Northeastern University in TREC 2009
Web Track

Shahzad Rajput*, Evangelos Kanoulas‡, Virgil Pavlu*, Javed Aslam*

* College of Computer and Information Science, Northeastern University, Boston, MA, USA
‡ Information Studies Department, University of Sheffield, Sheffield, UK

1 Introduction

In a typical retrieval scenario a user poses a query to a retrieval system in order to satisfy an information need generated during some task the user is undertaking. Retrieval systems access an underline collection of searchable material and rank them according to some definition of relevance of the material to the users request and returns this ranked list to the user. In the case of web search where typical users express their information needs by 2-3 keywords submitted queries often time have ambiguous meanings, representing more than one information need. Given a query, a good retrieval system should be able to satisfy all possible users by ranking documents in a way that their content covers as many information needs as possible.

The primary goal of our Web Track submission is to explore whether named entity tags can be utilized to diversify the returned ranked list of documents. Our hypothesis is that each information need could be represented by a certain named entity tag (or certain combination of them). For instance, in Table 1 one can see the example query taken from the Web Track web page. The query is "physical therapists". The subtopics that correspond to this query are listed in the left column of the table. To illustrate our hypothesis, next to each subtopic, in bold, we have manually identified a possible combination of entity tags that could represent each subtopic/information need.

Further, each document relevant to the original query could also be represented by a set of named entity tags. Instead of attempting to diversify documents based on the distance of their language models over text, we explored whether it is possible to diversify them according to the distance of their language model over entity tags. Entity tags could allow a further abstraction of documents avoiding issues like language mismatch. Our methodology highly depended on two assumptions: (1) retrieval methods based on a bug-of-words representation can retrieve many relevant documents in the top 2,000 positions, and (2) the relevant documents would be diverse enough at the first place. Then using our methodology we could abstract the representation of those documents and diversify the list based on their tag distributions.

A second goal of our Web Track submission was to develop a simple spam filter. By analyzing a small subset of the documents, selected at random from the top 2,000 documents ranked by Indri language model per query over the new ClueWeb09 collection (category B) [1], we observed that 44.5% of them were spam. A large subset of the spam documents were those that contained query terms way too many times. For this purpose, we decided to develop a simple spam filter to remove these documents from the ranked list.

[1] For the rest of the paper, we will refer to this set of approximately 100,000 documents – i.e. 2,000 documents per query for 50 queries – as *DOCSET*.

Query : physical therapist	
Description : The user requires information regarding the profession and the services it provides	
Subtopics	Entity Tags
What does a physical therapist do?	**JOBTITLE**
Where can I find a physical therapist?	**LOCATION/ ORGANIZATION/PERSON**
How much does physical therapy cost per hour?	**MONETARY UNITS/ TIME UNIT/PERSON**
What education or training does a physical therapist require? Where can I obtain this training? How long does it take?	**TIME/ORGANIZATION**
What is the American Physical Therapy Association? What is the URL of its Website?	**ORGANIZATION**
How much do physical therapists earn? What is the starting salary? What is the average salary for an experienced therapist?	**MONETARY UNITS/ DATE UNIT**
What is the difference between a occupational therapist and a physical therapist?	**JOBTITLE**
Information is required regarding physical therapist's assistants. What education do they require? How much do they make?	**ORGANIZATIONS/ MONETARY UNITS/ JOBTITLE**

Table 1: Web track query and subtopics example.

2 Named Entity Tagger

We believe that the problem of diversity is very much related to the structural content of the document. One way to determine the structural content is by identifying the named entities in the text. There are about 70 entities that we tag, some of which are person names, countries, cities, organizations, sports, accidents, crimes, moods, wars, etc. We have build a named entity tagger for this purpose, which identifies entities by using a lookup dictionary. The dictionary is built by the consolidation and modification of the publicly available data.

Before Named Entity Tagging In 1993, Drew Bledsoe and Rick Mirer were the top two picks in the NFL draft.
After Named Entity Tagging In $< year1 >$ 1993 $< year1 >$, $< person1 >$ DREW BLEDSOE $< /person1 >$ and $< person2 >$ RICK MIRER $< /person2 >$ were the top two picks in the $< company1 >$ NFL $< /company1 >$ draft.

