Multilingual Named Entity Recognition on Spanish-English Code-switched Tweets using Support Vector Machines

Daniel Claeser
Fraunhofer FKIE
Fraunhoferstraße 20
53343 Wachtberg
daniel.claeser@
fkie.fraunhofer.de

Samantha Kent
Fraunhofer FKIE
Fraunhoferstraße 20
53343 Wachtberg
samantha.kent@
fkie.fraunhofer.de

Dennis Felske
Fraunhofer FKIE
Fraunhoferstraße 20
53343 Wachtberg
dennis.felske@
fkie.fraunhofer.de

Abstract

This paper describes our system submission for the ACL 2018 shared task on named entity recognition (NER) in code-switched Twitter data. Our best result (F1 = 53.65) was obtained using a Support Vector Machine (SVM) with 14 features combined with rule-based post-processing.

1 Introduction

Named Entity Recognition (NER) is a part of information extraction and refers to the automatic identification of named entities in text. The ACL 2018 shared task invited participants to extract and classify the following named entities in code-switched data obtained from Twitter: person, location, organization, group, title, product, event, time, and other (Aguilar et al., 2018). The Tweets are either Spanish-English or Modern Standard Arabic-Egyptian, and participants were free to participate in either language pair. This paper describes our system for the Spanish-English NER task.

This particular NER task is challenging for two reasons. Firstly, NER has proved to be more difficult for Tweets than for longer text, as accuracy in NER ranges from 85-90% on longer texts compared to 30-50% on Tweets (Derczynski et al., 2015). One of the reasons for this difference is that Tweets contain non-standard spelling, unusual punctuation, and unreliable capitalization. Fromheide et al. (2014) also point out that another difficulty stems from the rapidly changing topics and linguistic conventions on Twitter. The 2015 and 2016 shared tasks for NER on Noisy User-generated Text (W-NUT) reported F1 scores between 16.47 and 52.41 for identifying 10 different NE categories (Baldwin et al., 2015; Strauss et al., 2016). NER methods range from bidirectional long short-term memory (LSTM) (Limsopatham and Collier, 2016) and Conditional Random Fields (CRF) (Toh et al., 2015), to Named Entity Linking (Yamada et al., 2015). The second added challenge for the data in this task is that the Tweets contain English and Spanish named entities. Both languages need to be taken into account in order to accurately identify the NEs in this data.

2 Data sets

The organizers provided three different English-Spanish data sets: a training set, a development set, and a test set. The data consists of multilingual Spanish-English Tweets and contains NEs in both languages. Table 1 provides an overview of the data and the total number of NEs available in each of the sets (Aguilar et al., 2018). The gold standard for the test set was not distributed and we are therefore not aware of the distribution of NEs in the test set.

Data set	#Tweets	#Tokens	#NEs
Train	50,757	616,069	12,366
Development	832	9583	152
Test	15,634	183,011	-

Table 1: Number of Tweets, tokens and Named Entities in the Spanish-English data sets.

3 System description

We used scikit-learn 0.19 (Pedregosa et al., 2011) to train and test five different types of classifiers using eight-fold cross validation:

- Support Vector Machine (SVM) (Chang and Lin, 2011)
- Decision Trees (DT)

Proceedings of The Third Workshop on Computational Approaches to Code-Switching, pages 132–137
Melbourne, Australia, July 19, 2018.

- K-nearest Neighbors (KNN)
- AdaBoost (Ada) (Freund and Schapire, 1995)
- Random Forest (RF) (Breiman, 2001)

We trained the classifiers with different training corpus sizes of 80.000, 120.000, 200.000, 300.000 and 550.000 tokens, and we reserved 10% of each size for testing to avoid overfitting on the training data. The best classifier is the Support Vector Machine using the default scikit-learn parameters and a Radial Basis Function (RBF) Kernel, which is defined as

$$K(x, x') = exp(-\frac{\|x - x'\|^2}{2\sigma^2}) \quad (1)$$

The results are obtained using the pre- and post-processing steps that are described in further detail in sections 3.1 and 3.3.

