Classification of medication-related tweets using stacked bidirectional LSTMs with context-aware attention

Orest Xherija
Department of Linguistics
University of Chicago
Chicago, IL, USA
orest.xherija@uchicago.edu

Abstract

This paper describes the system that team UChicagoCompLx developed for the 2018 Social Media Mining for Health Applications (SMM4H) Shared Task. We use a variant of the Message-level Sentiment Analysis (MSA) model of (Baziotis et al., 2017), a word-level stacked bidirectional Long Short-Term Memory (LSTM) network equipped with attention, to classify medication-related tweets in the four subtasks of the SMM4H Shared Task. Without any subtask-specific tuning, the model is able to achieve competitive results across all subtasks. We make the datasets, model weights, and code publicly available[1].

1 Introduction

The Shared Task of the 2018 Social Media Mining for Health Applications (SMM4H) workshop (Weissenbacher et al., 2018) proposed four subtasks in the domain of social media mining for health monitoring and surveillance. From a Natural Language Processing (NLP) viewpoint, these tasks present a considerable challenge since the nature of social media posts requires dealing with both a significant level of language variation and a widespread presence of noise (spelling mistakes, syntactic errors etc). Any classifier designed for this textual domain should take into account the above intricacies and should, furthermore, be able to deal with with semantic complexities in the various ways people express medication-related concepts and outcomes.

To address these challenges, we use a variant of the Message-level Sentiment Analysis (MSA) model of (Baziotis et al., 2017), originally developed for sentiment analysis of Twitter posts, to classify tweets in all four subtasks. The model is a word-level stacked bidirectional LSTM (BiLSTM) with context-aware attention that uses word-embeddings pretrained by (Baziotis et al., 2017) on a corpus of $\approx$ 330M tweets. Without additional hyperparameter tuning or subtask-specific modifications, the model outperforms the average of all submitted systems in subtasks 1 and 4 and achieves first place (by a F1-score margin of 0.234 from the next team) in subtask 2. In subtask 3 our model was placed 6th out of 9 systems.

In the following sections, we introduce the datasets, discuss preprocessing steps we took, present the model and its training setup, report results, and conclude with potential avenues for future research.

2 Datasets

In this section, we describe the datasets of each subtask. Subtasks 1, 3 and 4 are binary classification problems while subtask 2 is a three-class classification problem. The data was manually annotated by the organizers.

Subtask 1 is about the automatic detection of posts mentioning the name of a drug or dietary supplement, as defined by the United States Food and Drug Administration (FDA). A tweet is assigned label 1 if it contains the name of one or more drugs or supplements and 0 otherwise. **Subtask 2** poses the challenge of automatic classification of posts describing medication intake. A tweet is assigned label 1 if "the user clearly expresses a personal medication intake/consumption", 2 if the tweet suggests (without certainty) that "the user may have taken the medication", and 3 if the tweet mentions medication names but does not indicate personal intake. **Subtask 3** concerns the automatic classification of posts mentioning an adverse drug reaction (ADR). A tweet is assigned label 1 if it men-

[1] https://github.com/orestxherija/smm4h2018

Proceedings of the 3rd Social Media Mining for Health Applications (SMM4H) Workshop & Shared Task, pages 38–42
Brussels, Belgium, October 31, 2018. ©2018 Association for Computational Linguistics

	1	2	3	4
training	7011	13791	21062	6956
validation	780	1533	2341	CV
evaluation	5382	5000	5000	161

Table 1: Examples per split per task. CV indicates cross-validation, so no validation set was held out.

tions an ADR and 0 otherwise. Finally, **Subtask 4** deals with the automatic detection of posts mentioning vaccination behavior related to influenza vaccines. The annotators were asked the question "Does this message indicate that someone received, or intended to receive, a flu vaccine?" and a tweet was assigned label 1 if the answer was affirmative and 0 otherwise. Subtasks 1, 3 and 4 are evaluated using the F1-score for the positive class while subtask 2 uses the micro-averaged F1-score for classes 1 and 2. Subtask 1 is additionally evaluated on precision and recall for the positive class.

Due to Twitter privacy policies, the training set for any subtask did not contain the actual tweet text. To obtain said text, participants were provided with the tweet ID of each dataset example along with a script to use for downloading the text using this ID. The process inevitably resulted in fewer tweets than the number of IDs contained in the original dataset, primarily because a number of tweets had been removed (either by the users themselves, or by Twitter because e.g. the user deleted his account) while others failed to download (due to e.g. lag issues when requesting the HTML of the tweet). To avoid such issues in the evaluation datasets, the organizers decided to provide the tweet text along with the ID. Table [1] provides a short summary of the number of tweets that were available to our team for each subtask.

