Published in final edited form as:

Proc Conf. 2019 June ; 2019: 22–27. doi:10.18653/v1/w19-1903.

Extracting Adverse Drug Event Information with Minimal Engineering

Timothy Miller[1,2], **Alon Geva**[1,2], **Dmitriy Dligach**[3]

[1]Computational Health Informatics Program, Boston Children's Hospital

[2]Harvard Medical School

[3]Department of Computer Science, Loyola University Chicago

Abstract

In this paper we describe an evaluation of the potential of classical information extraction methods to extract drug-related attributes, including adverse drug events, and compare to more recently developed neural methods. We use the 2018 N2C2 shared task data as our gold standard data set for training. We train support vector machine classifiers to detect drug and drug attribute spans, and pair these detected entities as training instances for an SVM relation classifier, with both systems using standard features. We compare to baseline neural methods that use standard contextualized embedding representations for entity and relation extraction. The SVM-based system and a neural system obtain comparable results, with the SVM system doing better on concepts and the neural system performing better on relation extraction tasks. The neural system obtains surprisingly strong results compared to the system based on years of research in developing features for information extraction.

1 Introduction

Adverse drug events (ADEs) describe undesirable signs and symptoms that occur consequent to administration of a medication. ADEs may be identified in randomized controlled trials (RCTs), observational studies, spontaneous reports such as those gathered in the Food and Drug Administrations (FDAs) Adverse Event Reporting System (FAERS), or manual chart review of data in electronic health records (EHRs). RCTs have notable limitations for pharmacoepidemiology, including strict inclusion and exclusion criteria that limit their generalizability, small cohort sizes that make them under-powered for detecting rarer ADEs, and time-limited study periods that prevent detection of ADEs that occur with longer drug administration (Sanson-Fisher et al., 2007; Sultana et al., 2013; McMahon and Dal Pan, 2018). Although drug manufacturers are required to submit postmarket adverse event reports to the FDA, this information is not uniformly available to clinicians (Maxey et al., 2013). Therefore, the 21st Century Cures Act directs the FDA to use real-world data (RWD) in the drug approval process.

Timothy.Miller@childrens.harvard.edu.

Use of RWD is particularly important for medications that are commonly used off-label, for example, those targeted for treatment of rare diseases such as pulmonary hypertension in children (Maxey et al., 2013). Electronic health records (EHRs) provide an opportunity to capture such data reflecting real-world use of approved medications. Most studies of pharmacovigilance using RWD are based on health care insurance claims—for instance, the FDAs Sentinel program—because claims data contains longitudinal information about medication dispensing and clinical diagnoses (Platt et al., 2018). However, claims data may lack sensitivity for identification of ADEs, since not all signs and symptoms are submitted to insurers for billing purposes (Nadkarni, 2010). Reliance on claims data may also lead to incongruous results, such as a Mini-Sentinel study that found—contrary to data from several large RCTs—that dabigatran was associated with a lower risk of gastrointestinal bleeding than warfarin (Sipahi et al., 2014).

Limiting studies using RWD to structured data alone neglects the rich data that may be found in the unstructured, free text portion of the EHR. However, this data is not readily available for computation. Extracting this information requires natural language processing (NLP) methods. The NLP sub-task of information extraction is concerned with finding concepts in text and the relations between them (Jurafsky and Martin, 2014). Examples of information extraction are named entity recognition (e.g., finding the names of people, organizations, etc.) and relation extraction (e.g., determining whether the employment relation holds between a detected person like *Tim Cook* and a detected organization like *Apple*). A recent National NLP Clinical Challenge (n2c2)-hosted shared task annotated ADEs in clinical text in a style that is amenable to an information extraction approach. Specifically, annotations for things like drug names or drug attributes, including dosages, routes, and adverse events are entity-like spans, while the pairing of attributes and drugs are naturally represented as relations to be extracted. The benefit of framing the ADE task as an information extraction task is that decades of research in information extraction can be brought to bear on the task, before even considering the specifics of the domain or the task. In this work, we sought to evaluate a number of standard information extraction methods, including both standard clinical NLP tools and general domain methods, with the goals of setting strong baselines, learning how much performance is dependent on domain knowledge, and comparing classical machine learning to new deep learning approaches.

