Knowledge discovery from patient forums: gaining novel medical insights from patient experiences

Dirkson, A.R.

PART II:

EXTRACTING ADVERSE DRUG EFFECTS (ADES)

For some minutes it puffed away without speaking, but at last it unfolded its arms, took the hookah out of its mouth again, and said, “So you think you’re changed, do you?”
...
“I’m afraid I am, sir,” said Alice; “I can’t remember things as I used—and I don’t keep the same size for ten minutes together!"

Lewis Caroll, *Alice in Wonderland*

5

TRANSFER LEARNING FOR ADE EXTRACTION FROM TWITTER

Edited from: **Anne Dirkson** & Suzan Verberne (2019), Transfer Learning for Health-related Twitter Data. Proceedings of the Fourth Social Media Mining for Health Applications (SMM4H) Workshop & Task. Association for Computational Linguistics. 89-92.

In this chapter, we introduce the use of transfer learning methods for extracting and normalizing adverse drug events from Twitter data. We also apply transfer learning to the task of identifying personal health mentions in health-related tweets.

Transfer learning is an umbrella term for methods that re-use a model trained on one (usually larger) set of data as a starting point for training a model for another task. These methods are especially promising for domains that suffer from a shortage of annotated data or resources, such as health-related social media.

This work was done as part of the 2019 Social Media Mining for Health Applications (SMM4H) Shared Task.

5.1. INTRODUCTION

Transfer learning is promising for NLP applications, as it enables the use of universal pre-trained language models (LMs) for domains that suffer from a shortage of annotated data or resources, such as health-related social media. Universal LMs have recently achieved state-of-the-art results on a range of NLP tasks, such as classification [137] and named entity recognition (NER) [3]. For the Shared Task of the 2019 Social Media Mining for Health Applications (SMM4H) workshop we focused on employing state-of-the-art transfer learning with universal LMs to investigate its potential in this domain.

5.2. TASK DESCRIPTIONS

The SMM4H shared task consisted of four subtasks:

ADE extraction The purpose of **Subtask 1** (S1) is to classify tweets as containing an adverse drug event (ADE) or not. Subsequently, these ADE mentions are extracted in **Subtask 2** (S2) and normalized to MedDRA concept IDs in **Subtask 3** (S3). MedDRA (Medical Dictionary for Regulatory Activities) is an international, standardized medical terminology.[1]

Personal Health Mention Extraction The goal of **Subtask 4** (S4) is to identify tweets that are personal health mentions, i.e., posts that mention a person who is affected as well as their specific condition [152], as opposed to posts discussing health issues in general. Generalisability to both future data and different health domains is evaluated by including data from the same domain collected years after the training data, as well as data from an entirely different disease domain.

5.3. OUR APPROACH

5.3.1. PREPROCESSING

We preprocessed all Twitter data using the lexical normalization pipeline by Sarker [261]. We also employed an in-house spelling correction method (see Chapter 2). Additionally, punctuation and non-UTF-8 characters were removed using regular expressions.

5.3.2. ADDITIONAL DATA

Personal Health Mentions For S4, the training data consists of data from one disease domain, namely influenza, in two contexts: having a flu infection and getting a flu vaccination. To improve generalisability, we supplemented this data with six labeled data sets from different disease domains [152]. We refer to this combined data set as S4+. For each subset, 10% was used for a combined validation set. For fine-tuning the ULMfit universal language model based on 28,595 Wikipedia articles (Wikitext-103) [200], the DIEGO Drug Chatter corpus [263] was combined with the data from S1 and S4+ to form a larger unsupervised corpus of health-related Twitter data ('TwitterHealth'). For S4, fine-tuning was also attempted with only the S4+ data.

[1] `https://www.meddra.org/`

	S1	S2*	S3	S4	S4+
Dev	-	130	76	-	-
Train	14,634	910	1,756	6,996	11,832
Validation	1,626	130	76	777	1,314
Test	5000	1000	1000	ND	ND

Table 5.1: Data sets. *Only tweets containing an ADE were used for developing the system. ND: Not disclosed

Concept Normalization The MedDRA concept names and their aliases in both MedDRA and the Consumer Health Vocabulary[2] were used to supplement the data from S3. This data set is hereafter called S3+.