3 Pre-processing

We applied identical preprocessing to all datasets. We replaced Twitter specific strings with appropriate tokens (e.g. emojis were replaced by **$EMOJI$**, numbers were replaced by **$NUMBER$**, website urls by **URL** etc) to reduce the vocabulary size and to ameliorate the noisy nature of the text. All non-alphanumeric characters and all tokens that were too short (fewer than 2 characters) or too long (more than 15 characters) were removed. Finally, all text was converted to lower case and any excess whitespace (i.e. newlines and tabs) was removed.

4 Model description

4.1 Model architecture

We use a variant of the Message-level Sentiment Analysis (MSA) model of (Baziotis et al., 2017). The model consists of two stacked BiLSTMs with a context-attention mechanism à la (Yang et al., 2016) that identifies the maximally informative words for each label. We describe subsequently the individual network layers.

The input is a tweet, regarded as a sequence of words, which is projected to a vector space of fixed size via the **Embedding Layer**. The weights of the embedding layer are initialized using pre-trained word embeddings that (Baziotis et al., 2017) trained on a Twitter corpus of approximately $\approx 330M$ tweets. We opt for these embeddings instead of the standard Word2Vec (Mikolov et al., 2013a,b) ones since they have been trained on a similar textual domain to the tasks at hand.

A **LSTM Layer** placed on top of the embedding layer takes as input the embedding weights and produces a representation $\{h_i\}_{i=1}^{T}$ where, h_i is the hidden state of the LSTM at time-step i, intuitively corresponding to a summary of all the information of the sentence (viewed as a sequence $\{w_i\}_{i=1}^{T}$ of words) up to w_i. This constitutes a forward LSTM. Since we are using a bidirectional LSTM, we also have an LSTM that scans the sequence of words in the reverse direction. The final representation of a word is produced by concatenating the representations from the forward and backward LSTM:

$$h_i = \overrightarrow{h_i} || \overleftarrow{h_i} \tag{1}$$

where || denotes the concatenation operator. We opt for a stacked BiLSTM, and consequently we place an additional BiLSTM layer on top of the preceding layer. The motivation for this choice comes from the literature on the interpretation of hidden states of Recurrent Neural Networks (RNNs) (Belinkov et al., 2017; Belinkov, 2018) in which it has been claimed that deeper layers are able to learn more abstract semantic representations of sentences, thus achieving superior performance in downstream tasks.

To account for the fact that not all words contribute equally to the assignment of a label, we place an **Attention Layer** on top of the BiLSTMs following work like (Sutskever et al., 2014) who successfully used attention mechanisms for sequence-to-sequence neural machine translation. We use context-attention, following (Yang et al., 2016). A context vector u_h is initialized and is governed by the following update equations:

$$e_i = \tanh(W_h h_i + b_h) \tag{2}$$

$$a_i = \frac{\exp(e_i^\top u_h)}{\sum_{j=1}^{T} \exp(e_j^\top u_h)} \qquad \sum_{i=1}^{T} a_i = 1 \tag{3}$$

$$r = \sum_{i=1}^{T} a_i h_i \qquad r \in \mathbb{R}^{2L} \tag{4}$$

where W_h, b_h and u_h are learned parameters, h_i is the concatenation of the representations of the forward and backward LSTM, introduced in equation (1), and L is the number of cells in one LSTM layer.

Finally, we feed the representation r produced by the attention layer to a **Dense Layer** with sigmoid activation (softmax for subtask 2) and obtain a probability distribution over the classes. If the probability assigned to a tweet is greater than 0.5 we assign label 1, otherwise we assign 0.

4.2 Training setup

We train the model to minimize the negative log-likelihood loss using back-propagation with stochastic gradient descent and mini-batch size of 50. We use the Adam optimizer (Kingma and Ba, 2015) with gradient norm clipping (Pascanu et al., 2013) at 1. For subtasks 1, 2 and 3 we use a $90 - 10$ train-validation split, while for subtask 4 we use $10-$fold stratified cross-validation in consideration of the very small test set. Table [1] summarizes the information on train-validation splits.

4.3 Regularization

To make the model more robust to over-fitting, we employ, following (Baziotis et al., 2017), a number of regularization techniques. We add Gaussian noise at the embedding layer and use dropout (Srivastava et al., 2014) to ignore the signal from a set of randomly selected neurons in the network. Dropout is also applied after each LSTM layer as well as to the recurrent connections of the LSTM (Gal and Ghahramani, 2016). L_2 regularization along with class weights are applied to the loss function to prevent overly large weights and to account for class imbalance. Class weights are computed as follows: assuming that $\vec{x}$ is the vector of class counts, the weights are defined as $w_i = \max(\vec{x})/x_i$ for any class i. Finally, early-stopping (Caruana et al., 2001) is employed to terminate training after the validation loss has stopped decreasing.