2 Methods

2.1 Data

This work describes methods for participating in the National NLP Clinical Challenge (n2c2) Track 2 shared task: Adverse Drug Events and Medication Extraction in EHRs. The data consists of 500 discharge summaries from the MIMIC (Medical Information Mart for Intensive Care) III database (Johnson et al., 2016). The n2c2 data was labeled with eight concept types: Drugs, Strengths, Dosages, Durations, Forms, Routes, Reasons, and ADEs. In addition, seven relations are labeled, between Drug mentions and the other seven concept types.

We participated in all three tracks of the shared task: entity recognition, relation classification given entities, and end-to-end relation extraction.

2.2 Methods

Our methods explore how well standard information extraction methods perform. One of our primary motivations is the prevalence of neural network methods in recent work, often motivated by their elimination of resource-intensive manual feature engineering, and thus judged superior to classical machine learning methods even if accuracy is similar. Unfortunately, in work comparing neural networks to classical methods, baseline classical machine learning systems can appear to be under-developed, while one is left wondering how much effort was actually required to engineer the network architecture and tune hyperparameters for the neural system. We used this dataset and task as an opportunity to invert that dynamic. We design a comparison that uses well-engineered features in a simple linear classifier without actually doing the engineering ourselves – we use features engineered over years of research in information extraction, and packaged in open source software such as Apache cTAKES (Savova et al., 2010) and ClearTK (Bethard et al., 2014). We then complete the comparison by comparing against off-the-shelf neural network tools and architectures for information extraction.

2.2.1 Entity extraction—To classify entities, we used a BIO tagger over tokens with a support vector machine classifier, with one classifier for each entity type. These classify every token in a document as the [B]eginning, [I]nside, or [O]utside of the entity type that classifier handles. We used Apache cTAKES (Savova et al., 2010) default pipeline to pre-process the data and the ClearTK (Bethard et al., 2014) machine learning API to extract features and train the models with Liblinear (Fan et al., 2008). The features used by the classifiers are standard features from information extraction, including:

- The previous token’s BIO classification decision
- Word identity and part of speech for the current token
- Word identities and parts of speech in the surrounding context
- Sub-word character type features
- Word semantic features

For token and token context features, we represent features in two forms, first as bags of words within a window and also with relative positional information. Character type features extract the character sequence in both the target token and the context tokens to model the fact that many attributes are typically numbers, or include numbers. This feature maps tokens to strings representing character types inside the token—for example, lower case characters map to *l*, upper case to *u*, punctuation to *p*, and digits to *d*, so the phrase *Mar 10, 2019* would map to *Ull ddp dddd*. Finally, we used semantic type information of the current token, as extracted with the cTAKES dictionary lookup module, to create a feature representing whether a token is a sign/symptom, disease/disorder, procedure, drug mention (as detected by cTAKES), or anatomical site, as well as the UMLS (Bodenreider, 2004) Type Unique Identifier (TUI).

During development, we manually partitioned the data so that we could empirically optimize the value of C in the linear SVM classifier on held out data. We tuned a single value of C that optimized the micro-F score on the held-out part of the training data. It may be possible to squeeze out slightly better performance by tuning C separately for each classifier, but the classifiers were pretty stable in the range we experimented with. We compare this system to an off-the-shelf neural network-based system called Flair (Akbik et al., 2018). This system is pre-trained using one billion words of text (Chelba et al., 2013) to learn a multi-layer Long Short-Term Memory (LSTM) (Hochreiter and Schmidhuber, 1997) network language model. Given the pre-trained network, this system passes in the tokens for an input sequence, and receives back the values at the deepest hidden layer at each index of the multi-layer LSTM, and this sequence of vectors is called contextual embeddings. Like regular word embeddings (Turian et al., 2010; Mikolov et al., 2013; Pennington et al., 2014), there is one vector per input token, but since they are extracted from the output layer of the pre-trained LSTM they are expected to contain more information about the surrounding sentence context.

To train an entity extractor in Flair, we again model the task as a BIO tagging task, but instead of using linguistic features we simply pass the contextual embeddings for each token to a standard LSTM tagger. This LSTM has a hidden state with 256 dimensions, and is optimized with Adam (Kingma and Ba, 2014). We train for 50 epochs, and the model that performs best on the held out validation set during training is used to prevent overfitting.

