Using machine learning and deep learning methods to find mentions of adverse drug reactions in social media

Pilar López-Úbeda, Manuel Carlos Díaz-Galiano,
Maria-Teresa Martín-Valdivia, L. Alfonso Ureña-López

Department of Computer Science, Advanced Studies Center in ICT (CEATIC)
Universidad de Jaén, Campus Las Lagunillas, 23071, Jaén, Spain
{plubeda, mcdiaz, maite, laurena}@ujaen.es

Abstract

Today social networks play an important role, where people can share information related to health. This information can be used for public health monitoring tasks through the use of Natural Language Processing (NLP) techniques. Social Media Mining for Health Applications (SMM4H) provides tasks such as those described in this document to help manage information in the health domain.

This document shows the first participation of the SINAI group in SMM4H. We study approaches based on machine learning and deep learning to extract adverse drug reaction mentions from highly informal texts in Twitter.

The results obtained in the tasks are encouraging, we are close to the average of all participants and even above in some cases.

1 Introduction

An Adverse Drug Reaction (ADR) is an injury occurring after a drug (medication) is used at the recommended dosage, for recommended symptoms. This is a area that has already been researched in recent years (Sarker and Gonzalez, 2015; Karimi et al., 2015), and in which we will contribute with new systems.

The proposed shared tasks of SMM4H continue with NLP challenges in social media mining for health monitoring and surveillance (ws-, 2018; Weissenbacher et al., 2018).

We have decided to participate in 2 of the 4 tasks proposed by the organizers: automatic classifications of adverse effects mentions in tweets and extraction of adverse effect mentions.

In task *automatic classifications of adverse effects* the goal is a binary classification problem. The designed system for this sub-task should be able to distinguish tweets reporting an Adverse Effect (AE) from those that do not.

In the second task called *Extraction of Adverse Effect mentions*. This task includes identifying the text span of the reported ADRs and distinguishing ADRs from similar non-ADR expression. ADRs are multi-token, descriptive, expressions, so this subtask requires advanced Named Entity Recognition (NER) approaches.

2 Tweet data

The corpus are composed of tweets extracted from the famous social network called Twitter. This social network allows people to freely post short messages (called tweets) of up to 140 characters. Twitter has rapidly gained popularity worldwide, with more than 326 million active users generating more than 500 million tweets daily.

- Data set for task 1: For each tweet, the publicly available data set contains: (i) the user ID, (ii) the tweet ID, and (iii) the binary annotation indicating the presence or absence of ADRs.

  The training data is composed of 25,672 tweets (2,374 positive and 23,298 negative) and the test data contains 4,5175 tweets.

- Data set for task 2: This set contains a subset of the tweets from Task 1 tagged as *hasADR* plus an equal number of *noADR* tweets. The corpus contains: (i) the tweet ID, (ii) the start and (iii) end of the span, (iv) the annotation indicating an ADR or not and (v) the text covered by the span in the tweet.

  The training data is composed of 2,367 tweets (1,212 positive and 1,155 negative) and the test data contains 1,573 tweets.

3 Taking part in tasks

In this section we will explain the 3 methodologies applied to each task.

Proceedings of the 4th Social Media Mining for Health Applications (#SMM4H) Workshop & Shared Task, pages 102–106
Florence, Italy, August 2, 2019. ©2019 Association for Computational Linguistics

Before beginning to implement our approaches, it is necessary to clean the text of some rare characters that we find, these characters can make noise to our systems, therefore, we must treat them correctly. This pre-processing has been:

- Convert the text to lowercase.

- Substitution of characters HTML like: *&amp;*, *&lt;*, and *&gt;* to your representation: &, < and >.

3.1 Task 1: Automatic classifications of adverse effects mentions in tweets

In addition to the text processing already carried out and described above, for this task we have also decided to carry out another pre-processing:

- Expand contractions: the contractions in the text have been expanded as for example: *you're* to *you are*
- Remove hashtag: for this task we consider that the hashtag add noise to the text as we do not process them.
- Remove @ mentions: mentions of persons have been removed from the text.
- Remove non-alphanumeric words: we have only taken into account alphanumeric words.

For Task 1 systems we have used the automatic learning and deep learning approaches described below:

3.1.1 SVM

SVM (Vector Support Machines) is one of the best classifiers for a wide range of situations, so it is considered one of the references within the field of statistical learning and machine learning. We used SVM with linear kernel.

For tweet processing we have applied the TF-IDF schema with the following parameters: min_df = 3, max_df = 0.8, sublinear_tf = True, use_idf = True, lowercase = True and ngram_range = (1,3).

This will be our baseline, from which we will depart for better results.

3.1.2 SVM + features

For this system, we have used the SVM of the previous baseline adding some relevant features for this specific task. We believe it is interesting to use external resources referring to the medical domain.

