Proceedings of Machine Learning Research 90:31–38, 2018 Medication and Adverse Drug Event Detection Workshop

Knowledge-based Feature Engineering for Detecting Medication and Adverse Drug Events from Electronic Health Records

Duy-Hoa Ngo HOA.NGO@CSIRO.AU
Alejandro Metke-Jimenez ALEJANDRO.METKE@CSIRO.AU
Anthony Nguyen ANTHONY.NGUYEN@CSIRO.AU
The Australian eHealth Research Centre, CSIRO, Brisbane, Australia

Editor: Feifan Liu, Abhyuday Jagannatha, Hong Yu

Abstract

This paper presents a Conditional Random Field (CRF) learning model integrated with a series of knowledge-based feature engineering functions for detecting medication and adverse drug events from electronic health records (EHRs). Our experimental evaluation shows high performance in terms of the F-score measure (83.4% and 92.5% respectively for strict and relax modes) in the detection of medication and clinical finding named entities. It also shows promising results (overall 78.8% F-score) in the recognition of detailed properties of medication use as well as different types of clinical findings mentioned in electronic health records.

Keywords: CRF, SNOMED CT, ADE, Biomedical NER, Dictionary-based, Rule-based, Knowledge-based features

1. Introduction

These days, electronic health record (EHR) systems have been largely adopted in hospitals and medical practices. This vast information from EHR resources presents a great opportunity for automatically extracting medical insights from these patient cases through the use of text mining. One major focus of text mining research on EHRs is Biomedical Named Entity Recognition (Bio-NER), a crucial initial step in information extraction that is aimed at identifying chunks of text that refer to specific entities of interest in the biomedical domain.

Various approaches have been developed for Bio-NER, which can be categorised into three main approaches, namely, rule-based, dictionary matching based and machine learning based methods. Rule-based methods usually rely on regularities in naming conventions, and consist of a set of rules based on regular expressions. Though these rule-based methods seemed promising initially, they failed to perform well on larger datasets. Moreover, it is impossible for these systems to detect patterns that haven't been defined before. On the other hand, dictionary-based methods rely on terminology resources, containing large collections of names, serving as entries for a specific entity class. Matching entries exactly against text is simple and precise, but it results in low recall, especially due to terminology heterogeneity, such as language variation, use of shorthands to refer to the same thing, acronyms and abbreviations. In contrast to these two approaches, machine learning based methods learn patterns from existing training resources in order to detect named entities in

© 2018 D.-H. Ngo, A. Metke-Jimenez & A. Nguyen.

unseen data. In this approach, researchers do not have to compose complex rules manually. In addition, machine learning methods can also learn to identify new named entities not found in standard dictionaries. There are many machine learning methods that have been used for Bio-NER. For example, Conditional Random Fields (CRFs) have been used to extract medications and side effects from social media Karimi et al. (2015b,a); Metke-Jimenez and Karimi (2016).

In this paper, we present a hybrid method combining the benefits from all three aforementioned approaches to do detection of medications and adverse drug events (ADE) from EHRs. Specifically, we build a CRF-based machine learning model to detect biomedical named entities such as medications, dosages, routes, frequencies, durations, indications, ADEs, other signs or symptoms and its severity mentioned in sentences contained in the EHR. In order to train the model, we rely on existing knowledge resources including rules and dictionaries to extract representative features of observed data. The proposed approach was trained and evaluated on the MADE 1.0 dataset. Our experiments show promising results and also demonstrated that employing knowledge-based feature engineering is effective in improving system performance.

2. Methods

In this section, we first introduce our machine learning model used for Bio-NER. Next, we present an overview of our pre-processing pipeline, which transforms each EHR record from raw textual data into machine learning data used in the training and prediction phases. We also discuss in detail the knowledge-based feature engineering methods that extract representative features for every token contained in the EHRs.

