The University of Amsterdam at TREC 2008

Blog, Enterprise, and Relevance Feedback

Krisztian Balog **Edgar Meij** **Wouter Weerkamp** **Jiyin He** **Maarten de Rijke**

ISLA, University of Amsterdam
http://ilps.science.uva.nl/

Abstract: We describe the participation of the University of Amsterdam's ILPS group in the blog, enterprise and relevance feedback track at TREC 2008. Our main preliminary conclusions are that estimating mixture weights for external expansion in blog post retrieval is non-trivial and we need more analysis to find out why it works better for blog distillation than for blog post retrieval. For the relevance feedback track we observe two things: (i) in terms of statMAP, a larger number of judged non-relevant documents improves retrieval effectiveness and (ii) on the TREC Terabyte topics, we can effectively replace the estimates on the judged non-relevant documents with estimations on the document collection. Finally, since the enterprise track did not have any results yet, we only described our participation and do not draw any conclusions.

1 Introduction

This year the Information and Language Processing Systems (ILPS) group of the University of Amsterdam participated in three TREC tracks: blog, enterprise, and the new relevance feedback track. For the blog track our main main emphasis was on topical retrieval of blog posts and of blogs. In our participation in the enterprise track our main aim was to deploy query expansion technique using profiles of top ranked experts in document search and combine our document and candidate models for expert finding. And for the relevance feedback track our goal was to explicitly incorporate non-relevance information in the estimation of query models.

In this paper, we describe our participation for each of the three tracks mentioned above, in three largely independent sections: Section 3 on our blog track participation, Section 4 on our participation in the enterprise track, and Section 5 on our work in the relevance feedback track. We detail the runs we submitted, present the results of the submitted runs, and, where possible, provide an initial analysis of these results. Before doing so, we describe the shared retrieval approach in Section 2.1. We conclude in Section 6.

2 Background

2.1 Retrieval Framework

In this section we describe our general approach for each of the tracks in which we participated this year. We employ a language modeling approach to IR and rank documents by their log-likelihood of being relevant given a query. Without presenting details here we only provide our final formula for ranking documents, and refer the reader to (Balog et al., 2008b) for the steps of deriving this equation:

$$\log P(D|Q) \propto \log P(D) + \textstyle\sum_{t \in Q} P(t|\theta_Q) \cdot \log P(t|\theta_D). \quad (1)$$

Here, both documents and queries are represented as multinomial distributions over terms in the vocabulary, and are referred to as *document model* (θ_D) and *query model* (θ_Q), respectively. The third component of our ranking model is the *document prior* ($P(D)$), which is assumed to be uniform, unless stated otherwise. Note that by using uniform priors, Eq. 1 gives the same ranking as scoring documents by measuring the KL-divergence between the query model θ_Q and each document model θ_D, in which the divergence is negated for ranking purposes (Lafferty and Zhai, 2001).

Unless indicated otherwise, we estimate each document model by:

$$P(t|\theta_D) = (1 - \lambda_D) \cdot P(t|D) + \lambda_D \cdot P(t), \quad (2)$$

where λ_D is a parameter by that we use to tune the amount of smoothing. $P(t|D)$ indicates the maximum likelihood estimate (MLE) of term t on a document, i.e., $P(t|D) = n(t,D)/\sum_{t'} n(t',D)$, and $P(t)$ the MLE on the collection C:

$$P(t) = P(t|C) = \frac{\sum_D n(t,D)}{|C|}. \quad (3)$$

As to the query model θ_Q, we adopt the common approach to linearly interpolate the initial query with an expanded part (Balog et al., 2008b; Kurland et al., 2005; Rocchio, 1971; Zhai and Lafferty, 2001):

$$P(t|\theta_Q) = \lambda_Q P(t|\hat{\theta}_Q) + (1 - \lambda_Q) P(t|Q), \quad (4)$$

where $P(t|Q)$ indicates the MLE on the initial query and the parameter λ_Q controls the amount of interpolation. For each of the tracks in which we participated this year, we looked for ways of improving the query model $\hat{\theta}_Q$.

2.2 Significance testing

Throughout the paper we use the Wilcoxon signed-rank test to test for significant differences between runs. We report on significant increases (or drops) for $p < .01$ using $^{\blacktriangle}$(and $^{\blacktriangledown}$) and for $p < .05$ using $^{\vartriangle}$(and $^{\triangledown}$).

3 Blog Track

Like last year, the blog track consists of two separate tasks: *blog post retrieval* and *blog distillation*. Besides the task of finding topically relevant blog posts, The blog post retrieval task has two further tasks: finding blog posts that contain an opinion on the given topic and determining the polarity of the opinion. To test the opinion-ranking capabilities of participants' systems, participants were asked to rerank five baseline runs based on opinionatedness, besides submitting four full opinion retrieval runs. Our main interest this year lies with the topical retrieval of both blog posts and blogs. We did not participate in the polarity determination and only submitted very basic opinion finding runs.

3.1 Retrieval Models

In the blog post retrieval task we use an out-of-the-box implementation of Indri.[1] Results of previous years showed good overall performance of Indri compared to other systems and besides, it allows for easy use of query models (queries consisting of weighted terms).

In the blog distillation task we use our in-house expert retrieval model (Balog et al., 2006), which we translated to fit the task of blogger retrieval (Balog et al., 2008a; Weerkamp et al., 2008). The main reason for using this model is that we believe blog distillation should be solved using a post index (as opposed to a full blog index). Although last year's blog track showed good performance of blog indexes, we stick to a post index for three reasons: (i) a post index allows for easy incremental updating, (ii) posts are a natural unit for result presentation to the user, and most importantly, (iii) only one index is needed for both post retrieval and blog distillation.

