The University of Amsterdam at TREC 2009

Blog, Web, Entity, and Relevance Feedback

Krisztian Balog Marc Bron Jiyin He Katja Hofmann
Edgar Meij Maarten de Rijke Manos Tsagkias Wouter Weerkamp

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

Abstract: We describe the participation of the University of Amsterdam's ILPS group in the web, blog, web, entity, and relevance feedback track at TREC 2009. Our main preliminary conclusions are as follows. For the Blog track we find that for top stories identification a blogs to news approach outperforms a simple news to blogs approach. This is interesting, as this approach starts with no input except for a date, whereas the news to blogs approach also has news headlines as input. For the web track, we find that spam is an important issue in the ad hoc task and that Wikipedia-based heuristic optimization approaches help to boost the retrieval performance, which is assumed to potentially reduce the spam in top ranked documents. As for the diversity task, we explored different methods. Initial results show that clustering and a topic model-based approach have similar performance, which are relatively better than a query log based approach. Our performance in the Entity track was downright disappointing; the use of co-occurrence models led to poor results; an initial analysis shows that while our approach is able to find correct entity names, we fail to find homepages for these entities. For the relevance feedback track we find that a topical diversity approach provides good feedback documents. Further, we find that our relevance feedback algorithm seems to help most when there are sufficient relevant documents available.

1 Introduction

This year the Information and Language Processing Systems (ILPS) group of the University of Amsterdam participated in the Blog, Web, Entity and Relevance Feedback tracks. In this paper, we describe our participation for each of these four tracks, in four largely independent sections: Section 4 on our Web track participation, Section 3 on our Blog track participation, Section 5 on our participation in the Entity track, and Section 6 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. We conclude in Section 7.

2 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:

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

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).

2.1 Modeling

Unless indicated otherwise, we smooth each document model using a Dirichlet prior:

$$P(t|\theta_D) = \frac{n(t,D) + \mu P(t)}{\sum_t n(t,D) + \mu}, \quad (2)$$

where $n(t,D)$ indicates the count of term t in D and $P(t)$ indicates the probability of observing t in a large background model such as the collection:

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

μ is a hyperparameter that controls the influence of the background corpus which we set to the average document length.

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; 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, $P(t|\hat{\theta}_Q)$ indicates the MLE of the expanded part, and the parameter λ_Q controls the amount of interpolation.

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 $^{\triangle}$ (and $^{\triangledown}$).

2.3 Clueweb

Except for the Blog track, all the tracks we participated in this year make use of the Clueweb document collection. We do not use any form of stemming and remove a conservative list of 588 stopwords. We index the headings, titles, and contents as searchable fields and do not remove any HTML tags.

3 Blog track

This year's Blog track consisted of two tasks: *top stories identification* and *faceted blog distillation*. The latter task is very similar to the "regular" blog distillation task that ran during the previous two TREC years (2007 and 2008). The goal there was to return blogs that show a recurring interest in a topic; the task for 2009 is very similar, with the addition of a *facet* per topic. Not only should a blog be topically relevant ("show recurring interest"), but it should also be a "good" blog in that it complies with the facet. An example could be a user looking for blogs on U.S. politics that follow an "in-depth" facet, i.e., blogs that mainly have in-depth posts or mainly shallow posts. Participating systems are supposed to return two rankings: one of blogs that are relevant to the topic and to one value of the facet (e.g., "in-depth") and a second ranking of blogs that are relevant to the topic and to the other value of the facet (i.e., "shallow" in this case).

As the faceted blog distillation task is very similar to the previously run blog distillation task in 2007 and 2008, we felt that our focus should be on the new top stories task, and we therefore dedicated most of our time and effort for preparing its submissions. We tackle the task of blog distillation using off-the-shelf models. This is reflected by the larger part of this section being dedicated to top stories.

The second task, *top stories identification*, is new; the goal is to identify top stories for a given day using information from the blogosphere, and provide a listing of blog posts that support the selection of a top story. The underlying scenario is one of a news provider (in possession of news headlines) trying to rank these headlines based on what people write about news stories in their blogs. For the identification part, it calls for an approach described in the following steps:

1. construct a "query" from headline;
2. limit results to the given date;
3. count the number of relevant posts;
4. rank headlines based on these counts.

The steps above reveal two limitations: (i) headlines are needed in advance, and (ii) topics from the blogosphere can only emerge when they are about news events reported by mainstream media. In an effort to alleviate these limitations, we take on the task from a different angle:

1. observe posts from the given date;
2. see what differentiates these posts from previous posts;
3. display the emerging topics;
4. rank headlines by their similarity to the emerging topics.

Although the algorithm can stop one step short, the last step is designated to provide compatibility with the task at hand. In our participation we investigate the potential of both approaches and report on initial evaluation of the results. For the second step of the top stories identification task, namely, to provide evidence for the importance of a headline, we chose to select the top blog posts ranked by the number of their respective comments.

In the remainder of this section we first describe the data and preprocessing for both tasks (Section 3.1), then, we introduce our top stories identification approaches (Section 3.2) and report on the performance of the submitted runs. In Section 3.3 we briefly discuss our approach to the blog distillation task, introduce the "facet indicators" we derived and report on the performance of our runs. Finally, we report on some initial conclusions for this year's Blog track participation in Section 3.4.

