Named Entity Recognition and Hashtag Decomposition to Improve the Classification of Tweets

Belainine Billal
Univ. of Quebec in Montreal
201, President Kenney Av.
H2X 3Y7, Montreal-QC, Canada
belainine.billal@courrier.uqam.ca

Alexsandro Fonseca
Univ. of Quebec in Montreal
201, President Kenney Av.
H2X 3Y7, Montreal-QC, Canada
affonseca@gmail.com

Fatiha Sadat
Univ. of Quebec in Montreal
201, President Kenney Av.
H2X 3Y7, Montreal-QC, Canada
sadat.fatiha@uqam.ca

Abstract

In social networks services like Twitter, users are overwhelmed with huge amount of social data, most of which are short, unstructured and highly noisy. Identifying accurate information from this huge amount of data is indeed a hard task. Classification of tweets into organized form will help the user to easily access these required information. Our first contribution relates to filtering parts of speech and preprocessing this kind of highly noisy and short data. Our second contribution concerns the named entity recognition (NER) in tweets. Thus, the adaptation of existing language tools for natural languages, noisy and not accurate language tweets, is necessary. Our third contribution involves segmentation of hashtags and a semantic enrichment using a combination of relations from WordNet, which helps the performance of our classification system, including disambiguation of named entities, abbreviations and acronyms. Graph theory is used to cluster the words extracted from WordNet and tweets, based on the idea of connected components. We test our automatic classification system with four categories: politics, economy, sports and the medical field. We evaluate and compare several automatic classification systems using part or all of the items described in our contributions and found that filtering by part of speech and named entity recognition dramatically increase the classification precision to 77.3 %. Moreover, a classification system incorporating segmentation of hashtags and semantic enrichment by two relations from WordNet, synonymy and hyperonymy, increase classification precision up to 83.4 %.

1 Introduction

The automatic classification of text and the approaches for the extraction of hidden subjects have good performance when there is enough meta-information, the context is extended using knowledge from big collections, like Wikipedia (Sriram et al., 2010) or it uses meta-information from external sources such as Wikipedia (Genc et al., 2011) and include the use of lexical ontologies, like DBPedia (Cano et al., 2013). However, those approaches need online queries, what makes their performance decrease, and the extraction of knowledge from those external collections demand complex algorithms. Moreover, the use of those approaches makes the classification algorithms less general.

We present a classification method that uses offline knowledge extracted from WordNet to disambiguate and enrich information in tweets and also to group semantically connected words of tweets in order to decrease the size of our training matrix.

In Section 2 we present a review of some works on the classification of tweets and short texts in general. In Section 3 we detail the main steps in the pre-processing: tweet normalization, hashtag decomposition and named entity recognition. Section 4 presents our methodology: how WordNet is used

Proceedings of the 2nd Workshop on Noisy User-generated Text,
pages 102–111, Osaka, Japan, December 11 2016.

to disambiguate words in tweets, how we build a graph using the words in tweets and semantic relations extracted from WordNet and how connected components extracted from this graph is used to decrease the number of dimensions in our training matrix. In Section 5 we present the experiments and results obtained in the classification of tweets, in terms of precision. And in Section 6 and Section 7 we present the evaluation and conclusion, respectively.

2 Related Work

Recent works based on machine learning deal with the classification of tweets. Sriram et al. (2010) classify tweets into pre-defined generic classes, e.g. "news", "events", "advertising", etc., using information about the authors and some characteristics, such as abbreviations, words used to express opinion, etc.

Sahami and Heilman (2006) use short texts from tweets in search engines queries in order to increase the information in each tweet. However, those techniques need additional entity disambiguation approaches. For example, even though "jaguar" and "car" are close semantically, a disambiguation in the query results is needed to differentiate the results from "jaguar" as a car and as an animal. Therefore, the intervention of the user is necessary in order to guide the process of tweet expansion.

Genc et al. (2011) propose a method for the classification of tweets using Wikipedia, which calculates the semantic distance between words in a tweet and words in the description of Wikipedia pages to find the most similar page and category. Kinsella et al. (2011) analyze meta-data from objects (Amazon products, YouTube videos, etc.) to which links in tweets are pointing to determine their subjects.

