Filter-Stream Named Entity Recognition: A Case Study at the MSM2013 Concept Extraction Challenge

Diego Marinho de Oliveira[1], Alberto H. F. Laender[1], Adriano Veloso[1], Altigran S. da Silva[2]

[1] Universidade Federal de Minas Gerais, Departamento de Ciência da Computação, Belo Horizonte, Brazil
{dmoliveira,laender,adrianov}@dcc.ufmg.br
[2] Universidade Federal do Amazonas, Instituto de Computação, Manaus, Brazil
alti@icomp.ufam.edu.br

Abstract. Microblog platforms such as Twitter are being increasingly adopted by Web users, yielding an important source of data for web search and mining applications. Tasks such as Named Entity Recognition are at the core of many of these applications, but the effectiveness of existing tools is seriously compromised when applied to Twitter data, since messages are terse, poorly worded and posted in many different languages. In this paper, we briefly describe a novel NER approach, called FS-NER (Filter Stream Named Entity Recognition) to deal with Twitter data, and present the results of a preliminary performance evaluation conducted to assess it in the context of the Concept Extraction Challenge proposed by the 2013 Workshop on Making Sense of Microposts - MSM2013. FS-NER is characterized by the use of filters that process unlabeled Twitter messages, being much more practical than existing supervised CRF-based approaches. Such filters can be combined either in sequence or in parallel in a flexible way. Our results show that, despite the simplicity of the filters used, our approach outperformed the baseline with improvements of 4.9% on average, while being much faster.

Keywords: Twitter, Named Entity Recognition, FS-NER, CRF

1 Introduction

In this paper, we briefly describe a novel NER approach, called FS-NER (Filter Stream Named Entity Recognition), and present the results of a preliminary performance evaluation conducted to assess it in the context of the Concept Extraction Challenge proposed by the 2013 Workshop on Making Sense of Microposts - MSM2013[3]. Traditional approaches for Named Entity Recognition (NER) have demonstrated to be successful when applied to data obtained from typical Web documents, but they are ill suited to Twitter data [2, 3], since Twitter

[3] http://oak.dcs.shef.ac.uk/msm2013/challenge.html

Making Sense of Microposts Workshop @ WWW'13, May 13th 2013, Rio de Janeiro, Brazil

messages are composed of few words and usually written in informal, sometimes cryptic style. FS-NER is an alternative NER approach better suited to deal with Twitter data [1]. In this approach, the NER process is viewed as a coarse grain Twitter message flow (i.e., a Twitter stream) controlled by a series of components, referred to as *filters*. A filter receives a Twitter message coming on the stream, performs specific processing in this message and returns information about possible entities in the message (i.e., each filter is responsible to recognize entities according to some specific criterion). Specifically, FS-NER employs five lightweight filters, exploiting nouns, terms, affixes, context and dictionaries. These filters are extremely fast and independent of grammar rules, and may be combined in sequence (emphasizing precision) or in parallel (emphasizing recall).

In our performance evaluation, we run a set of experiments using micropost data made available by the challenge organizers. Our aim in this challenge was, given a short message (i.e., a micropost), to recognize concepts generally defined as "abstract notions of things". Thus, for the purpose of the challenge our task was constrained to the extraction of entity concepts found in micropost data, characterised by a type and a value, and considering four entity types: *Person*, *Organization*, *Location* and *Miscellaneous*. We also employed a state-of-the-art CRF-based baseline. Our results show that, despite the simplicity of the filters used, our approach outperformed the baseline with improvements of 4.9% on average, while being much faster.

2 Proposed Approach

FS-NER adopts filters that allow the execution of the NER task by dividing it into several recognition processes in a distributed way. Furthermore, FS-NER adopts a simple yet effective probabilistic analysis to choose the most suitable label for the terms in the message being processed. Because of this lightweight structure, FS-NER is able to process large amounts of data in real-time. In what follows, we briefly describe the main FS-NER aspects involved. More details can be found in [1].