3.1 Data and Preprocessing

The dataset provided by TREC is the new Blogs08 collection; the collection consists of a crawl of feeds, permalinks, and homepages of 1,303,520 blogs during early 2008–early 2009. This crawl results in a total of 28,488,766 blogs posts (or permalinks). In our experiments we only used feed data, that is, the textual content of blog posts distributed by feeds (e.g. RSS) and ignored the permalinks. Two main reasons underly this decision: (i) the tasks (and especially the top stories task) are precision-oriented and benefit from a very

clean collection; and (ii) using feed data requires almost no preprocessing of the data (e.g. no html-removal, etc.). Extracting posts from the feed data gave us a coverage of 97.7% (27,833,965 posts extracted). As a second preprocessing step we perform language detection and remove all non-English blog posts from our corpus. We construct two indexes, one based on the full content of blog posts and one on only the blog post titles. Both indexes consist of 16,869,555 blogs posts. Additional to the indexing, we extract features that can prove useful in both tasks. Extracted features are: number of comments, post length, number of spelling errors, number of shouted words, number of emoticons, and ratio of first person pronouns.

Part of the top stories task is a collection of 102,812 news headlines from the New York Times. We created a separate index of this collection, resulting in an average news headline length of 11 words. Finally, we have the topics for the two tasks: 55 dates designate the topics for the top stories task, and 50 query—facet ("in-depth", "opinionated", or "personal") pairs constitute the faceted blog distillation topics.

3.2 Top Stories Identification

As explained in the introduction, we contrast two main approaches in identifying top stories: (i) starting from the news headlines, or (ii) starting from the blog posts. In our participation we explore the potential of both approaches and compare their results.

News to Blogs Given the scenario where news headlines are known beforehand, they can be used as starting points for identifying top stories on a given date. As explained before, this scenario is limited, but definitely worth investigating. The approach we take, is simple: we want to estimate the probability of a news headline given a date, and rank news headlines based on this probability. We use an expert finding model from Balog et al. (2006) (more specific Model 2) and modify it to fit the data at hand. Although the model allows us to explicitly define a post's importance for a given date, we assume all posts to be equi-important (i.e., the probability of the post given the date is uniform).

We run the approach on both a post index (run **IlpsTSHlP**) and a title-only index (run **IlpsTSHlT**). The reason for using the title-only index is that we expect bloggers to use important (news) terms in their post titles, so that matching the headline to the title would result in acceptable rankings as well.

Blogs to News Following the second scenario where the news headlines are unknown, we need to extract information from the blog posts without any prior knowledge of what is in the news. To this end, we take the top 5,000 blog posts from a given date, ordered by their respective number of comments. We then combine these posts and identify distinguishing terms between them and a background corpus. The background corpus consists of the remainder of the blog posts. These steps (covering steps 1 and 2 from the introduction) result in a set of weighted terms, where the weight indicates a term's "distinctiveness" for the given date. Based on co-occurrence statistics, the terms are clustered, leaving us with the topics that emerge from the blog posts. So far, this approach is very general and has nothing to do with news headlines. For an example of the generated output, see Table 1. To use this approach for the task at hand, we need a way of matching the extracted information to the news headlines. We index the news headlines and use the extracted term clusters as queries to this headline index. Headlines are ranked based on the distinctiveness of the terms, and if more than one headline matches a query, we select a maximum of 10 headlines for this "topic".

Terms	News event
ledger heath actor	Actor Heath Ledger dies
roe abortion	Roe v. Wade case on abortion; March for Life 2008
romney mitt huckabee thompson gop ...	Republican primaries 2008
luther martin king african dr	Martin Luther King Day 2008

Table 1: Example of top emerging terms (left) and related news events (right) for January 22, 2008.

As with the previous approach, we run this on both a post index (**IlpsTSExP**) and a title-only index (**IlpsTSExT**). Here, we expect the title-only representation to contain less noise (less indistinctive terms) and therefore be able to better get the important terms on top.

Results The results of our submitted runs are displayed in Table 2. The top two lines represent the two approaches on the post index and the lower two lines on the title-only index. The first observation is that the blogs to news approach significantly outperforms the news to blogs approach on all metrics and for both indexes. Looking at each approach individually, we see that for the news to blogs approach the difference between the two indexes is not significant. For the blogs to news approach the performance of the post index is significantly better than the title-only index for MAP and MRR.

3.3 Faceted Blog Distillation

Since our focus of this year's participation in the Blog track is on the new top stories task, we limit ourselves to a basic approach to the faceted blog distillation task. As described in

Approach	MAP	P5	MRR	RunID
b-to-n (p)	**0.1354**▲	**0.2655**▲	**0.4271**▲	IlpsTSExP
n-to-b (p)	0.0083	0.0291	0.1119	IlpsTSHlP
b-to-n (t)	0.0756▲	0.2036▲	0.2670▲	IlpsTSExT
n-to-b (t)	0.0085	0.0545	0.0958	IlpsTSHlT

Table 2: Results of our submitted runs of top stories identification task for the blogs to news (b-to-n) and news to blogs (n-to-b) approaches on a (p)ost index or (t)itle index.

previous work (Balog et al., 2008a; Weerkamp et al., 2008; Weerkamp and de Rijke, 2009) models for expert finding can effectively be applied to the task of blog distillation. Given the choice for these models, we are handed with several estimation choices: we need a query model, and an estimate of the importance of a post for its blog. Below we detail on the choices made here and on the indicators we have used for the various facets. We perform all our experiments on the title-only index because of efficiency reasons.

Query modeling Previous work showed that query expansion based on external collections can be very beneficial in retrieval tasks in the blogosphere (Arguello et al., 2008; Weerkamp et al., 2009; Weerkamp and de Rijke, 2009). We use this observation here to construct a query model from two external collections: Wikipedia, and a news collection (Xinhua and Reuters). The query model is constructed using the EEM4 model from (Weerkamp et al., 2009) and contains 10 terms per topic. An example of a query model generated by this model is displayed in Table 3.

Original	Expanded
farm	farm
subsidies	subsidies
	trade
	subsidy
	agricultural
	farms
	agriculture
	farmers
	food
	bill

Table 3: Example of the original topic (left) and the generated query model (right) for topic 1103.

