LIA at TREC 2012 Web Track: Unsupervised Search Concepts Identification from General Sources of Information

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

Eric SanJuan
LIA - 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 2012 Web Track. We experimented a brand new system that models the latent concepts underlying a query. We use Latent Dirichlet Allocation (LDA), a generative probabilistic topic model, to exhibit highly-specific query-related topics from pseudo-relevant feedback documents. We define these topics as the latent concepts of the user query. Our approach automatically estimates the number of latent concepts as well as the needed amount of feedback documents, without any prior training step. These concepts are incorporated into the ranking function with the aim of promoting documents that refer to many different query-related thematics. We also explored the use of different types of sources of information for modeling the latent concepts. For this purpose, we use four general sources of information of various nature (web, news, encyclopedic) from which the feedback documents are extracted.

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.

The goal of the work presented in this paper is to accurately represent the underlying core concepts involved in a search process, hence indirectly improving the contextual information surrounding this search. For this purpose, we introduce an unsupervised framework that allows to track the implicit concepts related to a given query and improve document retrieval effectiveness by incorporating these concepts to the initial query. For each query, latent concepts are extracted from a reduced set of feedback documents initially retrieved by the system. These feedback documents can come from the target collection or from any other textual source of information.

The main strength of our approach is that it is entirely unsupervised and does not require any training step. The number of needed feedback documents as well as the optimal number of concepts are automatically estimated at query time. We emphasize that the algorithms have no prior information about these concepts. The method is also entirely independent of the source of information used for concept modeling. Queries are not labelled with topics or keywords and we do not manually fix any parameter at any time, except the number of words composing the concepts.

2 Query-Oriented LDA

w $P(w|k_1)$ $P(w|k_2)$ $P(w|k_3)$ $P(w|k_4)$ δ_1 δ_2 δ_3 δ_4

2.1 Latent Dirichlet Allocation

Latent Dirichlet Allocation is a generative probabilistic topic model (Blei et al., 2003). The underlying intuition is that documents exhibit multiple

topics, where a *topic* is a multinomial distribution over a fixed vocabulary W. The goal of LDA is thus to automatically discover the topics from a collection of documents. The documents of the collection are modeled as mixtures over K topics each of which is a multinomial distribution over W. Each topic multinomial distribution ϕ_k is generated by a conjugate Dirichlet prior with parameter β, while each document multinomial distribution θ_d is generated by a conjugate Dirichlet prior with parameter α. Thus, the topic proportions for document d are θ_d, and the word distributions for topic k are ϕ_k. In other words, $\theta_{d,k}$ is the probability of topic k occurring in document d (i.e. $P(k|d)$). Respectively, $\phi_{k,w}$ is the probability of word w belonging to topic k (i.e. $P(w|k)$). Exact LDA estimation was found to be intractable and several approximations have been developed (Blei et al., 2003; Griffiths and Steyvers, 2004). We use in this work the variational approximation algorithm implemented and distributed by Pr. Blei[1].

Each learned multinomial distribution ϕ_k is traditionally presented as list of the top words with the higher probabilities for topic k. Topics can then be easily identified by their most representative words.

2.2 Estimating the number of concepts

There can be a numerous amount of concepts underlying an information need. Latent Dirichlet Allocation allows to model the topic distribution of a given collection, but the number of topics is a fixed parameter. However we can not know in advance the number of concepts that are related to a given query. We propose a method that automatically estimates the number of latent concepts based on their word distributions.

Considering LDA's topics are constituted of the n words with highest probabilities, we define an $\mathrm{argmax}[n]$ operator which produces the top-n arguments that obtain the n largest values for a given function. Using this operator, we obtain the set $\mathbb{W}_k$ of the n words that have the highest probabilities $P(w|k) = \phi_{k,w}$ in topic k:

$$\mathbb{W}_k = \underset{w}{\mathrm{argmax}}[n]\ \phi_{k,w}$$

Latent Dirichlet Allocation needs a given number of topics in order to estimate topic and word distributions. Several approaches has been studied for automatically finding the right number of LDA's topics contained in a set of documents (Arun et al., 2010; Cao et al., 2009). Even though they differ at some point, they follow the same idea of computing similarities (or distances) between pairs of topics over several instances of the model, while varying the number of topics. It comes down to a clustering approach which delineates the different clusters. Here the clusters are the topics and the objective is to maximize the dissimilarity between topics. Iterations are done by varying the number of topics of the LDA model, then estimating again the Dirichlet distributions. The optimal amount of topics of a given collection is reached when the overall dissimilarity between topics achieves its maximum value.

