The complementary nature of different NLP toolkits for Named Entity Recognition in social media

Filipe Batista[1] and Álvaro Figueira[1]

CRACS / INESC TEC and University of Porto,
Rua do Campo Alegre, 1021/1055,
4169-007 Porto, Portugal
filipe.batista@fe.up.pt, arf@dcc.fc.up.pt

Abstract. In this paper we study the combined use of four different NLP toolkits — Stanford CoreNLP, GATE, OpenNLP and Twitter NLP tools — in the context of social media posts. Previous studies have shown performance comparisons between these tools, both on news and social media corporas. In this paper, we go further by trying to understand how differently these toolkits predict Named Entities, in terms of their precision and recall for three different entity types, and how they can complement each other in this task in order to achieve a combined performance superior to each individual one. Experiments on two publicly available datasets from the workshops WNUT-2015 and #MSM2013 show that using an ensemble of toolkits can improve the recognition of specific entity types - up to 10.62% for the entity type PERSON, 1.97% for the type LOCATION and 1.31% for the type ORGANIZATION, depending on the dataset and the criteria used for the voting. Our results also showed improvements of 3.76% and 1.69%, in each dataset respectively, on the average performance of the three entity types.

Keywords: Named entity recognition; Social medria; Ensemble of NLP toolkits; Text-mining; Machine learning

1 Introduction

Following the rapid growth of social networks, Named Entity Recognition (NER) on texts from social media sources such as Twitter, has received increasing attention over the last decade.

While Named Entity Recognition has been studied for a long time, and some tools achieved what could be considered very good results, most of these tools were essentially tested on formal texts, as news articles, scientific articles, or books. Two state-of-the-art tools for this task are Stanford CoreNLP [9], a complete NLP pipeline widely used as a NER reference and the OpenNLP library, a "machine learning based toolkit for the processing of natural language text" [11].

However, when applied to texts from social media, "out-of-the-box" tools tend to show significant decrease in performance [14], mainly due to the informal nature of those texts, which is expressed by the absence of context, the lack of proper punctuation, wrong capitalization, the use of characters to represent *emoticons*, spelling errors and even the use of different languages in the same text.

To overcome these problems different studies have been conducted at different levels in the NLP pipeline to deal with specific problems such as tokenization on tweets [8] or capitalization restoration [10].

Entire pipelines with the specific purpose of Named Entity Recognition on tweets have also been proposed. Twitter NLP tools, proposed by Ritter et al. [14], is an example of a rebuilt NLP pipeline, with part-of-speech tagging, chunking and named-entity recognition. Another example of an entire social media NLP pipeline is TwitIE [3], a sequence of modules including language identification, tokenization, spelling and orthographic corrector, Stanford POS tagger adapted to Twitter, and a Named Entity Recognizer.

Most of these tools implement different algorithms to perform NER, and their performances on different entity types varies significantly [1, 15, 12]. Moreover, it is common to find disagreements between these tools regarding specific tokens and their corresponding named entity. Therefore, it is our intuition that the simultaneous use of different toolkits might help achieve better results than using them separately. Apart from the obvious benefit that some of these toolkits predict different sets of entity types, complementing each other that way, we will analyze, for a standard set of core entities (PERSON, LOCATION, ORGANIZATION), if a ponderation between toolkits reveals to be beneficial.

In this regard, along this paper we will try to answer the following research questions (regarding the English language only):

- Can an ensemble of different toolkits achieve overall higher NER performance than any of the involved toolkits, independently, for the same task?
- What is the best way to resolve conflicts/disagreements between different toolkits regarding their entity predictions?

The remainder of this paper is presented as follows: in Section 2, we review previous toolkit comparisons and the conclusions regarding their individual performances per each entity type; in Section 3 we describe our experimental setup, including details on the datasets used, brief descriptions of the toolkits used and the necessary steps taken to obtain their results for our analysis, the ensemble itself and the different voting protocols tested, as well as the performance measures used; in Section 4 we present the results and discuss them in detail; finally, in Section 5 we present our conclusions and ideas for future work.