2.1. Learning Model

The machine learning model applied in our approach is a Linear-chain (first-order Markov) Conditional Random Field (CRF) model Elkan (2008); Sutton and McCallum (2012). Specifically, let $\bar{x}$ be a sequence of tokens and let $\bar{y}$ be a corresponding sequence of labels belonging to a predefined finite label set. A linear-chain CRF model posits that the probability of any particular $\bar{y}$, given the example $\bar{x}$ is:

$$p(\bar{y}|\bar{x};w) = \frac{1}{Z(\bar{x},w)} exp \sum_{j=1}^{J} w_j F_j(\bar{x},\bar{y})$$

Here, the denominator is a normalising factor that is constant given $\bar{x}$ and w; and for $j = 1$ to $j = J$, each feature function F_j is actually a sum along the output sequence, for $i = 1$ to $i = n$ where n is the length of $\bar{y}$:

$$F_j(\bar{x},\bar{y}) = \sum_{i=1}^{n} f_j(y_{i-1}, y_i, \bar{x}, i)$$

Training a CRF model means finding the parameter vector w that gives the best possible prediction:

$$\hat{y} = argmax_{\bar{y}} p(\bar{y}|\bar{x};w)$$

In our approach, the linear-chain CRF model was implemented on top of the CRFsuite library Okazaki (2007). It uses the Limited-memory BFGS (L-BFGS) algorithm with L2 regularisation to train the CRF model. The trained model takes as input a training sentence and produces a corresponding sequence of predicted BIO-formatted labels. The detail of the BIO format will be discussed in section 2.4.

2.2. Pre-processing Pipeline

The pre-processing pipeline is summarised in Figure 1. In our approach, a machine learning instance is a collection of representative features extracted from a single sentence. Therefore, a sentence segmentation module is needed to take the full text of an EHR as input and produce a list of sentences separated by full stops or new lines.

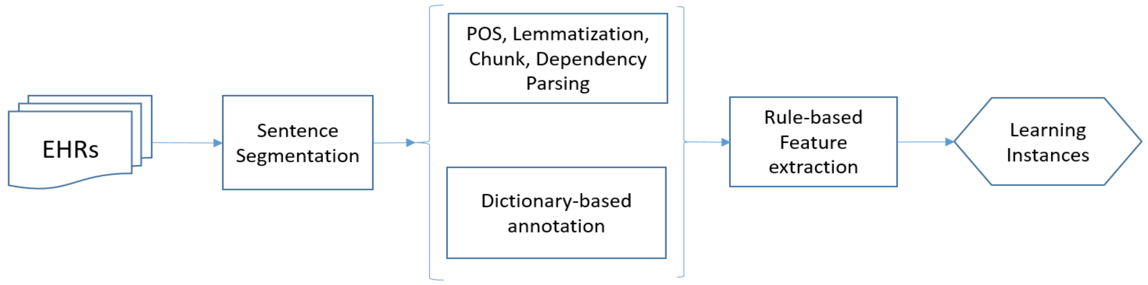

EHRs
Sentence Segmentation
POS, Lemmatization, Chunk, Dependency Parsing
Dictionary-based annotation
Rule-based Feature extraction
Learning Instances

Figure 1: Pre-processing pipeline

Each sentence is then processed by the GDep parser Sagae and Tsujii (2007) for tokenisation and linguistic processing such as lemmatisation, part-of-speech (POS) tagging, chunking and dependency parsing. In addition, a collection of dictionary-based matchers were used to assign concept labels to segments of tokens in a given sentence. This information was combined with predefined rules to produce representative features for every token. Finally, a machine learning instance is constructed by combining the extracted features from consecutive sequences of tokens in a given input sentence.

2.3. Dictionary-based Annotation

The selection of appropriate terminology resources to annotate entities from text plays an important role in our approach. To focus on Bio-NER in ADE detection, the following entity classes, namely, drug, dosage, route, duration, frequency and clinical findings were taken into consideration.

First, a drug dictionary was extracted from the DrugBank database Wishart et al. (2008), in which each entry consists of a mapping from a drug ID to its names, synonyms, parents and categories. Additionally, we supplement new entities extracted from descendants of the SNOMED CT [1] concepts 373873005 | *Pharmaceutical/biologic product* and 105590001 | *Substance*. Our assumption was that almost all prescribed medications and its ingredients could be found in these resources.

