Sarcasm Detection in Twitter
"All your products are incredibly amazing!!!" – Are They Really ?

Mondher Bouazizi
Graduate School of Science and Technology
Keio University
Yokohama, Japan
Email: bouazizi@ohtsuki.ics.keio.jp

Tomoaki Ohtsuki
Department of Information and Computer Science
Faculty of Science and Technology, Keio University
Yokohama, Japan
Email: ohtsuki@ics.keio.ac.jp

***Abstract*—Sarcasm is a special form of irony by which the person conveys implicit information, usually the opposite of what is said, within the message he transmits. Sarcasm is largely used in social networks and microblogging websites, where people mock or criticize in a way that makes it difficult even for humans to tell if what is said is what is meant. Recognizing sarcastic statements can be very useful when it comes to improving automatic sentiment analysis of data collected from social networks. It helps also enhance the efficiency of after-sales services or consumer assistance through understanding the intentions and real opinions of consumers when browsing their feedbacks or complaints. In this paper we propose a method to detect sarcasm in Twitter that makes use of the different components of the tweet. We propose four sets of features that cover different types of sarcasm we defined, and that will be used to classify tweets into sarcastic and non-sarcastic. We evaluate the performances of our approach. We study the importance of each of the proposed sets of features and evaluate its added value to the classification.**

I. Introduction

Twitter has become one of the biggest web destinations for users to express their opinions and thoughts. Throughout the last few years, Twitter content continued to increase: hundreds of millions of tweets are sent everyday by more than 285 million active users[1]. Many companies and organizations have been interested in these data for the purpose of studying the opinion of people towards political events [1], popular products [2] or movies [3].

However, due to the informal language used in Twitter and the limitation in terms of characters (i.e., 140 characters per tweet), understanding the opinions of users and performing such analysis is quite difficult. Furthermore, presence of sarcasm makes the task even more challenging: sarcasm is when a person says something different from what he means. Some people are more sarcastic than others, however, in general, sarcasm is very common, though, difficult to recognize.

Oxford dictionary[2] defines sarcasm as *"the use of irony to make or convey contempt"*. However, in general, people employ sarcasm in their daily life not only to make jokes and be humorous but also to criticize or make remarks about ideas, persons or events. Therefore, it tends to be widely used in social networks, in particular microblogging websites such as Twitter. That being the case, the state of the art approaches of sentiment analysis and opinion mining tend to have lower performances when analyzing data collected from such websites. Maynard et al. [4] show that sentiment analysis performance might be highly enhanced when sarcasm within the sarcastic statements is identified. Therefore, the need for an efficient way to detect sarcasm arises.

In this paper, we propose an efficient way to detect sarcastic tweet. Although it does not need an already-build user-knowledge-base as in the work of Rajadesingan et al. [5], our approach considers the different types of sarcasm and detect the sarcastic tweets regardless of their owners or their temporal context.

The remainder of this paper is structured as follows. Section II presents our motivations and describes some of the related work. Section III describes in details our proposed method, the different features we used and the tuning procedure of the parameters we defined. Section IV illustrates our experiments and results, and Section V concludes this work and proposes possible directions for future work.

II. Motivations and Related Work

A. Motivations

Sentiment analysis and opinion mining rely on emotional words in a text to detect its polarity (i.e., whether it deals *"positively"* or *"negatively"* with its theme). However, the appearance of the text might be misleading. A typical example of that is when the text is sarcastic. In Twitter, such sarcastic texts are very common. *"All your products are incredibly amazing!!!"* might be considered as a compliment. However, considering the following tweet *"Did I say incredibly?? Well, it's true, nobody would believe that. They break the second day you buy them -_-"*, the Twitterer explicitly explains that he did not mean what he said. Although some users indicate they are being sarcastic, most of them do not. Therefore, it might be indispensable to find a way to automatically detect any sarcastic messages.

B. Related Work

Sarcasm has been subject to deep studies form psychological [6] perspective. Nevertheless, it has been studied as a linguistic behavior characterizing the human being. In this

[1]https://about.twitter.com/company
[2]http://www.oxforddictionaries.com/

978-1-4799-5952-5/15/$31.00 ©2015 IEEE

context, researchers have recently been interested in sarcasm, trying to find ways to automatically detect it when it is present in a statement. Burfoot et al. [8] introduced the task of filtering satirical news articles from true newswire documents. They introduced a set of features including the use of profanity, slangs and "semantic validity"; and used Support Vector Machine (SVM) to recognize satire articles.

