Drug Mention Recognition in Twitter Posts Using a Deep Learning Approach

João F. Silva[1], Tiago Almeida[1], Rui Antunes[1], João R. Almeida[1,2], Sérgio Matos[1,*]

[1] Department of Electronics, Telecommunications and Informatics (DETI), Institute of Electronics and Informatics Engineering of Aveiro (IEETA), University of Aveiro, Portugal

[2] Department of Computation, University of A Coruña, Spain

* Team Leader and corresponding author. E-mail: aleixomatos@ua.pt

***Abstract*—In an era where medicine and technology are closely intertwined, sources of patient-generated data such as social media content are being explored to extract important information for the study of public health and patient trajectories. Drug related mentions present in Twitter posts are a particular use case, as the process of automatically extracting drug related mentions from tweets can provide novel relevant information for pharmacoepidemiologic studies. In this paper, we describe the system developed by the BIT.UA team from the University of Aveiro during the participation in BioCreative VII Track 3 on automatic extraction of medication names in tweets. The system consists of an end-to-end deep learning architecture based on transformers, and was used in all three submitted runs for the challenge. Run 1 obtained the best results on strict evaluation (F_1-score of 0.6810) whereas Run 3 performed better on overlapping evaluation (F_1-score of 0.7700).**

Keywords—NER, Twitter, transformer based model, deep learning, medication, patient-generated data

I. Introduction

The field of medicine has been the subject of much evolution during past years, benefiting from key progresses in other fields such as that of technology. With the increased growth in medical data, information availability and patient awareness, patient-generated data has become a valuable asset in the study of population health and patient trajectories by providing important unique information. Social media content such as Twitter posts is an example of patient-generated data that has already been explored for health research purposes, with existing applications leveraging Twitter data to study public health (1) or depression (2).

In recent years, international challenges have been created to foster research in this particular field, for instance Social Media Mining for Health Applications (#SMM4H) has organized several shared tasks focused on exploring tweets for different purposes. In the 2018 and 2020 editions (3,4), #SMM4H organized tracks on medication detection and extraction from tweets in an effort to improve the process of automatically extracting drug related information from tweets, as this data can be very important for pharmacoepidemiologic research. Even though both challenge tracks (3,4) provided valuable datasets and a venue for benchmarking solutions developed by researchers, the unrealistic equal distribution between relevant and irrelevant tweets limited the practicality of these resources.

Since the real scenario is closer to a “needle in a haystack” problem, as described in (5), where the number of tweets without entities of interest vastly outnumbers the amount of tweets effectively containing medication mentions, it is important to prepare and develop solutions capable of coping with such pronounced class imbalance. In fact, #SMM4H’20 track organizers developed a system which was evaluated on a class-balanced (50-50) and an imbalanced Twitter dataset, obtaining F_1-scores of 93.7% and 78.8% in these two corpora, respectively, clearly demonstrating the impact of an incorrect representation of the real scenario on the resulting system performance (6). Considering these concerns and following-up on the #SMM4H’20 shared task, 2021 BioCreative VII held a challenge track on automatic extraction of medication names in tweets (Track 3) where the provided dataset represented a more realistic scenario with high class imbalance. Although the dataset was also prepared for a Named Entity Normalization (NEN) task, the challenge was solely focused on the Named Entity Recognition (NER) component of extracting medication mentions.

In this paper, we describe the system developed by BIT.UA under the scope of BioCreative VII Track 3, which was used to submit three participating runs. The resulting end-to-end system explored the potential of transformer based architectures in natural language problems.

II. Methods

The system herein presented makes use of Deep Learning techniques to perform NER on Twitter data, retrieving detected medication and dietary supplement mentions along with the corresponding spans within the tweet. In this section, we provide more information on the used dataset, language model and model architecture, processing mechanisms and tests that were performed during the challenge.

J. F. S., T.A., R.A. and J. R. A. are funded by the FCT - Foundation for Science and Technology (Portuguese national funds) - under the grants PD/BD/142878/2018, 2020.05784.BD, SFRH/BD/137000/2018 and SFRH/BD/147837/2019, respectively.