We perform a simple heuristic that estimates the number of latent concepts of a user query by maximizing the information divergence D between all pairs (k_i, k_j) of LDA's topics. The number of concepts $\hat{K}$ estimated by our method is given by the following formula:

$$\hat{K}(m) = \underset{K}{\mathrm{argmax}}\ \frac{1}{K(K-1)} \sum_{(k_i,k_j)\in\mathbb{T}_{K,m}} D(k_i||k_j)$$

where K is the number of topics given as a parameter to LDA, and $\mathbb{T}_K$ is the set of K topics. In other words, $\hat{K}$ is the number of topics for which LDA modeled the most scattered topics. The Kullback-Leibler divergence measures the information divergence between two probability distributions. It is used particularly by LDA in order to minimize topic variation between two expectation-maximization iterations (Blei et al., 2003). It has also been widely used in a variety of fields to measure similarities (or dissimilarities) between word distributions (AlSumait et al., 2008). Considering it is a non-symmetric measure we use the Jensen-Shannon divergence, which is the symmetric version of KL divergence, to avoid obvious problems when computing divergences between all pairs of topics:

$$\begin{aligned} D(k_i||k_j) &= \frac{1}{2}\sum_{w\in W} p(w|k_i)\log\frac{p(w|k_i)}{p(w|k_j)} + \\ &\quad \frac{1}{2}\sum_{w\in W} p(w|k_j)\log\frac{p(w|k_j)}{p(w|k_i)} \end{aligned}$$

The word probabilities for given topics are obtained from the multinomial distributions ϕ_k.

[1] http://www.cs.princeton.edu/~blei/lda-c

Each word w of the vocabulary W has a probability of belonging to the topic k, which is expressed by $p(w|k) = \phi_{k,w}$. The final outcome is the optimal number of topics $\hat{K}$ and its associated topic model. The resulting $\mathbb{T}_{\hat{K},M}$ set of topics is considered as the set of $\hat{K}$ latent concepts modeled from a set of M feedback documents. We will further refer to the $\mathbb{T}_{\hat{K},M}$ set as a *concept model*.

The number of relevant documents can vary from one query to another, hence the number M of feedback documents used to model the latent concepts must also vary for each query. It is also highly dependent on the source of information from which the feedback documents are extracted. We propose in the following section a method for automatically choosing the right amount M of feedback documents based on *concept models* similarities.

2.3 How many feedback documents?

An obvious problem with pseudo-relevance feedback based approaches is that not-relevant documents can be included in the set of feedback documents. This problem is much more important with our approach since it could result with learned concepts that are not related to the initial query. We mainly tackle this difficulty by reducing the amount of feedback documents. Relevant documents concentration is higher in the top ranks of the list. Thus one simple way to reduce the probability of catching noisy feedback documents is to reduce their overall amount.

However an arbitrary number can not be fixed for all queries. Some information needs can be satisfied by only 2 or 3 documents, while others may require 15 or 20. Thus the choice of the feedback documents amount has to be automatic for each query. To this end, we compare the concept models generated from different amounts m of feedback documents. To avoid noise, we favor the concept models that contain concepts that are similar to others in other models. The underlying assumption is that all the feedback documents are essentially dealing with the same topics, no matter if they are 5 or 20. Concepts that are likely to appear in different models learned from various amounts of feedback documents are certainly related to query, while noisy concepts are not.

