FS-NER: A Lightweight Filter-Stream Approach to Named Entity Recognition on Twitter Data

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

[†]Departamento de Ciência da Computação
Universidade Federal de Minas Gerais
Belo Horizonte, Brazil
[†]{dmoliveira, laender, adrianov}@dcc.ufmg.br

[‡]Instituto de Computação
Universidade Federal do Amazonas
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. Also, Twitter follows a streaming paradigm, imposing that entities must be recognized in real-time. In view of these challenges and the inappropriateness of existing tools, we propose a novel approach for Named Entity Recognition on Twitter data called FS-NER (Filter-Stream Named Entity Recognition). 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. Moreover, because these filters are not language dependent, FS-NER can be applied to different languages without requiring a laborious adaptation. Through a systematic evaluation using three Twitter collections and considering seven types of entity, we show that FS-NER performs 3% better than a CRF-based baseline, besides being orders of magnitude faster and much more practical.

Categories and Subject Descriptors

H.4 [**Information Systems Applications**]: Miscellaneous

General Terms

Design, Experimentation, Performance

Keywords

Named Entity Recognition, CRF, Twitter, FS-NER

1. INTRODUCTION

Microblogging activity is reshaping the way people communicate. The major microblog platform, Twitter, has more than 500 million users and records over 340 million messages daily, yielding a unique source of data for web search and mining applications such as sentiment analysis, recommendation and entity relation extraction, to name a few. Such applications usually require identifying free-text references to named entities such as people, organizations, places, companies, and others [14] − a task commonly known as Named Entity Recognition (NER).

Dominant NER approaches are either based on linguistic grammar-based techniques or on statistical models. Grammar-based NER approaches are dependent on a specific language, while statistical NER approaches typically require a large amount of manually annotated training data. Both approaches have demonstrated to be successful when applied to data obtained from typical Web documents, but they are ill suited when it comes to Twitter data [6, 16]. Twitter messages are composed of a small amount of words and they are written in informal, telegraphic, sometimes cryptic style. These characteristics make hard the identification of entities. Also, Twitter messages keep coming at a fast pace in the stream, and we cannot afford to gather information from external sources on-the-fly, nor to produce training data continuously. Instead, given the restrictions imposed by the data streaming paradigm, NER approaches to Twitter data must operate with limited computing and training resources.

In this paper we propose a novel NER approach, called FS-NER (Filter Stream Named Entity Recognition), which is an alternative better suited to deal with Twitter data. Essentially, 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 (that is, each filter is responsible to recognize entities according to some specific criterion). Filters can be as simple as considering capitalized letters or using dictionaries, and thus are extremely fast, being able to perform real-time entity recognition. However, if used in isolation, these filters are not likely to provide satisfactory recognition performance. On the other hand, when these filters are used in combination, the aggregate performance increases significantly. Performance improvement is mainly explained by the independence and complementarity that exist among diverse filters. Specifically, FS-NER employs five lightweight filters, exploiting nouns, terms, affixes, context and dictionaries. These filters are extremely fast and

WWW 2013 Companion, May 13–17, 2013, Rio de Janeiro, Brazil.
ACM 978-1-4503-2038-2/13/05.

independent of grammar rules, and may be combined in sequence (emphasizing precision) or in parallel (emphasizing recall).

To evaluate the effectiveness of our approach, we performed a systematic set of experiments using Twitter data. We employed three collections: one containing messages in English, another containing messages in Portuguese, and a third one containing messages in different languages. Our evaluation is based on identifying seven types of entity and we employ state-of-the-art CRF-based baselines. Our results show that, despite the simplicity of the filters used, our approach is still able to outperform the baselines with improvements of 3% on average, while being orders of magnitude faster and thus more appropriate to the data streaming paradigm followed by Twitter.

Thus, this paper presents the following contributions: (1) a discussion on the main design challenges involving NER on Twitter data, (2) the proposal of a novel approach based on the filter-stream paradigm to tackle NER on Twitter data and (3) a detailed experimental evaluation that compares the performances of FS-NER and CRF-based approaches for the NER task.

The remainder of this paper is organized as follows. Section 2 addresses related work. Section 3 discusses the main challenges involved when performing NER on Twitter data. Section 4 describes our FS-NER approach. Section 5 presents our experiments and results. Finally, Section 6 presents our final considerations and discusses future work.

2. RELATED WORK

Problems related to Named Entity Recognition were first introduced in 1995 as part of MUC-6 (Message Understanding Conference). Identifying entities in unstructured text is a nontrivial task and several approaches have been developed to address it [8]. In general, these approaches are devised to recognize entities such as names of people, organizations, locations, among others. With the evolution of the area, new types of entity and domain were considered targets of interest. Traditionally, most work on entity recognition has been carried out on the context of a same subject area or preestablished domain, such as news [2, 3, 17]. We refer the reader to [14] for a comprehensive survey of the area.

