

Sentiment Analysis of User-Generated Content on Drug Review Websites

Jin-Cheon Na *

Wee Kim Wee School of Communication and Information, Nanyang Technological University
Singapore
E-mail: tjcna@ntu.edu.sg

Wai Yan Min Kyaing

CrimsonLogic Pte. Ltd., Singapore
E-mail: waiyanminkyaing@gmail.com

ABSTRACT

This study develops an effective method for sentiment analysis of user-generated content on drug review websites, which has not been investigated extensively compared to other general domains, such as product reviews. A clause-level sentiment analysis algorithm is developed since each sentence can contain multiple clauses discussing multiple aspects of a drug. The method adopts a pure linguistic approach of computing the sentiment orientation (positive, negative, or neutral) of a clause from the prior sentiment scores assigned to words, taking into consideration the grammatical relations and semantic annotation (such as disorder terms) of words in the clause. Experiment results with 2,700 clauses show the effectiveness of the proposed approach, and it performed significantly better than the baseline approaches using a machine learning approach. Various challenging issues were identified and discussed through error analysis. The application of the proposed sentiment analysis approach will be useful not only for patients, but also for drug makers and clinicians to obtain valuable summaries of public opinion. Since sentiment analysis is domain specific, domain knowledge in drug reviews is incorporated into the sentiment analysis algorithm to provide more accurate analysis. In particular, MetaMap is used to map various health and medical terms (such as disease and drug names) to semantic types in the Unified Medical Language System (UMLS) Semantic Network.

Keywords: Sentiment classification, drug reviews, linguistic approach, health and medical domains

Open Access

Accepted date: March 13, 2015
Received date: January 24, 2015

***Corresponding Author:** Jin-Cheon Na
Associate professor
Wee Kim Wee School of Communication and Information
Nanyang Technological University, Singapore
E-mail: tjcna@ntu.edu.sg

All JISTaP content is Open Access, meaning it is accessible online to everyone, without fee and authors' permission. All JISTaP content is published and distributed under the terms of the Creative Commons Attribution License (<http://creativecommons.org/licenses/by/3.0/>). Under this license, authors reserve the copyright for their content; however, they permit anyone to unrestrictedly use, distribute, and reproduce the content in any medium as far as the original authors and source are cited. For any reuse, redistribution, or reproduction of a work, users must clarify the license terms under which the work was produced.

1. INTRODUCTION

With the explosion of Web 2.0 platforms, there are enormous amounts of user-generated content. Weblogs, discussion forums, user review web sites, and social networking sites (e.g., Facebook and Twitter) are commonly used to express opinions about various subjects. Therefore, for the past decade many researchers have been studying effective algorithms for sentiment analysis of user-generated content (Liu, 2012). Sentiment analysis is a type of subjectivity analysis which analyzes sentiment in a given textual unit with the objective of understanding the sentiment polarities (i.e. positive, negative, or neutral) of the opinions regarding various aspects of a subject. It is still considered as a very challenging problem since user generated content is described in various and complex ways using natural language.

For sentiment analysis, most researchers have worked on general domains (such as electronic products, movies, and restaurants), but not extensively on health and medical domains. Previous studies have shown that this health-related user-generated content is useful from different points of view. First, users are often looking for stories from “patients like them” on the Internet, which they cannot always find among their friends and family (Sarasohn-Kahn, 2008). Moreover, studies investigating the impact of social media on patients have shown that for some diseases and health problems, online community support can have a positive effect (Jaloba, 2009; Schraefel et al., 2009). Because of its novelty as well as quality and trustworthiness issues, user-generated content of social media in health and medical domains is underexploited. It needs to be further studied, understood, and then leveraged in designing new online tools and applications.

For instance, when a new drug is released or used, users or patients publish their opinions about the drug on the social Web. The sentiment analysis results of drug reviews will be useful not only for patients to decide which drugs they should buy or take, but also for drug makers and clinicians to obtain valuable summaries of public opinion and feedback. Sentiment analysis can also highlight patients’ misconceptions and dissenting opinions about a drug. Therefore, the purpose of this paper is to develop an effective method for sentiment analysis of drug reviews on health infor-

mation service websites.

A clause-level sentiment analysis algorithm has been developed since each sentence can contain multiple clauses discussing multiple aspects, such as *overall opinion, effectiveness, side effects, condition, cost, and dosage*. Generally, each clause contains very few subjective terms, and therefore the method adopts a pure linguistic approach whereby a set of sentiment analysis rules are used to compute contextual sentiment scores by utilizing the grammatical relations, parts-of-speech (POS), and prior sentiment scores of terms in the clause. Since sentiment analysis is domain specific, domain knowledge on health and medical fields is very important to generate more accurate analysis. Therefore, MetaMap (Aronson & Lang, 2010) is used to map various health and medical terms (e.g., disease, symptom, and drug names) in the review documents to semantic types in the Unified Medical Language System (UMLS) Semantic Network, and the tagged semantic information is utilized for sentiment analysis.

We conducted a preliminary study where a clause-level sentiment classification algorithm was developed and applied to drug reviews on a discussion forum (Na et al., 2012). This study is based on our previous work, and we have improved the approach by adding additional rules for both handling more complex relations among words and utilizing domain knowledge further in drug reviews. Moreover, the neutral class is tested in addition to positive and negative classes in this study, and the details of sentiment analysis rules and implementation are described in the paper.

In the following sections, related work is described first. Then our sentiment analysis method is proposed, and its experiment results and issues are described and discussed. Finally, conclusion information is provided.

2. RELATED WORK

Researchers have used various approaches for sentiment classification (Liu, 2012; Pang & Lee, 2008). Sentiment analysis approaches often require resources such as sentiment lexicons to determine which words or phrases are positive or negative in general or domain context. General Inquirer (Stone et al., 1966) is a manually compiled resource often used in sentiment analysis. Many techniques have been proposed for

learning the polarity of sentiment words (Huang et al., 2014; Lu et al., 2011; Qiu et al., 2009). In our study, general sentiment lexicon was generated using publicly available sentiment lexicons and domain lexicon was compiled from the development dataset.

Most of the early studies were focused on document-level analysis for assigning the sentiment orientation of a document (Pang et al., 2002). However, a document-level sentiment analysis approach is less effective when in-depth sentiment analysis of review texts is required. More recently researchers have carried out sentence-level sentiment analysis to examine and extract opinions regarding various aspects of reviewed subjects (Ding et al., 2008; Jo & Oh, 2011; Kim et al., 2013). Our approach uses clause-level sentiment analysis so that different opinions on multiple aspects expressed in a sentence can be processed separately for each clause. For instance, the sentence “*I like this drug, but it causes me some drowsiness*” has two clauses expressing two aspects: *overall opinion and side effects*. Some researchers have studied phrase-level contextual sentiment analysis, but phrases are often not long enough to contain both sentiment and feature terms together for aspect-based analysis (Wilson et al., 2009).

Generally there are two main approaches for sentiment analysis: a machine learning approach and a linguistic approach (i.e. a natural language processing approach) (Shaikh et al., 2008). Since clauses are quite short and do not contain many subjective words, the machine learning approach generally suffers from data sparseness problems. Also, the machine learning approach cannot handle complex grammatical relationships among words in a clause. Some researchers have used various linguistic features in addition to word features in the machine learning approach to overcome the limitations of the bag-of-words (BOW) approach (Liu, 2012). In this study, we are using a pure linguistic approach to overcome these weaknesses in the machine learning approach. We also compare the results of our proposed linguistic approach with the ones of a machine learning approach.

