Ensemble of Convolutional Neural Networks for Medicine Intake Recognition in Twitter

Kai Hakala[1,2*], Farrokh Mehryary[1,2*], Hans Moen[1,3], Suwisa Kaewphan[1,2,4], Tapio Salakoski[1,4], Filip Ginter[1]
[1]Department of Future Technologies, University of Turku, Turku, Finland;
[2]University of Turku Graduate School, University of Turku, Turku, Finland;
[3]Department of Nursing Science, University of Turku, Turku, Finland;
[4]Turku Centre for Computer Science, Turku, Finland

Abstract

We present the results from our participation in the 2nd Social Media Mining for Health Applications Shared Task – Task 2. The goal of this task is to develop systems capable of recognizing mentions of medication intake in Twitter. Our best performing classification system is an ensemble of neural networks with features generated by word- and character-level convolutional neural network channels and a condensed weighted bag-of-words representation. A relatively strong performance is achieved, with an F-score of 66.3 according to the official evaluation, resulting in the 5th place in the shared task with performance close to the best systems created by other participating teams.

Introduction

Pharmacovigilance is the science of detecting, assessing and preventing drug-related adverse effects. A central focus and challenge is to detect adverse drug reactions (ADRs), which are undesired and harmful effects resulting from taking medications. Traditionally, ADRs are identified and recorded by health care professionals, and a part of their work includes weighting the risks and benefits of using medications. However, the number of documented ADRs is limited and it is believed that some of the more rare ADRs have not been revealed yet. As an alternative approach, pharmacovigilance has turned to social media. Social media represents a valuable forum for drug safety surveillance where text-mining techniques can be applied to extract potentially ADR-related events from a large population.

Our team participated in Task 2 of the 2nd Social Media Mining for Health Applications Shared Task at AMIA 2017. The goal of this shared task was to develop systems capable of classifying the mentions of medication intakes in tweets. Each of the provided tweets were to be assigned with one of the following three classes: *personal medication intake*, *possible medication intake* or *non-intake*. This is an important preliminary task for extracting ADRs from social media, since it can filter out the majority of tweets that mention drugs without any indications of personal intake. We participated with classifiers based on support vector machines (SVMs) and neural networks (NNs), as described in more detail in the Method section.

Data

The organizers provided a training dataset with manually assigned labels (*intake*, *possible intake*, *non-intake*) for each tweet. The *intake* class is defined as clearly expressing a personal intake of medication, whereas the *possible intake* is more ambiguous, yet still suggesting an intake by the tweet writer. The *non-intake* class includes the rest of the tweets, all of which include a mention of a drug, but refer to an intake by another person or discuss the drugs in general. Approximately 50% of the data belongs to the *non-intake* class, whereas the *intake* and *possible intake* classes constitute 19% and 31% of the data, respectively.

Due to the data sharing restrictions of Twitter, the organizers only provided the IDs of the tweets instead of their actual content. Since we started investigating this task much later than the data was released, we were only able to obtain the contents for 7444 tweets out of the total 8000 annotated tweets as some of the content had been already removed by the Twitter users, i.e. we had 7% less training data than teams who were involved in the shared task since the very

*: these authors have contributed equally.

beginning. The organizers also provided a separate development dataset, which consists of 2260 annotated tweets, of which we also lost roughly the same proportion. In the Results and Discussion section we evaluate the impact of the lost data in more detail.

Method

For our baseline approach we form term frequency–inverse document frequency (TF–IDF) weighted sparse bag-of-words (BOW) representations for all given tweets[1]. These representations are not only constructed for single tokens, but also token bigrams, trigrams and character n-grams of length 1 to 4. These representations are then fed as features to a linear SVM classifier[2]. The regularization parameter is selected to optimize the micro-averaged F-score of *intake* and *possible intake* classes, the official evaluation metric, on the development set. For the final submission in the shared task we merge the training and development sets and train the system on the combined dataset.

As the sparse representations are not able to generalize well to unseen vocabulary, we also test various NN approaches on this task. The final system is based on an ensemble of convolutional neural networks (CNNs)[3] and utilizes word and character information.

