LIA at TREC 2011 Web Track: Experiments on the Combination of Online Resources

Romain Deveaud
LIA - CERI
University of Avignon
Avignon, France
romain.deveaud@univ-avignon.fr

Eric SanJuan
LIA - CERI
University of Avignon
Avignon, France
eric.sanjuan@univ-avignon.fr

Patrice Bellot
LSIS
Aix-Marseille University
Marseille, France
patrice.bellot@lsis.org

Abstract

In this paper, we report the experiments we conducted for our participation to the TREC 2011 Web Track. The experiments we conducted this year aim at discovering how the combination of specific external resources in a language modeling fashion can help web search. We use Wikipedia and Google as external resources for different search contexts.

1 Introduction

When searching for a specific information, users query the retrieval system with a list of keywords, a question, a declarative sentence or maybe a long description of the search topic. However, this often does not fully describe the user information need, which may harm retrieval performance.

One way to better outline the topic of the search without the help of the user is to enrich the query with additional information. Such query expansion techniques have shown to significantly improve the effectiveness of retrieval systems in many TREC tracks before.

This year we experimented with the combination of two external and online resources for improving web search. Terms related to the information need are extracted from these resources and appended to the query, following a weighting scheme that reflects the relevance of each term to the initial query. We experimented expansions with Wikipedia, Google and both. When using only Wikipedia, we modeled the thematic links between the encyclopedic pages in order to generate a thematic graph. We used this graph to increase the thematic coverage and to expand the initial query with more terms linked to the topic.

The ClueWeb09 collection includes the English version of Wikipedia and is composed of approximately 504 million of English documents. Considering Google indexes roughly 45 billion web pages[1], we can assume that Google includes the ClueWeb09 collection itself. In the case of using Wikipedia as an external resource, we expand the query using a thematic graph extracted from an encyclopedic subset of the collection. In the case of using Google, we expand the query using terms extracted from a set of documents that include the collection. Our goal with these experiments is to compare both intra-collection and extra-collection approaches.

2 Retrieval system

This year we used a language modeling approach to retrieval. We follow the work done by Diaz and Metzler (Diaz and Metzler, 2006) who provided a framework allowing to interpolate relevance models computed using external collections with the maximum likelihood query estimate. This approach highlighted significant improvements over query likelihood alone when performing retrieval on news and web data with expansion terms extracted from different news and web collections.

2.1 Sequential Dependence Model

The sequential dependence model (SDM) is a special case of the Markov Random Field model for Information Retrieval introduced by Metzler and Croft (Metzler and Croft, 2005), and was used by several teams in previous Web Track editions (Bendersky et al., 2011; He et al., 2010; Smucker et al., 2010). The sequential dependence instantiation of MRF aims to model dependencies

[1] http://www.worldwidewebsize.com

between adjacent query terms.

The SDM provides two feature functions for two types of term dependence involving query bigrams.

The $f_O(q_i, q_{i+1}, D)$ feature function considers ordered matches of two adjacent query terms and is denoted by the O subscript. The second one is denoted by the U subscript and considers unordered matches within a window of 8 terms. Here, $c(\#1(q_i, q_{i+1}), D)$ is the number of occurrencies of the bigram (q_i, q_{i+1}) in the document D. On the other side, $c(\#uw8(q_i, q_{i+1}), D)$ is the number of occurrence of the two query terms q_i and q_{i+1} within an unordered window composed of 8 terms in the document D.

Finally, the query-document score using the above feature functions defined by the sequential dependence model is:

$$\begin{aligned} score_{SDM}(Q,D) &= \lambda_T \sum_{q \in Q} f_T(q,D) \\ &+ \lambda_O \sum_{i=1}^{|Q|-1} f_O(q_i, q_{i+1}, D) \\ &+ \lambda_U \sum_{i=1}^{|Q|-1} f_U(q_i, q_{i+1}, D) \end{aligned} \quad (1)$$

where λ_T, λ_O and λ_U are free parameters, and $f_T(q, D)$ is a maximum likelihood estimate of term q in a document D computed over the target collection with a Dirichlet smoothing. We will further refer to the SDM scoring function defined in Equation (1) as SDM.