Lee et al. (2011) classify tweets into 18 general categories, such as "sport", "politics", etc. They use two methods: bag of words and a classification based on a network, identifying the most influential users for each category. The number of influential users in common between already classified tweets and a tweet yet to be classified define the category of the new tweet.

Sankaranarayanan et al. (2009) build a news processing system called "TwitterStand" that identifies tweets corresponding to last news. Their objective is to reduce noise and identify tweets classes and groups of interest.

Saif et al. (2012) introduce a method based on the disambiguation of named entities. They add semantic information to the named entity identified (for example, adding "Apple" to "IPhone") and then associate negative/positive sentiment extracted from tweets. Michelson and Macskassy (2010) use Wikipedia to disambiguate, classify named entities found in tweets and identify the subjects of interest to the users and the most frequent categories related to named entities.

Since hashtags are essentials in understanding the subject of a tweet, most systems of analyze of opinions try to incorporate them in their calculations. Asur and Huberman (2010) show how to improve standard techniques of supervised classification by integration of polarity from most frequent hashtags.

Brun and Roux (2014a) show how to extract words from hashtags and use them to improve the detection of polarity in tweets. Their system represents each opinion according to the model proposed by Liu (2010) and compare a system that uses hashtag decomposition with a system that does not use it.

Almeida et al. (2016) present a supervised learning approach dedicated to the biomedical domain for supporting the production of literature on HIV using thesaurus MeSH (Medical Subject Headings).

3 Pre-processing

Our system for the classification of tweets is based on Weka[1] and uses the Twitter API to extract tweets[2].

We use Twitter API[3] to build our training and testing corpus. A language detector (Shuyo, 2010) is used to keep only tweets in English[4]. All tweets containing less than 80% of words in English, only URLs and empty tweets were deleted from the corpus.

Despite the existence of many pre-processing tools for short texts and tweets (e.g. TweetNLP[5]), we

1. http://www.cs.waikato.ac.nz/ml/weka
2. https://bitbucket.org/LyesBillal/twitterclassifier/overview
3. https://dev.twitter.com
4. https://github.com/shuyo/language-detection
5. http://www.cs.cmu.edu/ ark/TweetNLP/

have chosen Standford NLP[6] for its robustness.

In the next sections we explain the main pre-processing steps: tweet normalization, hashtag decomposition and named entity recognition.

3.1 Tweet Normalization

Tweet normalization consists in rewriting text in a standard language. It is based on the most common lexical mistakes made in social media and is divided in the following sub-tasks (Han et al., 2013):

1. Suppression of extra letters, e.g. "gooood". We use an English dictionary [7] and regular expressions to detect the closest possible correct word;

2. Minimal orthographic correction for most common mistakes, e.g. substitution of "scoll" for "scroll";

3. Substitution of common words used in social media, e.g. "2day" for "today". We use a dictionary[8] dedicated to this kind of problem.

3.2 Hashtag Decomposition

A hashtag always starts by the character "#", making easy its identification. They are usually composed words created by users and cause a problem for the linguistic analyzes because they are considered unknown words. In a tweet having only 140 characters ignoring the hashtags may cause an enormous loss of information (Brun and Roux, 2014b).

Usually, different hashtags are used for the same subject. For example, for Newsmax_Media we found:

#News_Media, #VanRE, #Vancouver...

Those three hashtags are related to the words "Newsmax", "Media", "Vancouver", "VanRE".

Our objective is to extract all words in hashtags and increase their frequency. For example:

#Newsmax_Media $\rightarrow$ (**Newsmax**, **Media**)
#News_Media $\rightarrow$ (**News**, **Media**)
#Vancouver $\rightarrow$ (**Vancouver**)
#VanRE $\rightarrow$ (**Van**, **RE**)

We propose a recursive algorithm that processes hashtags from left to right and separates the problem in three sub-tasks:

1. When each word in a hashtag start by an uppercase letter, we use a function to separate those words. Example: **#ParisClimateConference**

2. When the words are separated by special characters or by numbers, we use another function. Examples: **#3Novices**, **#Newsmax_Media**

3. When each word starts by a lowercase letter, we use a third function combined with an English dictionary[9]. This function tries to separate a hashtag in the fewest possible number of words, from left to right. For example, the hashtag **#renewableenergy** can be separated as:
#renewableenergy $\rightarrow$ (**renew**, **able**, **energy**)
#renewableenergy $\rightarrow$ (**renewable**, **energy**)
and we chose (**renewable**, **energy**), following the human tendency of choosing the longest possible sequence when decodifying a sequence of characters.

