# Markov Logic Networks for Adverse Drug Event Extraction from Text

Sriraam Natarajan, Vishal Bangera, Tushar Khot*, Jose Picado$^{+}$,
Anurag Wazalwar, Vitor Santos Costa$^{\#}$, David Page* and Michael Caldwell$^{++}$
Indiana University, *University of Wisconsin-Madison, $^{+}$ Oregon State University,
$^{\#}$University of Porto and $^{++}$Marshfield Clinic

**Abstract.** Adverse drug events (ADEs) are a major concern and point of emphasis for the medical profession, government, and society. A diverse set of techniques from epidemiology, statistics, and computer science are being proposed and studied for ADE discovery from observational health data (e.g., EHR and claims data), social network data (e.g., Google and Twitter posts), and other information sources. Methodologies are needed for evaluating, quantitatively measuring, and comparing the ability of these various approaches to accurately discover ADEs. This work is motivated by the observation that text sources such as the Medline/Medinfo library provide a wealth of information on human health. Unfortunately, ADEs often result from unexpected interactions, and the connection between conditions and drugs is not explicit in these sources. Thus, in this work we address the question of whether we can quantitatively estimate relationships between drugs and conditions from the medical literature. This paper proposes and studies a state-of-the-art NLP-based extraction of ADEs from text.

**Keywords:** Natural Language Processing; Adverse Drug Event Extraction; Markov Logic Networks; Statistical Relational Learning

## 1. Introduction

Adverse drug events (ADEs) have been receiving substantial national attention since an Institute of Medicine (IOM) report found serious gaps in pharmacosurveillance capacity, flagging this task as a top research priority and prompting a series of recommendations to the FDA's Center for Drug Evaluation and Research (CDER) [1]. The problem is felt worldwide, with initiatives stemming from international entities such as the European union's EU-ADR [2] and PROTECT[1]. Responses in US have included the FDA's Sentinel Initiative and

[1] http://www.imi-protect.eu/

Mini-Sentinel, the Observational Medical Outcomes Partnership (OMOP)[2] , the Reagan-Udall Foundation (RUF), and the Innovation in Medical Evidence Development and Surveillance (IMEDS) [3] arm of RUF that incorporates and builds upon content from the earlier OMOP and Mini-Sentinel. These organizations have driven, or currently are driving, significant research into statistical and computational methodologies [3, 4, 5, 6] for post-marketing surveillance of drugs by analyzing observational clinical data in the form of claims and electronic health record (EHR) databases, as well as in some cases social media and other semi-structured data, e.g., text and natural language processing (NLP). Meanwhile the problem of ADEs has continued to grow in impact and importance. In 2012, the Office of Disease Prevention and Health Promotion in the U.S. Department of Health and Human Services published a draft National Action Plan for Adverse Drug Event Prevention[4], which notes that in the U.S. alone ADEs are responsible for:

- one-third of all adverse events of any kind during hospital stays and affect two million stays annually [7]
- over 3.5 million physician office visits [8] and 1 million ER visits [9]
- $3.5 billion in U.S. health care costs [10]

To summarize, the primary contribution of our work is to present a probabilistic method for summarizing what the research community knows about ADE pairs. The approach is a novel application of recent advances in machine learning for information extraction and NLP. Our approach builds upon the use of Markov Random Fields (MRF) that have been successfully employed within the NLP community [11]. We use a template representation of these MRFs using a formalism called Markov Logic Networks (MLN) [12]. Our second contribution is a quantitative evaluation of the approach to the specific application of NLP for ADE discovery. In addition to the quantitative evaluation, we also give a qualitative evaluation that examines the strengths and weaknesses of the approach for this application. Based on this qualitative evaluation, we extend our framework to incorporate training of these MLNs and subsequently MRFs based on the data. The initial expert-based MLN exhibits competent performance but can be improved when data are used to "refine" the MLN.

It must be mentioned clearly that the aim of this paper is to demonstrate that for OMOP definitions and similar definitions, we can use the literature to verify complex definitions in our case, OMOP. As far as we are aware, not many NLP techniques are proposed for these definitions. Hence, we could not compare against any standard technique that uses NLP. Also, as shown clearly in our empirical evaluations (and the citations in there), our results are comparable or better than the current ADE methods that operate using OMOP definitions.

To summarize, we propose a probabilistic method that given a drug-effect pair searches PubMed for abstracts and converts these abstracts to standard NLP features. These features are then used in a probabilistic classifier based on MLNs to obtain a distribution over whether the drug-effect pair is indeed an ADE. In the rest of the paper, we first discuss the prior work and provide the required technical background on MLNs and NLP. We then present our

[2] http://omop.org/
[3] http://imeds.reaganudall.org/
[4] http://www.health.gov/hai/pdfs/ADE-Action-Plan-508c.pdf

approach in detail before the evaluation on OMOP ADE pairs. We next provide an in-depth discussion of the salient features of the approach before concluding by presenting avenues for future research.

## 2. Background

To make this paper self-contained, we begin with a discussion about the OMOP initiative, followed by a brief tutorial on Information Extraction, MLNs and their use for NLP.

**Initiative by OMOP:** In 2009, FDA, PhRMA, and the Foundation for the NIH initiated the Observational Medical Outcomes Partnership (OMOP) – which has now fed into the Reagan-Udall Foundation (RUF), specifically Innovation in Medical Evidence Development and Surveillance (IMEDS) – to evaluate and improve methods for discovering ADEs from observational medical data, such as health insurance claims data, Medicare and Medicaid data, or electronic health record (EHR) data [13]. To facilitate evaluation and comparison of methods and databases, OMOP established: a Common Data Model so that disparate databases could be represented uniformly; definitions for ten ADE-associated health outcomes of interest (HOIs); and drug exposure eras for ten widely-used classes of drugs. OMOP's 2010 evaluation had on average three different competing definitions for each HOI ranging from a most- to a least-stringent definition. These definitions employed ICD9 codes and other data types yielding a total of 30 HOI definitions. The end goal of this work was to encourage the development, quantitative evaluation, and comparison of methods for uncovering new (previously unknown and perhaps even unanticipated) ADEs. To evaluate methods, it is necessary to use known ADEs as ground truth and determine how well the new methods could have uncovered these ADEs had they been unknown. At its initiation, OMOP took a rigorous approach based on available drug label information to associate drug classes with HOI definitions [14]. Methods were then evaluated by their ability to correctly rank the pairs from most likely to least likely to be a true association. Ranking quality was evaluated by area under the receiver operating characteristic (ROC) curve, or AUCROC.

We use the OMOP definitions for the quantitative evaluation of our approach. We first evaluate our NLP approach on the 2010 OMOP ground truth and show that our approach yields a high AUCROC with respect to that ground truth. We then look more closely at where our system's results disagree with OMOP's ground truth. In some instances this investigation reveals probable errors in OMOP's ground truth, owing either to OMOP's high standard of evidence (drug labels) for ADEs or to discoveries occurring after OMOPs initiative. In other instances this investigation reveals shortcomings in our current approach that point to directions for further research.

**Information Extraction:** Information extraction (IE) [15, 16, 17, 18] is the process of automatically extracting structured information from unstructured data, where unstructured data consists of machine-readable documents. One of the tasks involving information extraction is relation extraction, which consists of identifying instances of entities in text and the relationships between those instances. Adverse drugs events discovery is a relation extraction problem, where the entities are drugs and health outcomes, and the relations indicate whether a health outcome is an adverse effect associated with taking a drug.

Many approaches have been developed to extract adverse drug events from a

large number of diverse information sources. Gurulingappa et al. [19] used supervised learning methods, such as Nave Bayes, Decision Trees, Maximum Entropy and Support Vector Machines, to perform automatic identification of adverse drug event assertive sentences, by exploiting lexical, syntactic and contextual features. Friedman [20] describes an approach that uses the Electronic Health Records (EHR) to extract novel adverse drug events based on coded data (structured) and narrative records (unstructured). Shetty and Dalal [21] performed disproportionality analysis on PubMed articles that mention a drug and adverse effect (AE) to discover new drug-AE associations. Bian et al. [22] used Twitter data to mine drug-related adverse events by building a classifier over textual and semantic features.

We use IE techniques in order to extract knowledge about text patterns and string similarities and assign scores to the proposed ADEs. Consider an expert (say an epidemiologist) that is evaluating a set of adverse events by scanning through a set of abstracts. He/she will scan each abstract looking for *patterns* that mark the presence or absence of adverse events. Only in very few cases can the expert have full confidence in a pattern. Instead, the expert will rely on sets of patterns, where some definitely will be stronger than others. Moreover, patterns may reinforce or weaken each other. The strength of a pattern thus depends both on the context where it is applied and on the other patterns being considered.

