Gaussian prior to avoid overfitting on the training data. Further details of using logistic regression for adaptive filtering, as well as computational issues are described in (Yang et al., 2005; Zhang & Yang, 2003).

In addition to the terms, we use the features described in the previous section as part of the feature space for the classifier. The class model (or query model) learned by classifier is a vector of weights that correspond to the contribution of each feature in determining the relevance of a passage to a query.

3.3. Data and Experimental Setup

TDT4 was the benchmark corpus in the TDT2002 and TDT2003 evaluations. The corpus consists of over $90,000$ news articles from multiple newswire sources published between October 2000 and January 2001, in the languages of Arabic, English and Mandarin. We restrict our evaluation to the $23,000$ English documents, for which we have named entity and event entity annotations as well as cross-document co-reference data from IBM's named entity tracker.

We use the 120 queries, associated answer keys, and the evaluation scheme as described in (Yang et al., 2007). We divide the four-month corpus into chunks of 5 days. The AF system is supposed to return a ranked list of 3-sentence passages at the end of each chunk, receive feedback, and then produce new ranked lists for the next chunk, and so on. We believe this is more realistic than a system that returns documents for the user's attention at arbitrary times, and expects feedback before it moves on to the next document. Instead, a user might choose to go through the system's output after every n days, where n can be controlled by the user.

3.4. Results

In Table 1, we present the results in terms of the nugget recall i.e. the number of nuggets that our system was able to retrieve at the end of the four-month period, and in terms of the NDCU score as described in (Yang et al., 2007), average over all queries over all the chunks. We compare the performance of our system based on the use of above-mentioned features with a system can uses no feedback (and hence only ranks based on bag-of-words features), and a system that uses bag-of-words based features for feedback.

The best performance is obtained when using entity wildcards in context, while using entity wildcards without context hurts the performance compared to bag-of-words feedback. Figures 5 and 6 show the scatterplot between the NDCU scores obtained with entity wildcards without and with context, as plotted against bag-of-words feedback. It is evident that many queries are negatively affected by using entity wildcards without context.

Table 1 also reports two additional experimental se-

Table 1. Performance on various feature sets

Feature Set	Nugget Recall	NDCU
No Feedback	0.42	0.24
Bag of words	0.52	0.34
Entity Wildcards	0.47	0.29
Entity Wildcards in Context	**0.55**	**0.38**
Specific Entities - 1	0.49	0.30
Specific Entities - 2F	0.52	0.33
Specific Entities - 2V	0.48	0.29

tups for e-type signature based scoring for specific entities. One of the setups (referred to as Specific Entities 2F in the table) fixed the maximum number of entities to be returned at 10 causing the top 10 (or lesser if fewer entities were present and ranked) entities to be used in query expansion. This was constant across all queries and time chunks. In the second setup (Specific Entities 2V in the table), we used the functional form in equation (3) where the number of entities used in query expansion was a function of the number of relevant documents returned. Our results show that there is very little to choose between the two approaches. This might be due to ad-hoc nature of the assumed relation between entities added and number of relevant documents returned. Only a quantitative analysis of the data can reveal whether such a correlation exists and can be exploited.

Secondly, both these approaches failed to improve over the baseline. Using fixed number of entities managed to score slightly higher recall compared to the bag-of-words feedback.

4. Conclusion

We evaluated several named-entity anchored feature sets for document-level and nugget-level adaptive filtering. The use of entity wilcards in context as an approach to generalizing from feedback showed encouraging improvements compared to traditional bag-of-words feedback approaches. Specific entity mentions showed very little improvement at the document or nugget level, and in some cases degraded performance significantly.