2.2 Online Resources Combination

After defining the basic retrieval models we use, we can now detail how external resources are incorporated to the SDM ranking function defined in Section 2.1. We incorporate terms extracted from different resources as feature functions in the model. The external resources we use in this work are Wikipedia and Google Search. Considering $\mathcal{W}$ a sequence of words extracted from a set of Wikipedia pages and $\mathcal{G}$ a sequence of words extracted from a set of Google pages, we rank documents according to the following scoring function:

$$\begin{aligned} score(Q,D) &= score_{SDM}(Q,D) \\ &+ \lambda_{\mathcal{W}} \sum_{w \in \mathcal{W}} H_{\mathcal{W}}(w) \cdot f_T(w,D) \\ &+ \lambda_{\mathcal{G}} \sum_{g \in \mathcal{G}} H_{\mathcal{G}}(g) \cdot f_T(g,D) \end{aligned} \quad (2)$$

where $\lambda_{\mathcal{W}}$ and $\lambda_{\mathcal{G}}$ are fixed parameters ($\lambda_{\mathcal{W}} = \lambda_{\mathcal{G}} = 1$ in our experiments).

The weights $H_{\mathcal{W}}(\cdot)$ and $H_{\mathcal{G}}(\cdot)$ are the entropy measures of words computed over the different resource sets. Considering $\mathcal{W}$ a sequence of words extracted from a set of Wikipedia pages, the entropy measure $H_{\mathcal{W}}$ we use is defined as follows:

$$H_{\mathcal{W}} = -\sum_{w \in \mathcal{W}} p(w|\mathcal{W}) \cdot \log p(w|\mathcal{W})$$

where word appearance probabilities $p_{\mathcal{W}}(\cdot)$ are computed within the whole set of Wikipedia pages. We chose an entropy measure to weigh the selected terms in order to reflect their relative informativeness within the set of pages they belong to. This measure behaves the same when using Google as an external resource.

3 Term extraction process

As described in Section 2.2, we use terms extracted from external resources as features in the ranking function. We detail in this section the processes involved in selecting informative terms from these resources for a given query. First, we explain how relevant documents are selected for each of the resources. Then, we give details about how terms are extracted and from the previously selected relevant documents.

3.1 Page Selection

The purpose of this general query expansion is to associate documents issued from external resources to a single query. The underlying principle is that adding knowledge related to the search topic will help to better understand the user's information need. In this work we use two different resources for query expansion, namely Wikipedia and the Google search engine. No query reformulation are made for page selection.

3.1.1 Wikipedia search API

Wikipedia pages are retrieved using the online API tool provided by the free encyclopedia[2]. We

[2] http://www.mediawiki.org/wiki/API:Search

use the query terms to query the API, which returns results of Lucene search engine[3]. The results given by the API consist of URLs of Wikipedia articles. When we ran our experiments during July 2011, the online version of Wikipedia had a total 3,641,203 encyclopedic articles. Given a query, the specific tool we developed automatically:

1. queries the API,
2. collects the resulting URLs,
3. gets the articles from URLs and concatenates them,
4. strips all HTML fields and filters stopwords,
5. computes frequencies of the words.

Then, we have a list of Wikipedia words $\mathcal{W}$ related to the user's query, with their frequencies in the selected pages.

3.1.2 Google search engine

When using Google, we query the search engine using the strict `<query>` text and collect the URLs of the top retrieved results. Again, we get the pages linked by the URLs, strip HTML tags and filter stopwords in order to obtain a list of words $\mathcal{G}$ which have been extracted from web pages retrieved by Google.

3.2 Term Extraction

We detailed in Section 3.1 how relevant documents from external resources are selected. Now we need to extract highly informative terms from the pages in order to add them to the query.

Each set of documents $\mathcal{W}$ or $\mathcal{G}$ is considered as a bag of terms that contain no stopwords. An entropy measure is computed for each term within its own set of documents. This measure allows to reflect the informativeness of each term considering the context of the user's search. Then, an informative weight is associated to each term. The terms are sorted by decreasing informativeness and the top-ranked ones are extracted.

4 Wikipedia Thematic Graphs

In the previous methods we expanded the query with words selected from pages directly related to the query. For our last run, we wanted to select broader and more general words, that could stretch topic coverage, at the risk of being too general. Considering that we can retrieve up to 10,000 documents for each query, we expect that extending topic coverage in large part will lead to better results. The main idea is to generate a thematic graph between Wikipedia pages in order to generate a set of articles that (ideally) completely covers the topic.

4.1 Thematic anchor texts

For this purpose we use anchor texts and their associated hyperlinks in the first Wikipedia page associated to the query. We keep the term extraction process detailed in Section 3 for selecting a Wikipedia page highly relevant to the query. We extract informative words from this page using the exact same method as above. But we also extract all anchor texts in this page.