6. http://nlp.stanford.edu/software/
7. http://gdt.oqlf.gouv.qc.ca/
8. https://github.com/coastalcph/cs_sst/blob/master/data/res/emnlp_dict.txt
9. https://github.com/dwyl/english-words

3.3 Named Entity Recognition

Different users write dates, names of places or people in different ways. For example: "2016-03-10" and "March 10th 2016", "John Kennedy" and "JFK", etc. Named Entity Recognition (NER), a set of techniques to deal with this problem, is used in different projects, such as the Gene/Protein Named Entity Recognition and Normalization Software(GNAT) (Wermter et al., 2009).

Unities of time, distance, currency can be normalized using the Stanford NLP API. A sequence of words that appear with high frequency is kept because they probably represent a unique entity. For the names of location and organization we search in Wordnet for the closest synonyms.

After NER, we remove the stop words, i.e. functional words carrying no meaning[10] and make the lemmatization, i.e. transformation of words into their canonical forms (e.g. nouns from plural to singular, verbs from a conjugated to a infinitive form, etc.), using Standford NLP lemmatizer[11].

4 Methodology

Figure 1 shows the main process in our methodology.

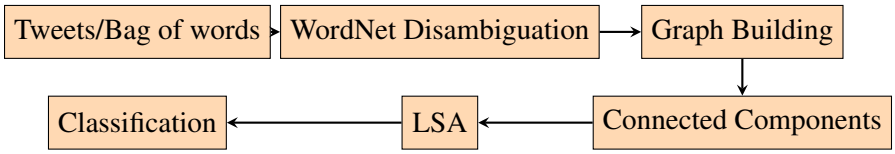

Figure 1: The main processes for building the matrix of connected components.

In the next sections we present the main steps in our methodology. Section 4.1 shows how we use WordNet to disambiguate terms. Section 4.2 presents our method for the construction of a graph of terms in tweets and how we extract connected components from this graph. In Section 4.3, we explain how we use connected components to lower the ranking in our classification matrix. Finally, Section 4.4 shows the classification algorithms used in this work.

4.1 Disambiguation of Tweets Using WordNet

Since we use Wordnet[12] for the tweets expansion, we decided to also use it for the task of disambiguation.

For each word, WordNet suggests many senses, each one containing a synset, i.e. a group of almost synonyms words. We adopt a structural method based on the semantic distance between concepts to choose the correct sense, according to the formula (Navigli, 2009):

$$\hat{S} = \textit{argmax}_{S \in Sense(w_i)} \sum_{w_j \in T : w_i \neq w_j} \textit{max}_{S' \in Sense(w_j)} Score(S, S'). \quad (1)$$

Where T is the set of terms in a tweet, w_i is the term we want to disambiguate, $Sense(w_i)$ is the set of candidates concepts for the term w_i, what in WordNet corresponds to the synsets containing this term, and $Score(S, S')$ is the function used to measure the similarity between two concepts S and S'.

There are many methods for measuring the similarity between concepts S and S'. After many comparisons, we have chosen an approach based on a path formed by the arcs of the graph (Wu and Palmer, 1994), which supposes that the similarity between two concepts depends on the distance between the nodes concerned and their common ancestor (Least Common Concept) in comparison to the distance to a root node in the graph.

To enrich the tweets, we use the technique introduced by (Audeh et al., 2013) from information search for the expansion of queries. The first option is to search for synonyms in the synset selected in the previous step. The second one is to use the synsets of the hypernym of the term.

