Semantic Linking for Event-Based Classification of Tweets

Amosse Edouard, Elena Cabrio, Sara Tonelli, Nhan Le Thanh

Semantic Linking for Event-Based Classification of Tweets

Amosse Edouard[1], Elena Cabrio[1], Sara Tonelli[2], and Nhan Le Thanh[1]

[1] Université Côte d'Azur, Inria, CNRS, I3S, France
firstname.lastname@unice.fr
[2] Fondazione Bruno Kessler, Trento, Italy
satonelli@fbk.eu

Abstract. Detecting which tweets are related to events and classifying them into categories is a challenging task due to the peculiarities of Twitter language and to the lack of contextual information. We propose to face this challenge by taking advantage of the information that can be automatically acquired from external knowledge bases. In particular, we enrich and generalise the textual content of tweets by linking the Named Entities (NE) to concepts in both DBpedia and YAGO ontologies, and exploit their specific or generic types to replace NE mentions in tweets. The approach we propose in this paper is applied to build a supervised classifier to separate event-related from non event-related tweets, as well as to associate to event-related tweets the event categories defined by the Topic Detection and Tracking community (TDT). We compare Naive Bayes (NB), Support Vector Machines (SVM) and Long Short-Term Memory (LSTM) classification algorithms, showing that NE linking and replacement improves classification performance and contributes to reducing overfitting, especially with Recurrent Neural Networks (RNN).

1 Introduction

The capability to understand and analyse the stream of messages on Twitter is an effective way to monitor what people think, what trending topics are emerging, and which main events are affecting people's lives. For this reason, several automated ways to track and categorise *events* on Twitter have been proposed in the literature. Such task is not trivial, given that most of the messages on Twitter are not related to events [12]. Moreover, processing Twitter messages is challenging because tweets are at most 140 characters long, contain little contextual information and often exhibit misspelled words and jargon.

Another challenge is related to the definition of 'event' in social media streams, since the different definition efforts carried out within the Natural Language Processing (NLP) community do not seem to fully capture the peculiarities of social media content [25]. Instead, more operational definitions have been preferred. For instance, [8] defines an event in the context of social media as "*An occurrence causing change in the volume of text data that discusses the associated topic at a specific time. This occurrence is characterized by topic and time, and often associated with entities such as people and location*". This definition, which we also adopt in this paper, highlights a

strong connection between events in the context of social media and the Named Entities (NEs) involved in such events (corresponding to events' participants, typically persons, organisations and locations). Despite their importance, however, using entity mentions as features in a supervised setting to classify event-related tweets does not generalise well, and may affect classification performance across different event categories.

We investigate this issue in the present paper, and we analyse the effect of replacing entity mentions in tweets with specific or generic categories automatically extracted from external knowledge bases. Specifically, we compare the classification performance linking named entities to DBpedia [14] and YAGO [15] in order to classify tweets in different event categories defined by the TDT community.

The main contributions of this paper are as follows : i) we propose and evaluate an approach for detecting tweets related to events as well as classifying them into event categories; ii) we show that supervised models can achieve good generalisation capabilities through semantic linking; iii) we evaluate how generic and specific types of NE affect the output of supervised models.

The paper is structured as follows: Section 2 reports on relevant related literature, Section 3 describes the proposed approach to detect and classify event-related tweets. Section 4 presents the experimental settings and discusses the obtained results. Conclusions end the paper drawing further remarks and proposing future works.

2 Related Work

In recent years, several approaches to build event-related services applied to social media have been investigated. Existing approaches to event detection on Twitter have been classified into two main categories: *closed domain* and *open domain* [2]. The first ones are mainly focused on detecting specific fine-grained event types (e.g. earthquakes, influenza epidemics), while the second ones do not target a specific event type, but try to detect real-world events belonging to various categories as they happen. Works in the closed domain scenario mostly rely on keywords to extract event-related messages from Twitter [22], recognise event patterns [20] or define labels for training classifiers [1].