A relatively small number of works have studied social media content in health and medical domains. For instance, Xia et al. (2009) developed a system to classify new posts on a forum according to their topic and polarity. The forum is called “Patient Opinion” and is an online review service for users of the British

National Health Service. Then the users can add reviews on food, staff, treatments, and so on. Both topic and polarity identification were achieved through a rather simple machine learning approach using BOW. Niu et al. (2005) applied Support Vector Machine (SVM) to detect four possibilities of clinical outcome sentences in medical publications (Clinical Evidence: *no, positive, negative, or neutral outcome*). They argued that combining linguistic features (such as unigrams, bigrams, polarity change phrases, and negation) and domain knowledge (i.e. the semantic types of the UMLS) led to the highest accuracy (79.42%). However, they analysed clinical outcomes instead of user-generated content. Denecke (2008) proposed a classification method for distinguishing affective from informative medical blogs by utilizing both the semantic types of the UMLS and the sentiment lexicon SentiWordNet (Baccianella et al., 2010). The paper work is similar to subjectivity analysis for distinguishing subjective from objective documents (Wiebe & Riloff, 2005). Nikfarjam and Gonzalez (2011) developed an information extraction system to dig out mentions of adverse drug reactions from drug reviews by using association rule mining. Their evaluation results were 70% precision, 66% recall, and 68% F-measure. Tsytsarau et al. (2011) worked on the problem of identifying sentiment-based contradictions. They applied the proposed approach to a data set of drug reviews collected from the DrugRatingz website (<http://drugratingz.com>), but they used an existing general sentiment classification tool without considering domain knowledge, and focused mainly on the detection of contradictions.

3. SENTIMENT CLASSIFICATION METHOD

3.1. Overview

For clause-level sentiment analysis of drug reviews, first, each sentence is broken into independent clauses, and their review aspects are determined. Since automatic methods for clause separation and aspect detection are also challenging problems, the clauses and their aspects identified by the system are validated by manual coders. Then, for each clause semantic annotation (such as disorder terms) is performed, and a prior sentiment score is assigned to each word. Then, for the clause the grammatical dependencies are de-

terminated using a parser, and the contextual sentiment score (between -1 and 1) is calculated by traversing the dependency tree based on its clause structure.

3.2. Clause Separation and Aspect Detection

3.2.1 Clause Separation

In order to produce more accurate and efficient sentiment analysis, sentences are broken down into multiple clauses that can stand alone. For instance, the sentence “*I took this drug and it worked great.*” is separated into two clauses: “*I took this drug*” and “*and it worked great.*” Clauses can be dependent or independent. An independent clause (or main clause) is a clause that can stand by itself, also known as a simple sentence. An independent clause contains a subject and a predicate, and it makes sense by itself. Multiple independent clauses can be joined by using a semicolon or a comma plus a coordinating conjunction (e.g., *for, and, nor, but, or, yet, so, etc.*). A dependent clause (or a subordinate clause) is a clause that augments an independent clause with additional information, but which cannot stand alone as a sentence. Dependent clauses can start with conjunctions such as *after, although, as, as if, because, and so on.* Thus, in our study each clause indicates an independent clause which may include a dependent clause.

For clause separation, we parsed sentences into Tree Structures (i.e. Parse Trees) using Stanford Parser (de

Marneffe, 2006), and investigated the structures of the Parse Trees and their clause separation points. Then we manually constructed heuristics rules to split sentences into clauses. In the study the automatically separated clauses were validated by manual coders to reduce possible errors from the clause separation process.

3.2.2 Aspect Detection

The drug reviewers discuss different aspects of a drug depending on their interest and expertise. After analysing different groups of user-generated documents on multiple drug review websites, we identified six types of aspects related to drugs reviews. Table 1 describes the aspect name, its description, and some example sentences or clauses related to each aspect.

To detect aspects of clauses, we compiled important terms and UMLS semantic types for each aspect from drug review websites. For instance, the *Cost* aspect is relatively easier to detect than the other aspects since there are only a few terms indicating cost aspect, such as *cost, price, pay, afford, etc.* From our experiments, we found that the detection of *Effectiveness, Side effects, and Condition* aspects is more challenging than the other three categories since sometimes contextual information is necessary in order to differentiate them. In this study, automatically tagged aspect data were validated by manual coders to reduce possible errors from the aspect detection process.

Table 1. Six Aspects of Drug Reviews

Aspect Name	Description	Example clauses
Overall	Corresponds to general opinion on a drug. We also use this category when a clause does not match any other aspects.	- <i>It is great.</i> - <i>I've been pleased with Actoplus Met overall.</i>
Effectiveness	Corresponds to the changes noticed after taking a drug. Difference from <i>Side effects</i> is that they are directly related to the condition of the patient or the disease.	- <i>But it didn't work well enough for me.</i>
Side Effects	<i>Side effects</i> are all the effects due to the drug that are not related to the drug.	- <i>I developed a serious rare stomach disorder from it.</i>
Dosage	Related to the quantity, the frequency, or the treatment period of drugs taken.	- <i>I take 50 mg as needed.</i>
Condition	A description of the patient's condition: mainly diseases, but also allergies and general health problems.	- <i>I had some sinus surgery.</i> - <i>I have a history of respiratory problems.</i>
Cost	Discusses the cost of the drugs.	- <i>Taking Ambien is a waste of money.</i>

3.3. Sentiment Lexicons

We have created a general lexicon (9,630 terms) and a domain lexicon (278 terms). For the general lexicon construction, first we collected positive and negative terms (7,611 terms) from Subjectivity Lexicon (SL) (Wilson et al., 2005). SL contains more than 8,000 subjective expressions manually annotated as strongly or weakly subjective, and as positive, negative, neutral, or both. We set the prior score +1 to strongly subjective positive terms and -1 to strongly subjective negative terms. Also, we set the prior score +0.5 to weakly subjective positive terms and -0.5 to weakly subjective negative terms. An additional 2,019 terms were collected from SentiWordNet, and they were manually tagged by three manual coders, and conflicting cases were resolved using a heuristic approach.

In addition, we added some domain specific terms to the domain lexicon during the development phase (e.g., *work (verb)*: +1; *sugar high*: -0.5; *sugar in control*: +0.5; *heartbeat up*: -1). To compensate for the small domain lexicon, MetaMap is used to tag disorder terms, such as “*pain*” and “*hair loss*,” using the Disorders semantic group (Bodenreider & McCray, 2003), and with them set to -1 sentiment score. The Disorders semantic group contains a list of UMLS semantic types related to disorder terms, such as “*Disease or Syndrome*” and “*Injury or Poisoning*.” In the study we excluded the “*Finding*” semantic type from it to reduce false positive disorder terms.

3.4. Dependency Tree

We used the Stanford NLP library (de Marneffe, 2006) to process the grammatical relationships of words in a clause. There are fifty-five Stanford typed dependencies which are binary grammatical relationships between two words: a governor and a dependent. For example, the sentence “*I like the drug.*” has the following dependencies among the words:

- *nsubj*[like-2, I-1]
- *root*[ROOT-0, like-2]
- *det*[drug-4, the-3]
- *dobj*[like-2, drug-4]

In the type dependencies, *nsubj*[like-2, I-1] indicates that the first word “*I*” is a nominal subject of the second word “*like*.” In the dependency relationship, “*I*” is

the dependent (or modifier) and “*like*” is the governor (or head). *dobj*[like-2, drug-4] indicates that the fourth word “*drug*” is the direct object of the governor “*like*.” *det*[drug-4, the-3] indicates that “*the*” in the third position is a determiner of “*drug*.” *root*[ROOT-0, like-2] indicates that “*like*” is a root node in the dependency tree. Based on the output set of grammatical dependencies, a dependency tree is constructed as shown in Fig. 1. Each node contains its POS and prior sentiment score.