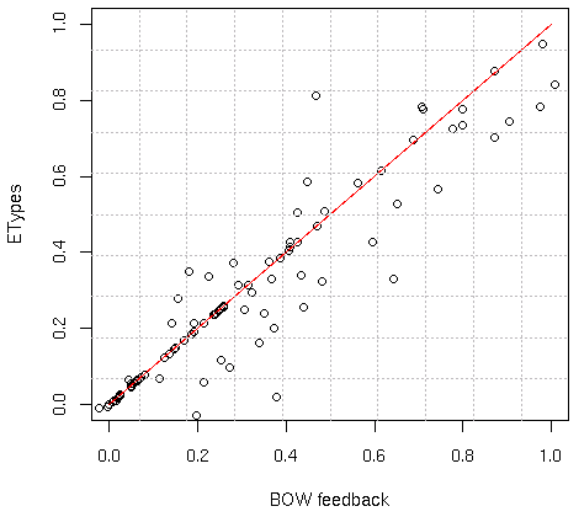

Figure 5. Scatterplot of NDCU scores of queries: Bag of words vs. entity wildcards.

References

Collins-Thompson, K., & Callan, J. (2005). Query expansion using random walk models. *CIKM '05: Proceedings of the 14th ACM international conference on Information and knowledge management* (pp. 704–711). New York, NY, USA: ACM Press.

Fiscus, J., & Duddington, G. (1998). Topic detection and tracking overview. *Topic Detection and Tracking: Event-based Information Organization*, 17–31.

Fiscus, J., & Wheatley, B. (2004). Overview of the TDT 2004 Evaluation and Results. *TDT Workshop. Dec*, 2–3.

Lavrenko, V., & Croft, W. B. (2001). Relevance-based language models. *Research and Development in Information Retrieval* (pp. 120–127).

Metzler, D., & Croft, W. B. (2004). Combining the language model and inference network approaches to retrieval. *Inf. Process. Manage.*, *40*, 735–750.

Raghavan, H., Allan, J., & Mccallum, A. (2004). An exploration of entity models, collective classification and relation description. *KDD'04*.

Robertson, S., & Hull, D. (2001). The TREC-9 filtering track final report. *Proceedings of the 9th Text REtrieval Conference (TREC-9)*, 25–40.

Robertson, S., & Soboroff, I. (2002). The TREC-10 Filtering Track Final Report. *Proceeding of the Tenth Text REtrieval Conference (TREC-10)*, 26–37.

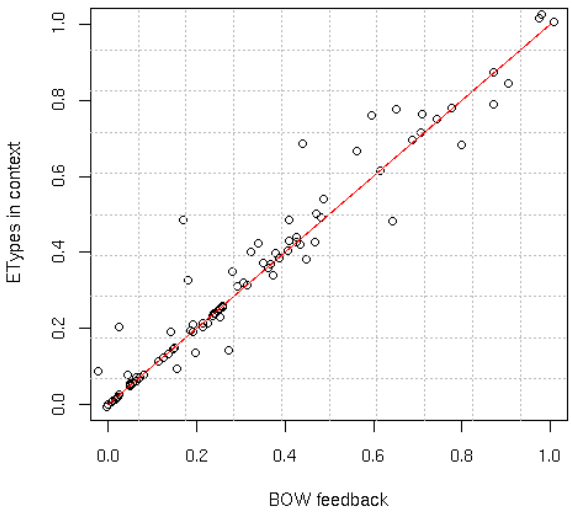

Figure 6. Scatterplot of NDCU scores of queries: Bag of words vs. entity wildcards in context.

Xu, J., & Croft, W. B. (1996). Query expansion using local and global document analysis. *Proceedings of the Nineteenth Annual International ACM SIGIR Conference on Research and Development in Information Retrieval* (pp. 4–11).

Yang, Y., Lad, A., & Lao, N. (2007). Utility-based Information Distillation over Temporally Sequenced Documents. *Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval.*

Yang, Y., Yoo, S., Zhang, J., & Kisiel, B. (2005). Robustness of adaptive filtering methods in a cross-benchmark evaluation. *Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval*, 98–105.

Zhang, J., & Yang, Y. (2003). Robustness of regularized linear classification methods in text categorization. *Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval*, 190–197.