Table 2: Example of Tagging

An example of tagged text is shown in Table 2. Once the named entities have been identified, we represent each document in the *DOCSET* as feature vector. Each feature corresponds to the normalized frequency of each named entity in that document. An example of feature vectors can be seen in Table 3. For *doc1* 10% of the entities belong to PersonName, 20% to Country and so on.

Document	PersonName	Country	City	⋯
Doc1	0.1	0.2	0.1	⋯
Doc2	0.1	0.2	0.2	⋯
Doc3	0.2	0.1	0.1	⋯
⋮	⋮	⋮	⋮	⋮

Table 3: Representation of Documents from DOCSET

3 Diversity using entity tags

We assume that the ranked list of documents retrieved by Indri search engine for each query contains documents related to different aspects of the query, which if identified correctly, could help us produce a diverse ranked list.

For this task, we applied two approaches: (1) we clustered the top 2,000 documents based on the distance of their tag language model, we ordered the documents in each cluster by their Indri score and used round-robin algorithm selecting a document at a time from each one of the clusters to populate the list, and (2) we considered the returned by Indri ranked list and rearranged the ranking of documents based on a combination of their original Indri score and the distance of their entity tag and entity value language model between the documents that have already populated the diverse ranked list and the remaining documents of the original ranked list. The two methods are described in detail in the following sections.

3.1 Clustering & Round-Robin

According to this first method, we consider the ranked list of documents returned by Indri. We eliminate the spam documents based on the simple spam filter we will describe in Section 4. We limit ourselves to the top 2,000 ”non-spam” documents. We employ the entity tagger to tag these 2,000 documents and construct the tag feature vector. We utilize the k-means algorithm to cluster documents based on the distribution of tags in each one of them. The number of clusters k was fixed to 7. The documents in each cluster were ranked based on their original Indri score. The final ”diverse” ranked list was produced by fusing the ranked-list of documents from each cluster in a Round-Robin manner. The results of the submitted run, **NEURRWeb300**, can be seen in Tables 4 and 5.

Run	alpha-ndcg5	alpha-ndcg10	alpha-ndcg20	# of queries with alpha-ndcg10 > median(statMAP)
NEURRWeb300	0.133	0.160	0.189	17
NEUDiv1	0.215	0.243	0.278	24
NEUDivW75	0.207	0.220	0.250	22

Table 4: Results for TREC 2009 Web track diversity task (α-NDCG measures)

As it can be observed the clustering approach led to the worst performing algorithm among the three employed in our submission. As an initial diagnostic of what went wrong, we first examined the ranked list of documents returned by Indri to explore whether a large number of diverse relevant documents were found in the first place and then we examined whether documents in certain clusters indeed corresponded to the different query aspects (subtopics). First, we observed that only a very small subset of the Indri ranked list of 2,000 documents were in fact relevant. Further, most of these relevant documents were actually relevant

Run	IA-P5	IA-P10	IA-P20
NEURRWeb300	0.057	0.062	0.063
NEUDiv1	0.126	0.131	0.134
NEUDivW75	0.122	0.119	0.124

Table 5: Results for TREC 2009 Web track diversity task (IA measures)

to one or two query aspects. Thus our first assumption that a ”bug-of-words” approach can return a large number of relevant and diverse documents at the first place was certainly not true. Regarding whether clustering over entity tags could improve diversity, we observed that there were cases, where documents relevant to a certain query aspect were clustered together and separately from documents relevant to a different query cluster. Given, however, the small number of relevant documents and the absence of diversity conclusions can only be tentative at this point.

3.2 Window-greedy maximization of diversity

In a second approach we again considered the ranked list of documents returned by Indri after eliminating the spam ones. We re-ranked the top 2,000 documents by some combination of the produced Indri score and a diversity measure.

Each document is represented by the distribution of entity tags along with the distribution of entity values. For instance, if a document contains the word ”Obama” twice and the word ”Mergel” once, and these words were tagged by the ”president” tag, we counted three times the entity tag ”president”, twice the entity value ”Obama” and once the entity tag ”Mergel”. The counts were then transformed into a Robertson’s TF -like score.

We also incorporate two weighting factors:

- entity tag weighting factor: We manual predefined an importance ratio between entity tags, e.g. tags over query terms are generally more important than other tags, or non-specific ”dates” do not count at all.
- entity value position in document: For each entity value (entity) we keep 5 counts: anchor, title, body-top, body-middle, body-bottom. These 5 categories have predefined importance weights, e.g. anchor counts 4 times more than body-bottom, etc.

