*Journal of Universal Computer Science, vol. 22, no. 6 (2016), 782-801*
*submitted: 30/11/15, accepted: 28/5/16, appeared: 1/6/16 © J.UCS*

# TwiSNER: Semi-supervised Method for Named Entity Recognition from Text Streams on Twitter[1]

**Van Cuong Tran**, **Dosam Hwang**[2]
(Department of Computer Engineering, Yeungnam University
Gyeongsan, Korea
{vancuongqbuni, dosamhwang}@gmail.com)

**Jason J. Jung**
(Department of Computer Engineering, Chung-Ang University
Seoul, Korea
j2jung@gmail.com)

**Abstract:** The data on Social Network Services (SNSs) has recently become an interesting source for researchers conducting different Natural Language Processing (NLP) experiments, such as sentiment analysis, information extraction, Named Entity Recognition (NER), and so on. The characteristics of SNS data are usually described as short, noisy, with insufficient supplemental information. They often contain grammatical errors, misspellings, and unreliable capitalization. Thus, standard NLP tools (e.g., NER systems) have difficulty obtaining good results when they are applied on these data, even if they perform well on well-formatted texts. Most of the traditional NER methods are based on supervised learning techniques that often require a large amount of standard training data to train a classifier. In this paper, we propose a method called TwiSNER to classify named entities in Twitter data (called tweets) by using a semi-supervised learning approach combined with the conditional random field model, hand-made rules, and the co-occurrence coefficient of the featured words surrounding entities. In the experiments, TwiSNER is applied on a dataset collected from Twitter, which includes 11,425 tweets for training with 4,716 labeled tweets and 1,450 tweets for testing. TwiSNER produces promising results, where the best F-measure is better than the baselines.

**Key Words:** Named Entity Recognition, SNS Analysis, Semi-supervised Learning
**Category:** H.4.m [Information Systems Applications]: Miscellaneous

## 1 Introduction

Named entity recognition also known as entity identification and entity extraction is a subtask of information extraction. It identifies entities in documents and classifies them into predefined categories such as person names, locations, organizations, etc [Abdallah, 12]. NER is a fundamental task and is the core

[1] This paper is significantly revised from an earlier version presented at The 2nd National Foundation for Science and Technology Development (NAFOSTED) Conference on Information and Computer Science (NICS'2015) in September 2015
[2] Corresponding author

of NLP systems. The extracted named entities can be utilized for various purposes such as entity relation extraction, document summarization [Nobata, 02], speech recognition [Meyer, 06], and term indexing in information retrieval systems [Chen, 98][Nguyen, 15].

NER systems include two tasks: first, identify the proper nouns in the document, and second, classify these proper nouns into a set of predefined categories of interest. The methods for current NER systems can be classified into three categories: i) hand-made rule-based systems; ii) machine learning approaches; iii) hybrid methods [Mansouri, 08]. First, hand-made rule-based systems use a set of rules created to extract patterns. The patterns mostly comprise grammatical, syntactic, and orthographic features in combination with a list of dictionaries that are manually pre-defined by humans [Alfred, 14][Van, 14]. Second, machine learning approaches normally use machine learning techniques to identify patterns and classify them into particular predefined classes, such as person, location, organization, etc. [Finkel, 09].

There are three categories of machine learning approaches for NER system. The supervised learning approach uses an algorithm that can learn to classify a given set of annotated data to produce classifiers, which can be applied NER task to new data. This method requires a large annotated training data and high accuracy to construct a statistical model for classifications. Typical models are the hidden Markov model, maximum entropy, and Conditional Random Field (CRF). Unsupervised learning is another machine learning method without any feedback. This approach does not need any annotated training data; the relation of objects in the unlabeled data will be found in order to construct classifiers. Semi-supervised learning (SSL) is a machine learning approach that utilizes a small amount of labeled data with a large amount of unlabeled data for the training phase. SSL falls between unsupervised learning and supervised learning. Third, hybrid methods combine the two above-mentioned methods with several NLP techniques to get better results.

Popular NER approaches are machine learning that uses either linguistic process techniques or statistical models, or an extension of famous NER approaches combined with knowledge base sources such as Wikipedia, Freebase, Wordnet, gazetteers, and so on. Machine learning approaches achieve high performance if they are applied to well-formatted text with proper sentences in terms of grammar and lexicons. However, the achievement results are not as expected when we apply these systems to short and noisy messages, such as tweets from Twitter. For example, the performance of the Stanford NER that uses the CRF model to train a classifier for the CoNLL03 data dropped from 90.8% to 45.8% when it was applied to tweets [Liu, 11]. This was caused by the characteristics of tweets, which include being short, informal, ungrammatical, noisy, and lacking in context. The length of a tweet is 140 characters at most, and tweets contain different kinds of information, such as text, hyperlinks, user mentions (e.g., @BrackObama) and hashtags (e.g., #NewYork). In addition, SNS users often post tweets with a free style and acronyms (e.g., NY for New York) and do not include extra information to explain the author's opinion. Another challenge for systems is the large volume and the dynamic content in terms of time. Currently, Twitter has more than 316 million monthly active users and 500 million tweets are sent per day[3]. The data from Twitter could be fed into processing systems as a stream of data.