We estimate the probability of a blog *blog* generating query Q as follows:

$$P(Q|\theta_{blog}) = \prod_{t \in Q} P(t|\theta_{blog})^{n(t,Q)}. \quad (5)$$

Next, we smooth the probability of a term given a blog with the background probabilities:

$$P(t|\theta_{blog}) = (1 - \lambda_{blog}) \cdot P(t|blog) + \lambda_{blog} \cdot P(t). \quad (6)$$

Finally, we estimate $P(t|blog)$ as follows:

$$P(t|blog) = \sum_{post \in blog} P(t|post, blog) \cdot P(post|blog). \quad (7)$$

We assume that the post and the blog are conditionally independent, thus $P(t|post, blog) = P(t|post)$, and approximate $P(t|post)$ with the standard maximum likelihood estimate. In Section 3.4 we detail our choices for estimating $p(post|blog)$.

3.2 Query Modeling

For both tasks we experimented with query models using external corpora. In short, we assume that documents in the target collection (the blog collection) are too noisy to generate good query models based on blind relevance feedback. Instead, we use different, less noisy external corpora for expanding our original query. As much of what goes on in the blogosphere is determined by news events, we use a contemporary news corpus AQUAINT-2[2] as our external corpus. Besides this, many queries directed towards blogs and blog posts contain named entities (persons, locations, organizations, products) or general concepts (especially in blog distillation). For this we also look at Wikipedia as an external corpus, since this source contains focused information on many general concepts and named entities.

For two post retrieval runs we use Lavrenko's relevance model 2 (Lavrenko and Croft, 2001) to select the top 10 terms from the top 10 external documents. After selecting weighted new terms, we combine these new query with the original query using Eq. 4.

In two opinion retrieval runs and two blog distillation runs we use a novel, experimental approach to query expansion. We estimate the probability of a expansion term t given the query Q and set of external corpora C:

$$P(t|Q, C) = \sum_{c \in C} \frac{P(t|c, Q) \cdot P(c|Q)}{\sum_{c' \in C} p(c'|Q)} \quad (8)$$

We estimate $p(t|c, Q)$ based on the probability of document d given the query and corpus, and the probability of term t given the document:

$$P(t|c, Q) = \sum_{D \in c; P(D|Q,c) > 0} P(t|D) P(D|Q, c) \quad (9)$$

Next, we estimate $P(D|Q, c)$, the probability of document D given corpus c and query Q:

$$P(D|Q, c) = \prod_{q \in Q} P(q|D) + \frac{n(Q, D) \cdot |Q|^{-1}}{|D|} \quad (10)$$

where $n(Q, D)$ is the count of phrase Q in document D and $P(q|d) = n(q, D) \cdot |D|^{-1}$. Finally, we estimate the probability

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

[2] http://trec.nist.gov/data/qa/2007_qadata/qa.07.guidelines.html#documents

of corpus c given query Q:

$$P(c|Q) = \sum_{D \in c; P(D|Q,c)>0} \frac{P(D|Q,c)}{|D \in c; p(D|Q,c) > 0|} \quad (11)$$

In the remainder of this section we detail our runs and results for blog post retrieval (Section 3.3), the blog distillation task is Section 3.4. We follow with a short discussion on the outcomes in Section 3.5.

3.3 Blog Post Retrieval

As explained in the introduction to this section, we use an out-of-the-box implementation of Indri as our retrieval system. Runs are evaluated on two topic sets: the new 2008 topics alone and the full set of 150 topics (2006–2008). We report on mean average precision (MAP), precision at 5 and 10 documents (P5, P10), and mean reciprocal rank (MRR).

We submitted 6 runs: *uams08n1o1*, *uams08n1o1sp*, *uams08class*, *uams08clspr*, *uams08qm4it1*, and *uams08-qm4it2*. Of these, *uams08n1o1* and *uams08class* are our baseline runs. Runs *uams08n1o1* and *uams08n1o1sp* use the news corpus as an external corpus for query expansion (Section 3.2). Runs *uams08class* and *uams08clspr* use both the news corpus and Wikipedia as external corpora. For the combination with the original query we need to estimate a parameter λ; we use two ways for this. Our baseline approach assigns equal weights to both components (i.e., $\lambda = 0.5$) and is used in runs *uams08n1o1* and *uams08n1o1sp*. The second way tries to estimate λ based on old topics: for each of the old (2006/2007) topics we know the performance of various parameter settings (weights of different corpora) in terms of MAP. We use this information in the following way: for each unseen topic t' we assing a similarity score to seen topics (t) based on overlapping documents in the result lists. Next, we multiply this overlap score by the MAP performance of each mixture setting and determine the "optimal" mixture weights this way. We use this method in runs *uams08class* and *uams08clspr*.

Besides external expansion, the four runs also use credibility priors: based on a combination of 6 credibility indicators (Weerkamp and de Rijke, 2008), we estimate the prior probability of the blog post being relevant. Since all runs use the same priors, we cannot determine its effectiveness here.

For runs *uams08qm4it1* and *uams08qm4it2* we use the model described in Section 3.2: The first run uses both a news corpus and Wikipedia, the second run uses a news corpus and the post index (treated as external corpus).

Looking at opinion retrieval, we explore the use of an opinionated prior. To construct this prior we use strongly "opinionated" terms from the OpinionFinder system[3] and calculate for each post the ratio of opinionated terms to the total number of terms. We use this prior on top of our two baseline runs *uams08n1o1* and *uams08class*, to come to runs *uams08n1o1sp* and *uams08clspr*.