3.1 Pre-processing

Early experiments showed that reducing the original tag set from two tags per category to one tag per category improved overall classification. 'B-LOC' refers to either the first word in a multi-word NE or a single word NE, and 'I-LOC' refers to any tokens in a multi-word NE that follows the initial 'B-' token. The information specific to the location of the NE within an NE sequence was removed and both tags are reduced to 'X-'. This improved classification performance as it reduced the number of different possible tags from 19 to 10 (one per NE category plus the "O" tag) and was easily reverted in the post-processing stage.

3.2 Feature selection

After testing numerous different features, and discarding ones such as 'proceeded by preposition or possessive pronoun' and 'difference in rank in the frequency dictionaries', we found that the features described below achieved the best result. There are three different types of features: token-centered features (1-5), context related features (6-9), and rank dictionary lookup features (10-14). To reduce dimensionality and computational workload, we condensed several mutually exclusive boolean features into common functions returning different integer values according to their outcome. For example, for the capitalization feature, rather than returning a boolean outcome for each of the four possible capitalization options (all lowercase, all uppercase length greater than 3, all uppercase length less than 3, first letter capitalized), they are combined into one feature that returns [0,1,2,3].

All rank features are obtained by sorting the corresponding list in order of frequency, with the most frequent occurrence in rank one. We normalized the ranks so that the value stays between 0 and 1, where 0 denotes the absence in the ranked lists and the closer the figure is to 1, the more highly ranked the token is.

For each feature, the possible outcomes that are inserted into the vector are provided in square brackets, where 'int' denotes the absolute rank, pairs of [0-1] boolean outcomes, and lists of numbers correspond to the exclusive outcomes of the function.

1. *Capitalization* – Check if the token is: all lowercase, all uppercase with length greater than 3, all uppercase with length less or equal to 3, or first letter only uppercase [0,1,2,3]
2. *Token length* - Returns the token-length [int]
3. *Contains non-ASCII* - Does the token contain non-ASCII characters? [0,1]
4. *Token first or last in Tweet* - Check if token is: first token, last token, or other [0,1,2]
5. *Token in majority language* - Check if the token language is the majority language of the tweet. Determined with a lexical lookup in frequency-ranked word lists for English and Spanish extracted from Wikipedia [0,1] (Claeser et al., 2018)
6. *Code-switch* - Returns true if the token's language is different from that of the token before [0,1] (Claeser et al., 2018)
7. *Previously tagged as, single-word* - The most common tag associated with the token in the training set. The outcome is either one of the nine NE categories or the token is not present in the training data [0-9]
8. *Previously tagged as, multi-word* - Same as above but for multi-word expressions [0-9]
9. *Is multi-word time* - Regular expressions to capture multi-word time expressions such as '23 de mayo' and 'april 29th' [0,1]
10. *Rank in family names* - Rank in list of last names extracted from the Wikipedia page

'Living people' [int]

11. *Rank in first names* - Rank in list of first names extracted from the Wikipedia page 'Living people' [int]
12. *Rank in cities list* - Rank in list of all United States census designated places (2016) ordered descending by population [int]
13. *Rank in Spanish Dictionary* - Rank in word list from Spanish Wikipedia [int]
14. *Rank in English Dictionary* - Rank in word list from English Wikipedia [int]

3.3 Post-processing

The first step in post-processing was to restore all the named entity categories that were simplified during the training of the SVM. All categories were reduced, for example, from 'B-PER' and 'I-PER' to X-PER in a pre-processing step, and were changed back to the original annotation.

The second step in post-processing was to address the misclassified multi-word tokens. For example, in a sequence of 'B-TITLE', 'I-TITLE', 'I-TITLE', if the middle token is misclassified as not being an NE, the tags shift to 'B-TITLE', 'O', 'B-TITLE' and the entire multi-word NE would therefore be misclassified.