The open domain scenario is more challenging. Its first step is the separation between event-related and non event-related tweets [11], a task that we also tackle in the present paper. [3] apply an online clustering and filtering framework to distinguish between messages about real-life events and non-events. The framework clusters streaming tweets using their similarity with existing clusters. In this framework, a set of event features including temporal, social, topical and Twitter-centric features is defined.

In [21], events are modelled using a 4-tuple representation including NEs, temporal expressions, event phrases and event types. The system recognizes event triggers as a sequence labelling task using Conditional Random Field. In addition, the authors measure the association strength between entities and calendar dates, which is used as key feature to separate event and non event-related tweets. Nevertheless, this assumption restricts the approach to tweets that explicitly contain temporal expressions.

More recently, researches have explored the usage of external knowledge sources to enrich the content of tweets. Genc et al. [9] introduced a Wikipedia-based classification technique to construct a latent semantic model that maps tweets to their most similar

articles on Wikipedia. Similarly [24] proposed a probabilistic framework to map terms in tweets to concept in the Probase knowledge base.

Cano et al. [5] exploit information obtained from different knowledge sources for tweet classification and evaluate their approach in the violence and emergency response domains. The evaluation shows that extracting semantic features from external knowledge sources outperform state-of-the-art techniques such as bag of words, bag of entities or part of speech features.

In our work, we exploit information acquired from external knowledge bases to enrich NEs mentioned in the tweets with additional information. Then enriched content is used to extract features for building word-embedding vectors which serve as feature model for training supervised models in the aim of identifying tweets related to events as well as classifying event-related into fine-grained event categories. Furthermore, while previous approaches have been evaluated on datasets collected during a short time period [3, 5, 19], we evaluate ours on two datasets collected over two different periods and covering different event types.

3 Detecting and Classifying Event Tweets

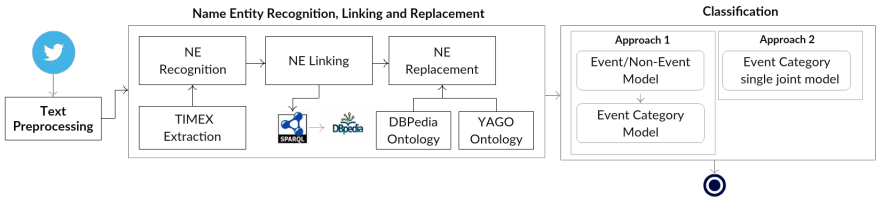

Fig. 1. The event detection pipeline (tweets are the input source). Rectangles are conceptual stages in the pipeline with data flowing in the order of the arrows. The solid circle at the bottom represents the output, i.e. tweets related to events and classified into categories.

This section describes the approach we propose to identify event-related tweets and classify them into categories. Given a set of tweets as input, the main steps of our framework include a **Preprocessing** step, to clean the input data, **Named Entity replacement**, based on NE recognition and linking, and **Tweet classification**. The goal of the last step is to classify the input tweets related to events into categories such as *Sports* and *Politics*. We propose two framework configurations: the first one carries out two steps in a row, in which the input tweets are first classified into event-related and not event-related, and the event-related ones feed the second classifier labelling them with different categories. The second configuration relies on a one-step solution, in which tweets that are not related to events are considered as an additional category together with the events categories in a multi-class classification step. Figure 1 shows the proposed framework architecture, detailed in the following sections.

3.1 NE Linking and Replacement

We first remove Twitter-specific features such as user mentions, emoticons and URLs, identified with a Twitter-specific Part-Of-Speech tagger [17]. Identical re-tweets are removed, as well as hashtags, if they cannot be mapped to a NE.

Then, we run the Entity Replacement module in our pipeline (Figure 1, Named Entity Recognition step) by calling the API of NERD-ML [26] to first recognise NEs in tweets, and then to link them to DBpedia[3] for the entity linking task. As a comparison, we also rely on the YAGO ontology [15], to assess how the two resources impact on our task. Since resources in DBpedia are also mapped to concepts in YAGO, linking tweets' NEs to this resource starting from the DBpedia categories labeled with NERD-ML is pretty straightforward. We rely on NERD-ML because it proved to outperform other NER systems on Twitter data [7].