A. *Data*

Task organizers provided a corpora containing tweets from 212 pregnant women, with the training set consisting of approximately 89,000 tweets (218 tweets mentioned at least one drug), the validation set containing almost 39,000 tweets (93 tweets mentioned at least one drug), and the test set holding nearly 54,000 tweets.

Dataset distribution was purposefully highly imbalanced (only approximately 0.2% of the tweets contained medication mentions) so as to capture the real scenario where *relevant* tweets are scarce among all existing tweets. The training and validations datasets were provided along with gold standard annotations, comprising medication entities and their corresponding textual span within the tweet.

Despite also having a normalized form for each annotated medication, since the focus of this challenge was not on NEN but on NER we did not explore this information in our solution. Additionally, a supplementary dataset was provided consisting of the training dataset from #SMM4H'18 shared tasks, which contained nearly 10,000 tweets. This dataset had a balanced distribution, with around 50% of the tweets having medication mentions. Since the objective of this challenge was precisely to develop practical solutions for the existing imbalanced real scenario, we opted to not explore this dataset in our solution.

B. *Model Architecture and Configurations*

The model used in this work (Fig. 1) has a simple architecture consisting of a language model, a Multilayer Perceptron (MLP) containing two Fully-Connected Layers (FCN), and a Conditional Random Field (CRF) layer. Regarding the language model, we opted for publicly available RoBERTa models which were already pretrained on Twitter. Despite initially intending to use the BERTweet model[1], this model had implementation problems regarding span retrieval during the tokenization procedure, which was a severe handicap for our solution. As an alternative, we used the base RoBERTa model for Twitter from Cardiff NLP[2]. In the MLP, the first FCN uses the Mish (7) activation function and has 128 hidden units, whereas both the second FCN and CRF layer have a size of N where N corresponds to the number of possible tags (in this case we used N=4).

Concerning model training, only the MLP and CRF were trained, and a weighted sample loss scheme was used to compensate for the imbalanced class distribution in the dataset, where negative samples (*i.e.* tweets where no "B" or "I" tag was detected) have their loss reduced by 60%, thus reducing their importance. Model performance was evaluated based on strict F_1-scores, where detected spans and entity text must exactly match the gold standard annotations.

C. *Pre and Post-processing Mechanisms*

Although it has been shown that the integration of heuristics mechanisms can play an important role in system performance for the present task (5), in this work we did not focus on developing a robust heuristics component but instead distributed our efforts on the system as a whole due to timing limitations. Nonetheless, some simple pre and post-processing heuristics were created and integrated in the system, as described next.

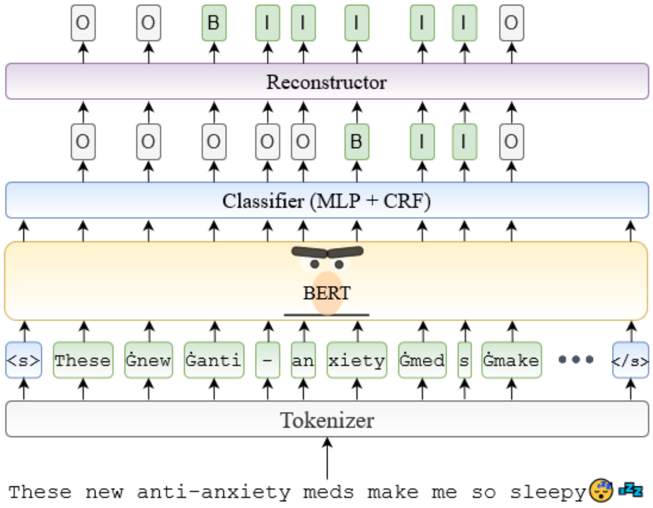

Fig. 1. Overview of the model architecture (BERT-MLP-CRF) demonstrating the functioning of the reconstructing sequence decoder.