4.4 Hyperparameter tuning

We use the similar hyperparameters to (Baziotis et al., 2017). In particular, we use 150 as the size of the LSTM hidden states (300 in total since we are using a BiLSTM), the Gaussian noise parameter is set to $\sigma = 0.3$, dropout rate on top of the embedding layer is set to 0.3 and dropout rate on top of the LSTM layers is set to 0.5. Dropout at the recurrent connections is also set to 0.3. L_2 regularization at the loss function is set to 0.0001. Finally, we initialize the learning rate at 0.001. Departing from (Baziotis et al., 2017), we use word embeddings of dimension 100. Vocabulary size and maximum sequence length are set to 7000 and 50 respectively for all subtasks and the patience level for early-stopping is set to 0.001 in 5 epochs.

5 Experiments and results

5.1 Experimental setup

The model was developed using Keras[2] with the Tensorflow (Abadi et al., 2016) backend. For data preparation and processing we use Scikit-learn (Pedregosa et al., 2011). Given the small size of the datasets, we do not use GPUs for training the model. A standard 8-core CPU is sufficient. Finally, for designing the network architecture, we use part of the code released by (Baziotis et al., 2017)[3].

5.2 Results

For subtasks 1 and 4, the organizers chose to disclose to each team only their respective score along with the average score of all submitted systems. These results are summarized in Table [2]. Our system performed better than the average in both subtasks, considerably so in subtask 1.

For subtasks 2 and 3, the organizers released the complete leaderboards, presented in Tables [3] and [4] respectively. Our system greatly outperformed all other systems by a significant margin in subtask 2. In subtask 3, our system ranked 6th (out of 9 participants), potentially because the other teams developed specialized systems for the particular

[2] https://keras.io/

[3] https://github.com/cbaziotis/datastories-semeval2017-task4

	P	R	F1
Subtask-1	**0.937**	**0.891**	**0.914**
	(0.890)	(0.872)	(0.880)
Subtask-4	0.791	0.923	**0.852**
	(0.826)	(0.858)	(0.840)

Table 2: Results on the evaluation set for subtasks 1 and 4. Average score of all participating systems in parentheses. Metric is F1-score for class 1. For subtask 1, precision and recall for class 1 are also used for evaluation.

	P	R	F1
UChicagoCompLx	**0.654**	**0.783**	**0.713**
Light	0.492	0.467	0.479
Tub-Oslo	0.464	0.466	0.465
IRISA_team	0.434	0.501	0.465
IIT_KGP	0.408	0.407	0.408
UZH	0.371	0.437	0.401
CLaC	0.402	0.366	0.383
Techno	0.327	0.432	0.372

Table 3: Subtask 2 final leaderboard. Metric is micro-averaged F1-score for classes 1 and 2.

subtask while we opted for a general model that can be used without modifications in all four subtasks.

6 Conclusion and future directions

We demonstrated that the variant of the MSA model of (Baziotis et al., 2017) performs competitively when applied to the domain of medication-related short text classification. Without hyperparameter tuning, major architectural modifications, or task-specific adjustments, the model obtained competitive results in subtasks 1 and 4 and ranked first in subtask 3, greatly outperforming all other models in terms of precision, recall and F1-score. The model's performance in this Shared Task is further testament to the ability of attentive RNNs to perform at state-of-the-art level in short text classification where individual word-meaning is essential.

	P	R	F1
THU_NGN	0.442	0.636	**0.522**
IRISA_team	0.378	**0.649**	0.478
UZH	0.455	0.436	0.445
Tub-Oslo	**0.638**	0.317	0.424
Art	0.332	0.547	0.413
UChicagoCompLx	0.370	0.464	0.411
CIC-NLP	0.314	0.529	0.394
Techno	0.434	0.344	0.383
IIT_KGP	0.189	0.643	0.292

Table 4: Subtask 3 final leaderboard. Metric is F1-score for class 1.

In the future, we aim to investigate whether ensembles of word- and character-level attentive RNNs can perform even better. The benefits of ensembling for text classification can be seen in numerous NLP tasks ranging from Natural Language Inference (Gong et al., 2018, among many others) to product categorization (Skinner, 2018). Word-level models perform well in capturing aspects of the semantics (Belinkov et al., 2017) while character-level models succeed in capturing syntactic information about the text. Ensembles of these diverse types of models can potentially lead to improved performance.

A second avenue to pursue would be multi-task learning, an area of active research that has shown promising results in text classification (Liu et al., 2016, 2017, among others). Given that all subtasks are nearly identical in nature (all but one of them being binary classification problems) and share a highly overlapping lexicon, they provide an excellent ground for testing the merits of multi-task learning.

Acknowledgments

This work was completed in part with resources provided by the University of Chicago Research Computing Center, whose contribution we gratefully acknowledge.