2.1 Structure and Design

Let $\mathcal{S} = < m_1, m_2, \ldots >$ be a stream of messages (i.e., tweets), where each m_j in $\mathcal{S}$ is expressed by a pair (X, Y), being X a list of terms $[x_1, x_2, \ldots x_n]$ that compound m_j and Y a list of labels $[y_1, y_2, \ldots, y_n]$, such that each label y_i is associated with the corresponding term x_i and assumes one of the values in the set {Beginning, Inside, Last, Outside, UnitToken}. While X is known in advance for all messages in $\mathcal{S}$, the values for the labels in Y are unknown and must be predicted. For example, the tweet "*RT: I love Mary*" could be represented by $([x_1 = RT:, x_2 = I, x_3 = love, x_4 = Mary], [y_1 = Outside, y_2 = Outside, y_3 = Outside, y_4 = UnitToken])$.

To properly predict labels for Y, we need to provide representative data to generate a recognition model. In FS-NER, a filter is a processing component that estimates the probability of the labels associated with the terms of a message. A set of features is used to support the training of the filters (such features include information like the term itself, or if the first letter of the term is in uppercase). If

a term in X satisfies one of these features, we say that the corresponding filter is activated by the term. Using the training set, we may count the number of times a filter is activated by a given term, and by inspecting the corresponding label we may calculate the likelihood of each pair $\{x_i, y_i\}$ for each filter as expressed by the equation

$$P(y_i = l | X \wedge F = k) \ = \theta_l \tag{1}$$

where F is a random variable indicating that a filter k is being used and θ_l is the probability of associating the label l with the term x_i. The probability θ_l is given by Equation 2, where TP is the number of true positive cases and FN is the number of false negative cases for the term x_i.

$$\theta_l = \frac{TP}{TP + FN} \tag{2}$$

Thus, after trained, a filter becomes able to recognize entities present in the upcoming messages. It is worth noting that each filter employs a different recognition strategy (i.e., a different feature), and thus different predictions are possible for different filters.

In sum, filters are simple abstract models that receive as input a list of terms X and a term $x_i \in X$, and provides as output a set of labels with the associated likelihood, denoted by $\{l, \theta_l\}$. Thus, a filter can be defined by

$$(X, x_i) \xrightarrow{input} F \xrightarrow{output} \{l, \theta_l\}.$$

During the recognition step, the set $\{l, \theta_l\}$ is used to choose the most likely label for the term x_i. However, if used in isolation, filters may not capture specific patterns that can be used for recognition. Fortunately, we may exploit filter combinations to boost recognition performance. Specifically, we may combine filters either in sequence (i.e., if we want to prioritize recognition precision), or in parallel (i.e., if we want to prioritize recognition recall). If combined in sequence, all filters must be activated by the input term, and the corresponding set $\{l, \theta_l\}$ is obtained by treating the combined filters as an atomic one using Equation 1. In this case, it is expected that filters when combined sequentially are able to capture more specific patterns. In contrast, if combined in parallel, the combined filters are not considered as an atomic one. Instead, they simply represent the average of the corresponding likelihoods, as expressed by the equation

$$\frac{1}{Z(\mathcal{F})} \sum_{k=1}^{K} P(y_i = l | X \wedge F = k) \tag{3}$$

where $Z(\mathcal{F})$ is a normalization function that receives as input a list of filters $\mathcal{F}$ and produces as output the number of filters activated by term x_i.

Once trained, the recognition models are used to select the most likely label for each term in the upcoming messages.

2.2 Filter Engineering

In FS-NER, features are encapsulated by five basic filters. They are the *term*, *context*, *affix*, *dictionary* and *noun* filters.

The *term filter* estimates the probability of a certain term being an entity. This estimation is based on the number of times a specific term has been assigned as an entity during the training step. The *context filter* is specially important since it is able to capture unknown entities. Hence, this filter analyzes only the terms around an observed term x_i considering a window of size n and infers whether it is an entity or not. The *affix filter* uses the fragments of an observation x_i to infer if it is an entity. Advantageously, this filter can recognize entities that have similar affix to the entities analyzed before. Thus, this filter makes use of the prefix, infix or suffix of the observation to infer its label y_i. The *dictionary filter* uses lists of names of correlated entities to infer whether the observed term is an entity. The dictionary is important to infer entities that do not appear in the training data. The *noun filter* only considers terms that have just the first letter capitalized to infer if the observed term is an entity.