In this paper, we propose a method, called TwiSNER, to recognize named entities in tweets by using an SSL approach combined with statistical models and hand-made rules. The SSL method can utilize unlabeled data for the training phase to compensate for the lack of labeled data and to improve the performance of supervised methods by iteratively adding self-labeled samples. In addition, we also use a richer linguistic context of linked websites in tweets to support identifying the named entities in them. In the initial phase, a cosine similarity measurement is applied to cluster unlabeled tweets in the training data into corresponding groups based on content. An available classifier is applied to categorize the named entities of hyperlink content embedded in tweets. These named entities are mapped into the tweets to overcome the lack of context. In the classification phase, the iterations will process the sequence of clusters. Initially, a CRF model is used to train a classifier based on the labeled tweets. The unlabeled tweets will be classified by both classifier and a set of hand-made rules. With each featured word in the labeled tweets that occurs around the specific named entity, a statistical method is utilized to calculate the co-occurrence coefficient toward that entity category. The named entity candidates in tweets are continuously examined as to whether they are a named entity, based on the average value of the co-occurrence coefficient of the featured words that occur around them. The entity label is decided based on the highest score of the entity category. To deal with the issues of shortage of content and lack of context information in tweets that do not have classified entities, the same proper nouns in one cluster are considered in order to classify in the same category. Finally, the labeled tweets of each cluster are added to the labeled training data to retrain the classifier.

In order to evaluate our proposal and show how the system works, we evaluated TwiSNER with a training data including 11,425 tweets, in which 4,716 tweets were manually labeled. The test tweets were obtained from a test set in Making Sense of Micro Posts (#MSM2013) which provided 1,450 tweets. Experimental results show that our model achieves good results.

The contributions of our method are summarized as follows:

- We propose a method called TwiSNER that combines the statistical models

[3] https://about.twitter.com/company, accessed 2015/9/28

and hand-made rules into a semi-supervised learning approach for NER on short text.

– To deal with the lack of context information about the data from Twitter, supplemental information on the tweets from hyperlinks in the tweets is used to support identifying the named entity. On the other hand, we assign the label of the named entity candidates based on the similarity of proper nouns in the same cluster.

– We also propose a statistical approach for classifying named entities based on the co-occurrence coefficient of featured words surrounding the named entity to overcome the weakness of informal text.

– We evaluated our system on the #MSM2013 test set and showed that our proposal outperforms the baselines.

The outline of this paper is as follows. Sections 2 and 3 present related works and basic notions. Section 4 describes our method for classifying named entities. Sections 5 and 6 show the experimental results of our system and conclude the paper, discussing issues for future work.

## 2 Related Works

Named entity recognition on Twitter is a hard challenge that has attracted more interest from researchers in recent years, and they have many applications in data mining. The first work that we want to mention here was contributed by [Ritter, 11]. They rebuilt the NLP tool beginning with part-of-speech tagging. The NER method leverages the redundancy inherent in tweets to achieve high performance by using labeled latent Dirichlet allocation to exploit Freebase dictionaries in a semi-supervised learning approach. Another approach is described by [Jung, 12], who proposed three heuristics (i.e., temporal association, social association, semantic association) of contextual association among the microtexts to discover contextual clusters in them. Instead of examining an entire dataset, the NER system is applied to each microtext cluster. As a case study, the author applied the proposed method on Twitter by using a maximum entropy approach-based method, which provided 90.3% precision as the best result.

[Liu, 11] proposed combining the K-Nearest Neighbors (KNN) algorithm with a linear CRF model in a semi-supervised approach. The general idea was to use the KNN model to classify tweets in a lexicon level first, and then apply the CRF model in order to execute a fine-grained tweet-level NER over the results obtained by the KNN algorithm. Finally, 30 gazetteers, which cover common names, countries, locations, and temporal expressions were used to compensate for the lack of training data.

Named entities tend to occur in multiple similar tweets, and it is easy to identify them for some tweets. [Liu, 13] describe a two-stage labeling system to harvest the redundancy for multiple similar tweets. First, a sequence tagger based on the CRF model labels each tweet. Then it clusters tweets to put similar tweets into the same groups. Finally, using an enhanced CRF model to refine the labels of each tweet. [Li, 12] also proposed a novel two-step unsupervised NER approach to recognize named entities in Twitter data, called TwinNER. Based on the gregarious properties of the named entities in a targeted tweets stream, for the first step, it leverages global context obtained from Wikipedia and a Web N-Gram corpus to partition tweets into valid segments that are the named entity candidates. In the second step, TwinNER constructs a random walk model to exploit the gregarious properties in the local context derived from the Twitter stream. The highly-ranked segments have a high opportunity to be true named entities. This approach deals with streams, however, it does not determine the class of the identified entity, but only determines if a phrase is an entity or not.

With experimentations that use different sources of distant supervision to guide unsupervised and semi-supervised adaptation of a part of speech (POS) and NER, [Plank, 14] proposed a semi-supervised approach to POS tagging and NER for Twitter data. They used the dictionaries and linked websites as a source of not-so-distant supervision to guide the bootstrapping. The content of URLs in tweets provides richer linguistic context than that available in the tweets themselves, and we can correct POS tagging and get the context of the tweet based on URL context. Their proposal does not require additional labeled in-domain data to correct for sample bias, but rather leverages pools of unlabeled Twitter data. Their results outperformed off-the-shelf taggers when evaluated across various datasets, and achieved average error reductions across the dataset of 5% on POS tagging and 10% on NER, compared to state-of-the-art baselines.