Davidov et al. [9] proposed a semi-supervised approach to identify sarcasm in amazon and Twitter. The results they obtained were interesting, though their approach relies on the frequency of appearance of words to define sarcastic patterns, and treats what is called "Context Words" in the same way regardless of their grammatical function. Maynard et al. [4] relied on hashtags that twitterers employ in their tweets to identify sarcasm. Riloff et al. [10] proposed a method to detect a specific form of sarcasm, where a positive sentiment contrasts with a negative situation. They introduced a bootstrapping algorithm that uses the single seed word *"love"* and a collection of sarcastic tweets to automatically detect and learn expressions showing positive sentiment and phrases citing negative situations.

Rajadesingan et al. [5] went deeper and dealt with the psychology behind sarcasm. They identified different forms of sarcasm and their manifestation in Twitter, and demonstrated the importance of historical information collected from the past tweets for sarcasm detection.

III. Proposed Approach

Given a set of tweets, we aim to classify each one of them depending on whether it is sarcastic or not. Therefore, from each tweet, we extract a set of features, refer to a training set and use machine learning algorithms to perform the classification. The features are extracted in a way that covers the different types of sarcasm we identified.

A. Data

Throughout the period ranging from December 2014 to March 2015, we collected tweets, using Twitter's streaming API. To collect sarcastic tweets, we queried the API for tweets containing the hashtag *"#sarcasm"*. As for non-sarcastic tweets, we collected tweets dealing with different topics and excluded ones that contain any hashtag referring to sarcasm. We prepared 3 datasets for our work as follows:

- **Set 1:** this set contains 6000 tweets, half of them are sarcastic, and the other half are not. The tweets on this dataset are manually checked and classified depending on their level of sarcasm from 1 (highly non-sarcastic) to 6 (highly sarcastic). This first set is used for training. In the rest of this works, it will be referred to as the *"training set"*. The number of sarcasm levels is also referred to as N_S and is equal to 6.
- **Set 2:** this set contains 1128 tweets having the hashtag *"#sarcasm"*, and 1128 non-sarcastic ones. No manual check is done, which makes it a noisy set. However, to reduce the noise, we filtered-out the non-English tweets, very short tweets (i.e., that have less than 3 words), and those which contain URLs. This dataset is used during our experimenting process to tune the parameters we defined for our features. In the rest of this works, we will refer to this set as the *"optimization set"*.
- **Set 3:** this set contains 500 sarcastic tweets, and 500 non-sarcastic ones. All tweets are manually checked and classified into sarcastic and non-sarcastic. This set will serve as a test set. Therefore, in the rest of this work, it will be referred to as the *"test set"*.

None of the tweets of any of the aforementioned sets is re-used in another. In addition, during our work, we removed the hashtag *"#sarcasm"* from all the tweets.

B. Features Extraction

Being a sophisticated form of speech, sarcasm, is used for different purposes, that fall in general in three categories:

a) sarcasm as wit: when used as a wit, sarcasm is employed with the purpose of being funny; the person employs some special forms of speeches, tends to exaggerate, or uses a tone that is different from that when he talks usually. In social networks, voice tones are converted into special forms of writing: use of capital letter words, exclamation and question marks, as well as some "sarcasm-related" emoticons.

b) sarcasm as whimper: when used as whimper, sarcasm is employed to show how annoyed or angry the person is. Therefore, it tempts to show how bad the situation is by using exaggeration, or by employing very positive expressions to describe a negative situation.

c) sarcasm as an avoidance: it refers to the situation when the person wants to avoid giving a clear answer, thus, makes use of sarcasm. In this case, the person employs complicated sentences, uncommon words and some unusual expressions.

We rely on these assumption to build our model. Four sets of features are extracted: sentiment-related features, punctuation-related features, syntactic features and pattern features.

1) Sentiment-related Features: A widely used form of sarcasm in both regular conversations as well as short messages, is when an emotionally positive expression is used in a negative context. Riloff et al. [10] show that this type of sarcasm can be identified and detected when a positive statement is collocated with a negative situation.