3.3.1 Analysis and Discussion

Run	MAP	P5	P10	MRR
All topics				
uams08n1o1	0.3329	0.5987	0.5693	0.7309
uams08n1o1sp	**0.3351**$^{\blacktriangle}$	**0.6040**	**0.5687**	0.7275
uams08class	0.3297	0.5840	0.5660	**0.7377**
uams08clspr	0.3323$^{\blacktriangle}$	0.5853	0.5647	0.7349
uams08qm4it1	0.2633$^{\blacktriangledown}$	0.4747$^{\blacktriangledown}$	0.4620$^{\blacktriangledown}$	0.6007$^{\blacktriangledown}$
uams08qm4it2	0.1969$^{\blacktriangledown}$	0.3480$^{\blacktriangledown}$	0.3587$^{\blacktriangledown}$	0.4539$^{\blacktriangledown}$
2008 topics				
uams08n1o1	0.3797	0.7080	**0.6620**	**0.8052**
uams08n1o1sp	**0.3823**$^{\blacktriangle}$	**0.7120**	0.6580	**0.8052**
uams08class	0.3685	0.6680	0.6420	0.7852
uams08clspr	0.3715$^{\blacktriangle}$	0.6640	0.6400	0.7852
uams08qm4it1	0.2927$^{\blacktriangledown}$	0.5360$^{\blacktriangledown}$	0.5300$^{\blacktriangledown}$	0.6567$^{\blacktriangledown}$
uams08qm4it2	0.2122$^{\blacktriangledown}$	0.4120$^{\blacktriangledown}$	0.4120$^{\blacktriangledown}$	0.5431$^{\blacktriangledown}$

Table 1: Opinion results on the blog post retrieval task. Significance of *uams08clspr* and *uams08n1o1sp* tested against their baselines, other runs tested against the first run, *uams08n1o1*.

Run	MAP	P5	P10	MRR
All topics				
uams08n1o1	0.4350	**0.7680**	0.7480	**0.8464**
uams08n1o1sp	**0.4366**$^{\blacktriangle}$	0.7667	0.7473	0.8419
uams08class	0.4313	0.7507	**0.7493**	0.8439
uams08clspr	0.4332$^{\blacktriangle}$	0.7520	0.7473	0.8441
uams08qm4it1	0.3627$^{\blacktriangledown}$	0.6800$^{\blacktriangledown}$	0.6713$^{\blacktriangledown}$	0.7780$^{\blacktriangledown}$
uams08qm4it2	0.2745$^{\blacktriangledown}$	0.5760$^{\blacktriangledown}$	0.5740$^{\blacktriangledown}$	0.6869$^{\blacktriangledown}$
2008 topics				
uams08n1o1	0.4644	**0.8040**	**0.7620**	**0.8892**
uams08n1o1sp	**0.4661**$^{\blacktriangle}$	0.8000	**0.7620**	**0.8892**
uams08class	0.4494	0.7680	0.7480	0.8358
uams08clspr	0.4513$^{\blacktriangle}$	0.7720	0.7500	0.8408
uams08qm4it1	0.3734$^{\blacktriangledown}$	0.6720$^{\blacktriangledown}$	0.6600$^{\blacktriangledown}$	0.8052$^{\triangledown}$
uams08qm4it2	0.2606$^{\blacktriangledown}$	0.5480$^{\blacktriangledown}$	0.5380$^{\blacktriangledown}$	0.6981$^{\blacktriangledown}$

Table 2: Topical results on the blog post retrieval task. Significance of *uams08clspr* and *uams08n1o1sp* tested against their baselines, other runs tested against the first run, *uams08n1o1*.

From the results in Tables 1 and 2 we have three initial observations: (i) The runs using the method for combining external corpora introduced in Section 3.2 (i.e., *uams08qm4it1* and *uams08qm4it2*) perform significantly worse than runs using relevance models and a linear combination of the expanded query and original query (*uams08n1o1* and *uams08class*). (ii) Looking at the runs using relevance models to construct query models (*uams08n1o1* and *uams08class*), we see that estimating the relative importance of the original query is not easy: the simple baseline approach ($\lambda = 0.5$) outperforms the slightly

[3] `http://www.cs.pitt.edu/mpqa/`

more advanced per-topic estimation. (iii) The runs using opinion priors (*uams08n1o1sp* and *uams08clspr*) significantly outperform their baseline counterparts in terms of MAP, not only on opinion retrieval, but also on topical retrieval.

3.4 Blog Distillation

Our blog distillation model allows for the estimation of the importance of individual posts to a blog, i.e., estimating association strengths between posts and their blog ($p(post|blog)$ in Eq. 7). Based on previous experiments (Weerkamp et al., 2008) and additional tests on the 2007 topics we use a combination of blog features to estimate this association strength: post length, recency, and number of comments. On top of this, we noticed that using information from the post title is an important indicator of relevance in the blog distillation task. To be able to use this information, we perform a linear combination between runs on the full post index and runs on a title-only index. This run is our baseline run, *uams08bl*.

We again experiment with expansion on external corpora using the novel method introduced in Section 3.2. In run *uams08nw* we use the news corpus and Wikipedia, in run *uams08pnw* we also use the post index as external corpus. The difference with the baseline is that we do not use the combination with the title-only index: for this submission we would like to look at the influence of the query expansion and scores of the two runs (using query expansion and the title-only run) are in a very different range, calling for other, more suitable ways of combining these scores.

The final run we submitted, *uams08nonr* is a highly experimental run: an important aspect of the blog distillation task is to return not just blogs that mention this topic, but mention it quite often. In that sense, we do not only want to determine the relevance of the blog for a given topic, but also the non-relevance for that topic (i.e. relevant regarding different topics). We tried to estimate this by looking at the performance of blogs on the 2007 topics and use this as indicator of non-relevance (assuming the 2008 topics are different from the 2007 topics); the relevance score of a blog (Eq. 5) is divided by the average relevance score of that blog on all 2007 topics. A blog with a high relevance score and low relevance scores on other topics will get a score (and rank) boost.

