Extracting Positive Mentions of Adverse Drug Reactions from Product Labels using a Machine Learning Centric Approach

Rave Harpaz, William DuMouchel
Oracle Health Sciences

1. Introduction

Product labels are an authoritative source of information about the risks, benefits, and pharmacological properties of drugs. As such, extracting the information stored in product labels, and making it available in the form of computationally accessible knowledge bases would benefit several applications in the area of drug safety surveillance and assessment[1-4]. For example, during post-marketing safety assessments is important to determine whether an investigated adverse drug reaction (ADR) is already labeled.

In the US electronic versions of product labels are called Structured Product Labels (SPLs). The primary challenge involved in utilizing SPLs for drug safety (and other) applications is that most of their content is captured as free-text. Consequently, unlocking their value necessitates the development of NLP techniques tailored to the application of SPL processing.

In parallel to existing initiatives and ongoing methodological development[5-8], the 2017 TAC ADR track was created to assess the utility of NLP techniques for extracting ADRs from product labels.

The 2017 TAC ADR track is defined by four different tasks[9]. Task 1 involves named entity recognition of *Adverse Reaction* mentions and five related entities. Task 2 involves the extraction of three types of relations (*Negated*, *Hypothetical*, and *Effect*) between *Adverse Reaction* named entities and the other entities. Task 3 involves the verbatim identification of all *positive* mentions of adverse reactions within a product label*.* A *positive Adverse Reaction* is defined as an adverse reaction that is not negated and is not related by a *Hypothetical* relation to a *Drug Class* or *Animal*. Finally, Task 4 involves MedDRA coding of the positive adverse reactions mentions identified in Task 3.

TAC ADR participants were provided with over 2,300 product labels as text documents in XML format. Of these, 101 drug labels formed the formal training set, and contained the gold standard annotations for each of the four tasks. Participants were then required to run their developed systems on the remaining 2,200 thousand labels, and were evaluated on a test/held-out set of 99 labels, which was not known in advance.

TAC ADR participants were permitted to work on any of the four task. We chose to address Task 3—the *identification of positive mentions of adverse reactions*.

Due to lack of annotated training data, the majority of previously developed NLP approaches to process product labels are rule-based. In light of the TAC ADR initiative, whereby carefully annotated training data has been made available, we sought to explore the use of machine learning (ML) approaches to accomplish Task 3.

The system we developed can be generally described as a three step process. (1) an ensemble of Conditional Random Fields (CRFs) is used to perform named entity recognition (NER) of adverse reactions (AR). (2) a rule-based approach is used to identify disjoint (non-consecutive word tokens)

AR mentions. (3) a rule-based approach is used to eliminate negated AR mentions identified in the previous two steps.

2. *Methods*

2.1 *low-level text processing operations*

To prepare the data for the NER step we applied sentence segmentation, word tokenization, and part of speech tagging. Using models developed for medical data to perform these three operations would have been preferable. However, because such models were not available to us we resorted to using Python's NLTK (version 3.2.4) library, to which we applied slight modifications in order to accommodate certain traits of the product labels data.

2.2 *Feature extraction*

Our CRF models for the NER step employed a set of commonly applied features, and two sets of application-specific features. The former included: lower-cased versions of the words being modeled, contextual features (e.g., words before and after the word being modeled), their POS, and word shape/orthographic features such as capitalization, suffix, and numeric patterns.

The first set of application-specific features included an indicator variable that encodes the sentence type being processed (e.g., list, table, header, bullet). Lists were identified by the presence of certain trigger terms (e.g., 'include ...', 'including …', 'commonly occurring adverse reactions …') and punctuation (e.g., ':', ','), and the ratio of tokens to commas in a sentence. Tables were identified by analyzing the ratio of token to spaces-between-tokens in a sentence. Headers and bullets were identified based on the appearance of special charters (e.g. '*') and numeric string patterns (e.g. '3.1') at the beginning of a sentence.