2.2.2 Relation Extraction—We built relation extraction classifiers relating each extracted attribute to drug mentions. Relation candidate pairs were extracted by comparing all drug mentions with the relevant attribute mention within the same paragraph, where paragraphs were defined to be delimited by two newline characters. We use the same feature set as previous work extracting relations to find anatomical site modifiers (Dligach et al., 2014). In the end-to-end version of the task, we considered drug mentions discovered both by the BIO tagger model and by cTAKES’s dictionary lookup module, which increased our recall. Any drug mentions discovered by cTAKES but not used in a relation were not output as Drug entities.

Finally, during preliminary work, we found that ADE and Reason entities actually behave more like relations, since they typically needed a nearby drug argument and some trigger words to be annotated. Therefore, instead of trying to detect ADE and Reason entities directly, we first train Drug-ADE and Drug-Reason relation classifiers, where the candidates for ADE and Reason arguments are all signs/symptoms and disease/disorders detected by cTAKES. If the relation classifier classifies a candidate pair as a Drug-ADE relation, we not only create the Drug-ADE relation but we create an ADE entity out of the non-Drug argument (and the Reason entity detector works the same way).

For relation extraction with the Flair neural model, we use a representation based on previous work on extracting temporal narrative container relations from sentences (Dligach et al., 2017). For each relation candidate consisting of a (Drug, Attribute) tuple, we insert xml-like start and stop tokens into the sentence around each of the candidate arguments indicating their position. For example, the sentence: *He does feel episodes of hypoglycemia if he does not eat following insulin* becomes: *He does feel episodes of <ADE> hypoglycemia </ADE> if he does not eat following <Drug> insulin </Drug>*. This augmented sentence representation is then passed into the pre-trained Flair bi-directional

LSTM sequence model, and the final states in each direction are concatenated into a feature vector. This feature vector is then passed through a linear layer to a softmax function over the output space to classify the relation.

For Track 3 (end-to-end relation extraction), the entity pairs found by the system in Track 1 were used to create candidate relations during training and testing. For Track 2, we used the gold standard entity pairs to create the candidate relations.

Results are scored with the scoring tool distributed by the organizers of the challenge. This tool reports scores for precision ($\frac{\#TruePositives}{\#Predictions}$), recall ($\frac{\#TruePositives}{\#GoldPositives}$), and F1 score ($\frac{2 * precision * recall}{precision + recall}$). For concepts, true positives can be strict (the system concept span must match a gold concept spans begin and end exactly) or lenient (a system concept span must overlap a gold concept span). For relations, a true positive is one where the gold set has a relation where both arguments match, and the relation category is the same. For both concepts and relations, we report micro-averaged results of the lenient evaluation, since that was the metric used to score the shared task.

3 Evaluation

The tables show results on the concept extraction (Table 1), relation classification (Table 2), and end-to-end relation extraction (Table 3). In the concept extraction task, the systems perform very similarly on average, with the SVM feature-engineered approach obtaining a micro-averaged F-score of 0.91 and the neural system scoring 0.90 (final row). By comparison, the best performing system at the n2c2 shared task scored 0.94 on the concept extraction task. The middle rows of Table 1 show the performance for different concept types. The two systems perform similarly across concept types, except that the SVM-based system performs much better on Route, while the neural system is much better at extracting Reason and Duration concepts.

For relation classification with gold standard concepts given as input (Table 2, top), the neural system is at least as good as the SVM-based system for every relation type, and the micro-averaged neural system is 0.93 compared to the 0.91 for the SVM-based system. Most improvement is seen in the Drug-Duration and Drug-Frequency categories. By comparison, the best performing system in the n2c2 challenge scored 0.96 on Track 2.

In the end-to-end relation extraction task (Table 3, bottom), the neural system is again two points better than the SVM in F1 score. The SVM performs better on Drug-Route and Drug-ADE, while the neural system performs better in Drug-Duration and Drug-Reason. The best performing system in the n2c2 challenge scored 0.89 on Track 3.

4 Conclusion

Despite minimal engineering effort, neural systems pre-trained on non-medical text obtain similar performance to feature engineered systems with features specific to clinical text. This is perhaps somewhat surprising, and provides some evidence that standard neural architectures for sequence tagging and relation extraction tasks are already quite mature.

One caveat to these results is that, while our feature-based approach used standard feature sets with history of success in the literature, one could argue that to mirror the tuning that is done with neural networks we could have done more extensive tuning of feature hyperparameters, by, for example, testing configurations where certain groups of features are turned on or off.