3.4.1 Analysis and Discussion

The results of our submitted runs, plus the evaluation of one additional run are presented in Table 3. The *baseline* run is similar to run *uams08bl*, except that we left out the combination with the title-only run. The results show some interesting things: (i) The experimental run using "non-relevance" fails completely, indicating we need different ways of incorporating this notion of non-relevance. (ii) Our baseline (*uams08bl*) is a pretty strong baseline and cannot be beaten by the other runs (except on MRR by *baseline*). (iii) Query expansion can improve over the absolute baseline in terms of MAP, but still performs less than the combination with the titles.

Run	**MAP**	**P5**	**P10**	**MRR**
baseline	0.2567	0.4480	0.4180	**0.7298**
uams08bl	**0.2638**$^{\triangle}$	**0.4600**	**0.4200**	0.7294
uams08nonr	0.0257$^{\blacktriangledown}$	0.1000$^{\blacktriangledown}$	0.0900$^{\blacktriangledown}$	0.2393$^{\blacktriangledown}$
uams08nw	0.2489	0.4080	0.3660	0.6515
uams08pnw	0.2620	0.4080	0.3900	0.6303$^{\triangledown}$

Table 3: Results on the blog distillation task. Significance tested against *baseline*.

3.5 Conclusions

In this year's participation in the blog track we mainly explored different ways of using external corpora to expand the original query. In the blog post retrieval task we did not succeed in improving over a simple baseline (equal weights for both the expanded and original query) and we need a thorough analysis to find out why this did not work. For the same task, further investigation is needed to determine the effectiveness of the credibility priors and to see what happens when the opinion prior is applied.

In the blog distillation task we tried to improve over our (strong) baseline using external expansion. Since this baseline also uses information from the title explicitly, it is hard to determine why the expanded runs do not improve over the baseline. Compared to a baseline without the title component, we see an improvement for the run using expansion on the combination of news, Wikipedia and blog posts. For this task, further research into the combination of title and full post components is needed, as well as the combination with expanded queries. The run that tried to capture non-relevance of a blog failed, but exploring this area further could lead to significant improvements over a baseline that looks only at "relevance."

Finally, looking at the two tasks combined, we see that query expansion on the blog distillation task is much more effective than on the blog post retrieval task. Further analysis is needed to find out why this difference occurs.

4 Enterprise Track

Similarly to last year, the enterprise track features two separate tasks: *document search* and *expert finding*. For both tasks, we experiment with a query expansion technique using profiles of top ranked experts and with encoding query-independent features as (document and candidate) priors. Further, concerning the expert search task we consider both candidate- and document-based models, as well as their combination. Since results were not available at the time of writing, we report only on the submitted runs.

4.1 Document search

The aim of the document search task is to retrieve documents that help a science communicator within an organization (in this case CSIRO) create an overview page for a given topical area. Relevant documents are therefore documents that discuss the given topic in detail and not the ones that only touch on the topic. Our aims for the document search task was to experiment with query models and with using a document prior.

4.1.1 Query models

We consider constructing the query model from three components according to the following equation:

$$\begin{aligned} P(t|\theta_Q) &= \lambda_Q \cdot P(t|\hat{\theta}_Q) \\ &+ \mu \cdot P(t|\check{\theta}_Q) \\ &+ (1-\lambda_Q-\mu)\cdot P(t|Q). \end{aligned} \quad (12)$$

Here, $P(t|\hat{\theta}_Q)$ is estimated using relevance models (method 2) of Lavrenko and Croft (2001), $P(t|\check{\theta}_Q)$ is constructed from profiles of candidate experts, and $P(t|Q)$ is the initial query.

Sampling expansion terms from expert profiles is performed using the following algorithm. First, we rank experts using expert finding Model 1B described in Section 4.2.1. Then, we obtain $P(t|S)$ by taking terms from the profiles of the top ranked M experts:

$$P(t|S) = \sum_{ca \in M} P(t|\theta_{ca}) \cdot P(ca|S), \quad (13)$$

where $p(t|\theta_{ca})$ is the probability of term t given the candidate's language model, and $P(ca|S)$ is proportional to how likely candidate ca is an expert, given the top M experts:

$$P(ca|S) = \frac{P(ca|Q)}{\sum_{ca' \in M} P(ca'|Q)}. \quad (14)$$

Calculating the sampling distribution $P(t|S)$ can be viewed as the following generative process:

1. Let the set of candidate experts $\{ca \in M\}$ be given
2. Select a candidate ca from this set with probability $P(ca|S)$.
3. From this candidate, generate the term t with probability $P(t|\theta_{ca})$

Finally, we take the top K terms from $P(t|S)$ to form $P(t|\check{\theta}_Q)$.

4.1.2 Document priors

Since we are looking for key pages, our intuition is that these pages have shorter URLs than non-key pages. This heuristic is encoded as document priors ($P(D)$ in Eq. 1):

$$P(D) \propto C - URL\_LENGTH(D), \quad (15)$$

where C is a constant (here set to 255), and $URL\_LENGTH(D)$ denotes the length of the URL (number of characters) of document D.

4.1.3 Runs

We submitted the following runs, all of which were automatic. To estimate the parameters of our models, such as the number of feedback documents and terms, and the interpolation weights in Eq. 12 we use last year's topic set.

UvA08DSbl the baseline run; uses only the initial query ($\lambda_Q = \mu = 0$) and document priors are set to be uniform.