10. http://members.unine.ch/jacques.savoy/clef/englishST.txt
11. http://nlp.stanford.edu/software/
12. http://wordnetcode.princeton.edu/wn3.1.dict.tar.gz

4.2 Graph Building

Once the synset having the closest sense to the context of the tweet is selected, we group the words w extracted from the tweets with their synsets in a graph $G = (V, E)$, defined as follows:

$$\begin{cases} V & = & \{\forall w \in E_{tweet} / \{w\} \cup Synset_w\} \\ \\ E & = & \{Synonym, Hyperonym \cdots\} \end{cases} \tag{2}$$

In this graph, each word from the corpus and each word in the synsets, as selected by the previous step, is represented by a node V and each relation (synonymy, hyperonymy) between those words, extracted from WordNet, is represented by an arc E. This creates a weakly connected graph that is used to the extraction of connected components.

The next step is to search for the connected components in the graph G. Each component is formed by nodes, corresponding to words, connected by arcs representing the semantic relations. The idea is to cluster the words w_1 and w_2 connected by an arc to another word $w \in \{w_1\} \cup Synset_{w_1}$, having this word w a relation $w \in \{w_2\} \cup Synset_{w_2}$. Mathematically:

$$G'\left(V', E'\right) \textit{ a connected component in G(V,E) } / V' \subset V \wedge \{w_1, w_2\} \in V'$$

Then we have:

$$\{\{w_1\} \cup Synset_{w_1}\} \cap \{\{w_2\} \cup Synset_{w_2}\} \neq \phi \tag{3}$$

A word, called the "representative", is selected to represent each component. The matrix of bag of words becomes a matrix of bags of representatives, or bag of connected components. For example, in Figure 2 the word "football" is the representative of the component:

football ->[football, football game, soccer/sports, association football, soccer]

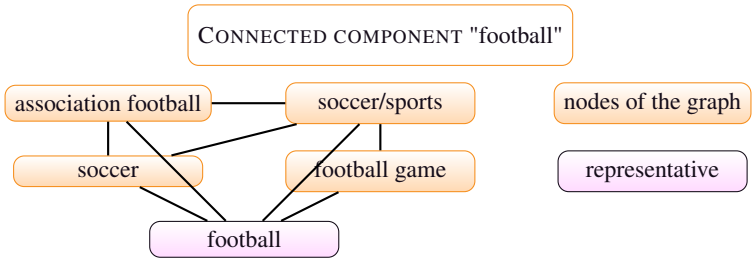

Figure 2: Example of a connected component represented by the word "football" in the vectorial space

Every time we find a word that belongs to this component in a tweet, we increment the frequency of the word "football", which is the representative of this component, leading to a matrix of fewer dimensions.

4.3 Rank Lowering Using Latent Semantic Analysis

Latent Semantic Analysis (LSA) (Bestgen, 2004) is a technique for analyzing the relation between terms and between terms and documents. Each term is represented as a column in a matrix and each document as a row. To represent a corpus of tweets, for example, row_1 represents $tweet_1$, row_2 represents $tweet_2$, etc. Each column represents the frequency or the tf-idf of a term in a tweet. Since tweets are short, most of the columns for a specific tweet are zeros since most of the words present in the entire corpus are not present in each specific tweet. And even in a big corpus, most of the words are rare, appearing only once and their rarity makes them sensible to aleatory variations (Bestgen, 2004).

In this work, we deal with the sparsity of such a matrix by representing the frequency of terms in connected components instead of the frequency or the tf-idf of single terms since we are more interested in the relation of tweets with concepts, represented by the connected components, than the relation of tweets with isolated terms.

Dimension of components / Vectors of tweets	**Component**$_1$	$\cdots$	**Component**$_i$	$\cdots$	**Component**$_n$
Tweet$_1$	$f_{1,1}$	$\cdots$	$f_{1,i}$	$\cdots$	$f_{1,n}$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
Tweet$_j$	$f_{j,1}$	$\cdots$	$f_{j,i}$	$\cdots$	$f_{j,n}$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
Tweet$_m$	$f_{m,1}$	$\cdots$	$f_{m,i}$	$\cdots$	$f_{m,n}$

Table 1: A matrix representing the frequency of each connected component in each tweet.

