Extracting Phrases Describing Problems with Products and Services from Twitter Messages

Narendra K. Gupta

AT&T Labs - Research, Inc.,
Florham Park, NJ 07932,
USA

ngupta@research.att.com

Abstract. Social media contain many types of information useful to businesses. In this paper we discuss a trigger-target based approach to extract descriptions of problems from Twitter data. It is important to note that the descriptions of problems are factual statements as opposed to subjective opinions about products/services. We first identify the *problem tweets* i.e. the tweets containing descriptions of problems. We then extract the phrases that describe the problem. In our approach such descriptions are extracted as a combination of *trigger* and *target phrases*. Triggers are mostly domain independent verb phrases and are identified by using hand crafted lexical and syntactic patterns. Targets on the other hand are domain specific noun phrases syntactically related to the triggers. We frame the problem of finding target phrase corresponding to a trigger phrase as a ranking problem and show the results of experiments with maximum entropy classifiers and voted perceptrons. Both approaches outperform the rule based approach reported before.

Keywords. Social media, information extraction, text classification.

Extracción de frases que describan problemas con productos y servicios de mensajes Twitter

Resumen. Medios sociales de comunicación contienen muchos tipos de información útil para las empresas. En este artículo se considera un enfoque orientado al método de “desencadenante-objetivo” para extraer descripciones de problemas de los datos de Twitter. Es importante mencionar que las descripciones de problemas son declaraciones de hechos a diferencia de opiniones subjetivos acerca de productos/servicios. En primer lugar se identifican los tweets de problema, es decir los tweets que contienen descripciones de problemas. En el enfoque propuesto tales descripciones se extraen como una combinación de frases de desencadenante y objetivo. Desencadenantes son en su mayoría frases verbales independientes del dominio y se identifican mediante patrones léxicos y sintácticos creados manualmente. Por otro lado, objetivos son frases nominales específicas del dominio particular y sintácticamente relacionadas con las desencadenantes. Se ataca el problema de encontrar la frase objetivo correspondiente a la frase desencadenante dada como un problema de ranking y se presentan los resultados de experimentos con clasificadores de máxima entropía y perceptrones de votación. El rendimiento de ambos enfoques es mejor que el del enfoque basado en reglas reportado anteriormente.

Palabras clave. Medios sociales de comunicación, extracción de información, clasificación de textos.

1 Introduction

In the past decade social media have become quite popular and a rich source of information. Twitter is one of the popular on-line social media. Whenever some event takes place, people make many tweets about it in real-time. They also talk about their personal experiences and preferences. This rich and real time source of information can also be used by businesses. Besides consumer opinions, businesses could learn the problems consumers face with their products/services, the questions they may have, or even specific information they may have discovered in real time. Examples of tweets containing different types of information for telecommunication products and services are shown in Figure 1.

In this paper we discuss a trigger-target based approach to extract descriptions of problems from Twitter data. It is important to note that the descriptions of problems are factual statements as opposed to subjective opinions about products/services. To the best of our knowledge there has been no previous work to extract this type of information from Twitter data. Although our method can be used to extract similar

Opinion Tweet: `Thank God tmobiles roaming partner is at&t.`
Problem Tweet: `can't place or receive calls on the network now.`
Question Tweets: `Hey AT&T. When are you going to let me tether?`
Information Tweets: `AT&T doesn't serve our condo community.`

Fig. 1. Examples of tweets containing different types of information

information from other textual data sources, like blogs and product reviews, we chose to focus on Twitter because: a) the limited size of tweets ensures that all the information is contained in a single sentence and very rarely co-reference resolution is needed, and b) information in tweets is real-time. Its quick discovery can allow businesses to take timely action and protect their brand reputation effectively.