Each tweet is represented as two separate sequences: words and characters, both of which are processed with separate convolutional channels. Each element in these sequences is represented with a latent feature vector, i.e. an embedding. The word embeddings are initialized using word2vec[4] trained with approximately 1 billion drug related tweets as provided by Sarker and Gonzalez[5]. We also tested GloVe vectors trained on 2B general domain tweets[6], but these experiments resulted in a decreased performance. The character embeddings are initialized randomly, but the network is allowed to backpropagate to both word and character embeddings.

The convolutional kernels are applied on the aforementioned two sequences using sliding windows. The outputs are subsequently max-pooled and concatenated. The concatenated vectors are further fed through two densely connected layers, the latter having the output dimensionality corresponding to the number of labels in the data set with softmax activation.

In addition to the convolutional layers we utilize the same TF–IDF weighted sparse vector representations as in the baseline method. As these representations have dimensionality in the order of hundreds of thousands, we first densify the representations to 4000 dimensional vectors using truncated singular-value decomposition (SVD)[7]. These vectors are concatenated alongside the CNN outputs. This dimensionality reduction is performed mainly due to computational reasons since the approach was prototyped on a consumer grade GPU with limited amount of memory. Projecting the sparse vectors to 4K dimensions preserves 74% of the variance in the data and may have caused a minor performance loss.

The network is trained on the official training data using the Nadam optimization algorithm. The network is regularized with dropout rate of 0.2 after the first dense layer, no explicit regularization is applied on the convolutional part of the network. The training is stopped once the performance on the development set is no longer improving, measured with the official evaluation metric. Table 1 shows the comprehensive list of used hyperparameters.

Hyper-parameter	**Optimal value**	**Tested Values**
Character embedding dimensionality	25	[25,50,75,100]
Word embedding dimensionality	400	pre-trained
Character CNN, number of filters per window size	50	[50,100,150,200]
Character CNN, window sizes	[2,3,4,5]	[2,3,4,5]
Word CNN, number of filters per window size	200	[100,200,300]
Word CNN, window sizes	[2,4]	any subset of [2,3,4,5]
Dimensionality of first dense layer	400	[100,200,300,400,500]
Dropout rate	0.2	[0,0.2,0.5]
Activation functions	tanh	[ReLU, tanh, sigmoid]

Table 1: The optimal and tested hyperparameter values of the CNN-based system.

Training the network on this dataset resulted in relatively large variance in the measured performance, caused by the random initialization of the weights. Thus we stabilize the system by training 15 networks, all identical apart from the initial (random) weights. We then select the optimal subset of these networks, as measured on the development set, for the final system where the final predictions are created by summing the confidences of all selected networks and choosing the label with the highest overall confidence. The final system included a subset of 6 neural networks out of the 15. We note that this approach may potentially overfit on the development set.

Other NN architectures experimented and tested during this shared task include various versions of BiLSTM and attention based networks[8,9] but none of these experiments resulted in better performance than the CNN architecture described in detail. However, due to the time limits of the shared task, we cannot reject the possibility of these approaches being competitive as well.

We also experimented with a way of (pre-)tuning the utilized word embeddings to this specific classification task in an attempt to give the word-level CNN a better starting point for the training. This was done using the principles underlying the random indexing (RI)[10] method. Unique index vectors are first assigned to each of the three classes (intake, possible intake and non-intake), and empty context vectors are assigned to each word in the data set. When traversing the training set, each word, in each tweet, have the index vector associated with the tweet's class added to their context vector. After training, the resulting word context vectors are normalized to unit length and summed with the corresponding word embeddings/vectors generated using word2vec[5] (also normalized to unit length). To make the signal provided by the RI approach have a modest impact on the conjoint vectors, these vectors are first multiplied with a weight of 0.3. However, the described approach did not seem to result in a positive performance impact, compared to using the original word2vec generated embeddings.

We also tested the potential benefits of including part-of-speech (POS) tags, which were produced using the Twitter NLP toolkit[11]. The sequences of POS tags were treated in similar fashion to the word and character sequences. Although the benefits of POS tagging are intuitive as for instance verbs in past tense are twice as common in the *intake* class as in the *possible intake*, we did not see any increase in the performance when POS tags were utilized.