Considering NER on Twitter data, few studies have been developed in this context so far. Ritter *et al.* [16] considered techniques usually applied to traditional NER and adapted them to Twitter. However, despite the reasonable results obtained, the dependency provided by the use of NPL techniques makes the framework slow and difficult to apply to a variety of situations. In another work [11], Liu *et al.* use the kNN (k-Nearest Neighbors) algorithm and a CRF-based approach for composing a semi-supervised system. The general idea is to use kNN to label tweets in a word level, and then apply linear CRFs in order to execute a fine-grained classification over the results obtained by the kNN algorithm. However, the use of two systems increases complexity. The choice of features becomes a major problem, since it is needed to deal with a satisfactory combination of them to fulfill the function of both systems together. Not only that, the combination of the two systems can decrease runtime performance. More recently, Li *et al.* [10], have proposed a two-step, unsupervised NER approach targeted to Twitter data, called TwiNER. This approach deals with streams, but due to the adopted strategies it is not capable of processing tweets in real time and only identifies if a phrase (text segment) is an entity or not, i.e., it does not determine the class of the identified entity.

Due to the scarcity and high cost to obtain a considerable amount of labeled tweets, learning transfer is a relevant issue. However, using formal sources to train an entity recognizer and then applying it to Twitter data, Locke and Martin [12] have concluded that due to the Twitter nature it is difficult to transfer learning from one domain to another. In another study [5], Finin *et al.* describe how to efficiently use the Amazon Mechanical Turk to annotate data from Twitter. Jung [9] suggests that using clusters of related tweets can alleviate the lack of contextual data. His results show an increase in precision. However, he did not investigate how precision impacts other metrics such as recall and F_1. Michelson and Macskassy [13] applied NER techniques to tweets in order to discover topics of interest to the users.

Despite the importance of supporting NER on Twitter data, only few works concern the main aspects needed to produce scalable and practical approaches for such an environment. As previously mentioned, social network platforms such as Twitter produce content in several languages and in real-time. Considering current approaches, most of them require long time for training a recognition model or are restricted to a specific language, making them very costly or unfeasible to adapt to other languages. On the other hand, our approach, which is based on the filter-stream paradigm, relies on filters that are lightweight processing components that receive messages coming on the Twitter stream. Although simple, our language-independent filters can be efficiently combined in order to boost recognition performance, thus alleviating many of the challenges related to NER on Twitter data.

3. DESIGN CHALLENGES

Recent work [11, 12, 16] has reported several difficulties and impediments for applying NER techniques to Twitter data, and called for more flexible and effective methods to carry out the NER task in such a more challenging environment. In this section, we discuss the main design challenges faced in this task.

Large volume of data. Twitter produces a huge volume of data every day due to the large number of users and the intense interaction among them. This means that more efficient methods and tools are needed to deal with NER on Twitter data. For instance, approaches that require an iterative process to generate their models may have their performance heavily affected. Considering real scenarios, the use of such approaches may become a bottleneck in terms of computing performance. Probabilistic approaches that rely on iterative learning process should use lighter and more efficient features to address NER in this environment.

Lack of formalism. Microblog platforms, and Twitter in particular, are environments that are dominated by the lack of language formalism. Thus, mispellings, abbreviations, punctuation misuse, and grammatical errors are very common in this context. This drastically affects the effectiveness of language-based NER approaches when recognizing entities and their relationships.

Language diversity. Despite the predominance of some languages, such as English, Japanese, Portuguese and Spanish, Twitter presents an enormous diversity in this aspect [7].

A particular challenge happens when it is necessary to identify entities in languages, such as Bengali [4], for which the NER process is intrinsically more complex due to specific grammar characteristics. Thus, the need of processing different languages may introduce difficulties to the NER task on Twitter and approaches that excessively rely on features provided for a specific language may become inadequate in this environment.

Real-time nature. Twitter is characterized to be very dynamic in terms of interaction between its users. Thus, large volumes of tweets are posted during short periods of time, which requires real-time processing capabilities in order to provide up-to-date information.

Lack of contextualization. The fact that tweets are short messages may result in insufficient contextual evidence on the text for judging the terms in order to recognize entities. Commonly, NER approaches employ the contextual information around a term or related sentences to discard terms that are not references to entities. As an example, suppose we want to recognize the occurrence of company names in the tweet "*RT: I bought at J&J.*". In this context, "*J&J*" may be evaluated as a candidate term. However, based only on the available information, recognizing "*J&J*" as a company name can be misleading, since there is no other evidence to ensure that.

Data stream orientation. Twitter is also characterized by transmitting messages in the form of data streams, which results in a quick spread of tweets over the network. Therefore, it is necessary to take into account the rapid emergence of new contexts and scenarios in which entities are mentioned.

4. PROPOSED APPROACH

The challenges discussed in the previous section make clear the need for alternative NER approaches to deal with Twitter data. In this section, we describe FS-NER (Filter-Stream Named Entity Recognition), our novel approach proposed to perform the NER task. FS-NER adopts filters that allow the execution of the NER task by dividing it into several recognition processes in a distributed way. Further, 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.

4.1 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 NEW YORK*" could be represented by $([x_1 = \textit{RT:}, x_2 = I, x_3 = \textit{love}, x_4 = \textit{NEW}, x_5 = \textit{YORK}], [y_1 = \textit{Outside}, y_2 = \textit{Outside}, y_3 = \textit{Outside}, y_4 = \textit{Beginning}, y_5 = \textit{Last}])$.

