Learning to Search for Recognizing Named Entities in Twitter

Ioannis Partalas, Cédric Lopez, Nadia Derbas and Ruslan Kalitvianski
Viseo R&D
Grenoble, France
`firstname.lastname@viseo.com`

Abstract

This paper describes our participation in the shared task *Named Entity Recognition in Twitter* organized as part of the 2nd Workshop on Noisy User-generated Text. The shared task comprises two sub-tasks, concerning a) the detection of the boundaries of entities and b) the classification of the entities into one of 10 possible types. The proposed approach is based on Linked Open Data for extracting rich features along with standard ones which are then used by a learning to search algorithm in order to build the tagger. The submitted system scored 46.16 and 60.24 in terms of F-measure and ranked 2nd and 3rd for the classification and segmentation tasks respectively.

1 Introduction

Named-Entity Recognition (NER) is a well-studied task for over two decades now with notable success for noise-free and grammatically well-structured documents. The final goal of a NER system is to classify textual segments in a predefined set of categories, for example persons, organizations, and locations. While current NER systems achieve very high performance for a narrow set of entities, in applications like in Twitter[1] where text is short, using an informal style and with an unreliable use of capitalization, the recognition of entities becomes a challenging task (Marrero et al., 2013; Ritter et al., 2011; Derczynski et al., 2015). For example, in the 2015 *NER in Twitter* competition the best system achieved an F-score of around 56% for ten entities (Baldwin et al., 2015).

In this context, the shared task for *Named-Entity Recognition in Twitter* has been organized in order to provide standard benchmarks for this task. In this paper we describe our participation in the 2nd edition of the *Named-Entity Recognition in Twitter* shared task which was organized in the framework of the Workshop on Noisy User-generated Text.[2] We cast the problem as a sequence labeling task and used a learning to search approach for solving it. The proposed system, called Talos,[3] combines an approach based on Linked-Open Data for extracting rich contextual features along with standard ones that are usually included in NER systems.

A total of 3,814 annotated tweets were provided by the organizers while the test set contained 3,850 examples. Our system ranked second in the classification and third in the segmentation (only detecting the boundaries) sub-tasks achieving F-scores of 46.16 and 60.24 respectively. In the following sections we present a description of the method we developed for tackling the tasks and provide evaluations of its performance.

2 Approach

In this section we detail our approach for the NER task. More specifically, we provide the different features that were extracted from the textual data as well as from external resources. Later, we present

[1]For example, one would like to track certain entities as a step towards influence detection, see SOMA project http://www.somaproject.eu/

[2]http://noisy-text.github.io/2016/

[3]In Greek mythology, Talos was a giant automaton constructed by the god Hepheastus and had as mission to protect Europa in Crete. https://en.wikipedia.org/wiki/Talos

the learning algorithm we used and finally an ensemble of rules that were applied on the predictions of the learned model in order to correct inconsistencies in the tagging.

2.1 Feature Engineering

We divide the extracted features into three major categories: a) lexical and morphosyntactic features for word units in the textual data, b) features using Linked Open Data (LOD) and c) features based on distributed representations of words.

2.1.1 Lexical and Morphosyntactic Features

The word-level features that we employ are the following:

- Current token w in its original form and words around it, $w[-n : n]$ for $n = 2$.
- Suffixes and prefixes of the current token of size 3 (empirically defined).
- Boolean features regarding the capitalization of the current token:
  - The token starts with capital letter.
  - All letters in the token are capital.
  - Previous words are capitalized.
  - The token contains a capitalized letter.
- Lexical features (is digit, is punctuation etc., starts with @, is hashtag, contains capital letter).

The presence of the current token in a standard English dictionary as well as in Wordnet are used as features along with the probability of its bigram prefix in order to be able to distinguish entities with rare n-grams (e.g. for XBOX the probability of encountering XB as a character bigram is zero). In the same line, we also included the presence of each token in the gazetters provided by the organizers.

Finally, we use the Stanford taggers in order to obtain the part-of-speech (POS) tag of the current and the previous tokens as well as the 3-class named-entity tag (location, person and organization) for a window of one token around the word (Kristina Toutanova, 2003). Both POS and named-entity tags have improved the performance of our system.

