Effective Tweet Contextualization with Hashtags Performance Prediction and Multi-Document Summarization

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 2013 Tweet Contextualization track and present our contributions. Our approach is the same as last year, and is composed of three main components: preprocessing, Wikipedia articles retrieval and multi-document summarization. We however took advantage of a larger use of hashtags in the topics and used them to enhance the retrieval of relevant Wikipedia articles. We also took advantage of the training examples from last year which allowed us to learn the weights of each sentence selection feature. Two of our submitted runs achieved the two best informativeness results, while our generated contexts where almost as readable as those of the most readable system.

1 Introduction

Tweets are short and ambiguous by nature and it can be hard for a user without any background knowledge to understand what the Tweet is about. The INEX Tweet Contextualization track makes the assumption that it is possible to overcome this lack of knowledge by providing the user with a bunch of sentences that give some context or additional information about the Tweet. While topical information may certainly be the most important for this task, one may also want some political context to understand a sarcastic Tweet for example. Our approach specifically focuses on the topical context, and aims at producing informative contexts.

Within the framework of this track, sentences must be extracted from the version of Wikipedia provided by the organizers. Our approach sequentially involves Information Retrieval (IR) and Text Summarization (TS) techniques. First, we extend the Tweet's topical context by retrieving related Wikipedia articles that are likely to contain contextually relevant sentences or passages. Then, we tackle the context generation step as a summarization task where we summarize the retrieved Wikipedia articles. The sentences achieving the best linear combination of weighted features are added to the context (in the 500 words limit established by the organizers). So far, this approach is the same as the one we experimented

last year [2,3,7], we however added a hashtag performance prediction component to the Wikipedia retrieval step.

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-level features that we used.

2 Candidate Sentence Extraction

Considering that the task is to provide context from Wikipedia text, one crucial step is to retrieve Wikipedia articles that are relevant to the Tweet. Hopefully, these articles contain sentences that provide enough contextual information to (fully) understand the meaning of the Tweet.

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 [8] 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 [5] 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 a Wikipedia article D according to the following function:

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

where α is the parameter which controls the influence of the hashtags with respect to the entire Tweet text. We describe in the following section how we

set this parameter. We used the Sequential Dependence 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 ($\mu = 2500$).

2.3 Hashtags performance prediction

The importance of hashtags is also contextual. Since they can sometimes be noise rather than useful pieces of information, we need an automatic way of setting a varying α for each Tweet. We thus rely on a well-known pre-retrieval query performance predictor: the clarity score [1]. This score being actually the Kullback-Leibler divergence between the hashtags language model and the background Wikipedia collection language model, it is formally defined as:

$$\alpha = \sum_{w \in V} P(w|H_{\mathcal{T}}) \frac{P(w|H_{\mathcal{T}})}{P(w|\mathcal{C})}$$

where V is the vocabulary. The hashtags language model is estimated through pseudo-relevant feedback:

$$P(w|H_{\mathcal{T}}) = \sum_{D \in R} P(w|D) P(D|H_{\mathcal{T}})$$

The set R of pseudo-relevant documents is composed of the top 5 ranked Wikipedia articles for a the $H_{\mathcal{T}}$ query. Then α achieves higher values when documents of R are homogeneous and different from the background documents of the collection. This parameter thus allows us to predict if hashtags are discriminative, and to weigh their importance in the query accordingly.

3 Sentence scoring

From the ranked list of Wikipedia articles, we only consider the top 5 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 think that 5 articles

is a reasonable number allowing us to avoid topic drift while hopefully providing a comprehensive coverage of the Tweet's topics. After selecting the 5 best ranked Wikipedia articles with respect to a Tweet $\mathcal{T}$, the next step is sentence segmentation. Each article is divided into sentences using the `nltk`[1] toolkit. We describe in this section the various scoring methods we use to estimate their importance with respect to the Tweet's context.

3.1 Sentence features

We computed several features for each candidate sentence in order to further rank them and produce the Tweet's context. There are four categories of features:

- centrality of the sentence within the Wikipedia article from which the sentence is extracted,
- relevance of the sentence with respect to the Tweet (also including hashtags),
- relevance of the sentence with respect to an URL embedded in the Tweet,
- relevance of the Wikipedia article from which the sentence is extracted.

All the computed features use cleansed versions of sentences and Tweets. We remove stopwords and stem remaining words using the standard Porter stemming algorithm.

Sentence centrality The importance of a sentence within the document where it appears is estimated using the TextRank [6] algorithm.

Sentence relevance regarding the Tweet We compute the word overlap and the cosine similarity between the candidate sentence and the entire Tweet, and also between the candidate sentence and the hashtags alone.

Sentence relevance regarding the URL Tweets sometimes provide link to external web pages which generally contain a lot of contextual information. Organizers consider these web pages as the "answer" of the question asked by the Tweet. This is why using these web pages (even automatically) is considered as a manual run in the Tweet Contextualization track. Considering it worked very well for us last year, we still computed some features using the text of these web pages. More specifically, we compute the word overlap and the cosine similarity between the candidate sentence and the entire text of the linked page, as well as with the title of the web page.

Wikipedia article relevance The articles from which candidate sentences are extracted contain different contextual information and thus have different importance. Then, a sentence belonging to a high ranked document has a higher chance of being relevant. We use as feature the probability of the document from which the candidate sentence has been extracted.

[1] `http://nltk.org/`

Final score of a candidate sentence We compute the final importance score of each sentence as a weighted linear combination of the above features. The weights were learned using the 2012 data and are presented in Table 1.