Table 1 shows the representation of a matrix of m tweets with the frequencies $f_{j,i}$ of words found in the connected components $Component_i$.

4.4 The Classifier

Liblinear (Fan et al., 2008) is a fast and simple linear classifier and has become one of the most promising machine learning technique for big data.

We use the library LIBLINEAR[13], which is based on the L2-regularized logistic regression (LR), L2-loss and L1-loss SVM linear vectors (Boser et al., 1992). It inherits many characteristics from the SVM library LIBSVM (Chang and Lin, 2011), like a rich documentation and free license (BSD license[14]).

LIBLINEAR is very efficient for training large scale problems: it takes some seconds for a text classification problem. For the same task, a SVM classifier like LIBSVM takes many hours. Furthermore, LIBLINEAR competes with the fastest linear classifiers like Pegasos (Shalev-Shwartz et al., 2007).

5 Experiments and Analysis of Results

We use the Twitter API to build our corpus, passing as queries words related to the chosen domains (sport, politics, economics and medicine). Table 2 presents the number of tweets, terms and lemmas in the corpus for each domain.

	Economy	Medicine	Sport	Politics
Simple Terms	13870	14784	16112	15346
Lemmas	7938	12138	12773	11976
Tweets	2504	2415	2493	2497

Table 2: Number of terms, lemmas and tweets for each category

To test our method we evaluate different criteria in the classification of tweets:

1. Corpus filtered by part-of-speech (POS) vs not filtered by POS. In the corpus filtered by POS, in addition to deleting a regular list of stop words (article, prepositions, etc.), we also delete adverbs and we keep only nouns, verbs and adjectives;

2. Applying vs not applying hashtags segmentation;

3. Using Wordnet relations (synonymy, hyperonymy and synonymy + hyperonymy) for clustering similar words in a graph for the construction of connected components;

4. Applying NER (and using WordNet to disambiguate them) vs not Applying NER.

13. http://www.csie.ntu.edu.tw/ cjlin/liblinear
14. The new BSD license (Berkeley Software Distribution License) approved by the Open Source initiative.

Table 3 presents a comparison of the classification results and the gain we obtain when NER is applied.

Filter type	Hashtag segmentation	WordNet relation	Precision w/o ^{-}NER (%)	Precision with ^{+}NER (%)	Gain(p.p.)
All words	No	Synonymy	38.3	36.2	-2.1
Only N, V and ADJ	No	Synonymy	68.2	69.7	+1.5
All words	Yes	Synonymy	42.1	41.7	-0.4
Only N, V and ADJ	Yes	Synonymy	73.4	74.8	+1.4
All words	No	Hyperonymy	41.7	40.6	-1.1
Only N, V and ADJ	No	Hyperonymy	70.9	71.3	+0.4
All words	Yes	Hyperonymy	43.2	42.4	-0.8
Only N, V and ADJ	Yes	Hyperonymy	76.4	77.3	+0.9
All words	No	Syn&Hyper	46.8	45.3	-1.5
Only N, V and ADJ	No	Syn&Hyper	72.3	73.2	+0.9
All words	Yes	Syn&Hyper	51.3	50.6	-0.7
Only N, V and ADJ	Yes	Syn&Hyper	81.2	83.4	+1.2

Table 3: Comparison of the precision for different classifications and the gain when named entity recognition is applied.

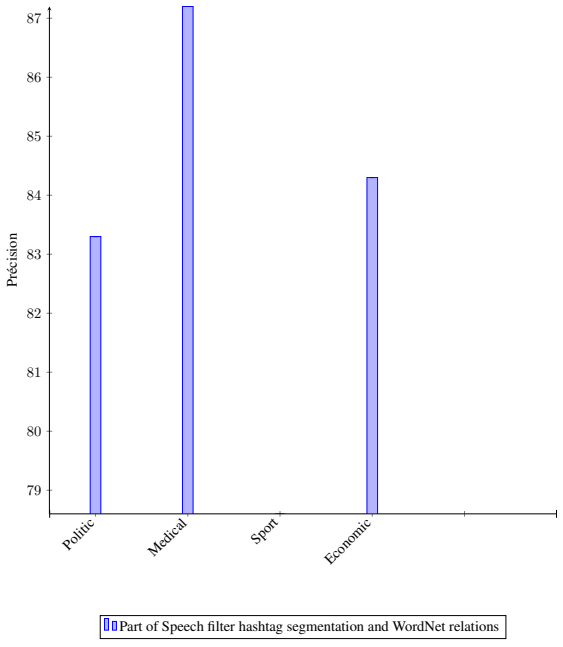

Figure 3: Best results obtained in the classification of each category.

