LIA/LINA at the INEX 2012 Tweet Contextualization track

Romain Deveaud, Florian Boudin

LIA/LINA at the INEX 2012 Tweet Contextualization track

Romain Deveaud[1] and Florian Boudin[2]

[1] LIA - University of Avignon
romain.deveaud@univ-avignon.fr

[2] LINA - University of Nantes
florian.boudin@univ-nantes.fr

Abstract. In this paper we describe our participation in the INEX 2012 Tweet Contextualization track and present our contributions. We combined Information Retrieval, Automatic Summarization and Topic Modeling techniques to provide the context of each tweet. We first formulate a specific query using hashtags and important words in the Tweets to retrieve the most relevant Wikipedia articles. Then, we segment the articles into sentences and compute several measures for each sentence, in order to estimate their contextual relevance to the topics expressed by the Tweets. Finally, the best scored sentences are used to form the context. Official results suggest that our methods performed very well compared to other participants.

1 Introduction

The INEX Tweet Contextualization tracks aims at providing a small bunch of text (less than 500 words) that gives insights or additional information about a given tweet. For example, when reading a tweet about Whitney Houston's funerals, the user might want to know who is this person, why is she famous and so on... One of the strict constraint was to extract this context from a Wikipedia collection provided by the organizers, so there were several challenges to tackle.

First, it was very important to retrieve relevant and important Wikipedia articles that were related to the Tweets, and that were likely to provide some useful context. Second, considering the word limit of the contexts, only very little parts of these articles had to be kept. For this purpose we segmented the top-ranked articles into sentences and used several measures to score them. These measures range from classic word overlap or cosine similarity to conceptual similarity using topic models.

The rest of the paper is organized as follows. Section 2 describes the process we followed to extract candidate sentences, which includes Tweet formatting and document retrieval on Wikipedia. Then, we describe in Section 3 the various sentence scoring methods we used in this work.

2 Candidate Sentence Extraction

Considering that the task is to provide context from Wikipedia text, one crucial step was to retrieve Wikipedia articles that are highly relevant to the Tweet. Hopefully, relevant articles contain important sentences that give lots of contextual information.

2.1 #HashtagSplitting and Tweet formatting

Hashtags in Tweets are very important pieces of information, since they are tags that were generated by the user. Making a parallel with TREC-like topics, we can view the hashtags as the title while the Tweet itself is the description.

However the main problem with hashtags is that they often are composed of several words concatenated together (e.g. #WhitneyHouston). We used an algorithm based on Peter Novig's chapter on "Natural Language Corpus Data" in [5] to split the hashtags. For each Tweet, all the hashtags we converted into a short keyword query.

We also removed all the retweet mentions (RT), user mentions (@somebody) and stopwords (based on the standard INQUERY stoplist) from the Tweets. The final output of this Tweet formatting process is a clean Tweet without stopwords or useless mentions, as well as a very short and user-generated representation of this Tweet.

2.2 Retrieving Wikipedia articles

Retrieving relevant Wikipedia articles is the first crucial part for finding contextually relevant sentences. For this purpose we use the well-known Markov Random Field model [3] to represent dependencies between query words. It has indeed performed consistently well on several variety of ad-hoc search tasks across the years.

Given an initial Tweet $\mathcal{T}$, the output of the method described in the previous section is a set of hashtags $H_{\mathcal{T}}$ and a set of terms $Q_{\mathcal{T}}$. We then score Wikipedia articles D according to the following function:

$$s(H_{\mathcal{T}}, Q_{\mathcal{T}}, D) = \lambda \times score_{MRF}(H_{\mathcal{T}}, D) + (1-\lambda) score_{MRF}(Q_{\mathcal{T}}, D)$$

where λ is a free smoothing parameter which was empirically set to 0.8 for all our experiments. We used the Sequential Dependance Model instantiation of MRF, which is defined as follows:

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

where the features weights are set according to the author's recommendation ($\lambda_T = 0.85$, $\lambda_O = 0.1$, $\lambda_U = 0.05$). f_T, f_O and f_U are the log maximum likelihood estimates of query terms in document D, computed over the target collection with a Dirichlet smoothing ($\mu = 2500$).