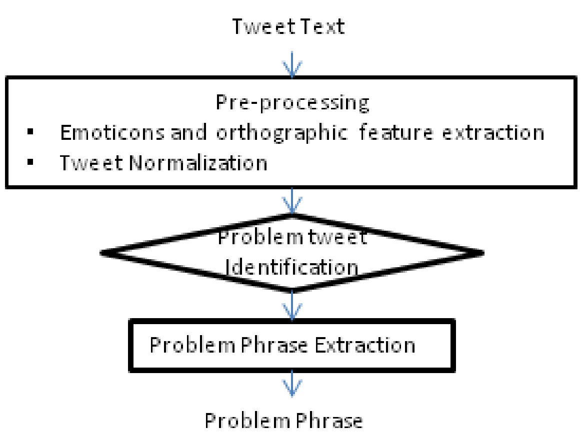

Fig. 2. The overall architecture of our solution

As shown in Figure 2, besides tweet normalization[1], our overall method consists of 2 major steps. In the first step tweets containing a description of some problem, henceforth referred to as *problem tweets*, are identified through text classification. We have identified several lexical and syntactic patterns specific to sentences describing problems. Most instances of these patterns are domain independent verb phrases. In this paper we will refer to them as *trigger phrases*. For example in `'my internet has stopped working'` the trigger phrase is `'stopped working'`. It indicates a problematic state of `'my internet'`. In [8] we have shown that the use of binary features indicating presence/absence of different types of trigger phrases significantly improves the classification of problem tweets.

In the second step, phrases describing the problem, henceforth referred to as *problem phrases*, are extracted from the problem tweets. This is the main focus of this paper. We extract problem phrases as a combination of the trigger phrases and related *target* noun phrases. In the above example `my internet` is the target phrase of the trigger `stopped working` and the problem phrase is a combination of target and trigger phrases i.e. `my internet stopped working`. Target phrases refer to domain dependent objects but have domain independent syntactic relationships with trigger phrases. In [8] we have shown that using simple syntactic pattern matching to extract target phrases yields practically usable results. In this paper we describe a data driven approach using products and services offered by AT&T as an example. Given a trigger phrase we rank the noun phrases of the sentence with respect to their appropriateness to be the target of a trigger phrase, and select the top ranking potential target. We show the results of our experiments with voted percepton [1] and maximum entropy classifiers. Both outperform the rule based approach reported in [8].

The rest of this paper is organized as follows. In Section 2 we discuss the relationship of our work with other natural language processing work in the context of social media data. In Section 3 we discuss the problem tweet identification through classification. In Section 4 we describe the problem phrase extraction. In Section 5 we describe our experiments and their results. We conclude the paper in Section 6 by summarizing the contributions of this work.

2 Relation to Prior Work

To best of our knowledge there exist only a few research papers[2] on extracting problem descriptions from textual data. Extracting problem descriptions from the Japanese Web document

[1] To pack information in limited-size messages, the language used in tweets is distorted. As a result, NLP tools trained on standard text corpora become unusable. Tweet normalization removes these distortions to get reasonable outputs from NLP tools.

[2] Thanks to the reviewers for the reference.

reported in [21] is very similar to our work. To identify trouble expression they use supervised text classification, and like us they also use manually encoded syntactic features. Unlike the syntactic patterns used in our work, their syntactic features are based on synonyms of word "trouble" and post-positions in Japanese. To identify the troubled object they also use a ranking procedure based on co-occurrences of the trouble expressions and the objects in the training data. In our approach we use various syntactic relationships between triggers and targets to predict trigger-target relationship, as a result, objects not seen in training data can also be identified as target.

Mining social media for opinions about products/services [7, 16], or about political candidates or even about some policy is very useful. In [3, 13, 18], the polarity (positive, negative and neutral) of opinions are found through text classification. This is similar to finding problem tweets through text classification. Since negative opinions/emotion[3] are often expressed in problem tweets, we use some of the same features for finding problem tweets as are used in opinion mining.

Text classification approaches for opinion mining make no attempt to ensure that opinions are directed towards the object of interest. Solution to this problem can be framed in terms of our trigger-target approach where opinion phrases act as triggers and those referring to the targets of opinions are the target phrases. For opinion mining problem also a large proportion of trigger phrases are domain independent while almost all target phrases are domain dependent but are syntactically related to trigger phrases in a domain independent manner. Several researchers have addressed this problem. Hu and Liu [10] use a simple heuristic and assume adjectives to be the opinion words (triggers) and nearby noun phrases to be their target. Jiang et al. [11] do not attempt to extract opinions and target phrases; instead, given a target, they generate the polarity of target specific opinions by using target specific syntactic features in the classifier model. Liu and Seneff [14], on the other hand, generate reviewer's ratings (strengths of opinions) for different topics. The first step of their algorithm is to extract descriptor-topic (opinion-target) pairs from a large corpus of reviews. To do this they also resort to hand coded syntactic rules. We have not experimented with our ranking approach for opinion mining, however in this paper we demonstrate its superiority over a rule based approach for extracting problem descriptions.