Our second set of applications-specific features included two indictor variables that encode domain knowledge. Specifically, these two indictor variables encode the presence or absence of an exact string match between words being modeled and two computationally generated vocabularies of medical disorders.

The first vocabulary was created by applying MetaMap (Ver 2014)[10] to the full set of 2,300 product labels in order to identify concepts associated with medical disorders. The second vocabulary was created by developing an independent annotator, similar to MetaMap, that scans through text to identify terms corresponding to medical disorders. In the following we refer to this annotator as the *ADR annotator*. The terminologies underlying this annotator were compiled from the UMLS (2016aa) metathesaurus[11].

To each of these two vocabularies we then added additional terms that were derived by three types of fuzzy string matching: standard, subset, and token order fuzzy matching. The matching was done between the original two vocabularies and all n-grams (n=1-5) appearing in the full set (2,300) of product labels. The matching method used the Levenshtein edit distance with cutoff ratios of 0.85 and above, depending on the number of tokens matched and the matching type.

2.3 *Adverse reaction named entity recognition using conditional random fields*

We applied CRFs for only the identification of the *Adverse Reaction* named entity. The other named entities required to accomplish Task 3 were indirectly identified using rule-based approaches. The direct identification of these other named entities was not necessary to accomplish Task 3.

Our CRF models were trained using the python sklearn-crfsuite (Ver. 0.3.6) library, which provides a API to the independent *CRFsuite*[12] software. Prior to training, the data was transformed to the standard IOB representation.

In addition to standard single model CRFs, we implemented and experimented with an ensemble of CRFs. The ensemble was created by iteratively training a base CRF model on resampled training data that focuses on prediction errors, similar to Boosting approaches. The results of the ensemble were combined by a voting scheme. To our knowledge this form of a CRF model ensemble has not been published.

The ensemble provides two main advantages over single model. Like other ensemble approaches, it is supposed to reduce prediction variance thereby improving generalization performance. By strategically resampling the data to focus on certain prediction errors (misclassified sentences), and by combing the results in a specific manner the ensemble allowed us finer control over the tradeoff between precision and recall. The main reason we utilized this ensemble was to increase recall. This was because the recall of our base CRF model was relatively low, and because it is easier to address precision than recall via post-processing.

The base CRF model was trained using a variant of L-BFGS optimization algorithm for fitting L_1-regularized models. The parameters that were varied for the base CRF model were the L_1 and L_2 regularization parameters. The ensemble requires two additional parameters: K - the number of models in the ensemble, and p_{miss} - the proportion of misclassified sentences to sample. The training of the ensemble is outlined using the pseudo code below

```
Input: $X$ – training sentences, $Y$ – training sentence labels
Initialize: $C^+ = C^- = (X, Y)$, ensemble $M^* = \emptyset$
For $k$ = 1 to $K$
    $X^+, Y^+$ = sample with replacement $(1 - p_{miss}) \times |X|$ sentences and their labels from $C^+$
    $X^-, Y^-$ = sample with replacement $p_{miss} \times |X|$ sentences and their labels from $C^-$
    $M_k$ = train base-CRF model on $(X^+ \cup X^-, Y^+ \cup Y^-)$
    apply model $M_k$ to $X$
    $e_k$ = error of $M_k$ applied to $X$
    $M^* = M^* \cup (M_k, e_k)$
    $C^+$ = set of sentences and their labels that have been misclassified by $M_k$ when applied to $X$
    $C^-$ = set of sentences and their labels that have been correctly classified by $M_k$ when applied to $X$
Output: $M^*$
```

Given that CRFs operate at the sentence level, and because each token within a sentence needs to be classified (e.g., labeled with IOB), a misclassified sentence can be defined in multiple ways. For the same reason, combing the ensemble results can be implemented in multiple different ways.