In order to properly predict labels for Y, we need to provide correct and representative data to generate a recognition model. In the case of 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 process of the filters (such features include information like as 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 number of times that a given label was assigned correctly, we may calculate the likelihood of a label being assigned to each term x$_i$ by 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 respective likelihood associated with each of them, 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 in order 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[1]. 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

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

[1] For example, consider the term "New". It would activate a filter by stating that *if the term is "New", then the likelihood of label l is* θ_l^T. The same term would also activate another filter by stating that *if the first letter of the term is in uppercase, then the likelihood of label l is* θ_l^N. If these two filters are combined sequentially, the combined filters would state that *if the term is "New" and the first letter is in uppercase, then the likelihood of label l is* $\theta_l^{T \wedge N}$.

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.

Therefore, we may propose specific recognition models, involving different combination alternatives such as the ones depicted in Figure 1. Each proposed model may be then formally described using the expressions defined by Equations 1 and 3. For example, the filter combination in Figure 1 comprises three filter sequences, (F_1, F_4), (F_2) and (F_3), which converge to filter F_5. Thus, the recognition model that describes this filter combination comprises three sequential filters given by $P(y_i = l|X \wedge F_1 \wedge F_4 \wedge F_5)$, $P(y_i = l|X \wedge F_2 \wedge F_5)$ and $P(y_i = l|X \wedge F_3 \wedge F_5)$, which are then combined in parallel. This leads to the following recognition model $\mathcal{M}$ for the filter combination in Figure 1:

$$\begin{aligned}\mathcal{M} \; &= \frac{1}{Z(\mathcal{F})}(P(y_i = l|X \wedge F_1 \wedge F_4 \wedge F_5) \\ &+ \; P(y_i = l|X \wedge F_2 \wedge F_5) \; + \; P(y_i = l|X \wedge F_3 \wedge F_5)).\end{aligned}$$

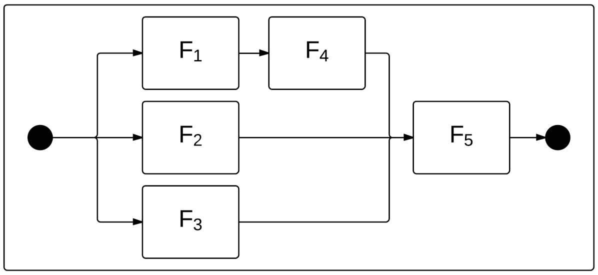

Figure 1: Example of a filter combination.

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

4.2 Filter Engineering

One of the most crucial steps in the recognition process is the choice of the features. In FS-NER, features are encapsulated by filters. Therefore, choosing the right filters are decisive for the proper performance of our approach. Thus, below we discuss the basic filters used by FS-NER. They are the *term*, *context*, *affix*, *dictionary* and *noun* filters. Although in this paper we consider only the aforementioned filters, many others may be devised.

Term. The *term filter* estimates the probability of a certain term being an entity. This filter has the ability to distinguish ambiguous terms discarding them when they present low probability. For example, given the need to recognize entities of type *Place*, the term "New" in "New York" would probably be discarded if analyzed separately. This happens because the term "New" is very common and appears in several situations where it does not correspond to an entity of type *Place*, thus requiring other features possibly provided by other filters. On the other hand, "Nashville" would possibly achieve a higher probability of being an entity, since it is a less common term usually assigned to the entity type *Place*.

Context. 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.

Affix. 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.

Dictionary. The *dictionary filter* uses a list 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.

Noun. The *noun filter* only considers terms that have just the first letter capitalized to infer if the observed term is an entity. Although capitalized terms are a weak evidence in Twitter data, this filter can recognize entities when wisely used.

5. EXPERIMENTAL EVALUATION

In this section we present a detailed experimental evaluation of FS-NER. This evaluation comprises two sets of experiments. The first examines the recognition performance of individual and combined filters, and the second compares the recognition and run-time performances of FS-NER and our CRF-based baselines. In all experiments, results are shown in terms of *precision*, *recall* and F_1, which are metrics widely used in the information retrieval realm. FS-NER has been implemented in Java, since this programming language is highly portable and facilitates the use of our framework in different domains and applications.

5.1 Setup

All five filters are used in ours experiments, *i.e.*, the *term*, *context*, *affix*, *dictionary* and *noun* filters. 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 entities considered in [16] and others created from Wikipedia pages. Three different collections of Twitter data, called OW, ETZ and WT, are employed in the experiments. All experiments adopt a 5-fold cross-validation and the final results are the average over the five runs.

OW **collection**. This collection consists of approximately 2,000 manually labeled tweets. These tweets are related to soccer teams playing in the Brazilian National League and are all in Portuguese. In this collection, we seek to identify three types of entity, namely: player names (*Player*), venue names (*Venue*) and team names (*Team*). It is worth mentioning that the Portuguese language uses spelling tones and accents, which further complicates the entity recognition task.

ETZ **collection**. This collection consists of approximately 2,400 manually labeled tweets and was used in [16]. Tweets in this collection were randomly crawled and are all in English. There are three relevant types of entity: company names (*Company*), geographic places (*Place*) and person names (*Person*). The small amount of samples available in relation to the large number of entities to be recognized is the major challenge in this collection. Other types of entity in this collection were discarded given that they correspond to a small fraction of the existing entities.