As for the knowledge bases, we focus on YAGO and DBpedia because they are among the most widely used general-purpose knowledge bases in the Semantic Web and NLP community, each having its own peculiarities. DBpedia relies on an ontology with around 685 classes and 2,795 properties, which have been manually mapped to Wikipedia infobox types. YAGO, instead, contains approximately 350,000 classes and is based on a much larger and deeper hierarchy derived partly from Wikipedia categories (the lower levels) and WordNet (the most general layers of the hierarchy) in a semi-automated fashion. More recently, it has also been enriched with concepts in GeoNames. This reflects the different coverage of the two resources: while DBpedia covers only the Wikipedia pages with an infobox that was mapped to its ontology, YAGO includes all Wikipedia pages having at least one category, thus it has a broader coverage. Such difference emerges also in our experiments, since we found that DBpedia URIs cover approximately 56% of the NEs detected in our tweet corpus, while YAGO accounts for 62% of the entities.

The NE linking submodule (Figure 1) relies on the DBpedia URI provided by NERD-ML to retrieve the categories to which an entity belongs in the order in which they appear in the hierarchy of the considered ontology (i.e DBpedia or YAGO). For example, for the geographical entity "New York", we retrieve from DBpedia ontology the categories "$Administrative\ Region \rightarrow Region \rightarrow Populated\ Place \rightarrow Place$". We then apply NE replacement, to investigate the impact of NE generalisation on event classification. We compare two strategies that replace the NEs in tweets with *i)* the first element in the hierarchy, i.e. *the most specific category* (e.g. "Administrative Region" in the example above); *ii)* the last element, i.e. the *most generic category* (e.g. "Place") of the entity. The rationale behind this replacement is to generalise over single mentions and to make classification more robust across different domains.

Beside NEs, also temporal information is relevant to event recognition and classification. Therefore, the Entity Replacement module extracts temporal expressions in the content of the tweets with the SUTime tool [6] and replaces them with one of the TIMEX3 types assigned by the tool: Date, Time, Duration and Set. Although SUTime

[3] When using DBpedia as external KB for entity linking, our approach is not limited to proper names (i.e. persons, location or organisations) but considers any term that has an associated URI in DBpedia (e.g. Nobel Prize).

was trained on news texts, it is expected to be precise enough on tweets, given that temporal expressions are relatively unambiguous [21]. Table 1 reports two example tweets in the original format and the version after replacement.

Original Tweet	**Generic Categories**	**Specific Categories**
Cambodia's ex-King Norodom Sihanouk dead at 89 http://q.gs/2IvJk #FollowBack	[Place] ex-king [Person] die at [number]	[Country] ex-king [Royalty] die at [number]
Amy Winehouse, 27, dies at her London flat http://bit.ly/nD9dy2 #amyWinehouse	[Person], [number], die at her [Place] flat [Person]	[Person], [number], die at her [Settlement] flat [Person]

Table 1. Output of the Entity Replacement module on two tweets with DBpedia categories.

3.2 Tweet classification

As shown in Figure 1, the third module is devoted to tweet classification. Two classification steps are performed. The first one is aimed at separating tweets that are related to real-world events from tweets that do not talk about events, and we cast it as a binary supervised classification. The second step aims at classifying tweets related to events into a set of categories. We cast the problem of classifying event-related tweets into categories as a supervised multi-class classification problem. We limit the scope of this work to the eight event categories from [16] (see Table 2).

4 Experimental setting and Results

This section presents the experiments we carried out to validate the proposed framework for event detection and classification on Twitter. We first describe the datasets (Section 4.1), then we present the algorithms and the results obtained on event detection (Sect. 4.3), and on event classification (Sect. 4.4).

4.1 Datasets

We use two different gold-standard datasets, i.e. the *Events 2012 Corpus* [16] and the *First Story Detection Corpus* (FSD) [19].