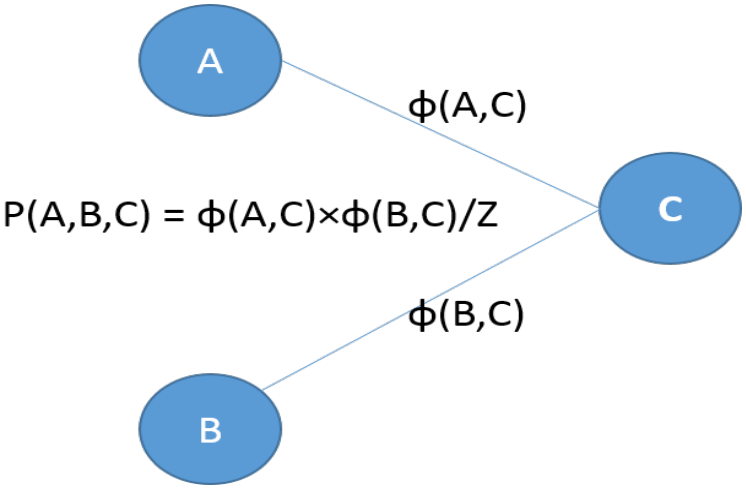
We aim to quantify these mental patterns by using a descriptive language such as first-order logic and model the uncertainty by weights (or probabilities). More precisely, we shall assume that drugs and conditions are *random variables* that may be *present* or *absent* in a empirical study. Patterns connect these random variables. We observe that the work of the expert is based on the principle that the same patterns will repeat in different abstracts. In other words, abstracts are not a random bunch of items, or random variables. Instead, the items are connected through a set of applicable rules.

The most commonly used NLP methods are Conditional Random Fields (CRFs) [23] which are essentially special cases of the more general Markov Random fields (MRFs). An MRF is an undirected graphical model that consists of a set of nodes (V) and edges (E). They factor the joint probability distribution over the variables as products of clique potentials[5]. Assuming that each node in V is a random variable, the MRF defines a distribution over V as a product of potentials. For example, in Figure 1, there are three nodes A, B and C. Since there is no clique of size 3 in the Figure (i.e., no triangle) , the joint distribution over the three variables,

$$P(A, B, C) = \frac{\phi(A, B)\phi(B, C)}{Z} \tag{1}$$

where $\phi$ is the potential of the clique and $Z$ is the normalization term ($Z = \sum_{A,B,C} \phi(A, B)\phi(B, C)$). Typically, the structure of the model (the cliques) are defined Apriori and parameters ($\phi$) are learned using data. While they are popular, designing specific MRFs for the problem at hand requires a machine learning expert. On the other hand, elucidating knowledge from domain experts is more natural if the formalism employed underneath is a general purpose one. First-

[5] A clique in a graph is a fully connected sub-graph of the original graph. A triangle is a clique of size 3, an edge is of size 2 and a fully connected square with both diagonals if of size 4.




[h]

**Fig. 1.** An example of a MRF. A,B,C are the three variables and $\phi$ are the potentials between the cliques. Note that the largest clique is of size 2. Given these parameters, the final joint distribution can be obtained by the product of these potentials (normalized).

order logic has been employed specifically in the Artificial Intelligence community for this purpose.

Also, Random Fields are most often used when there is a regular relation between items: a sequence of words in text CRFs or a pixel array in vision MRFs. They provide a very effective approach to compute a collective probability. Unfortunately, the textual ordering in the abstract is clearly not directly relevant to our task. This suggests that we need a flexible representation. well suited for irregular and structured problems.

**Markov Logic Networks:** Hence, we employ the first-order logic formalism of Markov Logic Networks (MLNs) [12] to model the relationships described in the textual data. An MLN consists of weighted first-order formulas where the first-order formula captures the structural (qualitative) relationship between objects of interest while the weight of the formula quantifies the strength of the relationship. Each first-order formula is called a clause. Consider the following two MLN clauses:

$$0.5 \quad smokes(x) \longrightarrow cancer(x)$$
$$1.0 \quad friends(x, y) \wedge smokes(x) \longrightarrow smokes(y)$$

The first clause expresses the knowledge that if a person (denoted by $x$), smokes, then he/she is likely to have cancer. The second clause expresses the idea that if two persons are friends and one of them smokes, the other is likely to smoke as well. Note that x and y are variables that can be instantiated with values such as *Ann, Bob, Cathy,* etc. The numbers in each of the MLN clauses are essentially log-odds and hence the probability of friends having similar smoking habits is $log(1/0.5)$ times more likely in the world compared to smoking causing cancer.

In order to apply this technique, we need : the set of rules; the set of weights; and an algorithm to compute probabilities.

One of the key reasons for using MLNs to capture relation extraction knowl-

[h]

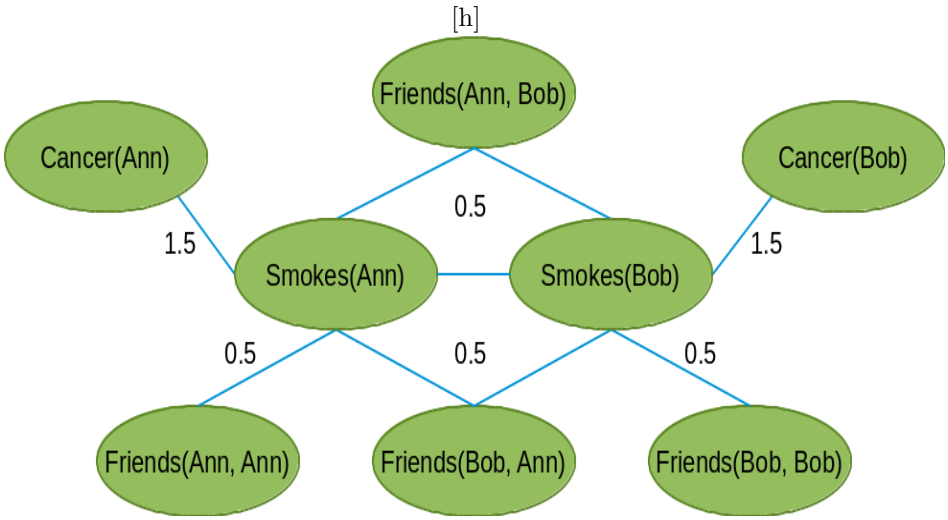



**Fig. 2.** Example MRF generated from the MLN. We consider only the second clause with two groundings *Ann* and *Bob*. Each predicate is instantiated with appropriate values for the variables leading to the MRF. This example serves to provide the intuition that MLNs can be simply viewed as templates for constructing MRFs. The cliques all share the same potential.

edge is that MLNs provide an easy way for a domain expert to specify the background knowledge as first-order logic clauses. We therefore assume that the rules were obtained from an expert. Most algorithms learn the weights of these rules from data. Given the variety of possible algorithms and how they depend on combinations of parameters, we asked our expert to propose a set of weights, In order to simplify this task, all our rules have a simpler form: implication statements of the form $a(X) \longrightarrow b(X)$.

MLNs can be seen as MRF generators. Given a MLN and the set of possible values for the variables, eg *Ann* and *Bob* in Figure 2 (called *groundings*), most algorithms construct a MRF. In the example shown in Figure 1, there are two people *Ann* and *Bob*. Correspondingly, there are two smoker nodes and four friends nodes in the MRF. Friends$(x, y)$ denotes that person $x$ is a friend of person $y$. A person can be a friend of him/herself, thus, *Ann* is a friend of her and *Bob* is a friend of himself. The weights are "shared" among all the instances of the same clause. For instance, the potential on the MRF corresponding to all instances of people smoking and being friends is the same and will be equal to 1 in this case. Similarly, the weights of the cancer-smokes clique will all be the same and equal to 0.5.

We use MLNs as a template for constructing irregular MRFs, that are the standard in the NLP literature. The use of MLNs allows us to generalize across multiple documents and ADE pairs. For instance, in our experiments, using 50 documents on 27 ADE pairs with 15 rules yielded a MRF of about $10,000$ nodes. Constructing this 10k node MRF manually will be extremely cumbersome and employing the use of MLNs allows us to bypass this issue and achieve effective generalization. What we exploit is an automatic construction of the ground MRF that requires minimal effort from the expert.

Most algorithms assume that an expert specifies the set of the rules and simply learn the weights of these rules from data. This weight learning process

must be distinguished from query time inference where the set of rules and the corresponding weights are provided and the system can be queried for a particular situation. For instance, in the example, one can query the probability of someone having a cancer given that his/her friend is a smoker.