2 Related work

While ensemble methods have been proposed in literature for the task of NER, usually these methods were applied at the level of the machine learning algorithms, rather than at the level of ready-to-use toolkits. An example of previous

use of ensemble methods for NER, proposed by Wu, Chia-Wei, et al. [17], consisted in applying a memory-based ensemble method on Chinese datasets to achieve better results than using individual classifiers. Another example of the same use was proposed by S. Saha and A. Ekbal [16], once again showing that combining different learning algorithms can improve the performance of Named Entity Recognition.

Differently from these works, in this study we will not be implementing algorithms from scratch, but instead using widely recognized toolkits which provide already solid out-of-the-box performances, presumably optimized by many contributors over the years. As a first approach, we chose Stanford CoreNLP[9], a reference toolkit in NER, OpenNLP and the Twitter specific NLP pipelines: "Twitter NLP tools" by Ritter et al.[14] and TwitIE [3].

To the best of our knowledge, there have been few attempts to simultaneously use different out-of-the-box toolkits to perform Named Entity Recognition on social media texts. The idea of combining toolkits was applied in one of the submissions to the Making Sense of Microposts challenge in 2013 [4]. In this study the authors combined different toolkits using machine learning techniques, and their results showed that several classification models could achieve better results than the best individual tools [4]. In our work, besides machine learning techniques we also tried manually defining protocols for the ensembles' voting system, and our experiments were conducted on a different set of toolkits, combining social media-oriented NLP toolkits with general text toolkits.

A more recent example of combining toolkits used two different toolkits (SpaCy and CoreNLP) together to create an hybrid NER tool [7]. This hybrid tool was tested on formal texts rather than social media texts, as in our study.

3 Experimental Setup

3.1 Datasets

For comparison purposes, every toolkit used equally pre-tokenized datasets, following Ritter's [14] tokenization method. We also chose to focus only on the entities PERSON, LOCATION and ORGANIZATION. The reason for this choice was that these entities are the only three entities detected by all the toolkits tested.

For the first experiment, an original dataset of tweets from our project (citation removed for anonymity) was partially used. This dataset consists of 840 entries: 420 tweets, 107 Facebook posts and 313 Facebook comments, retrieved by a crawler about 6 topics highly discussed in 2016: "Refugees Syria", "Elections US", "Olympic Games" , "Terrorism" , "Daesh" and "Referendum UK EU".

This original dataset was then tokenized. Therefore, instead of 840 entries, the tokenized dataset had 28172 entries (one per token). From the tokenized dataset, a subset of 3474 tokens was extracted. The final dataset contains one token per row, and one entity for each token. The ground truth for this dataset was manually annotated by the authors of this paper.

For the second experiment, a dataset from WNUT NER - Workshop on Noisy User-generated Text [2]- was used. This dataset used the same format seen in Twitter NLP tools by Ritter et al., including less common entity types that were dropped for the purpose of this study, which focuses only on the 3 core entities PERSON, LOCATION and ORGANIZATION.

In the third experiment, we tested the dataset from the 3rd workshop on 'Making Sense of Microposts' (#MSM13) [4], which took place in 2013. It is important to note that for this dataset we used the PTBTokenizer available as part of the CoreNLP libraries. The reason for this choice was that in the conversion process we had to tokenize both the entities and the text of the tweets, and for the tokenizations to match we needed a deterministic tokenizer.

Finally, in the last experiment, we used machine learning algorithms instead of manually defined voting rules. Dataset 1 was rather small for the purpose, and Dataset 2 had to suffer multiple conversions as will be explained in the next section. Therefore, we decided to partially use Dataset 3.

Therefore, our testing datasets were:

- **Dataset 1**: Our dataset - 3474 entries (tokens)
- **Dataset 2**: WNUT NER - 48 862 entries (tokens)
- **Dataset 3**: #MSM2013 - 62 494 entries (tokens)
- **Dataset 4**: Subset of #MSM2013 - 10 000 entries (tokens)

3.2 Toolkits and Data preparation

Stanford CoreNLP[1] was run using the default toolkit via command line [9]. This toolkit accepts as input format the tokenized text and the output format in a tab formatted file, convenient for this study.