2.1.2 Contextual Enrichment Features

As tweets are limited to 140 characters, the context is often not sufficient to disambiguate the mentions they contain. To overcome this constraint we propose to add context in a two fold manner: 1) using more tweets that contain the same mention, 2) using external LOD resources.

Regarding the first point, we used the Twitter API[4] and for each mention in the tweet we obtained a maximum of 100 tweets which is the limit imposed by Twitter API. For each set of tweets, we compute a score based on the presence of the immediate context (that is the two words before and after) in lexical fields defined for each class of entities. For instance, `fan`, `sing`, `dj` and `musician` are words contained in the `musicartist` lexical field. Such lexical fields were created manually from Wikipedia pages. We called this feature ***enrichTweetsImmediateContext*** *(ETIC)*. A variant of this feature is ***enrichTweets*** *(ET)*, that considers all the words in the tweet.

Concerning the second point, we generated features inspired from Entity Linking approaches. Indeed, Yamada et al. (2015) obtained the best results in the last edition of the challenge considering entity linking results as features dedicated to enhance the NER performance. For each tweet, whenever a mention is not provided, we generated all the possible n-grams of words (with $n \leq 5$). We directly use the DBpedia SPARQL endpoint in order to request the presence of each n-gram as an entry of this knowledge base, taking into account:

[4]We used the Twitter4J Java library: $http://twitter4j.org/en/$

- ***Exact Matching** (EM).* We search the type (`rdf:type`[5]) of the DBpedia ressource built according to the standard URI format (`http://dbpedia.org/resource/`+Entity). For instance, we obtain `<http://dbpedia.org/resource/Olympique_de_Marseille> rdf:type dbo:SportsTeam`.
- ***Redirection** (EM).* In order to deal with common misspelling, abbreviation and acronym cases, we also consider the `dbo:wikiPageRedirects`[6] predicate. For instance, "'L'oheme"' or "'Olympic de Marseille"' refers to the same URI namely `dbo:Olympique_de_Marseille`.
- ***Disambiguation** (D).* As most of the mentions are ambiguous, we use the `dbo:wikiPageDisambiguates`.[7] For instance, "OM" could be the Russian river `dbo:Om_River` or a television show `dbo:Om_(2003_film)`.

In these three cases, the mention receives a boolean value according to each category in the classification task (movie, geo-loc, person, etc.): 1 if a `rdf:type` corresponds to one of the 10 categories, 0 otherwise.

2.1.3 Word Clusters

We exploited unlabeled data using word representations which have been proved to enhance performance for a variety of NLP tasks (Ritter et al., 2011; Owoputi et al., 2012; Cherry et al., 2015). In this line, we used Brown clusters (Brown et al., 1992) as well as word clusters on top of GloVe word representations from external resources and a set of tweets collected in several periods between July and September 2016 resulting to a corpus of around 39 million tweets. The collection has been biased to include tweets in the context of each of the ten entity classes. For example, for the class `musicartist` we used the following keywords, {`band, musicgroup, livemusic, concert, musician, musical`}.

Regarding the external resources we used pre-trained Brown clusters from TweetNLP[8](Owoputi et al., 2012) and pre-trained GloVe word vectors of Wikipedia and Twitter collections[9] which were grouped in 1,000 clusters using K-means (Pennington et al., 2014). Several dimensions for the word vectors were tested, ranging from 50 to 200, and the best results were obtained with the 100 dimensional ones. For the collected tweets, around 36 million English tweets were selected using langpid.py (Lui and Baldwin, 2012). Then, we trained Brown clusters to partition the words in 1,000 classes with a minimum frequency of appearance in the collection of 10 times (Liang, 2005).

2.2 Learning Algorithm

The learning problem was cast as a sequence labeling one, and for solving it we used a learning to search (L2S) approach, which represents a family of algorithms for structured prediction tasks (Daumé III et al., 2014; Chang et al., 2015). These methods decompose the problem in a search space with states, actions and policies and then learn a hypothesis controlling a policy over the state-action space. In this case, each example in the training data is used as a reference policy (labels to be assigned at each token) and the learning algorithm tries to approximate it. This technique resembles reinforcement learning methods, with the difference that in the latter one is not provided with a reference policy but discovers it through trial-and-error.

The L2S framework is highly modular as it reduces structured problems to cost-sensitive multi-class classification ones. This allows us to use state-of-the-art algorithms for multi-class classification.

