Query-dependent Contextualization of Streaming Data

Nikos Voskarides, Daan Odijk, Manos Tsagkias,
Wouter Weerkamp, and Maarten de Rijke

ISLA, University of Amsterdam, Amsterdam, The Netherlands
{n.voskarides, d.odijk, m.tsagkias, w.weerkamp, derijke}@uva.nl

Abstract. We propose a method for linking entities in a stream of short textual documents that takes into account context both inside a document and inside the history of documents seen so far. Our method uses a generic optimization framework for combining several entity ranking functions, and we introduce a global control function to control optimization. Our results demonstrate the effectiveness of combining entity ranking functions that take into account context, which is further boosted by 6% when we use an informed global control function.

1 Introduction and related work

We keep track of what is happening around us through multiple dynamic sources. To keep on top of the resulting stream of information, we need useful signals. While human readers may be able to map different references as referring to the same entity, automated approaches aimed at analyzing a stream of messages might not. A key step in facilitating our understanding of streams is to move from raw text to entities. This can be achieved via a process known as *entity linking* [1, 2] that maps parts of text to entities in a knowledge base, thereby contextualizing the textual stream.

Entity linking is often performed in two steps: deciding what text to link and then where it should link to. It is common to consider entities from a general-purpose knowledge base such as Wikipedia or Freebase, since they provide sufficient coverage for most tasks and applications. For deciding what text to link *from*, the state-of-the-art performs lexical matching on the "surface forms" of entities [1, 3]. These surface forms are derived from the links created on the knowledge base, e.g., both "Barack Hussein Obama II" and "President Obama" can be used as anchor text referring to the entity "Barack Obama." Lexical matching of anchor texts can produce multiple candidate target entities that share the same surface forms (e.g., "the American President" can refer to any American president, or even the office). Deciding where to link *to* has been modeled as a learning to (re)rank problem [1, 4], where generic features represent the target entities. An alternative approach is to view this as an optimization problem [7], where the objective is to pick the best target entity for each link candidate.

When working with longer textual documents, the performance of entity linking can be enhanced by incorporating a sense of context or coherence [3, 5]. For shorter texts, context is often ignored, but even very limited textual contexts have shown to be helpful [1]. For *streaming* textual data, modeling context is not straightforward. Recent work on linking of streaming text has proposed to model context as a graph and to include it in the set of features in a learning to rerank approach [4].

To contextualize streams of related pieces of short text, one needs to resolve link candidate targets in the local short text context, while benefiting from the more global history of the stream. We propose a query-dependent contextualization approach tailored for streaming text. We use a generic optimization framework that can combine information about the surface forms of candidate entities, contextual features for short pieces of text and graph-based context modeling of the stream. Our approach combines the optimization approach that was shown effective for linking entities in short text with graph-based context modeling, shown to be effective on streaming text [4, 7].

We concentrate on a filtering task where a user is monitoring a document stream, e.g., the Twitter stream, for a specific information need (e.g., query, hashtag, trending topic) and is interested in entities that are relevant to their information need. As more data enters the stream, our system should be able to become better at "understanding" the topic of the stream and, therefore, linking new tweets to the correct entities.

Our main contribution is twofold: (i) an entity linking method that extracts and links entities using context both inside a document and in the history so far, and (ii) a manually annotated dataset [1]. The main research question we aim at answering is: (RQ1) what is the entity linking effectiveness of combining entity ranking functions that take into account context both from within a document and from the history of documents seen so far compared to individual entity ranking functions and their combinations? (RQ2) what is the effect in performance when we introduce a global control function in the optimization algorithm?

2 Method

Our entity linking method starts from the text of a tweet t and uses lexical matching to extract all n-grams that are also anchors in Wikipedia articles. As each anchor a can refer to multiple Wikipedia articles (concepts), which we call candidate concepts $Cand(a)$, the challenge is to find the correct concept $c \in Cand(a)$. The set of correct concepts that correspond to anchors in t constitute the final set of entities C for t. The main problem we are trying to solve, for a given tweet t, is twofold: (i) find an optimal set of concepts C, and (ii) generate a ranking of the concepts in C. We follow [6] for generating candidate sets of C and finding an optimal set. This algorithm starts from a set of anchors and generates a reasonable set of entities (e.g., based on their popularity). It keeps a mapping M between anchors and entities, and computes a score for this set. Then it iteratively tries to maximize the set score by replacing the entities in the set in an informed way, which guarantees to arrive to a local maximum. The optimal C is obtained by $\arg\max_C \sum_{c \in C} S(c; \cdot)$, where $S(c; \cdot)$ is the score of a concept c in C, and is defined as a linear interpolation over a set of ranking functions F:

$$S(c; \cdot) = g(\cdot) \sum_{f_i \in F} w_i f_i(\cdot), \tag{1}$$

[1] The dataset consists of 30 topics, with a total of 69,789 tweets and an average of 2,326 tweets per topic. On average, 2.8 concepts were judged as highly relevant and 44.86 as relevant per topic. See: http://ilps.science.uva.nl/resources/query-dependent-contextualization.

where g is a global control function, f_i is a ranking function (explained below) and w_i denotes the weight for f_i. Our next step is to instantiate g, and F. We build upon previous research and consider four readily used, state-of-the-art ranking functions: (i) commonness (CMNS), (ii) tweet coherence (TC), (iii) concept graph score (CGS), (iv) link probability (LP). In particular, we define $F = \{CMNS, TC, CGS\}$ and g can be either LP or 1. Below, we describe each ranking function in turn.

Commonness (CMNS). A simple function used for tweets is to rank a candidate concept $c \in Cand(a)$ by the prior probability of c being the target for the anchor a. Then, we select the best scoring candidate concept as the concept for that anchor. CMNS is defined as follows:

$$CMNS(c, a) = \frac{|L_{a,c}|}{\sum_{c' \in Cand(a)} |L_{a,c'}|},$$

where $L_{a,c}$ is the set of links with anchor text a that map to concept c.

Tweet coherence (TC). Commoness favors popular concepts. However, this is not always desirable. Think of a tweet that mentions "python" in a programming and biology context; CMNS will return the same concept regardless. One way to tackle this problem is to leverage contextual information from within the document to select candidate concepts that are topically coherent. For example, in a tweet containing the anchors "premier league" and "spurs," it is reasonably safe to assume that the anchor "spurs" should be linked to "Tottenham Hotspur F.C." For a concept $c \in C$ we compute TC as follows:

$$TC(c, C) = \frac{1}{|C| - 1} \sum_{c' \in C, c' \neq c} sim(c, c'),$$

where $sim(c, c')$ is computed using the variant of the Normalized Google Distance proposed in [3].

Concept graph score (CGS). Another way to leverage contextual information is to consider all candidate entities found in the history of documents seen so far. We create a graph $G(V, E)$ with vertices V, and edges E where V denotes candidate concepts and an edge between two vertices, c and c', exists if it holds that $sim(c, c') > 0$ and they are not linked from the same anchor. We update the graph for every incoming tweet, and rank the nodes by the standardized degree of the node (concept). We hypothesize that concepts with a higher degree are more likely to be relevant; see also [4].

Link probability (LP). Finally, for controlling our optimization algorithm, we use the link probability function [1] for g because it can penalize concepts that are refered by n-grams that are less likely to be observed as anchors in Wikipedia. For a candidate concept, we retrieve the respective n-gram via the mapping M (see above), and compute the link probability LP for the n-gram as the number of times that an n-gram is observed as anchor over its collection frequency.

We generate a final ranking of c in C for t using (1).

3 Experimental setup

To answer our research questions, we consider a stream of tweets related to a trending topic in Twitter. We contrast the performance of our method when it uses all possible functions with the performance of the method when it uses individual functions and combinations of them, for two global control functions: $g = 1$, and $g = LP$.

Dataset. We have crawled 700 Twitter trending topics between Sept. 3–Oct. 20, 2013. For each trending topic, we use the Twitter Search API to retrieve tweets that match the trending topic phrase, and were posted between 2 hours before and 3 hours after the topic started trending, ignoring retweets. We excluded "frivolous" topics such as "#IWishJacksonFollowedMe." For our evaluation exercise we sample 30 topics so that there is a good balance with regards to topic size, and to whether the topic is a hashtag or a phrase. As our knowledge base for linking tweets, we use an English Wikipedia dump dated March 18, 2013, with 4,070,588 articles. We filter out labels that are stop words or are raw numbers or have LP < 0.02, and entities with CMNS < 0.02 to exclude highly unlikely labels and entities.