In [Liao, 09], they proposed a semi-supervised learning algorithm for NER on documents by using a CRF model. The algorithm repetitively learned to improve the training data and the feature set from a small amount of gold data. The trained model is used to extract high-confidence data, which then discovers low-confidence data by using other independent features. These low-confidence data are then added to the training data to retrain the model. They give two ways to obtain independent evidence for entities. Another rule-based NER approach was proposed by [Alfred, 14]. Their system is designed based on Malay POS tagging features and contextual features that implement handling of Malay articles. With the POS results, proper nouns are identified or detected as the possible candidates for annotation, as well as symbols and conjunctions that are also to be considered in the process of identifying named entities. The dictionaries of three named-entity categories are utilized to recognize the named entities that are not identified by the predefined rules. Their experimental results show a reasonable output of 89.47% for the F-measure, in which the recall was 94.44% and the precision was 85%.

In this paper, we propose an SSL method to extract named entities from tweets on Twitter. Our system combines a hand-made rule-based classifier and statistic models into a semi-supervised learning method. We believe that our proposed method is an effective way to solve the NER task on Twitter.

## 3 Basic Notions

### 3.1 Tweet

Twitter is a free social networking microblogging service that allows registered members to broadcast short posts, called tweets. The maximum length of a tweet is 140 characters and it is a sequence of tokens defined as follows:

$$tw = (tk_i : i = 1, ..., n) \tag{1}$$

where $tk_i$ is a token in $tw$, and $n$ is the number of tokens of tweet $tw$.

An example of a tweet is:

*What keeps #Chicago small biz owners up at night? They tell @crainschicago: http://t.co/pgpTwYr6bE*

where tokens begin with the "#" character, like *#Chicago*, which is a hashtag usually used to mark keywords or denote one of the topics of a discussion. The hashtags can be used as pure metadata or serve as both a word and metadata. The "@" character followed by a username, like *@crainschicago*, is used for mentioning or replying to other users. The tokens that begin with "http://", like *http://t.co/pgpTwYr6bE*, are shortened web links.

### 3.2 Hand-made Rule

The form of the rule is

$$\mathrm{A} \mid \mathrm{B} \mid \mathrm{C} \rightarrow \mathrm{D}$$

where A is the left-hand side of the considered noun (possibly empty); B is the considered noun; C is the right-hand side of the considered noun (possibly empty); and D is the entity category.

The operators and symbols are utilized in the rule as follows:

- The comparison operators include: $=$, $!=$, $<$, $\leq$, $>$, $\geq$, and logical operators include $\wedge$ (and), $\vee$ (or).

- The wildcard characters are: *, ?. The asterisk (*) represents zero or more characters in a string of characters. The question mark (?) presents any one character.

- "norm" is the normalized form of the word.

- "POStag" is the part of speech to tag the tokens.

- "isAllCap" is a function that checks whether all characters in the word are capitals. Its value is *True* if all characters in the word are capitals and *False* otherwise.

- "isCap" is a function that checks whether the first letter in the word is a capital. Its value is *True* if the first letter in the word is a capital, and *False* otherwise.

- "length" is the number of characters in the token.

- "category" is the category prediction of the proper noun, where context satisfies the rule's condition.

*Example 1.*

*Rule:* [norm = "read"] [*] [norm = "by"] | [POStag = "NNPs"] | ⇒ [category = "Person"]

*Tweet:* LOVE reading Saunders.10 Stories to ***Read*** for Free Online ***by*** **George Saunders**$_{Person}$ http://t.co/Ku5LfrAz4r via

*Example 2.*

*Rule:* [Category = "Location"] [token = ","] | [POStag = "NNP" ∧ isAllCap ∧ length<4] | ⇒ [category = "Location"]

*Tweet:* Our 42nd Fellowship group in ***Darlington,*** **WI**$_{Location}$ is recruiting youths & adults interested in our traditional program. DM for info.

*Example 3.*

*Rule:* | [POStag = "NNPs"] | [norm = "city"] ⇒ [category = "Location"]

Applying the similar for the words {river, mountain, forest, street, road, place, square, ...}

### 3.3 Cosine Similarity for Clustering

Cosine similarity is a measure of similarity between two vectors of an inner product space. In document representation, each term of the text is assigned a different dimension and a document is characterized by a vector where the value of each dimension corresponds to the number of times that term appears in the document [Hong, 14][Tran, 15]. A document represents a data point in d-dimensional space, where d is the size of the word vocabulary in the corpus. Cosine similarity is a useful measure for calculating the similarity of two documents in terms of subject matter.

The cosine of two vectors can be derived by using the Euclidean dot product formula [Sidorov, 14]:

$$A \cdot B = ||A||||B||cos(\theta) \tag{2}$$

Given two vectors with attributes A and B, cosine similarity $cos(\theta)$ is represented using a dot product and magnitude as follows:

$$sim(A,B) = cos(\theta) = \frac{A \cdot B}{\|A\|\|B\|} = \frac{\sum_{i=1}^{n} A_i \times B_i}{\sqrt{\sum_{i=1}^{n}(A_i)^2} \times \sqrt{\sum_{i=1}^{n}(B_i)^2}} \quad (3)$$

where $\|A\|$ and $\|B\|$ denote the norm of vectors A and B, respectively.

To group documents into meaningful clusters, we can utilize equation (3) to measure textual similarity between two documents. It means that this measure will capture the similarity between the textual vectors.

A cluster is characterized by the vector of a cluster centroid, and the number of documents in the cluster is defined as follows [Yin, 13]:

- Textual centroid $Cent_i$ is a vector in which each element represents the average weight of the corresponding words for all documents in cluster $C_i$.
- Cluster size $|C_i|$ is defined as the number of documents belonging to cluster $C_i$.