It must be mentioned that while MLNs allow for the full first-order logic syntax to be employed in sentences, for the purposes of this work, we only use implication statements of the form $a(X) \longrightarrow b(X)$ which essentially states that attribute $b$ must be true whenever attribute $a$ is true for a particular object $X$. While in simple logic, this is a strong statement, MLNs allow for a softer form that is more probabilistic. If the weight of this statement is high, then $b$ will mostly likely be true when $a$ is true but if the weight is negative, it is mostly likely to be false. We refer to the book by Domingos and Lowd [12] for more details. In our work, we use the Tuffy system [24] to perform inference on the MLNs. One of the key advantages of Tuffy is that it can scale up to millions of documents.

**Using MLNs for NLP:** Several approaches have been proposed for knowledge extraction in general from biomedical literature. Riedel et al. [25] and Poon and Vanderwende [26] proposed approaches based on Markov Logic to perform biomedical event extraction, getting competitive results in the BioNLP09 Shared Task. These methods are shown to outperform standard machine learning algorithms on NLP tasks. They employed MLNs that used syntactic (word form) and semantic features (dependency paths) to capture the models for the extraction of nested-bio-molecular events from research abstracts, and then performed joint inference using these models. MLNs have become popular in biomedical extraction tasks, as has been demonstrated in the BioNLP11 Shared Task, where the top systems [27, 28] employed approaches based on Markov Logic. One of the key attractive features of MLNs is that they are based on first-order logic and hence allow for generalizable knowledge that can be used across multiple tasks. Another attractive feature of these MLNs is that the expert can simply write as many rules in first-order logic and efficient learning algorithms exist that can learn the weights (these weights reflect how true the rules are).

## 3. Extracting ADEs from Text

We now provide the details of our proposed method.

### 3.1. Markov Logic Networks for ADE Extraction

Our approach for evaluating adverse drug events is presented in Figure 3. The system can be defined as follows:

**Given:** A set of $\langle drug(s), condition(s) \rangle$ tuples
**To Do:** Determine **P(drug(s) cause condition(s))** i.e., output the probability that a given (possibly set of) condition(s) is an adverse event of (possibly a set of) drug(s) by using prior published research as evidence.

The aim of our work is to quantify what the research community knows about the drug-event (DE) pairs as a probabilistic function. Note that while we refer to the events as drug-event pairs, our methods are not restricted to just pairs

but can handle complex interactions such as multiple drugs/conditions causing multiple adverse conditions. In this work, we restrict ourselves to drug-event pairs (henceforth called as DE) only for simpler exposition of the ideas and for comparison to OMOP ground truth.

### *3.1.1. Searching for relevant abstracts:*

Given this problem definition, the first step is to obtain the set of previously published literature that provides evidence about the given drug-event pair, or DE. To this effect, we query PubMed for a given set of DEs. An example query is "ACEInhibitor Angioedema". For each DE, we obtain a set of articles. We consider only the abstracts of these articles in this work (but our model can handle full articles). For this step,

**Input:** A set of DEs
**Output:** A set of K PubMed abstracts for each DE.

These top K articles serve as the natural language textual evidence for the pair. For each article, we use two features to "weigh" the importance of the article: (1) Eigenfactor [29] of the journal and (2) the recency of the article. We use the eigen factor values for each article directly. The articles in PubMed are in different stages of verification and to ensure authenticity we only use the articles which are *finally* approved by PubMed. For recency, we used a discount factor of 0.9. We experimented with several other discount factors in the range of $0.6 - 0.95$ and found that beyond 0.7, the results did not change significantly. If the article is published within the last year, it has a weight of 1, and every preceding year after that has its weight lowered by a multiplicative factor of 0.9 (i.e., 2 years old article has a weight of 0.9, 3 years old article will have a weight of 0.81, fours years old has a weight of 0.72 etc). The intuition is that most recent articles published in high-quality journals will have a higher weight than more recent articles in low quality journals and older articles in high-quality journals.

The key idea is that we aim to model a human expert who does not rely on a single article to infer a meaningful association between drugs and events but rather rely on a broad set of articles. Not each of these articles are considered equally important by the expert and hence we use the weights accordingly. The choice of the parameters, although made before the experiments started, seem reasonable for the task at hand. The parameter K indicates how many publications will refer to the ADEs. Increasing K increases the sensitivity of the algorithm, but will also reduce precision and is computationally expensive. We chose K=50 before the experiments, in order to ensure the MLN system can compute the probabilities in feasible time These abstracts serve as the input to the next stage.

The second stage of our approach has two distinct phases (1) String similarity phase, and (2) Semantic relation extraction phase. We describe these two phases separately to demonstrate that standard information retrieval measures may not suffice in the task of identifying ADEs and that the task requires more semantic understanding of the text.

For each abstract obtained in the first step, we identify the sentences that contain the corresponding DE pair, i.e. sentences that contain both the drug and the condition. Note that we do not distinguish yet whether the DE is an adverse

[h]

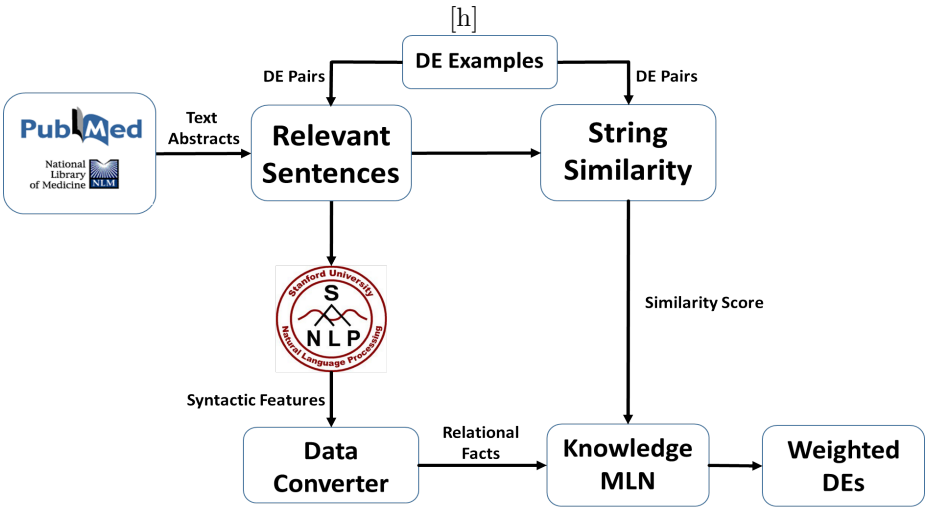



**Fig. 3.** Steps involved in the evaluation of adverse drug events (ADEs).

event or not, or simply if the drug and the condition are related, we just keep the sentences plain text to be used in the next step.

### *3.1.2. String Similarity:*

For the string similarity step,

**Input:** The given set of 50 abstracts and the current DE.
**Output:** The average string similarity scores between the DE and the considered abstracts.

In the string similarity phase, we use simple document matching metrics such as cosine similarity, Jaccard similarity, Jaro-Winkler similarity and Sorensens similarity [30]. The goal of this phase is to obtain a syntactic measure of the similarity between the DE pair and the abstract at hand. In other words, given a DE we find its support in the text. Note that this measure simply searches for mentions and thus does not distinguish between whether a given DE is an adverse event or not, or if there is no relationship between the drug and the condition. Cosine similarity measures the cosine of the angle between two vectors, where the vectors are the frequency of occurrence vectors of the documents. In our case, vectors store the occurrence of letters. While in this step we only compute string similarities between each DE and literature found on the web, these can also be seen as evidence in the MLN constructed in the second step. Note that the use of string similarities gives us a good baseline. This is typically the approach used by many systems that do not explicitly parse the entire medical abstract but compute some "distance" between the query and the abstract. The aim of this step is to demonstrate the value of *deeper* understanding of text to better improve the identification of ADE from text.

### *3.1.3. Semantic relation extraction:*

Semantic relation extraction on the other hand aims to identify features that can be employed for a deeper analysis of the given text.

For the semantic similarity step,

**Input:** The given set of 50 abstracts and the current DE.
**Output:** Probability that the current DE is actually an ADE based on the 50 extracted abstracts.

