Unsupervised Approach to Extracting Problem Phrases from User Reviews of Products

Elena Tutubalina[1] and **Vladimir Ivanov**[1,2,3]
[1]Kazan (Volga region) Federal University, Kazan, Russia
[2]Institute of Informatics, Tatarstan Academy of Sciences, Kazan, Russia
[3]National University of Science and Technology "MISIS", Moscow, Russia
{tutubalinaev,nomemm}@gmail.com

Abstract

This paper describes an approach to problem phrase extraction from texts that contain user experience with products. In contrast to other works, we propose a straightforward approach to problem phrase extraction based on syntactic and semantic connections between a problem indicator and mentions about the problem targets. In this paper, we discuss (i) grammatical dependencies between the target and the problem indicators and (ii) a number of domain-specific targets that were extracted using problem phrase structure and additional world knowledge. The algorithm achieves an average F1-measure of 77%, evaluated on reviews about electronic and automobile products.

1 Introduction

Automatic analysis of reviews can increase information about product effectiveness. This is especially important to a company if the information can be obtained with minimal costs. Customers write reviews regarding product issues that are too difficult to handle without technical support.

In this paper, we present a study about connections between a product (the target of a problem phrase) and words (problem indicators), describing unexpected situations specific to products. We define problem indicators as words describing phrases that contain obvious links to a problem (e.g., *problem*, *issue*). We also define problem indicators as words that mention implicit problems (e.g., *after*, *sometimes*). The problem indicator may be presented as an action verb with a negation expressing product failure.

The task is to identify which noun phrases (NPs) referred to the problem target in the sentence. The task is divided into two subtasks: (1) identify what phrase potentially contains information about a problem and (2) find possible targets using the set of nouns for a given problem expression. The first subtask, problem phrase identification, determines whether a given sentence contains problem phrases. The second subtask, target phrase extraction, identifies the targets of a given problem phrase.

The problem indicators are significant for problem sentences where the device doesn't work correctly. However, the presence of indicators in the sentence may have insufficient context to determine the problem's existence. In examples 1 and 2, the object that receives the action isn't defined ("I have not seen one", "something hasn't been right").

1. After looking for months, I have not seen one that I like at the local store and have expanded to other local stores.
2. Something hasn't been right for sometimes now - we have advised the support staff several times about this issue.

We present the dependency-based approach for extracting problem phrases and its target from user reviews of products. We suppose that problem indicators have a syntax connection with the target of a problem phrase. Domain-specific targets are extracted by determining a domain category of related target words in WordNet.

Proceedings of the AHA! Workshop on Information Discovery in Text, pages 48–53,
Dublin, Ireland, August 23 2014.

The rest of the paper is organized as follows. In section 2, we introduce related work in different areas of text classification of reviews and target detection. Section 3 describes our approach and classifies dependency relations between problem indicators and targets of problem phrases. We present experimental results in section 4. Section 5 presents our conclusions and future extensions of this work.

2 Related Work

Several different methods have been proposed in literature for the classification of product reviews in different areas of research. The primary area of related works is sentiment analysis. Turney (2002) presented an unsupervised learning algorithm for classifying reviews from a consumer platform on the Web. The author extracted phrases containing adjectives or adverbs from reviews of different domains. His algorithm achieved 74% accuracy for 410 reviews sampled from four different domains. Popescu and Etzioni (2007) focused on entity-level classification using extraction rules on sentences with features or entities. They proposed a method for opinion phrase extraction based on the semantic orientation of words in the context of product features. Authors commonly focused on explicit product features from noun phrases. They reported a recall of 89% and a precision of 86% for opinion phrase polarity determination with known product features and a precision of 79% and a recall of 76% for product feature identification. Hu et al. (2004) extracted sentences that contained one or more product features and identified the polarity of the opinion sentences using the adjective set from WordNet.

Another group of related works explores extracting information about subjectivity. Wiebe and Wilson (2002) recognized opinionated and evaluative (subjective) language in text. According to them, a sentence is subjective if it contains a significant expression of emotion, opinion, or an idea about why something has happened. Wilson et al. (2004) used dependency relations classifying the subjectivity of deeply nested clauses for their task of classifying the strength of the opinions. Breck (2007) used semi-supervised sequence modeling by conditional random fields (CRF) for entity-level opinion expressions. The best F1-measure (70.65%) was achieved for identifying opinion expressions.

There is much research on finding targets of objects using dependency relations (Quadir (2009); Lu (2010); Qiu et al. (2011)). Qadir (2009) identified opinion sentences that were specific to product features. The words forming the dependency relations were analyzed for frequent product feature. Qadir tagged each sentence with only one product feature. In contrast, we propose that problem phrases have multiple targets. Qiu et al. (2011) extracted sentiment words and aspects by using syntactic relations. They described a propagation approach that achieved a recall of 83% and a precision of 88%.