Results and Discussion

We measure the performance of our systems using micro-averaged F-score of *intake* and *possible intake* classes, following the official evaluation, and conduct all our experiments on the official development set. However, the reported results are not directly comparable with other systems as we only had access to a subset of the original data (see the Data section). The results on the test set are as reported by the organizers and thus comparable to other systems.

The overall performance of our baseline (i.e. SVM) and CNN-based systems are relatively strong, resulting in F-scores of 69.6 and 72.7 on the development set respectively (see Table 2). We suspect the main advantage of the CNN approach to be the generalizability of the word embeddings, which leads to the 3.1pp improvement in F-score. We also briefly tested a nonlinear multilayer perceptron with the same BOW features as used in the SVM, which led to a slight improvement over using the linear SVM model, but was not able to outperform our CNN-based system. Thus the model complexity alone does not explain the performance difference between the SVM and CNN approaches.

An unexpected observation is that the *intake* class seems to be harder to predict than the *possible intake*, although eyeballing the data suggests otherwise and the annotation guidelines provide more precise definition for the *intake* class. Also for the CNN-based system it seems that the precision and recall are rather well balanced, thus no performance improvements could have been gained through further fine tuning of these metrics.

The test set results follow the same patterns as the development set evaluation: SVM and CNN systems reach F-scores of 64.2 and 66.3, respectively. Thus it seems that either the test set is somewhat harder than the development set or both of the systems are overfitting equally on the development set, even though the implemented ensemble system with CNNs could have caused greater overfitting. According to the official evaluation, our best system loses to the winning system by 3pp in F-score, the difference being roughly the same in both precision and recall. This places our system in the 5th position in the shared task.

By inspecting the confusion matrices it can be concluded that our classifiers tend to confuse *intake* class with both

		Development set			Test set		
		Precision	Recall	F-score	Precision	Recall	F-score
SVM	Intake	70.5	64.5	67.4			
	Possible Intake	73.3	68.6	70.9			
	Overall	72.3	67.0	69.6	69.2	60.1	64.3
CNN	Intake	70.9	71.3	71.1			
	Possible Intake	76.3	71.1	73.6			
	Overall	74.2	71.2	72.7	70.1	63.0	66.3
InfyNLP	Overall				72.5	66.4	69.3

Table 2: Overall performance of our SVM and CNN-based systems. The development set results are measured with our own evaluation whereas the test set scores are as reported by the organizers. The class specific performance was not evaluated by the organizers and has been thus left out from the table. For comparison we have added the results of the best performing team: InfyNLP.

possible intake and *non-intake* classes equally often, whereas the *possible intake* is more often confused with the *non-intake* class.

As we only had access to a partial training data, we try to estimate how much the performance of the systems could have been improved with additional data. To accomplish this, we train the CNN-based system with different subsets of the training data, starting from 5K training examples and incrementally increasing the size in steps of 300 up to the whole training data available to us. After every increment we evaluate the system's performance on the development set. To reduce the variance caused by different initial random weights, we train 5 networks with each subset of the training data, and calculate the mean performance for each subset. Fitting a linear regression on the resulting measurements shows that in this region, the learning curve is fairly linear and decent performance improvements can be gained by adding more training data. Assuming a performance increase equal to the slope of the fitted regression line, having the full training dataset would have increased our performance by 0.7pp in F-score, placing our system close to the top 3 teams in the shared task.

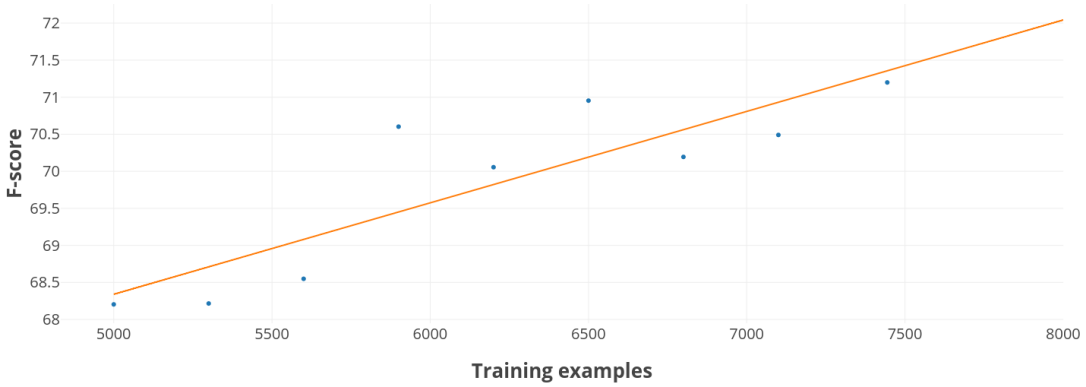

Figure 1: Influence of the number of training examples to the performance of the CNN system as evaluated on the development set.