To obtain the relevant features, we run the sentences obtained in the previous step through a standard NLP tool such as the Stanford NLP toolkit [31, 32] to create relational linguistic features. The created relational features are lexical, syntactic and semantic features, such as part-of-speech tags, phrase types, word lemmas, parse trees and dependency paths, which provide a representation of grammatical relations between words in a sentence. These are standard features used in the natural language processing literature and we find them to be very useful in our problem as well. These features are used to identify a deeper interaction between the drug and adverse event mentions in the text. For instance, it is useful to say that if the drug and object has a dependency path between them and the word "causes" appears in the dependency path, there is a chance that the drug causes the effect. Note that this is not always true and hence this knowledge is treated as probabilistic (weighted and uncertain) knowledge.

In addition to these features, we use an entity recognizer to identify drug and effect mentions. For example consider the following text: "There is evidence that MI is caused by the intake of Cox2ib". This sentence would lead to the features drug(Cox2ib) and effect(MI). These features (called as predicates in MLN literature) are then used as evidence to query the MLN for probable adverse event.

These relevant features along with the similarity scores are together considered while constructing the rules in the next step. A high-level flow of this step is presented in Figure 4. As can be seen, we run the 50 abstracts from PubMed through the NLP parser (Stanford NLP in our case). These are then used to create NLP features (they are presented in detail in Appendix). These features are then used in the MLN as we discuss below.

Drugs, effects, and the relationship between them form the evidence. Note that after having the information about drugs and effects, we use the features drug and effect to define a MLN clause that indicates that the drug $d$ with word $dw$, and effect $e$ with word $ew$, are present in an ADE r:

```
effect(e), effectWord(e,ew), drug (d), drugWord(d,dw),
     present(r,d), present (e,d) --> deADE(r,d,dw,e,ew)
```

$dw$ and $ew$ are variables and will be substituted by the corresponding values when performing reasoning. If we add a weight of infinity to this rule, then it means that this rule is always true. For the example in the above paragraph, $dw$ could correspond to *Cox2ib* and $ew$ could correspond to MI. Then the rule simply states that if the same sentence has *Cox2ib* as the effect and *MI* as the drug then it is always true that the adverse effect of *Cox2ib* is $MI$. Of course, since this is not always true for other drug and effect pairs, they are considered to be probabilistic and hence we soften them using weights that are then used to create potentials of the ground MRF.

[h]

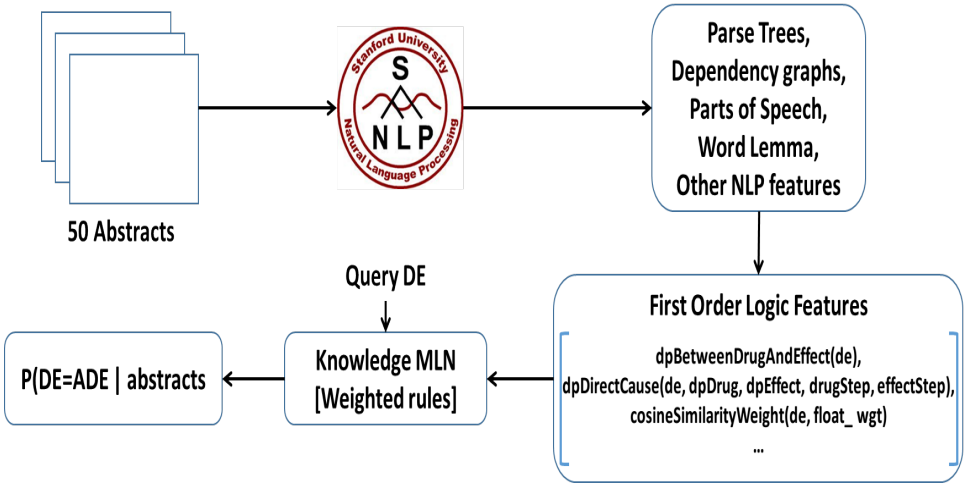



**Fig. 4.** Steps involved in the deeper semantic extraction phase that employs MLNs. The extracted 50 PubMed abstracts are given as input to a NLP parser. The resulting parse trees, dependency graphs, parts of speech and other NLP features are then converted to MLN format (first-order logic facts). They are then used as input along with the current query DE to obtain the final posterior distribution.

[h]

| Weight | Rules | Type |
|---|---|---|
| wgt | cosineSimilarityWeight(r, wgt) ⇒ adverse(r) | Sim |
| 1 | dpDE(r) ⇒ adverse(r) | Basic |
| 3 | deADE(r,d,dw,e,ew), preHW(wo,dw), postHW(wo,postwo), ws(postwo,"induced"), dt(ar,se,ew,wo,AMOD) ⇒ adverse(r) | Basic |
| 1.5-3 ($\propto$ l) | deADE(r,d,dw,e,ew), dp(ar,se,ew,dw,dp), contains(dp, "prep_after"), dpL(ar,se,ew,dw,l) ⇒ adverseC(r, l) | Prep |
| 1.5-3 ($\propto$ l) | deADE(r, d, dw, e, ew), word(wo), ws(wo, "risk"), dp(ar, se, wo, ew, dp1), dp(ar, se, wo, dw, dp2), dpL(ar, se, ew, dw, l), contains(dp1, "prep_of"), contains(dp2, "partmod") ⇒ adverseC(r, l). | Prep |

**Table 1.** A sample of the relation extraction knowledge. dpDE denotes that there is a dependency path between the drug and effect in a proposed ADE (they are in the same sentence), deADE denotes that drug and effect are in a proposed ADE, dp denotes the dependency path between two words, dpL denotes the length of the dependency path between two words, preHW denotes prehyphen word, postHW denotes posthyphen word, ws denotes word string, dt denotes dependency type. Rules are classified into rule types based on the features used shown in Type column.

adverseC is an intermediate target predicate that represents the length of the dependency path between the drug and the effect. The shorter length indicates a stronger correlation between the drug and the effect. Note that the weights are set based on the length of this path. Shorter the path, higher the weight. Please refer to appendix for more weights. The weights simply indicate the relative importance of one rule over the other.

Once we have the drugs, effects, DE pairs, string similarities and textual evidence, we employ an MLN that captures the relation extraction knowledge for identifying ADEs using rules about text patterns and string similarities. Some of the example rules (out of the 15 rules that we use) that we used to capture text patterns and string similarities are shown in Table 1. The first-order rules can be interpreted in English as,

**Rule 1:** If there is a cosine similarity between the DE pair and MEDLINE abstracts, the proposed ADE is true with a weight relative to the cosine similarity

**Rule 2:** If a drug and an effect are present in a proposed ADE and a sentence contains both the drug and effect, the ADE is true

**Rule 3:** If a drug and an effect are present in a proposed ADE, and a sentence contains both the drug and effect, and the sentence contains the pattern drug-induced effect, the ADE is true

**Rule 4:** If a drug and an effect are present in a proposed ADE, and a sentence contains both the drug and effect, and the sentence contains the pattern effect after drug, the ADE is true

**Rule 5:** If a drug and an effect are present in a proposed ADE, and a sentence contains both the drug and effect, and the sentence contains the pattern risk of effect and drug is a participial modifier of the word risk, the ADE is true

We divide the rules into three types: (1) *Text similarity based rule* (Sim) (2) *Dependency path based rules* that check for particular words occurring in a dependency path between the drug and effect word (Basic) (3) *Rules that check for dependency paths* as well as specific propositional dependencies in the paths (Prep). We evaluate the contribution of these rules in the evaluation section. Note that all rules mentioned above are considered as soft rules, and we manually assigned weights to the rules based on the lengths of the dependency paths and the specificity of the rule. Of course, these weights can be learned using data. Once the MLN is constructed and weights have been assigned, we query the MLN for the posterior probability on the adverse relation, using as evidence the relational linguistic features from the extracted abstracts, as well as drugs, effects, DEs and string similarities.

MLNs bring key advantages to this task. (1) We are able to specify rule relationships in the data about dependency graphs, parse trees etc besides the standard features used in NLP literature. This allows us to define richer MRFs than the ones typically employed in the literature. (2) The use of the template based formalism allows us to write as many rules as possible without worrying about the size of the grounded network. (3) The rules can be written by "experts i.e., the researchers who typically read these papers, without having understand the fundamentals of graphical models when writing these rules. Hence, to summarize our algorithm for identifying DEs consists of the following key steps:

– For each DE pair:

1. Search through the list of abstracts and identify the top K relevant abstracts (K=50).
2. Compute string similarity scores between the given DE and the retrieved abstracts.
3. Run each abstract through Stanford NLP parser and select the linguistic features relevant to the given DE.
4. Query the MLN for the probability of the DE being an adverse drug event pair($P(DE|evidence)$). During this step, the MLN engine computes the number of times (count) each rule is satisfied for the current DE pair across all these documents. Then it multiplies the corresponding weights with the counts, sums these weighted counts and normalizes them to obtain the final probability.
5. Store the probability and the DE pair to a global list

– Sort the global list and output the rankings of the DE pairs according to the posterior probability.