[3]Tweets containing negative opinion/emotions do not always accompany problem descriptions

Our task is closer to Semantic Role labeling [5] where given a predicate its arguments are extracted from within a sentence. In our case given a trigger we also would like to extract its targets from within a sentence. The typical solution to semantic role labeling involves identification and classification of a predicate's arguments. This is done by classifying internal nodes [19] of a syntactic parse tree. To identify target phrases we only focus on the noun phrase (NP) nodes since the targets are typically noun phrases.

For our task, NLP tools like sentence splitters and a syntactic parser are required. Since the language used in tweets is significantly distorted, NLP tools trained on standard text corpora perform poorly. Researchers have started focusing on Twitter specific tools like part of speech tagging [6], syntactic chunking [20] and even named entity extraction [20, 15]. Even though these tools perform better than those trained on the standard text corpora, non-standard language in the tweets inhibits their performance. To achieve better results therefore, researchers have also started to focus on tweet normalization [12, 9, 2]. In our work we have chosen to first normalize the text and use NLP tools trained on standard text corpora. Since we did not have access to tweet normalization software and since it was not the focus of our work, we use a simple text normalization approach based on table look-ups. Results presented in this paper will certainly improve with a more sophisticated and robust text normalization approach.

3 Problem of the Tweet Identification

For the problem of the tweet identification we trained a Maximum Entropy classifier [17]. In [8] we have shown that in addition to word ngrams, using the sentiment and a few well chosen syntactic features of the tweets improves the problem tweet identification F-measure from 0.66 to 0.742. The intuition behind using the sentiment features is that users having problems with some product/service often express negative sentiments. We use presence of a) emoticons,

b) repeated punctuation, and c) dictionary phrases expressing positive/negative sentiment in the tweets as sentiment features. In the rest of this section we describe the syntactic features used in classification model. These features were selected through manual analysis of a few hundred tweets.

3.1 Happening Verbs

A common way to describe the problem is by explaining what is happening. Such a description will have a verb phrase with specific head verbs. For example in `My network crashes frequently` the head of the verb phrase (VP) is `crashes` and in `preorder system is broken` it is `broken`. Some verbs are of general nature and can be used across multiple domains to describe a problem, while others are domain specific. For example the verb `freeze` in `My computer freezes often` is specific to the computer domain. We use 9 such verbs[4], we call "Happening verbs", that we believe with high probability[5], are used across the domain to describe a problem. These are `fail`, `crash`, `overload`, `trip`, `fix`, `mess`, `break`, `overcharge` and `disrupt`. To extract this feature, we test if the tweet contains a verb phrase headed by any of these verbs.

3.2 Not Happening Verbs

Another very common way to describe the problem is by explaining what is not happening; for example "`my internet is not working`" or "`my internet has stopped working`" etc. We identified 9 such verbs we call "Not happening verbs". These are `work`, `function`, `connect`, `get`, `perform`, `receive`, `send`, `run`, and `respond`. These verbs are often used, along with negation, or with the verbs `stop`, `refuse` or `cease` to describe what is not happening.

To extract this feature, we test if the tweet contains a verb phrase headed by any one of the not happening verbs and either the negation of the not happening verb, i.e. `neg(verb, not)`[6] or an open clausal complement dependency on `stop`, `refuse` or `cease`, i.e.,`xcomp(stop|refuse|cease, verb)` is present.

[4]These are obtained through manual analysis of a few hundred tweets and by no means are a complete set.

[5]This is based on the authors subjective judgment.

[6]In this paper we will use the dependencies generated by Stanford parser. See [4] for details.

3.3 Softer Verbs

These are like the happening verbs but are very often also used in contexts other than problem description. To extract this feature, we test if the tweet contains a verb phrase with head verb being one of `die, drop, bite, f**k, trouble, foil`.

3.4 Problem Noun

Problems are also described through noun phrases with specific head nouns. For example in `we have an internet failure` the head of the noun phrase is `failure` and in `we are having a 3G outage` it is `outage`. T o extract this feature, we test if the tweet has a noun phrase headed with one of the problem nouns. We used `crash`, `failure`, `issue`, `outage`, `problem` and `trouble` as problem nouns.