The words selected with the entropy measure are considering as a set $T_{\mathcal{W}}$, as well as each anchor text. We then compute and intersection between set $T_{\mathcal{W}}$ and each anchor text set. If the intersection is not null, we consider that the Wikipedia article that is linked with the anchor text is thematically relevant to the first retrieved Wikipedia article. We sum the previously computed entropies of the words in common between the anchors and the expansion words, which gives a relevance score to each anchor.

This method relies on the fact that anchor texts in Wikipedia are written by experienced users that are confident about the topic. Indeed, on the web hyperlinks are sometimes randomly constructed by robots, and some web pages can be linked together even if they do not share the same topic. Authors also have different writing styles that can affect the definition of anchor texts.

4.2 Building a complete weighted graph

We can iterate and construct a directed graph of Wikipedia articles linked together. Children node pages (or *sub-articles*) are weighted half that of their parents in order to minimize a potential *topic drift*. We avoid loops in the graph (i.e. a children node can not be linked to one of his elder) because it brings no additional information. It also could change weights between linked articles. Informative words are then extracted from the sub-articles and incorporated to the query as another resource, as described in Section 2.2.

The complete process of generating the weighted graph, from querying Wikipedia through

[3] `http://lucene.apache.org/`

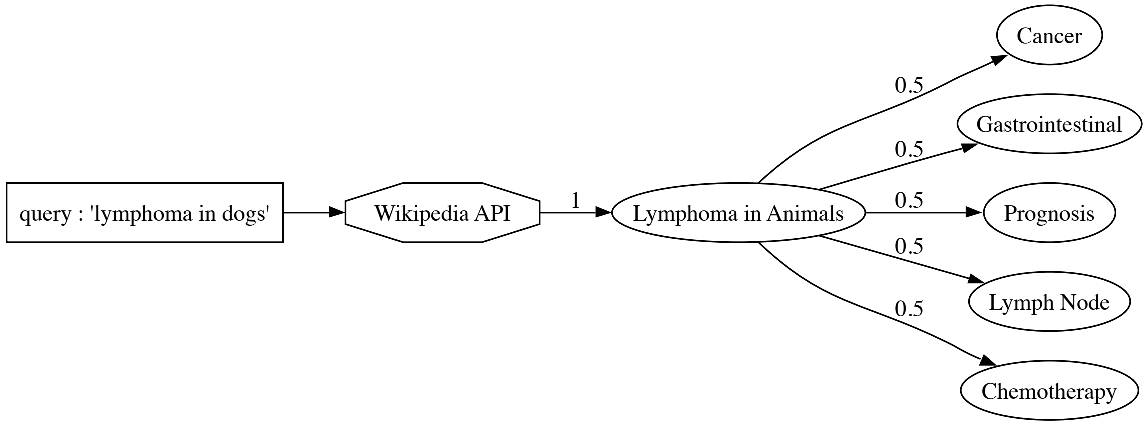

Figure 1: Generating a partial thematic graph (stopping at level 2) using anchor texts of Wikipedia pages. Query comes from topic 111.

its API to weighting the edges of the graph, is depicted on Figure 1. Here, the Wikipedia API ranks the article *Lymphoma in Animals* first for the query `lymphoma in dogs`, which is highly relevant. Informative words like ”Cancer”, ”Lymph” or ”Gastrointestinal” are extracted from this first article in order to expand the query. The algorithm also looks at the anchor texts and sees that they contain the previously selected words. If an article provides additional information about a concept or a topic that is used to expand the query, we can assume that the whole article is likely to be relevant. Hence we follow the hyperlinks and build the oriented graph.

5 Experimental Setup

We indexed the whole ClueWeb09 collection with Indri and two servers of 16 cores and 32GB of memory each.

We used the embedded stoplist along with the standard Krovetz stemmer.

We queried Wikipedia and Google simultaneously on July 2011. We used a plain Mozilla user profile to query anonymously Google, while we used the standard API for querying Wikipedia, as described in Section 3.1.1.

All the runs we submitted involve query expansion. We did not use any spam filter for all these runs. When expanding the query with a resource, with always select the top 20 words ranked according to their entropy measure. These entropies are also used to weigh the words to reflect their relative informativeness.

We submitted three runs for the Ad Hoc task:

liaQEWikiA Expanding the query with words extracted from the first Wikipedia page given by its API for a query. Document retrieval is performed over the full ClueWeb09 collection (category A).

liaQEWikiGoA Expanding the query with words extracted from the first Wikipedia page and the first Google page given by their APIs for a query. Document retrieval is also performed over the category A.

liaQEWikiGoo Same run as the previous one, but only retrieving category B documents. Considering that all our other runs are on category A, we will not discuss this run.