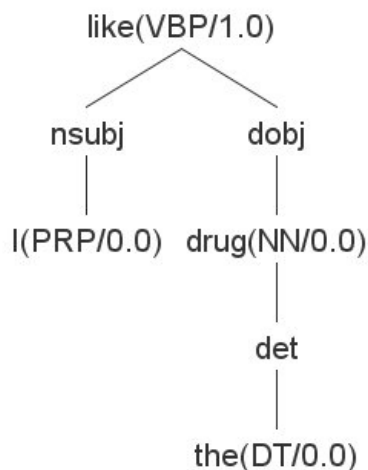


Fig. 1 Graphical representation of the dependency tree for the sentence “*I like the drug.*”

3.5. Sentiment Analysis Rules for Determining Contextual Sentiment Score

In our proposed approach, various calculation rules are used to determine the contextual sentiment score of a clause. We have developed the sentiment analysis rules using the development dataset prepared from the drug review website DrugsExpert. Most of the sentiment analysis rules are deduced by observing common patterns in the development dataset. Several special rules, such as Polarity Shifter and Positive and Negative Valence rules, are introduced by following existing literature (Polanyi & Zaene, 2006; Wilson et al., 2005). In the first subsection, general rules and formulas covering basic syntactic relations between words are described, and additional special rules for handling more complex relations between words are discussed in the second subsection.

3.5.1. General Rules for Contextual Sentiment Scores

Phrase. Phrase rules covering adjectival, verb, and noun phrases are used to calculate contextual sentiment scores of phrases which can be a subject, object, or verb phrase in English sentences.

Adjectival Phrase rules in Table 2 handle a relationship between an adverb and an adjective defined by the Adverbial Modifier relation *advmod()*. When inputs (i.e. an adverb and an adjective) are of the same sentiment orientation (F1 and F4 in Table 2), they tend to intensify each other. The absolute value of the output should be larger than or equal to the absolute values of the inputs but less than 1. Therefore, the formula $(+ (|Adjective| + (1-|Adjective|) * |Adverb|))$ is applied. For example, the adjective “*Enthusiastically*” in the phrase “*Enthusiastically Responded*” intensifies the neutral adjective “*Responded*.” Thus, the phrase “*Enthusiastically Responded*” becomes more positive than the adjective “*Responded*” by itself. The prior sentiment score of the adjective “*Responded*” and the adverb “*Enthusiastically*” are 0 and +1, respectively. The output sentiment score of the phrase “*Enthusiastically Responded*” is calculated as $(+ (0 + (1 - 0) * 1) = +1)$. Similarly, the adverb “*Strictly*” in the phrase

“*Strictly Prohibitive*” intensifies the negative adjective “*Prohibitive*.” Thus, the phrase “*Strictly Prohibitive*” becomes more negative than the adjective “*Prohibitive*” by itself.

When the adverb is positive and adjective is negative (i.e. F2 in Table 2), the adverb also intensifies the adjective. For example, the adverb “*Extremely*” in the phrase “*Extremely Disappointed*” intensifies the adjective “*Disappointed*.” Thus, the phrase “*Extremely Disappointed*” becomes more negative than the adjective “*Disappointed*” by itself (since “*Disappointed*” has the maximum negative value -1 already, it will remain as -1). The prior sentiment scores of the adjective “*Disappointed*” and the adverb “*Extremely*” are -1 and -1, respectively. The output sentiment score of the phrase “*Extremely Disappointed*” is calculated as $(-(1 + (1-1) * 1) = -1)$. When the adverb is negative and the adjective is positive (i.e. F3 in Table 2), the output is the value of the negative adverb. For example, the adjective “*Glaringly*” in the phrase “*Glaringly Stared*” determines the output score. The prior sentiment scores of the adverb “*Glaringly*” and the adjective “*Stared*” are -1 and 0, respectively. Thus, the output sentiment score of the phrase “*Glaringly Stared*” becomes -1.

Table 2. Adjectival Phrase Rules for adverb + adjective (*advmod*)

ID	(Ad)verb	(A)djective	Output	Examples Advmod(adjective, adverb)
F1	+ / 0*	+ / 0	+	<i>Enthusiastically Responded</i> (+1)
F2	+ / 0	-	-	<i>Extremely Disappointed</i> (-1)
F3	-	+ / 0	-	<i>Glaringly Stared</i> (-1)
F4	-	-	-	<i>Strictly Prohibitive</i> (-1)

*: 0 indicates neutral.

Formula	
F1	Positive Ad and Positive A (positive includes neutral) => $(A + (1- A) * Ad)$ E.g., <i>Enthusiastically Responded</i> : 1 and 0 => $(0 + (1-0) * 1) = +1$
F2	Positive Ad and Negative A => $-(A + (1- A) * Ad)$ E.g., <i>Extremely Disappointed</i> : -1 and -1 => $-(1 + (1-1) * 1) = -1$
F3	Negative Ad and Positive A => Ad E.g., <i>Glaringly Stared</i> : -1 and 0 => -1
F4	Negative Ad and Negative A => $-(A + (1- A) * Ad)$ E.g., <i>Strictly Prohibitive</i> : -1 and -0.5 => $-(0.5 + (1-0.5) * 1) = -1$

Verb Phrase rules handle a relationship between a verb and an adverb defined by the Adverbial Modifier relation *advmod()*. For calculating contextual scores of verb phrases, the same formulas used in Adjectival Phrase rules (Table 2) are applied with applicable POSs. Examples of verb phrases are as follows: *cheer happily*: +1; *gossip proudly*: -1; *liberate badly*: -1; *fail badly*: -1. *Noun Phrase* rules handle a relationship between an adjective and a noun defined by the Adjectival Modifier relation *amod()*. The subject or object of a clause can be a noun phrase consisting of a noun and an adjective. For calculating contextual scores of noun phrases, the same formulas used in Adjectival Phrase rules are also applied. Examples of noun phrases are as follows: *great drug*: +1; *big failure*: -0.5; *lousy drug*: -1; *worst disease*: -1. In implementation, *advmod()* is processed before *amod()* to handle three-word phrases, such as “*very nice drug*.” In other words, the contextual sentiment score of “*very nice*” is calculated first with *advmod()*, and then the resulting value is used as the value of “*nice*” to calculate the contextual score of “*nice drug*” using *amod()*. This allows calculating a contextual sentiment score for “*very nice drug*.” Also, relevant POSs (i.e. adverb, adjective, and noun) are checked with the two functional relations: *amod()* and *advmod()*.

Conjunct. A conjunct is the relation between two elements connected by a coordinating conjunction: *and*, *or*, *but*, and so on. Conjunct Rules handle a relationship between two terms connected by the Conjunct relations: *conj_and()*, *conj_or()*, and *conj_but()*.

When inputs are of the same sentiment orientation, they intensify each other if they are connected by “*and*” (e.g., *He is good and honest*: +1; *He is bad and dishonest*: -1). If they are connected by other connectors such as “*or*,” the average value of the two input values is used. When inputs have different sentiment orientation, their sentiment intensity values are compared. If they are the same, the output value becomes 0 (e.g., *He is handsome but dishonest*: 0). Otherwise, the higher intensity value becomes the output value.

