UNIBA: Exploiting a Distributional Semantic Model for Disambiguating and Linking Entities in Tweets

Pierpaolo Basile
University of Bari Aldo Moro
pierpaolo.basile@uniba.it

Annalina Caputo
University of Bari Aldo Moro
annalina.caputo@uniba.it

Giovanni Semeraro
University of Bari Aldo Moro
giovanni.semeraro@uniba.it

Fedelucio Narducci
University of Bari Aldo Moro
fedelucio.narducci@uniba.it

ABSTRACT

This paper describes the participation of the UNIBA team in the Named Entity rEcognition and Linking (NEEL) Challenge. We propose a knowledge-based algorithm able to recognize and link named entities in English tweets. The approach combines the simple Lesk algorithm with information coming from both a distributional semantic model and usage frequency of Wikipedia concepts. The algorithm performs poorly in the entity recognition, while it achieves good results in the disambiguation step.

Keywords

Named Entity Linking, Distributional Semantic Models, Lesk Algorithm

1. INTRODUCTION

In this paper we describe our participation in the Named Entity rEcognition and Linking (NEEL) Challenge [4]. The task is composed of three steps: 1) identify entities in a tweet; 2) link entities to appropriate concepts[1] in DBpedia; 3) cluster entities that belong to specific classes (entity types) defined by the organizers.

We propose two approaches that share the same methodology to disambiguate entities, while differing in the approach used to recognize entities in the tweet. We implement two algorithms for entity detection. The former ($UNIBAsup$) exploits PoS-tag information to detect a list of candidate entities, while the latter ($UNIBAunsup$) tries to find sequences of tokens (n-grams) that are titles of Wikipedia pages or surface forms which refer to Wikipedia pages.

The disambiguation and linking steps rely on a knowledge-based method that combines a Distributional Semantic Models (DSM) with the prior probability assigned to each DBpedia concept. A DSM represents words as points in a mathematical space; words represented close in this space are similar. The word space is built analyzing word co-occurrences in a large corpus. Our algorithm is able to disambiguate an entity by computing the similarity between the context and the glosses associated with all possible entity concepts. Such similarity is computed through the vector similarity in the DSM. Section 2 provides details about the adopted strategies for: 1) Entity Recognition and 2) Linking. The experimental evaluation, along with commentary about results, are presented in Section 3.

2. THE METHODOLOGY

Our methodology is a two-step algorithm consisting in an initial identification of all possible entities mentioned in a tweet followed by the linking (disambiguation) of entities through the disambiguation algorithm. DBpedia is exploited twice in order to 1) extract all the possible surface forms related to entities, and 2) retrieve glosses used in the disambiguation process. In this case we use as gloss the extended abstract assigned to each DBpedia concept.

2.1 Entity Recognition

In order to speed up the entity recognition step we build an index where each surface form (entity) is paired with the set of all its possible DBpedia concepts. The index is built by exploiting Lucene API[2], specifically for each surface form (lexeme) occurring as the title of a DBpedia concept[3], a document composed of two fields is created. The first field stores the surface form, while the second one contains the list of all possible DBpedia concepts that refer to the surface form in the first field. The entity recognition module exploits this index in order to find entities in a tweet. Given a tweet, the module performs the following steps: 1) Tokenizing and PoS-tagging the tweet via Tweet NLP[4]; 2) Building a list of candidate entity. We exploit two approaches: all n-grams up to five words ($UNIBAunsup$); all sequences of tokens tagged as proper nouns by the PoS tagger ($UNIBAsup$); 3) Querying the index and retrieving the list of the top 25 matching surface forms for each candidate entity; 4) Scoring each surface form as the linear combination of: a) the score provided by the search engine; b) a string similar-

[1] An entity can belong to several concepts.

[2] http://lucene.apache.org/

[3] We extend the list of possible surface forms using also the resource available at: http://wifo5-04.informatik.uni-mannheim.de/downloads/datasets/

[4] http://www.ark.cs.cmu.edu/TweetNLP/

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.

ity function based on the Levenshtein Distance between the candidate entity and the surface form in the index; c) the Jaccard Index in terms of common words between the candidate entity and the surface form in the index; 5) Filtering the candidate entities recognized in the previous steps: entities are removed if the score computed in the previous step is below a given threshold. In this scenario we set the threshold to 0.85. The output of the entity recognition module is a list of candidate entities in which a set of possible DBpedia concepts is assigned to each surface form in the list.