This method does not need to decide the number of clusters in advance. Initially, a document is chosen to form a cluster, and repeats with each document are calculated by the similarity between that document and any existing cluster, as seen in Equation (3). Document $d$ is assigned to a cluster whenever i) the similarity value between $d$ and the centroid of the cluster is the maximum, compared from its distance to other clusters, and ii) the similarity value is greater than the predefined similarity threshold $\gamma$. Otherwise, a new cluster will be created to contain $d$. Once a new document $d$ is added to cluster $C_i$, the cluster information will be updated by the following equations:

$$Cent_i = \frac{Cent_i \times |C_i| + d}{|Cent_i| + 1} \quad (4)$$

$$|C_i| = |C_i| + 1 \quad (5)$$

## 4 TwiSNER System

### 4.1 Overview

The workflow of TwiSNER is illustrated in Figure 1, and the algorithm for the training phase is described in detail in Algorithm 1. It inherits the idea of Lioa and Veeramachaneni to find new labeled data for the training data from

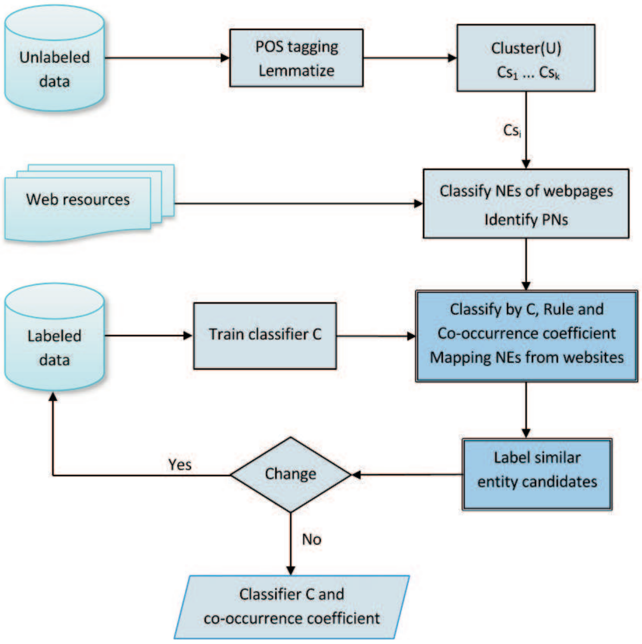


Figure 1: The workflow of TwiSNER system on Twitter for the training phase

unlabeled data based on a SSL approach [Liao, 09]. Assume that we have a training data distributed into two parts: a small set of manually labeled tweets and a bigger set of unlabeled tweets, which are denoted as $L$ and $U$, respectively. We manually construct a set of rules for identifying named entities based on grammatical, syntactic, and orthographic features and the writing style of social media messages. These rules are denoted by $R$. We also use a classifier of three classes trained on the mixing of CoNLL and MUC corpora. This classifier is utilized to classify named entities for the hyperlink content and classify named entities in unlabeled tweets along with the classifier trained on labeled dataset in each iteration. Cosine similarity is used to cluster tweets into groups of similar tweets in terms of content. The classification process deals with each cluster in each iteration. Initially, the labeled training data is used to train classifier $C$ based on a CRF model. In our work, we use the CRF framework provided by Stanford[4]. The classifier $C$ is used to classify the unlabeled tweets of each cluster. This classifier obtained high precision, but its recall is low, so the data are continually classified by other methods to improve the system's recall. Rule-based NER is a suitable method to assign the entity's label. We apply the set

[4] http://nlp.stanford.edu/software/CRF-NER.shtml

**Algorithm 1** Algorithm for training phase

**Input:** $L$ - labeled data
$U$ - unlabeled tweet data
$R$ - set of rules
**Output:** $C$ - classifier
$\delta$ - co-occurrence coefficient

Tag POS and Lemmatize tweets
Get proper nouns (PNs), mentions and content of hyperlinks from the web pages embedded in tweets
Use the CRF model to train classifier $C$ based on $L$
Classify named entity of hyperlink content
$Cs \leftarrow$ Cluster($U$)
**repeat**
  **for** each cluster $Cs_i$ in $Cs$ **do**
    Classify $Cs_i$ by $C$, $R$
    Mapping named entities from hyperlink content to tweet (if any)
    Calculate the co-occurrence coefficient $\delta$ of featured words
    Classify $Cs_i$ by the co-occurrence coefficient $\delta$
    Assign similar named entity candidates to have the same label
    Add $Cs_i$ to $L$
  **end for**
  Use the CRF model to train classifier $C$ based on $L$
**until** no new named entity can be identified
**return** $C$, $\delta$

of rules that was manually constructed, as described in Section 3.2, to classify entities.

Twitter users often use the hyperlinks to indicate detailed information on what they mentioned in their tweets [Plank, 14]. The content of the webpages is often from a newswire that provides more context and that is written in a more canonical language, so they are classified for the named entities more easily than tweets, and the accuracy is better. We can use the available classifiers trained on well-formatted text (e.g, trained on the mixing of CoLL and MUC corpora) to classify the named entities of these sources. And then, the named entities from the webpage are mapped into tweets, if any.

Microposts such as tweet data are often informal texts. So, the sequence models and rule-based systems miss a lot of named entities when they are applied to these data. In order to overcome the limitations of these issues, we propose an approach to classify named entities based on the featured words located

**Algorithm 2** Algorithm for testing phase

**Input:** $U$ - unlabeled data
$R$ - set of rules
$C$ - classifier
$\delta$ - co-occurrence coefficient
**Output:** $L$ - labeled data

**for** each tweet $tw \in U$ **do**
    POS($tw$)
    Get PNs, mention in $tw$
    Classify $tw$ by $C$, $R$
    Classify $tw$ by the co-occurrence coefficient $\delta$
**end for**
**return** $L$

around each entity category. A statistical model is applied to the labeled tweets to calculate the frequency of appearance and the impact of the featured words toward each entity category. For each proper noun, we consider it a named entity candidate, and we measure the average co-occurrence coefficient value of the featured words that occur around the named entity candidate. The average co-occurrence coefficient value will decide whether that proper noun is a named entity, and what the entity category is.