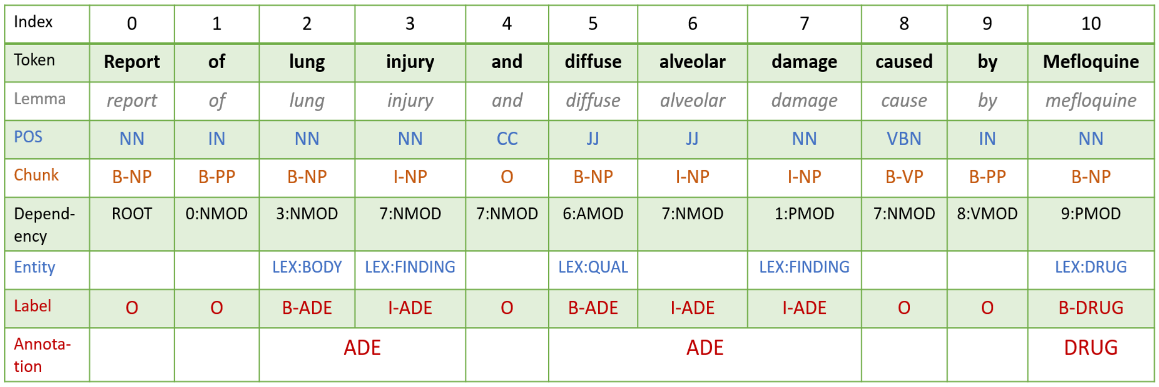
1. SNOMED CT International version - January 31 2018 Release

Next, our hypothesis about clinical findings mentioned in EHRs was that most of them were compound terms written in the form of a combination of a root disease or symptom category with a short descriptive modifier including anatomical location (e.g. *Breast Cancer*), qualifier value (e.g., *Chronic Pain*), etc. Therefore, we build a dictionary of core diseases/symptoms by extracting descendants of the SNOMED CT concepts 404684003 | *Clinical finding* and 118956008 | *Morphologic abnormality*. The anatomical location dictionary was extracted from the descendants of concept 123037004 | *Body structure*, and a qualifier value dictionary was extracted from the descendants of concept 362981000 | *Qualifier value*.

Similarly, we build dictionaries for handling the drug's dosage, duration and frequency by extracting the descendants of concepts 258666001 | *Unit*, 421967003 | *Drug dose form* and 7389001 | *Timeframe*.

2.4. Feature Extraction

The intuition of feature extraction is illustrated via a small example depicted in Figure 2. The first, second and third rows show indexes, tokens and corresponding lemmas in the sample sentence. The fourth and fifth rows describe part-of-speech and corresponding chunks for every token. The sixth row presents the chunk dependency relationship between tokens as a result of using the dependency parser GDep. Each cell in the seventh row contains types of dictionary entities that exactly matched to a chunk of tokens in the given sentence. In this example, token *lung* was found as a *Body structure* entity; *injury* and *damage* were found as *Clinical Findings* entities; *diffuse* was found as a *Qualifier value* entity, and finally, *mefloquine* was found as a *Drug* entity.



Index	0	1	2	3	4	5	6	7	8	9	10
Token	**Report**	**of**	**lung**	**injury**	**and**	**diffuse**	**alveolar**	**damage**	**caused**	**by**	**Mefloquine**
Lemma	*report*	*of*	*lung*	*injury*	*and*	*diffuse*	*alveolar*	*damage*	*cause*	*by*	*mefloquine*
POS	NN	IN	NN	NN	CC	JJ	JJ	NN	VBN	IN	NN
Chunk	B-NP	B-PP	B-NP	I-NP	O	B-NP	I-NP	I-NP	B-VP	B-PP	B-NP
Dependency	ROOT	0:NMOD	3:NMOD	7:NMOD	7:NMOD	6:AMOD	7:NMOD	1:PMOD	7:NMOD	8:VMOD	9:PMOD
Entity			LEX:BODY	LEX:FINDING		LEX:QUAL		LEX:FINDING			LEX:DRUG
Label	O	O	B-ADE	I-ADE	O	B-ADE	I-ADE	I-ADE	O	O	B-DRUG
Annotation			ADE			ADE					DRUG

Figure 2: Feature extraction example from an EHR extract.