UvA08DSbfb blind feedback run; query model uses the relevance model component ($\lambda_Q = 0.5$, top 10 terms from top 5 documents) but not the expert profiles component ($\mu = 0$). Document priors are set to be uniform.

UvA08DSexp query expansion using expert profiles; same as UvA08DSbfb but with $\lambda_Q = 0.4$ and using candidate profiles for expansion ($\mu = 0.2$, top 10 experts from top 5 experts). Document priors are set to be uniform.

UvA08DSall all features; query model is constructed as in UvA08DSexp and document priors are set based on URL character length.

4.2 Expert finding

Our approach to ranking candidates is as follows:

$$\log P(ca|Q) \propto \log P(ca) + \log P(Q|ca), \quad (16)$$

where $P(ca)$ is the *a priori* probability of the candidate ca being an expert, and $P(Q|ca)$ is the probability of ca generating the query Q. Our choice of setting $P(ca)$ is presented in Section 4.2.3. For estimating $P(Q|ca)$ we consider both candidate (Section 4.2.1) and document (Section 4.2.2) models.

4.2.1 Candidate model (Model 1B)

We use a proximity-based version of the candidate model, referred to as *Model 1B*. Here, a language model θ_{ca} is inferred for each candidate and the log-query-likelihood of a candidate producing the query is obtained as follows:

$$\log P(Q|ca) = \sum_{t \in Q} P(t|\theta_Q) + \log P(t|\theta_{ca}), \quad (17)$$

where $P(t|\theta_{ca})$ is a linear interpolation between an empirical candidate model ($P(t|ca)$) and the background (collection) language model ($P(t)$):

$$P(t|\theta_{ca}) = (1-\lambda_{ca}) \cdot P(t|ca) + \lambda_{ca} \cdot P(t). \quad (18)$$

The probability $P(t|ca)$ is estimated based on the co-occurrance of the term t and candidate ca in a particular window size w (which was set to 125 based on empirical exploration). The model we use corresponds to Model 1B with semantic document-candidate associations (SEM) described in (Balog and de Rijke, 2008).

We also used a web-based variation of Model 1B, where the candidate's name was used as a query, issued to a web search engine API (in our case: Yahoo!). Then, terms from top 100 result snippets were used to construct $P(t|ca)$.

4.2.2 Document model (Model 2)

Using a document-based model the estimation of $P(Q|ca)$ is goes as follows:

$$P(Q|ca) = \sum_D P(Q|D) \cdot P(D|ca). \qquad (19)$$

We use the approach developed for ranking documents to estimate $P(Q|D)$ (see Section 4.1). As to $P(D|ca)$, we use the semantic relatedness of document D and candidate ca; see Section 6.3.5 in (Balog, 2008) for details.

4.2.3 Candidate priors

We use candidate priors to filter out science communicators. To do this, we first extracted names and positions from contact boxes of CSIRO pages. Then, science communicators (SC) (often called *communication officer/manager/advisor* or *manager public affairs communication*) were assigned value the 0, while all other people were assigned the value 1 of candidate prior:

$$p(ca) = \begin{cases} 1, & ca \notin SC, \\ 0, & ca \in SC. \end{cases} \qquad (20)$$

4.2.4 Runs

We submitted the following 4 runs:

UvA08ESm1b Model 1B using the initial query (without expansion).

UvA08ESm2all Model 2 using expanded query models and all document search features (on top of document search run `UvA08DSall`)

UvA08EScomb linear combination of Model 1B (with weight 0.7) and Model 2 (with weight 0.3). Both models use the initial query (without expansion).

UvA08ESweb linear combination of the run `UvA08EScomb` (with weight 0.75) and the Web-based variation of Model 1B (with weight 0.25). The web run uses the query model from `UvA08DSexp`.

We employed candidate priors as described in Section 4.2.3 for all runs.

5 Relevance Feedback Track

Our chief aim for participating in this year's TREC Relevance Feedback track is to extend previous approaches, such as the one proposed by Lavrenko and Croft (2001), by explicitly incorporating non-relevance information. Such negative evidence is usually assumed to be implicit, i.e. in the case of estimating a model from some (pseudo-)relevant data, the absence of terms indicates their non-relevance status. This means, in a language modeling setting and for the sets of relevant documents R and non-relevant documents $\neg R$, $P(t|\theta_{\neg R}) = 1 - P(t|\theta_R)$. The TREC Relevance Feedback track gives us the opportunity to develop and evaluate models which explicitly capture non-relevance information and we participated to answer the following research questions. Can non-relevance information be effectively modeled to improve the estimation of a query model? Given our model, what is the effect of the relative size of the set of non-relevant documents with respect to the relevant documents on retrieval effectiveness? And, finally, we ask the question whether and when explicit non-relevance information helps. In other words, what are the effects when we substitute the estimates on the non-relevant documents with more general estimates, such as from the collection. Some previous work has already experimented with using negative weights for non-relevance information, either in an ad-hoc or more principled fashion, with mixed results (Dunlop, 1997; Ide, 1971; Wang et al., 2008; Wong et al., 2008).

The model we propose leverages the *distance* between each relevant document and the set of non-relevant documents, by penalizing terms that occur frequently in the latter, similar to the intuitions described by Wang et al. (2008). Instead of subtracting probabilities, however, we take a more principled approach based on the Normalized Log Likelihood Ratio (NLLR). Moreover, similar to other pseudo-relevance feedback approaches, such as the one proposed by Lavrenko and Croft (2001), we reward terms that appear frequently in the individual relevant documents. Although the NLLR is not a true distance between distributions (since it does not satisfy the triangle equality), we consider it to be a useful candidate for measuring the (dis)similarity between two probability distributions.