WT **collection**. This collection consists of approximately 44,000 tweets semi-manually labeled and supplied by the *WePS3* task [1]. Tweets in this collection are related to organizations, and the type of entity we are interested in recognizing is organization names (Org). The challenges related to this collection include the diversity of languages and the different contexts in which an entity may appear. Most tweets in this collection are written in English and Spanish, but there are also tweets in Japanese and Portuguese.

5.2 Performance Analysis

Next we analyze the recognition performance of our FS-NER approach. First we analyze the recognition performance of individual filters, and then the recognition performance of four specific filter combinations.

5.2.1 Analysis of Individual Filters

This analysis aimed to observe the behavior of the term (F_T), context (F_C), dictionary (F_D), affix (F_A) and noun (F_N) filters when individually applied. Table 1 presents the results. The term filter achieved the best F_1 results. In general, this filter was efficient to recognize entities and presented high values for precision and recall. However, when analyzing the terms, this filter was not able to generalize since it only recognizes terms that have been observed in the training set. The context filter achieved the best precision results for most cases. Furthermore, this filter was susceptible to recognize new entities, thus being quite useful. The dictionary filter also showed high precision values but relatively low recall ones. In most cases, this filter was also able to generalize. The affix and noun filters showed the highest values for recall, but low values for precision. From these results we can see that the term, context and dictionary filters are suitable for being individually applied whereas the affix and noun terms are not, i.e., they must be combined with other filters to improve their performance. Also notice that in some cases the dictionary and noun filters were not useful because during the training step they always predicted a wrong label when activated. However, as we will show next, these filters become useful when used in combination with the others.

Entity Type	Filter	Precision	Recall	F$_1$
Player	F$_T$	0.8914±0.05	0.6187±0.10	0.7276±0.08
	F$_C$	0.9470±0.05	0.2517±0.06	0.3941±0.08
	F$_D$	0.7990±0.09	0.4274±0.06	0.5539±0.05
	F$_A$	0.0965±0.01	0.9201±0.04	0.1743±0.02
	F$_N$	0.3028±0.05	0.7950±0.05	0.4373±0.06
Venue	F$_T$	0.8526±0.07	0.6693±0.08	0.7449±0.03
	F$_C$	0.9092±0.05	0.4058±0.03	0.5602±0.03
	F$_D$	0.9166±0.01	0.4581±0.10	0.6050±0.10
	F$_A$	0.0421±0.01	0.7723±0.07	0.0798±0.02
	F$_N$	0.0000±0.00	0.0000±0.00	0.0000±0.00
Team	F$_T$	0.8769±0.01	0.8406±0.03	0.8580±0.01
	F$_C$	0.9389±0.01	0.3317±0.03	0.4896±0.03
	F$_D$	0.8157±0.03	0.4431±0.03	0.5736±0.02
	F$_A$	0.3610±0.01	0.9049±0.02	0.5160±0.02
	F$_N$	0.5787±0.03	0.6034±0.02	0.5907±0.02
Company	F$_T$	0.6908±0.10	0.3796±0.12	0.4824±0.11
	F$_C$	0.7200±0.11	0.1788±0.07	0.2805±0.08
	F$_D$	0.0000±0.00	0.0000±0.00	0.0000±0.00
	F$_A$	0.0415±0.01	0.6353±0.10	0.0777±0.02
	F$_N$	0.0000±0.00	0.0000±0.00	0.0000±0.00
Place	F$_T$	0.6965±0.05	0.2499±0.08	0.3618±0.09
	F$_C$	0.7503±0.22	0.1018±0.06	0.1761±0.09
	F$_D$	0.9444±0.08	0.0775±0.03	0.1419±0.05
	F$_A$	0.0440±0.01	0.6466±0.05	0.0823±0.01
	F$_N$	0.0000±0.00	0.0000±0.00	0.0000±0.00
Person	F$_T$	0.8089±0.08	0.3161±0.01	0.4539±0.02
	F$_C$	0.9246±0.03	0.1180±0.03	0.2083±0.04
	F$_D$	0.0000±0.00	0.0000±0.00	0.0000±0.00
	F$_A$	0.0958±0.02	0.7903±0.02	0.1705±0.03
	F$_N$	0.3015±0.03	0.7478±0.04	0.4281±0.03
Org	F$_T$	0.7690±0.01	0.7503±0.01	0.7595±0.01
	F$_C$	0.7742±0.01	0.3109±0.00	0.4436±0.00
	F$_D$	0.4000±0.49	0.0002±0.00	0.0003±0.00
	F$_A$	0.1444±0.01	0.6591±0.00	0.2368±0.01
	F$_N$	0.0000±0.00	0.0000±0.00	0.0000±0.00

Table 1: Results when applying the filters individually.

5.2.2 Analysis of Specific Filter Combinations

In our previous set of experiments, we showed that the term, context and dictionary filters present a good performance when individually applied whereas the affix and noun filters must be applied with more caution. In addition, each filter has a particular property that is assessed by precision, recall and F_1. Due to this and the many ways that our filters can be combined, our next experiments seek by some intuition to propose and analyze specific filter combinations that might be useful for performing the NER task using FS-NER. Thus, in what follows we evaluate four specific recognition models centered on combinations of the term, noun and context filters. The proposed filter combinations are based on the results achieved by each filter individually and how complementary they are.