We estimate the similarity between two concept models by computing the similarities between all pairs of concepts of the two models. Considering that two concept models are generated based on different number of documents (i.e. different $\mathcal{R}_Q$ collections), they do not share the same probabilistic space. Since their probability distribution are not comparable, computing their overall similarity can be done solely by taking the concepts word distributions into account. We treat the different concepts as bags of words and use a document frequency-based similarity measure:

$$sim(\mathbb{T}_{\hat{K}(m)}, \mathbb{T}_{\hat{K}(n)}) = \sum_{k \in \mathbb{T}_{\hat{K}(m)}} \sum_{k' \in \mathbb{T}_{\hat{K}(n)}} \frac{|k_i \cap k'_j|}{|k_i|} \sum_{w \in W} p(w|k) p(w|k') \log \frac{N}{df_w}$$

where $|k_i \cap k_j|$ is the number of words the two concepts have in common, df_w is the document frequency of w and N is the number of documents in the target collection. The initial purpose of this measure was to track novelty (i.e. minimize similarity) between two sentences (Metzler et al., 2005), which is precisely our goal, except that we want to track redundancy (i.e. maximize similarity) while taking word probabilities inside the topics into account.

The final sum of similarities between each concept pairs produces an overall similarity score of the current concept model compared to all other models. Finally, the concept model that maximizes this overall similarity is considered as the best candidate for representing the implicit concepts of the query. In other words, we consider the top M feedback documents for modeling the concepts, where

$$M = \underset{m}{\operatorname{argmax}} \sum_{n} sim(\mathbb{T}_{\hat{K}(m)}, \mathbb{T}_{\hat{K}(n)})$$

In other words, for each query, the concept model that is the most similar to all other learned concept models is considered as the final set of latent concepts related to the user query.

2.4 Concept weighting

We previously detailed how we estimate the number of concepts and the number of feedback documents from which they are extracted. We face in this section the problem of appropriately weighting these concepts.

User queries can be associated with a number of underlying concepts but these concepts do not have the same importance. For example, the previous method for selecting the right amount of

feedback documents could still yield noisy concepts, and some concepts may also be barely relevant.Hence it is essential to emphasize appropriate concepts and to depreciate inappropriate ones. One effective way is to rank these concepts and weigh them accordingly: important concepts will be weighted higher, thus reflecting their importance.

Recent studies proposed different approaches to rank or score LDA topics (Alsumait et al., 2009; Newman et al., 2010; Wen and Lin, 2010), however .

Finally, the score δ_k of a concept k with respect to its overall coherence in the collection is given by:

$$\delta_k = \sum_{d \in \mathcal{R}_Q} p(d|Q)p(k|d)$$

where n is the number of words in each concept. The probability of a concept k appearing in document d is given by the multinomial distribution θ previously learned by LDA, hence $p(k|d) = \theta_{d,k}$.

Each concept is weighted with respect to its coherence in the target collection, but the actual representation of the concept is still a bag of words. These words are the core components of the concepts and intrinsically do not have the same importance. The easier way of weighting them is to use their probability of belonging to a concept k which are learned by Latent Dirichlet Allocation and given by the multinomial distribution ϕ_k. Probabilities are normalized across all words, the weight of word w in concept k is thus computed as follows:

$$\hat{\phi}_{k,w} = \frac{\phi_{k,w}}{\sum_{w' \in \mathbb{W}_k} \phi_{k,w'}}$$

Finally, a concept learned by our latent concept modeling approach is a set of weighted words representing a facet of the information need underlying a user query. The concept is itself weighted to reflect its relative importance with other concepts.

2.5 Document ranking

The previous subsections were all about modeling consistent concepts from reliable documents and modeling their relative influence. Here we detail how these concepts can be integrated in a retrieval model in order to improve ad-hoc document ranking.