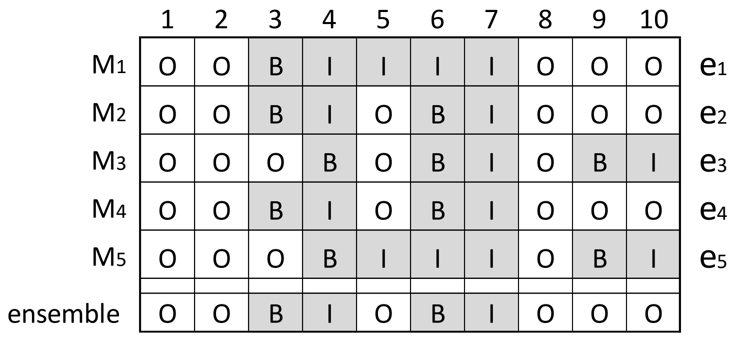
We experimented with three definitions of misclassification.

A. a sentence is misclassified if any of its predicted token labels is a false positive
B. a sentence is misclassified if any of its predicted token labels is a false negative

C. a sentence is misclassified if either A or B are true

Based on these definitions the training error e_k of the k-th unit model in the ensemble is simply the proportion of misclassified sentences. Using these definitions, we were able to better control the precision-recall tradeoff. For example, using the second definition forces the ensemble model to assign more weight to false negatives, thereby potentially improving recall at the expense of some precision.

Combining the ensemble results was done by aggregating I/O/B votes for each token in a given sentence. Figure 1 provides an illustration of this voting mechanism. The votes were aggregated by the classification weight of each unit model in the ensemble, with the weight defined as $\log(1 - e_k)/ e_k$.



	1	2	3	4	5	6	7	8	9	10	
M_1	O	O	B	I	I	I	I	O	O	O	e_1
M_2	O	O	B	I	O	B	I	O	O	O	e_2
M_3	O	O	O	B	O	B	I	O	B	I	e_3
M_4	O	O	B	I	O	B	I	O	O	O	e_4
M_5	O	O	O	B	I	I	I	O	B	I	e_5
ensemble	O	O	B	I	O	B	I	O	O	O	

Figure 1. Combining ensemble results. For this 10-token sentence, an I/O/B vote is cast by going through each token position (column) and summing the votes $\log(1 - e_k)/ e_k$ *for each possible I/O/B label, whereupon the label with the largest vote is selected.*

2.4 *Identifying disjoint mentions of adverse reactions*

Disjoint AR mentions were identified by developing regular expressions (regex) that captured different patterns of disjoint mentions. One such example was the regex

```
r'\bsuicidal\b[\s,]+(?:\w+[\s,]+){1,5}\bOTHER_WORD\b'
```

used to identify two-word disjoint AR mentions associated with suicide, that are separated by 1-5 words, start with the word '*suicidal*', and end with one of a list of other words (e.g., *'behaviors'*, *'ideation'*, *'attempt'*). The sentence below provides an example of three disjoint AR mentions (*Suicidal attempt, Suicidal behavior, Suicidal completion)*, which the regex above is able to identify.

Psychiatric Disorders : Suicidal ideation , attempt , behavior , or completion

The regular expressions were developed for only frequent disjoint mentions, which we defined as mentions that appeared at least five times in the training set. Each of these frequent disjoint mentions required a different regular expression, for a total of approximately 20 different expressions associated with approximately 250 different disjoint mentions.

2.5 *Adverse reaction negation detection*

Negation detection was performed using the ConText algorithm[13]. The algorithm identifies negated entities by searching for modifiers (e.g., the term 'no') that appear within a pre-specified token distance from a named entity. Applying the algorithm requires as input a list of modifiers. The modifiers are typically complied on a per-application basis. Each modifier is also associated with positioning (e.g., appearance before or after the named entity), which must be defined in advance.

