

LIA at the NTCIR-10 INTENT Task

Romain Deveaud

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

Eric SanJuan

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

ABSTRACT

This paper describes the participation of the LIA team in the English Subtopic Mining subtask of the NTCIR-10 INTENT-2 Task. The goal of this task was to specialize or disambiguate web search queries by identifying the different subtopics that could refer to these queries. Our motivation was to take a conceptual approach, therefore representing the query by a set of concepts before identifying the related subtopics. However we seem to have misunderstood the real point of this task, which was in fact focused on generating web query suggestion: official results therefore do not show support for our initial motivation.

Team Name

LIA

Subtasks

Subtopic Mining (English)

Keywords

topic modeling, latent concepts, entity linking

1. INTRODUCTION

Web queries can often be ambiguous or under-specialized. In other words, the set of keywords given by a user to a search engine may not fully describe the underlying information need. More, query misconstruction may be due to a lack of knowledge from the user. Discovering or inferring query subtopics is thus challenging in order to guide the user towards a better expression of her information need.

For our participation to the NTCIR-10 INTENT Subtopic Mining task [9], we wanted to model the core concepts of a query in order to better identify clusters of subtopics. For this purpose, we used the unsupervised latent concept modeling framework that we previously experimented on TREC Robust and Web collections [3, 4] (including the 50 english queries that were used for the Subtopic Mining task). For each query, latent concepts are extracted from a reduced set of feedback documents initially retrieved by the system from a textual source of information. We then used these concepts to identify related Wikipedia entities. Entities that are related to several concepts are given higher scores, and their title constitute the subtopic labels.

The remainder of this paper is organized as follows. Section 2 reviews the principles of latent concept modeling and

formalizes the generation of the subtopics. Section 3 describes the text collections we used for modeling the concepts as well as the runs we submitted, while Section 4 concludes the paper.

2. OUR APPROACH

2.1 Latent Concept Modeling

Latent Concept Modeling [4] is built based on the assumption of Relevance Models [7] (and more generally Pseudo-Relevance Feedback approaches) that the concentration of relevant information with respect to a query is higher in the top ranked documents retrieved for that query. It uses Latent Dirichlet Allocation [1] (LDA) to cluster words from these feedback documents into *topics* or *concepts*.

LDA is a generative probabilistic topic model. 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 . 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 [1, 6]. We use in this work the variational approximation algorithm implemented and distributed by Pr. Blei¹.

Latent concept modeling then comes down to performing LDA on the top- M feedback documents automatically retrieved using the initial query. However LDA needs a number of topics K as a parameter, but the number of latent concepts can vary from one query to another and cannot be fixed in advance. Likewise, an obvious problem with pseudo-relevance feedback based approaches is that not-relevant documents can be included in the set of feedback documents. The number M of top feedback documents thus also need to be estimated for each query. To tackle these problems, latent concept modeling provides an expectation-maximization algorithm that jointly estimates K^* and M^* .

Given a topic model T_K^M computed on the top- M feedback documents with K topics, K^* is estimated by computing the

¹<http://www.cs.princeton.edu/~blei/lda-c>

divergence between all pairs of topics (k_i, k_j) :

$$K_M^* = \operatorname{argmax}_K \frac{1}{K(K-1)} \sum_{(k_i, k_j) \in \mathbb{T}_K^M} D(k_i || k_j)$$

where $D(k_i || k_j)$ is the Kullback-Leibler divergence between topic k_i and topic k_j . Latent concept modeling thus estimates K^* values for various values of M .