Since there was not enough labeled data for training our own model, as the data was manually annotated by the authors and that is a very costly task time-wise, we used the "3 class model" provided by CoreNLP, which was trained on both MUC 6 and MUC 7 training data sets with some additional data (including ACE 2002 and other generated data).

GATE using TwitIE plugin[2] provides a graphical interface which was used in this study to run the TwitIE[3] pipeline, available as part of the Twitter plugin.

The output format consists in surrounding any detected entity with XML tags. In order to convert this type of output to the tab separated format, a small script using regular expressions was written in Python.

While GATE is able to detect many other entity types, we used only the three core entities (PERSON, LOCATION and ORGANIZATION). We used the default configurations of the TwitIE pipeline.

[1] https://stanfordnlp.github.io/CoreNLP/
[2] https://gate.ac.uk/download/

Twitter NLP tools[3] were run via command line, following the usage presented in the Twitter NLP tools Github repository.

Twitter NLP tools [14] output is by default in the IOB format [13] (B for beginning of a Named Entity (NE), I for inside an NE, O for outside of NE), and the "token/ENTITY" format. The IOB format was dropped, so instead of B-ENTITY and I-ENTITY we opted to use ENTITY only. Besides, 2 entity types were converted: COMPANY to ORGANIZATION, and GEO-LOCATION to LOCATION, while all the remaining entity types (except PERSON) were simply dropped.

We used this tool as is, without any re-training or tuning.

OpenNLP[4] is a Java library which supports several common NLP tasks, including Named Entity Recognition.

OpenNLP can be used directly as a tool, or via its API. We decided to use the API in a small Java project in order to easily output the entities to the tab-separated format.

We used the pre-trained models for the OpenNLP 1.5 series, for each entity type used.

3.3 Ensemble voting methods

In order to study the viability of a NER toolkit ensemble, all the outputs from the previous toolkits previously mentioned were merged to a single *comma-separated values* file, one column for the tokens, another column for the ground truth entities, and one column for each of the entities predicted from each toolkit.

The first step was to compute the precision, recall and F1 measure for each toolkit individually, using the ground truth obtained by manual labeling.

The second step was to define different voting protocols to resolve the conflicts between the different toolkits predictions.

Finally, we used different machine learning algorithms taking as input features the predictions of each tool.

Protocol use 1: A token is tagged with entity type A if and only if at least one of the following conditions are met:

- 50% of the toolkits predicted entity type A and the other 50% did not predict any entity type
- At least 75% of the toolkits predicted entity type A

Protocol use 2: A token is tagged with entity type A if and only if at least one of the following conditions are met:

- 50% of the toolkits predicted entity type A and the other 50% did not agree on any other entity between them.
- At least 75% of the toolkits predicted entity type A

[3] https://github.com/aritter/twitter_nlp
[4] https://opennlp.apache.org/

Machine learning approach: The models for predicting the combined output were obtained by running each of the following ML algorithms on a training set, with 10-fold cross validation, and then tested on an independent test set.

Both the train and test sets were subsets, each of 10 000 entries, of the previously mentioned MSM2013 dataset. Every ML experiment was performed in RapidMiner Studio. The algorithms used were Naïve Bayes, Random Forest, k-nearest neighbors (k-NN) and Neural Network. The features used consisted of the 4 individual outputs of each tool.

3.4 Performance evaluation

Performance in classification systems is measured by comparing the output of a classifier on unseen data with a golden standard - made by human annotators, and assumed as correct. A certain prediction can be either Positive or Negative, and according to the golden standard, that prediction can be True or False.

There are different ways of counting true positives. In the strict way, only exact matches are considered, while in the lenient way partially correct (shorter, longer, overlapping at either end) are also considered as correct [6]. In this study we chose to use the lenient way.