There are several ways of taking conceptual aspects into account when ranking documents. Here, the final score of a document d with respect to a given user query Q is determined by the linear combination of query word matches (standard retrieval) and latent concepts matches. It is formally written as follows:

$$s(Q,d) = P(d|Q) + \sum_{k \in \mathbb{T}_{\hat{K},M}} \hat{\delta}_k \sum_{w \in \mathbb{W}_k} \hat{\phi}_{k,w} \cdot P(w|d)$$

where $\mathbb{T}_{\hat{K},M}$ is the *concept model* that holds the latent concepts of query Q (see Section 2.4) and $\hat{\delta}_k$ is the normalized weight of concept k:

$$\hat{\delta}_k = \frac{\delta_k}{\sum_{k' \in \mathbb{T}_{\hat{K},M}} \delta_{k'}}$$

The $P(d|Q)$ and $P(d|w)$ probabilities are the likelihood of document d being observed given the initial query Q (respectively, word w). In this work we use a language modeling approach to retrieval (Lavrenko and Croft, 2001), $P(d|w)$ is thus the maximum likelihood estimate of word w in document d, computed using the language model of document d in the target collection $\mathcal{C}$. Likewise, $P(d|Q)$ is the basic language modeling retrieval model, also known as query likelihood, and can also be formally written as $P(d|Q) = \sum_{q \in Q} P(d|q)$. We tackle the null probabilities problem with the standard Dirichlet smoothing since it is more convenient for keyword queries (as opposed to verbose queries) (Zhai and Lafferty, 2004), which is the case here. We fix the Dirichlet prior parameter to 1500 and do not change it at any time during our experiments. However it is important to note that this model is generic, and that the word matching function could be entirely substituted by other state-of-the-art matching function (like BM25 (Robertson and Walker, 1994) or information-based models (Clinchant and Gaussier, 2010)) without changing the effects of our latent concept modeling approach on document ranking.

3 General Sources of Information

The approach described in the previous section requires a source of information from which the concepts could be extracted. This source of information can come from the target collection, like in traditional relevance feedback approaches, or from an external collection. In this work we use a set of different data sources that are large enough to deal with a broad range of topics. Then we can explore which effects does the nature, the size or the

quality of the information source have over latent concept modeling.

This set of data sources is composed of four general resources: Wikipedia as an encyclopedic source, the New York Times and GigaWord corpora as sources of news data and the category B of the ClueWeb09[2] collection as a web source. The English GigaWord LDC corpus consists of 4,111,240 newswire articles collected from four distinct international sources including the New York Times (Graff and Cieri, 2003). The New York Times LDC corpus contains 1,855,658 news articles published between 1987 and 2007 (Sandhaus, 2008). The Wikipedia collection is a recent dump from July 2011 of the online encyclopedia that contains 3,214,014 documents[3]. We removed the spammed documents from the category B of the ClueWeb09 according to a standard list of spams for this collection[4]. We followed authors recommendations (Cormack et al., 2011) and set the "spamminess" threshold parameter to 70. The resulting corpus is composed of 29,038,220 web pages.

Resource	# documents	# unique words	# total words
NYT	1,855,658	1,086,233	1,378,897,246
Wiki	3,214,014	7,022,226	1,033,787,926
GW	4,111,240	1,288,389	1,397,727,483
Web	29,038,220	33,314,740	22,814,465,842

Table 1: Information about the four general sources of information used in this work.

These four resources are heterogeneous in all possible ways. They vary in terms of vocabulary size, number of documents and, of course, type of information. We thus expect that latent concepts will be as diverse as the sources of information from which their are modeled.

4 Experiments

4.1 Experimental setup

We used Indri[5] for indexing and retrieval. The whole ClueWeb09 collection was stemmed during indexing with the well-known light Krovetz stemmer, and stopwords were removed using the standard english stoplist embedded within Indri. We also removed from our index all the documents that have a spam percentile lower than 70 according to Waterloo's list[4]. As seen in Section 2, concepts are composed of a fixed amount of weighted words For all our runs we fixed the number of words belonging to a given concept to $n = 10$.