Post importance One of the interesting features of our models is the possibility to estimate the importance of a post for its blog, that is, estimate the probability of a post given a blog. One could assume all posts being equally important, and thus assign a uniform probability, but it is more interesting to make meaningful distinctions. For this task we measured the KL-divergence between each post and its blog as an indication of the "centrality" or "consistency" of the post to the blog. A post that reflects the content of its blog better, gets assigned a higher importance.

Facet indicators and implementation To decide on the facet value of a blog, we use simple features extracted from the blogs; different combinations of features form different facet indicators. The facet indicator scores (the average score over all posts of a blog) for each blog is used in two ways: the retrieval score for a blog is multiplied by the facet score for one value of the facet (e.g., "personal") and the score is divided by the facet score for the other value (e.g. "company"). This is done for the top 300 results from the topical retrieval run. The feature to indicator translation is shown below:

opinionated *emoticons, comments, shouting*
Emoticons and shouting are used to express emotions in posts, while the number of comments is assumed to reflect post controversy.

personal *pronouns, shouting, spelling errors, emoticons*
Executive style of writing, e.g., a company's blog, contains less first person pronouns and is expected to have less spelling errors, emoticons, and shouting.

in-depth *post length*
In-depth thoughts are usually communicated with more words, and consecutively render post length a high promising feature.

Results We present the results of our submitted runs in Table 4. We alter the representation of the results and report scores for each of the three facets (the original results assessed the mixture of the three facets); we feel that this division is more informative and allows us to more directly identify "gaps" in our approach.

The results show quite low MAP scores, which is not a surprise, given that all runs use a title-only index. Overall, in terms of MAP the best performance is achieved by Model 2; this is not just true for the topical retrieval, but also for all facets. Combining Models 1 and 2 improves early precision in two cases, and achieves similar MRR scores as Model 2 alone. Finally, we see that the facet indicators are not very helpful: in most cases scores drop after using these indicators to rerank the results.

3.4 Conclusions

This year we focused on the new top stories identification task: use the blogosphere to rank news headlines. We follow two general approaches: news to blogs, and blogs to news. The former starts from the news headlines, uses them

Model	Indic.	MAP	P5	MRR	RunID
topical relevance					
Model 1	No	0.0182	0.1077	0.2313	IlpsBDm1T
Model 2	No	**0.0803**	0.1590	**0.3363**	IlpsBDm2T
Mixture	No	0.0402	**0.1795**	0.3351	IlpsBDmxT
opinionated facet					
Model 1	No	0.0094	0.0190	0.040	IlpsBDm1T
Model 2	No	**0.0466**	**0.0667**	0.1781	IlpsBDm2T
Mixture	No	0.0372	**0.0667**	**0.1889**	IlpsBDmxT
Mixture	Yes	0.0330	0.0476	0.1187	IlpsBDmxfT
personal facet					
Model 1	No	0.0346	0.0200	0.1251	IlpsBDm1T
Model 2	No	**0.1230**	**0.2200**	**0.3528**	IlpsBDm2T
Mixture	No	0.0784	0.1600	0.3126	IlpsBDmxT
Mixture	Yes	0.0837	0.1000	0.2780	IlpsBDmxfT
in-depth facet					
Model 1	No	0.0117	0.0316	0.0613	IlpsBDm1T
Model 2	No	**0.1032**	0.1579	**0.3680**	IlpsBDm2T
Mixture	No	0.0417	**0.1684**	0.3009	IlpsBDmxT
Mixture	Yes	0.400	0.1263	0.2649	IlpsBDmxfT

Table 4: Results of faceted blog distillation task for topical relevance, and each of the three facets.

as queries, and ranks these queries according to the headline likelihood. The latter is more general and tries to identify topics that emerge from the blogosphere. It is only in the final step that this approach tries to link these topics to news headline (by using the topics as a query against a headline index). The blogs to news approach outperforms the news to blogs approach on all reported metrics and does so significantly. Using a title-only representation of blog posts does not lead to improvement, neither on recall-based metrics (which makes sense) nor on precision-based metrics.

In the faceted blog distillation task we index a title-only representation of posts, and use (a mixture of) expert finding models. For estimating the facet values of blog posts, we use combinations of features like number of comments and spelling errors. We find that Model 2 outperforms Model 1 (and the mixture) and that the facet indicators hurt performance.

Future work focuses on applying more elaborate models to the top stories identification task and see how we can use additional (external) information to identify emerging topics, or use explicit links and references to news events for this task.

4 Web Track

This year's Web track consists of two tasks, namely the *ad hoc task* and the *diversity task*. The ad hoc task is similar to traditional ad hoc retrieval in a web setting. The goal is to return a list of documents from a static document collection, ranked by decreasing relevance, where document relevance is considered independent from the rest of the documents within the list. The second task, diversity, is new; the goal is to return a ranked list of documents which *together provide complete coverage for a query, while avoiding excessive redundancy in the result list*. Here, in contrast to the ad hoc task, document relevance is dependent on the presence of other documents in the same ranked list.

In the remainder of this section, we first describe the data and the pre-processing we use for both tasks in Section 4.1, followed by detailed description of our submissions to the ad hoc task in Section 4.2 and diversity task in Section 4.3. Section 4.4 concludes the description of our participation in this year's Web track.

4.1 Data and preprocessing

For both tasks in the Web track, we use the category A set of the Clueweb collection (the full collection). We use the parameter settings for indexing as described in Section 2.3. Our approaches retrieve against the text content of the web pages and leave out information provided by anchor texts or hyperlinks among web pages.