Problem detection and extraction of problem phrases from texts are less studied. We used a clause-based approach to problem-phrase extraction from user reviews of products. This method is based on dictionaries and rules and performed well compared to the simple baseline given by supervised machine learning algorithms. They achieved a recall of 77% and a precision of 74% for user reviews about electronic products. The current task of this research is identifying the targets of problem phrases to reduce classification errors. Gupta (2013) studied the extraction of problems with AT&T products and services from English Twitter messages. The author used a supervised method to train a maximum entropy classifier. Gupta reported the best performance F1-measure of 75% for identification of problem target. In contrast, our method is based on grammatical domain-independent relations in a sentence.

3 Target Extraction

In this section, we describe our method for extracting problem phrases, related to targets, from customer reviews. PW is an initially empty set of the problem indicators, T is an empty target set. The common approach starts from an initially empty set of pairs (problem indicator, target), $PTWs = (PW, T)$, where $PW = T = \emptyset$. The approach is divided into three steps: problem-phrase identification, target extraction, and domain-specific target detection. After three steps, the algorithm marks the sentence as a problem sentence if at least one pair (indicator, target) is extracted (i.e. $PTWs = \{(pw_1, t_1), (pw_2, t_2), ...\}$, $pw_i \in PW, t_i \in T$). We describe problem-phrase identification in section 3.1. Section 3.2 describes dependency relations for target extraction. Section 3.3 explains identifying domain-specific targets using semantic knowledge from WordNet.

3.1 Problem Phrase Identification

A common appoach to problem-phrase extraction uses problem indicators (i.e., words, that indicate a problem in a sentence). We briefly describe the manually created ProblemWord dictionary, defined in Ivanov and Tutubalina (2014). The ProblemWord dictionary includes problem indicators such as *problem*, *error*, *failure*, *malfunction*, *crash*, *break*, *reboot*, *have to replace*, etc. We collected synonyms for these problem indicators. The manually created dictionary consists of about 300 terms.

Problem-phrase identification. In this step, the algorithm looks for the problem indicators in the sentence. The algorithm looks for verb phrases headed by an action verb with a negation or looks for problem words from manually created dictionaries (without related negations). As a result, the set PW is collected, which includes all problem indicators $pw_i \in PW$, found in the sentence S.

3.2 Dependency Relations for Target Extraction

The method for target extraction uses syntax dependencies to determine connections between possible targets and problem indicators. The phrase describing the problem is a combination of the target and the problem indicators. The structure of a sentence with selected collapsed dependencies is shown in Figure 1. We use the Stanford typed dependencies from the Stanford parser[1]. For this sentence, the Stanford dependencies representation with the selected problem indicator is $\{$*dobj(use, firefox), nn(software, printer), prep_with(use, software), vmod(software, added), pre_to(added, computer)*$\}$. These dependencies are written as relation_type(head, dependent), where the head and the dependent have a dependency direction associated with them in the sentence.

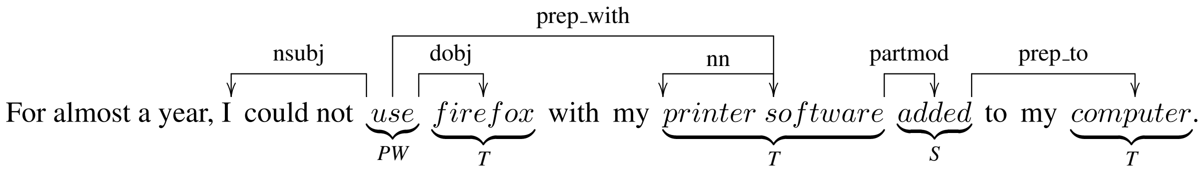

Figure 1: The syntactic structure of a sentence with collapsed dependencies.

To identify connections between the problem indicator and the targets of problem phrase, we use direct and indirect dependency relations between two words that are defined in (Qiu et. al., 2011). The intuition behind a choice of these types is that only syntactically and semantically rich mentions of targets are extracted, reducing noise in the extracted set of nouns. The direct dependencies allow for recognizing some relations directly between the target and the problem indicator. Indirect dependency indicates that one word depends on the other word through some additional words, which we call *successors*. A successor word connects to a problem indicator and replaces a problem indicator in relation with a target. The connection of the successor with the problem indicator and the target can possibly indicate the relation between the indicator and the target of the problem phrase through context.