Predicate. Each clause consists of a subject and a predicate. The predicate indicates all the syntactic components except its subject. Predicate rules handle a relationship between a verb phrase and an object / complement defined by the Direct Object, Indi-

rect Object, or Adjectival Complement relationship: *dobj()*, *iobj()*, or *acomp()*. For calculating contextual scores of predicates, the same formulas used in Adjectival Phrase rules (Table 2) are also applied. Examples of phrases are as follows: *provide goodness*: +1; *looks beautiful*: +1; *provide problems*: -0.5; *lose award*: -0.5; *suffers pain*: -1.

Clausal Complement Relation rules handle a relationship between a verb phrase / adjective phrase and a clausal complement defined by the Open Clausal Complement relationship, *xcomp()*. These complements do not have their own subjects and are always non-finite, which includes participles, infinitives, or gerunds. For instance, for clauses with “*to*” dependency, the sentiment score of the second clause is intensified when the first clause is positive (e.g., *I would love to use this great drug again*: +1; *I will advise anyone to throw away the drug*: -0.5), and the sentiment score of the second clause is negated when the first clause is negative (e.g., *It is hard to like this drug*: -1.0; *It is hard to find a problem*: +0.5).

Clause. Each clause consists of a subject and a predicate. The rules in Table 3 handle a relationship between a subject and a predicate defined by the nominal subject, clausal subject, passive nominal subject, and clausal passive subject relationships: *nsubj()*, *csubj()*, *nsubjpass()*, and *csubjpass()*. As before, when inputs are of the same sentiment orientation, they intensify each other (F1 and F4 in Table 3). When the subject is positive and the predicate is negative (F2 in Table 3), the output is the value of the negative predicate. However, when the subject is negative and the predicate is positive (F3 in Table 3), the output can be either positive or negative. Therefore, the values of the subject and the predicate are compared, and if the absolute value of the subject is larger than that of the predicate, the output becomes the sentiment score of the subject, and vice versa. For example, in the clause “*The pain started at night time*,” the absolute value of the subject “*The Pain*” is greater than that of the predicate “*started at night time*.” Thus, the output is the sentiment score of “*Pain*,” which is negative. However, in the clause “*This expensive drug was amazing*,” the absolute value of the predicate “*was amazing*” is greater than the value of the subject “*This expensive drug*,” and thus the output is the sentiment score of “*was amazing*,” which is positive.

Table 3. Clause Rules for subject + predicate (nsubj, csubj, nsubjpass, csubjpass)

ID	(S)ubject	(P)redicate	Output	Examples nsubj (predicate, subject) csubj (predicate, clausal subject) nsubjpass (predicate, subject) csubjpass (predicate, clausal subject)
F1	+ / 0	+ / 0	+	<i>The drug did great.</i> (+1)
F2	+ / 0	-	-	<i>It performed poorly.</i> (-1)
F3	-	+ / 0	+/-	<i>This expensive drug was amazing.</i> (+1) <i>The pain started at night time.</i> (-1)
F4	-	-	-	<i>The bad drug spoiled everything.</i> (-1)
Formula				
F1	Positive S and Positive P => $(P + (1- P) * S)$ E.g., 0 and +1 => $(1 + (1-1) * 0) = +1$			
F2	Positive S and Negative P => P E.g., 0 and -1 => -1			
F3	Negative S and Positive P => If $ S > P $ Then S Else P E.g., -0.5 and +1 => +1 -1 and 0 => -1			
F4	Negative S and Negative P => $- (P + (1- P) * S)$ E.g., -1 and -1 => $-(1 + (1-1) * 1) = -1$			

Default Rule. Since the previously defined rules (and subsequently defined rules) cannot comprehensively cover all the grammatical dependencies of words in clauses, the default calculation rules in Table 4 are applied to unmatched phrases. The formulas are generalized since the output sentiment orientation can vary in such situations. When both terms are either positive or negative, they intensify each another and the output maintains their original sentiment orientation (F1 and F4 in Table 4). However, when their sentiment orientations are different, the term with a greater sentiment score is used as the output (F2 and F3 in Table 4). For instance, in the case of “*the drug’s flaws*,” “*drug*” (noun) and “*flaws*” (noun) are associated with the possession modifier relation. This case is processed using formula F3 in Table 4.

Clause connector. A clause connector is used to connect two clauses in a clause, and it is used to merge their contextual sentiment scores into one value. For instance, the adverbial clause modifier functional dependency defines a relation between a verb phrase and a clause modifying the verb. Therefore, the Clause

Connector rules should be applied after the involved clauses’ sentiment scores are calculated, using the clause rules (i.e. rules in Table 3). The rules handle adverbial clause modifier, clausal complement, and purpose clause modifier relations: *advcl()*, *ccomp()*, and *purpcl()*. The Clause Connector rules apply the similar formulas used in Default rules (Table 4). Examples clauses are as follows: *He says that the drug works well:* +0.5; *He says that the drug does not work well:* -0.5; *He misuses drugs in order to sleep well:* -1; *The pain increased as the night was falling:* -1.

Negation of Term. Handling of negation is one of the key processes in sentiment analysis. The negation modifier *neg()* is the relation between a negation word (such as *not*, *never*, and *none*) and the word it modifies. For example, in the phrase “*not good*,” the negativity of the word “*not*” is -1 and the sentiment score of the term “*good*” is +0.5. The output becomes -0.5 through multiplying +0.5 by -1. For the phrase “*not lousy*,” the output becomes +1. In addition, a special rule is defined to output the negation of a neutral term as -0.5. For instance, when the score of “*perform*” is 0,

Table 4. Default Rules (Any Undefined Relations)

ID	(G)overnor	(D)ependent	Output	Examples Type-dependency (Governor, Dependent)
F1	+ / 0	+ / 0	+	<i>my tolerance</i> (+1) possession (tolerance, my)
F2	+ / 0	-	+/-	<i>in terms of tension</i> (-1) prep_of (terms, tension)
F3	-	+ / 0	+/-	<i>the drug's flaws</i> (-0.5) possession (flaws, drug)
F4	-	-	-	<i>A depression that really was very bad</i> (-1) rmod (depression, bad)
Formula				
F1	Positive G and Positive D => + (G + (1- G) * D) E.g., + 1 and + 0 => + (1 + (1-1) * 0) = +1			
F2	Positive G and Negative D => If G >= D Then G Else D E.g., 0 and -1 => -1			
F3	Negative G and Positive D => If G >= D Then G Else D E.g., -0.5 and 0 => -0.5			
F4	Negative G and Negative D => - (G + (1- G) * D) E.g., -1 and -1 => - (1 + (1-1) * 1) = -1			

the sentiment score for “*not perform*” becomes -0.5. Besides neg() dependency checking, a negation lexicon (a total of 13 negation terms, such as *not, never, none, nobody, nowhere, nothing, and neither*) is used to detect negation terms without considering their grammatical dependencies with other terms.

Polarity Shifter. Some polarity shifting (or negating) words, called polarity shifters, are not detected as the neg() type dependency and should be handled in other type dependencies, such as advmod(), acomp(), xcomp(), and dobj().

Adjectival Polarity Shifter rules handle the negation relation between an adjective and an adverb defined by the Adverbial Modifier relation advmod(), where the negating adverb shifts the original sentiment orientation of an adjective. When an adjective is positive, the result becomes negative, and vice versa. For example, in the verb phrase “*hardly good*,” negating the positive adjective “*good*” with “*hardly*” makes the phrase negative (-0.5). However, the phrase “*hardly bad*” becomes positive (+0.5).