In our work, we opt for a more straight-forward, yet more general, approach. We consider any kind of inconsistency between sentiments of words as well as other components within the tweet. Therefore, to identify and quantify such inconsistency, we extract sentimental components of the tweet and count them. For this purpose, we maintain two lists of words qualified as "positive words" and "negative words". The two lists contain respectively words that have positive emotional content (e.g., *"love"*, *"happy"*, etc.) and negative emotional content (e.g., *"hate"*, *"sad"*, etc.). Using these two lists, we extract two features we denote respectively pw and nw by counting the number of positive and negative words in the tweet.

TABLE I
POS-TAGS FOR WORDS CONSIDERED AS HIGHLY EMOTIONAL

Part of Speech	Part of Speech Tag
Adjectives	"JJ", "JJR", "JJS"
Adverbs	"RB", "RBR", "RBS"
Verbs	"VB", "VBD", "VBG", "VBN", "VBP", "VBZ"

Adjectives, verbs and adverbs have higher emotional content than nouns; therefore positive and negative words that have the associated PoS-tag, shown in TABLE I, are counted again and used to create two more features that we denote PW and NW and which represent the number of highly emotional positive words and highly emotional negative words.

We then add three more features by counting the number of positive, negative and sarcastic emoticons. Emoticons are the facial expressions such as smile that are formed by typing a sequence of keyboard symbols, and that are usually used to convey the writer's sentiment, emotion or intended tone. In particular, sarcastic emoticons are ones used sometimes with sarcastic or ironical statements (e.g., *":P"*).

Hashtags also have emotional content. In some cases, they are used to disambiguate the real intention of the twitterer conveyed in his message. Therefore, we count also the number of positive and negative hashtags.

In addition to the aforementioned features, we extract features related to the contrast between these sentimental components. We first calculate the ratio of emotional words $\rho(t)$ defined as

$$\rho(t) = \frac{(\delta \cdot PW + pw) - (\delta \cdot NW + nw)}{(\delta \cdot PW + pw) + (\delta \cdot NW + nw)} \quad (1)$$

where t is the tweet, and δ is a weight given to the highly emotional words. In case the tweet does not contain any emotional word, $\rho(t)$ is set to 0. In the rest of this work, δ is set to 3.

We then define 4 features that represent whether there is a contrast between the different components. By contrast we mean the coexistence of a negative component and a positive one within the same tweet: we denote respectively the existence of such contrast between words, between hashtags, between words and hashtags and between words and emoticons by $\psi^{ww}(t)$, $\psi^{hh}(t)$, $\psi^{wh}(t)$ and $\psi^{we}(t)$ and use them as extra features.

2) Punctuation-Related Features: Sarcasm is a sophisticated form of speech: not only it plays with words and meanings, but also it employs behavioral aspects such as low tones, facial gestures or exaggeration. These aspects are translated into a certain use of punctuation or repetition of vowels when the message is written. To detect such aspects, we extract this second set of features. For each tweet, we calculate the following features: number of exclamation marks, that of question marks, that of dots, that of all-capital words, and that of quotes. We also add a sixth feature by checking if any of the words contain a vowel that is repeated more than twice (e.g. *"looooove"*). If such a word exists, the feature is set to *"true"*, otherwise, it is set to *"false"*. We define one last feature by counting the number of words in the tweet.

3) Lexical and Syntactic Features: Along with the punctuation-related features, some common expressions are used usually in a sarcastic context. It is possible to correlate these expressions with the punctuation to decide whether what is said is sarcastic or not. Besides, in other cases, people tend to make complicated sentences or use uncommon words to make it ambiguous to the listener/reader to get a clear answer. This is common when sarcasm is used as avoidance, where the person's purpose is to hide his real feeling or opinion by burying them in humor. Hence, we extract the following features that reflects these aspects:

- Use of uncommon words[3]
- Number of uncommon words
- Existence of common sarcastic expressions
- Number of interjections
- Number of laughters

In particular, the feature "Existence of common sarcastic expression" is extracted in the same way we extract the features qualified as "pattern-related" (this will be described in detail in the next subsection). Here we used a noisy set of 3000 tweets having the hashtag *"#sarcasm"*. We extracted all possible patterns of length varying from 3 to 6, we selected the patterns that appeared more than 10 times. Being few in number, we manually checked the list and removed the irrelevant ones. We obtained a list of 13 main patterns including [*love* PRONOUN *when*] (e.g., *"I love it when I am called at 4 a.m. by an unknown number"*), etc.