Dealing with the short length and the lack of context information of tweets, the cluster phase put the tweets that have similar content into the same group. Thus, similar proper nouns in different tweets are automatically assigned the same label for detected entities, if any. This solution can improve the accuracy of the classifier during the next iteration. Finally, the labeled tweets of the cluster are added to the labeled training data to retrain the classifier. The number of iterations of the model depends on the detection results. If there are no more named entities detected, the training process finishes.

### 4.2 Features

A POS tagger assigns a label for the role of the words or tokens (e.g., noun, verb, adjective, etc.) and a lemmatizer determines the lemma for given words (i.e., we use publicly available tools developed by Stanford[5]). The proper nouns are extracted based on the results of a POS tagger. In our work, we only focus on words that are labeled by NNP, NNPS, NN, and NNS. An example of the POS tagging derived from the POS tagger is as follows.

Original text: *I'm at Bicycle Ranch in Scottsdale, AZ.*

[5] http://nlp.stanford.edu/software/

| Tag | Description |
|---|---|
| DT | Determiner |
| PRP | Personal pronoun |
| VB | Verb, base form |
| VBP | Verb, non-3rd person singular present |
| IN | Preposition or subordinating conjunction |
| NN | Noun, singular or mass |
| NNS | Noun, plural |
| NNP | Proper noun, singular |
| NNPS | Proper noun, plural |

**Table 1:** Some tags of POS tagging

POS tagging text: *[I/PRP, 'm/VBP, at/IN, Bicycle/NNP, Ranch/NNP, in/IN, Scottsdale/NNP, ,/,, AZ/NNP, ./.]*

In this example, we can extract the following proper nouns: *Bicycle Ranch*, *Scottsdale*, and *AZ* based on the POS tagging.

To extract the featured words, the mentions and hyperlinks are separated; meanwhile, symbols and hashtags are removed from tweets [Nguyen, 15]. Stop words are also removed by using a list of predefined stop words[6]. In addition, some kinds of POS tagging are not considered, such as coordinating conjunctions, cardinals, determiners, symbols, modals, etc.

To represent textual information in tweets, we use a vector of the bag of featured words. A tweet $tw$ is represented by a vector of the featured words where its dimension is equal to the number of featured words in the training data:

$$tw = (w_1, w_2, ..., w_n) \tag{6}$$

where $w_i = 1$ if the word at the $i^{th}$ position in the list of featured words occurs in the tweet $tw$; otherwise, $w_i = 0$; $n$ is the quantity of the featured words in the training data.

### 4.3 Co-occurrence Efficient Model

The most typical characteristic of microposts in SNSs is that they are usually informal in nature. These are the situations where the sequence NER models do

[6] http://www.textfixer.com/resources/common-english-words.txt

| Tweet | Feature words |
|---|---|
| @FakeSportsCentr yes cuz $Chicago_L$ is a border city #ignorant | yes cuz Chicago be border city |
| Man extradited to $U.S._L$ from $Mexico_L$ over slaying of border patrol agent - http://t.co/g544XWzdXM | man extradite U.S. Mexico slay border patrol agent |
| Leaked photos show immigrant children packed in crowded $Texas_L$ border facilities - http://t.co/t3ybJj7OXY | leak photo show immigrant children pack crowd Texas border facility |
| This is the worst thing to happen to $Chicago_L$ since border patrol | be worst thing happen Chicago border patrol |

**Table 2:** Examples of tweets and their featured words

not achieve high performance when they are applied to these texts. To overcome the weakness of informal text, we only consider the featured words instead of considering all of the words in the sequence. With each named entity $x_i$, we examine the previous featured words and the next featured words of the entity in the consideration window $(x_{i-3}, x_{i-2}, x_{i-1}, x_i, x_{i+1}, x_{i+2}, x_{i+3})$. The number of featured words in the consideration window is seven, with the named entity located in the middle of the window. The co-occurrence coefficient of each featured word toward an entity category is calculated as follows.

$$\delta_x^{\varepsilon} = \frac{\sum_{i=1}^{k} \frac{1}{\lambda_i}}{n} \tag{7}$$

where $\varepsilon$ is the entity category (person, location, organization, etc.); $x$ is a featured word in the current consideration window; $\lambda$ is the distance from $x$ to the named entity (i.e., it is the number of featured words from $x$ to the named entity); $k$ is the number of tweets that contain featured word $x$ together with entity category $\varepsilon$; $n$ is the number of the featured words $x$ that are contained in the labeled data.

*Example 3.* Assume that we have 10 tweets that contain the word *border*, in which there are four tweets that contain entity category Location as described in Table 4.3. These named entities and the featured word *border* satisfy the condition of the featured word window in the above definition. Apply Equation (7) to calculate the co-occurrence coefficient of the featured word *border* toward the entity category Location for those four tweets.

$$\delta_{border}^{Location} = \frac{\frac{1}{2} + \frac{1}{2} + \frac{1}{1} + \frac{1}{1}}{10} = 0.3$$

The entity candidates are classified based on the co-occurrence coefficient of featured words in the consideration window. With each proper noun that has not been classified as a named entity yet, the average co-occurrence coefficient value between the featured words and the named entity candidate is calculated according to each entity category as follows:

$$\Psi_X^{\varepsilon} = \frac{\sum_{j=1}^{m} \delta_{x_j}^{\varepsilon}}{m} \tag{8}$$

where $X$ is a named entity candidate, and $m$ is the number of featured words in the consideration window.

