UniMiB: Entity Linking in Tweets using Jaro-Winkler Distance, Popularity and Coherence

Davide Caliano
Università degli Studi di Milano-Bicocca, Italy
d.caliano@campus.unimib.it

Elisabetta Fersini
Università degli Studi di Milano-Bicocca, Italy
fersini@disco.unimib.it

Pikakshi Manchanda
Università degli Studi di Milano-Bicocca, Italy
pikakshi.manchanda@disco.unimib.it

Matteo Palmonari
Università degli Studi di Milano-Bicocca, Italy
palmonari@disco.unimib.it

Enza Messina
Università degli Studi di Milano-Bicocca, Italy
messina@disco.unimib.it

ABSTRACT

This paper summarizes the participation of UNIMIB team in the Named Entity rEcognition and Linking (NEEL) Challenge in #Microposts2016. In this paper, we propose a knowledge-base approach for identifying and linking named entities from tweets. The named entities are, further, classified using evidence provided by our entity linking algorithm and type-casted into Microposts categories.

Keywords

Knowledge base; Named entity recognition; Named entity linking

1. INTRODUCTION

Microblogging platforms such as Twitter have become a rich source of real-time information. Today, information is being readily extracted from such platforms, in the form of named entities, relations and events. The tasks of this challenge comprise identification and classification of named entities from a set of tweets, and linking the identified entities to corresponding KB resources if a match is found, or to a NIL reference if no candidate resources can be retrieved [5].

In order to identify named entities, we use a pre-trained, state-of-the-art Named Entity Recognition (NER) system [4]. Using this system, we tokenize and segment the tweets to identify entities and non-entities. Further, our linking algorithm is based on a greedy approach which disambiguates and links all the identified entities with DBpedia resources. Finally, the entities are classified using evidence from the linking phase.

2. METHODOLOGY

2.1 Named Entity Identification

For the task of identifying named entities, we use a state-of-the-art NER system, T-NER [4] which is a supervised model based on Conditional Random Fields (CRF), pre-trained on a state-of-the-art gold standard of tweets [4]. The CRF model of T-NER has been used to identify, given a tweet t as input, the candidate entities $e_1, e_2, ..., e_n$ in t. In other words, the CRF model segments a tweet into entities and non-entities.

For performing entity recognition using T-NER, we remove the special characters (@, #,..) as a pre-processing step and process the tweets in UTF-8 format in order to deal with emoticons. T-NER is not trained to recognize @usernames as entities and the current version of our system does not resolve username references. This has a significant impact on the overall performance of our system.

2.2 Candidate Resource Selection & Ranking

For the task of selecting a candidate resource for an entity, we use DBpedia[1] as our KB. We perform a pre-processing step here, wherein, for every identified entity which consists of a segment that begins with a capital letter, we segment that entity into a set of tokens based on the capital letter. For instance, the entity mention 'StarWars' is treated as 'Star Wars' during the candidate retrieval phase so as to obtain better candidate matches. To this end, we extract all the Titles of all *Wikipedia articles*[2] from DBpedia using *rdfs:label* and index them using LuceneAPI[3]. For each identified entity, top-k candidate KB resources are retrieved using a high-recall approach. Here we set $k = 500$. We estimate a knowledge-base score, called $KB(c_k)$, for each candidate resource c_k of an entity e_j as follows:

$$KB(e_j, c_k) = (\alpha \cdot lex(e_j, l_{c_k}) + (1 - \alpha) \cdot (cos_k(e_j^*, a_{c_k}))) + R(c_k) \quad (1)$$

where:

- $lex(e_j, l_{c_k})$ denotes a lexical similarity between an entity e_j and the label of a candidate resource l_{c_k};
- $cos_k(e_j^*, a_{c_k}))$ represents a discounted cosine similarity between an entity context e_j^* and a candidate KB abstract description a_{c_k};

[1]http://wiki.dbpedia.org/
[2]http://dbpedia.org/Downloads2015-04
[3]http://lucene.apache.org/

Published as part of the #Microposts2016 Workshop proceedings, available online as CEUR Vol-1691 (http://ceur-ws.org/Vol-1691)

#Microposts2016, Apr 11th, 2016, Montréal, Canada.