This DE pairs order the different drug event pairs based on their likelihood of being an adverse event.

### 3.2. Bringing Expert and Data Together - Refined MLNs

As mentioned earlier, we rely on the expert to write the rules and we simply "soften" the rules by assigning the weights. While reasonable, there is a burden on the expert to list all the rules that he/she uses in identifying the ADE. We now propose to relax this requirement. The key idea is that *the expert writes as much rules as possible and the system discovers more rules that rely on data to complement the contribution of the expert.* To achieve this, we require a learning algorithm to revise the expert's theory as needed. We use a non-parametric MLN learning algorithm based on functional-gradient boosting [33] as this method has been proven to be effective for complex data.

Functional-Gradient Boosting (FGB) is an iterative procedure where the mistakes committed in the previous step are "fixed" in the next step. For every positive example (i.e., a true ADE), indicator value (say $I_i$ for the current ADE $i$) is set to 1 else it is set to 0. Now, the probability of every ADE being a true ADE (i.e., $P(ADE_i = 1) = P_i$ for the current ADE $i$) is computed given the current model. The difference between the indicator value and the computed probability (i.e., $I_i - P_i$) is computed for each ADE and this term becomes the *weight* of the ADE. We refer to the prior work [34] for details of the derivation.

Intuitively, the weight reflects the error made by the current model for each example. If the example is a positive ADE, then the model should predict this as positive with a probability of 1. The difference between 1 and the current predicted probability is the *magnitude* of the error and becomes its weight. If the example is a negative example, the model should predict this as positive with probability 0 and the difference (negative number) is the magnitude of the error of the negative examples. Hence, the positive example weights are always $\geq 0$ and negative example weights are always $\leq 0$. Once the weights of the examples are set, this method learns more clauses that focus on the higher weighted examples i.e., examples that have higher errors in the previous step. Then these clauses are added to the earlier set of clauses, new predictions are made, new weights

[h]

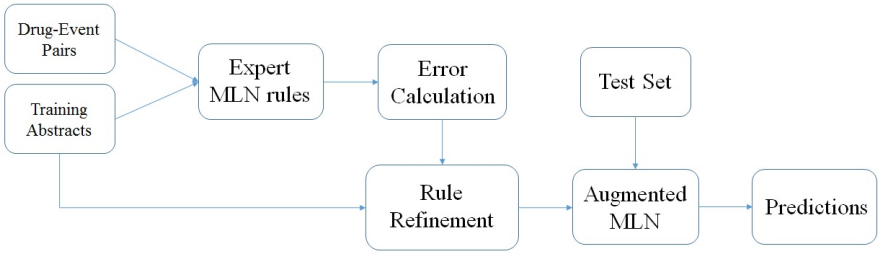



**Fig. 5.** Flowchart of the refinement approach. First the rules of the expert are evaluated on the training set to identify potential errors. These errors are then fixed using an ensemble method that learns from textual data.

are computed and the process is repeated. Thus, this method simply pushes all positive examples towards probability of 1 and negatives towards 0 at each iteration.

So the next question is: *how can we employ this iterative procedure to improve our expert designed MLN?* We essentially make predictions using the current MLN that the expert provides. These predictions are used to compute the weights of the different ADEs at iteration 0. Then functional gradient boosting is applied as described above to learn more clauses in the next few iterations. We run the boosting algorithm for 5 more time steps since the initial MLN (as we show in the next section) exhibits quite reasonable performance. The hypothesis is that further refining the MLN (i.e., by adding more weighted clauses) can result in a more robust model that can improve upon the errors of the human expert. Note that this step requires learning as against our earlier step which only used the MLN for reasoning about every ADE. Subsequently, as we describe later, we need more examples than the original approach for improving the expert specified MLN. As with the earlier case, given a set of instantiations, this MLN is grounded to an MRF and then is used for ADE detection.

We call this approach as *refinement of MLNs* and present results for this approach in the second half of the next section.

## 4. Results

In this section, we present the results of empirically validating our proposed approaches by evaluating the proposed adverse drug events (ADEs). We aim to explicitly answer the following questions:

**Q1:** Is the use of expert MLNs necessary? Will string similarities suffice for medical abstracts?

**Q2:** Does the use of data improve the expert's knowledge?

### 4.1. Evaluation of the basic approach

We evaluate our approach on the OMOP ground truth that can be obtained at OMOPs website [6] . OMOP provided 9 ADE pairs, which are composed of widely-used drug classes and health outcomes of interest (HOIs). We referred to these HOIs as effects in our earlier discussion. These ADE pairs were classified by OMOP as positive risks. OMOP also provided 44 negative control ADE pairs. When evaluating the algorithm, we simply queried for the probability that the given HOI is actually an ADE of the given drug. While plotting the area under the curve of the ROC curve (AUCROC), we used the ground truth values.

For each ADE, we extracted 50 MEDLINE abstracts by querying PubMed[7]. Our experiments showed that when considering different number of documents - $10, 25, 50, 75, 100$, the results improved till 50 documents. Beyond 50, there were no significant improvements. From these abstracts, we identified the sentences that contain both the drug class and the HOI, resulting in a total of 2140 sentences. We ran these sentences through the Stanford NLP toolkit to create relational linguistic features lexical, syntactic and semantic features, which were used as evidence. We also stored the drug classes and HOIs, as well as their relationships with ADE pairs, to be used as evidence when querying the MLN.

We compared the performance of different MLN rule types in this domain. We compared five different set of MLN rules to evaluate the importance of each rule type. In the first setting, we used the full relation extraction knowledge to evaluate the proposed ADEs (full MLN). In the second setting, we only used the string similarity rules (Sim). In the third setting, we used just the basic dependency rules to evaluate their contribution (Basic). In the fourth setting, we used the full extraction knowledge except the string similarities (i.e. Basic + Prep). In the fifth setting, we used just the basic dependencies with similarity rules to evaluate the importance of prepositional features (Sim + Basic). Since the prepositional rules inherently depend on the basic dependency rules, we do not evaluate on using prepositional rules without the dependency rules.

We use Receiver Operating Characteristic (ROC) and Precision-Recall (PR) curves to perform performance evaluation. In all settings, we performed ten runs, and averaged the area under the ROC curve (AUC ROC) and PR curve (AUC PR). Since we employ an approximate inference technique for obtaining the distribution over the drug condition pairs, we repeat the experiment multiple times.

As shown in Table 2, using all the rules in the MLN performs the best with AUC ROC of 0.83 and AUC PR of 0.68. It can be noted that adding the similarity metrics to the MLN (Basic+Prep) is not improving the performance significantly. This shows that our method is capable of going beyond simple mentions of the drug, condition pairs in the text. Just using the similarity rules also performs reasonably well as it removes all the negative ADEs that are never even mentioned together. As can be seen, the most effective method is the one that uses all the different clauses and the similarity rules as well and has statistically significant difference in the area under PR curves. The PR curves are considered to be a conservative estimate over ROC curves and hence are considered as more

[6] http://omop.fnih.org/sites/default/files/ground%20truth.pdf
[7] If there are less than 50 abstracts for a particular ADE pair, we use only the returned set of documents

[h]

| Rule Type | AUC ROC | | AUC PR | |
|---|---|---|---|---|
| | Mean | Variance | Mean | Variance |
| Sim | 0.69 | 0.0020 | 0.47 | 0.0073 |
| Basic | 0.73 | 0.0010 | 0.49 | 0.0010 |
| Sim + Basic | 0.80 | 0.0017 | 0.57 | 0.0013 |
| Basic + Prep | 0.83 | 0.0006 | 0.68 | 0.0008 |
| Full MLN | 0.83 | 0.0005 | 0.68 | 0.0010 |

**Table 2.** AUC ROC and AUC PR values for five MLNs averaged over ten runs. For each run, we used 50 abstracts for each drug condition pair.

