Combining Multiple Signals for Semanticizing Tweets: University of Amsterdam at #Microposts2015

Cristina Gârbacea, Daan Odijk, David Graus, Isaac Sijaranamual, Maarten de Rijke

University of Amsterdam, Science Park 904, Amsterdam, The Netherlands
{G.C.Garbacea, D.Odijk, D.P.Graus, I.B.Sijaranamual, deRijke}@uva.nl

ABSTRACT

In this paper we present an approach for extracting and linking entities from short and noisy microblog posts. We describe a diverse set of approaches based on the Semanticizer, an open-source entity linking framework developed at the University of Amsterdam, adapted to the task of the #Microposts2015 challenge. We consider alternatives for dealing with ambiguity that can help in the named entity extraction and linking processes. We retrieve entity candidates from multiple sources and process them in a four-step pipeline. Results show that we correctly manage to identify entity mentions (our best run attains an F1 score of 0.809 in terms of the strong mention match metric), but subsequent steps prove to be more challenging for our approach.

Keywords

Named entity extraction; Named entity linking; Social media

1. INTRODUCTION

This paper describes our participation in the named entity extraction and linking challenge at #Microposts2015. Information extraction from microblog posts is an emerging research area which presents a series of problems for the natural language processing community due to the shortness, informality and noisy lexical nature of the content. Extracting entities from tweets is a complex process typically performed in a sequential fashion. As a first step, *named entity recognition (NER)* aims to detect mentions that refer to entities, e.g., names of people, locations, organizations or products (also known as *entity detection*), and subsequently to classify the mentions into predefined categories (*entity typing*). After NER, *named entity linking (NEL)* is performed: linking the identified mentions to entries in a knowledge base (KB). Due to its richness in semantic content and coverage, Wikipedia is a commonly used KB for linking mentions to entities, or deciding when a mention refers to an entity that is not in the KB, in which case it is referenced by a NIL identifier. DBpedia aims to extract structured information from Wikipedia, and combines this information into a huge, cross-domain knowledge graph which provides explicit structure between concepts and the relations among them.

Our participation in this challenge revolves around the existing open-source entity linking software developed at the University of Amsterdam. We use *Semanticizer*[1], a state-of-the-art entity linking framework. So far Semanticizer has been successfully employed in linking entities in search engine queries [1] and in linking entities in short documents in streaming scenarios [6]. Moreover, it has been further extended to deal with additional types of data like television subtitles [3]. In what follows we explain how we use Semanticizer for the task at hand, and describe each of our submitted runs to the competition.

2. SYSTEM ARCHITECTURE

Our system processes each incoming tweet in four stages: mention detection, entity disambiguation and typing, NIL identification and clustering, and overlap resolution. We explain each stage in turn.

Mention detection: The first step aims to identify all entity mentions in the input text, and is oriented towards high recall. We take the union of the output of two mention identification methods:

Semanticizer: the state-of-the-art system performs lexical matching of entities' surface forms. These surface forms are derived from the KB, and comprise anchor texts that refer to Wikipedia pages, disambiguation and redirect pages, and page titles as described in Table 1. For this, we use two instances of Semanticizer, running on two Wikipedia dumps: one dated May 2014 (the version used to build DBpedia 3.9), and a more recent one, dated February 2015.

We perform three separate preprocessing steps on the tweet text, the results of which get sent to the Semanticizer. These steps are: *i)* the raw text, *ii)* the cleaned text (replacing @-mentions with corresponding Twitter account names, and splitting hashtags using dynamic programming), and *iii)* the normalized text (e.g., case-folding, removing diacritics).

NER: For identifying entity mentions that do not exist in Wikipedia, i.e., out of KB entities, we employ a state-of-the-art named entity recognizer, previously applied to finding mentions of emerging entities on Twitter [2]. We train five different NER models, three using the ground truth data from the Microposts challenges (2013 through 2015), one using pseudo-ground truth (generated by linking tweets as in [2]) and one trained on all data.

Given the candidate mentions identified by NER and Semanticizer, we include a binary feature to express whether the mention has been detected by both systems. For each mention we end up with the set of features described in Table 1 that we use in training a Random Forest classifier (using 100 trees and rebalancing the classes per tweet by modifying instance weights), to predict whether a candidate mention is an entity mention (actually refers to an entity).

Published as part of the #Microposts2015 Workshop proceedings, available online as CEUR Vol-1395 (http://ceur-ws.org/Vol-1395)

#Microposts2015, May 18th, 2015, Florence, Italy.

[1] https://github.com/semanticize/semanticizer

Table 1: Features used for mention detection.

Feature	Description
linkOccCount	no. of times mention appears as anchor text on Wikipedia
linkDocCount	no. of docs in which mention appears as anchor text
occCount	no. of times the mention appears on Wikipedia
senseOccCount	no. of times the mention is anchor to Wikipedia title
senseDocCount	no. of docs the mention is anchor to Wikipedia title
priorProbability	% of docs where anchor links to target Wikipedia title
linkProbability	% of docs where mention is anchor for a Wikipedia link
senseProbability	% of docs where mention links to target Wikipedia article
isCommon	the mention is found by both NER and Semanticizer

Entity disambiguation and typing: Given the entity mentions from the previous stage, the next step is to identify referenced entities. We retrieve the full list of candidate entities, extract features, and cast the disambiguation step of identifying the correct entity for a mention as a learning to rank problem.