5.1 Modeling non-relevance

Kraaij (2004) defines the NLLR measure as being equivalent to determining the negative KL-divergence for document retrieval. It is formulated as:

$$\mathrm{NLLR}(Q|D) = H(\theta_Q, \theta_C) - H(\theta_Q, \theta_D), \qquad (21)$$

where $H(\theta, \theta')$ is the cross-entropy between two multinomial language models:

$$\begin{aligned} H(\theta, \theta') &= H(\theta) + \mathrm{KL}(\theta || \theta') \\ &= -\sum_t P(t|\theta) \log P(t|\theta) + \end{aligned}$$

$$\begin{aligned} & \sum_t P(t|\theta)\log\frac{P(t|\theta)}{P(t|\theta')} \\ = & \ -\sum_t P(t|\theta)\log P(t|\theta'). \end{aligned}$$

Eq. 21 can be interpreted as the relationship between two language models θ_Q and θ_D, normalized by a third language model θ_C (these three models are estimated using Eq. 4, Eq. 2, and Eq. 3 respectively). The NLLR is a measure of average surprise; the better a document model 'fits' a query distribution, the higher the score will be; $H(\theta_Q,\theta_D)$ will be smaller than $H(\theta_Q,\theta_C)$ for relevant documents. In other words, the smaller the cross entropy between the query and document model (i.e., when the document language model better fits the observations from the query language model), the higher it will be ranked.

Based on the NLLR measure, we have developed the following model by which we estimate $P(t|\hat{\theta}_Q)$ in Eq. 4. The intuition is to determine for each term, the probability that it was sampled from each relevant document as well as the probability that it was sampled from the set of non-relevant documents:

$$P(t|\hat{\theta}_Q) \quad \propto \quad \sum_{D\in R} P(t|\theta_D)P(\theta_D|\theta_R),$$

We weigh each term by the divergence from R to $\neg R$ and its importance in the current document by setting:

$$P(\theta_D|\theta_R) = \frac{\text{NLLR}(D|R)}{\sum_{D'}\text{NLLR}(D'|R)}, \qquad (22)$$

where

$$\begin{aligned} \text{NLLR}(D|R) &= H(\theta_D,\theta_{\neg R}) - H(\theta_R,\theta_D) \qquad (23) \\ &= \sum_t P(t|\theta_D)\log\frac{P(t|\theta_R)}{P(t|\theta_{\neg R})} \\ &= \sum_t P(t|\theta_D)\log\frac{(1-\delta_1)P(t|R)+\delta_1 P(t)}{(1-\delta_2)P(t|\neg R)+\delta_2 P(t)}. \end{aligned}$$

The δ parameters provide us with the means to control the individual influence of each set of relevant and non-relevant documents versus a background model. $P(t|R)$ and $P(t|\neg R)$ are estimated by considering the MLE on the documents in the respective set, i.e., for the set of relevant documents R:

$$P(t|R) = \frac{\sum_{D\in R} P(t|D)}{|R|}.$$

5.2 Runs

We have submitted 2 runs, each consisting of 5 separate runs (one for each set of provided relevance judgements). The capital letters in each run indicate the relevance judgements per topic used for that run: (A) no relevance judgements, (B) 3 relevant documents, (C) 3 relevant and 3 non-relevant documents, (D) 10 judged documents (division unknown), (E) large set of judgements (division and number unknown).

	Set	MAP	P5	P10
	A	0.1364	0.2516	0.2452
met6	B	**0.1732**$^{\triangle}$	0.2645	0.2677
met6	C	0.1568	**0.3484**	**0.3129**
met6	D	0.1584	0.3097	**0.3129**
met6	E	0.1689	0.2645	0.2677
met9	B	0.1769$^{\triangle}$	0.3161	0.3194
met9	C	0.1699$^{\triangle}$	0.3161	0.3032
met9	D	0.1738$^{\triangle}$	**0.4000**$^{\triangle}$	**0.3710**$^{\triangle}$
met9	E	**0.1959**$^{\triangle}$	0.2903	0.2871

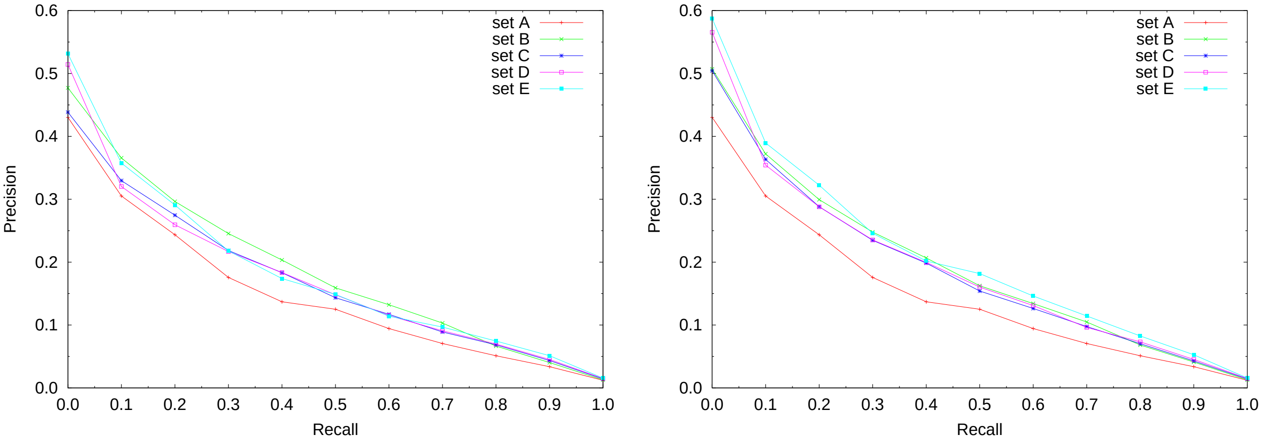
Table 4: Evaluation on the 31 TREC Terabyte topics (top10): significance tested against the baseline (set A).