The set of modifiers needed for our application was derived from the list of annotated *Negated*, *Factor*, *Drug Class*, and *Animal* named entities in the training set. Determining the positioning of these modifiers was done by analyzing examples. Similar to the detection of disjoint entities, we only used frequent modifiers, and modifiers that resulted in reasonable performance.

2.6 *Post-processing*

Partly due to method errors and partly due to textual artifact, a series of post-processing operations was needed to cleanse the candidate set of positive AR mentions produced by previous steps. These cleansing operations were identified by analyzing frequent error patterns and by developing rules to rectify these errors.

The cleansing operations generally involved removing mentions that do not correspond to proper ARs, and editing mentions that appear to be ARs but not in their original identified form.

Examples of the former include single-word mentions such as *'disease'*, *'syndrome'*, and *'outcome'*, which are not proper ARs. Examples of the latter include removing punctuation from mentions, stripping extra spaces as in 'raynaud_'s phenomenon', and stripping random charters that appear at the end of proper ARs as in 'anxiety d'.

3. Results

3.1 Data statistics

The number of sentences parsed by our sentence segmentation algorithm was 15,030 and 14,871 for the training and test sets respectively. Of these sentences roughly 43% contained AR mentions.

Tables 1 provides mention statistics relevant to Task 3. The training and test sets contain a total of 12,792 and 11,611 non-disjoint AR mentions respectively. The unique number of non-disjoint AR mentions are 3,104 and 2,937 respectively. The test set contains 1,254 unique mentions that are not part of the training set. Additionally, 51% of the unique positive ARs (the goal of Task 3) in the test set were not included in the training set. The absence of such a large proportion of AR terms from the training set may have impacted generalization performance of our system.

Of the total 13,795 AR mentions for the training set, 7.3% represent disjoint mentions. For the test set the proportion of disjoint mentions is 8.5%. These proportions grow substantially when examining the number of unique mentions (22% and 24% respectively). Together, these two statistics suggest that the detection of disjoint AR entities represents a relatively large proportion of the overall problem, and that the test set contains more of them. 88% of the disjoint mentions in the test set were not part of the training set, which impacted our ability to develop regexes for these disjoint mentions.

Tables 2-3 provide negation statistics. The total number of negated relations for the training set is 761. Despite containing fewer labels, the test set contains more negated relations (866). For the training and test sets each negated term covered 1.2 and 1.8 AR mentions respectively. The majority of negated relations were related to drug class (67% and 46% respectively), highlighting the importance of this class of negations in the task. As with the AR mentions, 70% of the unique negation terms in the test set were not included in the training set, which impacted our ability to identify generalizable negation modifiers for the ConText algorithm.

Table 1. mention statistics

	training		*test*		*training + test*		*test - training*
	mentions	unique	mentions	unique	mentions	unique	unique
non-disjoint AR mentions	12792	2540	11611	2341	24403	3794	1254
disjoint AR mentions	1003	685	1082	714	2085	1312	627
negations	420	136	460	136	880	230	94
positive ARs	7038	2927	6343	2715	13381	4532	1605

Table 2. negated relation statistics

	training	*test*
negation	163	288
animal	84	178
drug class	514	400
total	761	866

Table 3. negation by type statistics

	training		*test*		*training + test*		*test - training*
	mentions	unique	mentions	unique	mentions	unique	unique
Negation	98	20	173	24	271	37	17
Factor	29	11	37	13	66	23	12
Animal	44	7	86	16	130	17	10
Drug Class	249	98	164	83	413	153	55

3.2 Feature selection

Table 4 summarizes the performance of each of the features (and combinations thereof) examined for the NER step. The performance statistics are provided for the classification of AR B/I labels, and are based on training five-fold-cross validation. Differences between feature selections is summarized by the F1 proportion of error reduction (F1-PER), which is the proportion of additional distance covered towards a perfect F1 score of 100%. PER is relative to a baseline CRF model that includes only the word being modeled as a feature.