The first processing step implemented in the system was a post-processing mechanism regarding the reconstruction of predicted entities. Since we used a modified BIO (Beginning, Inside, Outside) tagging schema, where a fourth tag named PAD was introduced to represent padding tokens, and a BERT (Bidirectional Encoder Representations from Transformers) derived language model which splits some words in several subtokens, there may exist situations where the model predicts only part of the "medication entity" as the actual entity (e.g. tagging "xiety meds" as entity instead of "anti-anxiety meds", as shown in Fig. 1), which results in a positive match when using approximate evaluation but in a miss if using a strict evaluation. In our solution all models were validated using a strict evaluation, thus we implemented a reconstructor that checks for incomplete entity spans and adds the missing subtokens so that only full tokens are considered. Additionally, the reconstructor also corrects some of the non-entity tokens (*e.g.* punctuation, emoji derived tokens) that are wrongfully tagged as entities by resetting their tag to O.

The implementation of the entity reconstructing heuristic faced some technical problems due to the presence of emojis in the tweets, which is frequent as emojis are widely used in social media communication such as Twitter posts. Due to the language model used in this work, the tokenization process resulted in emojis being split into numerous "dummy" subtokens with special characters, which had to be disregarded in the entity reconstructing heuristic. Three possible approaches were defined to address the problem of emojis: 1) convert emojis to their corresponding text variant, 2) replace emojis with a punctuation char, and 3) maintain emojis and compile a list of all possible "dummy" subtokens originated from emoji tokenization, using the resulting compiled list in the entity

[1] https://huggingface.co/vinai/bertweet-base

[2] https://huggingface.co/cardiffnlp/twitter-roberta-base

reconstructor. Even though emojis can be easily converted to text using the emoji Python package, this package does not cover all existing emojis in the corpus. Furthermore, converting emojis to text affects final sentence span, which must be factored in when computing the entity spans for the predicted annotations. Owing to both reasons, method 1) for emoji handling was side-lined. The second method is more straightforward as the sentence maintains its span intact, and was tested using the "." and "_" punctuation characters. However, models trained using this pre-processing approach had worse performance comparatively to using sentences with the emojis, showing that emojis might actually provide relevant information to the model (e.g. emojis can be representative of human sentimental state, being an important feature for sentiment analysis tasks). Due to the previously mentioned problems of approaches 1) and 2), we selected the third approach for the emoji pre-processing mechanism as it does not affect sentence spans nor remove emoji information, whilst allowing the use of the token reconstruction procedure.

Finally, in an attempt to filter out some "irrelevant" tweets (e.g. tweets containing only emojis) and feed the deep learning model with cleaner input data, an additional pre-processing step was introduced consisting of two simple rules: 1) remove tweets with less than four subtokens and 2) remove tweets where less than 40% of the characters are alphanumeric. The use of this simple heuristic mechanism resulted in the removal of 2,084 tweets out of nearly 128,000 tweets (training and validation datasets combined). Since the dataset already contains a scarce amount of true positives, it is important to ensure that none are removed during this procedure. After checking the list of removed tweets we verified that no true positive was incorrectly eliminated with this mechanism.

D. Submitted Runs

The described system was empirically tested with several modifications, resulting in the three final submitted runs which will be further detailed next. The models were implemented using TensorFlow and trained using the log-likelihood loss function and the AdamW optimizer. All models were executed on a machine with 20 CPU cores, 126GB of memory and an Nvidia Tesla K80 GPU. Model performance was evaluated using strict F1-score.

- Run 1

For the first run, which we defined as our baseline, the model was trained on the training dataset, validated on the validation dataset, and used the normal sequence decoder to evaluate model performance. The model checkpoint that attained the highest F_1-score in the validation dataset was selected to be used for inference in test time, resulting in the submitted prediction for run 1.

- Run 2

The second run was similar to the baseline, differing in the sequence decoder selected to be used during evaluation. Here, the reconstructing sequence decoder was used to evaluate model performance in the validation dataset. The model checkpoint that attained the highest "reconstructed" F_1-score in the validation dataset was selected for inference in the test dataset, resulting in the submitted prediction for run 2.