The *Events 2012 Corpus* contains 120 million tweets collected from October to November 2012 from the Twitter streaming API[4], of which 159,952 tweets were labeled as event-related. 506 event types were gathered from the Wikipedia Current Event Portal, and Amazon Mechanical Turk was used to annotate each tweet with one of such event types. Besides, each event was also associated with an event category following

[4] `https://dev.twitter.com/streaming/overview`

the Topic Detection and Tracking (TDT) annotation manual[5] (see Table 2). Events covered by this dataset include, e.g., the US presidential debate between Barack Obama and Mitt Romney, the US presidential election results or the Chemistry Nobel prize. After removing duplicated tweets and those that are no-longer available, we are left with 42,334 tweets related to one out of 506 events.

The *First Story Detection Corpus (FSD)* contains 50 million Twitter messages collected from July 2011 until September 2011 using the Twitter API. Human annotators annotated the tweets related to events with one out of 27 event types extracted from the Wikipedia Current Event Portal. In total, 3,035 tweets were labeled as related to events and annotated with the corresponding event topic (e.g. 'death of Amy Winehouse', 'earthquake in Virginia' or 'plane crash of the Russian hockey team'). After removing tweets that are no more available, we are left with $\sim$31 million tweets from which 2,340 are related to events. Contrary to the Event 2012 corpus, the events in the FSD corpus are not associated with event categories. Therefore, in order to merge the two corpora in a single dataset for our experiments, we extended the FSD corpus by labelling each event topic with one of the event categories of the Event 2012 corpus. The task was manually performed by 3 annotators: the labels were first assigned independently, and then adjudicated by majority vote in case of disagreements (Inter Annotator Agreement: Krippendorff's α=0.758). Since both corpora contain much more non-event related than event related tweets, resulting in a very skewed class distribution, we reduced the number of negative instances by randomly selecting a sample of non event-related tweets.

Event Category	**Event 2012**	**FSD**
Arts, Culture, Entertainment	2,589	710
Armed conflicts & Attacks	7,079	56
Law, Politics & Scandals	16,383	58
Sports	8,812	0
Business & Economy	2,881	342
Science & Technology	1,537	296
Disasters & Accidents	2,479	778
Miscellaneous	574	100
Total	42,334	2,340

Table 2. Tweets in each event category.

4.2 Experimental setup

We compare two external knowledge sources to generalize over NE mentions: *1)* the DBpedia ontology, and *2)* the YAGO ontology. We also test the integration of YAGO for missing categories in DBpedia ontology, but given that this configuration did not improve the performance of the classifiers compared to DBpedia or YAGO alone, we do not report the results in this paper. For the two knowledge bases, we also analyse which generalisation strategy works better by replacing NEs in the tweets either with the

[5] `https://catalog.ldc.upenn.edu/docs/LDC2006T19/TDT2004V1.2.pdf`

most generic or the most specific category in each ontology. As a baseline, we compute the classification performance without entity replacement, using the entity mentions as features. To simulate a real scenario, where large streams of tweets to be classified may describe events and domains different from those in the training data, all results presented in this paper are obtained by training the models on the Event 2012 corpus via cross-validation[6], and testing them on the FSD corpus.

Classification algorithms: We compare different classification algorithms including Naive Bayes (NB), Support Vector Machines (SVM) trained with a degree-2 polynomial kernel, and Recurrent Neural Networks (RNN), which have recently shown to advance state of the art in several NLP tasks [23]. We use the implementations included in the sckit-learn library [18] to train NB and SVM models. As for RNN, we use a multi-layered feed-forward NN with Long Short-Term Memory (LSTM) [10].