The table shows that the sentence type and the lexical features used (vocabulary matching) add the most (8%) to performance. Of the lexical features (D-H), the vocabularies that include derived fuzzy-matched medical terms (G-H) are better than their counterparts (E-F), which do not include the derived terms.

Adding contextual features (words pre/post the word being modeled) provided a significant boost in performance (26%-31%). This led us to select the three word pre/post feature set as our final contextual feature set, to which we added certain combinations of features A-H. With the exclusion of the sentence type feature (C), each of features A-H added to the performance of the three word pre/post feature set. Our final feature set was the one marked with a '*' in Table 4.

Table 4. Feature selection for NER

		precision	recall	F1	F1-PER
	Baseline	77.4%	67.2%	71.9%	
A	POS	77.1%	68.5%	72.6%	2%
B	Shape	75.9%	67.6%	71.5%	-1%
C	Sentence type	79.9%	69.1%	74.1%	8%
D	MetaMap	77.0%	69.7%	73.2%	4%
E	ADR annotator	76.9%	70.0%	73.3%	5%
F	MetaMap + derived terms	77.3%	70.7%	73.9%	7%
G	ADR annotator + derived terms	76.8%	70.4%	73.5%	6%
H	Metamap + ADR + derived terms	77.8%	70.8%	74.1%	8%
	One word pre/post	85.1%	74.0%	79.1%	26%
	Two words pre/post	86.0%	75.7%	80.5%	31%
	Three words pre/post	86.7%	75.6%	80.8%	31%
	Three words pre/post + A	86.7%	77.0%	81.6%	3%
	Three words pre/post + B	85.5%	77.0%	81.0%	1%
	Three words pre/post + C	86.6%	75.9%	80.9%	0%
	Three words pre/post + D	86.4%	79.0%	82.6%	6%
	Three words pre/post + E	86.3%	79.1%	82.6%	6%
	Three words pre/post + F	86.7%	79.8%	83.1%	8%
	Three words pre/post + G	86.0%	79.8%	82.8%	7%
	Three words pre/post + H	86.2%	80.9%	83.5%	10%
	Three words pre/post + D + E	86.4%	80.1%	83.1%	8%
	Three words pre/post + F + G	86.6%	80.5%	83.4%	9%
	Three words pre/post + D + E + H	86.5%	80.8%	83.5%	10%
*	Three words pre/post + A + B + C + F + G	86.8%	81.4%	84.0%	11%

3.3 Adverse reactions named entity recognition

Tables 5-6 display a performance comparison of our CRF models for the identification of AR named entities. The performance statistics are provided for the classification of AR B/I labels. The comparison is made between a single CRF model and two CRF ensemble models. The first aims to improve recall ('CRF ensemble - R'), and is based on error definition B in Section 2.3. The second aims to improve both precision and recall ('CRF ensemble - PR'), and is based on error definition C in Section 2.3. The feature set used in our models is the one labeled with '*' in Table 4, i.e., the best performing feature set identified throughout our feature selection analysis. The CRF L_1 and L_2 regularization parameters, were set to 0.3 and 0.1 respectively, which were slightly larger than the cross-validated optimal parameters. The number of models in the ensemble (K) was set to 20, and the sampling misclassification proportion p_{miss} was set to 0.4. The former was determined by identifying the point at which the ensemble starts to converge, and the latter by examining a series of values in the range 0.3-0.7.

Table 5 displays five-fold-cross-validation training performance statistics, whereas Table 6 displays the results of applying the trained models to the test (evaluation/hold-out) set. Cross validation was done at the label level and not at the sentence level. The tables demonstrate relatively high performance of our NER models across both the training and test sets. Moving from the training to the test set lowered the F1 score by 1%-2%. This may indicate a minimal amount of overfitting, but is more likely due to the large difference in AR mentions appearing in both sets (as discussed in Section 3.1). The tables also show that the ensemble models always improve, though by a modest amount (F1-PER of 1%-5%), upon the single CRF model when cross-validating the training set. However, when applying the trained models to the test set, there was virtually no improvement. The recall ensemble ('CRF ensemble - R') provided the improvement it was designed to provide adding 2%-3% to recall at the expense of about 1.6% in precision. The ensemble that aimed to improve both recall and precision ('CRF ensemble - RP'), appears to improve only precision. The tables also suggest that the recall ensemble provides slightly better performance than the other ensemble.