While the performance of the neural system in this work is impressive, one might expect them to perform even better if they could be pre-trained on clinical text. Future work will investigate language model pre-training in Flair and other neural architectures on large amounts of clinical data from electronic health record systems. The code developed to participate in the n2c2 challenge and run these experiments is available open source.[1]

Acknowledgements

Research reported in this publication was supported by the National Library Of Medicine of the National Institutes of Health under Award Number R01LM012918. The content is solely the responsibility of the authors and does not necessarily represent the official views of the National Institutes of Health.

References

Akbik Alan, Blythe Duncan, and Vollgraf Roland. 2018. Contextual string embeddings for sequence labeling. In Proceedings of the 27th International Conference on Computational Linguistics, pages 1638–1649.

Bethard S, Ogren PV, and Becker L. 2014. ClearTK 2.0: Design Patterns for Machine Learning in UIMA. LREC.

Olivier Bodenreider. 2004. Theunifiedmedical language system (UMLS): integrating biomedical terminology. Nucleic acids research, 32(suppl 1):D267–D270. [PubMed: 14681409]

Chelba Ciprian, Mikolov Tomas, Schuster Mike, Ge Qi, Brants Thorsten, Koehn Phillipp, and Robinson Tony. 2013. One Billion Word Benchmark for Measuring Progress in Statistical Language Modeling. arXiv:1312.3005 [cs]. ArXiv: 1312.3005.

Dligach Dmitriy, Bethard Steven, Becker Lee, Miller Timothy, and Savova Guergana K. 2014. Discovering body site and severity modifiers in clinical texts. Journal of the American Medical Informatics Association: JAMIA, 21(3):448–54. [PubMed: 24091648]

Dligach Dmitriy, Miller Tim, Lin Chen, Bethard Steven, and Savova Guergana. 2017. Neural Temporal Relation Extraction. In Proceedings of the 15th Annual Meeting of the European Association for Computational Linguistics, pages 746–751, Valencia, Spain.

Fan Rong-En, Chang Kai-Wei, Hsieh Cho-Jui, Wang Xiang-Rui, and Lin Chih-Jen. 2008. LIBLINEAR: A Library for Large Linear Classification. Journal of Machine Learning Research, 9:1871–1874.

Hochreiter S and Schmidhuber J. 1997. Long short-term memory. Neural computation.

Johnson Alistair E. W., Pollard Tom J., Shen Lu, Lehman Liwei H., Feng Mengling, Ghas-semi Mohammad, Moody Benjamin, Szolovits Peter, Celi Leo Anthony, and Mark Roger G.. 2016. MIMIC-III, a freely accessible critical care database. Scientific Data, 3:160035. [PubMed: 27219127]

Jurafsky Dan and Martin James H. 2014. Speech and language processing, volume 3. Pearson London.

Kingma Diederik P and Ba Jimmy Lei. 2014. Adam: Amethod for stochastic optimization. In Proc. 3rd Int. Conf. Learn. Representations.

Maxey Dawn M, Dunbar Ivy D, Ogawa Michelle T, and Feinstein Jeffrey A 2013. Food and Drug Administration (FDA) postmarket reported side effects and adverse events associated with

[1] https://github.com/tmills/ctakes-ade

pulmonary hypertension therapy in pediatric patients. Pediatric cardiology, 34(7):1628–1636. [PubMed: 23532466]

McMahon Ann W and Dal Pan Gerald. 2018. Assessing Drug Safety in ChildrenThe Role of Real-World Data. The New England journal of medicine, 378(23):2155. [PubMed: 29874532]

Mikolov Tomas, Chen Kai, Corrado Greg, and Dean Jeffrey. 2013. Distributed Representations of Words and Phrases and their Compositionality arXiv: 1310.4546v1 [cs. CL] 16 Oct 2013. arXiv preprint arXiv:1310.4546, pages 1–9.

Nadkarni Prakash M. 2010. Drug safety surveillance using de-identified EMR and claims data: issues and challenges. Journal of the American Medical Informatics Association, 17(6):671–674. [PubMed: 20962129]

Pennington Jeffrey, Socher Richard, and Manning Christopher D. 2014. GloVe: Global Vectors for Word Representation. In Proceedings of EMNLP.