4) Pattern-Related Features: The patterns selected in the previous subsection, and qualified of "common sarcastic expression" are common, even in spoken language. However, they are small in number and they are not unique. That being the case, we dig further and extract another set of features. The idea of our pattern-related features is inspired from the work of Davidov et al. [9]. However, we propose more efficient and reliable way to extract patterns. We divide words into two classes: A first one referred to as *"CI"* containing words of which the content is important and a second one referred to as *"GFI"* containing those of which the grammatical function is more important. If a word belongs to the first category, it is transformed into its form; otherwise, we replace it by a certain expression. The expressions used to replace these words are shown in TABLE II. The classification into classes is done based on the PoS-tag of the word in the tweet as given in TABLE III.

We generate the vector of words for each tweet according to the rule defined. We define a pattern as an ordered sequence of words. The patterns are extracted from the training set and are taken such as their length satisfies

$$L_{min} \leq Length(pattern) \leq L_{max} \quad (2)$$

where L_{min} and L_{max} represent the minimal and maximal allowed length of patterns in $words$ and $Length(pattern)$ is

[3]We referred to http://www.wordfrequency.inf/ to obtain the list of the most common English words

TABLE II
EXPRESSIONS USED TO REPLACE THE WORDS OF GFI

PoS-tag	Expression
“CD”	CARDINAL
“FW”	FOREIGNWORD
“UH”	INTERJECTION
“LS”	LISTMARKER
“NN”, “NNS”, “NNP”, “NNPS”,	NOUN
“PRP”, “PRP$”	INTERJECTION
“MD”	MODAL
“PB”, “RBR”, “RBS”	ADVERB
“WDT”, “WP”, “WP$”, “WRB”	WHDETERMINER
“SYM”	SYMBOL

TABLE III
PART-OF-SPEECH TAGS CLASSES

Class	PoS Tags
CI	“CC”, “DT”, “EX”, “IN”, “JJ”, “JJR”, “JJS”, “PDT”, “POS”,“RP”, “TO”, “VB”, “VBD”, “VBG”, “VBN”, “VBP”, “VBZ”
GFI	“CD”, “FW”, “LS”, “NN”, “NNS”, “NNP”, “NNPS”, “PRP”, “PRP$”, “SYM”, “UH”
CI / GFI	“MD”, “RB”, “RBR”, “RBS”, “WDT”, “WP”, “WP$”, “WRB”

the length of the pattern in $words$. The number of pattern lengths is $N_L = (L_{max} - L_{min} + 1)$.

Only patterns that appear at least N_{occ} times in our training set are kept; the others are discarded. In the rest of this work, N_{occ} is set to 2. In addition, a pattern that appears in a sarcastic tweet and in a non-sarcastic tweet is discarded. This step is done to filter out patterns that are not related to sarcasm. After the selection, we classify the resulted patterns into N_F sets, where

$$N_F = N_L \times N_S. \tag{3}$$

We create N_F features, as shown in TABLE IV. Each feature F_{ij} in the table represents the degree of resemblance of the tweet to the patterns of degree of sarcasm i and length j. Therefore, given a tweet t, we calculate the resemblance degree [9] $res(p,t)$ of each pattern in the training set p to the tweet t.

$$res(p,t) = \begin{cases} 1, & \text{if the tweet vector contain the pattern as it is, in the same order,} \\ \alpha * n/N, & \text{if } n \text{ words out of the } N \text{ words of the pattern appear in the tweet in the correct order,} \\ 0, & \text{if no word of the pattern appear in the tweet.} \end{cases}$$

Given N_{ij} the number of patterns collected from the training set having a sarcasm degree i and a length j, the value of the feature, F_{ij}, is

$$F_{ij} = \beta_j * \sum_{k=1}^{N_{ij}} res(p_k, t) \tag{4}$$

TABLE IV
PATTERN FEATURES

		Pattern length			
		L_1	L_2	$\cdots$	L_N
Sarcasm	1	F_{11}	F_{12}	$\cdots$	F_{1N}
	2	F_{21}	F_{22}	$\cdots$	F_{2N}
level	$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$
	6	F_{61}	F_{62}	$\cdots$	F_{6N}