Recognition centered on the term filter (TRM). This combination is the simplest one and aims to recognize entities based on terms previously analyzed. Because it relies on the existence of these terms in the training set, this filter combination is not able to generalize. The recognition model of this filter combination is given by

$$\begin{aligned}\mathcal{M} \;=\; & \frac{1}{Z(\mathcal{F})}(P_1(y_i = l|X \wedge F_T) \;+\; P_2(y_i = l|X \wedge F_T \wedge F_C) \\ & +\; P_3(y_i = l|X \wedge F_T \wedge F_N) \\ & +\; P_4(y_i = l|X \wedge F_T \wedge F_C \wedge F_N)).\end{aligned}$$

Table 2 shows the results for the term filter combination. As we can see, this filter combination was able to recognize various entity types with high precision and good recall.

Entity Type	Precision	Recall	F$_1$
Player	0.8916±0.05	0.6213±0.10	0.7294±0.09
Venue	0.8608±0.07	0.7304±0.10	0.7857±0.06
Team	0.8746±0.01	0.8495±0.03	0.8616±0.01
Company	0.7039±0.09	0.3993±0.12	0.5022±0.10
Place	0.6972±0.05	0.2550±0.08	0.3676±0.08
Person	0.8103±0.08	0.3181±0.01	0.4600±0.02
Org	0.7768±0.01	0.7985±0.01	0.7875±0.01

Table 2: Results for the term filter combination.

Recognition centered on the term filter with generalization ($GTRM$). This combination aims to provide a strategy to analyze, integrally or partially, the terms of a message. Thus, this strategy keeps the characteristics of the term filter combination, at the same time that provides some generalization ability. The recognition model of this filter combination is given by

$$\begin{aligned}\mathcal{M} \;=\; & \frac{1}{Z(\mathcal{F})}(P_1(y_i = l|X \wedge F_T) \;+\; P_2(y_i = l|X \wedge F_A \wedge F_C) \\ & +\; P_3(y_i = l|X \wedge F_D \wedge F_N)).\end{aligned}$$

Table 3 shows the results for this filter combination. As we can see, it achieved better results than the combination solely centered on the term filter (TRM). This was due to its ability to generalize, since this filter combination does not rely only on the terms present in the training set.

Entity Type	Precision	Recall	F$_1$
Player	0.8411±0.04	0.6930±0.08	0.7573±0.06
Venue	0.8468±0.05	0.6809±0.09	0.7499±0.04
Team	0.8557±0.01	0.8667±0.02	0.8610±0.01
Company	0.6969±0.09	0.3858±0.10	0.4900±0.09
Place	0.7439±0.04	0.3102±0.07	0.4329±0.07
Person	0.6345±0.05	0.6098±0.03	0.6195±0.02
Org	0.7453±0.01	0.7924±0.01	0.7681±0.01

Table 3: Results for the generalized term filter combination.

Recognition centered on the noun filter (NON). This combination aims to analyze the ability of our filters to recognize entities based mainly on noun evidence found in terms present in the tweets. The recognition model of this filter combination is given by

$$\begin{aligned}\mathcal{M} &= \frac{1}{Z(\mathcal{F})}(P_1(y_i = l|X \wedge F_T \wedge F_N) \\ &+ P_2(y_i = l|X \wedge F_D \wedge F_N) + P_3(y_i = l|X \wedge F_A \wedge F_N) \\ &+ P_4(y_i = l|X \wedge F_C \wedge F_N)).\end{aligned}$$

Table 4 presents the results of the noun filter combination. As we can see, when properly applied, this filter is able to provide good results. However, to obtain such results, it is necessary to apply the noun filter in conjunction with more reliable ones. This means that, despite being a weak evidence in Twitter data, capitalization helps to recognize entities with relative high precision.

Entity Type	Precision	Recall	F$_1$
Player	0.8305±0.04	0.6288±0.07	0.7137±0.06
Venue	0.8515±0.06	0.5852±0.12	0.6866±0.09
Team	0.8349±0.02	0.5670±0.02	0.6750±0.02
Company	0.7147±0.19	0.2178±0.07	0.3240±0.08
Place	0.6963±0.08	0.2023±0.04	0.3107±0.06
Person	0.6309±0.05	0.5765±0.03	0.6000±0.02
Org	0.7691±0.02	0.5325±0.01	0.6292±0.01

Table 4: Results for the noun filter combination.

Recognition centered on the context filter (CTX). This combination exploits the ability to recognize entities based only on the context around the current observation, thus softening problems derived from terms out of the vocabulary. Therefore, all filters but the term one are incrementally combined with the context filter. The recognition model of this filter combination is given by

$$\begin{aligned}\mathcal{M} &= \frac{1}{Z(\mathcal{F})}(P_1(y_i = l|X \wedge F_C) + P_2(y_i = l|X \wedge F_C \wedge F_A) \\ &+ P_3(y_i = l|X \wedge F_C \wedge F_D) + P_4(y_i = l|X \wedge F_C \wedge F_N) \\ &+ P_5(y_i = l|X \wedge F_C \wedge F_A \wedge F_D) \\ &+ P_6(y_i = l|X \wedge F_C \wedge F_A \wedge F_N) \\ &+ P_7(y_i = l|X \wedge F_C \wedge F_D \wedge F_N) \\ &+ P_8(y_i = l|X \wedge F_C \wedge F_A \wedge F_D \wedge F_N)).\end{aligned}$$

Table 5 presents the results for the context filter combination. As we can see, among the filter combinations analyzed, this was the one that achieved the highest precision values. When considering recall, however, it presented the lowest values. Despite that, this filter combination is the most restrictive and reliable among all proposed combinations due to its high precision.