The metrics we used to measure performance of classification tasks include Precision, Recall and F1-score.

Although it is important to understand how the system is behaving, recall and precision measures are not sufficient when used independently, meaning that knowing recall without knowing precision, or vice-versa, does not provide enough information about the performance of the system. The most common way to combine Recall and Precision in one single measure is the F-measure.

F-measure: Calculates the harmonic mean of precision and recall. The relative importance (weight) of each component (precision and recall) is controlled by the β parameter (higher values of β mean more weight on recall) [5].

$$F_\beta = \frac{(\beta^2 + 1) \times P \times R}{\beta * P + R} \equiv F_1 = \frac{2 \times P \times R}{P + R}, \beta = 1 \tag{1}$$

F1-score: Used when both measures have the same importance ($\beta = 1$)

4 Experimental results

In this Section we explore the performances of each toolkit and compare them to the ensembles' performances using different protocols and datasets. $Ensemble_n$ (E_n) will be the notation used to refer to the ensemble using protocol number n, previously defined. Bold will be used to highlight the highest results.

For the first dataset we provide a more extensive analysis, providing not only the F1-score results but also the Precision and Recall. For the other datasets we present only the F1-scores and discuss them briefly, given that the results of precision and recall led to the same conclusions in every experiment.

For dataset 4 more experiments were added using ML algorithms.

4.1 Dataset 1 - Our dataset

Table 1. Precision, Recall and F_1 scores on Dataset 1

	Person			Location			Organization			Average		
	Precision	Recall	F1	Precision	Recall	F1	Precision	Recall	F1	Precision	Recall	F1
CoreNLP	58.18	**80**	67.37	96.92	65.63	78.26	**100**	16.67	28.57	85.03	54.1	58.07
TwitIE	67.5	67.5	67.5	89.77	**82.29**	85.87	84.62	**30.56**	**44.90**	80.63	**60.12**	**66.09**
TwitterNLP	37.93	55	44.90	88.33	61.46	72.84	80	5.56	10.39	69.11	40.67	42.71
OpenNLP	80.77	52.5	63.63	88.33	55.21	67.95	37.5	16.67	23.08	63.13	41.46	51.55
Ensemble1	**85.71**	75	**80.00**	**98.61**	73.96	84.52	**100**	18.06	30.59	**94.79**	55.67	65.04
Ensemble2	73.17	75	74.07	96.30	81.25	**88.14**	**100**	18.06	30.59	89.82	58.10	64.27

F1-score analysis

Ensemble 1 :

Looking at Table 1 we can see that $Ensemble_1$ achieved the highest F1-score for detecting the entity PERSON. In terms of LOCATION and ORGANIZATION entities, while $Ensemble_1$ was better than CoreNLP, Twitter NLP tools and OpenNLP, it did not perform better than TwitIE.

On average, TwitIE still achieved the best F1 measure, with 66%, followed immediately by the ensemble, which achieved an average F_1 of 65%. This is not surprising, given that TwitIE was, among all the 4 toolkits, the one to achieve better results for every entity type.

Nevertheless, it was possible to achieve an improvement of 12.5% on the detection of the entity Person by using $Ensemble_1$.

Ensemble 2 :

$Ensemble_2$ also achieved the best F_1 score for the entity type PERSON when compared to any other toolkit individually, however its F_1 score was lower than $Ensemble_1$.

On the other hand, $Ensemble_2$ scored higher than $Ensemble_1$ and any other toolkit and in terms of detecting the entity LOCATION.

On average, $Ensemble_2$ was worse than $Ensemble_1$, which in turn was worse than TwitIE.

While for this dataset our ensembles did not outperform the best individual toolkit, TwitIE, there were still visible improvements in specific entity types, namely PERSON and LOCATION.

Recall analysis

Ensemble 1 :

In terms of recall, it is possible to see in Table 1 that the $Ensemble_1$ ranked second for every entity type. The toolkit able to detect more PERSON entities was Stanford CoreNLP, while TwitIE was the toolkit to achieve higher recall for the entities LOCATION and ORGANIZATION.