The Polarity Shifter rules for verb phrases handle the negation relation between a verb and an adverb

defined by the Adverbial Modifier relation advmod(). The negating adverb shifts the original sentiment orientation of a verb (e.g., *rarely succeed*: -0.5; *hardly fail*: +0.5). For the negating governor terms in dobj(), acomp(), and xcomp() relations, the Polarity Shifter rules for predicates are used (e.g., *ceased boring* (dobj): +1; *stopped success* (dobj): -0.5; *stopped interesting* (acompany): -1; *stopped to use* (xcomp): -0.5). For the negating verbs, the Polarity Shifter rules for clauses are used (e.g., *The wild dreams stopped*: +1; *The effect stopped*: -0.5). Currently we are using 11 polarity shifter terms collected manually.

3.5.2. Special Rules for Contextual Sentiment Scores

Intensify, mitigate, maximize, and minimize. The general rules do not handle intensify, mitigate, maximize, and minimize relationships. Thus, new special rules are used to handle these relationships using additional lexicon terms including intensifiers (e.g., *tremendously* and *greatly*), mitigators (e.g., *slightly* and *more or less*), maximizers (e.g., *unquestionably* and *absolutely*), and minimizers (e.g., *little* and *scarcely*). The lexicon terms (a total of 138 terms) are collected main-

ly from the given lists from Quirk et al. (1985). These modifier terms are checked before the corresponding functional dependencies, such as `amod()` and `advmod()`, are applied.

In the Intensify rule, the polarity score of the modified word is doubled, but limited to a value of ± 1 (e.g., *enormously good*: +1). If the modified word is neutral, the score becomes +0.5. Conversely, the polarity of the modified word is halved in the Mitigate rule (e.g., *slightly better*: +0.5). The polarity score is maximized to ± 1 in the Maximize rule (e.g., *totally bad*: -1), and in the Minimize rule, the polarity score of the modified word is reduced to a quarter of its original score (e.g., *minimal passion*: +0.25).

Positive and negative valence. A positive or negative valence term can determine overall sentiment polarity of a whole clause no matter how it is modified by other terms. We are using 13 positive valence terms (e.g., *help* and *improve*) and 8 negative valence terms (e.g., *hate* and *suffer*) that are all verbs. If a positive or negative valence term is found in a clause, its sentiment prior score becomes an output sentiment score of the clause (e.g., *helped me escape from pains*: +0.5; *hate high level drugs*: -1). In case the sentiment prior score of the valence term is neutral, the output value becomes +0.5 for a positive valence term and -0.5 for a negative valence term.

Decrease disorder. Rules are defined to handle “decrease-type verb + disorder object” cases, such as “*It reduces the pain.*” If the general predicate rules are applied, the output of the clause “*It reduces the pain*” will be -1, which is wrong. The general rules can handle only “increase-type verb + disorder object” cases, such as “*It increases the pain.*” The rule in Table 5 handles the direct object relation between a decrease-type verb

and a disorder object defined by the Direct Object `dobj()`. The output becomes +1 by multiplying the disorder object value (i.e. -1) by -1. We have collected 23 decrease-type verbs, such as *reduce*, *decrease*, *lessen*, and so on. The Disorders semantic group in UMLS is used to tag disorder terms in clauses using MetaMap. For example, in the sentences “*It releases the pain*” and “*The drug causes me hair loss*,” the terms “*pain*” and “*hair loss*” are tagged as disorder terms with -1 sentiment score since they are concepts under the Disorders semantic group.

Similarly the Decrease Disorder rules for clauses are used to handle “disorder subject + decrease-type verb” situations (e.g., *Symptoms subsided*: +1) and the Decrease Disorder rules for noun phrases are used to handle “decrease-type adjective + disorder noun” cases (e.g., *Less side effects*: +1).

Preposition (for, as, with) and disorder. Rules are defined to handle “(prep_for, prep_as, or prep_with) type dependency + disorder term” cases. For instance, without these rules, the phrase “*good drug for fever*” would have a negative score -1 by using existing rules, such as Default rules. But by introducing these new rules, “*good drug for fever*” gets a positive score of +0.5 since the new rules set the sentiment score of the disorder term to 0 and ignore the “*for fever*” part. Examples of “(prep_for or prep_as) type dependency + disorder term” cases are as follows: *Ineffective drug for pain*: -0.5; *A drug for pain*: 0. Additional similar rules are defined to handle “(prep_with) type dependency + disorder term” cases (e.g., *Helped with pain*: +0.5; *Dying with pain*: -1). Particularly, in `prep_with(G, D)`, if G is neutral the value of D (i.e. the disorder term) is returned (e.g., *A guy with ADHD*: -1).

Preposition (because of, due to) and disorder. When

Table 5. Decrease Disorder Rules for Predicates (`dobj`)

ID	(V)erb Phrase	(O)bject	Output	Examples
F1	Decrease-Type Verb (+/0/-)	Disorders Semantic Group (-)	+	<i>Decrease the pain</i> (+1)
Formula				
F1	Decrease-Type Verb V and Disorders Semantic Group Term O => -1 * O E.g., <i>decrease the pain</i> : -0.5 and -1 => -1 * -1 = +1			

a “because of” or “due to” term occurs with a disorder term, it can affect the sentiment score result of the main clause, and so we set the sentiment score of the disorder term to 0. For instance, with the rule the system can return a right negative sentiment score for the clause “*I hardly get my work done because of headaches.*” since “*because of headaches*” is set to score 0 and does not affect the sentiment score calculation of the main part. This rule is applied when the distance between a “because of” or “due to” term and a disorder term is within the predefined range (i.e. 5).

Disorder and medication. This rule is used to handle “disorder term + medication (drug)” cases. For instance, without this rule the noun phrase “*cancer drug*” would have a negative score -1 by using Default rules. But by introducing this rule, it becomes 0.

Phrasal verb. A phrasal verb (such as *back off* and *break up*) is identified by using the phrasal verb particle relation prt(verb, particle), which identifies a relation between the verb and its particle. In the case of “*back off*” that is negative (-1.0); if “*back*” and “*off*” are handled by Default rules, it will become +0.75 since “*back*” and “*off*” have the prior scores +0.5 and +0.5 respectively. Currently we are using 121 phrasal verbs collected manually (e.g., *blow up*: -1; *get along*: +1).

Contradicting connectors. Using this rule, dependent clauses having contradicting connectors such as *although*, *though*, *however*, *but*, *on the contrary*, and *notwithstanding* are ignored, and only their main clauses are considered to calculate contextual sentiment scores. For instance, in the sentence “*Although the drug worked well, it gave me a headache,*” the dependent clause “*Although the drug worked well*” is ignored or neutralized.

Question. This is used to detect question clauses using the question mark “?” It will set the sentiment polarity of a whole clause to the neutral value 0.

3.6. Implementation

We use a bottom-up approach in applying the rules in the dependency tree, and leaf nodes are evaluated first and the resultant polarity scores propagated to upper-level nodes for further evaluation. Since a node in the dependency tree can have multiple relations with its children nodes, we set rule priorities among the relations between a parent node and its direct children nodes (see Fig. 2). Generally, these rule priorities

allow the system to calculate the scores of phrases (i.e. small components) first in a clause, and use the calculated phrase scores to calculate a predicate score, and finally the clause score is calculated using the scores of the subject and the predicate.