Ground truth. Our ground truth is produced as follows. For every tweet in a topic, we extract all n-grams that are found as anchors in Wikipedia and retrieve their corresponding candidate concepts. We collect these mappings over each topic and filter out labels that appear only once in the topic stream. We assume each label that appeared in the topic stream maps to the same concept for all the tweets in that topic. Three annotators assessed each labels selecting the relevant concepts for each label. We use graded relevance for a concept: *highly relevant* if it is central to the topic (e.g., the WWE wrestler "Dustin Rhodes" for the topic "#Goldust"), *relevant* if it is related (e.g., "Vladimir Putin" for the topic "President Obama") and *non-relevant* otherwise, i.e. for labels unrelated to the topic's central subjects or not worthy of linking (e.g., "10", "lol").

This evaluation setup allows us to capture what people discuss about a topic in Twitter and is not based on pre-conceived ideas on what should be related to the topic at hand. For example, for the topic "#AfricasXtinction," which refers to the extinction of rhinoceros and elephants in Africa, the basketball player "Yao Ming" is judged as relevant, as he was involved in a campaign against the extinction.

We used 5-fold cross validation and chose the best weights of the objective function using line search over the parameters for each fold. We report on precision at N (P@N with N=1, 2, 3) at tweet level, both for highly relevant and for highly relevant or relevant. Statistical significance is tested using a two-tailed paired t-test and is marked as $^{\blacktriangle}$ (or $^{\blacktriangledown}$) for significant differences for $\alpha = .05$.

4 Results and analysis

In our first experiment, we set the global control function $g = 1$ and assess the effectiveness of three individual ranking functions and their combinations in the optimization framework we consider. We denote the experimental setting where we use all three ranking functions as ALL. We report our results in Table 1 (top). ALL is one of the top-performing combinations and no individual ranking function or other combination outperforms it in a stastically significant manner, for both all relevant and only high

Table 1. Performance for three individual ranking functions, i.e., commonness (CMNS), tweet coherence (TC), concept graph score (CGS) and their combinations in F, and two global control functions for g, i.e., 1 (top), and link probability (LP, bottom). ALL denotes using all three functions in F. Statistical significance is tested against ALL. Boldface marks best performance in the corresponding metric and ground truth: all relevant (relevant), only highly relevant (highly).

	P@1		P@2		P@3	
	relevant	highly	relevant	highly	relevant	highly
$g = 1$						
ALL	**0.6709**	0.4264	0.5416	0.2903	0.4481	0.2190
CMNS	0.6461	**0.4392**	0.5022$^{\blacktriangledown}$	0.2857$^{\blacktriangledown}$	0.3868$^{\blacktriangledown}$	0.2110$^{\blacktriangledown}$
TC	0.6209	0.3311$^{\blacktriangledown}$	0.4771$^{\blacktriangledown}$	0.2430$^{\blacktriangledown}$	0.4012$^{\blacktriangledown}$	0.2046$^{\blacktriangledown}$
CGS	0.6485	0.3268$^{\blacktriangledown}$	0.5368	0.2857	0.4475	**0.2404**
CMNS+TC	0.6624	0.4213	**0.5476**	**0.2974**	0.4501	0.2324
CMNS+CGS	0.6624	0.4213	0.5371	**0.2974**	0.4435	0.2244
TC + CGS	0.6592	0.3002$^{\blacktriangledown}$	0.5459	0.2744$^{\blacktriangledown}$	**0.4593**	0.2397
$g = LP$						
ALL	0.7573	0.4995	0.6044	0.3510	0.4807	0.2695
CMNS	0.7206$^{\blacktriangledown}$	**0.5111**	0.5428$^{\blacktriangledown}$	0.3265	0.4252$^{\blacktriangledown}$	0.2340$^{\blacktriangledown}$
TC	0.6801$^{\blacktriangledown}$	0.4611$^{\blacktriangledown}$	0.5399$^{\blacktriangledown}$	0.3192$^{\blacktriangledown}$	0.4420$^{\blacktriangledown}$	0.2397$^{\blacktriangledown}$
CGS	**0.7615**$^{\blacktriangle}$	0.5032	**0.6104**	**0.3593**	**0.4855**	**0.2719**
CMNS+TC	0.7289$^{\blacktriangledown}$	0.5096	0.5640	0.3373	0.4538$^{\blacktriangledown}$	0.2465$^{\blacktriangledown}$
CMNS+CGS	0.7273$^{\blacktriangledown}$	0.4980	**0.6104**	**0.3593**	**0.4855**	**0.2719**
TC + CGS	0.7557	0.4900	0.6045	0.3491	0.4817	0.2709

relevant entities. This supports our hypothesis that combining context from within the document and from the history of documents seen so far helps linking effectiveness.

