AMIA Annual Symposium Proceedings Archive

AMIA Annu Symp Proc. 2011; 2011: 1019–1026.
Published online 2011 Oct 22.

Pattern Mining for Extraction of mentions of Adverse Drug Reactions from User Comments

Azadeh Nikfarjam, MS[1] and Graciela H. Gonzalez, PhD[1]

Abstract

Rapid growth of online health social networks has enabled patients to communicate more easily with each other. This way of exchange of opinions and experiences has provided a rich source of information about drugs and their effectiveness and more importantly, their possible adverse reactions. We developed a system to automatically extract mentions of Adverse Drug Reactions (ADRs) from user reviews about drugs in social network websites by mining a set of language patterns. The system applied association rule mining on a set of annotated comments to extract the underlying patterns of colloquial expressions about adverse effects. The patterns were tested on a set of unseen comments to evaluate their performance. We reached to precision of 70.01% and recall of 66.32% and F-measure of 67.96%.

Introduction

It is well-known that adverse reactions to medications present a significant health problem. According to a systematic review of twenty-five prospective observational studies including 106,586 patients who were hospitalized, approximately 5.3% of all hospital admissions are associated with adverse drug reactions, with higher rates (a median of 10.7%) reported for elderly patients[1]. An earlier review of 39 prospective studies had similar findings, calculating the overall incidence of serious ADRs at 6.7% of all hospital admissions, with fatal ADRs at 0.32% [2]. Based on their results and the number of admissions, the later study estimated that there were 2 216 000 patients hospitalized with serious ADRs and 106 000 had fatal ADRs in 1994, placing adverse drug reactions between the fourth and sixth leading cause of death for that year. The economic impact of ADRs is also important: approximately $136 billion is spent annually on treating ADRs in the U.S., with other nations facing similar difficulties [3,4].

Some of the drug adverse effects are discovered during Phase III trials, however, there are some that are only revealed after a long time use or at the end of treatment. New ADRs also appear when the drug is used by groups of patients not included in the trial (for example, patients with chronic diseases). Uncovering adverse effects is often a difficult and lengthy process that could greatly reduce the impact that ADRs have on human health if it could be accelerated. There have been

some attempts at mining information about drug effects from unstructured text in clinical records. However, it is often difficult to access such records due to privacy concerns. There exists an unexplored, yet valuable resource that is openly available: the comments that patients themselves make about their experience with the drugs on health-related social network websites such as DailyStrength[1].

Extracting ADRs from text using natural language processing has been studied in recent years, but mainly in the context of electronic health records (EHRs) [5–7]. Aramaki et al.[5] reported 7.7% of electronic records in their study included ADRs, with their automatic extraction system detecting 59% of them. The authors applied MedLEE to encode and extract entities from narrative discharge summaries and used co-occurrence statistics to extract relations between drugs and adverse effects[7]. Text mining to discover knowledge from colloquial text in the context of social networks has focused on product reviews, but, to the best of our knowledge, its application in the context of health information, including drug adverse effects, has not been studied. Most of the approaches try to detect whether a comment is positive or negative regarding a product or service rather than extracting specific concepts[8,9]. Turney[10] applied sentiment analysis on people's phrases to classify film reviews. The method is based on the notion that people express their feelings about objects by using "sentiment" words (those that convey positive or negative feelings about things, such as "good", "excellent" or "terrible"). Hu and Liu [11,12], who are considered pioneers in extracting patterns from colloquial text, proposed a method called "feature-based opinion mining" which applied association rule mining to extract comments about product features (such as quality, size, and usability) in websites such as Amazon.com.

In previous work [13], we showed that user comments from health-related social networking sites can indeed reflect known ADRs and potentially serve as an early warning system about unknown ADRs. Our method to automatically extract ADRs from user comments (with 78.3% precision, 69.9% recall and 73.9 f-measure) was primarily based on inexact matching using an augmented lexicon of adverse drug reactions. In this paper we propose a new method to automatically extracting ADRs from user comments using natural language processing techniques that go beyond lexicon matching. We applied association rule mining, a supervised learning method, to extract mentions of ADRs in user reviews about drugs in health social networks. The hypothesis that drives our method is that even if the language they use is highly informal, people write their comments using some converging patterns that can be identified to facilitate the extraction of interesting pieces of information in those comments.

Methods

In next sections we describe the steps followed in our method to generate frequent patterns of language for expressing opinions about drugs using association rule mining.