From the ranked list of Wikipedia articles, we only consider the 5 top articles as relevant. The underlying assumption is that a Tweet may discuss only a very limited amount of topics, due to the 140 characters limit. Since encyclopedic topics are very well delimited between Wikipedia articles, we thought 5 articles would treat roughly 4-5 to 10 different topics.

3 Sentence scoring

After selecting the 5 best ranked Wikipedia articles with respect to a Tweet $\mathcal{T}$, the next step is sentence segmentation. Each article is split into sentences which are the context candidates. We describe in this section the various scoring methods we used to estimate their importance with respect to the Tweet context.

3.1 Automatic summarization

First, we used some NLP scores that are widely used in the field of automatic summarization. For each candidate sentence S we computed:

- the word overlap between S and $Q_\mathcal{T}$, and between S and $H_\mathcal{T}$,
- the cosine similarity between S and $Q_\mathcal{T}$, and between S and $H_\mathcal{T}$,
- the TextRank [4] score of S in the context of the article from which it belongs.

3.2 Conceptual similarity

We the conceptual similarity measure, we wanted to estimate at which point a candidate sentence is close to a thematic or a topic that may be related to the Tweet. We used two sources from which we extracted the concepts: Wikipedia and the Web.

The Wikipedia source is a dump from July 2011 of the online encyclopedia that contains 3,214,014 documents[1]. For the Web source, we removed the spammed documents from the category B of the ClueWeb09 according to a standard list of spams for this collection[2]. We followed authors recommendations [2] and set the "spamminess" threshold parameter to 70. The resulting corpus is composed of 29,038,220 web pages.

We model the concepts using Latent Dirichlet Allocation [1], a generative probabilistic topic model. We want to model topics that are highly related to the Tweet, hence we perform LDA on the top-ranked Wikipedia or Web documents originally retrieved using the scoring function defined in 2.2. The documents of the collection are modeled as mixtures over K topics each of which

[1] `http://dumps.wikimedia.org/enwiki/20110722/`
[2] `http://plg.uwaterloo.ca/~gvcormac/clueweb09spam/`

is a multinomial distribution over the vocabulary 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. $P(w|k)$).

In our sense, a concept is a topic generated by LDA from these top-ranked and supposedly highly relevant documents. Given a sentence S, a Tweet $\mathcal{T}$ and the learned topics $K_{\mathcal{T}}$, the conceptual score of S is given by:

$$\sigma(S) = \frac{1}{|K_{\mathcal{T}}|} \sum_{k \in K_{\mathcal{T}}} \left(\sum_{d} P(k|d) P(d|\mathcal{T}) \sum_{w \in W} p(w|k) \log \frac{N}{df_w} \right)$$

where N is the total number of documents in the collection, and df_w is the document frequency of w.

3.3 Tweeted URLs as context

A large part of the Tweets of the collection come along with an URL. This URL is the most important piece of context available, however the organizers judged to label as "manual" all the runs that used this information. We were not aware of this limitation and computed measures that are similar to the automatic summarization ones.

When a URL is present in the Tweet, we download the page and extract its title as well as the content of the body. For each candidate sentence S we computed:

- the word overlap between S and the title of the web page, and between S and the body content of the web page,
- the cosine similarity between S and the title of the web page, and between S and the body content of the web page.

3.4 Forming context

Our three runs follow the three types of measures we described above. After every sentence have been attributed a score, they are ordered and the top-ranked sentences are selected to form context (within the limit of 500 words).

4 Conclusions

In this paper we presented our contributions to the INEX 2012 Tweet Contextualization Track. We used various techniques involving automatic summarization and topic modeling algorithms to score the candidate sentences.

References

1. David M. Blei, Andrew Y. Ng, and Michael I. Jordan. Latent dirichlet allocation. *J. Mach. Learn. Res.*, 3:993–1022, March 2003.
2. Gordon Cormack, Mark Smucker, and Charles Clarke. Efficient and effective spam filtering and re-ranking for large web datasets. *Information Retrieval*, 2011.
3. Donald Metzler and W. Bruce Croft. 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, pages 472–479, New York, NY, USA, 2005. ACM.
4. Rada Mihalcea and Paul Tarau. Textrank: Bringing order into text. In *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing*, EMNLP '04, pages 404–411, 2004.
5. Toby Segaran and Jeff Hammerbacher. *Beautiful Data: The Stories Behind Elegant Data Solutions.* O'Reilly Media, 2009.