Platt Richard, Brown Jeffrey S., Robb Melissa, Mark McClellan Robert Ball, Nguyen Michael D., and Sherman Rachel E.. 2018. The FDA Sentinel Initiative - An Evolving National Resource. The New England journal of medicine, 379(22):2091–2093. [PubMed: 30485777]

Sanson-Fisher Robert William, Bonevski Billie, Green Lawrence W, and DEste Cate. 2007. Limitations of the randomized controlled trial in evaluating population-based health interventions. American journal of preventive medicine, 33(2):155–161. [PubMed: 17673104]

Savova GK, Masanz JJ, and Ogren PV. 2010. Mayo clinical Text Analysis and Knowledge Extraction System (cTAKES): architecture, component evaluation and applications. Journal of the American Medical Informatics Association.

Sipahi Ilke, Celik Seden, and Tozun Nurdan. 2014. A comparison of results of the US food and drug administrations mini-sentinel program with randomized clinical trials: the case of gastrointestinal tract bleeding with dabigatran. JAMA internal medicine, 174(1):150–151. [PubMed: 24247291]

Sultana Janet, Cutroneo Paola, and Trifir Gianluca. 2013. Clinical and economic burden of adverse drug reactions. Journal of pharmacology & pharmacotherapeutics, 4(Suppl1):S73. [PubMed: 24347988]

Turian Joseph, Ratinov Lev, and Bengio Yoshua. 2010. Word representations: A simple and general method for semi-supervised learning. In Proc. of ACL 2010.

Table 1:

Results of entity recognition experiments with SVM vs. Neural systems.

Track 1	**Precision**		**Recall**		**F1**	
	SVM	**Neural**	**SVM**	**Neural**	**SVM**	**Neural**
Drug	0.96	0.96	0.92	0.90	0.94	0.93
Strength	0.98	0.97	0.95	0.97	0.97	0.97
Duration	0.82	0.91	0.63	0.65	0.71	0.76
Route	0.96	0.95	0.91	0.83	0.94	0.89
Form	0.97	0.93	0.92	0.95	0.95	0.94
ADE	0.66	0.58	0.20	0.18	0.31	0.27
Dosage	0.94	0.92	0.88	0.92	0.91	0.92
Reason	0.78	0.71	0.38	0.56	0.51	0.63
Frequency	0.98	0.98	0.93	0.95	0.95	0.96
Average	0.95	0.94	0.86	0.87	0.91	0.90

Table 2:

Results of relation classification experiments (gold standard entity arguments) with SVM vs. Neural systems.

Track 2	Precision		Recall		F1	
	SVM	Neural	SVM	Neural	SVM	Neural
Drug-Strength	0.93	0.99	0.96	0.98	0.94	0.98
Drug-Duration	0.81	0.93	0.83	0.86	0.82	0.89
Drug-Route	0.93	0.97	0.95	0.94	0.94	0.96
Drug-Form	0.96	0.99	0.97	0.95	0.97	0.97
Drug-ADE	0.75	0.77	0.78	0.80	0.76	0.79
Drug-Dosage	0.95	0.98	0.96	0.93	0.95	0.95
Drug-Reason	0.74	0.91	0.76	0.65	0.75	0.76
Drug-Frequency	0.90	0.98	0.92	0.94	0.91	0.96
Average	0.90	0.97	0.92	0.90	0.91	0.93

Table 3:

Results of relation extraction experiments (system-generated entity arguments) with SVM vs. Neural systems.

Track 3	Precision		Recall		F1	
	SVM	Neural	SVM	Neural	SVM	Neural
Drug-Strength	0.92	0.96	0.91	0.94	0.91	0.95
Drug-Duration	0.73	0.83	0.51	0.57	0.60	0.67
Drug-Route	0.92	0.94	0.86	0.77	0.89	0.85
Drug-Form	0.95	0.94	0.89	0.89	0.92	0.91
Drug-ADE	0.60	0.50	0.18	0.15	0.28	0.23
Drug-Dosage	0.92	0.92	0.84	0.84	0.88	0.88
Drug-Reason	0.66	0.65	0.31	0.46	0.42	0.54
Drug-Freq	0.90	0.96	0.86	0.87	0.88	0.92
Average	0.90	0.90	0.76	0.78	0.82	0.84