To solve this issue, we used a dictionary lookup approach and compared possible multi-word NE sequences to lists of multi-word tokens based on the types of tokens present in the training data. The '-GROUP', '-PERSON' and '-OTHER' lists stems from Wikipedia, and the '-TITLE' list contains titles of video games available from Steam. We found post-processing to be most effective when the multi-word NE consisted of at least two tokens and was no longer than five tokens. We started by checking the longest NEs first, so that, for example, 'Tomb Raider' would not split the longer NE 'Rise of the Tomb Raider'. If a match was found in any of the lists, the tags gained from post-processing replaced those tagged by the SVM.

The final step addresses specific tokens that are very frequent in many of the categories and are therefore not learned correctly by the classifiers. The Spanish particle 'de', was often classified as an NE, but should have been classified as 'O'. So, if 'de' was tagged as an NE, but not proceeded by a token with a 'B-', the NE tag was removed. A similar rule applies to the article 'the', which was frequently tagged as 'O', and caused issues for multi-word NEs starting with 'the'. If 'the' is followed by a NE, the tag is switched to match the rest of the tokens in the multi-word sequence.

Classifier	Macro F1	FB1
Support Vector Machine	0.49	0.48
Decision Tree	0.61	0.43
KNN	0.50	0.44
Random Forest	0.59	0.45
AdaBoost	0.41	0.39

Table 2: Results for the train/test set without post-processing (Macro F1) and the held-out test set (FB1).

4 Results

Table 2 shows the best result obtained with a training size of 550.000 tokens for each of the five classifiers using 8-fold cross validation and the results of those five classifiers when applied to the held-out test data. Note that all figures are without post-processing. We only performed post-processing on the SVM to achieve the final result of 53.56. Table 2 shows that the Macro F1, which is the performance of the classifiers when splitting the training data into 90% train and 10% test, is higher for the Decision Tree, KNN and Random Forest classifiers. However, when applying the classifiers on the held-out test set, the FB1 is highest for the SVM. It is also clear that while a certain degree of overfitting is to be expected, it is much higher for the Decision Tree based classifiers than for the SVM. For the SVM, the Macro F1 and the FB1 is very similar, in contrast to the Decision Tree classifier where the difference is much larger.

Size	SVM	DT	KNN	RF	Ada
30k	0.25	0.33	0.34	0.40	0.23
80k	0.38	0.39	0.39	0.43	0.27
120k	0.43	0.45	0.44	0.48	0.28
200k	0.40	0.48	0.45	0.48	0.28
300k	0.43	0.56	0.49	0.58	0.33

Table 3: Performance of the classifiers with the different training sizes.

We also tested the classifiers with different sizes of training data. Table 3 provides the Macro

Category	Precision	Recall	FB1
EVENT	31.25%	11.11 %	16.39
GROUP	58.82 %	20.62 %	30.53
LOC	58.88 %	58.14 %	58.51
ORG	32.99 %	15.84 %	21.40
OTHER	100.00 %	3.45 %	6.67
PER	75.32 %	58.91 %	66.11
PROD	71.19 %	43.64 %	54.11
TIME	57.14 %	2.65 %	5.06
TITLE	22.45 %	14.93 %	17.93

Table 4: Results of best performing SVM per category including post-processing.

F1 from our train/test split data for the training sizes 30.000, 80.000, 120.000, 200.000, 300.000 and 550.000 tokens. The performance of all five classifiers improves significantly with increased amounts of training data.
The evaluation of the results per named entity category using the best performing SVM show that some of the categories were classified more accurately than others. The best results were obtained for person (66.11), location (58.51) and product (54.11). The most challenging categories were time (5.06) and other (6.67).