Sentences 3–4 show examples with direct and indirect dependency relations taken from the review sentences. PW refers to a problem indicator, and T refers to a target of a problem phrase, S refers to a successor of a problem indicator. In example 3, *give* is the action verb with the related negation, which indicates the problem and has a direct connection with the target *stars*. Example 4 contains problems with the *cord*, and there is no problems with domain-specific products in example 3.

nsubj dobj

3. The only reason I didn't $\underbrace{give}_{PW}$ it five $\underbrace{stars}_{T}$ is because I am unfamiliar with any of the bottles.

nsubj xcomp dobj

4. They still don't $\underbrace{allow}_{PW}$ you to $\underbrace{unwind}_{S}$ the $\underbrace{cord}_{T}$ while it's plugged in the cord would twist.

[1]`http://nlp.stanford.edu/software/lex-parser.shtml`

Target extraction. In second step, for each problem indicator $pw_i \in PW$ the approach extracts all related targets. The targets are found in the dependencies representation of the sentence S. The target is related to the problem word if there is direct or indirect dependency between the target word and the problem indicator or the problem indicator's successor, respectively, in the sentence representation. We add to the set $PWTs$ one $pair(pw_i, t_j)$ for each target t_j, i.e., $PWTs = PWTs \cup \{pair(pw_i, t_j)\}$, $T = T \cup \{t_j\}$.

3.3 Domain-Specific Target Detection Using WordNet Categories

Domain-specific targets can be extracted by using additional world knowledge. Domain-specific targets are objects that have important meanings in a particular domain. The method uses WordNet for choosing domain-specific targets. WordNet organizes related words in synsets as synonym sets. It extracts domain categories from WordNet for selected problem targets in the sentence. WordNet assigns multiple domain semantic labels to terms in the domain. The method extracts categories for each target and its hypernyms, hyponyms, and holonyms. A hyponym is a lexical relation between a more general term and a more particular term (e.g., *machine/computer*). A hypernym is a word that is more generic than a given entity (e.g., *portable computer/laptop*). A holonym is a term that denotes a complete object whose part is denoted by given word (e.g., *computer/keyboard*).

Figure 2 shows the relations that classified targets into WordNet domains. Each target from the problem phrase is represented by a row in the first table. Hyponym, hypernym and holonym relations are drawn between rows in the tables. Dashed arrows are represented type-of relations, and solid arrows are represented part-of relations between the target and the WordNet synset. Solid lines are represented links to the WordNet domain category. In the example, the target *TouchPad* has an unknown category.

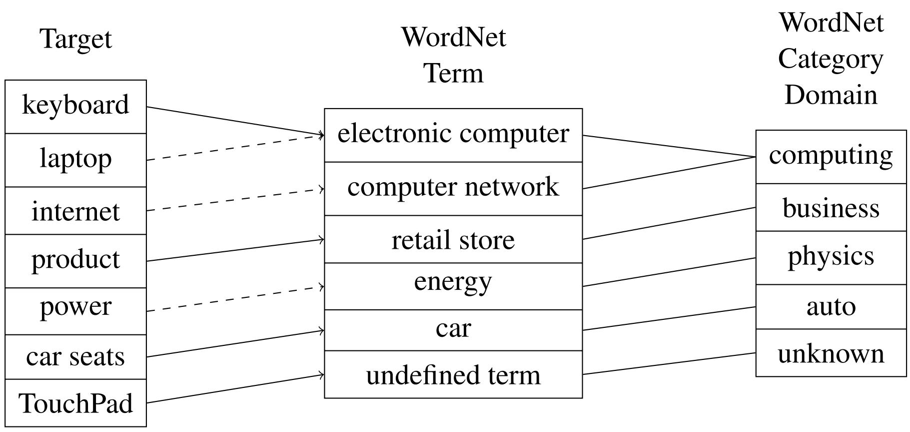

Figure 2: The example of relating targets into WordNet categories

Domain-specific target detection. In the third step, the algorithm reduces pairs with non-domain-specific problem targets from the set $PWTs$ using semantic knowledge. The pair $pair(pw_i, t_j)$ is reduced if there is no term with a domain category in the hierarchy of WordNet synsets induced by the hypernym, hyponym, or holonym relations with target t_j.

4 Evaluation and Discussion

For our experiments we collected 734 sentences from the HP website[2]. We employed 953 sentences from Amazon reviews[3] about automobile products. Of the total sentences, 1,288 sentences (506 + 782 from electronic and automobile domains) were classified as problem sentences, and 399 (228 + 171) sentences were labeled as part of the no-problem class. Class labels for each sentence were obtained by using the Amazon Mechanical Turk service. Each sentence does not have any particular label for targets and contains at least one problem indicator that the approach can find. We propose that the problem

[2] `http://reviews.shop.hp.com`

[3] The dataset is available at `https://snap.stanford.edu/data/web-Amazon.html`

phrase with the problem indicator always has targets, but not necessarily domain-specific targets. For our performance metrics, we view that a text classification task is to identify whether a target is a domain-specific problem target. We computed precision (P), recall (R), accuracy (Acc.) and F1-mesure (F1).