2.3 Post-application of Rules

The annotations produced by our system contained inconsistencies which we corrected with a post-processing step. For instance, in the BIO encoding, a label "I" can not exist if the previous label is not a "B". In order to avoid such cases, we applied the following rules:

[5]$http://www.w3.org/1999/02/22-rdf-syntax-ns\#type$
[6]$http://dbpedia.org/ontology/wikiPageRedirects$
[7]$http://dbpedia.org/ontology/wikiPageDisambiguates$
[8]http://www.cs.cmu.edu/ ark/TweetNLP/
[9]http://nlp.stanford.edu/projects/glove/

1. $O - \mathbf{I} \Rightarrow O - \mathbf{B}\ orB - \mathbf{I}\ orB - I - \mathbf{I}\ orB - I - I - \mathbf{I}$ regarding the presence of capital letters in previous words.
2. $I - \mathbf{B} \Rightarrow I - \mathbf{I}$
3. $B - \mathbf{B} \Rightarrow B - \mathbf{I}$

Concerning the last two rules, we take into account the fact that the phenomenon of composition (when an entity is included in another entity) is not frequently present in the training set.

3 Experiments

3.1 Data

The organizers provided a training set with 2,814 tweets and a development set of 1,000 tweets which corresponds to the test set of the 2015 edition of the challenge. Table 1 presents the distribution for each entity type in the training and development sets. We observe that `person`, `geo-loc` and `other` are the dominating classes in both sets.

type	train	dev	test
company	204	39	621
facility	111	38	253
person	522	171	482
musicartist	68	41	191
geo-loc	322	116	882
movie	37	15	34
other	272	132	584
product	106	37	246
sportsteam	86	70	147
tvshow	40	2	33

Table 1: Size for each class of entities in the training, development and test sets.

Tweets were already tokenized and we applied normalization in order to reduce noise induced by spelling errors or the use of abbreviations and slang. There are three kinds of token normalizations (Baldwin et al., 2015):

- one-to-one normalization, when one out-of-vocabulary (OOV) token is replaced by one in-vocabulary equivalent,
- one-to-many (OTM), when one token is replaced by several,
- many-to-one (MTO), when several tokens are replaced by one.

In this work we perform a simple one-to-one normalization, by substituting OOV terms for which a normalized equivalent exists in the W-NUT normalization shared task corpus of 2015 by their normalized counterparts (Baldwin et al., 2015). This has the advantage of not requiring to perform a token-level alignment between the unnormalized tweet and the potentially longer (or shorter) tweet that would result from an OTM or MTO normalization.

No further preprocessing has been applied to the data. Nevertheless, during an inspection of the data we observed some inconsistencies in the annotation of some entities like for example "Justin Bieber" which was annotated as `person` in the training data while in the test set as `musicartist`. During development, we decided not to alter these annotations although it could have slightly improved the performance of our system.

It is interesting to note here that the intersection of the unique entities between the `train` and `dev` sets is around 14.6% while the corresponding ratio for `train + dev` and the `test` set is around 8.3%. This means that we should expect a harder prediction task than this of 2015 which will test the generalization capabilities of our approach.

Namespace	Segmentation	Classification
LM	$w_{-2:2}$, $N_{-1:1}$, $POS_{-1:0}$, $Cap_{-1:0}$, $Lexical$	$w_{-2:2}$, $N_{-1:1}$, $POS_{-1:0}$, $Cap_{-1:0}$, $\alpha_0^{\pm 3}$, $Lexical$
LOD	LOD_{EM}, LOD_R, LOD_D	LOD_{ETIC} LOD_{ET}, LOD_R, LOD_{EM}, LOD_R, LOD_D
WordClusters	B_0, $B_0^{indomain}$, GV_0^{wiki}, GV_0^{tweets}	B_0, $B_0^{indomain}$, GV_0^{wiki}, GV_0^{tweets}

Table 2: Template features for the two tasks. w corresponds to the word, N to the 3-class NER tag, Cap to the capitalization features and α to the affix.