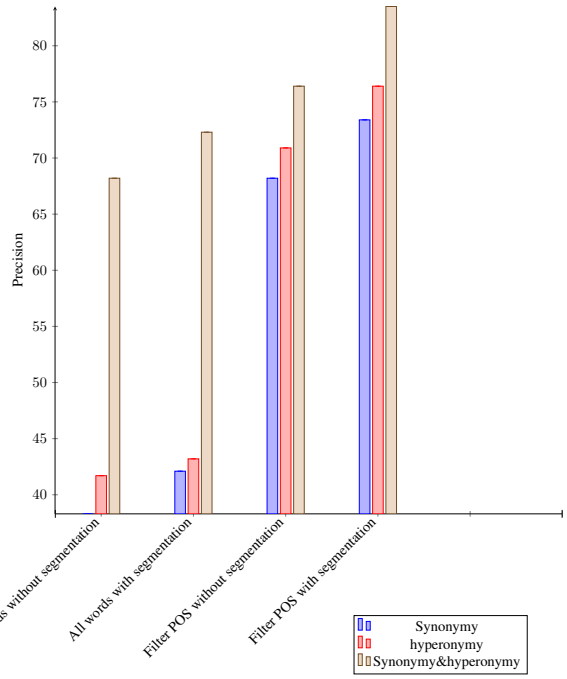

Figure 4: Precisions obtained when applying named entity recognition (NER).

When NER is not applied and we keep only nouns, verbs and adjectives, combined with synonymy, we have a gain of 29.9 p.p. in precision (from 38.3% to 68.2%), which grows to 81.2% when combined with synonymy and hyperonymy. This can be explained by the fact that the words removed, like adverbs, do not contribute for the identification of a tweet subject.

The hashtags segmentation improves the result in 3.8 p.p., when using all the words and 5.2 p.p. when the POS filtering is applied. This is explained by the fact that usually hashtags carries the most significant words for the identification of the subject of a tweet.

The use of hyperonyms gives a better result than the use of synonyms. For example, the precision is 73.4% when we use hashtag segmentation, POS filter and synonyms and it increases to 76.4% when we use hyperonyms instead of synonyms. This can be explained by the fact that hyperonyms by themselves are a better indication of a word group since they represent relations of the type "part-of". Moreover, adding synonyms and hyperonyms together gives the best precision, 81.2%, when using hashtag segmentation and POS filter.

The application of NER increases the precision in the corpus having only nouns, verbs and adjectives, and decreases when the corpus have all words (except stop words).

The histogram in Figure 3 compares the precision in the classification of each category using NER, hashtag segmentation and POS filter.

The histogram in Figure 4 is a graphic representation of the results shown by the column "Precision with NER" in Table 3. We note a progressive increment in precision when segmentation of hashtags and then POS filtering are applied.

6 Evaluation

The hashtags carry important information about the tweets subjects. Moreover, words extracted from them enrich the connected components that contain those words. For example, a tweet that contains the hashtag #ParisClimateConference does not share any word with the following connected component: climate→ environmental condition, clime, climate. However, after the hashtag segmentation, the word $climate$ appears in the vector representing the tweet that contains this hashtag:

$$\text{\#ParisClimateConference} \rightarrow (\text{paris}, \text{climate}, \text{conference})$$

The tweets containing the word $climate$ in their texts or in their hashtags will share the same connected component.

Keeping only verbs, adjectives and adverbs (POS filter) helps improving the classification precision since the sense of the text is usually given by words in those grammatical categories.

Despite the existence of polysemic words, the disambiguation using WordNet, as explained in section 4.1, helps us find the correct sense of a word.