5 Discussion

The large variation in F1 per category, for example in '-TIME', is partly due to the inconsistent annotation of tokens. Table 5 below shows the days of the week present in the training data in both Spanish and English and all the tags associated with these tokens. It shows that all of these tokens are inconsistently annotated in that they are sometimes annotated as '-TIME' and sometimes annotated as 'O'. For example in Tweets (1) and (2) below, 'Happy Friday' is used in the same context, but is only tagged as 'B-TIME' in the first Tweet.

(1) Happy Friday Familia!!! #ElvacilonDeLaGatita #battingcage #HappyHour 17 ave NW 7 Calle http://t.co/fbPk0sER05

(2) RT @isazapata : Challenge yourself and move away from your comfort zone! Happy Friday!! http://t.co/OK320hNQ

Some variation in the annotation of tokens such as 'Friday' is to be expected, as the token may not always refer to a day of the week but a title or another type of named entity, but the SVM will discard the information from the feature vector if 'Friday' is 'tagged as 'O' more often than '-TIME'.

TOKEN	-TIME	O
lunes	21	74
monday	7	11
martes	23	51
tuesday	2	3
miercoles	7	20
wednesday	1	4
jueves	18	68
thursday	5	10
viernes	48	87
friday	13	35
sabado	6	21
saturday	6	9
domingo	34	63
sunday	16	18

Table 5: Number of times the tag '-TIME' occurs for the days of the week in the training Tweets.

Whilst training the classifiers, we noticed a large amount of variation in the results for the train/test data. To find out exactly how much the results fluctuate, we used the random split function in scikit-learn and split the training data into two chunks: 90% training and 10% testing and retrained the classifier with the new version of the training data. Consequently, the intermediate results for each of the classifiers was always on a different 10% test set. The difference between the best and the worst result can be up to an increase in macro F1 of 0.12 with the same classifier and the same size training set. The results also showed that by increasing the number of tokens in the training data, the performance of the classifiers improved.

To illustrate why this may be the case, table 6 below contains the number of overlapping NEs for three different splits for each training size. It shows the large amount of variance in the results depending on how the random split occurred. We counted all types that were tagged as an NE in the training data in total, compared to how many of those NEs were in the train and test sets. For example, for the first random 30.000 tokens split, there were 456 NEs in the training data, and 65 NEs in the training test set. A total of 17 NEs in the training test set were also present in the training data, meaning that the SVM had already en-

countered these tokens. Depending on how the data was split, the overlap already encountered in the training data varies from 0.19 to 0.26 for 30.000 tokens. This difference is not as large for 550.000 tokens, where it varies between 0.6 and 0.63.

Size	Total	Train	Test	Overlap
30k	504	456	65	0.26
30k	504	464	51	0.22
30k	504	454	62	0.19
80k	1096	1003	142	0.35
80k	1096	1007	147	0.39
80k	1096	996	169	0.41
120k	1561	1443	215	0.45
120k	1561	1439	227	0.46
120k	1561	1440	223	0.46
200k	2262	2085	362	0.51
200k	2262	2066	408	0.52
200k	2262	2092	365	0.53
300k	3074	2818	545	0.53
300k	3074	2824	550	0.55
300k	3074	2822	557	0.55
550k	4705	4369	854	0.61
550k	4705	4390	857	0.63
550k	4705	4331	927	0.60

Table 6: Distribution of NEs in the training data. The overlap refers to the percentage of types that was present in both the training set and the test set extracted from the training.

Table 6 also illustrates that the number of overlapping tokens increases immensely when the number of tokens in the training data increases. It ranges from .19 to .63, which means that the higher the number of tokens in the training set, the likelihood that NEs in the test set are also present in the training data increases. Therefore, the classifier does not need to classify as many unseen tokens and overall performance increases.