5.3.3. TEXT CLASSIFICATION (S1 AND S4)

Text classification was performed with fast.ai ULMfit [137]. As recommended, the initial learning rate (LR) of 0.01 was determined manually by inspecting the log LR compared to the loss. Default language models were fine-tuned using AWD_LSTM [201] with (1) 1 cycle (LR = 0.01) for the last layer and then (2) 10 cycles (LR = 0.001) for all layers.

Subsequently, this model is used to train a classifier with F_1 as the metric, a dropout of 0.5 and a momentum of (0.8,0.7), in line with the recommendations. Training is done with (1) 1 cycle (LR = 0.02) on the last layer; (2) unfreezing of the second-to-last layer; (3) another cycle running from a 10-fold decrease of the previous LR to this LR divided by 2.6[4] (as recommended in the fast.ai MOOC).[3] This is repeated for the next layer and then for all layers. The last step consists of multiple cycles until F_1 starts to drop.

As an alternative classifier for S1, we used the absence of ADEs (noADE) according to the Bert embeddings NER method (see below) which was developed for the subsequent sub-task (S2) and aims to extract these ADE mentions. As a baseline for text classification, we used a Linear SVC with unigrams as features. The C parameter was tuned with a grid of 0.0001 to 1000 (steps of x10).

5.3.4. NAMED ENTITY RECOGNITION (S2)

We experimented with different combinations of state-of-the-art Flair embeddings [3], classical Glove embeddings and Bert embeddings [84] using the Flair package. We used pretrained Flair embeddings based on a mix of Web data, Wikipedia and subtitles; and the 'bert-base-uncased' variant of Bert embeddings. We also experimented with Flair embeddings combined with Glove embeddings (dimensionality of 100) based on FastText embeddings trained on Wikipedia (GloveWiki) or on Twitter data (GloveTwitter). Training for all embeddings was done with an initial LR of 0.1, batch size of 32, and max epochs set to 150.

As a baseline for NER, we used a CRF with the default L-BFGS training algorithm with Elastic Net regularization. As features for the CRF, we used the lower-cased word, its suffix, the word shape and its POS tag.[4]

[2] https://www.nlm.nih.gov/research/umls/sourcereleasedocs/current/CHV/
[3] https://course.fast.ai/
[4] https://sklearn-crfsuite.readthedocs.io/en/latest/tutorial.html

5.3.5. CONCEPT NORMALIZATION (S3)

Pretrained Glove embeddings were used to train document embeddings on the extracted ADE entities in the S3 data including or excluding the aliases from CHV (S3+) with concept IDs as labels. We used the default RNN in Flair with a hidden size of 512. Glove embeddings (dim = 100) were based on FastText embeddings trained on Wikipedia. Token embeddings were re-projected (dim = 256) before inputting to the RNN.

5.4. RESULTS

For all four subtasks, our best transfer learning system consistently performs better than the average over all runs submitted to SMM4H. For classifying ADE mentions, our overall best performing system is a ULMfit model trained on the TwitterHealth corpus (see Table 5.2). Yet, the highest recall is attained by using the absence of named entities (noADE) as a classifier. This is in line with our validation results (see Table 5.3). For extracting ADEs, our best system combines Bert with Flair embeddings without a separate classifier for sentences containing ADE mentions (see Table 5.4). However, using Bert embeddings alone *with* the ULMfit classifier from S1 appears to be more precise. During validation, we found that a combination of Glove embeddings (based on Twitter or Wikipedia) and Flair embeddings performed poorly compared to the submitted systems (see Table 5.5). For mapping the ADEs to MedDRA concepts, we only submitted one system with different preceding NER models (see Table 5.6), since adding the alias information (S3+) decreased both precision and recall (see Table 5.7). Our RNN document embeddings with only the S3 data, however, performed better than average. Lastly, for the classification of personal health mentions, our best classifier was a ULMfit model fine-tuned on the S4+ data (see Table 5.8), which outperformed the average result and the ULMfit model trained on the larger TwitterHealth corpus on all metrics. This system similarly outperformed the other ULMfit model on the validation data (see Table 5.9).