2.2 Linking

We exploit an adaptation of the distributional Lesk algorithm proposed by Basile et al. [1] for disambiguating named entities. The algorithm replaces the concept of word overlap initially introduced by Lesk [2] with the broader concept of semantic similarity computed in a distributional semantic space. Let $e_1, e_2, ...e_n$ be the sequence of entities extracted from the tweet, the algorithm disambiguates each target entity e_i by computing the semantic similarity between the glosses of concepts associated with the target entity and its context. This similarity is computed by representing in a DSM both the gloss and the context as the sum of words they are composed of; then this similarity takes into account the co-occurrence evidences previously collected through a corpus of documents. The corpus plays a key role since the richer it is the higher is the probability that each word is fully represented in all its contexts of use. We exploit the word2vec tool[5] [3] in order to build a DSM, by analyzing all the pages in the last English Wikipedia dump[6]. The correct concept for an entity is the one whose gloss maximizes the semantic similarity with the word/entity context. The algorithm consists of four steps.

1. Building the glosses. We retrieve the set $C_i = \{c_{i1}, c_{i2}, \dots, c_{ik}\}$ of DBpedia concepts associated to the entity e_i. For each concept c_{ij}, the algorithm builds the gloss representation g_{ij} by retrieving the *extended abstract* from DBpedia.
2. Building the context. The context T for the entity e_i is represented by all the words that occur in the tweet except for the surface form of the entity.
3. Building the vector representations. The context T and each gloss g_{ij} are represented as vectors (using the vector sum) in the DSM.
4. Sense ranking. The algorithm computes the cosine similarity between the vector representation of each extended gloss g_{ij} and that of the context T. Then, the cosine similarity is linearly combined with a function that takes into account the usage of the DBpedia concepts. We analyse a function that computes the probability assigned to each DBpedia concept given a candidate entity. The probability of a concept c_{ij} is computed as the number of times the entity e_i is tagged with the concept c_{ij} in Wikipedia. Zero probabilities are avoided by introducing an additive (Laplace) smoothing.

We exploit the *rdf:type* relation in DBpedia to map each DBpedia concepts to the types defined in the task. In particular, we provide a manual map for all the types defined in the *dbpedia-owl* ontology to the respective types provided by the organizers.

3. EVALUATION AND RESULTS

This section reports results of our system on the development set provided by the organizers. The dataset consists of 500 manually annotated tweets. Results are reported in Table 1. The first column shows the entity recognition strategy, the other columns report respectively the F-measure of: strong link match (SLM), strong typed mention match (STMM), mention ceaf (MC). SLM measures the linking performance, while STMM takes into account both link and type. MC measures both recognition and classification.

ER Strategy	F-SLM	F-STMM	F-MC
UNIBAsup	0.362	0.267	0.389
UNIBAunsup	0.258	0.191	0.306

Table 1: Results on the development set

We cannot discuss the quality of the overall performance since we have not information about both baseline and other participants. However, we can observe that the recognition method based on PoS-tags obtains the best performance. We performed an additional evaluation in which we removed the entity recognition module and took entities directly from the gold standard. The idea is to evaluate only the linking step. Results of this evaluation are very encouraging, we obtain a F-SLM=0.563, while excluding the NIL instances we achieve a link match of 0.825. These results prove the effectiveness of the proposed disambiguation approach based on DSM.

Acknowledgments

This work fulfils the research objectives of the PON project EFFEDIL (PON 02_00323_2938699). The computational work has been executed on the IT resources made available by two PON projects financed by the MIUR: ReCaS (PONa3_00052) and PRISMA (PON04a2_A).

4. REFERENCES

[1] P. Basile, A. Caputo, and G. Semeraro. An Enhanced Lesk Word Sense Disambiguation Algorithm through a Distributional Semantic Model. In *Proc. of COLING 2014: Technical Papers*, pages 1591–1600. ACL, August 2014.

[2] M. Lesk. Automatic Sense Disambiguation Using Machine Readable Dictionaries: How to Tell a Pine Cone from an Ice Cream Cone. In *Proc. of SIGDOC '86*, SIGDOC '86, pages 24–26. ACM, 1986.

[3] T. Mikolov, K. Chen, G. Corrado, and J. Dean. Efficient estimation of word representations in vector space. In *Proc. of ICLR Work.*, 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 M. Rowe, M. Stankovic, and A.-S. Dadzie, editors, *5th Workshop on Making Sense of Microposts (#Microposts2015)*, pages 44–53, 2015.

[5] https://code.google.com/p/word2vec/

[6] We use 400 dimensions for vectors analysing only terms that occur at least 25 times.