The diversity measure was calculated as follows. Let’s assume that we have populated the ”diverse” list with a number of documents. For all the documents already in the ”diverse” list, we aggregated all entity tag and entity value counts. Regarding the documents that are not in the ”diverse list” already, at each round of the algorithm we only consider W of them, the top W documents based on their original Indri scores (i.e., a window of length W). The distance between the entity tag and value distribution of each one of these W documents is computed against the aggregated distribution of entity tags and values of the documents already in the ”diverse list” and the most ”diverse” document is added to the ranked list. The first document listed in the ”diverse” list is the highest ranked document by Indri, since there are no ”prior” documents to measure diversity against.

We varied W, that is the number of documents considered to be added in the ”diverse” ranked list and submitted 2 runs: **NEUDiv1** with W=25, and **NEUDivW75** with W=75. The results can be viewed in Tables 4 and 5. As it can be observed this approach outperforms the clustering one. Further, a window of size 25 leads to better results than a window of size 75. The fact that a small window outperforms both a larger window and the clustering approach seems to highlight the importance of the relevance score over the importance of diversity. Given that most of the top 2,000 documents considered are non-relevant an approach that puts a large weight to diversity will lead to non-relevant documents ranked high in the ”diverse” ranked list, since on average the distance between non-relevant documents is larger than the distance between relevant ones.

4 Adhoc Spam Filter

The spam filter is based on the idea that the term frequency in a spam document does not follow the same distribution as the term frequency in a non-spam document. Essentially, our hypothesis is that the distribution of terms in non-spam documents is more random than the distribution of terms in spam documents. By measuring the randomness of the text in a document with entropy, we attempted to identify all the documents whose entropy is very close to the entropy of the general English. The latter was approximated with a training set of non-spam documents.

We selected about 100 documents, *SPAMDOCSET* from the *DOCSET* at random, which were used as a training set. All of these documents were labeled manually as spam or non-spam. We computed the entropy of the text for each of these documents. The entropy is defined as,

$$H(W) = \sum_{i=1}^{n} p(w_i) \log_2 p(w_i) \tag{1}$$

where *W* is the set of all the words in the document, w_i is the i^{th} word in *W* and p_i is the probability of w_i in *W*.

Thus, the training data is a set of *l* labeled examples $(e_1, y_1), \cdots, (e_l, y_l)$, where e_i denotes the entropy of i^{th} document and y_i is its label: 1 for spam and 0 for non-spam. A linear classifier is then trained over this training data set to obtain the optimal entropy e^* such that e^* minimizes $f(e^*)$, where

$$f(e^*) = \sum_{i=1}^{100} \{y_i - (e_i/e^*)\}^2 \tag{2}$$

In other words, e^* minimizes the square error in the prediction of our spam filter.

For the documents in *SPAMDOCSET*, the value of e^* was found to be 918.399. For this value of e^*, 9.78% of the labeled spam documents were not identified as spam and 3.26% of the labeled non-spam documents were identified as spam. Given that this 3.26% of non-spam documents that were labeled as spam are documents ranked high in the ranked list returned by Indri with a number of terms repeated more than normal, we assumed that some of these terms could be query terms and thus these filtered documents could be relevant documents. Hence, we decided to drop the entropy threshold e^* down to 600 and 300. Based on these two thresholds we submitted two runs, the **NeuLMWeb600** and the **NeuLMWeb300**.

4.1 Results for TREC 2009 Web track adhoc task

In total we submitted the following three runs to TREC 2009 Web track adhoc task:

1. **NeuLMWebBase:** This is the ranked list returned by Lemur search engine. It is used as a baseline to compare results after removing spam.

2. **NeuLMWeb300:** We remove some spam documents from the ranked list returned by Lemur search engine. We use the threshold of 300 to control the spam filtering.

3. **NeuLMWeb600:** We remove some spam documents from the ranked list returned by Lemur search engine. We use the threshold of 600 to control the spam filtering which removes more spam than in the previous case.

The results are summarized in Table 6.

Run	eMAP (MTC)	statMAP	# of queries with statMAP > median(statMAP)
NEULMWebBase	0.042828	0.1763	30
NEULMWeb300	0.043899	0.1865	34
NEULMWeb600	0.044242	0.1869	32

Table 6: Results for TREC 2009 Web track adhoc task