Now we will see how the system processes the sentence “*It could not completely reduce the stomach pain*” with actual rules. The dependency tree for the sentence is shown in Fig. 3, in which words in the sentence are nodes and grammatical relations are edge labels. First, the contextual sentiment score of the object “*stomach pain,*” nn(pain, stomach), at the lowest bottom level is calculated using Default rules (formula F3 in Table 4). The prior sentiment scores of the nouns “*stomach*” and “*pain*” are 0 and -1, respectively. Thus, the contextual sentiment score of the noun phrase “*stomach pain*” is calculated to -1 (note that the determiner relation between “*the*” and “*pain*” is ignored). Subsequently, the root node “*reduce*” is processed with its five children nodes. First, advmod(reduced, completely) is processed since it has a higher priority than the other four relations. The contextual sentiment score of the verb phrase “*completely reduce,*” advmod(reduce, completely), is calculated by using the Maximize rule since “*completely*” is a maximize term. The contextual score of the verb phrase is calculated to +0.5 (the Maximize rule converts the neutral verb term to +0.5). For the predicate “*completely reduce the stomach pain*” defined by dobj(reduce, pain), the sentiment score is calculated using *Decrease Disorder* rules (formula F1 in Table 5) since “*reduce*” is a decrease-type verb and “*pain*” is a disorder term. The sentiment scores of the verb phrase and object are +0.5 and -1, respectively. Thus, it is calculated as $(-1 * -1)$ which is equal to +1 (the decrease disorder rule shifts the original sentiment orientation of the disorder object). Then, the score for the clause “*It completely reduce the stomach pain*” defined by nsubj(reduce, It) is calculated by using Clause rules (formula F1 in Table 3). Since the prior score of the subject “*It*” is 0, the result score of the clause remains +1 (note that the auxiliary relation between “*reduce*” and “*could*” is ignored). Finally, the system processes neg(reduce, not) using Negation of Term rules which negate the computed score of the intermediate clause “*It completely reduce the stomach pain*” and returns the final score of -1.

1. Question rule: check for question mark
2. Phrasal verb rules: prt()
3. Preposition (because of, due to) and disorder rules: any type dependencies, (because of, due to) + disorder
4. Positive and negative valence rules: any type dependencies
5. Disorder and medication rule: any type dependencies, disorders + medication
6. Adjectival phrase rules: advmod (adjective, adverb)
7. Noun phrase rules: amod()
8. Conjunct rules: conj_and(), conj_or(), and conj_but()
9. Verb phrase rules: advmod (verb, adverb)
10. Predicate rules: dobj(), iobj(), and acomp()
 - Handle decrease disorder rule as a special case: dobj()
11. Clausal complement relation rules: xcomp()
12. Default rules: e.g., prep_of(), possession(), etc.
13. Preposition (for, as, with) and disorder rules : (prep_for, prep_as, or prep_with) + disorder
14. Clause rules: csubj(), csubjpass(), nsubj(), and nsubjpass()
 - Handle intensify, mitigate, maximize, and minimize rules as special cases: advmod (adjective, adverb), amod(), advmod (verb, adverb), and dobj().
15. Negation of term rules
 - Handle polarity shifter rules as special cases: advmod (adjective, adverb), advmod (verb, adverb), and dobj().
16. Contradicting connectors rule: advcl (contradicting connector terms), ccomp (contradicting connector terms), purpcl (contradicting connector terms)
17. Clause connectors rules: advcl (), ccomp(), and purpcl()

Fig. 2 Rule priorities for relations between a parent node and its direct children nodes

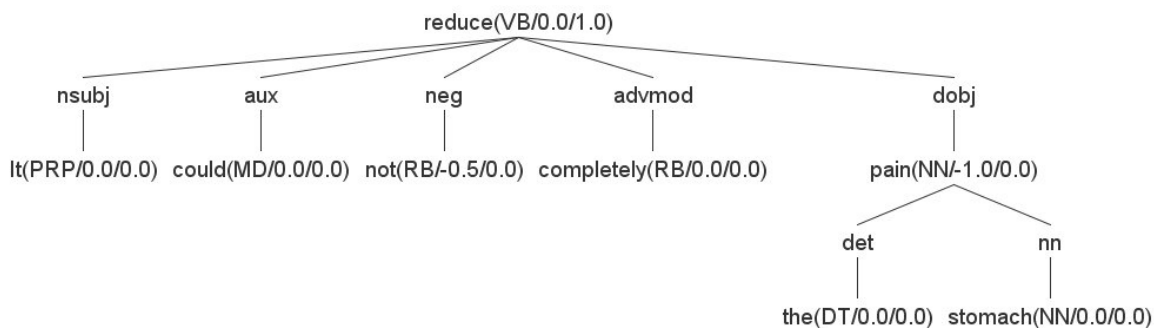


Fig. 3 Dependency tree for the clause "It could not completely reduce the stomach pain."

4. EXPERIMENT RESULTS AND DISCUSSION

4.1. Datasets

Drug review sentences were collected from the drug review website WebMD (www.webmd.com) to evaluate the developed algorithm. The target drugs are mainly diabetes, depression, ADHD (Attention Deficit Hyperactivity Disorder), slimming pills, and sleeping pills. For the algorithm development, we used the development dataset prepared from the drug review website Drugs-Expert, which is no longer in service. First, two manual coders worked on the same 300 clauses sampled from the development dataset and tagged them into positive, negative, or neutral classes with corresponding aspects. The agreement rate between them is 84% and the Cohen Kappa value is 0.65, which is considered as substantial agreement. We noted that the neutral class has a higher disagreement rate than the other two classes since some neutral clauses can be interpreted as either positive or negative. For instance, “*Ambien (zolpidem) gives me no side effects except mild occasional amnesia.*” was tagged as positive class by one coder but as neutral class by the other coder because of mixed sentiments. For the evaluation dataset the two coders, trained with the 300 clauses, tagged a total of 4,200 clauses from WebMD, and we selected randomly a dataset of 2,700 clauses (900 for each class) for sentiment classification. Then the tagged labels were validated by one of the authors.

4.2. Results

In order to provide benchmarks for comparison with the proposed linguistic approach, we also conducted experiments with a machine learning approach, SVM. 10-fold cross validation was used, and precision, recall, F-score, and accuracy are calculated using the following formulas:

$$\text{Precision} = \frac{\text{number of correctly classified positive, negative, or neutral clauses}}{\text{number of automatically classified positive, negative, or neutral clauses}} \quad (1)$$

$$\text{Recall} = \frac{\text{number of correctly classified positive, negative, or neutral clauses}}{\text{number of manually classified relevant positive, negative, or neutral clauses}} \quad (2)$$

$$F\text{-score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \quad (3)$$

$$\text{Accuracy} = \frac{\text{number of correctly classified positive, negative, and neutral clauses}}{\text{number of manually classified relevant positive, negative, and neutral clauses}} \quad (4)$$

In the first machine learning approach (SVM-1), we used BOW (with term presence) and negation document features for sentiment classification. For negation handling, when negation terms such as *neither*, *never*, *no*, *non*, *nothing*, *not*, and *none* occur an odd number of times in a clause, the negation feature becomes 1, otherwise it becomes 0. In the second approach (SVM-2), we added an additional linguistic bi-gram feature to consider grammatical relations between words and to overcome data sparseness problems. For each typed dependency of all the 55 Stanford typed dependencies, we use four document features as follows: TD(+,+), TD(+,-), TD(-,+), and TD(-,-). The governor and dependent terms in type dependencies are converted to + or - using the general and domain lexicons (note that + includes neutral) to utilize prior scores of subjective terms. For instance, for *amod(drug, worst)*, we have the following four type dependency features: *amod(+,+)*: 0, *amod(+,-)*: 1, *amod(-,+)*: 0, and *amod(-,-)*: 0. Table 6 shows precision, recall, F-score, and accuracy of the two baseline machine learning approaches and our linguistic approach by comparing the system results to the answer keys (gold standard).