We have used the medical entity recognizer for English called MetaMap (Aronson, 2001). MetaMap is a widely available program providing access to the concepts in the unified medical language system (UMLS[1]) Metathesaurus from biomedical text. In addition, this resource provides additional information about the medical concept detected. For example, we can know the Concept Unique Identifier (CUI), the preferred name or the semantic type for the concept.

We make use of the semantic type of the concepts detected, and specifically, we use the semantic groups: ”dsyn”, ”fndg”, ”inpo”, ”menp”, ”mobd”, ”neop”, ”patf”, ”phsf”, ”sosy”, ”topp” creating a vector of 10 positions, we insert 1 in the case in which it finds a concept in the tweet with that semantic group, 0 in other cases.

These semantic groups can be understood as: Disease or Syndrome, Finding, Injury or Poisoning, Mental Process, Mental or Behavioral Dysfunction, Neoplastic Process, Pathologic Function, Physiologic Function, Sign or Symptom and Therapeutic or Preventive Procedure respectively.

We decided to use these semantic types thanks to the ADR mentioned in Task 2 corpus, these ADR were introduced in MetaMap and we chose to use the 10 most repeated semantic groups.

3.1.3 CNN

For the third system, we implemented a Convolutional Neural Network (CNN). CNN are a category of neural networks that have proven very effective in areas such as image recognition and classification.

The architecture of the network is as follows:
- Embedding layer.
- 1D convolution layer: filters = 32, convolution window = 3, activation = relu and the other default values.
- 1D Max pooling layer: size of the max pooling windows = 2 and the other default values.
- 1D convolution layer: filters = 32, convolution window = 3, activation = relu and the other default values.
- Global max pooling layer with default values.
- Dense layer for output with 1 output unit and activation = sigmoid.

We have used the Twitter pre-trained word vec-

[1] https://www.nlm.nih.gov/research/umls/

tors of GloVe[2]. These embeddings are composed of 2B tweets, 27B tokens, 1.2M vocab and 200 dimension.

3.2 Task 2: Extraction of Adverse Effect mentions

In the second task, our team has focused on the use of Conditional Random Field (CRF) algorithm, applying characteristics to it in such a way that they provide extra information to each word of the document.

3.2.1 CRF

CRF classifier is a stochastic model commonly used to label and segment data sequences or extract information from documents. We used CRFsuite, the implementation provided by Okazaki, as it is fast and provides a simple interface for training/modifying the input features.

The CRF classifier is trained on annotated mentions of ADRs and indications, and it attempts to classify individual tokens in sentences. Therefore, it learns to distinguish five different labels: ADR and 0.

Below, we define some characteristics for each word in the document used in all our models:

- Characteristics of the context: Context is defined by three characteristics that include the current word (word), the previous word (word-1) and the subsequent word in the sentence (word+1).

- POS: Part of speech of the token, which was generated using the Spacy[3] library for Python.

- Lemma: Lemma of the token, which was generated using the Spacy.

- Other features: we incorporate some basic features of each word such as isLower, isUpper, isTitle, isDigit, isAlpha, isBeginOfSentence and isEndIfSentece.

3.2.2 CRF + W2V

We want to use embedded word vectors as feature in existing conditional random field (CRF) with gazetteer features for sequence labeling task in text.

We have again used the Twitter pre-trained word vectors of GloVe but with 50 dimension.

To make this possible, we added 50 new features to each word, to the previous word and to the next word. These 50 characteristics refer to each of the dimensions of that word. In this way the algorithm will learn where the words are within the axes in order to improve in context.

Bitchain	Word	Count
011110100000	unmotivated	754
011110100000	knackered	2407
011110100000	tired	232683
011110100000	exhausted	19368
011110100000	drained	3333

Table 1: Example content of Brown cluster.

3.2.3 CRF + BC + W2V

For the last system developed for this task, the word representations feature induced by Brown clustering method was introduced as an additional feature.

Brown clustering (Brown et al., 1992) is a greedy, hierarchical, agglomerative hard clustering algorithm to partition a vocabulary into a set of clusters with minimal loss in mutual information. Intuitively, the Brown clustering method will merge the tokens with similar contexts into the same cluster.

The implementation of Brown clustering method by Liang and described by Owoputi et al. is adopted in our system. The clustering used contains 216,846 words, is grouped in 1000 clusters and processed more than 56 million tweets.

Some examples of Brown clustering are shown in Table 1. In this table we can see how different words are in the same cluster (011110100000) and the number of occurrences found.

The feature that was finally added to the method was the bitchain to which each word belonged.

4 Results

In this section we show the results obtained by the group SINAI in the participation of SMM4H Shared Task 2019.

4.1 Task 1

The average of all participants in Task 1 and the results obtained by our group in Task 1 are those shown in Table 2.