The category of the entity candidate is decided depending on the value of $\Psi$ and $\varepsilon$. If the value $\Psi$ is greater than threshold $\alpha$, then the named entity candidate $X$ will be classified into entity category $\varepsilon$.

Assume that we have a tweet "*Do you have to cross a border to go to Scotland? - Spencer the droman*". In this tweet, *Scotland* is a proper noun and is considered a named entity candidate. The entity category of *Scotland* is decided according to the average co-occurrence coefficient value of the featured words in the consideration window. We utilize the values in Example 3 and apply Equation (8) to the proper noun *Scotland* according to three entity categories: Person, Location, and Organization.

$$\Psi_{Scotland}^{Person} = \frac{\delta_{cross}^{Person} + \delta_{border}^{Person} + \delta_{go}^{Person}}{3}$$

$$\Psi_{Scotland}^{Location} = \frac{\delta_{cross}^{Location} + \delta_{border}^{Location} + \delta_{go}^{Location}}{3}$$

$$\Psi_{Scotland}^{Organization} = \frac{\delta_{cross}^{Organization} + \delta_{border}^{Organization} + \delta_{go}^{Organization}}{3}$$

Assuming that the highest value of three above values is $\Psi_{Scotland}^{Location}$ and this value is greater than the threshold $\alpha$, then the proper noun *Scotland* is classified into the entity category Location.

In an SNS, users often utilize mentions in their post to refer to another user by using a simple syntax "@username" such as *@BarackObama*, *@TheDemocrats*, *@Chicago_Reader*, etc. There is a high probability that an account name of a user contains the user's real name or casual name. So, if an entity candidate of a tweet occurs in their mentions, it has high potential to be a named entity. In some case, the values of an entity category are calculated by Equation (8) are not enough to decide the entity category according to the selected threshold $\alpha$, even though it is a named entity. A solution is proposed whereby these values are increased a certain $\beta$ times. For example, we have a tweet:

*@ArianaGrande Hey Ari Would you mind retweeting the Tweet I Retweeted to Get @justinbieber To Follow #oomf*

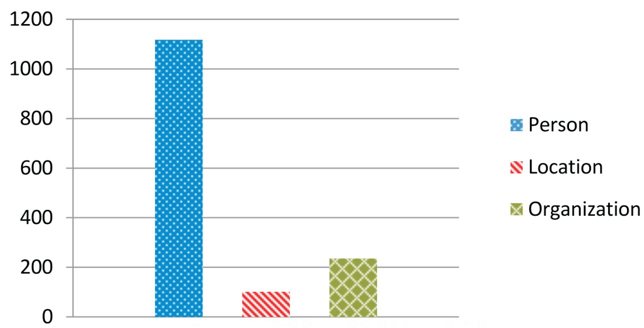


**Figure 2:** The distribution of the entity categories in the test set

Suppose that we set threshold $\alpha = 0.3$, $\beta = 3$, and the average co-occurrence coefficient values of featured words surrounding the proper noun $Ari$ are calculated as follows:

$$\Psi_{Ari}^{Person} = 0.114; \quad \Psi_{Ari}^{Location} = 0.034; \quad \Psi_{Ari}^{Organization} = 0.002$$

All three above values do not satisfy the threshold $\alpha$ for classifying the entity $Ari$. Because proper noun $Ari$ appears in the mention of $@ArianaGrande$, it has high probability to be a named entity so these values are multiplied by three. In these cases, $\Psi_{Ari}^{Person}$ is a highest value because its final value is $0.114 * 3 = 0.342$. This value is greater than threshold $\alpha$ for classifying the named entity, thus proper noun $Ari$ is classified into the entity category Person.

## 5 Experimental Results

### 5.1 Dataset

To evaluate the performance of our proposal, we apply the TwiSNER system to the tweet dataset. The training data consists of tweets from the training data of the #MSM2013 challenge[7] and was collected from Twitter by using the public Java library for the Twitter API[8]. We drop tweets that are only hashtags, mentions, hyperlinks or emoticons, etc., and finally, we have 11,425 tweets for the training phase. The test set ($TS$) includes 1,450 tweets from the #MSM2013 challenge as the gold standard ($GS$) to assess performance.

The dataset is annotated with three named entity categories: Person, Location, Organization. If a named entity does not belong to three of the above categories; it is labeled as Other.

[7] http://oak.dcs.shef.ac.uk/msm2013/ie_challenge/
[8] http://twitter4j.org

Each token in the labeled data is marked </Type>, where "Type" is the entity category[9]. The entity distribution over the test set is represented in Figure 2.

### 5.2 Evaluation Measures

The performance of this task is calculated following #MSM2013's measures [Basave, 13]. Precision ($P$), Recall ($R$), and F-measure ($F_1$) are calculated for each entity category, and the final results for overall entity categories are the average performance of three defined categories.