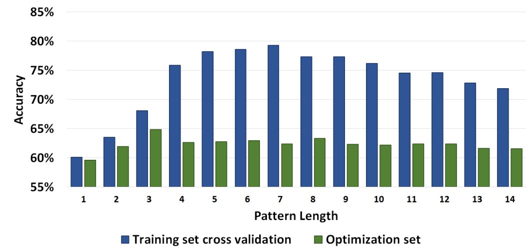

Fig. 1. Accuracy per pattern length for fixed values of $\alpha, \beta_1, \cdots, \beta_{N_L}$

where β_j is a weight given to patterns of length L_j (regardless of their level of sarcasm). We give different weights for each length of pattern since longer patterns are more likely to have higher impact. F_{ij} as defined measures the degree of resemblance of a tweet t to patterns of level of sarcasm i and length j.

Extension of the training set Patterns: Being relatively small in size to cover all possible patterns (i.e., only 6000 tweets), our training set has been enriched to obtain more patterns. We collected 18 959 more tweets containing the hashtag *“#sarcasm”* and 18 959 more tweets that do not. We checked if the tweets having the hashtag *“#sarcasm”* contain any of the sarcastic patterns we already extracted from the training set and that have a length equal to or more than 4. If that is the case, we extract the different patterns from the tweet and add them to the list of patterns of the training set. We did the same to the non-sarcastic tweet.

Pattern-related features as defined give a high flexibility to tune the features depending on their contribution. In total we have the following parameters to optimize:

- L_{min} and L_{max}
- α
- $\beta_1, \ldots, \beta_{N_L}$

To optimize L_{min} and L_{max}, we fixed α to be equal to 0.1 and β_n $(n = 1, \cdots, N_L)$ to be equal to 1.

We run a first simulation on our training set and optimization set, for each pattern length. We obtained the results shown in Fig. 1. The results present the accuracy of the classification of tweets into sarcastic and non-sarcastic. The obtained results show that the patterns having a length are from 4 to 10 give the highest accuracy (i.e., more than 75% accuracy during 10-folds cross validation). Pattern length 3 gives the highest accuracy on our optimization set. Given that the average number of words per tweet is equal to 11.48, we set the to parameters L_{min} and L_{max} respectively to 3 and 10.

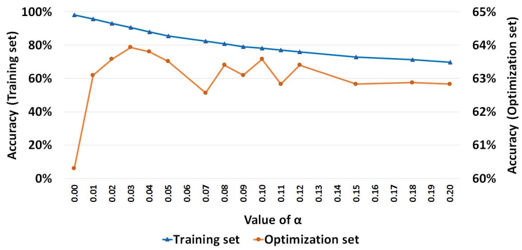

Fig. 2. Accuracy of classification for different values of α

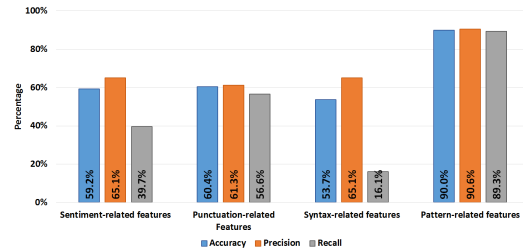

Fig. 3. Accuracy of classification during cross-validation for each family of features

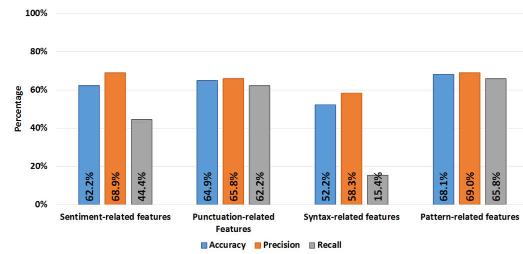

Fig. 4. Accuracy of classification of the test set for each family of features

Afterwards, we set L_{min} and L_{max} as mentioned, kept the values of $\beta_1, \cdots, \beta_{N_L}$ as they are (i.e., equal to 1). We run different simulations on the same datasets using pattern features, for different values of α. Results of the test are given in Fig. 2. The accuracy of classification varies highly depending on the value of α, that is, the lower the value is, the better the performances are during the cross validation. This is due to the unicity of the patterns. In other terms, the tweet will be classified as the closest to its own patterns. However, in the optimization set, the optimal accuracy we obtained was for $\alpha = 0.03$.