As most approaches we tested, as well as the systems created by other teams, resulted roughly in the same performance level, we wanted to assess what would be a theoretical performance limit for this task. To this end, we manually annotated a random subset of 100 tweets from the development set and evaluated the annotations against the gold standard. Surprisingly our manual annotations reached only an F-score of 59.3, notably lower than the developed systems or what the official inter-annotator agreement would suggest[12]. This indicates that the task is complex even for humans and deep understanding of the annotation guidelines is required for high quality annotations.

Conclusions and Future Work

We have shown that strong results in detecting tweets describing personal medication intake can be achieved using convolutional neural networks and word embeddings. However, more traditional methods relying on bag-of-words features and linear classifiers also result in competitive performance. Considering that such a system can be implemented in less than an hour with the existing tools and libraries, and is easily interpretable, the simpler methods may be a more practical choice in many use cases.

Since the amount of training data for this task is fairly limited, we plan to explore various approaches for pretraining NN classifiers as a future task. The goal here is to find a suitable proxy task related to the domain for initializing the network before the actual training. Such a task could be, for instance, sentiment detection as many of the tweets expressing drug intake also express a certain sentiment about the condition of the user or the effects of the drug.

Acknowledgements

This work was supported by ATT Tieto käyttöön grant and Tekes – Räätäli project (no. 644/31/2015). Computational resources were provided by CSC - IT Center For Science Ltd., Espoo, Finland.

References

1. Salton G, Buckley C. Term-weighting approaches in automatic text retrieval. Information processing & management. 1988;24(5):513–523.

2. Joachims T. Making large-scale SVM learning practical. In: Schölkopf B, Burges C, Smola A, editors. Advances in Kernel Methods – Support Vector Learning. Cambridge, MA: MIT Press; 1999. p. 169–184.

3. LeCun Y, Bengio Y, et al. Convolutional networks for images, speech, and time series. The handbook of brain theory and neural networks. 1995;3361(10):1995.

4. Mikolov T, Sutskever I, Chen K, Corrado GS, Dean J. Distributed representations of words and phrases and their compositionality. In: Advances in Neural Information Processing Systems 26; 2013. p. 3111–3119.

5. Sarker A, Gonzalez G. A corpus for mining drug-related knowledge from Twitter chatter: language models and their utilities. Data in Brief. 2017;10(Supplement C):122 – 131.

6. Pennington J, Socher R, Manning C. Glove: global vectors for word representation. In: Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP); 2014. p. 1532–1543.

7. Halko N, Martinsson PG, Tropp JA. Finding structure with randomness: stochastic algorithms for constructing approximate matrix decompositions. 2009;.

8. Hochreiter S, Schmidhuber J. Long short-term memory. Neural computation. 1997;9(8):1735–1780.

9. Luong MT, Pham H, Manning CD. Effective approaches to attention-based neural machine translation. arXiv preprint arXiv:150804025. 2015;.

10. Kanerva P, Kristofersson J, Holst A. Random indexing of text samples for latent semantic analysis. In: Proceedings of 22nd Annual Conference of the Cognitive Science Society. Philadelphia, PA, USA; 2000. p. 1036.

11. Owoputi O, O'Connor B, Dyer C, Gimpel K, Schneider N, Smith NA. Improved part-of-speech tagging for online conversational text with word clusters. Association for Computational Linguistics; 2013. .

12. Klein A, Sarker A, Rouhizadeh M, O'Connor K, Gonzalez G. Detecting personal medication intake in Twitter: an annotated corpus and baseline classification system. In: BioNLP 2017. Vancouver, Canada,: Association for Computational Linguistics; 2017. p. 136–142.