- $R(c_k)$ is a popularity measure of a given candidate in the KB.

More formally, $lex(e_j, l_{c_k})$ is defined as follows:

$$lex(e_j, l_{c_k}) = lcs(e_j, l_{c_k}) + W_D \left(\frac{JW(e_j, l_{c_k})}{W_D + 1} \right) \quad (2)$$

where $lcs(e_j, l_{c_k})$ denotes a normalized Lucene Conceptual Score[4] between e_j and l_{c_k}, while $W_D \left(\frac{JW(e_j, l_{c_k})}{W_D + 1} \right)$ represents a string distance measure, based on the well-known *Jaro-Winkler distance*, between an entity and the label of a candidate resource. The coefficient W_D is set equal to 3.0 and represents a boosting coefficient that allows us to weigh more syntactically close matches. The asymmetric Jaro-Winkler distance weighs more edit distances occurring in the first subsequences of two strings, and is defined as:

$$JW(e_j, l_{c_k}) = Jaro(e_j, l_{c_k}) + \frac{P'}{10} \cdot (1 - Jaro(e_j, l_{c_k})) \quad (3)$$

where *Jaro* is a similarity metric [2] and P' is a measure that takes into account the length of the longest common prefix of e_j and l_{c_k}. Moreover, in situations where a candidate label l_{c_k} is composed of more than one token, we calculate $JW(e_j, l_{c_k})$ as follows:

$$JW(e_j, l_{c_k}) = max(JW(e_j, P_1^{l_{c_k}}), ..., JW(e_j, P_n^{l_{c_k}})) \quad (4)$$

where $P_i^{l_{c_k}}$ denotes one of every possible permutation of tokens in l_{c_k}. This particular step is undertaken because users may refer to an entity in a tweet using a concise, more popular substring of the entity, which may not necessarily be the first token of the entity itself. For instance, in the tweet,

> @steph93065 shes hates me but she's no bigot, intelligent and correct most of the time. #Trump

we observe that candidate KB resources for the entity mention 'Trump' comprise of *Trump* (card game, rdf:type Thing), *Donald_Trump* (rdf:type Person), and *Trump_(comics)* (rdf:type CartoonCharacter), amongst other resources. By using the afore-mentioned equation (4), we are able to compute the JW distance for the entity mention 'Trump' not only with 'Donald Trump', which yields a low JW similarity, but also with 'Trump', which yields a high JW similarity.

To evaluate the second component $cos_k(e_j^*, a_{c_k})$ of the KB score in equation (1), we have indexed the *extended abstracts* of all DBpedia resources. This has been done with an objective to be able to disambiguate an entity with a candidate label using an entity's usage context in the tweet, on one hand, and contextual evidence from the KB on the other. The measure $cos_k(e_j^*, a_{c_k})$, which is used for denoting contextual similarity between an entity e_j and a KB candidate resource c_k, is defined as:

$$cos_k(e_j^*, a_{c_k}) = \begin{cases} cos(e_j^*, a_{c_k}) & \text{if } k = 1 \\ \dfrac{cos(e_j^*, a_{c_k})}{\log_2(k)} & k \geq 2 \end{cases} \quad (5)$$

where $cos(e_j^*, a_{c_k})$ denotes the cosine similarity between an entity context e_j^* and a candidate KB abstract description a_{c_k}. To compute equation (5), we retrieve the abstracts for all the top-k candidate resources $c_1, c_2, ..., c_k$ from DBpedia. An entity context, denoted as e_j^*, is modelled as a vector composed of an identified entity e_j in a tweet t_i and the words in the tweet which have been tagged as noun / verb / adjective. Equation (5) allows us to scale the similarity with respect to each candidate abstract according to its ranking position.