Finally, for $\beta_1, \cdots, \beta_{N_L}$, we tried different combinations maintaining the condition

$$\beta_1 \leq \cdots \leq \beta_{N_L}. \tag{5}$$

The observed results are not very different for all the combinations we tried. The optimal performances we obtained were observed when

$$\beta_n = \frac{n-1}{n+1}. \tag{6}$$

The final values of parameters we set for pattern-related features is as follows:

$$\begin{cases} N_{occ} & = & 2, \\ L_{min} & = & 3, \\ L_{max} & = & 10, \\ \alpha & = & 0.03, \\ \beta_n & = & (n-1)/(n+1) \qquad \forall n \in \{3, \ldots, 10\}. \end{cases}$$

In the next section, we evaluate the model we built and present the results of our experiments.

IV. Experimental Results

Once the features are extracted, we proceed to our experiments. The Key Performance Indicators (KPIs) used to evaluate the approach are:

- **Accuracy:** which represents the overall correctness of classification. It measures the fraction of all correctly classified instances over the total number of instances.
- **Precision:** which represents the fraction of retrieved sarcastic tweets that are relevant. It measures the number of tweets that have correctly been classified as sarcastic over the total number of tweets classified as sarcastic.
- **Recall:** which represents the fraction of relevant sarcastic tweets that are retrieved. It measures the number of tweets that have successfully been classified as sarcastic over the total number of sarcastic tweets.

We run the classification using the classifier *"Random Forest"* [11]. We first checked the performances of classification of each set of features apart.

A. Performances of Each Set of Features

1) During cross-validation: Fig. 3 shows the performances of classification during cross-validation. We notice that the performances of the pattern-related features is very high during cross-validation. This is because the value of α as chosen makes each tweet in the training set the closest to itself.

On the other hand, we notice that the syntax-related features performances are very low. One reason is that is the low presence of these features in the dataset. TABLE V shows the existence rate of each of the features in the training set. However, the precision given by this set of features, and which exceeded 65% shows the importance of such features to detect sarcastic components.

Punctuation-related features and sentiment-related features have higher prediction rate. Furthermore, the precision of sentiment-based features is remarkably high. This can be explained by the fact that tweets having contrasting emotional

TABLE V
RATIO OF PRESENCE OF SYNTAX-RELATED FEATURES IN THE TRAINING SET

	True	False	Ratio
Presence of uncommon words	243	5757	4.05%
Presence of common sarcastic patterns	115	5885	1.92%
Presence of interjections	410	5590	6.83%
Presence of laughters	224	5776	3.73%

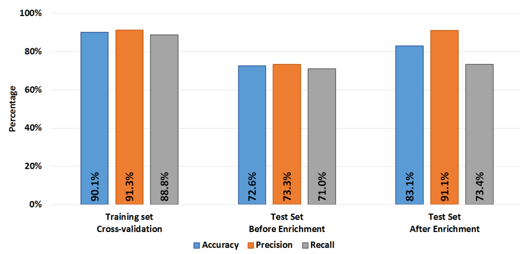

Fig. 5. Accuracy of classification using all features during training set-cross-validation and on the test set

content are likely to be sarcastic.Thus, if detected, they would be classified as sarcastic.

2) On a test set: Fig. 4 shows the performances of classification on our test set. Performance of the classification on unknown data is clearly lower than that during cross-validation. However, we notice that the sets of features that have the highest merit during cross-validation, are the same ones that have the highest merit during the classification of test set tweets. Pattern-related features have higher performances. Accuracy and precision have very close values. This can be explained by the fact that, contrarily other sets of features, which check the existence on some characteristics related to sarcasm in the tweets, patterns are extracted from both sarcastic and non-sarcastic tweets, and the closeness to these patterns is checked.

B. Overall Performances of the Proposed Approach

Together, the sets of features perform better than each one by itself. Fig. 5 show the performance of the proposed approach when all the features are used.

During cross validation, both the accuracy and precision are higher than 90%. The recall is lower than 89%. More interestingly, the accuracy obtained for the test set, before enrichment of the patterns, exceeds 72% with a precision higher than 73%. This shows that, if combined, the different sets of features, perform better. The enrichment process added more potential to the approach and increased the accuracy of the classification noticeably. The precision remarkably increased compared to that without enrichment.