Ensemble 2 :

$Ensemble_2$ ranked better than $Ensemble_1$ for the entity type LOCATION, but scored the same for PERSON and ORGANIZATION.

The fact that protocol 2 was less strict than protocol 1 is the likely reason for the improve in recall from $Ensemble_1$ to $Ensemble_2$.

Precision analysis

Ensemble 1 :

In terms of precision, $Ensemble_1$ ranked first for every entity type, as we can see in Table 1. This result makes sense and indicates that using this protocol helped significantly in detecting entities efficiently, by eliminating predictions with less than a certain level of confidence (see protocol 1).

Ensemble 2 :

$Ensemble_2$ overall precision dropped when compared to $Ensemble_1$, 12.54% on PERSON and 2.31% on ORGANIZATION. Once again it makes sense that reducing the strictness of the protocol would likely reduce the precision.

4.2 Dataset 2 - WNUT NER

Table 2. F1-scores on Dataset 2

	PERSON	LOCATION	ORGANIZATION	Avg.
CoreNLP	56.62	32.5	20	36.37
TwitIE	59.95	**48.14**	38.23	48.77
TwitterNLP	52.78	34.9	**45.12**	44.27
OpenNLP	43	34.79	6.59	28.13
$Ensemble_1$	**70.57**	41.45	42.37	51.46
$Ensemble_2$	70.44	44.53	41.73	**52.53**

In Table 2 we can see that for this dataset the results were generally low for all the toolkits, when compared to the performances obtained from the other datasets tested. Since this dataset used Twitter NLP tools format, it had to

suffer the same conversion explained in Section 3.2.3, which probably led to the worse results.

Nevertheless, we can see that both Ensembles achieved better F-scores on average than any other toolkit alone, which is the question we sought to answer in this work.

4.3 Dataset 3 - #MSM2013

Table 3. F1-scores on Dataset 3

	PERSON	LOCATION	ORGANIZATION	Avg.
CoreNLP	69.20	54.18	27.09	50.16
TwitIE	77.06	**67.96**	43.95	62.99
Ritter	55.04	41.91	16.18	37.71
OpenNLP	55.40	47.68	25.47	42.85
$Ensemble_1$	79.93	62.20	41.37	61.17
$Ensemble_2$	**82.36**	66.42	**45.26**	**64.68**

Looking at Table 3 it is possible to see that once again ensemble 2 performed better on average than any other toolkit individually. $Ensemble_1$, while not better than TwitIE on average still performed reasonably well with only 1.82% less F1-score.

Also, once again, both Ensembles outperformed every toolkit on the entity type PERSON, and $Ensemble_2$ on the entity type ORGANIZATION.

4.4 Dataset 4 - Subset of #MSM2013

Table 4. F1-scores on Dataset 4

	PERSON	LOCATION	ORGANIZATION	Avg.
CoreNLP	54.21	65.64	40.20	53.35
TwitIE	71.13	82.20	**61.02**	71.45
TwitterNLP	51.73	55.59	15.85	41.06
OpenNLP	52.80	63.05	41.20	52.35
$Ensemble_1$	80.08	77.07	53.57	70.24
$Ensemble_2$	**81.26**	81.45	57.74	73.48
Random Forest	80.68	82.58	51.4	71.55
Naïve Bayes	80.88	83.82	57.37	**74.02**
kNN, k=3	75.09	**84.17**	47.76	69.00
kNN, k=10	80.68	82.53	53.16	72.12
Neural net	79.62	83.71	58.68	74.00

We extracted a subset of 20000 entries (i.e. tokens) from #MSM2013 and split it into two equally sized datasets for training and testing purposes.

Looking at Table 4, we can see that Naïve Bayes was the best method on average (74.02% F1), followed by the Neural Network (74.00% F1), and our manually defined $Ensemble_2$ (73.48% F1). Every ML algorithm that we experimented, except kNN with k=3, performed better than TwitIE (the best among the tools).