Table 5. training five-fold-cross-validation performance statistics

	precision	recall	F1	TP	pred	system	F1-PER
single CRF model	86.8%	81.4%	83.98%	17239	19870	21184	
CRF ensemble - R	85.2%	84.4%	84.80%	17883	20991	21184	5%
CRF ensemble - PR	87.3%	81.1%	84.08%	17173	19667	21184	1%

Table 6. test set performance statistics

	precision	recall	F1	TP	pred	system	F1-PER
single CRF model	83.3%	82.0%	82.7%	15981	19175	19489	
CRF ensemble - R	81.6%	83.9%	82.7%	16357	20055	19489	0.3%
CRF ensemble - PR	84.1%	81.0%	82.5%	15785	18765	19489	-0.8%

3.4 Identifying adverse reaction disjoint entities

On the training set, the regexes we developed could identify at most only 22% of the disjoint mentions. This was by design since we developed regexes to detect only frequent disjoint mention patterns. The precision, recall, and F1 of our method was 68%, 87%, and 76% respectively. Relative to the goal of task 3, applying our method provided a F1-PER (relative improvement) of 13%, thus its effect was overall positive. For the test set our regexes were able to identify at most only 11% of the mentions, but applying them to the test set resulted in a slight negative affect, due to a precision < 50%.

3.5 Negation detection

Applied to the training set, the precision, recall, and F1 of our method was 82%, 72%, and 77% respectively. Relative to the goal of task 3, applying our method provided a F1-PER (relative improvement) of 39%. Applied to the test set our method resulted in a precision, recall, and F1 of 79.8%, 67.9%, and 73.4% respectively. Relative to the goal of task 3, applying our method improved the F1 score by 32%. Hence, the performance of our method had a slightly lower overall effect on the test set.

3.6 Overall System performance

Table 7 displays overall system performance, i.e., the application of our AR NER method followed by our methods for identifying AR disjoint mentions, negation detection, and post processing. The performance statistics are also provided at the macro level (averaged across product labels), and each row in the table represents one of our three submissions for the task. The submissions varied by the approach taken for the NER step. The table shows that both ensemble approaches provided a slight improvement over a single CRF model, and the best result was obtained for the ensemble that focuses on recall. According to our evaluation the three steps following the NER step contributed 1%-1.5% to the overall F1 score.

Table 7. final system results

	Micro			Macro							
	precision	recall	F1	precision	recall	F1	TP	FP	FN	pred	system
single CRF model	81.28	79.32	80.28	81.10	78.81	79.20	5031	1159	1312	6190	6343
CRF ensemble - R	81.18	79.69	80.43	81.47	79.28	79.67	5055	1172	1288	6227	6343
CRF ensemble - PR	82.71	78.05	80.31	82.64	77.73	79.42	4951	1035	1392	5986	6343

4. *Discussion*

To our knowledge annotated product labels data has not been available prior to its release for TAC ADR track. For the first time, this enabled the investigation of machine learning approaches for extracting adverse reactions from product labels. Based on our results, it appears that extracting positive mentions of adverse reactions from product labels can be done with reasonable accuracy using machine learning-centric approaches.

According to our estimates 93% of the overall problem is associated with the NER step. It appears that the quality and size of annotated training data made available for the task was sufficient to develop and train a machine learning approach for the NER step. However, the performance of our NER approach may have been hindered by the large proportion of AR terms that were part of the test set but missing from the training set. This discrepancy between the two sets also suggests that a larger and more inclusive training set would benefit future development.