[2] http://boston.lti.cs.cmu.edu/clueweb09/

[3] http://dumps.wikimedia.org/enwiki/20110722/

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

[5] http://www.lemurproject.org

4.2 Runs

We submitted four runs in which we explore the influence of the number of feedback documents used for concept modeling, the concept weights and combining the general sources of information.

lcm-web This is our reference run. It uses the complete concept modeling approach described in this paper, but the feedback documents from which the concepts are modeled are solely extracted from the Web source of information (see Section 3).

lcm-web-noW This run is the same as above, except that we removed the concept weights (the $\hat{\delta}_k$s). The word weights (the $\hat{\phi}_k$s) are still present in the ranking function.

lcm-web-10p This run is identical to **lcm-web**, except that we fix the number of feedback documents to $M = 10$.

lcm-4res This last run uses our concept modeling on the four general sources of information presented earlier. The *concept models* issued from the different sources are combined in the final document ranking function:

$$s(Q,d) = P(d|Q) + \frac{1}{|\mathcal{S}|} \sum_{\sigma \in \mathcal{S}} \sum_{k \in \mathbb{T}_{\hat{K},M}(\sigma)} \hat{\delta}_k \sum_{w \in \mathbb{W}_k} \hat{\phi}_{k,w} \cdot P(w|d)$$

where $\mathcal{S}$ is the set of sources of information and $\mathbb{T}_{\hat{K},M}(\sigma)$ is the *concept model* composed of $\hat{K}$ concepts modeled from M feedback documents which were extracted from a source σ.

4.3 Results

We report in this section the results of our runs for both the Ad Hoc (Table 2) and the diversity metrics (Table 3). We also present the results of a standard competitive baseline, the Markov Random Field for IR (Metzler and Croft, 2005), as a mean of comparison. We chose the Sequential Dependance Model instantiation of this model and set the various weights as recommended by the authors ($\lambda_T = 0.85$, $\lambda_O = 0.1$ and $\lambda_U = 0.05$).

This baseline showed to be highly effective in previous TREC tracks, and especially in those involving web documents. For both table of results, we use two sided paired wise t-test to determine statistically significant differences with MRF-IR ($^* : p < 0.1$; $^{**} : p < 0.05$; $^{***} : p < 0.01$).

Run	ERR@20	nDCG@20
MRF-IR	0.1038	0.1041
lcm-web	0.1334**	0.1306**
lcm-web-noW	0.1352**	0.1337*
lcm-web-10p	0.1364***	0.1339*
lcm-4res	**0.1428*****	**0.1401*****
term-web (2011)	0.1470**	0.1328**
term-4res (2011)	0.1649***	0.1511***

Table 2: Ad Hoc results for our four submitted runs.

Although there is not much difference in averaged scores between our four runs, we see that **lcm-4res** achieves highly significant improvements over the MRF-IR baseline. More, the three other runs fail to retrieve any relevant document in the top 20 ranks (ERR@20 = nDCG@20 = 0) for 13 topics, while the **lcm-4res** approach only fails for 9 topics. It is however interesting to note that MRF-IR fails on the same topics as our Latent Concept Modeling (LCM) approaches. It may be an language modeling issue, and it may be interesting to compare with other participants that explored other retrieval models. The indexing of only non-spammed documents could also be an explanation and needs further exploration.

When looking at runs individually, fixing the number of feedback documents to 10 achieves better results on average than using an adaptive method. Despite improvements of **lcm-web-10p** over MRF-IR are less significant than **lcm-web** for nDCG@20, the gain in computation time seems to be worth fixing M.

As for the diversity, removing the concept weights seems to improve the results on average, however **lcm-4res** achieves again higher statistically significant improvements than the other runs. It also reduces the number of topic failures to only one compared to 4 for the other runs and 5 to MRF-IR.

Overall, the influence of concept weighting is rather low. When comparing results topic per topic between **lcm-web** and **lcm-web-noW**, we see no significant differences. This is certainly due to the fact that all the concepts refer to common thematics and share the same vocabulary. Plus, using a small amount of feedback documents leads to computing LDA in a reduced probabilistic space. Hence, some very important words w.r.t to the query are present in every concept, thus diminishing the effect and the interest of concept weighting.