Feature Representation: We represent tweets through word embeddings, which have shown a good generalization power, outperforming feature models such as Bag of Words in many NLP tasks [13, 27]. Following the approach proposed in [27], we use tweets in the training set to build word embeddings using the Word2Vec tool with 300-dimensional vectors, context window of size 2 and minimum word frequency of 10. Before building the embeddings, we apply the preprocessing and entity replacement steps (see Section 3.1) to clean up the dictionary and replace NEs by their semantic categories. Thus, we build three variants of the word embedding vectors: *i)* NEs are replaced by their *generic* category; *ii)* NEs are replaced by their *specific* category; *iii)* no NE replacement (i.e. our baseline). We use the same embeddings as features for all the three classification algorithms. Concerning RNN, we train a 5-layer model with 128 hidden nodes consisting in one input layer, three hidden layers and one output layer. In the input layer, tweets are represented by concatenating the vector in the word embeddings corresponding to each word in the input tweets. Words in tweets that do not appear in the word embedding vectors are initialized randomly [13]. The model takes mean of the outputs of all LSTM cells to form a feature vector, and then uses logistic regression and tangent transformation as activation function for the feature vector. We use LSTM as our first hidden layer with batches of 512 examples using Logistic regression as activation function. We use recurrent sum and dropout as second and third layer, respectively. The dropout layer is considered as regularization method for the network. Finally, we use Softmax as the output layer and compute the cost with cross entropy. The implementation is done with Neon[7], a Python-based deep learning library with a GPU back-end.

4.3 Task 1 Evaluation

The first task is the detection of event-related tweets (Section 3.2). We cast the problem as a binary classification task, in which event-related tweets are considered as positive instances and non-event related ones are negative instances. We carry out approximate randomization test to evaluate the statistical significance of our results, allowing us to

[6] For lack of space, results obtained with cross-validation on Events2012 can be found at: `https://goo.gl/b8704o`

[7] `http://neon.nervanasys.com/`

validate our hypothesis. The results reported in Table 3 show that, for the three classification algorithms, entity linking and replacement is effective and always contributes to outperform the baseline. Moreover, the replacement strategy using the most generic ontological category achieves always a better performance than the most specific option. More importantly, the best results are obtained using YAGO to extract NE categories, i.e. relying on WordNet synsets that represent the upper level of YAGO ontology. We got highly significant results ($P<0.001$) when comparing YAGO vs. DBpedia both for the generic and for the specific replacement strategies (RNN), and significant results ($P<0.009$) for YAGO vs DBpedia in both replacement strategies (SVM). Finally, we observe that LSTM-RNNs based on word embeddings yield better results compared to SVM (highly significant for yago:spec (SVM) vs. yago:spec (RNN) and $P<0.06$ for yago:gen (SVM) vs. yago:gen (RNN)).

	NB			SVM			RNN		
Approach	Prec	Rec	F1	Prec	Rec	F1	Prec	Rec	F1
dbp:generic	0.79	0.79	0.79	0.80	0.78	0.79	0.88	0.87	0.87
dbp specific	0.79	0.79	0.79	0.79	0.78	0.77	0.87	0.86	0.86
yago:generic	0.80	0.80	0.80	0.83	0.82	0.82	**0.88**	**0.88**	**0.88**
yago:specific	0.78	0.78	0.78	0.82	0.82	0.82	0.85	0.84	0.84
Baseline (no NER)	0.68	0.67	0.67	0.69	0.67	0.67	0.71	0.71	0.71

Table 3. Results of Task 1: tweet classification as event-related or non event-related (weighted average).

As a comparison, we run the same classification task using cross-validation only on the Event 2012 dataset. In this setting, the baseline outperforms all the other approaches with every algorithm considered. For example, the F-measure of the baseline for the RNN classifier is 0.94 compared to 0.93 when the NEs are replaced by their generic class in YAGO. The difference in performance between the two settings shows that classification based only on in-domain data can be affected by overfitting. This confirms the importance of evaluating the task in a more realistic setting, with training and test data coming from different domains.

4.4 Evaluation of Task 2: Classification of event-related tweets

We evaluate the proposed approach on the task of classifying event-related tweets into event categories. For this task, we consider only tweets related to events in each of the datasets presented in Section 4.1. Table 4 shows the results obtained for the 8 event categories listed in Table 2. In line with the findings on the previous task, LSTM-RNNs outperform the other classifiers in all settings.

Contrary to the findings of Task 1, the classifier performance is higher when using specific categories in DBpedia and YAGO (see for instance, the difference between yago:gen vs. yago:spec, which is highly significant, $p<0.001$). Although the difference between specific and generic categories is in some cases very small, the setting in which NEs are replaced by their most specific category seems more suitable to the multi-class classification of Task 2, while the generic setting targets better the binary classification

of Task 1. The yago:specific categories are more fine-grained than dbp:specific ones, yielding better results in this scenario (results of this comparison are statistically significant, $p<0.01$). Among the different event categories, the worst results are obtained for *Miscellaneous*, that in most cases are assigned to the categories *Politics* and *Economy*.