[h]

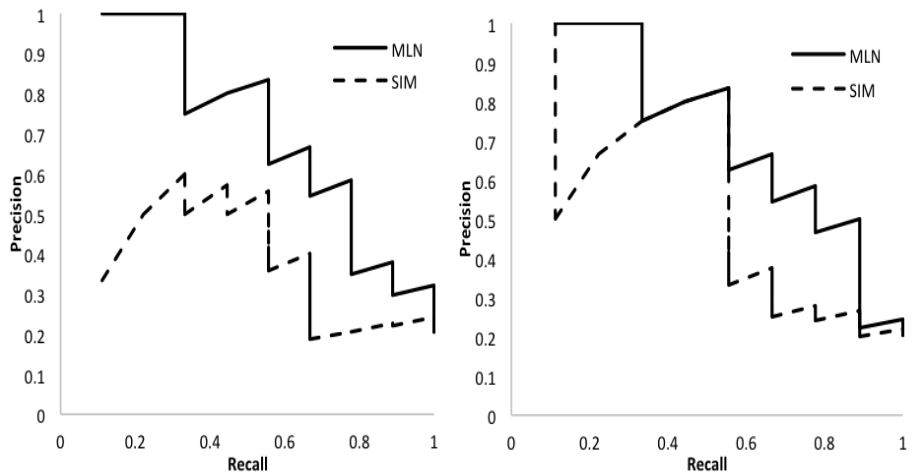



**Fig. 6.** Sample Precision-Recall curves. We compare the results between SIM(String Similarity) and with the expert designed MLN.

rigorous estimators. Under this estimation, the use of the entire MLN yields far superior results than any other combination.

We present two sample precision-recall curves from two of the runs in Figure 6. The dashed line represents the use of only similarity measures while the other line is the full MLN. The shapes of the curves are very similar in most of the runs. The key observation is that the use of the entire MLN helps to identify the more complex negatives. For instance, if a drug condition pair is mentioned in a sentence that uses complex word formations to explain the negative correlation between them, simple similarity measures will not suffice while the full MLN can possibly identify this as negatives. We discuss this in greater detail in the next section.

We can answer Q1 affirmatively that the use of expert's knowledge that encodes MRFs as MLNs improves upon the use of simple string similarity metrics.

In addition, we also performed another experiment to understand the importance of the number of articles. The key question that we aimed to understand was: do a small number of strongly relevant articles exhibit higher performance or is their support further reinforced by a few more weakly supportive documents. To this we extracted 20, 30, 50 and 100 articles for each ADE pair. We

[h]

| Num of Abstracts | AUCROC | AUCPR |
|---|---|---|
| 20 | 0.73 ± 0.001 | 0.43 ± 0.001 |
| 30 | 0.79 ± 0.001 | 0.48 ± 0.001 |
| 50 | 0.83 ± 0.00 | 0.68 ± 0.001 |
| 100 | 0.75 ± 0.00 | 0.29 ± 0.001 |

**Table 3.** AUC ROC and AUC PR values for different number of abstracts across 10 runs.

ensured that the 20 documents is a subset of the 30 documents which is a subset of the 50 which in turn is a subset of 100. We performed 10 different runs and averaged the results over these 10 runs. As can be seen, the method exhibits the best performance using 50 documents which appears to be a sweet spot for the number of articles. Using 100 documents introduces more noise and hence the performance decreases drastically. Using lower number of documents do not provide sufficient evidence to obtain a useful performance. Hence, we have chosen 50 documents for evaluation.

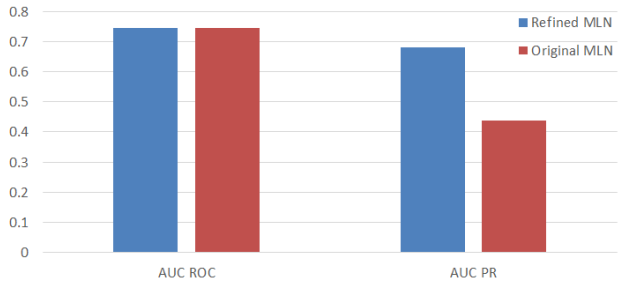
It must be mentioned that since we are simply performing inference using the documents, our relation extraction method is quite effective. When using the OMOP ADE definitions and 50 documents per ADE-pair, on a quad-core machine, the inference process was completed in under 30 minutes. This is because of the fact that the rules are essentially horn clauses (of the form *if then*) and the fact that the if part is observed, probabilistic inference is efficient.

## 4.2. Evaluation of the refinement approach

Note that the previous experiment evaluated whether the MLN was useful in identifying OMOP specified ADEs from text. While the results showed improvement over standard string based methods, they can still be improved. As mentioned earlier, we used refinement of MLNs to improve upon the MLN created earlier. A key issue is that since we are *learning*, we require more examples than the 9 positive ADE pairs from OMOP data set. To this effect, we used 30 more ADE pairs from literature (PubMed) as positive examples. 60 negative examples were created randomly using these 30 drugs and event pairs. These were then used as the training set along with the 9 OMOP ADEs as input to the refinement algorithm and performed 5-fold cross validation. It must be mentioned that the 2011-2012 version of OMOP is much more restricted than the original data-set and has only 4 categories of drugs. It did not provide any more information than what we have already, i.e., there is no statistical significance in adding these definitions to our enhanced data set. When refining, we learn 10 trees for the refinement. Using beyond 10 trees did not significantly improve the results and we restricted ourselves to 10 trees.

Table 5 lists all the positive cases that we have considered in the current work. We constructed the negative controls from the positives by considering all possible drug-disease combinations and removing the positive pairs from that list of combinations. This is called closed-world assumption, which means whatever that is not observed is false. This is a standard assumption in many machine learning/Artificial Intelligence algorithms and we employ the same assumption here.

The results are presented in Figure 7 using AUC-ROC and AUC-PR values for the two algorithms - refined MLN and the original MLN. As can be seen from




[h]

**Fig. 7.** Results of using the refinement algorithm for MLNs on 39 ADE pairs using 5-fold cross validation.

[h]

| Drug | Adverse event |
|---|---|
| ADE Inhibitor | Angioedema |
| Amphotericin B | Acute Renal Failure |
| Anaesthesia | Headache |
| Antibiotic | Acute Liver Failure |
| Antibiotic | Deafness |
| Antidepressant | Erectile Dysfunction |
| Antiepileptic | Aplastic Anemia |
| Antihistamine | Drowsiness |
| Antipsychotic | Myocardial Infarction |
| Antipsychotic | Diabetes |
| Aspirin | Intenstine Bleeding |
| Benzodiazepine | Hip Fracture |
| Benzodiazepine | Seizures |
| Bisphosphonate | Upper GI Ulcer |
| Chemotherapy | Anemia |
| Chemotherapy | Hairloss |
| Contraceptive | Melasma |
| Contraceptive | Thrombosis |
| Corticosteroid | Glaucoma |
| Corticosteroid | Mania |
| Ephedrine | Hypertension |
| Fluoxetine | Suicide |
| Interferon | Depression |
| Interferon | Hepatic Injury |
| Metformin | Lactic Acidosis |
| Methylphenidate | Insomnia |
| Metoclopramide | Tardive Dyskinesia |
| Misoprostol | Uterine Hemorrhage |
| Orlistat | Diarrhea |
| Paracetamol | Liver Damage |
| Propofol | Death |
| Sildenafil | Heart Attack |
| Sildenafil | Priapism |
| Statins | Rhabdomyolysis |
| Stavudine | Lactic Acidosis |
| Tricyclic Antidepressant | Acute Myocardial Infarction |
| Vaccination | Fever |
| Warfarin | Bleeding |

**Table 4.** List of all the positive ADE pairs.

[h]

| Category | Adverse drug event | Probability |
|---|---|---|
| Positive OMOP ADE pairs | ACE Inhibitor causes Angioedema | 1.000 |
| | Benzodiazepines cause Hip Fracture | 0.997 |
| | Amphotericin B causes Acute Renal Failure | 0.986 |
| Negative OMOP ADE pairs | ACE Inhibitor causes Aplastic Anemia | 0.624 |
| | Typical Antipsychotic causes Upper GI Ulcer | 0.626 |
| | Warfarin causes Aplastic Anemia | 0.617 |
| Negative OMOP ADE pairs but positive NLP ADE pairs | Bisphosphonates cause Acute Renal Failure | 0.998 |
| | Antibiotics cause Bleeding | 0.991 |
| | Warfarin causes Acute Renal Failure | 0.965 |

**Table 5.** Examples of ADE pairs.

the figure, the use of data on top of the expert knowledge provides significantly better results on the cross-validated ADE pairs. The improvement in PR is significant (around 50% with the PR). This initially answers Q2 in that data can help improve upon the expert knowledge. We employ AUC-PR for preliminary analysis as it has been shown to be a more conservative estimate of the learning performance compared to AUC-ROC [35]. Further experimental evidence is necessary and this is an important direction that we will pursue in the future.

## 5. Discussion

The results on OMOP data show that the system performs significantly better than chance, and compares very well with systems designed to extract ADE information from EHRs (for instance, see Ryan et al. [14]). While string similarities can be used to remove most of the negative ADEs, the use of text patterns and semantic understanding improves the accuracy further.