4.2 Ad hoc Task

The goal of the ad hoc task can be considered as one of the most basic ones in IR: to rank documents according to their relevance to a given query. Despite its simplicity, the nature and the size of the new Clueweb collection render the task challenging and interesting again. We did not apply spam filtering on the collection, although insights from preliminary data exploration suggest that it holds the potential for being the most improving feature in any ad hoc retrieval system on this collection. For now, we try two basic approaches, and use two optimization techniques. Below we describe the two approaches and the optimizations.

Markov Random Fields Following the ideas from Metzler and Croft (2005), we use Markov Random Fields (MRF) to rewrite our initial query. The goal of applying this technique is to be better able to grab phrases present in the query. A three term query like "obama white house" would result in all possible phrases (e.g., "obama white", "white house", "obama house", and "obama white house") and the single terms. Previous TREC years showed that this technique is very effective without losing efficiency.

External expansion Given that we are dealing with a web collection that can be quite noisy, we use a technique that proved useful in retrieval in the blogosphere: external query expansion (Arguello et al. (2008); Weerkamp et al. (2009); Weerkamp and de Rijke (2009)). The goal here is to use an

"external" collection that is less noisy than the target collection to re-compute our query model. The Clueweb collection offers a natural "external" collection: Wikipedia. We can be quite certain that this part of the collection is free of spam and relatively clean, and it would therefore be usable in modeling our query. We run our queries on the Wikipedia collection, select the top 10 terms (using relevance models from Lavrenko and Croft (2001)) and mix these with the original query terms.

Optimizing our approaches We use two ways of optimizing our runs: (i) Wikipedia filtering, and (ii) Wikipedia promotion. The first technique is used to filter out Wikipedia pages that do not contain real content. These pages are for example the link-to, category, and disambiguation pages that are mainly included for navigational purposes. We feel that these pages can be removed without danger of missing relevant documents, thereby possibly pushing relevant documents higher up the ranking. The second technique, Wikipedia promotion, is based on the observation that Wikipedia pages are pages we can certainly trust, whereas other web documents could very well be spam. We translate this observation into the promotion of all Wikipedia pages in the results to the top of the ranking (maintaining their relative order).

Our three final runs for the ad hoc task use: (i) Markov Random Fields and Wikipedia filtering, (ii) Markov Random Fields and Wikipedia filtering and promotion, and (iii) External expansion and Wikipedia filtering. We report on the results of the runs in the next paragraph.

Results The results of our runs are displayed in Table 5. The obvious observation is that the run using Wikipedia promotion outperforms the other two runs significantly. The difference with its baseline, MRF with just filtering, is huge, especially on the precision metrics. Comparing the two approaches, external expansion and MRF, in their "baseline" setting, we see a marginal advantage for external expansion, but differences are not significant.

Approach	MAP	P10	MRR	runID
MRF + filter	0.0626	0.0940	0.1255	uvamrf
MRF + filter + prom.	**0.1092**	**0.4100**	**0.5272**	uvamrftop
EE + filter	0.0682	0.1100	0.1627	uvaee

Table 5: Results of our submitted runs for the ad hoc task.

4.3 Diversity Task

For the diversity task, we experimented with 3 types of approach: *Single Pass Clustering (SPC)*, a *topic model-based approach*, and *AOL query suggestion*. The first two approaches share common features: they re-rank an initially retrieved list of documents for generating the final result list, and try to model the topical facets contained in the initial retrieved ranking list without using external resources. The difference between the two approaches mainly lies in the methods used for topic detecting and for re-ranking. For topic detection, the first approach, SPC, clusters documents into a number of topics and each document is assigned to one topic, while the topic model-based approach uses LDA for topic extraction and each document is represented as a mixture of the set of topics. For re-ranking, the SPC approach selects documents from different clusters so that selected documents are supposedly about different topics, while the topic model-based approach tries to maximize the probability that most if not all topics being present in the selected document list. The third approach, *AOL query suggestion* uses an external resource, i.e., the AOL query logs, for modeling the topical facets of a query. It also generates the final result list in a different fashion which will be further described in the following subsection.

Single Pass Clustering The first method we employed is Single Pass Clustering (SPC) (Hill., 1968), which provides not only an efficient clustering algorithm, but also mimics a reasonable heuristic that a user might employ. That is, start at the top and work down the initial retrieved list of documents, and assign each to a cluster. The process for assignment is performed as follows: The first document is taken and assigned to the first cluster. Then each subsequent document is compared against each cluster with a similarity measure (in our case a standard cosine measure using a TF.IDF weighting scheme). A document is assigned to the most likely cluster, as long as the similarity score is higher than a certain threshold (set to 0.2 for our run); otherwise, the document is assigned to a new cluster.

Once this (single pass) clustering has been performed on the initial result list, we re-rank documents as follows. First, we output a single document from each cluster, specifically, the ones that were ranked the highest initially. Second, we iterate over the initial list of documents, and output each that has not been returned in the first phase.

Topic Model Approach This approach is inspired by previous work on diversifying a ranked list with Maximal Marginal Relevance (MMR) by Carbonell and Goldstein (1998) and based on a topic modeling approach, i.e., LDA (Blei et al., 2003). It treats the reranking problem as a procedure of selecting a sequence of documents, where a document is selected depending on both its relevance with respect to the query and the documents that have already been selected before it, so as to have a set of documents that (i) are most relevant to the query and (ii) represent most if not all topical aspects.

We proceed as follows. First, we use LDA to extract 10 topics from the top 2,500 documents in the initial retrieved set of results, where the initial results are generated from the ad hoc run *uvamrf* as described above, and each document

Topic	AOL query	Frequency
dinosaurs	remote control dinosaurs	30
dinosaurs	jim henson dinosaurs	25
dinosaurs	allosaurus dinosaurs	24
dinosaurs	flying dinosaurs	21
dinosaurs	walking with dinosaurs	16

Table 6: Example of using the AOL query logs for diversification.

can be represented as a mixture of the 10 topics. On top of that, we start the re-ranking procedure by selecting the top relevant document in the initial list as the first document in the new ranked list. Then, we select a next document that can maximize the expected joint probability of presence of all topics in the selected result set. Since the sum of topic proportions within a document equals to 1, the maximum joint probability (i.e., product of the probabilities of presence of each topic) occurs when the topics have equal proportion in the selected set. On the other hand, we use the retrieval score from the initial run as a prior probability that a document is selected as the next one, so as to take into account the relevance relation between the document and the original query.