Run	ERR-IA@20	α-nDCG@20	P-IA@20
MRF-IR	0.2662	0.3653	0.1955
lcm-web	0.3166*	0.4160**	0.2501***
lcm-web-10p	0.3110*	0.4115*	0.2427***
lcm-4res	0.3176**	**0.4240*****	0.2479***
lcm-web-noW	**0.3205***	0.4194**	**0.2503*****
term-web (2011)	0.3417**	0.4302**	0.2475***
term-4res (2011)	0.3522***	0.4557***	0.2666***

Table 3: Diversity results for our four submitted runs.

5 Conclusion

This paper detailed the run we submitted to the TREC 2012 Web track. Our approach was to model the latent concepts that are underlying an information need. The goal was to broaden the scope of the search and ultimately promoting retrieval diversity, without hurting topical relevance.

Official results suggest that our approach works quite well for both ad hoc and diversity metrics. The use of several sources of information (instead of sticking to the target collection) is found useful in this context.

6 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

Loulwah AlSumait, Daniel Barbará, and Carlotta Domeniconi. 2008. On-line LDA: Adaptive Topic Models for Mining Text Streams with Applications to Topic Detection and Tracking. In *Proceedings of the 2008 Eighth IEEE International Conference on Data Mining*, ICDM '08.

Loulwah Alsumait, Daniel Barbará, James Gentle, and Carlotta Domeniconi. 2009. Topic Significance

Ranking of LDA Generative Models. In *Proceedings of the European Conference on Machine Learning and Knowledge Discovery in Databases*, ECML PKDD '09.

R. Arun, V. Suresh, C. Veni Madhavan, and M. Narasimha Murthy. 2010. On Finding the Natural Number of Topics with Latent Dirichlet Allocation: Some Observations. In *Advances in Knowledge Discovery and Data Mining*, volume 6118 of *Lecture Notes in Computer Science*.

David M. Blei, Andrew Y. Ng, and Michael I. Jordan. 2003. Latent Dirichlet Allocation. *Journal of Machine Learning Research*, 3.

Juan Cao, Tian Xia, Jintao Li, Yongdong Zhang, and Sheng Tang. 2009. A density-based method for adaptive LDA model selection. *Neurocomputing*, 72(7-9).

Stéphane Clinchant and Eric Gaussier. 2010. Information-based models for ad hoc IR. In *Proceedings of the 33rd international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '10.

Gordon Cormack, Mark Smucker, and Charles Clarke. 2011. Efficient and effective spam filtering and reranking for large web datasets. *Information Retrieval*.

David Graff and Christopher Cieri. 2003. English Gigaword. *Philadelphia: Linguistic Data Consortium*, LDC2003T05.

Thomas L Griffiths and Mark Steyvers. 2004. Finding scientific topics. *Proceedings of the National Academy of Sciences of the United States of America*, 101 Suppl.

Victor Lavrenko and W. Bruce Croft. 2001. Relevance based language models. In *Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '01.

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.

Donald Metzler, Yaniv Bernstein, W. Bruce Croft, Alistair Moffat, and Justin Zobel. 2005. Similarity measures for tracking information flow. In *Proceedings of the 14th ACM international conference on Information and knowledge management*, CIKM '05.

David Newman, Jey Han Lau, Karl Grieser, and Timothy Baldwin. 2010. Automatic evaluation of topic coherence. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, HLT '10.

S. E. Robertson and S. Walker. 1994. Some simple effective approximations to the 2-Poisson model for probabilistic weighted retrieval. In *Proceedings of the 17th annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '94.

Evan Sandhaus. 2008. The New York Times Annotated Corpus. *Philadelphia: Linguistic Data Consortium*, LDC2008T19.

Zhen Wen and Ching-Yung Lin. 2010. Towards Finding Valuable Topics. In *Proceedings of the SIAM International Conference on Data Mining*, SDM '10.

Chengxiang Zhai and John Lafferty. 2004. A study of smoothing methods for language models applied to information retrieval. *ACM Transactions on Information Systems*, 22(2).