In terms of individual entity types, our $Ensemble_2$ was the best for PERSON, achieving 81.26% of F1, an improvement of 10.13% against TwitIE. For LOCATION, the best achieved was 84.17%, using kNN with k=3, an increase of 1.97% (again against TwitIE). For the entity type ORGANIZATION none of our ensembles was able to perform better than TwitIE.

An interesting fact to note is that the best ensemble on average (Naïve Bayes) was not the best ensemble for any specific entity type alone.

4.5 Results summary

Differently from results previously shown in literature [14, 12], in our experiments Twitter NLP tools achieved overall worse performances than other toolkits across all the 3 tested datasets. We believe this performance difference was related to the way we converted the output of this toolkit for our study. We expose our rationale for this.

Firstly, Twitter NLP recognizes multiple entity types, but those entities do not include ORGANIZATION nor LOCATION. Instead, they include COMPANY and GEO-LOCATION, which were converted directly to ORGANIZATION and LOCATION. We are aware that the former is probably not optimal, since a company does not need to be an organization and vice-versa.

Secondly, there is also the fact that Twitter NLP tools recognizes other entity types that we decided to ignore in this study (such as SPORTSTEAM, BAND, and MOVIE) which could be, in some cases, sub-categories of more general entity types (for example a SPORTSTEAM could be seen as an ORGANIZATION/COMPANY). Therefore, ignoring such entity types could be another reason for the comparatively worse results obtained by Twitter NLP tools in our experiments.

Finally, we did not include optional features based on POS and chunk tags, which leads to faster but lower quality results [14].

For the first dataset, while TwitIE has remained better than both ensembles on average, we witnessed a positive boost of PERSON detection using Protocol 1, achieving more 12.5% F1-score than the best individual toolkit (TwitIE with 67.5%), and a boost in LOCATION detection using Protocol 2, achieving more 2.27% F1-score than the best individual toolkit (TwitIE with 85.871%).

On the second dataset, both ensembles have beaten the best individual toolkit. The performance boost was very noticeable on the entity type PERSON (up to 10.62%), and the ensembles managed to keep a reasonable performance on the detection of ORGANIZATIONS (42.37% and 41.73% respectively), given that two of the toolkits (CoreNLP and OpenNLP) achieved very low results for this entity type (20% and 6.59% respectively).

In our third experiment, the boost on the entity type person remained noticeable for both ensembles (2.87% and 5.3% higher than the best toolkit). $Ensemble_2$ performed better on average than any other toolkit, achieving 1.69% higher F1-score than TwitIE, the best individual toolkit with 62.99% F1-score.

In terms of precision and recall, the conclusions were the same as for every dataset: the stricter protocol ($Ensemble_1$) had less recall but more precision than the less strict protocol ($Ensemble_2$).

Finally, our last experiment showed that there were some ML algorithms able to outperform TwitIE and even our $Ensemble_2$, namely Naive Bayes and the Neural Network.

5 Conclusions and Future Work

The first conclusion of this study is that using an ensemble of toolkits with a voting system can improve the performance of NER on tweets, answering the first question of our research.

As for the second question, we can say that both manually defined protocols were, to some extent, naïve yet they achieved promising results. This indicates that a more refined protocol will probably improve these results even further. It proves to be false, this approach could still be used with a combination of both protocols for the entities PERSON and LOCATION, and keeping ORGANIZATION predicted by TwitIE. We also showed that using machine learning algorithms for predicting entities based on the outputs of each toolkit is viable.

As future work we intend to train most of the toolkits using a training dataset, instead of using already trained models, since in some of these toolkits the models were not trained on social media texts. We also want to set up an "out-of-the-box" multi-threading ensemble NER toolkit, available and easy to use for anyone intending to extract entities from social media posts.

In terms of the results, a deeper analysis could be conducted in the future in order to better understand the behaviours observed in each toolkit, as well as the differences across corpora. Statistical tests would also be interesting to check if improvements between tools are statistically significative or not.