We categorised feature engineering functions into two main categories: *individual* and *contextual* features. The individual features of each token show the properties of the token itself that consist of its text string, lemma, POS, chunking, word shape and entity types discovered by dictionary-based matchers. For example, token *injury* has features such as `LEMMA=injury`, `POS=NN`, `CHUNK=I-NP` and `LEX=FINDINGS`.

The contextual features show the relationship properties of a token with its neighbours. Five types of feature engineering functions have been implemented to extract contextual features for every token. Table 1 summarises all these feature engineering functions.

Firstly, a *window function* returns the tokens surrounding the given token. This kind of feature function was inspired from the widely used n-grams feature engineering method in text mining, in which, for each token inside the window, the function can use an individual feature such as lemma, POS or entity types. For example, within a window [-1,1] of token *alveolar*, a window function exploiting the token's lemma returns feature: *LEMMA=diffuse@-1_&_LEMMA=alveolar_&_LEMMA=damage@1*.

Secondly, instead of taking all consecutive tokens inside a predefined window of a given token, an *offset conjunction function* was only interested in tokens located at some offset distance to the given token. Similarly to the *window functions*, an *offset conjunction function* can extract either lemma, POS or entity type of each token of interest. For example, given offsets (-2,-1), a LEMMA-LEMMA conjunction function of token *lung* produces the feature: *LEMMA=report@-2_&_LEMMA=of@-1*. The intuition is that ADEs, indications and other disease signs in an EHR may be distinguished by some *signal words* appearing before or after clinical findings terms.

Thirdly, a *mixture feature conjunction function* was an extension of the *offset conjunction function*. This kind of function can extract different types of features for different tokens in the context. For example, given offsets (-2, -1, 0), a POS-POS-LEX function of token *damage* will produce feature *POS=JJ@-2_&_POS=JJ@-1_&_LEX=FINDING*. The intuition was that if the current token has been annotated as a disease name, and its previous token was an adjective, then the chunk of the two tokens will likely be a compound disease name.

Fourthly, contextual information such as negation, temporality and experiencer of clinical findings entities were also used as contextual features. For example in Figure 2, the terms *injury* and *damage* have been found in the clinical findings dictionary. After running rule-based detection tools Harkema et al. (2009) of negation, temporality and experiencer for the given sentence, we assigned features *NEG=FALSE*, *TEMPORALITY=RECENT* and *EXPERIENCER=PATIENT* to *injury* and *damage*.

Finally, a *dependency function* returns a combination of token string values that were dependent on the given token according to the dependency structure after parsing the sentence. In our approach, dependency structure were exploited in two ways. The first one was inspired from Gimli system Campos et al. (2013), which was a high-performance biomedical name recogniser. Similarly to Gimli's system, features were extracted based on the following dependent types: SUB, OBJ, NMOD, AMOD, PMOD and VMOD De Marneffe and Manning (2008). Each function could extract either lemma or entity type from the dependent tokens. For example from Figure 2, the following dependency features would be assigned to the term *damage*: *NMOD-BY=alveolar*, *NMOD-BY=and*, *NMOD-BY=injury*. On the other hand, in the second way, paths from the tree dependency structure were exploited to generate token features. For example, within a radius size of 2, token *damage* had the following path features: *PREV=report of*, *POST=caused by*, *POST=lung injury*. Similar to the *offset conjunction function* mentioned above, we expected that this kind of features would contain *signal words* to separate ADE, Indication and other disease signs. In this example, *PREV=report of* and *POST=caused by* are those signal words.