In the meantime we submitted one run for the Diversity task:

liaQEWikiAnA This run generates a thematic graph as described in Section 4. The two subarticles whose anchor texts contain the most expansion words are considering for building the graph. Hence, expansion words are taken from three sources: the first Wikipedia page and two children. Document retrieval is performed over the category A.

6 Discussion

6.1 Results

Results are reported in Table 1. We use a Sequential Dependence Model (**SDM**) as a competitive baseline. We set the weights in (1) as recommended by Metzler and Croft in (Metzler and Croft, 2005): $\lambda_T = 0.85$, $\lambda_O = 0.1$ and $\lambda = 0.05$.

We observe that the combination of terms extracted from Wikipedia and Google achieves the best results in terms of MAP and early nDCG and precision. However using Google alone as a source of external information achieves slightly

Run	Resources	ERR@20	nDCG@20	MAP	P@20	ERR-IA@20	α-nDCG@20
SDM (unofficial)	-	0.0409	0.0963	0.1111	0.1270	0.1661	0.2609
liaQEWikiA	**Wiki**	0.0519^{**}	0.1567^{**}	0.1323^{**}	0.2500^{***}	0.2138^{**}	0.3116
liaQEWikiAnA	**Wiki graph**	0.0606^{**}	0.1630^{**}	0.1218	0.2610^{***}	0.2287^{***}	0.3161
liaQEWikiGoA	**Wiki + Google**	0.0765^{***}	**0.1978*****	**0.1566*****	**0.2780*****	0.2769^{***}	0.3876^{***}
GooA (unofficial)	**Google**	**0.0825*****	0.1868^{***}	0.1438^{***}	0.2140^{***}	**0.3030*****	**0.3998*****

Table 1: Comparison of the retrieval performance of three of the four submitted runs and two additional runs. We use two sided paired wise Wilcoxon test (* : $p < 0.1$; ** : $p < 0.05$; *** : $p < 0.01$) to determine significant differences with baseline.

better results in terms of ERR. Using Wikipedia alone for expanding the query performs significantly better than the baseline, as well as our thematic graph approach. It is important to note that this Wikipedia thematic graph run outperforms the ”standard” use of Wikipedia for all measures excepting MAP. There was no significant differences between these two runs though.

The average P@10 and P@20 scores of our best run (liaQEWikiGoA) is slightly lower than the average of median participant score for each topic, but differences are not significant (p-value= 0.76 using wilcoxon test), meanwhile these differences are significant for runs using only one resource (p-value between 0.9 and 0.1).

It appears that each resource significantly improves the baseline on its own. MAP, P@10 and P@20 scores for GooA are significantly higher (p-value$<$ 0.001). The same significant improvement exists for P@10 and P@20 scores considering the liaQEWikiA run (p-value$<$ 0.001), even though the improvement in MAP is a bit less significant (p-value$<$ 0.1). Using Wikipedia alone performs lower than using Google alone. Apart from the PageRank effects, the fact that Google englobes the ClueWeb09 collection plays a major role. However the inherent quality of Wikipedia helps the retrieval of relevant documents at early ranks (P@20= 0.2500). More surprisingly, when combining these two external resources using a language model, the resulting run liaQEWikiGoA significantly outperforms both GooA and liaQEWikiA runs considering MAP and P@20 measures (p-value$<$ 0.1 using Wilcoxon test). We also constructed a run that optimally combines the two resources (i.e. it selects the best resource for each topic). We observe that its MAP score (0.1700) is significantly higher (p-value$<$ 0.1) than the MAP of our automatic combination based on the language model (liaQEWikiGoA). This opens perspectives for future improvements. This optimal combination could not highlight significant improvements in terms of early precision nor graded metrics, but it however achieves the best results for almost all metrics (ERR@20= 0.0790, nDCG@20= 0.2102, P@20= 0.2810).

6.2 Spam filtering

All the runs we submitted did not have any policy concerning spam documents. We wanted to see if the combination of information extracted from different resources could automatically filter this spam. Indeed, the quality of Wikipedia's content is very good because it is edited by contributors and reviewed by moderators. On the other side, one of Google's search engine last improvement consists of lowering the ranks of ”low-quality” sites such as content farmers.