Type of targets	Examples
Domain-specific	internet, printer, monitor, screen, laptop, processor, driver
Undefined category	Apple, HP printer, win7, printhead, adapter, reboots
Other targets	marketing, stars, box, unit, letters, shipment, results

Table 1: Problem targets related to WordNet categories.

Table 1 gives examples of the problem targets that are extracted in connection with problem indicators and are related to the categories. The total number of targets extracted from 506 problem sentences about electronic products were 258 domain-specific targets, 458 targets without a WordNet category and 138 other targets. The approach collected 151 compound targets (e.g., *auto configuration*, *network settings*), that were related to the undefined category.

We used different methods to compare the performance of our approach, as follows:

1. We considered the targets extracted by direct dependencies (we did not use any lexical knowledge).
2. We considered all targets extracted by direct and indirect dependencies as domain-specific targets.
3. We considered only domain-specific targets extracted by direct and indirect dependencies. We extracted the targets, if the selected target was a term related to computing, electronic, or automobile terminology from WordNet. We also considered the targets that don't relate to any WordNet category and did not have a lexical meaning such as "time" or "person".

Method name	Electronics				Cars			
	P	R	Acc.	F1	P	R	Acc.	F1
Direct Dependencies	.74	.48	.53	.58	.83	.67	.62	.74
+ Indirect Dependencies	.73	.90	.71	.81	.83	.92	.78	.87
+ WordNet caterogies	.74	.71	.62	.72	.84	.79	.70	.81

Table 2: Performance metrics of the dependency-based approach.

Performance metrics are calculated with this dataset and provided in able 2. The average recall and the average precision of problem-phrase extraction related to domain-specific targets are 75% and 79%, respectively. WordNet is limited and does not include many proper nouns (e.g., *MacBook*, *Honda Odyssey*) related to a particular domain. Thus, domain-specific target detection using WordNet categories led to a decrease in the average recall (from 91% to 75%). Another type of sentence, that decreases the average recall, is user sentences with a situation description, not just a report about a problem ("Something hasn't been right for sometimes now", for example).

5 Conclusion

In this paper, we aim to identify problem phrases and to connect the proper targets of phrases with problems or difficult situations. Without using domain-specific knowledge about products, we focus our attention on dependency-based syntactic information between the target and the problem indicators in the text. We use WordNet synsets with domain labels to reduce targets that aren't related to a product domain. The average value of the F1-measure (about 84% for all targets and 77% for domain-specific targets) is better than the F1-measure in Ivanov and Tutubalina (2014). Our research shows that dependencies and syntactic information combine with each other to collect different parts of problem targets for target phrase extraction. For future work, we plan to extend the evaluation corpus with new domains to explore more kinds of syntactic information between problem indicators and the dependent context.

References

Turney P. D. 2002. *Thumbs up or thumbs down?: semantic orientation applied to unsupervised classification of reviews*. Association for Computational Linguistics, 417–424.

Popescu A. M., Etzioni O. 2007. *Extracting product features and opinions from reviews*. //Natural language processing and text mining, Springer, London, 9–28.

Hu M., Liu B. 2004. *Mining and summarizing customer reviews*. Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining, 168–177.

Wiebe J., Wilson T. 2002. *Learning to disambiguate potentially subjective expressions*. Proceedings of the 6th conference on Natural language learning-Volume 20, Association for Computational Linguistics, 1–7.

Wilson T., Wiebe J., Hwa R. 2004. *Just how mad are you? Finding strong and weak opinion clauses*. Aaai, 761–769.

Breck E., Choi Y., Cardie C. 2007. *Identifying Expressions of Opinion in Context*. IJCAI, 2683–2688.

Narenda Gupta. 2013. *Extracting phrases describing problems with products and services from twitter messages*. Conference on Intelligent Text Processing and Computational Linguistics, CICling2013.

Ivanov V., Tutubalina E. 2014. *Clause-based approach to extracting problem phrases from user reviews of products*. Proceeding of the 3d International Conference on Analysis of Images, Social Networks, and Texts AIST'14, Russia, 2014.

Lu B. 2010. *Identifying opinion holders and targets with dependency parser in Chinese news texts*. Proceedings of the NAACL HLT 2010 Student Research Workshop. Association for Computational Linguistics, 46–51.

Qadir A. 2009. *Detecting opinion sentences specific to product features in customer reviews using typed dependency relations*. Proceedings of the Workshop on Events in Emerging Text Types. Association for Computational Linguistics, 38–43.

G. Qiu, B. Liu, J. Bu, C. Chen 2011. *Opinion word expansion and target extraction through double propagation*. Computational linguistics, 9–27.