The idea of association rule mining originated from the "shopping cart" problem, where the challenge is to identify which set of items are more likely to be bought together. Supermarkets use this information in positioning the items in the shelves and control the way customers traverse in the supermarket.

Explica que son las reglas de asociación

Association rules are represented as a set of expressions of the form $\{X_1,X_2,X_3,..X_n\} => Y$, which indicates that if we find $X_1,X_2,X_3,\ldots X_n$ in a shopping cart (a transaction), the probability of finding another product Y in that transaction will be high. This probability is called the *confidence* of the rule, and usually we seek the rules with confidence above a defined threshold. In addition, the number of transactions that include all the items $X_1,X_2,X_3,.. X_n$ and Y together is called 'support' of the rule. A *Frequent item set* is a set of items which have the support and confidence higher than a defined threshold. The Apriori algorithm is an influential algorithm in mining frequent association rules. It iteratively traverses transactions to find item sets with cardinality from 1 to K (K-items). The dominant rule behind Apriori method is that if S is a frequent item set, every subset of S should be frequent also. Once the frequent item set is found, it is used to generate the rules.

Mining patterns in text can be modeled as an association rule mining problem in which the sentences in text are the transactions and the words in the sentence are considered as items in the transactions.

Annotated Corpus

Dataset usado

We used DailyStrength, a health-related social network, as the source of user comments in this study. DailyStrength users can create profiles, communicate with other users, ask questions from people with similar conditions, and express their experiences in using various treatments. We used the same annotated data set as our previous research[13], which contains user comments relating to the following 4 drugs: carbamazepine, olanzapine, trazodone, and ziprasidone. These drugs were chosen since they are known to cause adverse reactions and the results could be verified with known adverse effects.

The corpus contains a total of 6,890 comment records, randomly selected for annotation and independently annotated by two experts. Each comment was annotated for mentions of adverse effects (a reaction to the drug experienced by the patient, which the user considered negative), beneficial effects (a reaction to the drug which was experienced by the user and considered positive), indications (the condition for which the patient is taking the drug) and other. Each annotation included the span of the mention and the name of the concept found, using entries from a lexicon we constructed by combining terms and concepts primarily from the COSTART vocabulary created by the U.S. Food and Drug Administration for postmarketing surveillance of adverse drug reactions (a subset of the UMLS Metathesaurus), and the Canada Drug Adverse Reaction Database, or MedEffect, among others. A total of 3,600 comments were annotated. The annotators found 1,260 mentions of adverse effects, 391 indications, 157 beneficial effects and 78 other, for a total of 1,886 mentions.

System Architecture

Figure 1 illustrates the general architecture of our system during the training phase. There are three main steps in generating patterns based on training data: 1. Term Sequence Generation, 2. Frequent Rule Identification and 3. Frequent Pattern Generation. We will detail each step in the following sub sections.

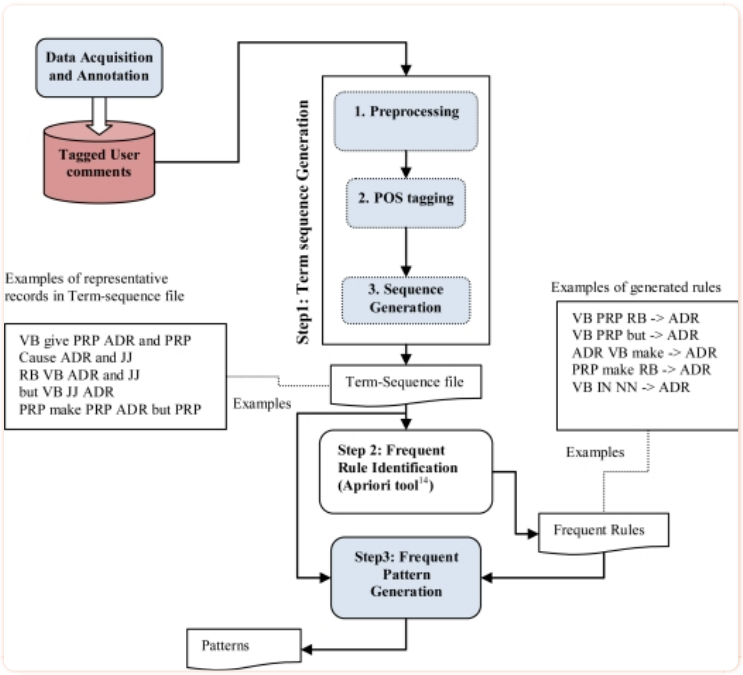

Figure 1.