Not all named entities can be detected by WordNet. However, the most common names of people, places and organizations used in tweets can be successfully identified. The application of NER helps increase the precision. For example, in tweets we find "United States of America", "United States" and "USA". Identifying that the three expressions are a unique entity helps understand the connection between the tweets that contain them.

We show how NER can affect the identification of the subject of a text. Without NER, we could have the following connected components:

$$\begin{aligned} \text{United} &\rightarrow \text{unite, unify} \\ \text{States} &\rightarrow \text{government, authorities, regime} \end{aligned}$$

However, NER gives:

$$\begin{aligned} \text{United States} \rightarrow &\text{United States, United States of America, America,} \\ &\text{the States, US, U.S., USA, U.S.A} \end{aligned}$$

Moreover, the detection of the sense of some acronyms using WordNet, for example:

$$\begin{aligned} \text{FBI} &\rightarrow \text{Federal Bureau of Investigation, FBI; or} \\ \text{UN} &\rightarrow \text{United Nations, UN} \end{aligned}$$

helps to connect the sense of a tweet containing *«UN»* with tweets containing *«United Nations»*. Finally, we have a reduction in the number of dimensions in our training matrix.

7 Conclusion

Before discussing results, it is important to stress some limits of this research. First, we use a local WordNet, instead of an online one, due to performance. Second, not all terms or named entities found in our corpus are present in WordNet.

The idea of connected components, based on graph theory, reduces the training matrix based on bag of words, increasing the performance of our classification. Our POS filter improves the precision. And the hashtag segmentation helps us extract more information from tweets and also helps increase the precision of the classification.

NER improves the precision when we keep only nouns, verbs and adjectives. When all words are used, the precision decreases.

In this work we do not consider multiword expressions (MWEs). Finally, the hashtag segmentation does not take into account hashtags written in more than one language, even if one of the languages is English, like, for example, the hashtag #Japanにほん, which is composed by a word in English and another in Japanese.

References

H. Almeida, M. J. Meurs, L. Kosseim, and A. Tsang. 2016. Data sampling and supervised learning for hiv literature screening. *IEEE Transactions on NanoBioscience*, 15(4):354–361, June.

S. Asur and B. A. Huberman. 2010. Predicting the future with social media. In *Web Intelligence and Intelligent Agent Technology (WI-IAT), 2010 IEEE/WIC/ACM International Conference on*, volume 1, pages 492–499, Aug.

Bissan Audeh, Philippe Beaune, and Michel Beigbeder. 2013. Expansion sémantique des requêtes pour un modèle de recherche d'information par proximité. In Chantal Soulé-Dupuy, editor, *INFORSID 2013*, pages pages 83–90, Paris, France, May. https://liris.cnrs.fr/inforsid/?q=Actes

Yves Bestgen, 2004. *Analyse sémantique latente et segmentation automatique des textes*. Cahiers du Cental. Presses universitaires de Louvain, Louvain-la-Neuve.

Bernhard E. Boser, Isabelle M. Guyon, and Vladimir N. Vapnik. 1992. A training algorithm for optimal margin classifiers. In *Proceedings of the Fifth Annual Workshop on Computational Learning Theory*, COLT '92, pages 144–152, New York, NY, USA. ACM.

Caroline Brun and Claude Roux. 2014a. Decomposing hashtags to improve tweet polarity classification (décomposition des « hash tags » pour l'amélioration de la classification en polarité des « tweets ») [in french]. In *Proceedings of TALN 2014 (Volume 2: Short Papers)*, pages 473–478. Association pour le Traitement Automatique des Langues.

Caroline Brun and Claude Roux. 2014b. Decomposing hashtags to improve tweet polarity classification (décomposition des « hash tags » pour l'amélioration de la classification en polarité des « tweets ») [in french]. In *Proceedings of TALN 2014 (Volume 2: Short Papers)*, pages 473–478. Association pour le Traitement Automatique des Langues.