At this point, M concept models are generated. The estimation of the best model among them is done by computing similarities between models, and choosing the one that is the most similar with respect to the others. The underlying assumption is that relevant documents are essentially dealing with the same topics, regardless of their number. 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. Since concepts are computed from different documents, their probability distributions are not comparable. They are thus treated as bags of words and compared using a document frequency-based similarity measure:

$$M^* = \operatorname{argmax}_M \sum_{n=1}^M \sum_{k_j \in \mathbb{T}_{K_M^*}^M} \sum_{k_i \in \mathbb{T}_{K_N^*}^N} \frac{|k_i \cap k_j|}{|k_i|} \sum_{w \in k_i \cap k_j} \log \frac{N}{df_w}$$

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

The resulting concept model $\mathbb{T}_{K_M^*}^{M^*}$ represents the latent concepts of the initial query used to retrieve the top- M^* feedback documents. In our submissions for the Submining task, we let K and M vary between 1 and 20. Concepts were truncated and only the 10 words with highest probabilities were kept.

2.2 Subtopic labels generation

After having generated a set of latent concepts for the query, the next step of our approach is to find candidate labels for these concepts and score them. We thought Wikipedia could provide interesting entries to user in order to specialize or disambiguate their queries, this is the reason why the candidate labels actually are titles of Wikipedia articles. For each concept, the 4 words with highest probabilities are considered as a query. All queries are submitted to a static index of the May 2012 version of Wikipedia and to the live version of Wikipedia through its API². The titles of the articles that are retrieved from these two searches are considered as candidate labels. Those that appear in only one of the two ranked lists are discarded.

We based our scoring function on the work done by Mei et al. [8] for automatically labeling topic models. We compute co-occurrence probabilities between concepts words and candidate labels in a reference collection \mathcal{C} to determine the score of a label l :

$$s(l) = \sum_{k \in \mathbb{T}_{K_M^*}^{M^*}} \hat{\delta}_k \sum_{w \in k} P(w|k) \log \frac{P(w, l|\mathcal{C})}{P(w|\mathcal{C})P(l|\mathcal{C})}$$

²http://www.mediawiki.org/wiki/API:Main_page

where:

$$\hat{\delta}_k = \frac{\delta_k}{\sum_{k'} \delta_{k'}}$$

and:

$$\delta_k = \sum_D P(Q|D)P(k|D)$$

Hence, labels that have the highest probabilities of occurrence with respect to words of all latent concepts are given the highest scores.

3. OFFICIAL EXPERIMENTS

3.1 External collections

In this work we use a set of different data sources from which the latent concepts are modeled: Wikipedia as an encyclopedic source, the New York Times and GigaWord corpora as sources of news data and the category B of the ClueWeb09³ 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 [5]. The New York Times LDC corpus contains 1,855,658 news articles published between 1987 and 2007 [10]. The Wikipedia collection is a dump from July 2011 of the online encyclopedia that contains 3,214,014 documents⁴. We removed the spammed documents from the category B of the ClueWeb09 according to a standard list of spams for this collection⁵. We followed authors recommendations [2] and set the "spamminess" threshold parameter to 70. The resulting corpus is composed of 29,038,220 web pages.

All collections were indexed by Indri⁶ with the exact same parameters: tokens were stemmed with the well-known light Krovetz stemmer, and stopwords were removed using the standard English stoplist embedded with Indri.

3.2 Runs

LIA-S-E-1A.

In this run, we modeled the concepts from the four external collections described above. The subtopics solely contain titles Wikipedia articles.

LIA-S-E-2A.

This run is similar to the previous one, except that M^* is not estimated but fixed at $M = 10$ for all queries. Concepts are thus modeled from the top-10 feedback documents.

LIA-S-E-3A.

This run is the same as LIA-S-E-1A, except that we use the commercial search engines suggestions provided by the organizers. The query used in this case is the concatenation of all completions available (without duplicates).

LIA-S-E-4A.

This run is the same as LIA-S-E-1A, except that the original query is inserted before each subtopic label.