As shown in Table 6, the accuracy of the SVM-2 approach is significantly better than for SVM-1 (two-sided t-test, $p \leq 0.01$) since the additional linguistic features help for sentiment classification. Also, our linguistic approach performed significantly better than both baselines (two-sided t-test, $p \leq 0.01$). Table 7 shows precision, recall, F-score, and accuracy of our linguistic approach for the six aspects. Accuracy of *Dosage* clauses is the highest (82%), and accuracy of *Cost* clauses is the lowest (50%).

Table 6. Precision, Recall, F-score, and Accuracy of Three Sentiment Classification Methods

Approach	Term Features for SVM			Polarity	Precision	Recall	F-Score	Accuracy
	BOW	Negation	Type Dependencies					
SVM-1	*	*		Positive	0.65	0.57	0.61	0.62 (1676/2700)
				Negative	0.57	0.63	0.60	
				Neutral	0.45	0.62	0.52	
SVM-2	*	*	*	Positive	0.70	0.66	0.68	0.66 (1791/2700)
				Negative	0.63	0.67	0.65	
				Neutral	0.45	0.63	0.53	
Linguistic Approach				Positive	0.73	0.74	0.73	0.69** (1871/2700)
				Negative	0.69	0.70	0.70	
				Neutral	0.44	0.67	0.54	

** : significant at the 0.01 level

4.3. Error Analysis

We have conducted error analysis on 829 error clauses that were misclassified by the proposed linguistics approach. Based on the nature of errors, we categorized the sources of errors into seven groups. Fig. 4 shows the names of seven error groups and their distribution.

• Inference Problem (168 clauses, 20%)

The inference problem includes modality and indirect expressions. Modality (Palmer, 2001) is related to the attitude of the speaker towards his/her statements in terms of degree of certainty, reliability, subjectivity, sources of information, and perspective. As an example of modality problems, the clause *"I know that if I exercised more, I would obviously lose more weight"* expresses a counterfactual mood where a certain situation or action is not known to have happened at the time of the author's writing. This was tagged as a neutral clause, but the system predicts it as a positive one. Currently, we are not handling modality problems and we leave it as our future work. For indirect expressions, the example clause *"I can actually throw out my really 'big girl clothes' for good"* expresses the author's positive sentiment towards the effectiveness of the slimming pill drug. However, it is tough for the machine to infer the true sentiment orientation of the given clause.

• System Error (161 clauses, 19%)

In the system error, negation handling is a challenging issue since a negation term is meant to negate a specific component in a clause, such as the verb, object, complement, or following clause. We noticed in many cases that the system negated a wrong component in the dependency tree of a clause, and this caused errors. In addition, incorrect grammatical parsing caused the system to trigger irrelevant rules and that affected the eventual polarity prediction. The parsing errors usually happened when clauses are grammatically wrong or incomplete (such as no subject in a clause). A limited number of the rules also caused system errors since more specialized rules are needed to handle various complex expressions in clauses.

• Lexicon Error (108 clauses, 13%)

Inappropriate assignment of the prior scores of words prevents the system from making accurate predictions. Idiomatic expressions contributed the major proportion of lexicon errors. For example, the clause *"It keeps my life on track"* implies a positive sentiment. But when the system calculated the sentimental score of this expression, the result was neutral because the term *"track"* holds neutral polarity in the general lexicon. To solve the lexicon error problem, we need to add commonly

Table 7. Precision, Recall, F-score, and Accuracy of Aspect-Level Sentiment Classification using a Linguistic Approach

Aspect	Polarity	Precision	Recall	F-Score	Accuracy
Overall	Positive	0.63	0.63	0.63	0.63 (817/1298)
	Negative	0.56	0.64	0.60	
	Neutral	0.67	0.62	0.65	
Effectiveness	Positive	0.94	0.77	0.85	0.75 (402/535)
	Negative	0.53	0.76	0.62	
	Neutral	0.08	0.29	0.13	
Side Effect	Positive	0.43	0.72	0.54	0.67 (263/390)
	Negative	0.89	0.71	0.79	
	Neutral	0.02	0.05	0.03	
Condition	Positive	0.39	0.88	0.54	0.68 (64/94)
	Negative	0.84	0.73	0.78	
	Neutral	0.52	0.48	0.50	
Dosage	Positive	0.59	0.82	0.69	0.82 (280/340)
	Negative	0.47	0.49	0.48	
	Neutral	0.93	0.88	0.90	
Cost	Positive	0.00	0.00	0.00	0.50 (3/6)
	Negative	1.00	1.00	1.00	
	Neutral	0.00	0.00	0.00	

Distribution of error sources

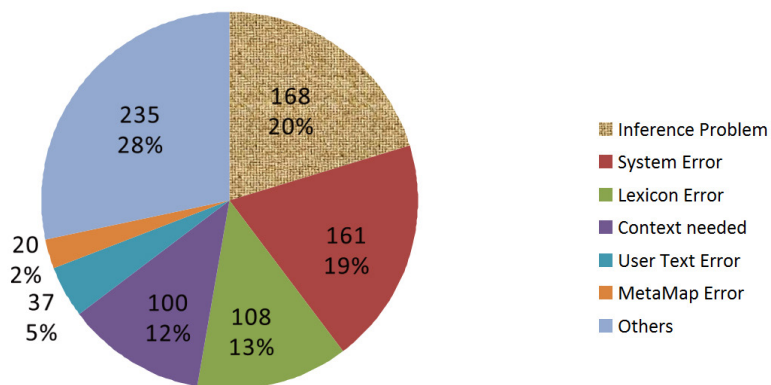


Fig. 4 Distribution of seven error groups

used idiomatic expressions to the domain lexicon.

- **Context Needed (100 clauses, 12%)**

It is hard for the system to determine the polarity of an individual clause without knowing the whole context. For instance, the clause “*when I am on this pill, I have no appetite at all!*” becomes positive for a slimming pill’s reviews, but on the other hand it becomes negative for other drugs’ reviews. So the introduction of drug specific rules may help to resolve context related issues.

- **User Text Error (37 clauses, 5%)**

User text errors such as spelling and grammatical mistakes raised a series of issues in grammatical parsing, prior score assignment, and semantic detection, and led the system to make wrong predictions.

- **MetaMap Error (20 clauses, 2%)**

MetaMap cannot detect certain disorder terms correctly. So we are using our own disorder term list to compensate for MetaMap errors. However, the size of our disorder lexicon is relatively small compared to MetaMap and more work needs to be done to develop a comprehensive disorder terms list.

- **Other Error (235 clauses, 28%)**

We noticed that common sense knowledge is required to detect the effectiveness or side effects of a drug. For instance, in the clause “*went from a bmi of 33 to 31,*” it is difficult for the system to determine the user’s sentiment towards the effectiveness of the drug without knowing the desired value range of the BMI value. The acquisition of common sense knowledge by the machine is another challenging research area and we did not consider it in this paper. We also observed that some clauses are ambiguous even for human coders.