- Run 3

Since the use of the reconstructing sequence decoder led to performance improvements during the training phase, for the last run we decided to maintain the reconstructing sequence decoder but train the model on a combined dataset containing both the training and validation splits of the corpus. Here, the model saved in the last checkpoint (end of model training) was used for inference in the test dataset.

III. Results and Discussion

Herein we report some of the results obtained while performing experiments with the model, as well as the official test results from our challenge submissions.

Table I presents model performances during development time, and provides a comparison between using the normal sequence decoder and the reconstructing sequence decoder to evaluate model performance. In runs 1 and 2, the models were trained on the training dataset and evaluated on the validation dataset. Therefore, the reported values correspond to the evaluation metrics obtained on the validation dataset. As observable in the results from runs 1 and 2, using the additional post-processing mechanism to reconstruct entity predictions resulted in a performance improvement in every configuration during model training.

Since during our tests the reconstructing sequence decoder seemed to constantly improve model performance, a first system setting (test run 1) with normal sequence decoder was selected to be used as a baseline. Then, to directly assess the impact of this post-processing mechanism, the second system configuration (test run 2) used the reconstructing sequence decoder. Finally, we were interested in evaluating the impact of training the model using more data. As no external data was used in the present work, in the third scenario (test run 3) the model was trained in the training and validation datasets, at the cost of having no data left to evaluate model performance and select the optimal model checkpoint. Hence, in run 3 we used the last model which is saved after the training procedure ends.

In Table II it is possible to observe 1) official results from the three submitted test runs, and 2) some official benchmarking metrics provided by track organizers, which they computed using only the best submission from each participating team (16 teams participated in the challenge). Highlighted in bold are the best results (only for the 3 submitted runs) for each metric per type of evaluation (strict and overlapping). Surprisingly, run 1 obtained the best performance concerning strict evaluation by a margin of 4 percentage points to the second best (run 3), and run 2 obtained the worst performance in the test dataset, showing a completely opposed behavior from that observed during development time (Table I). This best submission (run 1) had below average performance in the challenge as it is below both the mean and median challenge performances.

On the other hand, when analyzing results regarding overlapping evaluation, run 2 was closer but still worse than run 1, and run 3 improved significantly with a higher recall than run 1. Comparing obtained results with the overall challenge metrics, run 3 had a system performance higher than the median

TABLE I. RESULTS OBTAINED IN DEVELOPMENT TIME WITH THE NORMAL SEQUENCE DECODER (LEFT) AND WITH THE RECONSTRUCTING SEQUENCE DECODER (RIGHT). NO RESULTS ARE REPORTED FOR RUN 3 AS IT WAS TRAINED BOTH IN THE TRAINING AND VALIDATION DATASETS.

Runs	**Strict Evaluation – No Reconstruction**			**Strict Evaluation - Reconstruction**		
	Precision	***Recall***	***F_1-score***	***Precision***	***Recall***	***F_1-score***
Run 1	0.7684	0.6952	0.7300	0.7766	0.6952	0.7337
Run 2	0.7604	0.6952	0.7264	0.7708	0.7048	0.7363
Run 3	—	—	—	—	—	—

TABLE II. RESULTS OBTAINED IN TEST TIME AND AGGREGATED CHALLENGE STATISTICS COMPUTED BY THE ORGANIZERS ON THE BEST SUBMISSIONS FOR ALL PARTICIPANTS.