Finally, the last contribution provided in equation (1) is provided by $R(c_k)$, which allows us to take into account the popularity of a given candidate in the KB for the final ranking. To this purpose, we computed the popularity $R(c_k)$ of a KB resource c_k by using the following boosted Page Rank coefficient:

$$R(c_k) = \beta \cdot PR(c_k) \quad (6)$$

where $PR(c_k)$ is the normalized PageRank coefficient [6], and β is a damping coefficient, which lies in the range [0,1], and has been experimentally determined as equal to 0.6.

In order to determine the optimal configuration of our system, the parameters have been experimentally evaluated. The top-k candidates are ranked using equation (1) where the score of each candidate resource is denoted by $KB(c_k)$. Finally, the value of α in equation (1) has been investigated varying between the range [0,1] and the optimal value $\alpha = 0.7$ results as the best configuration.

2.3 Entity Linking and Type Classification

We followed an unsupervised, greedy approach to link an entity with a DBpedia resource. In this way, we link every identified entity with a corresponding candidate resource with the highest candidate score achieved using equation (1). However, entities for which no candidate matches are retrieved from the index have been mapped to a NIL reference with an assigned type *Thing*. The entities are, further, classified using the relation *rdf:type* with the help of *dbpedia-owl Ontology*[5]. For this purpose, we indexed the *mapping-based types* dataset of DBpedia classes[6].

Moreover, we established a mapping between the DBpedia Ontology and Microposts categories (*Thing, Person, Location, Organization, Event, Character and Product*) by following the description of the Microposts categories [5] by the challenge organizers. Every DBpedia Ontology class that can not be mapped intuitively following this description, such as the Ontology class *Species*, has been mapped to the Microposts category *Thing*. We adopted only one exception to this rule, where we mapped the DBpedia Ontology class *Name*, with its subclasses, *GivenName, Surname* to the Microposts category *Person*. GivenNames and Surnames are used in tweets mostly to refer to a person in the real world, i.e., they are mentions of entities that would be re-classified under the Microposts category *Person*. This interpretation of mapping for names and surnames is inspired by previous work on mapping semantics [1].

2.4 Entity Boundary Re-Scoping

We performed an additional post-processing step, where an identified entity's boundary is re-scoped based on the label of the resource linked to the entity in the previous phase. We apply this step when the resource label is a substring of

[4]https://lucene.apache.org/core/4_6_0/core/org/apache/lucene/search/similarities/TFIDFSimilarity.html

[5]http://mappings.dbpedia.org/server/ontology/classes/

[6]http://dbpedia.org/Downloads2015-04

Table 1: Performance: Entity Linking and Classification

Dataset		SLM	STMM	Mention_Ceaf
Training	without Re-scoping	0.327	0.297	**0.380**
	with Re-scoping	**0.336**	**0.300**	0.378
Dev	without Re-scoping	0.194	**0.139**	0.237
	with Re-scoping	**0.221**	0.134	**0.250**

Table 2: Performance: Entity Recognition

Dataset		Precision	Recall	F_1 Measure
Training	without Re-scoping	**0.627**	**0.362**	**0.459**
	with Re-scoping	0.625	0.347	0.446
Dev	without Re-scoping	0.514	0.166	0.251
	with Re-scoping	**0.545**	**0.178**	**0.268**

the entity mention. In this way, we are able to filter out noisy tokens in entities that were identified in the first step by the entity recognition system. For instance, in the tweet,

> Day 9: Wearing a StarWars T-Shirt each day until 'The Force Awakens'. We're half way there! https://t.co/QoAOxoSCJk

the entity recognition system identifies 'StarWars T-Shirt' as an entity, due to a capitalization issue, however, our linking algorithm is able to link this entity correctly with the KB resource *Star_Wars*, based on contextual and KB evidence. As a result, we re-scope the boundary of the identified entity 'StarWars T-Shirt' to 'StarWars' to improve the identification performance of the system. We evaluate our system using two configurations, viz. without entity boundary re-scoping and with entity boundary re-scoping, as reported in Section 3 below.