4.4. Limitations

We have several limitations in this study. First, automatically separated clauses and tagged aspect data were validated by manual coders to reduce possible errors from clause separation and aspect detection processes. We plan to improve our algorithms to remove the manual steps. Second, the system has a limited number of sentiment analysis rules, and it caused various errors. So we plan to mine more specialized rules using a machine learning approach, which could handle various complex expressions in clauses. Third, we conducted evaluation experiments with a relatively small dataset of 2,700 clauses (900 for each class) because manual tagging requires a great deal of human time and effort. We plan

to prepare a larger dataset to validate the effectiveness of the proposed algorithm.

5. CONCLUSION

We have applied the proposed approach to health and medical domains, particularly focusing on public opinions of drugs in various aspects. Experimental results show the effectiveness of the proposed approach, and it performed significantly better than the machine learning approach with SVM. Since there are still various issues as discussed in the Error Analysis section, we plan to continue improving our linguistic algorithm. For instance, we plan to handle modality, and the developed new rules will be used to detect whether facts described in clauses happened or not. In addition, additional rules will be mined and added by using a machine learning approach, and we will improve aspect detection and clause separation algorithms.

REFERENCES

- Aronson, A. R., & Lang, F. M. (2010). An overview of MetaMap: Historical perspective and recent advances. *Journal of American Medical Informatics Association (JAMIA)*, 17, 229-236.
- Baccianella, S., Esuli, A., & Sebastiani, F. (2010). SentiWordNet 3.0: An enhanced lexical resource for sentiment analysis and opinion mining. *Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC 2010)* (pp. 2200–2204).
- Bodenreider, O., & McCray, A. T. (2003). Exploring semantic groups through visual approaches. *Journal of Biomedical Informatics*, 36, 414-432.
- de Marneffe, M.-C., MacCartney, B., & Manning, C. D. (2006). Generating typed dependency parses from phrase structure parses. *Proceedings of the 5th International Conference on Language Resources and Evaluation* (pp. 449-454).
- Denecke, K. (2008). Accessing medical experiences and information. *Proceedings of the Workshop on Mining Social Data, European Conference on Artificial Intelligence*.
- Ding, X., Liu, B., & Yu, P. S. (2008). A holistic lexi-

- con-based approach to opinion mining. *Proceedings of the International Conference on Web Search and Web Data Mining* (pp. 231-240). New York: ACM.
- Huang, S., Niu, Z., & Shi, C. (2014). Automatic construction of domain-specific sentiment lexicon based on constrained label propagation. *Knowledge-Based Systems*, 56, 191-200.
- Jaloba, A. (2009). The club no one wants to join: Online behaviour on a breast cancer discussion forum. *First Monday*, 14(7).
- Jo, Y., & Oh, A. H. (2011). Aspect and sentiment unification model for online review analysis. *Proceedings of the Fourth International Conference on Web Search and Data Mining (WSDM)* (pp. 815-824). Hong Kong.
- Kim, S., Zhang, J., Chen, Z., Oh, A., & Liu S. (2013). A hierarchical aspect-sentiment model for online reviews. *Proceedings of the Twenty-Seventh AAAI Conference on Artificial Intelligence* (pp. 526-533).
- Liu, B. (2012). *Sentiment analysis and opinion mining*. San Rafael, CA: Morgan & Claypool.
- Lu, Y., Castellanos, M., Dayal, U., & Zhai, C. X. (2011). Automatic construction of a context-aware sentiment lexicon: An optimization approach. *Proceedings of the 20th International Conference on World Wide Web (WWW 2011)* (pp. 347-356). Hyderabad, India.
- Na, J.-C., Kyaing, W. Y. M., Khoo, C., Foo, S., Chang, Y.-K., and Theng, Y.-L. (2012). Sentiment classification of drug reviews using a rule-based linguistic approach. *Proceedings of ICADL (International Conference on Asian Digital Libraries) '2012* (pp. 189-198). Taipei, Taiwan.
- Nikfarjam, A., & Gonzalez, G. H. (2011). Pattern mining for extraction of mentions of adverse drug reactions from user comments. *Proceedings of AMIA Annual Symposium* (pp. 1019-1026).
- Niu, Y., Zhu, X., Li, J., & Hirst, G. (2005). Analysis of polarity information in medical text. *Proceedings of the American Medical Informatics Association Symposium (AMIA)* (pp. 570-574).
- Palmer, F. R. (2001). *Mood modality*. 2nd Edition. New York: Cambridge University Press.
- Pang, B., & Lee, L. (2008). Opinion mining and sentiment analysis. *Foundations and Trends in Information Retrieval*, 2(1-2), 1-135.
- Pang, B., Lee, L., & Vaithyanathan, S. (2002). Thumbs up? Sentiment classification using machine-learning techniques. *Proceedings of the 2002 Conference on Empirical Methods in Natural Language Processing* (pp. 79-86).
- Polanyi, L., & Zaenen, A. (2006). Contextual valence shifters. In J. G. Shanahan, Y. Qu, & J. Wiebe (Eds.), *Computing attitude and affect in text: Theory and applications* (pp. 1-10), Information Retrieval Series Volume 20. Dordrecht: Springer Netherlands.
- Qiu, G., Liu, B., Bu, J., & Chen, C. (2009). Expanding domain sentiment lexicon through double propagation. *Proceedings of the 21st International Joint Conference on Artificial Intelligence* (pp. 1199-1204). San Francisco: Morgan Kaufmann.
- Quirk, R., Greenbaum, S., Leech, G., & Svartvik, J. (1985). *A comprehensive grammar of the English language*. London: Longman.
- Sarasohn-Kahn, J. (2008). The wisdom of patients: Health care meets online social media. California Healthcare Foundation, Oakland. Retrieved from <http://www.chcf.org/publications/2008/04/the-wisdom-of-patients-health-care-meets-online-social-media>.
- Schraefel, M. C., White, R. W., André, P., & Tan, D. (2009). Investigating Web search strategies and forum use to support diet and weight loss. *Proceedings of the 27th international conference extended abstracts on Human factors in computing systems (CHI '09)* (pp. 3829-3834).
- Shaikh, M. A. M., Prendinger, H., & Ishizuka, M. (2008). Sentiment assessment of text by analyzing linguistic features and contextual valence assignment. *Applied Artificial Intelligence*, 22(6), 558-601.
- Stone, P. J., Dunphy, D. C., Smith, M. S., & Ogilvie, D. M. (1966). *General Inquirer: a computer approach to content analysis*. Cambridge, MA: MIT Press.
- Tsytsarau, M., Palpanas, T., & Denecke, K. (2011). Scalable detection of sentiment-based contradictions. *Proceedings of the First International Workshop on Knowledge Diversity on the Web*.
- Wiebe, J., & Riloff, E. (2005). Creating subjective and objective sentence classifiers from unannotated texts. *Proceedings of Sixth International Conference on Intelligent Text Processing and Computational Linguistics (CICLing)* (pp. 486-497).
- Wilson, T., Wiebe, J., & Hoffmann, P. (2005). Recogniz-

ing contextual polarity in phrase-level sentiment analysis. *Proceedings of the Conference on Human Language Technology and Empirical Methods in Natural Language Processing* (pp. 347-354).

- Wilson, T., Wiebe, J., & Hoffmann, P. (2009). Recognizing contextual polarity: An exploration of features for phrase-level sentiment analysis. *Computational Linguistics*, 35(3), 399-433.
- Xia, L., Gentile, A. L., Munro, J., & Iria, J. (2009). Improving patient opinion mining through multi-step classification. *Lectures Notes in Artificial Intelligence*, 5729, 70-76.