Overall Architecture of the System

Step 1: Term Sequence Generation

This step creates a collection of sequences of words that appear in sentences where an adverse effect is mentioned and stores them in a "Term-Sequence" file. Each line of this file is thus representative of a sentence with an ADR mention. Either a generic POS (part of speech) or the original words in the sentence constitute the lines in the file and we refer to them as terms.

Preprocessing. We replace all the mentions of ADRs in user comments with the keyword "ADR" based on the annotator's tags. This keyword was used as a fixed term in all the representative sentences.

Identifying sentences can be a challenging task in colloquial text. Writers pay less attention to correctly using punctuation marks in informal context, making it more difficult to automatically split text to sentences. We split text at periods and exclamation marks and removed some unimportant marks such as quotations and parenthesis. Since we were interested in patterns of expression around mentions of ADRs, this allows us to just select from the corpus the sentences that included the "ADR" keyword. A subset of standard stop words (such as 'the' or 'this') are also removed from the sentences.

POS Tagging. Part of speech tagging performed on each of the sentences, using the Stanford parser[2]. Noun phrases also were identified and considered as a single term.

Sequence Generation. To generate representative sentences, a few terms around each side of the ADR mention were selected. We used part of speech as representatives for all words except for connectors (such as “and”, ”but”) and a few manually selected verbs such as (make, cause, give). In addition, we examined keeping verb lemmas instead of their POS tag, using the Wordnet[3] root of the verb.

When selecting the terms before and after the ADR, if we reach a comma, we stop adding terms in that direction. Each sentence representative is then inserted into the Term-Sequence file.

Step 2: Frequent Rule Identification

At this stage we are interested in extracting the rules that tell us which combination of terms are more likely to be present with a mention of an ADR. The Term-Sequence file generated in the previous step is the input to this stage. In order to generate the rules we applied Apriori tool[14] which is an implementation of the Apriori algorithm[15] for association rule mining. When running the association rule miner, the minimum and maximum support, the confidence, and the minimum number of items per rule can be set. We were interested in the rules that had “ADR” as the consequence, to enable us to conclude that: if we have a set of terms in the condition of the rule, it would be highly probable that a mention of an ADR is present in that sentence also (with the precise probability equal to the defined confidence). For example, a rule such as: “make PRP RB CC => ADR” means that the combination of a verb (“make”), a preposition (PRP), adverb (RB), and a connector(CC) often occurs with an ADR (in no particular order). This allows us to infer that if the combination in the condition is present in a sentence, a mention of an ADR is likely to be present in the sentence as well.

Step 3: Frequent Pattern Generation

The goal in this step is to create patterns from extracted rules that can be applied on new sentences to find mentions of adverse effects. Using the rules from the previous step we have the frequent terms sets, however the order of these terms should be identified to generate the patterns. We used the Term-Sequence file, which includes all the possible sequences, to find the orders. For each rule, the sequences from the Term-Sequence file that contained all the present terms in the rule were selected to build patterns from them using regular expressions. Therefore, a rule could have more than one corresponding pattern. Considering the order of the terms in the selected sequence, if the term was present in the rule it was added to the pattern, otherwise a placeholder was added which can accept any unseen term in the test sequence. Such placeholders that occurred at the beginning and the end of the pattern were removed from the pattern. The patterns with a placeholder right after the ADR keyword in the sequence and short patterns with no noun or verb phrase or adjective were removed from the system. ADR keyword in the sequence was replaced with a placeholder which catches the term(s) presented in that position in the test sequence. An example of the way that a rule is converted to some of possible patterns is presented in Table 1.

Table 1.

Converting a rule to possible patterns

Rule	Possible Sequence	Patterns
PRP and make -> ADR	make PRP ADR and	make PRP (.*) and
	and make PRP ADR	and make PRP (.*)
	make PRP JJ and ADR	make PRP (?:[^]*)* and (.*)
VB RB and -> ADR	VB ADR RB and	VB (.*) RB and
	RB ADR and VB	RB (.*) and VB

Among the original comments, there are some very short sentences (3 words or less) that end up being the single keyword “ADR” after preprocessing. Sometimes the whole sentence was tagged as ADR (E.g. “Very addictive.”). These are not incorporated in the generated rules (since our system settings require a minimum 4 number of terms per rule). We manually added a few patterns to handle short sentences. For example, if a sentence consists of just an adjective and noun or just a verb phrase, then the whole sentence is considered an ADR.