We have followed the following intuition for our submissions: given that we have knowledge on which documents are relevant and not relevant to the query, can we use this information to obtain a better estimate of our query model? We hypothesize that our model gains the most when the set of non-relevant documents is large enough to give a proper estimate on non-relevance. We expect the background collection to be a better estimate of non-relevance when the set of judged non-relevant documents is small, but expect to obtain an increasingly good estimate using the non-relevant documents as the size of this set increases. Thus, we compare our model using explicit non-relevance information to the same model using the collection as a non-relevance model, by submitting two distinct runs: **met6**, using the set of non-relevant documents, and **met9**, using only the collection ($\delta_2 = 1$, viz. Eq. 23).

Preprocessing and Parameter settings We did not perform any preprocessing of the data besides standard stopword removal and stemming using a Porter stemmer. For our models we need to estimate four parameters: δ_1, δ_2, λ_D, and λ_Q. We have used the odd numbered topics from the TREC Terabyte track (topics 701-850) and from the TREC Million Query track (topics 1-10000) as training data. We have performed sweeps (with steps of 0.1) over possible values for these parameters and select the parameter settings with the highest resulting MAP scores. The resulting set of parameters that we have used for **met6** is given by: $\lambda_D = 0.2$, $\lambda_Q = 0.4$, $\delta_1 = 0.2$, and $\delta_2 = 0.6$. The settings for **met9** are: $\lambda_D = 0.2$, $\lambda_Q = 0.4$, and $\delta_1 = 0.2$.

5.3 Results and Discussion

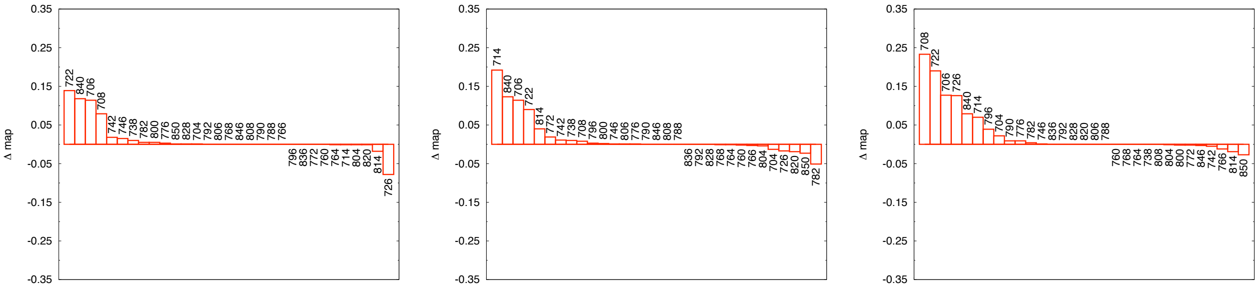
The results of our 10 individual runs are listed in Table 4 and Table 5. Note that the baseline runs (set A) are the same for both methods, since neither uses (non-)relevance information. The same holds for set B: in this set only relevant in-



(a) **met6**

(b) **met9**

Figure 1: Precision-recall plots of **met6** (a) and **met9** (b) on the various feedback sets and the 31 TREC Terabyte topics (top10).



(a) Set C

(b) Set D

(c) Set E

Figure 2: Per topic difference in MAP between **met6** and **met9** on the 31 TREC Terabyte topics and the various sets of relevance feedback information (a positive value indicates that **met9** outperforms **met6** and vice versa). The labels indicate the respective topic identifiers.

formation is available and the two methods should therefore result in the same scores. Due to a small bug in the implementation, however, parameter δ_2 was not properly normalized, causing a slight difference in the retrieval results for **met6** on set B.

As stated earlier, we submitted our runs to explore three main research questions:

- Can non-relevance information be effectively modeled to improve the estimation of a query model?
- What is the effect of the relative size of the set of non-relevant documents with respect to the relevant documents on retrieval effectiveness?
- What are the effects when we substitute the estimates on the non-relevant documents with more general estimates, such as from the collection.

The results reported in Table 4 and Figure 1 with respect to **met6** give an answer to the first question. In all conditions, i.e. in all three measures as well as different settings of relevance feedback sets, the retrieval performance improves over the baseline, which confirms that our model can effectively incorporate non-relevance information for query modeling. Given a limited amount of non-relevant documents (sets C and D), our model especially improves early precision, although not significantly. A larger amount of non-relevant documents (set E) decreases overall retrieval effectiveness. From Figure 1a we observe that set E only outperforms the other sets at the very ends of the graph. Figure 2 shows a per-topic breakdown of the difference in MAP between the two submitted runs. We observe that most topics are helped more using the collection-based estimates. We have to conclude that, for the TREC Terabyte topics, the estimation on the collection yields the highest retrieval performance and is thus a better estimate of non-relevance than the judged non-relevant documents.