Entity Type	Precision	Recall	F$_1$
Player	0.9470±0.05	0.2517±0.06	0.3941±0.08
Venue	0.9092±0.05	0.4058±0.03	0.5602±0.03
Team	0.9391±0.01	0.3330±0.03	0.4911±0.03
Company	0.7200±0.11	0.1788±0.07	0.2805±0.08
Place	0.7503±0.22	0.1018±0.06	0.1761±0.09
Person	0.9246±0.03	0.1180±0.03	0.2083±0.04
Org	0.7205±0.01	0.3178±0.00	0.4410±0.00

Table 5: Results for the context filter combination.

In summary, when analyzing the above proposed filter combinations, we see that each one presents a particularity. For example, the term filter combination showed good F_1 results, but since it is not able to generalize, its use is quite restricted. The noun filter combination, on the other hand, presented good F_1 results and was able to generalize, but its results were, in general, inferior to those presented by the term filter combination. The generalized term filter combination, in turn, provided the best general results in terms of F_1 for all entity types but *Venue*, *Company* and *Org*. Finally, the context filter combination was the most reliable and restrictive among the filter combinations analyzed.

Table 6 presents a summary of the results obtained by the proposed filter combinations for each entity type. Looking at the figures, we find that, among the proposed filter combinations, $GTRM$ is the one that showed the best overall performance. Thus, for the comparative experiments reported next, we consider this filter combination as the representative of the FS-NER approach.

Entity Type	Filter Combinations			
	$F_1(TRM)$	$F_1(GTRM)$	$F_1(NON)$	$F_1(CTX)$
Player	0,73±0,09	0,76±0,06	0,71±0,06	0,39±0,08
Venue	0,79±0,06	0,75±0,04	0,69±0,09	0,56±0,03
Team	0,86±0,01	0,86±0,01	0,68±0,02	0,49±0,03
Company	0,50±0,10	0,49±0,09	0,32±0,08	0,28±0,08
Place	0,37±0,08	0,43±0,07	0,31±0,06	0,18±0,09
Person	0,46±0,02	0,62±0,02	0,60±0,02	0,21±0,04
Org	0,79±0,01	0,77±0,01	0,63±0,01	0,44±0,01
Average	0,64	0,67	0,56	0,36
Std. Dev.	0,19	0,16	0,17	0,14

Table 6: Summuary of the results obtained by the proposed filter combinations.

5.3 Comparison with CRF-Based Approaches

The comparative analysis involves assessing the efficiency both in terms of recognition performance and execution time. As baselines, we used CRF-based approaches available at `http://crf.sourceforge.net`. All experiments were performed in similar conditions, also considering the OW, ETZ and WT collections and all viable combinations of non-linear filters for the approach FS-NER.

Table 7 shows the results obtained by competing approaches for the different entity types, in terms of precision, recall and F_1. The CRF-based approaches are presented in two distinct configurations. The first configuration, called SCRF(1), represents the SCRF in its standard configuration. The second configuration, called SCRF(2), is a modified version of SCRF(1) that also uses the features exploited by FS-NER. As can be seen in Table 7, for the *OW* collection the influence of noise caused by spelling errors does not significantly affect the efficiency of the NER process. On the other hand, the results obtained for the *ETZ* collection were not as impressive. From an analysis of entity distribution for the three collections, we noticed that the *ETZ* collection presents high percentage of entities that are out of the known vocabulary, *e.g.*, 72% for entity type *Person*. This difference contributes to the achievement of poor results as those presented by the three evaluated approaches. Moreover, the small number of samples in the *ETZ* collections, affected the recognition process as a whole. For the *WT* collection, all approaches obtained similar results. In general, the hardest problem in this collection is to recognize precisely entities in different contexts and subtle variations of entity names.