We used the ”Fusion” set of spam scores for the ClueWeb09 provided by (Cormack et al., 2010)[4]. For each document in the collection, the spam list contains a percentile score, which indicates the percentage of the documents in the corpus that are ”spammier”. The authors recommend to label the documents with a percentile score below 70 as spam, and the others as non-spam. We followed these indication and pruned the spammed documents from the output of our submitted runs.

The results of our runs without any spam are reported in Table 2. It is important to note that, contrary to the runs that contain spam, there is no significant difference for all the six metrics. This behavior was also noted by other participants during the workshop: it was indeed difficult to highlight significant differences between runs that performed over non-spammed documents. The SDM baseline performs very well for all metrics and achieves the best results in terms of MAP. The best results in terms of nDCG, MAP and early precision are achieved when using both Wikipedia

[4] http://plg.uwaterloo.ca/ gvcormac/clueweb09spam/

Run	Resources	ERR@20	nDCG@20	MAP	P@20	ERR-IA@20	α-nDCG@20
NoSpamSDM	-	0.1216	0.2390	**0.1651**	0.3370	0.4184	0.5218
NoSpamWikiA	**Wiki**	0.1121	0.2425	0.1488	0.3440	0.3848	0.4765
NoSpamGooA	**Google**	0.1185	0.2426	0.1574	0.3280	0.4315	**0.5433**
NoSpamWikiGoA	**Wiki + Google**	**0.1230**	**0.2635**	0.1628	**0.3600**	**0.4345**	0.5319

Table 2: Comparison of the retrieval performance of the runs presented in Table 1 with all spammed documents pruned from the result list.

and Google as external resources. This can be explained by the fact that these two valuable and clean resources bring redundant information. The terms used for expanding the queries are often very closely related, if not synonyms. Hence a lot of similar documents are retrieved, improving topical relevance.

On the other side, using Google alone seems to favor the diversity of retrieved documents. The run NoSpamGooA achieves the best results in terms of α-nDCG@20 and its performance in ERR-IA@20 are very close to the best run. When looking again at Table 1, we see that the run GooA that only uses Google achieves the best diversity results by far.

7 Conclusions

In this paper, we detailed the runs we submitted and the experiments we conducted for the TREC 2011 Web Track. We experimented with the combination of two external resources available online, namely Wikipedia and Google, for improving web search. For this purpose we select words in first top ranked Wikipedia and Google pages in a pseudo-relevance feedback fashion and expand the original query. We also proposed a method generating thematic graphs using anchor texts and hyperlinks of Wikipedia pages. Results highlight significant improvements over a competitive baseline when searching over the entire and spammed collection, despite some effects can be attributed to the internal adjustments of the online resources we used. We also see that using several Wikipedia pages thematically linked together for selecting expansion terms helps retrieval. When applying a spam filter that removes 70 percent of the documents that are judged spams, our approaches performs better than the (strong) baseline. However these results are not statistically significant.

The results of this resource combination opens some perspectives for the future. First, we aim to do without online resources and reproduce the results with local indexes. Then we want to explore the use of more resources in order to help several search contexts and scenarios.

8 Acknowledgments

This work was supported by the French Agency for Scientific Research (Agence Nationale de la Recherche) under CAAS project (ANR 2010 CORD 001 02).

References

Michael Bendersky, David Fisher, and W. Bruce Croft. 2011. UMass at TREC 2010 Web Track: Term Dependence, Spam Filtering and Quality Bias. In *Proceedings of the Nineteenth Text REtrieval Conference (TREC 2010).*

Gordon V. Cormack, Mark D. Smucker, and Charles L. A. Clarke. 2010. Efficient and Effective Spam Filtering and Re-ranking for Large Web Datasets. *CoRR*, abs/1004.5168.

Fernando Diaz and Donald Metzler. 2006. Improving the estimation of relevance models using large external corpora. In *Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '06.

J. He, K. Balog, K. Hofmann, E. Meij, M. de Rijke, E. Tsagkias, and W. Weerkamp. 2010. Heuristic Ranking and Diversification of Web Documents. In *Proceedings of the Eighteenth Text REtrieval Conference (TREC 2009).*

Donald Metzler and W. Bruce Croft. 2005. A Markov random field model for term dependencies. In *Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '05.

Mark D. Smucker, Charles L. A. Clarke, and Gordon V. Cormack. 2010. Experiments with ClueWeb09: Relevance Feedback and Web Tracks. In *Proceedings of the Eighteenth Text REtrieval Conference (TREC 2009).*