3 Evaluation

We performed the preliminary evaluation of our approach with the training data made available for the MSM2013 Concept Extraction Challenge. This data includes microposts that refer to entities of types *Person* (PER), *Organization* (ORG), *Location* (LOC) and *Miscellaneous* (MISC). For this, we performed a 5-fold cross validation. To reduce noise, we applied simple preprocessing techniques like removing repeated letters and repeated adjacent terms within a micropost. We also used additional labeled Twitter data from [3] for improving recognition results for entities of types PER and LOC. The standard filter combination adopted for FS-NER was the generalized term filter combination that includes all five proposed filters and presented the best performance in [1]. In the *term* filter, the terms are case sensitive. The context filter, uses prefix and suffix contexts with a window of size three, which presented the best result for F_1 in all collections analyzed. The affix filter uses a prefix, infix and postfix size of 1 to 3. The dictionary filter, specifically, uses the same lists of names of correlated entities considered in [3] and others created from Wikipedia pages. The CRF-based framework used as baseline was the one available at `http://crf.sourceforge.net`, with features functionally similar to the FS-NER filters.

Table 1 presents the obtained results. The line *AVG-Diff* shows the average difference between the FS-NER and CRF-based framework results for all entity types. These results show that, on average, FS-NER outperformed the CRF-based framework by 4.9% for the F_1 metric.

Regarding the test dataset labeling, we followed the same procedure adopted in the preliminary experiment discussed above. In addition, we trained our approach for each entity type separately and then submitted all results together. In case of any intersection between distinct entity types, we chose the entity type that presented the most precise result among them (i.e., PER > LOC > ORG > MISC).

Entity Type	Approach	Precision	Recall	F_1
PER	FS-NER	0.7508	0.7546	0.7520
	CRF	0.7688	0.5350	0.6309
ORG	FS-NER	0.6924	0.4741	0.5612
	CRF	0.7188	0.4702	0.5685
LOC	FS-NER	0.6961	0.5400	0.6069
	CRF	0.7160	0.4656	0.5643
MISC	FS-NER	0.5734	0.3322	0.4185
	CRF	0.5610	0.2847	0.3777
AVG-Diff		-0.0130	0.0864	0.0493

Table 1: Results for FS-NER and the CRF-based framework on the challenge training dataset.

4 Conclusion

In this paper, we have briefly described a novel NER approach, called FS-NER (Filter Stream Named Entity Recognition), and presented the results of a performance evaluation conducted to assess it in the context of the Concept Extraction Challenge proposed by the 2013 Workshop on Making Sense of Microposts - MSM2013. In this challenge, our task was constrained to the extraction of entity concepts found in micropost data, characterised by a type and a value, and considering four entity types: Person, Organization, Location and Miscellaneous. We also employed a state-of-the-art CRF-based baseline. Following previous results [1], our approach outperformed the baseline with improvements of 4.9% on average, while being much faster.

Acknowledgments

This work was partially funded by InWeb - The Brazilian National Institute of Science and Technology for the Web (grant MCT/CNPq 573871/2008-6), and by the authors' individual grants from CNPq, FAPEMIG and FAPEAM.

References

1. D. M. de Oliveira, A. H. F. Laender, A. Veloso, and A. S. da Silva. FS-NER: A Lightweight Filter-Stream Approach to Named Entity Recognition on Twitter Data. In *Proceedings of the 22nd International World Wide Web Conference (Companion Volume)*, pages 597–604, 2013.
2. K. Gimpel, N. Schneider, B. O'Connor, D. Das, D. Mills, J. Eisenstein, M. Heilman, D. Yogatama, J. Flanigan, and N. A. Smith. Part-of-Speech Tagging for Twitter: Annotation, Features, and Experiments. In *Proceedings of the Association for Computational Linguistics (Short Papers)*, pages 42–47, 2011.
3. A. Ritter, S. Clark, Mausam, and O. Etzioni. Named Entity Recognition in Tweets: An Experimental Study. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 1524–1534, 2011.