If we classify only in-domain data using cross-validation with the Event 2012 dataset, the baseline always outperforms the other approaches, like in the binary classification task. Again, this may be due to overfitting and shows the importance of evaluating the task in a different scenario and choosing an approach that generalises over specific entity mentions.

	NB			SVM			RNN		
Approach	Prec	Rec	F1	Prec	Rec	F1	Prec	Rec	F1
dbp:generic	0.75	0.38	0.50	0.72	0.25	0.37	0.82	0.70	0.75
dbp:specific	0.73	0.30	0.43	0.72	0.23	0.35	0.85	0.74	0.79
yago:generic	0.75	0.42	0.54	0.75	0.35	0.48	0.87	0.74	0.80
yago:specific	0.71	0.39	0.50	0.74	0.32	0.45	**0.87**	**0.75**	**0.81**
Baseline (no NER)	0.63	0.22	0.33	0.61	0.22	0.32	*0.72*	*0.62*	*0.67*

Table 4. Experimental results of Task 2: classification of event-related tweets into event categories.

4.5 Combining Task 1 and Task 2

In the experiments described so far, the identification and the classification of event-related tweets have been carried out separately. Specifically, for event classification (Task 2) we considered only tweets related to events. In a real scenario, a combination of the two tasks would be needed if, given a set of tweets, the goal is to understand which event categories can be detected in the data. In this section, we compare two models, one combining Task 1 and 2 in a pipeline, and the other based on a single classification step. We train the classifiers on the Event 2012 dataset and we test on the FSD dataset.

Pipeline model: A model for classifying tweets as event-related or non event-related provides the input to a second model, which classifies event-related tweets into categories (see the box 'Approach 1' in Figure 1). We consider all tweets labeled as event-related by the binary classification algorithm (i.e. both true positive and false positive instances) as input for the second model.

In Table 5, we report the performances of the complete pipeline (the performance of the binary model remains the same as in Table 3). As expected, combining Task 1 and Task 2 in a pipeline yields a performance drop compared with Task 2 in isolation, due to error propagation. Nevertheless, the drop is only around 0.03 points F-measure, that can still be considered as satisfactory. The main issue is precision: since the second model is not trained to handle non event-related tweets, all misclassified instances in the first model are also misclassified by the second one, which lowers precision. However, the recall of the pipeline model is higher than the classification recall in Task 2, due to a lower number of tweets per category, because of event-related tweets misclassified

as non-event related tweets by the binary model (i.e. false negatives). Similar to the results reported in Section 4.4 for Task 2, LSTM-RNNs with entity replacement using the most specific YAGO category outperform all the other settings (the results obtained comparing YAGO vs. DBpedia both for the generic and for the specific replacement strategies (RNN) are significant at $p<0.01$, while the difference between yago:spec vs. yago:gen (RNN) is not significant $p<0.216$). Again, LSTM-RNN baseline achieves a better performance.

	NB			SVM			RNN		
Approach	Prec	Rec	F1	Prec	Rec	F1	Prec	Rec	F1
dbp:generic	0.64	0.48	0.55	0.76	0.29	0.42	0.64	0.82	0.72
dbp:specific	0.56	0.39	0.46	0.72	0.31	0.44	0.60	0.81	0.69
yago:generic	0.57	0.40	0.47	0.73	0.33	0.46	0.72	0.84	0.77
yago:specific	0.59	0.47	0.52	0.77	0.36	0.49	**0.73**	**0.83**	**0.78**
Baseline (no NER)	0.48	0.29	0.36	0.57	0.21	0.30	0.59	0.69	0.64

Table 5. Experimental results of the pipeline model.