Amparo E. Cano, Andrea Varga, Matthew Rowe, Fabio Ciravegna, and Yulan He. 2013. Harnessing linked knowledge sources for topic classification in social media. In *Proceedings of the 24th ACM Conference on Hypertext and Social Media*, HT '13, pages 41–50, New York, NY, USA. ACM.

Chih-Chung Chang and Chih-Jen Lin. 2011. Libsvm: A library for support vector machines. *ACM Trans. Intell. Syst. Technol.*, 2(3):27:1–27:27, May.

Rong-En Fan, Kai-Wei Chang, Cho-Jui Hsieh, Xiang-Rui Wang, and Chih-Jen Lin. 2008. Liblinear: A library for large linear classification. *J. Mach. Learn. Res.*, 9:1871–1874, June.

Yegin Genc, Yasuaki Sakamoto, and Jeffrey V. Nickerson, 2011. *Discovering Context: Classifying Tweets through a Semantic Transform Based on Wikipedia*, pages 484–492. Springer Berlin Heidelberg, Berlin, Heidelberg.

Bo Han, Paul Cook, and Timothy Baldwin. 2013. Lexical normalization for social media text. *ACM Trans. Intell. Syst. Technol.*, 4(1):5:1–5:27, February.

Sheila Kinsella, Alexandre Passant, and John G. Breslin, 2011. *Topic Classification in Social Media Using Metadata from Hyperlinked Objects*, pages 201–206. Springer Berlin Heidelberg, Berlin, Heidelberg.

Kathy Lee, Diana Palsetia, Ramanathan Narayanan, Md. Mostofa Ali Patwary, Ankit Agrawal, and Alok Choudhary. 2011. Twitter trending topic classification. In *Proceedings of the 2011 IEEE 11th International Conference on Data Mining Workshops*, ICDMW '11, pages 251–258, Washington, DC, USA. IEEE Computer Society.

Bing Liu, 2010. *Sentiment analysis and subjectivity*, chapter 26, pages 627–666. Chapman and Hall/CRC, second edition, February.

Matthew Michelson and Sofus A. Macskassy. 2010. Discovering users' topics of interest on twitter: A first look. In *Proceedings of the Fourth Workshop on Analytics for Noisy Unstructured Text Data*, AND '10, pages 73–80, New York, NY, USA. ACM.

Roberto Navigli. 2009. Word sense disambiguation: A survey. *ACM Comput. Surv.*, 41(2):10:1–10:69, February.

Mehran Sahami and Timothy D. Heilman. 2006. A web-based kernel function for measuring the similarity of short text snippets. In *Proceedings of the 15th International Conference on World Wide Web*, WWW '06, pages 377–386, New York, NY, USA. ACM.

Hassan Saif, Yulan He, and Harith Alani, 2012. *Semantic Sentiment Analysis of Twitter*, pages 508–524. Springer Berlin Heidelberg, Berlin, Heidelberg.

Jagan Sankaranarayanan, Hanan Samet, Benjamin E. Teitler, Michael D. Lieberman, and Jon Sperling. 2009. Twitterstand: News in tweets. In *Proceedings of the 17th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, GIS '09, pages 42–51, New York, NY, USA. ACM.

Shai Shalev-Shwartz, Yoram Singer, and Nathan Srebro. 2007. Pegasos: Primal estimated sub-gradient solver for svm. In *Proceedings of the 24th International Conference on Machine Learning*, ICML '07, pages 807–814, New York, NY, USA. ACM.

Nakatani Shuyo. 2010. Language detection library for java.

Bharath Sriram, Dave Fuhry, Engin Demir, Hakan Ferhatosmanoglu, and Murat Demirbas. 2010. Short text classification in twitter to improve information filtering. In *Proceedings of the 33rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '10, pages 841–842, New York, NY, USA. ACM.

Joachim Wermter, Katrin Tomanek, and Udo Hahn. 2009. High-performance gene name normalization with geno. *Bioinformatics*, 25(6):815–821.

Zhibiao Wu and Martha Palmer. 1994. Verbs semantics and lexical selection. In *Proceedings of the 32Nd Annual Meeting on Association for Computational Linguistics*, ACL '94, pages 133–138, Stroudsburg, PA, USA. Association for Computational Linguistics.