IRISA at SMM4H 2018: Neural Network and Bagging for Tweet Classification

Anne-Lyse Minard, Christian Raymond, Vincent Claveau

IRISA at SMM4H 2018: Neural Network and Bagging for Tweet Classification

Anne-Lyse Minard[1] **Christian Raymond**[1,2] **Vincent Claveau**[1]
(1) CNRS, IRISA, Univ Rennes
(2) INSA Rennes, Rennes
Campus de Beaulieu, 35042 Rennes, France
`firstname.lastname@irisa.fr`

Abstract

This paper describes the systems developed by IRISA to participate to the four tasks of the SMM4H 2018 challenge. For these tweet classification tasks, we adopt a common approach based on recurrent neural networks (BiLSTM). Our main contributions are the use of certain features, the use of Bagging in order to deal with unbalanced datasets, and on the automatic selection of difficult examples. These techniques allow us to reach 91.4, 46.5, 47.8, 85.0 as F1-scores for Tasks 1 to 4.

1 Introduction

IRISA has participated in the four tasks of the SMM4H challenge (Weissenbacher et al., 2018). Yet, we have focused on Task 2 and 3, which are the most challenging ones, in particular because they have unbalanced data. Moreover, for Task 2, the three classes have very fuzzy boundaries, which makes some tweets difficult to classify even for humans. Our main contribution is to rely on Bagging (Bootstrap Aggregating) in order to deal with this problem of unbalanced data.

2 Methods

2.1 RNN: BiLSTM

For the four tasks, we have developed classifiers based on recurrent neural networks which consists in one Bidirectional LSTM layer (Graves et al., 2013) and a dense layer with a softmax activation as hidden layer. The input layer takes a representation of a tweet which consists in the word embeddings of each token and, depending of the task, a one-hot vector for each token or a one-hot vector for some medical terms in the tweet. Metamap Lite (Demner-Fushman et al., 2017) is used to extract specific medical terms from the tweets. We restrict the number of semantic types according to the task: for Task 1, we have selected only terms related to drugs or substances; for Task 2, only to procedural terms; and for Task 3, we have selected both terms related to drugs and terms related to symptoms. For Task 1 and Task 2, we observe an improvement while using medical terms, whereas for Task 4 the use of metamap has no influence on the results. We use the word embeddings distributed by Grave et al. (2018). They have been trained with FastText (Bojanowski et al., 2017).

2.2 Bonzaiboost

During the development phase, we have used BONZAIBOOST, an implementation of the boosting algorithm adaboost.MH (Laurent et al., 2014) on decision trees. The results obtained are a bit lower than those of recurrent neural network methods. Yet, the experiments done with BONZAIBOOST allowed us to extract the most discriminating words, to choose the better features for the RNN, and to select the difficult examples (see Section 2.4). For Task 1, the important words found are drug names, such as *xanax*. For Task 2, the useful words are verbs indicating the action of taking a drug, the results of its intake, or the fact that a drug is needed (e.g. *took*, *need*). For Task 3, the discriminating words include symptom names (e.g. *dizzy*, *headache*). Finally for Task 4, no relevant discriminating words have been found. These findings help us to determine the semantic types of the medical terms to be used in the feature set.

2.3 Bagging

Bagging (Breiman, 1996) is a technique that consists in combining the prediction of different learners, where each "learner" uses only a sample of the original training set. We learn several models, with, for each, a subset of the training dataset, different training parameters (number of epochs, number of hidden layers...) and different feature sets. To deal with unbalanced datasets in Tasks

		Model			Input		Data	
		BiLSTM	Bagging	epochs	metamap	tokens	cleaned	F1
T1	R1	x		3	x			**90.2**
	R2	x		3	x		x	90.0
	R3	x		3		x		88.1
T2	R1	x		3	x			67.9
	R2		x	3	x	x		67.7
	R3		x	1 to 5	x	x		**68.1**
T3	R1	x		3		x	x	50.1
	R2		x	3		x		**52.0**
	R3		x	1 to 7	x	x		46.6
T4	R1	x		3				87.7
	R2	x		3			x	**92.2**
	R3	x		3	x			87.2

Table 1: Description of the submitted runs and results obtained on the training dataset.

2 and 3, for Task 2 an equal number of instances of each class are chosen (2000 examples) and, for Task 3, every positive example is selected, and 20% of the negative ones are randomly selected.

Bagging seems to improve the results, especially because it allows the RNN to deal with more balanced datasets. For Task 1 and Task 4, bagging does not improve the results; this may be due to the results already being high (F1 $>$ 0.90), and for Task 1, to the data being already balanced.

2.4 Automatic cleaning of the datasets

Every manually annotated dataset may contain annotation errors or uncertain annotation due to the difficulty of the task. In order to improve the classification performance of our system, we have tried to clean up the training data by removing potential errors. We have considered that the tweets to be removed are those incorrectly classified although it was part of the training data used to train the model. More precisely, we proceed as follows: a model is trained on the complete training dataset; this model is then applied to predict the class of every example of this training dataset; the misclassified tweets are finally removed from the data; the cleaned dataset is then used to train the final model. We have removed 234, 183 and 250 examples respectively for Tasks 1, 3 and 4. For the Task 2, we have not observed improvement while removing difficult examples.

3 Evaluation

For each experiment the data is split into train set (80%) and dev set (20%). Evaluation is performed with a 5-fold cross validation, except when using bagging techniques. For the experiments with bagging, we train 20 models (with more models we do not get any improvement). The description of all the submitted runs and the obtained results on the training data are given in Table 1.

The official results are given in Table 2. The use of bagging techniques enables us to improve from 1.9 to 3.9 points the performance of our systems for Task 2 and Task 3. The automatic cleaning of the datasets is also a reason for a light improvement for Task 1 and Task 4.

Task	Run 1	Run 2	Run 3
T1	91.1	**91.4**	90.6
T2	43.6	45.5	**46.5**
T3	43.9	46.2	**47.8**
T4	84.4	**85.0**	82.4

Table 2: Final results in terms of F1-score.

References

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2017. Enriching Word Vectors with Subword Information. *Transactions of the Association for Computational Linguistics*, 5:135–146.

Leo Breiman. 1996. Bagging predictors. *Mach. Learn.*, 24(2):123–140.

Dina Demner-Fushman, Willie J Rogers, and Alan R Aronson. 2017. Metamap lite: an evaluation of a new java implementation of metamap. *Journal of the American Medical Informatics Association*, 24(4):841–844.

Edouard Grave, Piotr Bojanowski, Prakhar Gupta, Armand Joulin, and Tomas Mikolov. 2018. Learning Word Vectors for 157 Languages. In *Proceedings of the International Conference on Language Resources and Evaluation (LREC 2018)*.

Alex Graves, Abdel-rahman Mohamed, and Geoffrey Hinton. 2013. Speech recognition with deep recurrent neural networks. In *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 6645–6649. IEEE.

Antoine Laurent, Nathalie Camelin, and Christian Raymond. 2014. Boosting bonsai trees for efficient features combination : application to speaker role identification. In *InterSpeech*, Singapour.

Davy Weissenbacher, Abeed Sarker, Michael Paul, and Graciela Gonzalez-Hernandez. 2018. Overview of the Third Social Media Mining for Health (SMM4H) Shared Tasks at EMNLP 2018. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing (EMNLP), 2018*.