³<http://boston.lti.cs.cmu.edu/clueweb09/>

⁴<http://dumps.wikimedia.org/enwiki/20110722/>

⁵<http://plg.uwaterloo.ca/~gvcormac/clueweb09spam/>

⁶<http://www.lemurproject.org>

3.3 Results

Table 1 reports the official results of our 4 runs. It turns out that our first 3 runs are the ones that achieved the lowest results. However we see that run LIA-S-E-4A achieves strong improvements over the others. This run is exactly the same as LIA-S-E-1A but with the original query prepended to each subtopic labels. It thus seems that we misconceived the goal of the track and did not expect that there would be such a dramatical gap between the results of LIA-S-E-1A and LIA-S-E-4A.

run	I-rec@10	D-nDCG@10	D#-nDCG@10
LIA-S-E-1A	0.0291	0.0420	0.0355
LIA-S-E-2A	0.0328	0.0474	0.0401
LIA-S-E-3A	0.0377	0.0329	0.0353
LIA-S-E-4A	0.2000	0.2753	0.2376
S-E-1A*	0.2000	0.2753	0.2376
S-E-2A*	0.0392	0.0569	0.0481
S-E-3A*	0.0812	0.0773	0.0793

Table 1: Official evaluation of our four runs for the English Subtopic mining task. Unofficial experiments are marked with *.

In the earlier version of the relevance judgments, LIA-S-E-3A achieved higher results than the two others (1A and 2A). However this ranking changed completely with the revised version of the judgments, preventing us from learning anything from these results. We nonetheless conducted additional experiments in order to see whether adding original queries to the subtopics labels would at least reach the performance of LIA-S-E-4A. We used the evaluation toolkit and the revised relevance judgments provided by the organizers.

The resulting runs are presented in the lower part of Table 1, and show very little improvements over their “no-query” versions. In these new runs, new subtopic labels are generated and they may not be present in the relevance judgments. Since no other team chose to extract labels from Wikipedia titles, there is little chance that the exact same labels have been judged. S-E-1A achieves logically the same results as LIA-S-E-4A.

4. CONCLUSIONS

We tried an unsupervised clustering approach on pseudo-relevant feedback documents in order to discover the latent concepts of a query. These concepts were used to find Wikipedia entries which titles were supposed to represent the subtopics of the query. However it turns out that we somewhat misconceived the purpose of the task, and the results offer little insights on which parts of our system failed.

5. REFERENCES

- [1] D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent dirichlet allocation. *J. Mach. Learn. Res.*, 3:993–1022, Mar. 2003.
- [2] G. Cormack, M. Smucker, and C. Clarke. Efficient and effective spam filtering and re-ranking for large web datasets. *Information Retrieval*, 2011.
- [3] R. Deveaud, E. SanJuan, and P. Bellot. LIA at TREC 2012 Web Track: Unsupervised Search Concepts Identification from General Sources of Information. In E. M. Voorhees and L. P. Buckland, editors, *TREC*. National Institute of Standards and Technology (NIST), 2012.
- [4] R. Deveaud, E. SanJuan, and P. Bellot. Unsupervised Latent Concept Modeling to Identify Query Facets. In *Proceedings of the 10th International Conference in the RIAO series on Open Research Areas in Information Retrieval*, OAIR ’13, 2013.
- [5] D. Graff and C. Cieri. English Gigaword. *Philadelphia: Linguistic Data Consortium*, LDC2003T05, 2003.
- [6] T. L. Griffiths and M. Steyvers. Finding scientific topics. *Proceedings of the National Academy of Sciences of the United States of America*, 101 Suppl, 2004.
- [7] V. Lavrenko and W. B. Croft. Relevance based language models. In *Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR ’01, pages 120–127, New York, NY, USA, 2001. ACM.
- [8] Q. Mei, X. Shen, and C. Zhai. Automatic labeling of multinomial topic models. In *Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD ’07, pages 490–499, New York, NY, USA, 2007. ACM.
- [9] T. Sakai, Z. Dou, T. Yamamoto, Y. Liu, M. Zhang, and R. Song. Overview of the NTCIR-10 INTENT-2 Task. In *Proceedings of NTCIR-10*, 2013.
- [10] E. Sandhaus. The New York Times Annotated Corpus. *Philadelphia: Linguistic Data Consortium*, LDC2008T19, 2008.