Runs	**Strict Evaluation**			**Overlapping Evaluation**		
	Precision	***Recall***	***F_1-score***	***Precision***	***Recall***	***F_1-score***
Run 1	**0.7380**	**0.6330**	**0.6810**	**0.8100**	0.6940	0.7470
Run 2	0.6670	0.5990	0.6310	0.7670	0.6940	0.7290
Run 3	0.6720	0.6120	0.6410	0.8010	**0.7410**	**0.7700**
Mean - Challenge	0.7544	0.6583	0.6960	0.8105	0.7088	0.7491
Std - Challenge	—	—	0.0720	—	—	0.0596
Median - Challenge	—	—	0.6970	—	—	0.7585

and mean values for the challenge. Despite attaining a basic insight on how the different system configurations are performing, more experiments are required to correctly evaluate the impact of using pre and post-processing mechanisms such as the reconstructing sequence decoder. After a quick error analysis it was found that the reconstruction mechanism is not working optimally, which negatively impacted on test performances.

IV. CONCLUSION

In this work we performed medication identification in a highly imbalanced Twitter corpus. The best performing run under strict evaluation obtained an F_1-score of 0.6810, whereas the top run in overlapping evaluation attained 0.7700 F_1-score. Overall challenge statistics demonstrate that there is much margin for improvement, but also show positive signs considering our best overlapping evaluation score.

Regarding future work, there are several aspects that can be improved. Firstly, due to the high class imbalance, it would be advantageous to have an efficient triage system (e.g. heuristics-based) capable of reducing the number of irrelevant tweets being forwarded through the model. Next, a more elaborated pre-processing stage could be used to assess the impact of using a more "digested" input text. Finally, the reconstructing sequence decoder should be revised to enable better support for emoji handling without harming the correct reconstruction of actual medication entities, as a brief error analysis posterior to the challenge revealed that this negatively impacted on model performance.

REFERENCES

1. Paul, M., & Dredze, M. (2021) You Are What You Tweet: Analyzing Twitter for Public Health. *Proceedings of the International AAAI Conference on Web and Social Media*, 5(1), 265-272.
2. Chen, X., Sykora, M.D., Jackson, T.W., Elayan, S. (2018) What about Mood Swings: Identifying Depression on Twitter with Temporal Measures of Emotions. *Companion Proceedings of the The Web Conference 2018 (WWW '18).* International World Wide Web Conferences Steering Committee, Republic and Canton of Geneva, CHE, 1653–1660. DOI:https://doi.org/10.1145/3184558.3191624
3. Weissenbacher, D., Sarker, A., Paul, M.J., Gonzalez-Hernandez, G. (2018) Overview of the third social media mining for health (SMM4H) shared tasks at EMNLP 2018. *Proceedings of the 2018 EMNLP Workshop SMM4H: The 3rd Social Media Mining for Health Applications Workshop & Shared Task*. Association for Computational Linguistics
4. Klein, A.Z., Alimova, I., Flores, I., Magge, A., Miftahutdinov, Z., Minard, A.L., O'Connor, K., Sarker, A., Tutubalina, E., Weissenbacher, D., Gonzalez-Hernandez, G. (2020) Overview of the fifth social media mining for health applications (#smm4h) workshop & shared task at coling 2020. *Proceedings of the Fifth Social Media Mining for Health Applications (#SMM4H) Workshop & Shared Task.* Association for Computational Linguistics
5. Weissenbacher D., Rawal S., Magge A., Gonzalez-Hernandez G. (2021) Addressing Extreme Imbalance for Detecting Medications Mentioned in Twitter User Timelines. In: Tucker A., Henriques Abreu P., Cardoso J., Pereira Rodrigues P., Riaño D. (eds) *Artificial Intelligence in Medicine. AIME 2021. Lecture Notes in Computer Science*, vol 12721. Springer, Cham. https://doi.org/10.1007/978-3-030-77211-6_10
6. Weissenbacher, D., Sarker, A., Klein, A., O'Connor, K., Magge, A., Gonzalez-Hernandez, G. (2019) Deep neural networks ensemble for detecting medication mentions in tweets, *Journal of the American Medical Informatics Association*, Volume 26, Issue 12, December 2019, Pages 1618–1626
7. Misra, D. (2020) Mish: A Self Regularized Non-Monotonic Activation Function, *31st British Machine Vision Conference 2020, BMVC 2020*, Virtual Event, UK, September 7-10, 2020. BMVA Press