As we can see the mean has a low measure, so we can intuit that it is a difficult task. In our case, the neural network learns better than machine learning systems, although we add features

[2] https://nlp.stanford.edu/projects/glove/

[3] https://spacy.io/

Approach	**F1**	**Prec**	**Recall**
Average particp.	**0.5019**	0.5351	**0.5054**
SVM	0.4509	**0.6393**	0.3482
SVM + features	0.4829	0.6222	0.3946
CNN	0.4969	0.5517	0.4521

Table 2: Result obtained for Task 1.

Approach	**F1**	**Prec**	**Recall**
Average particip.	0.5383	0.5129	**0.6174**
CRF	0.496	0.633	0.408
CRF + W2V	0.532	**0.616**	0.468
CRF + BC + W2V	**0.542**	0.612	0.486

Table 3: Result obtained for Task 2 relaxed matching.

to these models.

Although the use of features added to SVM improves our baseline in F1 and recall, they are not sufficient and we do not get a substantial increase. We can observe that systems 2 and 3 worsen the precision. For future work we can try to choose some features more related to the task.

4.2 Task 2

In this task two measures of agreement were computed: strict and relaxed matching.

- Relaxed matching

  The average scores for this task with relaxed matching and our results are showing in Table 3.

  In different measures such as F1 and precision we are above average. In terms of precision, we exceeded it by 20%, although the average recall does not reach it and that hurts us.

- Strict matching

  Our results and the average scores for all participants in this task with strict matching are presented in Table 4.

  In this system, we can see that the same thing happens as in the case of relaxed matching, we surpass the F1 and precision measures, but not in recall. For next participation we will pay special interest in the exhaustiveness for relevant instances that we have recovered.

We will be able to analyze the results once the organizers provide us with the complete test. With this, we will be able to carry out an analysis of errors and see the failures obtained and how to improve them.

Approach	**F1**	**Prec**	**Recall**
Average particip.	0.3169	0.3026	**0.3581**
CRF	0.326	0.419	0.267
CRF + W2V	0.352	0.408	0.31
CRF + BC + W2V	**0.36**	**0.408**	0.322

Table 4: Result obtained for Task 2 strict matching.

5 Conclusions

In this document, we expose the first participation of the SINAI group in SMM4H, we created 3 strategies for Task 1 and 3 strategies for Task 2. For Task 1 different approaches of machine learning and deep learning were implemented, whereas for Task 2 the effectiveness of several classification characteristics was explored in the training of the CRF model and it was found that context and cluster integration were the most contributing characteristics.

In both tasks we managed to overcome our baseline and improve in each method. In Task 1 we get a F1 of 0.486 being a little below the average of all participants, in Task 2 we managed to obtain a measure F1 of 0.322 in the strict system and 0.486 in relaxed system.

Our future work will involve exploring the effectiveness of training a deep learning neural network, rather than the CRF, to learn features and classify labels and improve our neural networks and add new text features. As well as participate in all tasks proposed to implement our systems and expose them to the scientific community.

Acknowledgments

This work has been partially supported by Fondo Europeo de Desarrollo Regional (FEDER) and REDES project (TIN2015-65136-C2-1-R) from the Spanish Government.

References

2018. *Proceedings of the 2018 EMNLP Workshop SMM4H: The 3rd Social Media Mining for Health Applications Workshop & Shared Task*. Association for Computational Linguistics, Brussels, Belgium.

Alan R Aronson. 2001. Effective mapping of biomedical text to the umls metathesaurus: the metamap program. In *Proceedings of the AMIA Symposium*, page 17. American Medical Informatics Association.

Peter F Brown, Peter V Desouza, Robert L Mercer, Vincent J Della Pietra, and Jenifer C Lai. 1992. Class-based n-gram models of natural language. *Computational linguistics*, 18(4):467–479.

Sarvnaz Karimi, Chen Wang, Alejandro Metke-Jimenez, Raj Gaire, and Cecile Paris. 2015. Text and data mining techniques in adverse drug reaction detection. *ACM Computing Surveys (CSUR)*, 47(4):56.

Percy Liang. 2005. *Semi-supervised learning for natural language*. Ph.D. thesis, Massachusetts Institute of Technology.

Naoaki Okazaki. 2007. Crfsuite: a fast implementation of conditional random fields (crfs).

Olutobi Owoputi, Chris Dyer, Kevin Gimpel, and Nathan Schneider. 2012. Part-of-speech tagging for twitter: Word clusters and other advances.

Abeed Sarker and Graciela Gonzalez. 2015. Portable automatic text classification for adverse drug reaction detection via multi-corpus training. *Journal of biomedical informatics*, 53:196–207.

Davy Weissenbacher, Abeed Sarker, Michael J. 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 EMNLP Workshop SMM4H: The 3rd Social Media Mining for Health Applications Workshop and Shared Task*, pages 13–16, Brussels, Belgium. Association for Computational Linguistics.