AOL — Diversifying using query logs This approach employs queries from a query log to discern and obtain diverse query formulations. The intuition is that terms that are frequently queried in conjunction with the current query terms provide a diverse set of *aspects*. We proceed as follows. First, we normalize all the queries in the AOL query logs and remove web addresses and non-alphabet characters. We then look up for each topic whether it appeared as a phrase in the query logs. If so, we take the top 25 queries with a minimum occurrence of 5. An example is given in Table 6. For each of these "expanded" queries we generate a weighted proximity query (Metzler and Croft, 2005; Mishne and de Rijke, 2005) and, on the basis of this, a new ranking. Each of these ranked lists now represents a ranking of documents based on one aspect of the initial topic. In order to arrive at a final ranking, the lists are merged. We do so by first sorting them by aspect occurrence frequency (as found in the query log) and, then, adding the highest ranked document that has not been selected yet to the final ranking in a round-robin fashion.

Results Table 7 shows the results of our submitted runs for the diversity task. The results can only be considered indicative considering the heuristic selection of parameter values. Nevertheless, we observe that the clustering method and the topic model based method yield similar performance. Intuitively, this is likely due to the common features shared during the topic detection process: given that LDA can also be seen as a method for clustering, the resulting clustering/topic

Approach	α-ndcg@5	α-ndcg@10	α-ndcg@20
AOL	0.055	0.074	0.098
SPC	0.068	0.093	**0.127**
TM	**0.090**	**0.097**	0.125

Approach	IA-P@5	IA-P@10	IA-P@20
AOL	0.023	0.030	0.037
SPC	0.036	**0.043**	**0.051**
TM	**0.047**	0.041	0.043

Table 7: Result of diversity task. The names of approaches correspond to AOL query suggestion (AOL), single pass clustering (SPC) and topic model based approach(TM).

structure may be similar. However, in order gain insight into the similarities and differences in behavior of the two approaches, further comparison and analysis are needed.

4.4 Conclusion

In this year's web track, we submitted runs to both ad hoc task and diversity task. For the ad hoc task, we explored a basic retrieval approaches, namely Markov Random Fields for modeling query term proximity and external query expansion. On top of that, we applied two types of heuristic optimization approaches, i.e., Wikipedia filtering and Wikipedia promotion. Combining the basic approaches with the optimization methods, we submitted three runs: i) Markov Random Fields with Wikipedia filtering, ii) Markov Random Fields with Wikipedia filtering and promotion, iii) External Expansion with Wikipedia filtering. Although we did not explicitly apply any spam filtering techniques, the preliminary results suggest that spam is an important issue in web retrieval. For the diversity task, we explored three types of approach: (i) Single Pass Clustering iii) topic model-based approach, and (iii) diversifying using a query log. Although the results are not exactly comparable across methods, we were able to identify issues shared by all three methods. For example, the heuristic method for choosing parameters calls for systematic experiments that will allow us to gain further insights in to the algorithms' performance under different parameter settings. On the other hand, intuitively, the performance of the clustering and topic model-based methods depends heavily on the initial retrieval run used for re-ranking, which is an interesting issue for further analysis.

5 Entity Track

5.1 Approach

In this first year of the entity track we formulated the entity ranking problem as follows: to rank candidate entities (e) according to $P(e|E,T,R)$, where E is the input entity, T

is the target type, and R is the relation described in the narrative. This probability ($P(e|E,T,R)$) is decomposed into the following four components: entity priors, a context-independent entity-entity co-occurrence model, a context dependent entity-entity co-occurrence model and entity type detection. Below, we briefly introduce and discuss these components.

Entity priors This component expresses the a-priori probability of an entity being relevant, independent of the information need; it can be used to favor certain (e.g., popular) entities. In our approach we assume that all candidate entities are equally likely to be returned (i.e., the probability mass is distributed uniformly).

Context-independent entity-entity co-occurrence model We use this component to express the strength of associations between the input and candidate entities, without considering the nature of their relation. We use pointwise mutual information (PMI, computed on the basis of the number of documents in which entities co-occur) as an estimate of the degree of association between entities.

Context-dependent entity-entity co-occurrence model In this component we model the relations that hold between the input entity and candidate entities, represented as statistical language models. Such co-occurrance language models are constructed from contexts (windows of text) in which the entities co-occur, and are used then to estimate the probability that a pair of entities are in a specific relation (described in the narrative).

Entity type detection The final component is used to filter entities by type. The type of an entity is determined in one of two ways. In one approach, a named entity tagger is used to tag an entity with one of its type labels; the other approach checks whether an entity matches exactly a Wikipedia page title, and in that case we use the Wikipedia category structure.

We model a target type (person, organization, product) by a top category and its subcategories, up to 4 levels deep, e.g., the person category in Wikipedia and its subcategories. An entity's type is modeled by matching a term to a Wikipedia page and then using the categories that the page belongs to. If the Wikipedia categories of the entity overlap with one of the target type categories we consider it of that type.

5.2 Implementation

Since we are not able to run named entity tagging and normalization on the whole Clueweb collection, given our current infrastructure, we need to apply some reasonable heuristics in order to realize an efficient implementation of our approach. We do this by limiting the set of entities for which $P(e|E,T,R)$ is calculated. Therefore, we only look at documents in which the input entities occur (we use each input entity as a query to the collection to get a ranked list of documents; we consider the top N documents for each, where N is set to 1000 in our experiments).