When we zoom out and look at the full range of available topics (Table 5), we observe that both models improve statMAP over the baseline (set A) for the full set of topics. When the feedback set is small, **met9** improves statMAP

	setA	setB	setC	setD	setE
met6	0.2289	0.2595▲	0.2750▲	0.2758▲	**0.2822**▲
met9	0.2289	0.2608▲	0.2787▲	0.2777▲	**0.2810**▲

Table 5: Evaluation with statMAP: significance tested against baseline (set A).

more effectively than **met6**, i.e. the background model is performing better than the non-relevant documents. On the largest set of feedback documents (set E) **met6** obtains the highest statMAP score (although the difference with **met9** is not significantly different for this set, tested using a Wilcoxon sign rank test). The difference does seem to suggest that the amount of non-relevance information needs to reach a certain size to outperform the estimation on the collection. Since we select the terms that are most likely to be sampled from the distribution of the relevant documents rather than non-relevant documents, it is crucial that the underlying relevant and non-relevant distributions can be accurately estimated. While the relevant documents are topically concentrated, i.e. they are all related to a given query, the non-relevant documents can be topically diverse and therefore more difficult to be estimated when the number of examples is limited. The background information is generally a good approximation of the distribution of non-relevant documents, given that most of the documents in the collections are not relevant. On the other hand, as the size of the set of non-relevant examples increases, especially the query-specific top-ranked non-relevant documents, we can more accurately estimate the true distribution of the non-relevant information, which enables our model to have more discriminative power. Where this cut-off point lies remains a topic for future work.

5.4 Conclusion

The results presented here provide us with mixed evidence regarding the hypothesis we stipulated in Section 5.2. Some of the presented results (statMAP and Figure 1a) confirm the premise that, using **met6**, a larger number of judged non-relevant documents improve retrieval effectiveness most. On the other hand, the overall results obtained on the 31 TREC Terabyte topics suggest that the collection is a viable and sufficient alternative. We would like to further explore the problem in two directions. First, we intend to investigate the impact of the available judged (non-)relevant documents and their properties with respect to the estimates on the collection. Second, given the relevance assessments, we will try to find better ways of estimating the true distribution of the (non-)relevant information within our framework. We believe that, instead of using maximum likelihood estimates, more sophisticated estimation methods may be explored and applied.

6 Conclusions

We described our approaches, submissions, and initial results of this year's TREC participation. For blog track we found that estimating optimal weights for external expansion for blog post retrieval is not trivial, and that currently a simple linear combination works best. In blog distillation we observe a positive effect for more advanced combinations of external corpora, and more analysis on the differences between the two tasks are therefore necessary. As to the relevance feedback track, we found that a larger number of judged non-relevant documents can improve retrieval effectiveness in terms of statMAP. Moreover, we can effectively replace the estimates on the judged non-relevant documents for the TREC Terabyte topics with estimations on the document collection. Conclusions on our enterprise track participation are postponed due to the lack of evaluation results at the time of writing.

7 Acknowledgments

This research was supported by the E.U. IST program of the 6th FP for RTD under project MultiMATCH contract IST-033104, by the DuOMAn project carried out within the STEVIN programme which is funded by the Dutch and Flemish Governments (`http://www.stevin-tst.org`) under project number STE-09-12, and by the Netherlands Organisation for Scientific Research (NWO) under project numbers 220-80-001, 017.001.190, 640.001.501, 640.002.-501, 612.066.512, 612.061.814, 612.061.815, and by the Virtual Laboratory for e-Science project (`http://www.vl-e.nl`), which is supported by a BSIK grant from the Dutch Ministry of Education, Culture and Science and is part of the ICT innovation program of the Ministry of Economic Affairs.

8 References

Balog, K. (2008). *People Search in the Enterprise*. PhD thesis, University of Amsterdam.

Balog, K., Azzopardi, L., and de Rijke, M. (2006). Formal models for expert finding in enterprise corpora. In *SIGIR'06*.

Balog, K. and de Rijke, M. (2008). Non-local evidence for expert finding. In *CIKM '08*.

Balog, K., de Rijke, M., and Weerkamp, W. (2008a). Bloggers as experts. In *SIGIR '08*.

Balog, K., Weerkamp, W., and de Rijke, M. (2008b). A few examples go a long way: constructing query models from elaborate query formulations. In *SIGIR '08*.

Dunlop, M. D. (1997). The effect of accessing nonmatching documents on relevance feedback. *ACM Trans. Inf. Syst.*, 15(2):137–153.

Ide, E. (1971). New experiments in relevance feedback. In Salton, G., editor, *The SMART Retrieval System – Experiments in Automatic Document Processing*, pages 337–354. Prentice-Hall.

Kraaij, W. (2004). *Variations on Language Modeling for Information Retrieval*. PhD thesis, University of Twente.

Kurland, O., Lee, L., and Domshlak, C. (2005). Better than the real thing?: iterative pseudo-query processing using cluster-based language models. In *SIGIR '05*.

Lafferty, J. and Zhai, C. (2001). Document language models, query models, and risk minimization for information retrieval. In *SIGIR '01*.

Lavrenko, V. and Croft, B. W. (2001). Relevance based language models. In *SIGIR '01*.

Rocchio, J. (1971). Relevance feedback in information retrieval. In Salton, G., editor, *The SMART Retrieval System: Experiments in Automatic Document Processing*, pages 313–323. Prentice Hall.

Wang, X., Fang, H., and Zhai, C. (2008). A study of methods for negative relevance feedback. In *SIGIR '08*.

Weerkamp, W., Balog, K., and de Rijke, M. (2008). Finding key bloggers, one post at a time. In *ECAI 2008*.

Weerkamp, W. and de Rijke, M. (2008). Credibility improves topical blog post retrieval. In *HLT/NAACL '08*.

Wong, W. S., Luk, R. W. P., Leong, H. V., Ho, K. S., and Lee, D. L. (2008). Re-examining the effects of adding relevance information in a relevance feedback environment. *Information Processing & Management*, In Press, Corrected Proof.

Zhai, C. and Lafferty, J. (2001). Model-based feedback in the language modeling approach to information retrieval. In *CIKM '01*.