Single joint model We compare the pipeline model with a single joint model trained on 9 classes, including the 8 event categories plus a non event-related class (a single multi-class classification step). The input and output are the same as those used for the pipeline experiment. Table 6 reports the evaluation of the joint model (we consider the weighted average Precision, Recall and F1-measure for the event-related classes only). The results between the specific and the generic replacement strategies for YAGO are significant with RNN, while they are not significant when comparing the two replacement strategies using DBpedia.

A comparison between the two approaches shows that the single joint model yields lower results than the pipeline (results are highly significant, $p<0.001$). Recall is particularly affected by several event-related tweets that are classified as non event-related. For example, 60% of the tweets of the category *Economy* were classified as non event-related by SVM.

	NB			SVM			RNN		
Approach	Prec	Rec	F1	Prec	Rec	F1	Prec	Rec	F1
dbp:generic	0.63	0.41	0.49	0.71	0.23	0.34	.78	0.64	0.71
dbp:specific	0.55	0.33	0.42	0.70	0.20	0.31	0.79	0.63	0.70
yago:generic	0.56	0.36	0.44	0.71	0.27	0.39	0.76	0.64	0.70
yago:specific	0.51	0.33	0.40	0.72	0.23	0.35	**0.81**	**0.67**	**0.73**
Baseline (no NER)	0.45	0.23	0.30	0.58	0.11	0.18	0.70	0.58	0.63

Table 6. Experimental results of the single joint model.

5 Conclusions and Future Work

In this paper, we have presented a framework for identifying and classifying event-related tweets by exploiting information automatically leveraged from DBpedia and YAGO. In general we observed that information extracted from YAGO contributes better to improve classification performances than DBpedia. This is probably due to: i) the better coverage of YAGO, and ii) YAGO class hierarchy is deeper than the DBpedia ontology, which has an impact especially when using specific categories for the multi-class classification task. The fact that DBpedia ontology was manually created, while YAGO was built semi-automatically does not affect much our experiments.

In all the experiments, LSTM-RNNs outperform SVM and NB, confirming previous findings on the effectiveness of RNNs when applied to several NLP tasks [23]. Our experiments on different classification tasks show that performing binary classification first and then passing the output to the second classification step in a pipeline is more accurate than the single-step model. A possible future extension of this work could be to exploit domain-specific ontologies for certain categories, for example geographical names. Furthermore, we aim at further specifying event types inside each event category using unsupervised methods. Using event categories collected from diverse knowledge bases could be beneficial also in this case [4]. Exploring the combination of word embeddings and clustering may be another interesting research direction to pursue.

References

1. Anantharam, P., Barnaghi, P., Thirunarayan, K., Sheth, A.: Extracting city traffic events from social streams. ACM Transactions on Intelligent Systems and Technology 9(4) (2014)
2. Atefeh, F., Khreich, W.: A survey of techniques for event detection in twitter. Computational Intelligence 31(1), 132–164 (2015)
3. Becker, H., Naaman, M., Gravano, L.: Beyond trending topics: Real-world event identification on twitter. ICWSM 11, 438–441 (2011)
4. Bryl, V., Tonelli, S., Giuliano, C., Serafini, L.: A Novel Framenet-based Resource for the Semantic Web. In: Proceedings of the 27th Annual ACM Symposium on Applied Computing. pp. 360–365. SAC '12 (2012)
5. Cano, A.E., Varga, A., Rowe, M., Ciravegna, F., He, Y.: Harnessing linked knowledge sources for topic classification in social media. In: Proceedings of the 24th ACM Conference on Hypertext and Social Media. pp. 41–50. ACM (2013)
6. Chang, A.X., Manning, C.D.: Sutime: A library for recognizing and normalizing time expressions. In: LREC. pp. 3735–3740 (2012)
7. Derczynski, L., Maynard, D., Rizzo, G., van Erp, M., Gorrell, G., Troncy, R., Petrak, J., Bontcheva, K.: Analysis of named entity recognition and linking for tweets. Information Processing & Management 51(2), 32–49 (2015)
8. Dou, W., Wang, K., Ribarsky, W., Zhou, M.: Event detection in social media data. In: IEEE VisWeek Workshop on Interactive Visual Text Analytics-Task Driven Analytics of Social Media Content. pp. 971–980 (2012)
9. Genc, Y., Sakamoto, Y., Nickerson, J.V.: Discovering context: classifying tweets through a semantic transform based on wikipedia. In: International Conference on Foundations of Augmented Cognition. pp. 484–492. Springer (2011)
10. Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural computation 9(8), 1735–1780 (1997)