When considering string similarity only, we observed that several false positive ADEs have high string similarities with literature found on the web. Several of these are even higher than similarities of positive ADEs. Note that the string similarities are simply computing the frequencies that the pair has been mentioned. In some cases, while the number of times the given DE pair is mentioned could be high, these were essentially negative ADE mentions. The similarity metric ignores phrases such as negative, not an effect, no association, etc. Using text patterns on the other hand, we were able to make a better evaluation of the proposed ADEs since they consider the type of the mention as positive or negative, resulting in a performance improvement.

In Table 5, we show some examples of ADE pairs found in the MLN setting in three categories: true positives, i.e., OMOP pairs that we also found to be positive (where the probability of the event being an ADE is high), true negatives, i.e., negative OMOP pairs that we found to be negative (where the probability of the event being an ADE is low), and false negatives, negative OMOP pairs that we found to be positive (where the probability of the event being an ADE is high).

As expected, some of our results agree with the OMOP ground truth (the top two sets of rows in the table). Note that some of our results have no perfect agreement with OMOP ground truth. This means that some of the negative control ADEs given by OMOP are actually found to be positive by our method.

This disagreement reveals some probable directions of investigation for OMOPs ground truth. For instance, consider the ADE

`Bisphosphonate causes Acute Renal Failure.`

This ADE is classified as negative control by OMOPs ground truth. However, it received a high score in our method. When looking closely at the sentences related to this ADE, we found that there is text to support the fact that the ADE is a positive risk, which may contradict OMOPs ground truth. An example sentence that we found from PubMed article (PMID 11887832) is:

`Bisphosphonates have several important toxicities: acute renal failure, worsening renal function, reduced bone mineralization, and osteomalacia.`

This may happen because of several reasons, such as OMOPs high standard of evidence for ADEs or discoveries occurring after OMOP initiation. Results like this show that our method can be used for ADE evaluation of the ground truth. More importantly, given that the literature is vast, we can find with less human effort ADEs that are already known or have been discovered previously.

A current limitation of our approach is that although it finds evidence for ADEs that were not in the OMOP ground truth (such as a link between bisphosphonates and acute renal failure and a link between antibiotics and increased risk of bleeding with warfarin use) it also falsely interprets some other relationships. For example, it falsely assigns hip fracture as a warfarin ADE on the basis of sentences such as this one from a PubMed Central article (PMC3195383)

`There is a need for a national policy for reversing warfarin anticoagulation in patients with hip fractures requiring surgery.`

Another error occurs when our approach falsely interprets evidence for a protective effect as evidence for an ADE, interpreting PubMed Central article with PMID 11826008 as providing evidence that amphotericin B might cause aplastic anemia. Of the ten highest-ranked false ADEs by our method from OMOPs ground truth this is the lowest ranked.

`We describe a case of primary cutaneous mucormycosis (zygomycosis) in a patient with idiopathic aplastic anemia which responded to surgical debridement and therapy with liposomal amphotericin B.`

Other disagreements with OMOP ground truth among the top ten were actual positive evidence for ADEs but with weak evidence in the form of single cases or animal studies.

Our primary goal in this work is to develop a nimble, general tool for evaluating a wide variety of ADE discovery methods that might be based on search engine queries, social network data, or observational medical data such as health insurance claims or electronic health records. It is possible that the best approach will be an ensemble of all of these, and might itself include our scientific literature-based approach as well. Nevertheless, we see the primary role of this literature-based approach as being for evaluation, since we expect results confirmed and published in the scientific literature to necessarily lag behind the initial signals of an ADE likely to appear in EHRs and claims data, in internet searches, and in social media.

## 6. Conclusion

We present a novel approach for extracting adverse drug events (ADEs), a major social concern that accounts for 770 000 injuries and deaths each year [36], from text. Our method exploits publicly available biomedical literature to estimate the probability that a drug may cause a certain event. We do so by using state-of-the-art text mining and multi-relational machine learning techniques. We evaluate our performance on the reference OMOP ground truth, find agreement better than state-of-the-art ADE discovery methods, and find that in some of the cases of disagreement our method appears to be correct. Nevertheless, we find that in an equal number of cases our method is incorrect. In the remaining cases of disagreement our method has only weak evidence in support of its findings. We expect these weaknesses in our method can be addressed in part by further improvements in its natural language processing and in part by performing parameter learning in its Markov logic network.

## References

[1] L. Pray, S. Robinson, Enhancing postmarket safety monitoring. Challenges for the FDA: The Future of Drug Safety, Workshop Summary, The National Academies Press, 2007.
[2] J. L. Oliveira, P. Lopes, T. Nunes, D. Campos, S. Boyer, E. Ahlberg, E. Mulligen, J. Kors, B. Singh, L. Furlong, et al., The EU-ADR Web Platform: delivering advanced pharmacovigilance tools, Pharmacoepidemiology and drug safety 22 (5) (2013) 459–467.
[3] A. PS, C. Z, C. C. ad Tai BC, Data mining spontaneous adverse drug event reports for safety signals in singapore - a comparison of three different disproportionality measures., Expert Opin Drug Saf.
[4] N. D, K. Y, T. S, Y. H, Adverse events associated with incretin-based drugs in japanese spontaneous reports: a mixed effects logistic regression model, Peer J.
[5] T. J, L. RJ, Time-to-event analysis., JAMA.
[6] I. H, S. A, A. A, S. E. A, Mining association patterns of drug-interactions using post marketing fda's spontaneous reporting data, J Biomed Inform.
[7] A. Baldini, M. Von Korff, E. H. Lin, A review of potential adverse effects of long-term opioid therapy: a practitioners guide, The primary care companion to CNS disorders 14 (3).
[8] L. Manchikanti, S. Abdi, S. Atluri, C. Balog, R. Benyamin, M. Boswell, et al., American society of interventional pain physicians (asipp) guidelines for responsible opioid prescribing in chronic non-cancer pain: Part i–evidence assessment., Pain Physician 15 (3 Suppl) (2012) S1–65.
[9] Va/dod clinical practice guideline for management of opioid therapy for long-term pain, d.o.d, Department of Veterans Affairs.
[10] M. Kahan, L. Wilson, A. Mailis-Gagnon, A. Srivastava, Canadian guideline for safe and effective use of opioids for chronic noncancer pain clinical summary for family physicians. part 2: special populations, Canadian Family Physician 57 (11) (2011) 1269–1276.
[11] H. Poon, P. Domingos, Unsupervised semantic parsing, in: Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing, EMNLP 2009, 6-7 August 2009, Singapore, A meeting of SIGDAT, a Special Interest Group of the ACL, 2009, pp. 1–10.
[12] P. Domingos, D. Lowd, Markov logic: An interface layer for artificial intelligence, Synthesis Lectures on Artificial Intelligence and Machine Learning 3 (1) (2009) 1–155.
[13] P. Ryan, E. Welebob, A. G. Hartzema, P. Stang, J. M. Overhage, Surveying us observational data sources and characteristics for drug safety needs, Pharmaceutical Medicine (2010) 231–238.
[14] P. Ryan, D. Madigan, P. Stang, J. Overhage, J. Racoosin, A. Hartzema, Empirical assessment of methods for risk identification in healthcare data: results from the experiments of the observational medical outcomes partnership, Statistics in Medicine 31 (30) (2012) 4401–15.
[15] R. Navigli, P. Velardi, S. Faralli, A graph-based algorithm for inducing lexical taxonomies

from scratch, in: Proceedings of the Twenty-Second International Joint Conference on Artificial Intelligence - Volume Volume Three, IJCAI'11, AAAI Press, 2011, pp. 1872–1877. doi:10.5591/978-1-57735-516-8/IJCAI11-313.
URL http://dx.doi.org/10.5591/978-1-57735-516-8/IJCAI11-313

[16] G. Boella, L. D. Caro, A. Ruggeri, L. Robaldo, Learning from syntax generalizations for automatic semantic annotation, J. Intell. Inf. Syst. 43 (2) (2014) 231–246. doi:10.1007/s10844-014-0320-9.
URL http://dx.doi.org/10.1007/s10844-014-0320-9

[17] R. J. Mooney, R. Bunescu, Mining knowledge from text using information extraction, SIGKDD Explor. Newsl. 7 (1) (2005) 3–10. doi:10.1145/1089815.1089817.
URL http://doi.acm.org/10.1145/1089815.1089817

[18] M. Mintz, S. Bills, R. Snow, D. Jurafsky, Distant supervision for relation extraction without labeled data, in: Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP: Volume 2 - Volume 2, ACL '09, Association for Computational Linguistics, Stroudsburg, PA, USA, 2009, pp. 1003–1011.
URL http://dl.acm.org/citation.cfm?id=1690219.1690287

[19] H. Gurulingappa, J. Fluck, M. HofmannApitius, T. L., Identification of adverse drug event assertive sentences in medical case reports, in: First International Workshop on Knowledge Discovery in Health Care and Medicine, 2011.