There are cases where combinations marginally outperform ALL but not in a statistically significant manner. One reason for this may be due to the large intervals we used in the line search over the parameters in the training phase. Another reason can be that we ignore the decrease of an entity's CGS score over time as more nodes are added to the graph. Therefore, we are not able to achieve significant gains over the other combinations (e.g., CMNS+CGS). We plan to address this problem in future work.

In our second experiment, we set the global control function $g = LP$ and report on results in Table 1 (bottom). On average, the performance of all systems is statistically significantly better when using $g = LP$ compared to $g = 1$, with an average improvement of 6% over both highly relevant and all relevant entities. This confirms our intuition that n-grams with high LP need to be boosted during optimization, and stresses the importance of using LP in controlling the optimization function.

Our findings are similar to what we found above: no individual ranking function or combination of ranking functions significantly outperforms ALL, except for CGS at P@1. It is interesting that by controlling CGS with LP we are able to balance the problem with the CGS scores changing over time. This finding confirms the importance of modeling the stream history as a graph in the setting of Twitter topics. Future work should consider better optimization methods for combining ranking functions.

To better understand our results, we analyze the linked entities per topic. We find that our methods tend to have lower effectiveness for trending topics that are not hash-

tags. We looked closer at such topics and found that they are noisy, and therefore may not be centered around a particular entity, or set of entities. With regards to errors due to linking and not to noise in the stream, we found that in topic "#Goldust," for example, although the main entity is the wrestler "Dustin Rhodes," the objective function using ALL gives a higher score to the entity "WWE Raw" whereas CMNS always favors "Dustin Rhodes" because the high commonness score between the anchor "#Goldust" and the entity "Dustin Rhodes." A possible way to overcome this problem is to use a different function for optimization and ranking and take into account the number of times we have seen each entity in the stream history.

5 Conclusions

We presented an entity linking method for linking short documents in a stream of documents that uses context both from inside a document and the history of documents seen so far. Our experiments demonstrated that using context from previously seen documents leads to robust improvements in entity linking effectiveness. In terms of optimization, we introduced a global control function for a candidate entity, which improved early precision by 6% across all our experimental settings. The outcomes of our work can be used in applications which need to extract the most popular relevant entities in the stream so far. In future work, we envisage exploring other global control and ranking functions, and optimizing our method for efficiency.

Acknowledgments. This research was supported by the European Community's Seventh Framework Programme (FP7/2007-2013) under grant agreement nr 288024 (LiMoSINe project), the Netherlands Organisation for Scientific Research (NWO) under project nrs 640.004.802, 727.-011.005, 612.001.116, HOR-11-10, the Center for Creation, Content and Technology (CCCT), the QuaMerdes project funded by the CLARIN-nl program, the TROVe project funded by the CLARIAH program, the Dutch national program COMMIT, the ESF Research Network Program ELIAS, the Elite Network Shifts project funded by the Royal Dutch Academy of Sciences (KNAW), the Netherlands eScience Center under project number 027.012.105 and the Yahoo! Faculty Research and Engagement Program.

Bibliography

[1] E. Meij, W. Weerkamp, and M. de Rijke. Adding semantics to microblog posts. In *WSDM '12*. ACM, 2012.

[2] R. Mihalcea and A. Csomai. Wikify!: Linking documents to encyclopedic knowledge. In *CIKM '07*, pages 233–242. ACM, 2007.

[3] D. Milne and I. H. Witten. Learning to link with Wikipedia. In *CIKM '08*, pages 509–518. ACM, 2008.

[4] D. Odijk, E. Meij, and M. de Rijke. Feeding the second screen: Semantic linking based on subtitles. In *OAIR '13*, 2013.

[5] L. Ratinov, D. Roth, D. Downey, and M. Anderson. Local and global algorithms for disambiguation to Wikipedia. In *HLT '11*, pages 1375–1384. ACL, 2011.

[6] W. Shen, J. Wang, P. Luo, and M. Wang. Liege:: link entities in web lists with knowledge base. In *KDD '12*, pages 1424–1432. ACM, 2012.

[7] W. Shen, J. Wang, P. Luo, and M. Wang. Linking named entities in tweets with knowledge base via user interest modeling. In *KDD '13*, pages 68–76. ACM, 2013.