As a next step we use a named entity tagger (Stanford NE tagger by Finkel et al. (2005)) to recognize entities in these documents. The tagger recognizes 4 entity types: person, organization, location and miscellaneous. No special category for products was used. To identify products and to improve recall for the other classes we also tag terms that match Wikipedia titles with a label derived from the category structure.

Once documents containing the input entity have been tagged with named entities, we extract contexts in which the input entity co-occurs with candidate entities. The context we use is a window of text to the left and to right of the entities; the size of the window is set to 40, measured in term positions. For each entity pair the contexts are added to a document that forms an entity co-occurrence model. In order to reduce the number of distinct entities we perform variant detection, i.e., we recognize that "B. Obama" and "Barack Obama" are the same entity (see paragraph below). The variant models are merged into a single co-occurrence model.

Once these context dependent co-occurrence language models have been created, the narrative is used as a query against an index of the models to obtain a ranked list of entities. In the last step we find homepages for each entity; this, too, is discussed below.

Name variant detection To prevent our rankings from being dominated by variants of only a hand full of entities, we perform variant detection. We implemented a set of heuristic rules that depend on the topic type. For example, typical person name variants are "B. Obama," "Obama" and "Barrack Obama," but for a company variants as "Apple," "Apple Inc." and "Apple Computer" are more common.

To find variants that are more difficult to derive by simple rules, for example "Schumacher" and "Schumi," we use Wikipedia redirects. Redirects are used to link title variants to a basic page title. So whenever an entity has a Wikipedia page we find its variants.

Primary homepages We use the query "official homepage of ⟨ENTITY⟩" to obtain a list of webpages related to an entity name. To find homepages we assume that the name occurs somewhere as part of the url of those pages. In the case of a person or company the homepage usually has the name as the main part of the url, for example `www.barackobama.com` and `www.apple.com`. For products the name is usually at the end of the URL, for example `http://www.apple.com/iphone/iphone-3gs`. We sum the edit distance between the entity name and each part of the URL, i.e., delimited by backslash and average over the

number of parts. The 3 documents that belong to the highest scoring urls are returned as homepages.

Supporting documents When we extract context from a document we also store its document id and the entity the context belongs too. We return up to 3 supporting documents from this set for each entity.

5.3 Runs

In our runs we focus on methods to find entity names for a topic. We only implemented a simple heuristic to find homepages and left out Wikipedia pages, as we are less interested in this second task.

ilpsEntBL Baseline run with type filter
In this run we use only two components: the context independent co-occurrence model and type detection. Candidate entities are ranked according to their co-occurrence score and filtered based on type. We ignore entities of which the type is unknown. This run serves as a baseline and shows which entities are most often seen with the input entity.

ilpsEntem Entity model run with type filter
In our main run we combine all four components. We fire the narrative as a query against the context dependent co-occurrence models. The retrieval scores are combined with scores of the context independent co-occurrence models. Finally, the ranked list of entities is filtered by type.

ilpsEntcf Entity model run with category filter
In this run we constructed a topic category model for each topic, based on the narrative and used it as an alternative to the type filter component. The narrative usually gives a more specific instantiation of the type than the given topic type, e.g., woman vs. person, company vs. organization.

The category model is estimated by using the narrative as a query against an index of Wikipedia categories. The top 50 categories are taken to represent a topic. A category model for an entity is constructed by taking the Wikipedia categories it directly belongs to. The final ranking is obtained by scoring candidate entities as in our main run and filtering based on their category models instead of there type.

ilpsEntcr Entity model run with category reranking and type filtering
In our last run we re-ranked our main run (ilpsEntem) by weighting the rank of an entity with the overlap between the entity category model and the topic category model. The probability that an entity category model belongs to a topic category model is estimated by the number of categories these sets have in common.[1]

5.4 Results

runID	nDCG_R	P10	pri_ret	rel_ret
ilpsEntBL	0.0161	0.0	1	30
ilpsEntem	0.0105	0.0	0	17
ilpsEntcf	0.0128	0.0	0	25
ilpsEntcr	0.0161	0.0	1	30

Table 8: Total score for each of our Entity track runs.

The focus on entity names instead of homepages did not result in favorable scores, see Table 8. Given the disappointing results there is little sense in comparing our runs; instead, we analyze the components and identify possible causes.

First of all, it is worth noting that while the task is (related) entity finding, it actually consists of two phases: (1) finding related entities and (2) locating the (primary) homepages for each of these entities; the current evaluation methodology employed by the track measures only the overall performance. An initial analysis shows that while our approach was able to locate the correct entities, we failed to locate the (primary) homepages for these entities. It suggests that we need more sophisticated methods than edit distance between entity names and URLs, as currently phase (2) forms the bottleneck of our system.

We also suspect that the context dependent co-occurrence models are sensitive to the size of the window used for context extraction. A wide context introduces terms unrelated to the relation between the input entity and candidate entity into the model and negatively influences the ranking. One of the next steps is to perform a sensitivity analysis w.r.t. this parameter.

Finally, there is the dependence of our entire pipeline on the quality of the named entity recognition method. We focused on entities that have a page in Wikipedia, complemented with a basic tagger. Our initial analysis shows that many entities were not recognized and consequently not returned by any of our runs.

6 Relevance Feedback Track

This year, the goal of the Relevance Feedback track is to evaluate how well a system can find good documents to serve as input for the relevance feedback algorithm, as well as the improvement gained by the feedback algorithm itself.