With each named entity $t$ in the $TS$, we perform strict matching between the output results and the $GS$. Each entity is presented in the tuples (entity value, entity category). Let $(x, y) \in TS_t$ denote the set of tuples for an entity category $t$ extracted by TwiSNER and $(x, y) \in GS_t$ denotes the set of tuples for an entity category $t$ in the $GS$. The set of True Positives ($TP$), False Positives ($FP$) and False Negatives ($FN$) are defined as follows:

$$TP_t = \{(x, y)|(x, y) \in TS_t \cap GS_t\} \tag{9}$$

$$FP_t = \{(x, y)|(x, y) \in TS_t \wedge (x, y) \notin GS_t\} \tag{10}$$

$$FN_t = \{(x, y)|(x, y) \notin TS_t \wedge (x, y) \in GS_t\} \tag{11}$$

where $TP_t, FP_t, FN_t$ are the set of true positives, false positives, and false negatives for entity category t, respectively. We compare the set of tuples $(x, y)$ of the output results with the set of tuples $(x, y)$ of the $GS$ based on the strict matches for both detection of the correct entity value ($x$) and the correct entity category ($y$).

Precision ($P_t$) and Recall ($R_t$) for a given entity category $t$ are defined as follows:

$$P_t = \frac{|TP_t|}{|TP_t| + |FP_t|} \tag{12}$$

$$R_t = \frac{|TP_t|}{|TP_t| + |FN_t|} \tag{13}$$

Precision ($\overline{P}$) and Recall ($\overline{R}$) of the overall entity categories are the average value for all entity categories, and we combine them into F-measure ($F_1$) which is defined as follows:

$$F_1 = 2 \times \frac{\overline{P} \times \overline{R}}{\overline{P} + \overline{R}} \tag{14}$$

[9] http://nlp.stanford.edu/software/crf-faq.shtml

### 5.3 Baselines

Two systems of the #MSM2013 competition were used as the baselines: one is Das's system [Das, 13] and the other is Genc's system [Genc, 13]. These two systems are called $SRI-JU$ and $SIT$, respectively. #MSM2013 challenge restricts the classification to four entity categories (i.e., Person, Location, Organization and Misc), in our work, we only consider three entity categories (i.e., Person, Location and Organization). The baseline results are extracted from [Basave, 13].

$SRI - JU$ is a CRF-based system. They apply a method for identification and classification of named entities based on the features (i.e., capitalization, out of vocabulary words, and gazetteers). The gazetteers of location category include 220 country names and 100 popular city names. Samsad & NICTA dictionary is used to augment for the cases out of vocabulary words. The system is examined with some combinations of features and the best result is obtained from the combination of all features.

$SIT$ system leverages the content of Wikipedia articles to classify the entity candidates in tweets. This system has two main stages. First, it recognizes the candidate concepts parts-of-tweets that may be valid entities in the tweet, and then these candidates are classified into named entity categories. The candidate concepts are identified by mapping tweets to Wikipedia pages, and then the networks of these concepts in Wikipedia are utilized for filtering and classifying named entity categories.

### 5.4 Results

The results for each entity category of TwiSNER and the baseline systems are shown in Tables 5.4 to 5.4. In terms of precision, SRI-JU achieves good performance for location and organization categories, and the best for all entity categories. TwiSNER only obtains the highest performance for the person category. There is 2.79% results that are correct entity category but they are incorrect entity value. For example, for this tweet "*Give your San Francisco Santa (John Toomey) his job back*", our method mistakenly labels *San Francisco Santa* as a Location instead of *San Francisco*. There is 3.0% results that are incorrect entity category when they are matched with the values in the $GS$. For example, the system identifies *Chicago* as a Location instead of Organization in the tweet "*Welcome to the game, Chicago. Glad you decided to wake up in the 4th quarter*".

TwiSNER obtains the highest recall score for overall entities (i.e., it is better than SRI-JU by 5.8% and it is better than SIT by 12.5%). It means that TwiSNER can find more entities than the others.

The overall assessment of classification performance is calculated by combining the precision and the recall score together using the F-measure score.

| System | Person | Location | Organization | All |
|---|---|---|---|---|
| $SRI-JU$ | 80.9 | **74.6** | **70.7** | **75.4** |
| $SIT$ | 76.5 | 71.1 | 67.4 | 71.7 |
| $TwiSNER$ | **87.4** | 58.9 | 61.5 | 69.3 |

**Table 3:** The precision scores over the different concept types (%)

| System | Person | Location | Organization | All |
|---|---|---|---|---|
| $SRI-JU$ | **87.7** | 51.8 | 24.8 | 54.8 |
| $SIT$ | 86.4 | 69.2 | 29 | 61.5 |
| $TwiSNER$ | 86.9 | **73** | **42.1** | **67.3** |

**Table 4:** The recall scores over the different concept types (%)

| System | Person | Location | Organization | All |
|---|---|---|---|---|
| $SRI-JU$ | 84.6 | 61.6 | 36.7 | 63.5 |
| $SIT$ | 81.5 | **70.5** | 40.5 | 66.2 |
| $TwiSNER$ | **87.1** | 65.2 | **50** | **68.3** |

**Table 5:** The F-measure scores over the different concept types (%)

Table 5.4 presents the F-measure score across three entity categories. TwiSNER significantly outperforms SIT and SRI-JU (i.e., 4.8% and 2.1%, respectively) in terms of F-measure. The best TwiSNER F-measure is 68.3%. Although SRI-JU achieves the highest precision, its performance is the lowest overall from among the entities.

## 6 Conclusion

In this paper, we propose an NER method called TwiSNER that combines statistical models based on featured words and hand-made rules to classify named entities from tweets. This is a semi-supervised learning approach. TwiSNER does not require a large number of manually labeled tweets. The experimental results show that our proposed method achieves high performance with only a small amount of labeled training data and obtains an F-measure better than the baselines.

In the future, we will improve our proposed method by using extra knowledge base information, such as gazetteers, Freebase, and so on. In addition, we also plan to extend TwiSNER to many entity categories.