The CRF ensemble approach we developed for the NER step is currently in an experimental state and requires additional research. We did not derive theoretical guaranties (error bounds) for the ensemble, and its construction should currently be viewed as heuristic. The ensemble did improve performance, and also allowed us better control of the precision-recall tradeoff. However, the improvement it provided was modest. This is expected given the relatively high performance provided by the base CRF on which the ensemble was built.

The lexical features we employed had a large positive effect on the performance of our NER method (added 2%-3% to the F1 score) and highlight the importance of incorporating domain knowledge into the system. They do however require a considerable effort to develop and refine. We conjecture that refining them further than we did would have provided an even greater benefit.

We found it impracticable to develop a machine learning approach for the identification of disjoint entities and for negation detection. This was due to the limited number of training examples and the relatively large diversity of terms involved. The performance of our rule-based methods for these two sub-problems was mediocre and did not generalize well to the test set. This is somewhat expected given that they were manually developed for a training set that was substantially different than the test set with respect to the disjoint entities and negated terms appearing in both. Despite their relatively lower importance in the overall solution, improving methods to solve these two sub-problems should not be overlooked.

In conclusion, extraction of positive mentions of ARs can be done with reasonable accuracy using approaches whose core rests on machine learning. Similar to other NLP problems, it appears that additional training data will benefit development and system performance. The extent in which the current training set needs to be augmented is currently unknown, but would be an interesting project to pursue.

5. *References*

1. Harpaz R, Callahan A, Tamang S, et al. Text mining for adverse drug events: the promise, challenges, and state of the art. *Drug Saf.* 2014;37(10):777-790.
2. Boyce RD, Ryan PB, Noren N, et al. Bridging islands of information to establish an integrated knowledge base of drugs and health outcomes of interest. *Drug Safety (in press).* 2014.

3. Duke J, Friedlin J, Li X. Consistency in the safety labeling of bioequivalent medications. *Pharmacoepidemiology and drug safety.* 2013;22(3):294-301.
4. Voss EA, Boyce RD, Ryan PB, van der Lei J, Rijnbeek PR, Schuemie MJ. Accuracy of an automated knowledge base for identifying drug adverse reactions. *Journal of biomedical informatics.* 2017;66:72-81.
5. Friedlin J, Duke J. Applying Natural Language Processing to Extract Codify Adverse Drug Reaction in Medication Labels http://omop.fnih.org/OMOPWhitePapers2010.
6. Kuhn M, Campillos M, Letunic I, Jensen LJ, Bork P. A side effect resource to capture phenotypic effects of drugs. *Molecular systems biology.* 2010;6:343.
7. Fung KW, Jao CS, Demner-Fushman D. Extracting drug indication information from structured product labels using natural language processing. *Journal of the American Medical Informatics Association.* 2013;20(3):482-488.
8. Adverse Drug Reactions Database. http://www.imi-protect.eu/adverseDrugReactions.shtml.
9. Adverse Drug Reaction Extraction from Drug Labels, TAC 2017. https://bionlp.nlm.nih.gov/tac2017adversereactions/.
10. Aronson AR, Lang FM. An overview of MetaMap: historical perspective and recent advances. *Journal of the American Medical Informatics Association : JAMIA.* 2010;17(3):229-236.
11. Lindberg DA, Humphreys BL, McCray AT. The Unified Medical Language System. *Methods of information in medicine.* 1993;32(4):281-291.
12. Okazaki N. CRFsuite: a fast implementation of Conditional Random Fields (CRFs). http://www.chokkan.org/software/crfsuite/. 2007.
13. Harkema H, Dowling JN, Thornblade T, Chapman WW. ConText: an algorithm for determining negation, experiencer, and temporal status from clinical reports. *Journal of biomedical informatics.* 2009;42(5):839-851.