There are two phases to the track. In the first phase, systems were to return two ranked lists (with a maximum of 5

[1]The ilpsEntcr run had a bug in it; results for the corrected run are: nDCG_R: 0.0131, P10: 0.0, pri_ret: 1, rel_ret: 30.

documents) for each topic. In the second phase, all participating systems were given their own ranked list and a number of ranked lists from other groups from phase 1 and relevance assessments to perform relevance feedback. For our submitted runs in phase 1 we used the Category B subset of Clueweb, while for the runs in phase 2 we used Category A.

6.1 Phase 1

For the first phase we generated two runs based on different approaches. The first run was inspired by our approach to the diversity task of the Web track (cf. Section 4.3), whereas the second run was a standard combination of pseudo-relevance feedback and query modeling.

Diversity This run (ilps.1) tries to select documents that reflect different topical facets of a given query for relevance feedback. Intuitively, a query may have different topical facets, where some are relevant while others are irrelevant. From a clustering point of view, a set of documents that are representative for different topical facets would provide more information than documents that all focus on a single topical facet, since we can easily use a "prototypical" model to represent the single-topic set of documents.

For detecting different topical facets of the documents associated with each topic, we run hierarchical clustering on the top 50 documents from an initial retrieval run. For this kind of clustering one needs to pre-define a cut-off threshold which determines the number of clusters. However, in our scenario, we are not interested in getting a perfect clustering of the documents. Instead, we only want to detect the *significant* topical facets contained in the documents associated with a particular query. We measure the significance of a cluster with two measures: *stability* and *cluster quality*. A cluster is stable when it repeatedly occurs given different cut-off threshold and is of high quality when it results in a high Silhouette value (a measure for the quality of a cluster (Rousseeuw, 1987)). Additionally, in order to prevent outliers dominating the top ranked clusters, we also take the cluster size into account. We rank the clusters by combining these scores, i.e., the stability, silhouette values, and cluster size, in a heuristic way. Once we have obtained a ranked list of clusters, we select the top scoring documents from each cluster as our ranking.

Pseudo Relevance Feedback For this run (ilps.2) we apply a standard combination of pseudo relevance feedback and structured query modeling. We first transform each query into a full-dependency query model (Metzler and Croft, 2005). We then perform a retrieval run and select the 10 top-ranked documents. From these documents we generate relevance models (RM-1 (Lavrenko and Croft, 2001)) and keep the 50 terms with the highest probability. We use the expanded query to retrieve our final ranked set of documents.

Results Table 9 shows the aggregate score of our submitted runs for phase 1. We observe that ilps.1 is a better source of feedback documents than most other runs, whereas the opposite is true for ilps.2.

runID	score
ilps.1	0.8281
ilps.2	0.2885

Table 9: Results of our submitted relevance feedback runs in phase 1.

6.2 Phase 2

The main goal of phase 2 is to see how well each participants' relevance feedback algorithm performs, by running them on a set of 8 baseline runs (constructed in phase 1). Using each of the baseline runs and the relevance assessments, we need to identify new relevant documents. Participants were allowed to submit only one run and we suffice by reporting on our approach and its results. Comparing it to other, standard approaches remains as future work.

The leftmost columns of Table 10 show the baseline runs we were assigned, and the number of retrieved documents and the number of relevant documents in each run. As can be observed from this table, the information available from just the relevant documents is limited (the best run has 44% of its returned documents judged relevant). We believe that making our feedback approach dependent solely on these few documents is not a good idea and we feel we need to incorporate the non-relevant information as well to obtain the best relevance feedback results.

The general goal of a relevance feedback algorithm is to extract terms from relevant documents that distinguish them from other, non-relevant documents. One way of approaching this would be to use the non-relevant documents as a language model against which to compare the relevant documents (Meij et al., 2008). From this comparison one could, for example, extract terms that distinguish between the relevant and non-relevant documents. Even though this is a valid approach, we feel that in the current situation this approach might not work optimally: first, the total number of judged documents is very limited (maximum of 5 documents per topic), which makes it hard to put confidence in comparing the two sets. Second, for a significant portion of the topics we have neither relevant nor non-relevant documents, and for these cases this approach would not work at all.

Building on the observations above, we arrive at the following wishlist. First, a sensible approach to feedback should make use of each individual judged document as much as possible. Second, the approach should be able to handle cases in which no relevant or no non-relevant documents are known. Finally, as mentioned before, the approach

	Documents		IlpsRF	
runID	retrieved	relevant	MAP	P10
QUT.1	248	36	0.0688	0.1286
Sab.1	250	98	0.0581	0.0939
WatS.1	250	**110**	**0.0915**$^{\triangle}$	**0.2878**$^{\blacktriangle}$
fub.1	250	81	0.0792$^{\triangle}$	0.1694$^{\triangle}$
ilps.1	250	87	0.0705	0.2020$^{\blacktriangle}$
ilps.2	250	92	0.0680	0.1898$^{\triangle}$
twen.1	250	83	0.0776	0.1837$^{\blacktriangle}$
twen.2	250	71	0.0694	0.1816$^{\triangle}$
—	—	—	0.0639	0.0959

Table 10: Main results of our system. The second and third column indicate the number of retrieved documents and the number of relevant documents for each of the baseline runs assigned to us. The rightmost columns contain the resulting performance of applying our feedback algorithm. Significance is tested against the run without any relevance feedback information (last row).

should take non-relevance into account and not depend on relevant documents only. Based on these requirements we take a four-step relevance feedback approach:

1. Extract key terms from each individual document.
2. Use the extracted terms as queries.
3. Combine the result lists from step 2 in two rankings: a relevant and a non-relevant one.
4. Combine both rankings from step 3 into a final ranking.

Below we elaborate on these steps.