Entity Type	Approach	Precision	Recall	F$_1$
Player	SCRF(1)	0.9245±0.05	0.5942±0.10	0.7207±0.09
	SCRF(2)	0.8918±0.05	0.6358±0.07	0.7407±0.06
	FS-NER	0.8411±0.04	0.6930±0.08	0.7573±0.06
Venue	SCRF(1)	0.9300±0.03	0.7135±0.07	0.8058±0.04
	SCRF(2)	0.8737±0.08	0.6665±0.09	0.7502±0.04
	FS-NER	0.8468±0.05	0.6809±0.09	0.7499±0.04
Team	SCRF(1)	0.8898±0.01	0.8368±0.03	0.8620±0.01
	SCRF(2)	0.8659±0.01	0.8543±0.03	0.8598±0.01
	FS-NER	0.8557±0.01	0.8667±0.02	0.8610±0.01
Company	SCRF(1)	0.8240±0.09	0.3782±0.11	0.5125±0.12
	SCRF(2)	0.7281±0.10	0.3858±0.11	0.4981±0.10
	FS-NER	0.6969±0.09	0.3858±0.10	0.4900±0.09
Place	SCRF(1)	0.7824±0.09	0.2346±0.09	0.3534±0.10
	SCRF(2)	0.6952±0.08	0.2703±0.10	0.3834±0.11
	FS-NER	0.7439±0.04	0.3102±0.07	0.4329±0.07
Person	SCRF(1)	0.7208±0.37	0.3107±0.04	0.3801±0.15
	SCRF(2)	0.8041±0.06	0.3243±0.03	0.4613±0.04
	FS-NER	0.6345±0.05	0.6098±0.03	0.6195±0.02
Org	SCRF(1)	0.7598±0.02	0.7123±0.01	0.7351±0.01
	SCRF(2)	0.7506±0.03	0.7531±0.02	0.7511±0.01
	FS-NER	0.7453±0.01	0.7924±0.01	0.7681±0.00

Table 7: Detailed results for F$_1$ considering the FS-NER and CRF-based approaches.

Table 8 shows the comparison between the recognition performance obtained by FS-NER and the CRF-based baselines, in terms of F_1. In addition to the results obtained by these approaches, the table also shows the results obtained by the approach proposed in [16], which we call RCME (name derived from the surnames of the authors). The RCME column is used to verify how close are the results obtained by FS-NER and the CRF-based approaches for the RCME solution (which is considered an upper-bound, since it uses additional information about the NER process). The *Diff* column refers to the difference in terms of F_1 between FS-NER and SCRF(2). Column t represents the sum of the difference values obtained by Student's t-test and *p-value* is the probability value associated with the t-test.

From Table 8, we can note that the differences between F_1 results obtained by different approaches are minimal. Analyzing the difference we observed that FS-NER achieves results that are, on average, 3% superior than those obtained by the CRF-based approaches. In the cases of entity types *Venue*, *Place* and *Person* the differences are above 3%. For the cases of entity type *Company*, the CRF-based approaches presented better F_1 results.

Regarding the results obtained by the RCME approach, it is clear that there is a significant difference when compared with the results obtained by FS-NER and CRF-based approaches. In principle, the RCME approach is constituted by context, word clustering, dictionary, spelling, pos-tagging and chunk features. Furthermore, this approach separates the recognition process into two phases: segmentation and classification. The first phase is related to the recognition of the entity regardless of whether it belongs to one of the three types of entity presented in the *ETZ* collection. The second phase is responsible for associating one of the three types of entity to the entity term. In this phase, the authors use a supervised topic model called LabeledLDA [15]. Because of the lack of possibility of training the RCME approach in order to recognize entities in the other collections and the high cost to prepare an adequate model considering the needs of this approach, we are only able to speculate. We speculate that the process of recognition being split into two phases produces better results for the *ETZ* collection.

Entity Type	RCME	FS-NER	SCRF(2)	Diff.	t	p-value
Player	-	0.76±0.06	0.74±0.06	0.02	1.33	0.25
Venue	-	0.75±0.04	0.75±0.04	0.00	-0.04	0.97
Team	-	0.86±0.01	0.86±0.01	0.00	0.43	0.69
Company	0.58±0.07	0.49±0.09	0.50±0.10	-0.01	-1.76	0.15
Place	0.73±0.05	0.43±0.07	0.38±0.11	0.05	2.06	0.11
Person	0.78±0.04	0.62±0.02	0.46±0.04	0.16	6.65	0.00
Org	-	0.77±0.01	0.75±0.01	0.02	5.71	0.01
Average	0.69	0.67	0.63	0.03	-	-
St. Dev.	0.10	0.15	0.18	0.06	-	-

Table 8: Detailed results for F$_1$ considering the RCME, FS-NER and CRF-based approaches. Diff represents the diference between the FS-NER and CRF results.

5.4 Execution Time Comparison

In the last set of experiments we used 22,000 tweets from the *WT* collection. We adopted this collection to better highlight the difference of computational cost between the competing approaches. For best judgment, the experiments were executed 100 times for each iteration. During each iteration about 2,200 new tweets were added to the previous training set.

Figure 2 shows the results for the comparison of average runtime, involving FS-NER and the CRF-based approaches. From the results we observed a large difference in runtime performance between the CRF-based approaches when compared to FS-NER. The main difference between runtime results is due to the fact that FS-NER does not perform any iterative training procedure in order to build the recognition model. On the other hand, the CRF-based approaches require an iterative process to adjust their weights for the features associated during the NER process. This could be exacerbated for the CRF-based approaches, considering the need to update the model for recognition. In this case, due to excessive retraining, the performance of the CRF-based approaches would be deteriorated. In contrast, because of the lightweight structure of FS-NER, the cost for updating the model is almost negligible when compared to the cost associated with the CRF-based approaches.