References

Martín Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, Michael Isard, Manjunath Kudlur, Josh Levenberg, Rajat Monga, Sherry Moore, Derek G. Murray, Benoit Steiner, Paul Tucker, Vijay Vasudevan, Pete Warden, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. 2016. TensorFlow: A System for Large-Scale Machine Learning. In *12th USENIX Symposium on Operating Systems Design and Implementation (OSDI 16)*, pages 265–283, Savannah, GA, USA. USENIX Association.

Christos Baziotis, Nikos Pelekis, and Christos Doulkeridis. 2017. DataStories at SemEval-2017 Task 4: Deep LSTM with Attention for Message-level and

Topic-based Sentiment Analysis. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 747–754, Vancouver, Canada. Association for Computational Linguistics.

Yonatan Belinkov. 2018. *On Internal Language Representations in Deep Learning: An Analysis of Machine Translation and Speech Recognition*. Ph.D. thesis, Massachusetts Institute of Technology.

Yonatan Belinkov, Lluís Màrquez, Hassan Sajjad, Nadir Durrani, Fahim Dalvi, and James Glass. 2017. Evaluating Layers of Representation in Neural Machine Translation on Part-of-Speech and Semantic Tagging Tasks. In *Proceedings of the Eighth International Joint Conference on Natural Language Processing*, volume 1, pages 1–10, Taipei, Taiwan. Asian Federation of Natural Language Processing.

Rich Caruana, Steve Lawrence, and Lee Giles. 2001. Overfitting in Neural Nets: Backpropagation, Conjugate Gradient, and Early Stopping. In Todd K. Leen, Thomas G. Dietterich, and Volker Tresp, editors, *Advances in Neural Information Processing Systems*, volume 13, pages 402–408. MIT Press, Denver, CO, USA.

Yarin Gal and Zoubin Ghahramani. 2016. Dropout As a Bayesian Approximation: Representing Model Uncertainty in Deep Learning. In *Proceedings of The 33rd International Conference on Machine Learning*, volume 48 of *Proceedings of Machine Learning Research*, pages 1050–1059, New York, NY, USA. PMLR.

Yichen Gong, Heng Luo, and Jian Zhang. 2018. Natural Language Inference over Interaction Space. In *International Conference on Learning Representations*, Vancouver, Canada.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A Method for Stochastic Optimization. In *International Conference on Learning Representations*, San Diego, CA, USA.

Pengfei Liu, Xipeng Qiu, and Xuanjing Huang. 2016. Deep Multi-Task Learning with Shared Memory for Text Classification. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 118–127, Austin, Texas. Association for Computational Linguistics.

Pengfei Liu, Xipeng Qiu, and Xuanjing Huang. 2017. Adversarial Multi-task Learning for Text Classification. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics*, volume 1, pages 1–10, Vancouver, Canada. Association for Computational Linguistics.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013a. Efficient Estimation of Word Representations in Vector Space. In *International Conference on Learning Representations*, Scottsdale, AZ, USA.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013b. Distributed Representations of Words and Phrases and Their Compositionality. In Christopher J. C. Burges, Léon Bottou, Max Welling, Zoubin Ghahramani, and Kilian Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 26, pages 3111–3119. Curran Associates, Inc., Lake Tahoe, CA, USA.

Razvan Pascanu, Tomas Mikolov, and Yoshua Bengio. 2013. On the Difficulty of Training Recurrent Neural Networks. In *Proceedings of the 30th International Conference on Machine Learning*, volume 28 of *Proceedings of Machine Learning Research*, pages 1310–1318, Atlanta, GA, USA. PMLR.

Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, Jake VanderPlas, Alexandre Passos, David Cournapeau, Matthieu Brucher, Matthieu Perrot, and Edouard Duchesnay. 2011. Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

Michael Skinner. 2018. Product Categorization with LSTMs and Balanced Pooling Views. In *SIGIR 2018 Workshop on eCommerce (ECOM 18)*, SIGIR '18, Ann Arbor, MI, USA. ACM.

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: A Simple Way to Prevent Neural Networks from Overfitting. *Journal of Machine Learning Research*, 15:1929–1958.

Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to Sequence Learning with Neural Networks. In Zoubin Ghahramani, Max Welling, Corinna Cortes, Neil D. Lawrence, and Kilian Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 27, pages 3104–3112. Curran Associates, Inc., Montréal, Canada.

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 3rd Workshop Social Media Mining for Health Applications (SMM4H)*, Brussels, Brussels. Association for Computational Linguistics.

Zichao Yang, Diyi Yang, Chris Dyer, Xiaodong He, Alex Smola, and Eduard Hovy. 2016. Hierarchical Attention Networks for Document Classification. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1480–1489, San Diego, CA, USA. Association for Computational Linguistics.