3. RESULTS

We use the training and dev datasets to test the performance of the pre-trained NER system (supervised approach) and, use the identified entities for testing the performance of our linking algorithm (unsupervised approach). The training and dev gold standards consist of ≈6000 and 100 tweets, annotated with a total of 8665 and 338 entities, respectively.

Table 1 shows the performance of our entity linking and classification approach for Strong Link Match (SLM), Strong Typed Mention Match (STMM) and Mention_Ceaf. As evident, the performance of the linking approach (SLM) improves when entity boundary re-scoping is applied, for both the datasets. An overall low performance of the entity linking system could be attributed to poor performance of the entity recognition system, as illustrated in Table 2. On the other hand, the performance for type classification approach (STMM) improves for the training dataset with entity re-scoping, however, the improvement is not significant.

As shown in table 2, significant precision values are obtained on both the datasets, however, recall as well as F_1 scores on the dev dataset are poor. A possible reason could be attributed to the presence of a lot of #hashtags and @usernames recognized as entities in the ground truth, which leads to a poor performance of the entity recognition system, even if @ and # are removed. An important observation is that by applying entity boundary re-scoping, precision and recall fall for the training dataset, however, its the opposite for the dev dataset. This can again be attributed to the presence of lot of #hashtags and @usernames in the dev dataset, due to which the entity recognition system exhibits entity segmentation errors.

Table 3: NER Oracle: Entity Linking Performance

Dataset	Precision	Recall	F_1 Measure
Training*	0.524	0.459	0.489
Dev*	0.452	0.387	0.417

Finally, table 3 summarizes the performance of our entity linking algorithm in terms of precision, recall and F_1 scores assuming a NER Oracle. To this end, we use a modified version of the Training and Dev gold standards, denoted as Training* and Dev* which comprise of linkable entities only, i.e., void of NIL mentions. They are annotated with 6371 and 253 linkable entities, respectively. Our linking approach is able to link correctly ≈ 50% of the entities in the modified ground truth. When a NER Oracle is used, the performance of the system obviously falls for entity boundary re-scoping. Hence, we report the results without entity boundary re-scoping for the Training* and Dev* datasets. For the test set evaluation, we provide 2 runs of our system on the test dataset for both configurations.

In previous work we defined a more sophisticated entity classification method, which combines evidence from the LabeledLDA component of T-NER and from the types of candidate entities [3]. In this challenge we could not apply this method due to problems in integrating the LabeledLDA component in our current pipeline, but we plan to use this method again in the near future.

4. REFERENCES

[1] M. Atencia, A. Borgida, J. Euzenat, C. Ghidini, and L. Serafini. A formal semantics for weighted ontology mappings. In *The Semantic Web–ISWC 2012*, pages 17–33. Springer, 2012.

[2] M. A. Jaro. Probabilistic linkage of large public health data files. *Statistics in medicine*, 14(5-7):491–498, 1995.

[3] P. Manchanda, E. Fersini, and M. Palmonari. Leveraging entity linking to enhance entity recognition in microblogs. In *Proceedings of the 7th International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management*, pages 147–155, 2015.

[4] A. Ritter, S. Clark, O. Etzioni, et al. Named entity recognition in tweets: an experimental study. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 1524–1534. Association for Computational Linguistics, 2011.

[5] G. Rizzo, M. van Erp, J. Plu, and R. Troncy. Making Sense of Microposts (#Microposts2016) Named Entity rEcognition and Linking (NEEL) Challenge. In D. Preoţiuc-Pietro, D. Radovanović, A. E. Cano-Basave, K. Weller, and A.-S. Dadzie, editors, *6th Workshop on Making Sense of Microposts (#Microposts2016)*, pages 50–59, 2016.

[6] A. Thalhammer and A. Rettinger. Browsing dbpedia entities with summaries. In *The Semantic Web: ESWC 2014 Satellite Events*, pages 511–515. Springer, 2014.