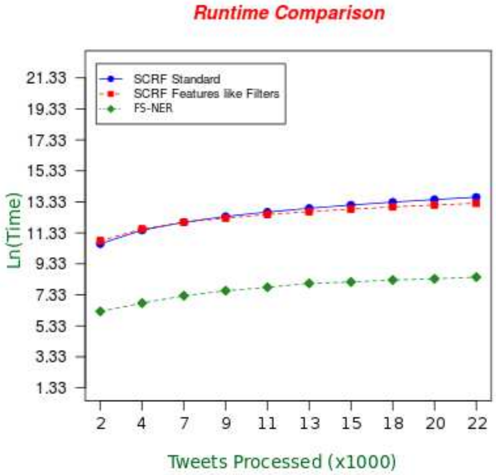

Figure 2: Comparative results for the runtime performance in ms between FS-NER and the CRF-based approaches.

6. CONCLUSIONS AND FUTURE WORK

This paper focuses on the important problem of Named Entity Recognition (NER) on Twitter data. Devising practical and effective approaches to NER in such scenario is particularly challenging. We have introduced a new approach, FS-NER (Filter-Stream Named Entity Recognition), which is more suitable to deal with Twitter data. The proposed approach is based on a efficient structure composed of lightweight filters. These filters exploit distinct features and can be combined in sequence or in parallel. In addition, they are independent of grammar rules and more suitable to the data streaming paradigm followed by Twitter. To evaluate the effectiveness of FS-NER, we used multi-lingual Twitter data obtained from different domains and involving diverse entity types. Our results reveal that FS-NER achieves similar recognition performance when compared to CRF-based approaches. On the other hand, in terms of computational performance, FS-NER surpassed by large the CRF-based approaches, indicating to be more practical to the Twitter environment. As future work we intend to alleviate the dependence on manually annotated data, automate the process of filter combination and identify other application environments in which FS-NER is suitable.

7. 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.

8. REFERENCES

[1] E. Amigó, J. Artiles, J. Gonzalo, D. Spina, B. Liu, and A. Corujo. WePS3 Evaluation Campaign: Overview of the On-line Reputation Management Task. In *Proc of CLEF*, 2010.

[2] G. Crane and A. Jones. The Challenge of Virginia Banks: An Evaluation of Named Entity Analysis in a 19th-Century Newspaper Collection. In *Proc. of JCDL*, pages 31–40, 2006.

[3] G. Doddington, A. Mitchell, M. Przybocki, L. Ramshaw, S. Strassel, and R. Weischedel. The Automatic Content Extraction (ACE) Program–Tasks, Data, and Evaluation. In *Proc. of LREC*, pages 837–840, 2004.

[4] A. Ekbal and S. Saha. Maximum Entropy Classifier Ensembling using Genetic Algorithm for NER in Bengali. In *Proc. of LREC*, 2010.

[5] T. Finin, W. Murnane, A. Karandikar, N. Keller, J. Martineau, and M. Dredze. Annotating named entities in Twitter data with crowdsourcing. In *Proc. of the NAACL HLT 2010 Workshop on Creating Speech and Language Data with Amazon's Mechanical Turk*, pages 80–88, 2010.

[6] 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 *Proc. of ACL (Short Papers)*, pages 42–47, 2011.

[7] L. Hong, G. Convertino, and E. H. Chi. Language Matters In Twitter: A Large Scale Study. In *Proc. of ICWSM*, 2011.

[8] W. Hua, D. T. Huynh, S. Hosseini, J. Lu, and X. Zhou. Information Extraction From Microblogs: A Survey. *Int. J. Soft. and Informatics*, 6(4):495–522, 2012.

[9] J. J. Jung. Online Named Entity Recognition Method for Microtexts in Social Networking Services: A Case Study of Twitter. *Expert Systems with Applications*, 39(9):8066–8070, 2012.

[10] C. Li, J. Weng, Q. He, Y. Yao, A. Datta, A. Sun, and B.-S. Lee. TwiNER: named entity recognition in targeted twitter stream. In *Proc. of SIGIR*, pages 721–730, 2012.

[11] X. Liu, S. Zhang, F. Wei, and M. Zhou. Recognizing Named Entities in Tweets. In *Proc. of ACL*, pages 359–367, 2011.

[12] B. Locke and J. Martin. Named Entity Recognition: Adapting to Microblogging. Technical report, University of Colorado, 2009.

[13] M. Michelson and S. A. Macskassy. Discovering Users' Topics of Interest on Twitter: a First Look. In *Proc. of the Fourth workshop on Analytics for Noisy Unstructured Text Data*, pages 73–80, Oct. 2010.

[14] D. Nadeau and S. Sekine. A Survey of Named Entity Recognition and Classification. *Linguisticae Investigationes*, 30(1):3–26, 2007.

[15] D. Ramage, D. Hall, R. Nallapati, and C. D. Manning. Labeled LDA: A Supervised Topic Model for Credit Attribution in Multi-Labeled Corpora. In *Proc. of EMNLP*, pages 248–256, 2009.

[16] A. Ritter, S. Clark, Mausam, and O. Etzioni. Named Entity Recognition in Tweets: An Experimental Study. In *Proc. of EMNLP*, pages 1524–1534, 2011.

[17] M. Rössler. Using Markov Models for Named Entity Recognition in German Newspapers. In *Proc. of the Workshop on Machine Learning Approaches in Computational Linguistics*, pages 29–37, 2002.