Testing Phase: Extracting ADRs from unseen comments

For each comment in our corpus we have a list of expected ADRs that were manually tagged by annotators and used as our gold standard for evaluation. Preprocessing is performed on test comments in the same way as for training (see Step 1, Term-Sequence generation): sentences are identified, the stop words are removed, and the POS tagger is run. A representative sequence is generated for each sentence in the comment. Note that for testing, we do not have an ADR keyword in the representative term sequence, and the goal is to extract ADR mentions by applying the patterns.

All the patterns generated in Step3 were applied on test sentences and the ADR mentions were found. The extracted ADRs from sentences of the comment were then compared with the gold standard ADRs for that comment. In some cases, there existed more than one pattern that matched the sentence. We selected a set of *unique* extracted ADRs for each comment. In post processing, extracted ADRs that did not contain an adjective or noun phrase or verb phrase were removed.

Evaluation

We used 1200 comments from our annotated corpus. While only the comments containing ADR mentions were used to generate the patterns in the training phase, the test set also contains comments that do not contain an ADR mention. We ran 10-fold cross validation to evaluate the performance of the method. At each run 90% of the comments used for the training and 10% for the testing. The system performance was measured using three standard measures: recall (R), precision

(P) and F-measure (F). In defining the true positive for the system we used an approximate matching strategy whereby each extracted ADR string is considered as true positive if it includes the corresponding ADR in the gold standard; for example if the system extracts “lots of weight gain” and the gold standard ADR is “weight gain”, then it is accepted as true positive. False positives are ADRs extracted by the system for which there is no match in the gold standard. The gold standard ADRs that cannot be matched with an extracted ADR constitute false negatives.

Results

We reached a precision of 70.01% with 66.32% recall for an f-measure of 67.96%. A combination of various tuning parameters in different steps of our approach affects the final result. Results on alternative options for the values of some of the parameters discussed in Step 1 (Term Sequence Generation) and Step 2 (Frequent Rule Identification) are summarized in Tables 3–5. For generating the Term-Sequence file in Step 1, the best result is achieved by setting both the maximum number of terms to the left and right of the ADR to 4. Also, simply using the POS of all the verbs except (make, cause, give) generates better results than using the lemma for all verbs (Table 4),

Table 3.

Results on different lengths for sentence representatives

Max# terms (left side of ADR)	**Max # terms (right side of ADR)**	**Precision (%)**	**Recall (%)**	**F-Measure (%)**
4	**4**	**70.01**	**66.32**	**67.96**
3	3	62.20	72.65	67.02

Table 4.

Results on alternative options for representing verbs in the Term-Sequence file

Verb representative	**Precision (%)**	**Recall (%)**	**F-Measure (%)**
POS	**70.01**	**66.32**	**67.96**
Lemma (Wordnet root)	74.2	56.31	64.2

Table 5.

Results on alternative options for association rule miner

Min Support	Max Support	Min# terms per rule	Precision (%)	Recall (%)	F-Measure (%)
4	-	4	64.10	71.24	67.26
6	-	4	66.70	69.36	67.80
4	7	4	67.86	68.0	67.89
4	**6**	**4**	**70.01**	**66.32**	**67.96**
4	7	3	63.2	71.9	67.26

Alternative options for generating frequent rules are presented in Table 5. Minimum and maximum support of the rule and the minimum number of items per rule affect the precision and recall in a reverse way; while changing the confidence does not change the result in our case. The best result is achieved by setting the min support to 4, max support to 6 and minimum number of terms per rule to 4. The number of generated rules and corresponding patterns are highly dependent to the size of the training data (number of sentences which have ADR mention) and the defined support and confidence of the rules; with our settings, at each run of cross validation average number of 40 frequent rules were extracted (Step 2) and average number of 300 patterns were generated based on the rules and term sequences (Step 3).

Discussion

The results of the evaluation of the rule based text mining approach presented here is highly dependent on the richness of the training data and whether the test data includes sentences matching the generated rules. Therefore, having a large corpus of training data significantly increases the performance.

Some parameters can be set in a way that leads to more restrictive rules and consequently results in higher precision and lower recall. For example considering representative records with longer length in the Term-sequence file (Table 3) increases precision and decreases the recall. In the same way, limiting the minimum and maximum support and increasing the minimum number of terms in generating rules at Step 2 (Table 5) decreases the recall and increases precision. This issue can be addressed in future through post processing; the recall can be increased by generating more general rules and allowing the post processing step to remove the false positives and increase the precision.