[20] C. Friedman, Discovering novel adverse drug events using natural language processing and mining of the electronic health record, in: Proceedings of the 12th Conference on Artificial Intelligence in Medicine: Artificial Intelligence in Medicine, AIME '09, 2009, pp. 1–5.

[21] K. Shetty, S. Dalal, Using information mining of the medical literature to improve drug safety, JAMIA 18 (5) (2011) 668–674.

[22] J. Bian, U. Topaloglu, F. Yu, Towards large-scale twitter mining for drug-related adverse events, in: Proceedings of the 2012 International Workshop on Smart Health and Wellbeing, 2012, pp. 25–32.

[23] J. Lafferty, A. McCallum, F. Pereira, Conditional random fields: Probabilistic models for segmenting and labeling sequence data.

[24] F. Niu, C. Ré, A. Doan, J. Shavlik, Tuffy: Scaling up statistical inference in markov logic networks using an rdbms, Proceedings of the VLDB Endowment 4 (6) (2011) 373–384.

[25] S. Riedel, H. Chun, T. Takagi, J. Tsujii, A markov logic approach to bio-molecular event extraction, in: Proceedings of the Workshop on Current Trends in Biomedical Natural Language Processing: Shared Task, BioNLP '09, Association for Computational Linguistics, 2009, pp. 41–49.

[26] H. Poon, L. Vanderwende, Joint inference for knowledge extraction from biomedical literature, in: Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics, HLT '10, Association for Computational Linguistics, 2010, pp. 813–821.

[27] S. Riedel, A. McCallum, Robust biomedical event extraction with dual decomposition and minimal domain adaptation, in: Proceedings of the BioNLP Shared Task 2011 Workshop, BioNLP Shared Task '11, Association for Computational Linguistics, 2011, pp. 46–50.

[28] S. Riedel, D. McClosky, M. Surdeanu, A. McCallum, C. Manning, Model combination for event extraction in bionlp 2011, in: Proceedings of the BioNLP Shared Task 2011 Workshop, BioNLP Shared Task '11, Association for Computational Linguistics, 2011, pp. 51–55.

[29] C. T. Bergstrom, J. D. West, M. A. Wiseman, The eigenfactor metrics, The Journal of Neuroscience 28 (45) (2008) 11433–11434.

[30] R. Clayton, Calculating similarity (part 1): Cosine similarity [internet] (Dec. 2010).

[31] J. Finkel, T. Grenager, C. Manning, Incorporating non-local information into information extraction systems by gibbs sampling, in: Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics, ACL '05, Association for Computational Linguistics, 2005, pp. 363–370.

[32] D. Klein, C. Manning, Accurate unlexicalized parsing, in: Proceedings of the 41st Annual Meeting on Association for Computational Linguistics-Volume 1, Association for Computational Linguistics, 2003, pp. 423–430.

[33] T. Khot, S. Natarajan, K. Kersting, J. Shavlik, Learning markov logic networks via functional gradient boosting, in: International Conference in Data Mining, 2011.

[34] S. Natarajan, T. Khot, K. Kersting, B. Gutmann, J. Shavlik, Gradient-based boosting for statistical relational learning: The relational dependency network case, Invited contribution to special issue of Machine Learning Journal 86 (1) (2012) 25–56.

[35] J. Davis, M. Goadrich, The relationship between precision-recall and ROC curves, in: ICML, 2006.
[36] N. Tatonetti, G. Fernald, R. B. Altman, A novel signal detection algorithm for identifying hidden drug-drug interactions in adverse event reports, JAMIA 19 (1) (2012) 79–85.

| Predicates |
|---|
| adverse(rule) |
| rule(rule) |
| dpBetweenDrugAndEffect(rule) |
| dpH(rule, wo, subwostr, wordType) |
| dpRCount(rule, dp1, n, n) |
| dpWhileReceivingCount(rule, dp2, n, n) |
| dpWhileTakingCount(rule, dp3, n, n) |
| dpDirectCause(rule, dp4, dp44, n, n) |
| dpDirectIncrease(rule, dp5, dp55, n, n) |
| dpRisk(rule, dp6, dp66, n, n) |
| dpRiskAssociated(rule, dp7, dp77, n, n) |
| dpAssociated(rule, dp8, dp88, n, n) |
| dpConsequence(rule, dp9, n, n) |
| dpSideWithEffect(rule, dp11, dp11, n, n) |
| dpProduce(rule, dp12, dp12, n, n) |
| dpPromote(rule, dp12, dp12, n, n) |
| adverseC(rule, n, n) |
| cosineSimilarityWeight(rule, float_ wgt) |

[!b]

**Table 6.** List of predicates in MLN

# 7. Appendix

| Rules |
|---|
| // effect after drug<br>dpRCount(r, dp, up, down), [contains(dp, "PREP_AFTER") OR<br>contains(dp, "PREP_FOLLOWING")] =>adverseC(r, up, down). |
| // effect on xxxx following drug<br>dpRCount(r, dp, up, down), [contains(dp, "PREP_ON") AND<br>contains(dp, "PREP_FOLLOWING")] =>adverseC(r, up, down). |
| // effect prep_while <receiving—taking>drug<br>dpWhileReceivingCount(r, dp, up, down), [contains(dp, "PREPC_WHILE")]<br>=>adverseC(r, up, down).<br>dpWhileTakingCount(r, dp, up, down), [contains(dp, "PREPC_WHILE")]<br>=>adverseC(r, up, down). |
| // drug nsubj <cause—increase>dobj effect<br>dpDirectCause(r, dp1, dp2, up, down), [contains(dp1, "NSUBJ") AND<br>contains(dp2, "DOBJ")] =>adverseC(r, up, down).<br>dpDirectIncrease(r, dp1, dp2, up, down), [contains(dp1, "NSUBJ") AND<br>contains(dp2, "PREP_OF")] =>adverseC(r, up, down). |
| // risk prep_of effect partmod drug<br>dpRisk(r, dp1, dp2, up, down), [contains(dp1, "PREP_OF") AND<br>contains(dp2, "PARTMOD")] =>adverseC(r, up, down). |
| // risk <prep_of—prep_for>effect associated prep_with drug<br>dpRiskAssociated(r, dp1, dp2, up, down), [(contains(dp1, "PREP_OF") OR<br>contains(dp1, "PREP_FOR")) AND contains(dp2, "PREP_WITH")]<br>=>adverseC(r, up, down). |
| // drug nsubjpass associated prep_with effect<br>dpAssociated(r, dp1, dp2, up, down), [contains(dp1, "NSUBJPASS") AND<br>contains(dp2, "PREP_WITH")] =>adverseC(r, up, down). |
| // effect consequence prep_of drug<br>dpConsequence(r, dp, up, down), [contains(dp, "PREP_OF")]<br>=>adverseC(r, up, down). |
| //effect side effect of drug<br>dpSideWithEffect(r, dp1, dp2, up, down), [contains(dp1, "PREP_OF")]<br>=>adverseC(r, up, down). |
| //drug promotes effect<br>dpPromote(r, dp1, dp2, up, down) =>adverseC(r, up, down). |
| //drug produced effect<br>dpProduce(r, dp1, dp2, up, down) =>adverseC(r, up, down). |

**Table 7.** Rules to deduce adverseC predicates, which subsequently influence the posterior probability of the adverse predicate.

| Weight | Rules |
|---|---|
| wgt | cosineSimilarityWeight(r, wgt) =>adverse(r) |
| 1 | dpBetweenDrugAndEffect(r) =>adverse(r) |
| 3 | dpH(r, wo, str, str1), [(str = "induced" OR str = "associated") AND (contains(str1, "AMOD"))] =>adverse(r) |
| 3 | adverseC(r, up, down), [up+down = 1] =>adverse(r) |
| 2.5 | adverseC(r, up, down), [up+down = 2] =>adverse(r) |
| 2 | adverseC(r, up, down), [up+down = 3] =>adverse(r) |
| 1.5 | adverseC(r, up, down), [up+down >= 4] =>adverse(r) |
| -0.5 | !adverse(r) |

**Table 8.** Final MLN Rules