6 Conclusion and Future Work

We presented a named entity recognition system for Spanish-English code-switched Tweets based on a combination of classical machine learning algorithms and post-processing. The best performing classifier was a Support Vector Machine with an RBF kernel, allowing it to be flexible and less prone to overfitting compared to other classifiers on the held-out test data. We used a small set of features which were selected based on frequency observations in the training data. This provides a classifier with low computational costs and could allow for easy adaptation for other language pairs. Overall, the task of recognizing named entities in multilingual Twitter data proved to be quite challenging. We managed to achieve an overall F1 of 53.65 and thus modestly outperformed the baseline provided by Aguilar et al. (2018). The results show that there is a large amount of variation in classifier performance depending on the specific NEs present in the training and test sets. The classifiers could be improved by incorporating gazetteer resources more specific to Spanish-speaking countries, for example for geographical entities similar to that of the United States census list. Currently, the focus lies on English NEs as there are more resources available. Furthermore, the current approach relies heavily on gazetteering, and the wider context of a token could be taken into account by, for example, determining correlations of certain types of NEs with related verbs in the same Tweet.

References

Gustavo Aguilar, Fahad AlGhamdi, Victor Soto, Mona Diab, Julia Hirschberg, and Thamar Solorio. 2018. Overview of the CALCS 2018 Shared Task: Named Entity Recognition on Code-switched Data. In *Proceedings of the Third Workshop on Computational Approaches to Linguistic Code-Switching*, Melbourne, Australia. Association for Computational Linguistics.

Timothy Baldwin, Marie Catherine de Marneffe, Bo Han, Young-Bum Kim, Alan Ritter, and Wei Xu. 2015. Shared tasks of the 2015 workshop on noisy user-generated text: Twitter lexical normalization and named entity recognition. ACL Association for Computational Linguistics.

Leo Breiman. 2001. Random forests. *Machine Learning*, 45(1):5–32.

Chih-Chung Chang and Chih-Jen Lin. 2011. Libsvm: a library for support vector machines. *ACM transactions on intelligent systems and technology (TIST)*, 2(3):27.

Daniel Claeser, Dennis Felske, and Samantha Kent. 2018. Token level code-switching detection using wikipedia as a lexical resource. In *Language Technologies for the Challenges of the Digital Age*, pages 192–198, Cham. Springer International Publishing.

Leon Derczynski, Diana Maynard, Giuseppe Rizzo, Marieke van Erp, Genevieve Gorrell, Raphal Troncy, Johann Petrak, and Kalina Bontcheva. 2015. Analysis of named entity recognition and linking for

tweets. *Information Processing & Management*, 51(2):32 – 49.

Yoav Freund and Robert E. Schapire. 1995. A desicion-theoretic generalization of on-line learning and an application to boosting. In *Computational Learning Theory*, pages 23–37, Berlin, Heidelberg. Springer Berlin Heidelberg.

Hege Fromreide, Dirk Hovy, and Anders Søgaard. 2014. Crowdsourcing and annotating ner for twitter #drift. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC-2014)*. European Language Resources Association (ELRA).

Nut Limsopatham and Nigel Collier. 2016. Bidirectional lstm for named entity recognition in twitter messages. In *Proceedings of the 2nd Workshop on Noisy User-generated Text (WNUT)*, pages 145–152. The COLING 2016 Organizing Committee.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. 2011. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

Benjamin Strauss, Bethany Toma, Alan Ritter, Marie-Catherine de Marneffe, and Wei Xu. 2016. Results of the wnut16 named entity recognition shared task. In *Proceedings of the 2nd Workshop on Noisy User-generated Text (WNUT)*, pages 138–144. The COLING 2016 Organizing Committee.

Zhiqiang Toh, Bin Chen, and Jian Su. 2015. Improving twitter named entity recognition using word representations. In *Proceedings of the Workshop on Noisy User-generated Text*, pages 141–145. Association for Computational Linguistics.

Ikuya Yamada, Hideaki Takeda, and Yoshiyasu Takefuji. 2015. Enhancing named entity recognition in twitter messages using entity linking. In *Proceedings of the Workshop on Noisy User-generated Text*, pages 136–140. Association for Computational Linguistics.