11. Ilina, E., Hauff, C., Celik, I., Abel, F., Houben, G.J.: Social event detection on twitter. In: Web Engineering, pp. 169–176. Springer (2012)
12. Java, A., Song, X., Finin, T., Tseng, B.: Why we twitter: understanding microblogging usage and communities. In: Proceedings of the 9th WebKDD and 1st SNA-KDD 2007 workshop on Web mining and social network analysis. pp. 56–65. ACM (2007)
13. Kim, Y.: Convolutional neural networks for sentence classification. arXiv preprint arXiv:1408.5882 (2014)
14. Lehmann, J., Isele, R., Jakob, M., Jentzsch, A., Kontokostas, D., Mendes, P., Hellmann, S., Morsey, M., van Kleef, P., Auer, S., Bizer, C.: DBpedia - a large-scale, multilingual knowledge base extracted from wikipedia. Semantic Web Journal (2014)
15. Mahdisoltani, F., Biega, J., Suchanek, F.: Yago3: A knowledge base from multilingual wikipedias. In: 7th Conference on Innovative Data Systems Research. CIDR (2014)
16. McMinn, A.J., Moshfeghi, Y., Jose, J.M.: Building a large-scale corpus for evaluating event detection on twitter. In: Proceedings of the 22nd ACM international conference on Conference on information & knowledge management. pp. 409–418. ACM (2013)
17. Owoputi, O., O'Connor, B., Dyer, C., Gimpel, K., Schneider, N., Smith, N.A.: Improved part-of-speech tagging for online conversational text with word clusters. Association for Computational Linguistics (2013)
18. Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., Duchesnay, E.: Scikit-learn: Machine learning in Python. Journal of Machine Learning Research 12, 2825–2830 (2011)
19. Petrović, S., Osborne, M., Lavrenko, V.: Streaming first story detection with application to twitter. In: Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics. pp. 181–189 (2010)
20. Popescu, A.M., Pennacchiotti, M., Paranjpe, D.: Extracting events and event descriptions from twitter. In: Proceedings of the 20th international conference companion on World wide web. pp. 105–106. ACM (2011)
21. Ritter, A., Etzioni, O., Clark, S., et al.: Open domain event extraction from twitter. In: Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining. pp. 1104–1112. ACM (2012)
22. Sakaki, T., Okazaki, M., Matsuo, Y.: Earthquake shakes twitter users: real-time event detection by social sensors. In: Proceedings of the 19th international conference on World wide web. pp. 851–860. ACM (2010)
23. Socher, R., Perelygin, A., Wu, J.Y., Chuang, J., Manning, C.D., Ng, A.Y., Potts, C.: Recursive deep models for semantic compositionality over a sentiment treebank. In: Proceedings of EMNLP. vol. 1631, p. 1642. Citeseer (2013)
24. Song, Y., Wang, H., Wang, Z., Li, H., Chen, W.: Short text conceptualization using a probabilistic knowledgebase. In: Proceedings of the Twenty-Second International Joint Conference on Artificial Intelligence - Volume Volume Three. pp. 2330–2336. IJCAI'11, AAAI Press (2011), http://dx.doi.org/10.5591/978-1-57735-516-8/IJCAI11-388
25. Sprugnoli, R., Tonelli, S.: One, no one and one hundred thousand events: Defining and processing events in an inter-disciplinary perspective. Natural Language Engineering (2017), https://doi.org/10.1017/S1351324916000292
26. Van Erp, M., Rizzo, G., Troncy, R.: Learning with the web: Spotting named entities on the intersection of nerd and machine learning. In: # MSM. pp. 27–30. Citeseer (2013)
27. Yang, X., Macdonald, C., Ounis, I.: Using word embeddings in twitter election classification. arXiv preprint arXiv:1606.07006 (2016)