3.2 Setup

We trained two different systems for the corresponding tasks, that is the detection of boundaries and the classification of types of entities. Note, that we use the predictions of the system that detects the boundaries in order to extract the Linked Open Data features discussed in Section 2.1.2. While this approach can have a low recall rate, it has the advantage of being very fast and precise and thus not injecting noise in the features. On the other hand, one could have generated all the n-grams from left to right for each tweet, letting the system increase the coverage of the entities but with a high computational cost and a low precision rate. Table 2 presents the templates we used for each of the two tasks where w, N, Cap, $Lexical$ and α refer to the form of word, named-entity obtained by the Stanford tagger, lexical features and affixes of a word. For each feature, a subscript denotes the slice that has been used around the current token. For the word clusters B and $B^{indomain}$ refer to the pre-trained brown clusters and the ones learned over our collection of tweets (see Section 2.1.3). Finally, GV clusters correspond to the trained clusters for Wikipedia and tweets GloVe vectors respectively.

For both tasks, parameter tuning was performed through a random search with cross-validation for the estimation of the performance. In order to evaluate the performance of our system with respect to the results of the 2015 edition of the competition, we performed the tuning only on the training set and then test on the development set. For our final submission the same procedure was repeated on the union of the train and development sets.

We used the Vowpal Wabbit[10] suite, which implements several L2S algorithms and provides a framework for compiling structured learning tasks. We tune the following parameters for L2S: *learning_rate* $\in [0.05, 0.3]$, *passes* $\in [2, 20]$ and *history_length* $\in \{1, 2, 3\}$. The parameter for the hash function was optimized beforehand and kept fixed to 29 bits throughout the experiments. Finally, we used the default setting for solving the multi-class cost-sensitive classification problems.

3.3 Results

Tables 3 and 4 present the results for the segmentation and classification tasks respectively. As the development set for 2016 corresponds to the test set of 2015 edition of the challenge, we include the best two systems of 2015 for comparison, namely **Ousia** (Yamada et al., 2015) and **NLANGP** (Toh et al., 2015). Our system, Talos, was ranked third and second respectively for the two tasks.

Regarding the segmentation task, Talos achieved a high precision score but with a lower recall in both the development and the test sets.

Table 4 presents the results in the same fashion as previously for the entities classification task. We include a version of our system without using the LOD features and a version without normalization in order to assess their contribution. Talos achieved a high F-score in the test set and ranked second in this task. We also observe that on the development set it outperformed the best system of the 2015 edition of the challenge. However, noticeably we note the lower scores in the `test` set for the best systems with respect to the results of 2015. This could come from the fact that many mentions never appear in the training data and thus stressing the generalization capabilities of the systems. Also, as tweets come from different periods of time one should account for concept-drift phenomena, which we did not consider in our case.

[10]http://hunch.net/vw

System	dev 2016			test		
	P	R	F	P	R	F
Talos	71.29	55.98	62.71	70.53	52.58	60.24
1st-2016	-	-	-	73.49	59.72	65.89
2nd-2016	-	-	-	64.18	62.28	63.22
Ousia	72.20	69.14	70.63	-	-	-
NLANGP	67.74	54.31	60.29	-	-	-

Table 3: Results for the no-types task in terms of Precision, Recall and F-score. The development test of 2016 corresponds to the test set of 2015 for which we present the best two systems.

Concerning LOD features, we note that they improve considerably the performance of the system showing how rich contextual features can compensate for the lack of context. Finally, considering normalization, we observe an increase in both the development test set F-scores. Nevertheless, one should be careful as in some cases normalization may change semantics in the text and thus deteriorate the performance.

System	dev 2016			test		
	P	R	F	P	R	F
Talos	63.64	51.89	57.17	58.51	38.12	46.16
Talos_no_LOD	62.90	44.63	52.21	54.32	36.37	43.57
Talos_no_norm	63.55	51.44	56.86	58.19	37.83	45.86
1st-2016	-	-	-	60.77	46.07	52.41
3rd-2016	-	-	-	51.70	39.48	44.77
Ousia	57.66	55.22	56.41	-	-	-
NLANGP	63.62	43.12	51.40	-	-	-

Table 4: Results for the no-entity task. The development test of 2016 corresponds to the test set of 2015 for which we present the best two systems.

Table 5 presents the results per entity class for Talos. As expected, the system performs well for the person, company and geo-loc classes which are the most populated ones.

Type	Precision	Recall	F
company	64.51	36.88	46.93
facility	60.67	21.34	31.58
geo-loc	71.31	65.65	68.36
movie	20.00	2.94	5.13
musicartist	45.00	4.71	8.53
other	50.94	18.49	27.14
person	46.12	59.13	51.82
product	33.33	6.91	11.45
sportsteam	42.42	28.57	34.15
tvshow	0.00	0.00	0.00

Table 5: Results per entity type for Talos.