Next to the features in Table 1, we use additional full-text search features. We index Wikipedia using ElasticSearch (ES), and issue the tweet as a query for candidate entities' retrieval scores. We also retrieve the 10 most similar entities for each candidate, using a *more like this* query. Finally, we incorporate Wikipedia page view statistics[2] from April 2014 as features. We use these features to train RankSVM to rank the entity candidates for each mention, and take the top ranked candidate as the entity to link. We map the entity to its DBpedia URI, and determine its type through a manual mapping of DBpedia classes to the #Microposts2015 taxonomy.

NIL identification and clustering: To decide whether the top-ranked entity is correct, or the mention refers to an out-of-KB entity, we compute meta-features based on the RankSVM classifier's scores. We use these meta-features to train a Random Forest classifier for NIL detection. We cluster NILs by linking identical mentions to a single NIL identifier based on their surface forms.

Overlap resolution: Finally, we resolve all overlapping mentions that are output by the mention identification step. We create a graph of all non-overlapping mentions, and assign them their link score (non-linked mentions get a fixed score). We then find the highest scoring path through the graph using dynamic programming, and return the mentions of this path as our resolved list of mentions.

Our submitted runs rely on this scheme and variations thereof. See Table 2 for an overview of the runs. We hypothesize that the Semanticizer will yield high entity recall, but low precision. Filtering the resulting candidates by *senseProbability* will increase precision. We expect the NER runs to be superior to Semanticizer or ES-only runs. Finally, we believe that combining the NER and Semanticizer outputs with additional candidates returned by ES will outperform all our other runs.

3. RESULTS

We evaluate our approach on the dev set consisting of 500 tweets made available by the organizers [4], [5]. In Table 3 we report on the official metrics for entity detection, tagging, clustering and linking. Our best performing runs (Run 1, Run 2) in terms of mention detection and typing rely mainly on NER and ES features. Even though Semanticizer detects candidates with high recall, our analysis indicates that most errors occur when the system fails to recognize mentions correctly, which negatively impacts the linking scores. Since each step in the pipeline relies on the output from the previous step, cascading errors influence our results, and we believe a more in-depth error analysis of each stage is desirable. Despite its simplicity, our clustering approach performs reasonably well.

[2] `https://dumps.wikimedia.org/other/pagecounts-raw/`

Table 2: Description of our runs.

RunID	NER	Semanticizer	Disambiguation	Filter
Run 1	2015	-	-	-
Run 2	2015	-	full-text search	-
Run 3	2015	-	full-text search	NIL
Run 4	all	-	full-text search	-
Run 5	-	2014	senseProbability	-
Run 6	*Same as Run 5 without overlap resolution.*			
Run 7	all	all	full-text search	NIL
Run 8	2015	all	RankSVM	NIL
Run 9	all	all	RankSVM	NIL
Run 10	*Same as Run 9 with a lower mention detection threshold.*			

Table 3: F1 scores on the dev set for strong mention match (SMM), strong typed mention match (STMM), strong link match (SLM), and mention ceaf (MC) metrics.

RunID	SMM	STMM	SLM	MC
Run 1	**0.809**	0.456	0.164	0.715
Run 2	**0.809**	**0.460**	0.330	**0.731**
Run 3	**0.809**	0.455	0.291	0.730
Run 4	0.554	0.311	0.213	0.497
Run 5	0.411	0.288	0.280	0.374
Run 6	0.620	0.389	0.280	0.567
Run 7	0.533	0.330	0.210	0.486
Run 8	0.732	0.418	**0.334**	0.633
Run 9	0.577	0.365	0.247	0.525
Run 10	0.566	0.355	0.280	0.515

4. CONCLUSION

We have presented a system that performs entity mention detection, disambiguation and clustering on short and noisy text by drawing candidates from multiple sources and combining them. We observe that our simple NER and ES runs perform better than our more complex runs. We believe that more robust methods are needed to deal with the errors introduced at each step of the pipeline. For future work we plan on improving mention detection with additional Semanticizer features.

Acknowledgements. This research was partially supported by the Netherlands Organisation for Scientific Research (NWO) under project numbers 727.011.005, SEED and 640.006.013, DADAISM; Amsterdam Data Science, and the Dutch national program COMMIT.

REFERENCES

[1] D. Graus, D. Odijk, M. Tsagkias, W. Weerkamp, and M. de Rijke. Semanticizing search engine queries: the University of Amsterdam at the ERD 2014 challenge. In *The first international workshop on Entity recognition & disambiguation*, 2014.

[2] D. Graus, M. Tsagkias, L. Buitinck, and M. de Rijke. Generating pseudo-ground truth for predicting new concepts in social streams. In *ECIR 2014*. Springer, 2014.

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

[4] G. Rizzo, A. E. Cano Basave, B. Pereira, and A. Varga. Making Sense of Microposts (#Microposts2015) Named Entity rEcognition and Linking (NEEL) Challenge. In Rowe et al. [5], pages 44–53.

[5] M. Rowe, M. Stankovic, and A.-S. Dadzie, editors. *Proceedings, 5th Workshop on Making Sense of Microposts (#Microposts2015): Big things come in small packages, Florence, Italy, 18th of May 2015*, 2015.

[6] N. Voskarides, D. Odijk, M. Tsagkias, W. Weerkamp, and M. de Rijke. Query-dependent contextualization of streaming data. In *ECIR 2014*. Springer, 2014.