After applying feature engineering functions to every token, the tokens in the training set were associated with their target labels using the BIO (beginning-inside-other) format. For example, the term *lung injury* was annotated in the gold standard as an *ADE*, so the

Function type	Feature Combination	Location Parameters
Window	LEMMA	[-1,1], [-3,3]
Window	LEX	[-3,3]
Offset Conjunction	LEMMA-LEMMA	(-2,-1), (-1,0), (0,1), (-1,1)
Offset Mix features	POS-LEMMA	(-1,0), (0,1)
Offset Mix features	POS-POS-LEMMA	(-2,-1,0), (-1,0,1),(0,1,2)
Offset Mix features	POS-POS-LEX	(-2,-1,0), (-1,0,1),(0,1,2)
Offset Mix features	LEMMA-LEX	(-1,0), (0,1)
Offset Mix features	LEX-LEMMA	(-1,0), (0,1)
Dependency	LEMMA	dependency structure
Dependency	LEX	dependency structure
Context	Negation, Temporality, Experiencer	-

Table 1: Contextual feature engineering functions used in our approach

tokens *lung* and *injury* were assigned `B-ADE` and `I-ADE`, which refer to a beginning and inside of an ADE annotation respectively. Other tokens not associated with any predefined annotations will be assigned the label `O` meaning outside of any annotation. A training sentence is a sequence of training tokens. In Figure 2, the eighth row shows the resulting labels for the sentence in the second row.

3. Evaluation

We evaluated our approach on the MADE 1.0 dataset[2], in which, training data contained 876 EHR records, and testing data included 271 additional EHR records. The challenges in detecting Bio-NER in the MADE dataset came from its noisy data, its incomplete sentences, phrases and irregular use of language as well as the use of abundant abbreviations, rich medical jargons, and variations of named entities.

In general, the following biomedical entities were of interest in the recognition process: i) medication information including drug name, dosage, route, frequency, duration; ii) clinical findings and their severity information. Three types of clinical findings detected were: adverse drug event - ADE, Indications and other signs and symptoms - SSLIF. Here, SSLIF refers to medical sign, symptom or disease names that were neither being actively treated (i.e., Indication) nor were they an adverse side effect (i.e., ADE) of a drug.

	Drug	Dose	Route	Duration	Severity	Frequency	Indication	ADE	SSLIF	Overall
Recall	0.811	0.789	0.828	0.684	0.498	0.754	0.404	0.381	0.801	0.746
Precision	0.902	0.844	0.947	0.843	0.816	0.867	0.682	0.845	0.807	0.835
F-score	0.854	0.815	0.883	0.755	0.619	0.807	0.507	0.525	0.804	0.788

Table 2: Strict evaluational results on discovery of all Bio-medical name entities

Table 2 presents the evaluation results from all entities from the MADE 1.0 challenge. The overall Recall, Precision and F-score values were 74.6%, 83.5% and 78.8% respectively.

2. https://bio-nlp.org/index.php/announcements/39-nlp-challenges

It was noticeable that our approach achieved high *precision* and *recall* scores in detecting medication and its attributes. This behaviour was exactly what we expected when using large drug terminology resources such as Drugbank and comprehensive terminology resources about measurement units for dose, route, duration and frequency in SNOMED CT.

On the other hand, an initial error analysis of the results revealed that our model had some challenges in distinguishing between *ADEs*, *Indications* and *SSLIFs*. For example in the test case `45bae618`, our model classified *Hodgkin disease* as an *SSLIF*, but in fact, it was annotated as an *Indication*. Similarly, *Nausea*, *vomiting*, *hypotension*, *chills*, *fever*, *dyspnea*, *bronchospasm* and *tachycardia* were classified as *SSLIF*, but they were indeed *ADE*. This kind of misclassification occurred quite frequently in test cases. One of the main reasons behind this limitation was that our features for every token were extracted using a small window around the tokens' position in the sentence or in dependency tree paths, whereas, the information used to detect ADEs and Indications were usually found using larger window sizes.