Extract key terms and run as queries We compare each judged document to a background collection and identify key terms that distinguish this document. As background collection we take the full collection and we select only terms that occur at least four times in the document (to avoid selecting infrequent terms and typos). The weights of the resulting terms are normalized, leaving us with a weighted representation (or "query") for each document. We use this query to retrieve a set of new documents. We now have, for each judged document, a ranked list of documents which are highly similar. An example of two queries, a relevant and non-relevant one, are displayed without their weights in Table 11. Additionally, we create a baseline ranking based on the original query terms.

Construct relevant and non-relevant rankings We then combine the ranked lists from the previous step into two separate rankings: one for the relevant documents and one for the non-relevant documents. We do so by normalizing the retrieval scores for each topic and ranking using min-max normalization (Lee, 1995) and use CombMNZ (Fox and Shaw, 1994) to combine the relevant rankings into one, and the non-relevant rankings into one. We are now left with two new rankings, one being a ranking of relevant documents and the other a ranking of non-relevant documents.

Relevant	greyhounds, rescuing, doberman purebred, adoption, shih, collie, rescues
Non-relevant	adoption, transracial, photolisting

Table 11: Examples of the key terms from a relevant and non-relevant document for topic RF09-38, "dogs for adoption".

Construct final ranking The final ranking is then constructed from the relevant and non-relevant rankings: we simply subtract the non-relevant score for each document from its relevant score. The idea behind this step is that a document that is returned high for many relevant documents, but is hardly ever returned for non-relevant documents, receives a high final score. Documents that are mixed, i.e., showing up in both rankings, would get ranked below these documents, and documents that are ranked high in the non-relevant ranking and are nowhere to be found in relevant rankings, drop all the way to the bottom.

The approach described above fulfills our requirements in that it (i) takes full advantage of each individual document, (ii) can handle cases where no relevant or no non-relevant information is available, and (iii) takes non-relevance into account.

Results Table 10 shows the result of applying our relevance feedback algorithm to our assigned input rankings from phase 1. From this table we observe that there seems to be a correlation between the number of relevant documents in the phase 1 ranking and the resulting, final performance. The last row of the table indicates the performance of our system without any relevance feedback information. We note that using relevance feedback information helps in all cases but one. Further, the improvement of applying our relevance feedback algorithm is significant for early precision in most cases. Finally, we observe that the absolute MAP values are quite low.

7 Conclusion

We have described the participation of the University of Amsterdam's ILPS group in the web, blog, web, entity, and relevance feedback track at TREC 2009. We arrived at the following preliminary conclusions. For the Blog track we find that for top stories identification a blogs to news approach

outperforms a simple news to blogs approach. This is interesting, as this approach starts with no input except for a date, whereas the news to blogs approach also has news headlines as input. In the Web track we found that spam is an important issue in the ad hoc task and that Wikipedia-based heuristic optimization approaches help to boost the retrieval performance, which is assumed to potentially reduce the spam in top ranked documents. As for the diversity task, we explored different methods. Initial results show that clustering and a topic model-based approach have similar performance, which are relatively better than a query log based approach. Our performance in the Entity track was downright disappointing; the use of co-occurrence models led to poor results; an initial analysis shows that while our approach is able to find correct entity names, we fail to find homepages for these entities. For the Relevance Feedback track we found that a topical diversity approach provides good feedback documents. Further, we found that our relevance feedback algorithm seems to help most when there are sufficient relevant documents available.

8 Acknowledgments

This research was supported 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 640.001.501, 640.002.501, 612.066.512, 612.061.814, 612.061.815, 640.004.802, 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.

9 References

Arguello, J., Elsas, J., Callan, J., and Carbonell, J. (2008). Document representation and query expansion models for blog recommendation. In *ICWSM 2008*.

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

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*.

Blei, D. M., Ng, A. Y., and Jordan, M. I. (2003). Latent dirichlet allocation. *J. Mach. Learn. Res.*, 3:993–1022.

Carbonell, J. and Goldstein, J. (1998). The use of mmr, diversity-based reranking for reordering documents and producing summaries. In *SIGIR '98: Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval*, pages 335–336, New York, NY, USA. ACM.

Finkel, J. R., Grenager, T., and Manning, C. (2005). Incorporating non-local information into information extraction systems by gibbs sampling. In *ACL '05: Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, pages 363–370, Morristown, NJ, USA. Association for Computational Linguistics.

Fox, E. and Shaw, J. (1994). Combination of multiple searches. In *The Second Text REtrieval Conference (TREC-2)*, pages 243–252. NIST.

Hill., D. R. (1968). A vector clustering technique. In Samuelson, editor, *Mechanised Information Storage, Retrieval and Dissemination*, North-Holland, Amsterdam.

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*.

Lee, J. (1995). Combining multiple evidence from dierent properties of weighting schemes. In *Proceedings of the 18th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 180–188.

Meij, E., Weerkamp, W., He, J., and de Rijke, M. (2008). Incorporating non-relevance information in the estimation of query models. In *Seventeenth Text REtrieval Conference (TREC 2008)*.

Metzler, D. and Croft, W. B. (2005). A markov random feld model for term dependencies. In *SIGIR '05: Proceedings of the 28th annual international ACM SIGIR conference on research and development in information retrieval*, pages 472–479, New York, NY, USA. ACM Press.

Mishne, G. and de Rijke, M. (2005). Boosting Web Retrieval through Query Operations. In Losada, D. and Fernández-Luna, J., editors, *Advances in Information Retrieval. Proceedings 27th European Conference on IR Research (ECIR 2005)*, pages 502–516.

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.

Rousseeuw, P. (1987). Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *J. Comput. Appl. Math.*, 20(1):53–65.

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

Weerkamp, W., Balog, K., and de Rijke, M. (2009). A generative blog post retrieval model that uses query expansion based on external collections. In *ACL-ICNLP 2009*.

Weerkamp, W. and de Rijke, M. (2009). External query expansion in the blogosphere. In *Seventeenth Text REtrieval Conference (TREC 2008)*. NIST.

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