We performed an analysis to find the main sources of error in the system. We randomly selected 100 test comments and determined the reasons for 21 false positives (FPs) and 34 false negatives (FNs).

FNs happen in our system when the pattern of the target sentence has no matching pattern among the ones identified as frequent patterns. The largest number of FNs (55.8%) happened because of two main reasons. First, we had limited the parameters in the Frequent Rule Identification step (e.g. set the min support to 4% and maximum support to 6%) to prevent generating too many rules that could cause many FPs; as a result some of the effective rules and consequently their related patterns were not accepted. The second main reason for FNs is that some of the comments were written in a way which was not general enough to be matched with our patterns. In addition, 23.5% of FNs are due to the short sentence having words less than the minimum number of parameters present in a rule (which is set to 4). The limited number of patterns that we added manually for handling short sentences did not cover all the possible patterns that include an adverse effect. This problem can be solved by creating a training set of short sentences and running the system separately to create patterns for shorter sentences.

A number of FPs (67%) were phrases extracted by matching with an existing rule, but were not actually ADR mentions. Some of the errors in this group could be addressed by performing sentiment analysis. Often sentences that include mentions of ADRs contain sentiment bearing phrases that can be categorized as being pleasant or unpleasant or contain polar words such as (very, quite, awfully …). Therefore sentiment analysis can be performed as a post processing step to exclude extracted ADRs of non sentiment bearing sentences and improve the precision. 8% of FPs were due to grammatical errors in the comments, and another 8% were adverse effects that were not tagged by the annotators.

Conclusion and future work

We have presented a new method for using association rules for colloquial text mining. We applied our method on user comments to find mentions of adverse reactions to drugs by extracting frequent patterns. Since we are dealing with highly informal colloquial text, the idea of using extracted patterns might, at first, seem counter-intuitive. However, we indeed found consistencies in the user comments. Our evaluation measured the effectiveness of our technique in extracting frequent patterns in this context. However, our method can easily be generalized for other contexts and languages.

One of the advantages of this approach over a lexicon based approach is that it can detect expressions not included in the lexicon. Most new idiomatic expressions do not exist in standard medical terminologies, and even an augmented lexicon such as the one we developed will not include all of them; this issue as reported was the largest source of error in our previous method[13]. Another problem with the lexicon based method was the approximate string matching, which was not able to sufficiently correct for the misspelled words in user comments. The current pattern based system eliminates the need for string matching by looking for ADRs through templates rather than the exact mention; however this flexibility lowers the system precision. Although the precision and recall of our previous lexicon based method is currently higher than our new proposed method (8.29% in precision and 3.58% in recall), the nature of the source of the errors are different. We plan to combine these two methods by adding the results of a lexicon based method modified for high precision to our system and using the flexibility of the pattern based system to recover the recall. In addition, we expect a machine learning classifier that includes features related to the lexicon and the extracted patterns to be effective in ADR extraction.

Furthermore, a significant number of errors in the system could be addressed by using semantic analysis of the sentences by measuring the positivity and negativity of the sentences.

In health websites such as DailyStrength, sometimes more than 10,000 free text comments exist for just a single drug. Aside from detecting signals for ADRs, one of the important advantages of applying a pattern-based system such as the one proposed here is that it can also aid in creating structured summaries (for human consumption) of all adverse effects experienced and reported by patients during the lifetime of a drug. We plan to explore this direction in the future.

Table 2.

Example of test sentences that match with a given pattern

pattern	Example sentences	Extracted ADR
make PRP VB (.*)	It\'s okay-seems make me have really bad dreams	really bad dreams
	it made me lose weight	weigh loss
	Worked but it made me feel drugged	drugged
PRP (.*) VB RB	It worked but weight gain was not good for me	weight gain
PRP (.*) and VB	Puts me sleep and gets me out of MOMENT	sleep
cause PRP (.*)	It works, but swell too much and get heart murmurs	swell too much
	I think it actually causes me have more headaches	have more headaches

Acknowledgments

We would like to thank Robert Leaman for his support and helpful suggestions and Ehsan Emadzadeh for his helpful contributions during development and reviewing.

Footnotes

[1] http://www.dailystrength.org/

[2] http://nlp.stanford.edu/software/tagger.shtml

[3] http://wordnet.princeton.edu/

References