	Strict			Relax		
	Precision	Recall	F-score	Precision	Recall	F-score
Drug	0.902	0.811	0.854	0.977	0.885	0.918
Finding	0.859	0.799	0.828	0.964	0.893	0.927
Average	0.870	0.801	0.834	0.967	0.886	0.925

Table 3: Evaluation results on discovery of Drug and Disease named entities

Next, we were highly interested in the detection of drug and clinical finding named entities because they were considered important in any clinical context. The results in Table 3 show that the performance of our model was quite promising. In the relax evaluation mode where a predicted term was found overlapping a gold standard term, the F-score values were around 92%, whereas for the strict mode, i.e., exact matching, the F-score values were around 83%. After analysing the false positive (i.e., wrong classified entities) and false negative (i.e., undetected entities) we found other weaknesses in our learning model. For example in the test case `ed7798df`, after scanning the whole text, our system detected that tokens *Savella*, *Cymbalta*, *Paroxetine*, *Pilocarpine* and *Restasis* were found in our drug-based dictionary, but, at the end of classification process, none of them were detected as a *Drug* named entity. This matter also happened the same with disease named entities. We plan for further investigation on these weaknesses in our future work. However, these results show using both dictionary-based and rule-based feature engineering in this task was a promising approach.

4. Conclusion

We presented a learning model based on CRFs with a combination of dictionary-based and rule-based feature engineering. Our trained model not only discovered drug and disease names with high accuracy in terms of F-score, but also shows promising performance on the discovery of detailed medication usage properties accompanied by different types of clinical findings mentioned in EHRs.

References

David Campos, Srgio Matos, and Jos Lus Oliveira. Gimli: open source and high-performance biomedical name recognition. *BMC Bioinformatics*, page 54, 2013.

Marie-Catherine De Marneffe and Christopher D. Manning. Stanford typed dependencies manual. 2008. URL http://nlp.stanford.edu/software/dependencies_manual.pdf.

Charles Elkan. Log-linear models and conditional random fields. 2008. URL http://cseweb.ucsd.edu/~elkan/250Bwinter2012/loglinearCRFs.pdf.

Henk Harkema, John N. Dowling, Tyler Thornblade, and Wendy W. Chapman. Context: An algorithm for determining negation, experiencer, and temporal status from clinical reports. *J. of Biomedical Informatics*, 42(5):839–851, October 2009. ISSN 1532-0464.

Sarvnaz Karimi, Alejandro Metke-Jimenez, Madonna Kemp, and Chen Wang. Cadec: A corpus of adverse drug event annotations. *Journal of Biomedical Informatics*, 55:73 – 81, 2015a. ISSN 1532-0464.

Sarvnaz Karimi, Alejandro Metke-Jimenez, and Anthony Nguyen. Cademiner: A system for mining consumer reports on adverse drug side effects. In *Proceedings of the Eighth Workshop on Exploiting Semantic Annotations in Information Retrieval*, ESAIR '15, pages 47–50, New York, NY, USA, 2015b. ACM. ISBN 978-1-4503-3790-8.

Alejandro Metke-Jimenez and Sarvnaz Karimi. Concept identification and normalisation for adverse drug event discovery in medical forums. In *BMDID@ISWC*, 2016.

Naoaki Okazaki. Crfsuite: a fast implementation of conditional random fields (crfs), 2007. URL http://www.chokkan.org/software/crfsuite/.

K. Sagae and J. Tsujii. Dependency parsing and domain adaptation with lr models and parser ensembles. In *Proceedings of the CoNLL 2007 Shared Task in the Joint Conferences on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL'07 shared task)*, pages 1044–1050, 2007. Prague, Czech Republic.

Charles Sutton and Andrew McCallum. An introduction to conditional random fields. *Found. Trends Mach. Learn.*, 4(4):267–373, 2012. ISSN 1935-8237.

David S. Wishart, Craig Knox, Anchi Guo, Dean Cheng, Savita Shrivastava, Dan Tzur, Bijaya Gautam, and Murtaza Hassanali. Drugbank: a knowledgebase for drugs, drug actions and drug targets. *Nucleic Acids Research*, 36:901–906, 2008.