To measure the potential of our method, we consider the approach proposed by Riloff et al. [10] as well as the *n*-gram approaches as our baseline. In addition to the aforementioned KPIs, we define a fourth one, which is the F-Score defined as follow:

$$F_1 = 2 \cdot \frac{precision \cdot recall}{precision + recall}. \tag{7}$$

It combines the precision and recall, therefore it represents a more reliable KPI to compare different approaches.

The results of the comparison of our approach to the baseline ones is given by TABLE VI. Our proposed approach clearly outperforms the baseline ones, for the used dataset: not only it has a higher accuracy and precision, our method's F-Score is neatly higher than the baseline ones. Although it performs well when detecting a specific type of sarcasm, the approach proposed by Riloff et al. [10], performs poorly in our dataset since most of the sarcastic tweets do not fall in the type of sarcasm where a positive sentiment contrasts with a negative situation.

TABLE VI
PERFORMANCE OF THE PROPOSED APPROACH COMPARED TO THE BASELINE ONES

	Accuracy	Precision	Recall	F-Score
n-grams	65.9%	82.2%	40.6%	65.9%
Riloff et al. [10]	59.4%	65.0%	40.8%	50.1%
Proposed approach	**83.1%**	**91.1%**	**73.4%**	**81.3%**

V. CONCLUSION

In this work, we proposed a new method to detect sarcasm in Twitter. The approach makes use of Part-of-Speech-tags to extract patterns characterizing the level of sarcasm of tweets. It has shown good results, though might have even better results if we use a bigger training set since the number of patterns we extracted from the current one is 346 541, which might not cover all possible sarcastic patterns.

In a future work, we will study how to use the output of the current one to enhance the performances of sentiment analysis and opinion mining.

ACKNOWLEDGMENT

The research results have been achieved by ”Cognitive Security: A New Approach to Securing Future Large Scale and Distributed Mobile Applications,” the Commissioned Research of National Institute of Information and Communications Technology (NICT) , JAPAN.

REFERENCES

[1] J. M. Soler, F. Cuartero, and M. Roblizo, “Twitter as a tool for predicting elections results,” in *Proc. IEEE/ACM ASONAM,* pp. 1194–1200, Aug. 2012.

[2] S. Homoceanu, M. Loster, C. Lofi, and W-T. Balke, “Will I like it? Providing product overviews based on opinion excerpts,” in *Proc. IEEE CEC,* pp. 26–33, Sept. 2011.

[3] U. R. Hodeghatta, “Sentiment analysis of Hollywood movies on Twitter,” in *Proc. IEEE/ACM ASONAM,* pp. 1401–1404, Aug. 2013.

[4] D. Maynard, and M. Greenwood, “Who cares about sarcastic tweets? Investigating the impact of sarcasm on sentiment analysis,” in *Proc. 9th Int. Conf. Language Resources Evaluation,* pp. 4238–4243, May 2014.

[5] A. Rajadesingan, R. Zafarani, and H. Liu, “Sarcasm detection on Twitter: A behavioral modeling approach,” in *Proc. 18th ACM Int. Conf. Web Search Data Mining,* pp. 79–106, Feb. 2015.

[6] F. Jr. Sting, *The meaning of irony.* New York, State University of NY, 1994.

[7] J. Tepperman, D. Traum, and S. S. Narayanan, “Yeah right: Sarcasm recognition for spoken dialogue systems,” in *Proc. InterSpeech,* pp. 183–184, Sept. 2006.

[8] C. Burfoot and T. Baldwin., “Automatic satire detection: Are you having a laugh?” in *Proc. ACL-IJCNLP,* pp. 161–164, Aug. 2009.

[9] D. Davidov, O. Tsur, and A. Rappoport, “Semi-supervised recognition of sarcastic sentences in Twitter and Amazon,” in *Proc.14th Conf. Computational Natural Language Learning*, pp. 107–116, July 2010.

[10] E. Riloff, A. Qadir, P. Surve, L. De Silva, N. Gilbert, and R. Huang, “Sarcasm as contrast between a positive sentiment and negative situation,” in *Proc. Conf. Empirical Methods Natural Language Processing,* pp. 704–714, Oct. 2013.

[11] L. Breiman, “Random Forest,” *Machine Learning,* vol. 45, no. 1, pp. 5–32, Jan. 2001.