	Method	F_1 (range)	P	R
Average*		0.502 (0.331)	0.535	0.505
Run1	ULMfit[1]	**0.533**	**0.642**	0.455
Run2	noADE	0.418	0.284	**0.792**

Table 5.2: Results for ADE Classification (S1). *over all runs submitted [1]TwitterHealth data

Method	F_1	P	R
Baseline: Linear SVC (C=1.0)	0.475	0.526	0.433
ULMfit[1]	**0.574**	**0.574**	0.574
noADE	0.330	0.207	**0.823**

Table 5.3: Validation results for ADE classification (S1) [1]TwitterHealth data

		Relaxed			Strict		
	Method	F_1 (range)	P	R	F_1 (range)	P	R
Average*		0.538 (0.486)	0.513	0.615	0.317 (0.422)	0.303	0.358
Run1	Bert+Flair$^+$	**0.625**	0.555	**0.715**	**0.431**	0.381	**0.495**
Run2	Bert$^+$	0.622	0.560	0.701	0.427	0.382	0.484
Run3	Bert+ADEClassifier	0.604	**0.718**	0.521	0.417	**0.494**	0.360

Table 5.4: Results for ADE Extraction(S2). *over all runs submitted $^+$No separate classifier for sentences containing ADE

Method	Micro-F_1	P	R
Baseline: CRF	0.235	0.560	0.149
Flair+ GloveWiki	0.596	0.666	0.540
Flair+ GloveTwitter	0.577	0.655	0.515
Bert	0.640	**0.699**	0.590
Bert+Flair	**0.649**	**0.699**	**0.606**

Table 5.5: Validation results for ADE extraction (S2)

		Relaxed			Strict		
	Method	F_1 (range)	P	R	F_1 (range)	P	R
Average*		0.297 (0.242)	0.291	0.312	0.212 (0.247)	0.205	0.224
Run1$^+$	RNN Docemb.	**0.312**	**0.370**	0.270	**0.250**	**0.296**	0.216
Run2$^+$	RNN Docemb.	0.303	0.272	0.343	0.244	0.218	0.277
Run3$^+$	RNN Docemb.	0.302	0.267	**0.347**	0.246	0.218	**0.283**

Table 5.6: Results for concept normalization (S3). *over all runs submitted $^+$Runs same as S2 prior to concept normalization

Method	F_1	P	R
RNNDocembeddings with S3	**0.623**	**0.566**	**0.694**
RNNDocembeddings with S3+	0.253	0.171	0.482

Table 5.7: Validation results for concept normalization (S3)

Method			Acc. (range)	F_1 (range)	P	R
Average*			0.781 (0.263)	0.701 (0.464)	0.902	0.585
Run1	ULMfit S4+ data	*Domain1*	0.869	0.859	0.952	0.781
		Domain2	0.638	0.419	0.750	0.290
		Domain3	0.786	0.539	1.000	0.368
		Mean	**0.793**	**0.726**	**0.940**	**0.591**
Run2	ULMfit TwitterHealth data	*Domain1*	0.863	0.849	0.969	0.756
		Domain2	0.609	0.342	0.700	0.226
		Domain3	0.768	0.480	1.000	0.316
		Mean	0.786	0.716	0.928	0.583

Table 5.8: Results for personal health mention classification (S4). *over all runs submitted

Method	F_1	P	R
Baseline: Linear SVC (C=0.1)	0.615	0.678	0.572
ULMfit with S4+ data	**0.712**	**0.743**	**0.701**
ULMfit with TwitterHealth data	0.692	0.738	0.676

Table 5.9: Mean validation results for personal health mention classification (S4) averaged over eight data sets of S4+

5.5. CONCLUSIONS

Transfer learning using default settings offers above average results for various NLP tasks using health-related Twitter data. More research is necessary to investigate whether state-of-the-art performance may be possible with further domain-specific adaptation, for instance by tuning hyper-parameters, training embeddings on medical data or by dealing with domain-specific vocabulary absent in the language model.