For the machine learning algorithms, more complex features and hyperparameters could be tried and analyzed. It would also be interesting to apply the ML approach to different datasets and compare the results.

Acknowledgments

This work is supported by the ERDF European Regional Development Fund through the COMPETE Programme (operational programme for competitiveness) and by National Funds through the FCT (Portuguese Foundation for Science and Technology) within project "Reminds/UTAP-ICDT/EEI-CTP/0022/2014".

References

1. S. Atdağ and V. Labatut. A comparison of named entity recognition tools applied to biographical texts. In *Systems and Computer Science (ICSCS), 2013 2nd International Conference on*, pages 228–233. IEEE, 2013.
2. T. Baldwin, M. C. De Marneffe, B. Han, Y.-B. Kim, A. Ritter, and W. Xu. Shared tasks of the 2015 workshop on noisy user-generated text: Twitter lexical normalization and named entity recognition. In *Proceedings of the Workshop on Noisy User-generated Text (WNUT 2015), Beijing, China*, 2015.
3. K. Bontcheva, L. Derczynski, A. Funk, M. A. Greenwood, D. Maynard, and N. Aswani. Twitie: An open-source information extraction pipeline for microblog text. In *RANLP*, pages 83–90, 2013.
4. A. E. Cano Basave, A. Varga, M. Rowe, M. Stankovic, and A.-S. Dadzie. Making sense of microposts (# msm2013) concept extraction challenge. 2013.
5. A. Clark, C. Fox, and S. Lappin. *The handbook of computational linguistics and natural language processing*. John Wiley & Sons, 2013.
6. Gate.ac.uk - wiki/twitie.html. https://gate.ac.uk/wiki/twitie.html. (Accessed on 06/10/2017).
7. R. Jiang, R. E. Banchs, and H. Li. Evaluating and combining named entity recognition systems. *ACL 2016*, page 21, 2016.
8. G. Laboreiro, L. Sarmento, J. Teixeira, and E. Oliveira. Tokenizing micro-blogging messages using a text classification approach. In *Proceedings of the fourth workshop on Analytics for noisy unstructured text data*, pages 81–88. ACM, 2010.
9. C. D. Manning, M. Surdeanu, J. Bauer, J. R. Finkel, S. Bethard, and D. McClosky. The stanford corenlp natural language processing toolkit. In *ACL (System Demonstrations)*, pages 55–60, 2014.
10. K. Nebhi, K. Bontcheva, and G. Gorrell. Restoring capitalization in# tweets. In *Proceedings of the 24th International Conference on World Wide Web*, pages 1111–1115. ACM, 2015.
11. Apache opennlp. https://opennlp.apache.org/. (Accessed on 06/10/2017).
12. A. Pinto, H. Gonçalo Oliveira, and A. Oliveira Alves. Comparing the performance of different nlp toolkits in formal and social media text. In *OASIcs-OpenAccess Series in Informatics*, volume 51. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2016.
13. L. A. Ramshaw and M. P. Marcus. Text chunking using transformation-based learning. In *Natural language processing using very large corpora*, pages 157–176. Springer, 1999.
14. A. Ritter, S. Clark, O. Etzioni, et al. Named entity recognition in tweets: an experimental study. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 1524–1534. Association for Computational Linguistics, 2011.
15. K. J. Rodriquez, M. Bryant, T. Blanke, and M. Luszczynska. Comparison of named entity recognition tools for raw ocr text. In *KONVENS*, pages 410–414, 2012.
16. S. Saha and A. Ekbal. Combining multiple classifiers using vote based classifier ensemble technique for named entity recognition. *Data & Knowledge Engineering*, 85:15–39, 2013.
17. C.-W. Wu, S.-Y. Jan, R. T.-H. Tsai, and W.-L. Hsu. On using ensemble methods for chinese named entity recognition. In *Proceedings of the 5th SIGHAN Workshop on Chinese Language Processing*, pages 142–145, 2006.