4 Conclusion

We presented in this work our participation in the “2nd Named Entity Recognition for Twitter” shared task. The task has been cast as a sequence labeling one and we employed a learning to search approach in order to tackle it. We also leveraged LOD for extracting rich contextual features for the named-entities. Our submission achieved F-scores of 46.16 and 60.24 for the classification and the segmentation tasks and ranked 2nd and 3rd respectively. The post-analysis showed that LOD features improved substantially the performance of our system as they counter-balance the lack of context in tweets.

The shared task gave us the opportunity to test the performance of NER systems in short and noisy textual data. The results of the participated systems shows that the task is far to be considered as a solved one and methods with stellar performance in normal texts need to be revised.

5 Acknowledgments

We would like to thank the organizers of the shared-task for their time and effort that they allocated in order to provide a standard benchmark. This work has been supported by the SOMA Eurostars program 9292/12/19892.

References

Timothy Baldwin, Young-Bum Kim, Marie Catherine De Marneffe, Alan Ritter, Bo Han, and Wei Xu. 2015. Shared tasks of the 2015 workshop on noisy user-generated text: Twitter lexical normalization and named entity recognition. In *Association for Computational Linguistics (ACL)*. ACL Association for Computational Linguistics, August.

Peter F. Brown, Peter V. deSouza, Robert L. Mercer, Vincent J. Della Pietra, and Jenifer C. Lai. 1992. Class-based n-gram models of natural language. *Computational Linguistics*, 18:467–479.

Kai-Wei Chang, He He, Hal Daumé III, and John Langford. 2015. Learning to search for dependencies. *CoRR*, abs/1503.05615.

Colin Cherry, Hongyu Guo, and Chengbi Dai. 2015. Nrc: Infused phrase vectors for named entity recognition in twitter. In *Proceedings of the Workshop on Noisy User-generated Text*, pages 54–60, Beijing, China, July. Association for Computational Linguistics.

Hal Daumé III, John Langford, and Stéphane Ross. 2014. Efficient programmable learning to search. *CoRR*, abs/1406.1837.

Leon Derczynski, Diana Maynard, Giuseppe Rizzo, Marieke van Erp, Genevieve Gorrell, Raphal Troncy, Johann Petrak, and Kalina Bontcheva. 2015. Analysis of named entity recognition and linking for tweets. *Information Processing & Management*, 51(2):32 – 49.

Christopher D. Manning Yoram Singer Kristina Toutanova, Dan Klein. 2003. Feature-rich part-of-speech tagging with a cyclic dependency network. In *In Proceedings of NAACL*.

Percy Liang. 2005. Semi-supervised learning for natural language. In *PhD Thesis, MIT*.

Marco Lui and Timothy Baldwin. 2012. Langid.py: An off-the-shelf language identification tool. In *Proceedings of the ACL 2012 System Demonstrations*, ACL '12, pages 25–30. Association for Computational Linguistics.

Mónica Marrero, Julián Urbano, Sonia Sánchez-Cuadrado, Jorge Morato, and Juan Miguel Gómez-Berbís. 2013. Named entity recognition: fallacies, challenges and opportunities. *Computer Standards & Interfaces*, 35(5):482–489.

Olutobi Owoputi, Chris Dyer, Kevin Gimpel, and Nathan Schneider. 2012. Part-of-speech tagging for twitter: Word clusters and other advances. Technical report, CMU.

Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. Glove: Global vectors for word representation. In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543.

Alan Ritter, Sam Clark, Mausam, and Oren Etzioni. 2011. Named entity recognition in tweets: An experimental study. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, EMNLP '11, pages 1524–1534.

Zhiqiang Toh, Bin Chen, and Jian Su. 2015. Improving twitter named entity recognition using word representations. In *Proceedings of the Workshop on Noisy User-generated Text*, pages 141–145, Beijing, China, July. Association for Computational Linguistics.

Ikuya Yamada, Hideaki Takeda, and Yoshiyasu Takefuji. 2015. Enhancing named entity recognition in twitter messages using entity linking. In *Proceedings of the Workshop on Noisy User-generated Text*, pages 136–140, Beijing, China, July. Association for Computational Linguistics.