## Acknowledgment

This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIP) (NRF-2014R1A2A2A05007154). Also, this work was supported by the 2014 Yeungnam University Research Grant and supported by the BK21+ program of the National Research Foundation (NRF) of Korea.

## References

[Abdallah, 12] Abdallah, S., Shaalan, K., Shoaib, M.: Integrating rule-based system with classification for Arabic named entity recognition, Computational Linguistics and Intelligent Text Processing, Springer Berlin Heidelberg (2012), 311-322.

[Alfred, 14] Alfred, R., Leong, L. C., On, C. K., Anthony, P.: Malay Named Entity Recognition Based on Rule-Based Approach, International Journal of Machine Learning and Computing, 4(3) (2014), 300-306.

[Basave, 13] Basave, A. E. C., Varga, A., Rowe, M., Stankovic, M., Dadzie, A. S.: Making Sense of Microposts (#MSM2013) Concept Extraction Challenge, Proc. MSM2013, 1-15.

[Chen, 98] Chen, H. H., Ding, Y. W., Tsai, S. C.: Named entity extraction for information retrieval, Computer Processing of Oriental Languages, 12(1) (1998), 75-85.

[Das, 13] Das, A., Burman, U., Balamurali, A. R., Bandyopadhyay, S.: NER from Tweets: SRI-JU System@ MSM 2013, Making Sense of Microposts (2013), 62–66.

[Finkel, 09] Finkel, J. R., Manning, C. D.: "Nested named entity recognition"; Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing, Volume 1-Volume 1 (Aug 2009), 141-150.

[Genc, 13] Genc, Y., Mason, W. A., Nickerson, J. V.: Classifying Short Messages using Collaborative Knowledge Bases: Reading Wikipedia to Understand Twitter, MSM (2013), 50-53.

[Hong, 14] Hong, T. P., Liou, Y. L., Wang, S. L., & Vo, B.: Feature selection and replacement by clustering attributes, Vietnam Journal of Computer Science, 1 (1) (2014), 47-55.

[Jung, 12] Jung, J. J.: Online named entity recognition method for microtexts in social networking services: A case study of twitter, Expert Systems with Applications, 39(9) (2012), 8066-8070.

[Li, 12] Li, C., Weng, J., He, Q., Yao, Y., Datta, A., Sun, A., Lee, B. S.: Twiner: named entity recognition in targeted twitter stream, Proceedings of the 35th international ACM SIGIR conference on Research and development in information retrieval, ACM (Aug 2012), 721-730.

[Liao, 09] Liao, W., Veeramachaneni, S.: A simple semi-supervised algorithm for named entity recognition, Proceedings of the NAACL HLT 2009 Workshop on Semi-Supervised Learning for Natural Language Processing (Jun 2009), 58-65.

[Liu, 13] Liu, X., Zhou, M.: Two-stage NER for tweets with clustering, Information Processing Management, 49(1) (2013), 264-273.

[Liu, 11] Liu, X., Zhang, S., Wei, F., Zhou, M.: Recognizing named entities in tweets, Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1 (Jun 2011), 359-367.

[Mansouri, 08] Mansouri, A., Affendey, L. S., Mamat, A.: Named entity recognition approaches, International Journal of Computer Science and Network Security, 8(2) (2008), 339-344.

[Meyer, 06] Meyer, C., Schramm, H.: Boosting HMM acoustic models in large vocabulary speech recognition, Speech Communication, 48(5) (2006), 532-548.

[Nguyen, 15] Nguyen, T. T., & Jung, J. J.: "Exploiting geotagged resources to spatial ranking by extending HITS algorithm"; Computer Science and Information Systems, 12(1) (2015), 185-201.

[Nguyen, 15] Nguyen, H. L., Nguyen, T. D., Hwang, D., Jung, J. J.: KELabTeam: A Statistical Approach on Figurative Language Sentiment Analysis in Twitter, Proceedings of the 9th International Workshop on Semantic Evaluation (2015), 679–683.

[Nobata, 02] Nobata, C., Sekine, S., Isahara, H., Grishman, R.: Summarization System Integrated with Named Entity Tagging and IE pattern Discovery, LREC (May 2002)

[Plank, 14] Plank, B., Hovy, D., McDonald, R., Søgaard, A.: Adapting taggers to Twitter with not-sodistant supervision, Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers, Dublin, Ireland (Aug 2014), 17831792.

[Ritter, 11] Ritter, A., Clark, S., Etzioni, O.: Named entity recognition in tweets: an experimental study, Proceedings of the Conference on Empirical Methods in Natural Language Processing (Jul 2011), 1524-1534.

[Sidorov, 14] Sidorov, G., Gelbukh, A., Gómez-Adorno, H., & Pinto, D.: Soft similarity and soft cosine measure: Similarity of features in vector space model, Computación y Sistemas (2014), 18(3), 491-504.

[Tran, 15] Tran, V. C., Hwang, D., Jung, J. J.: Semi-supervised approach based on co-occurrence coefficient for named entity recognition on Twitter, Proceedings of Information and Computer Science (NICS), 2nd National Foundation for Science and Technology Development Conference, IEEE (2015), 141-146.

[Van, 14] Van, T. T., Vo, B., & Le, B.: IMSR_PreTree: an improved algorithm for mining sequential rules based on the prefix-tree, Vietnam Journal of Computer Science, 1(2) (2014), 97-105.

[Yin, 13] Yin, J.: Clustering microtext streams for event identification, Proc. IJCNLP (2013), 719–725.