Feature	Name	Value	Significance
$c1$	**TextRank**	8.996	$p < 2^{-16}$
$c2$	**Overlap Tweet**	2.496	$p = 2.38^{-6}$
$c3$	**Cosine Tweet**	5.849	$p = 4^{-15}$
$c4$	Overlap *hashtags*	−2.051	$p = 0.1368$
$c5$	Cosine *hashtags*	0.671	$p = 0.3074$
$c6$	Overlap title URL	1.373	$p = 0.2719$
$c7$	Cosine title URL	0.788	$p = 0.6287$
$c8$	Overlap text URL	0.543	$p = 0.4337$
$c9$	**Cosine text URL**	10.374	$p = 0.0195$
$c10$	**Document score**	0.782	$p < 2^{-16}$

Table 1. Feature weights used for our 2013 runs, learned on the 2012 available data.

After every sentence has been attributed a score, they are ordered and the top-ranked sentences are selected to form context (within the limit of 500 words). If two sentences are extracted from the same document, we keep their original order to improve readability and coherence.

4 Runs

We submitted three different runs this year, which we describe in this section.

LIA-title-only-notrain This first run only uses features $c2$, $c3$ and $c10$ without using the trained weights. Sentences of the context are thus ordered using their linear combination of three features.

LIA-all-notrain For this run we use all features described in the previous section without using the trained weights.

LIA-all-train Finally, this runs uses all features combined with the weights from Table 1.

5 Official Results

We report in Table 2 the official results released by the organizers of the 10 best performing systems. The evaluation measure computes divergences [7], hence lower scores are better. We see that our approach performed very well and

Run	All.skip	All.bi	All.uni
*LIA-all-notrain**	**0.8861**	**0.881**	**0.782**
LIA-title-only-notrain	0.8943	0.8908	0.7939
275	0.8969	0.8924	0.8061
273	0.8973	0.8921	0.8004
274	0.8974	0.8922	0.8009
*LIA-all-train**	0.8998	0.8969	0.7987
254	0.9242	0.9229	0.8331
276	0.9301	0.927	0.8169
270	0.9397	0.9365	0.8481
267	0.9468	0.9444	0.8838

Table 2. Official informativeness results of the 2013 Tweet Contextualization track (top 10 best performing systems). Starred runs are manual runs.

achieved the best results of the track. Although our best performing run was tagged as manual, we did not manually intervene at any time in our contextualization process.

In Table 3 are reported the readability results of the top 10 best systems. Although our best informative run does not achieve the best readability results, we see that it is very close to the run 275. It also produces the less redundant contexts overall. We however do not clearly understand why our three runs achieve such different readability results since the context generation process is the same. We can for example hypothesize from Table 2 that the contexts which *LIA-title-only-notrain* outputs are very similar to those of *LIA-all-notrain*. Then what could explain such a huge readability difference between the two (very similar) approaches? We think that these problems are worth further investigation.

Run	Mean Average	Relevancy	Non redundancy	Soundness	Syntax
275	**72.44%**	**76.64%**	67.30%	**74.52%**	**75.50%**
LIA-all-notrain	72.13%	74.24%	**71.98%**	70.78%	73.62%
274	71.71%	74.66%	68.84%	71.78%	74.50%
273	71.35%	75.52%	67.88%	71.20%	74.96%
LIA-all-train	69.54%	72.18%	65.48%	70.96%	72.18%
254	67.46%	73.30%	61.52%	68.94%	71.92%
LIA-title-only-notrain	65.97%	68.36%	64.52%	66.04%	67.34%
276	49.72%	52.08%	45.84%	51.24%	52.08%
267	46.72%	50.54%	40.90%	49.56%	49.70%
270	44.17%	46.84%	41.20%	45.30%	46.00%

Table 3. Official readability results of the 2013 Tweet Contextualization track (top 10 best performing systems).

6 Conclusions

We presented in this paper our contributions to the INEX 2013 Tweet Contextualization Track as well as the official results released by the organizers. We saw that a simple contextualization system composed of an effective Wikipedia retrieval system and a multi-document summarizer achieved the best informativeness results of the track. While it did not achieve the best readability results, it was very close to the best system.

References

1. Steve Cronen-Townsend and W. Bruce Croft. Quantifying Query Ambiguity. In *Proceedings of the Second International Conference on Human Language Technology Research*, HLT '02, pages 104–109, San Francisco, CA, USA, 2002. Morgan Kaufmann Publishers Inc.
2. Romain Deveaud and Florian Boudin. LIA/LINA at the INEX 2012 Tweet Contextualization track. In Forner et al. [4].
3. Romain Deveaud and Florian Boudin. Contextualisation automatique de Tweets à partir de Wikipédia. In *Proceedings of the 10th French Information Retrieval Conference*, CORIA'13, pages 125–140, 2013.
4. Pamela Forner, Jussi Karlgren, and Christa Womser-Hacker, editors. *CLEF 2012 Evaluation Labs and Workshop, Online Working Notes, Rome, Italy, September 17-20, 2012*, 2012.
5. 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.
6. 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.
7. Eric SanJuan, Véronique Moriceau, Xavier Tannier, Patrice Bellot, and Josiane Mothe. Overview of the inex 2012 tweet contextualization track. In Forner et al. [4].
8. Toby Segaran and Jeff Hammerbacher. *Beautiful Data: The Stories Behind Elegant Data Solutions*. O'Reilly Media, 2009.