3.5 Phrase

Phrases are frequently used to describe the problems; for example `screwed up`, `hang up`, `knock off`, `knocked out`, `acting up` etc. In these phrases propositions act as the particle of the verb. To identify such phrases we check for a `prt(verb, proposition)` dependency between the verbs `f**k`, `hang`, `screw`, `cut`, `knock`, `act` and the propositions `up`, `off`, `out`.

Besides phrases with particles, there are other phrases that are also used to describe problems. See [8] for details.

4 Problem Phrase Extraction

In our trigger-target approach, the problem phrases are composed trigger and target phrases. Triggers are instances of syntactic patterns described in the previous section. They indicate that something is not working as desired. Their presence triggers the identification of the target phrases referring to the object that are not working. In [8] we described a rule based approach for target phrases identification. In the rest of this section we describe a data driven approach and show improvement in its performance over that of the rule base approach.

4.1 Data Driven Target Phrase Extraction

As in semantic role labeling, we classify internal nodes of the syntactic parse tree in the context of known triggers. Since targets can only be noun phrases, instead of classifying all internal nodes, we only focus on NP nodes. Specifically we rank the NP nodes with respect to their appropriateness of being the target of a trigger phrase, and select the top ranking noun phrase. We used two different methods to perform this ranking i.e., the voted perceptron [1] algorithm and maximum entropy classification.

We experimented with voted perceptrons because they have been effectively used for semantic role labeling [22] and they also posses some very desirable properties [1] i.e., 1) if there is a parameter vector $\mathbf{U}$ which makes no error on the training data then the algorithm converges to that in finite iterations; 2) the number of mistakes the algorithm makes is independent of the number of candidates to choose from for each example. It only depends on the separability of the data; 3) in contrast to classification approaches, using even low frequency features in the training data can yield better results.

For each trigger phrase x^i there could be n_i candidate noun phrases in a sentance. We indexed these from 1 to n_i. For notational convenience, in the training algorithm, the correct target phrase is indexed by 1 [1]. We represent the candidate problem phrase formed by the trigger phrase x^i and the jth target phrase by $x_{i,j}$. Using this notation we outline in Figure 3 the voted perceptron training and decoding algorithms.

To use a maximum entropy classifier to rank target phrases corresponding to each trigger phrase, we trained a binary classifier that outputs the probability $p($*target* $= j|$*trigger* $= x^i)$ for each target, and then select the jth. noun phrase such that $j = \arg\max_{j=1\dots n_i} p($*target* $= j|$*trigger* $= x^i)$. To train such a classifier we generate a set of positive and negative examples. For each trigger x^i, $h(x_{i,1})$ is the positive example and $h(x_{i,j})$ for $2 \leq j \leq n_i$ are the negative examples.

Voted perceptron training.

Input:Training data $(x^i : \{x_{i,1}, x_{i,2} \dots x_{i,n_i}\})$ for $i = 1$ to m.
$x_{i,1}$ is the correct problem phrase for trigger x^i.
$h : x_{i,j} \rightarrow \Re^d$,Feature vector representation of each problem phrase $x_{i,j}$.
Output: d dimensional parameter vectors $\mathbf{w}^0, \mathbf{w}^1, \dots \mathbf{w^m}$ for ranking function $F(x) = \mathbf{w} \cdot h(x)$.

1. set $\mathbf{w}^0 = 0$
2. for $i = 1$ to m
3. $j = \arg\max_{j=1\dots n_i}(\mathbf{w^{i-1}} \cdot h(x_{i,j}))$
4. if$(j \neq 1)\mathbf{w^i} = \mathbf{w^{i-1}} + h(x_{i,1}) - h(x_{i,j})$
5. else $\mathbf{w^i} = \mathbf{w^{i-1}}$

Voted perceptron decoding.

Input:$(x^i : \{x_{i,1}, x_{i,2} \dots x_{i,n_i}\})$ for each trigger x^i
$h : x_{i,j} \rightarrow \Re^d$,Feature vector representation of each problem phrase $x_{i,j}$.
d dimensional parameter vectors
parameter vectors $\mathbf{w}^0, \mathbf{w}^1, \dots \mathbf{w^m}$.
Output: Index j of the correct problem phrase

1. for $i = 1$ to m
2. $j = \arg\max_{j=1\dots n_i}(\mathbf{w^i} \cdot h(x_{i,j}))$
3. $V(j)++$
4. $j = \arg\max_{j=1\dots m} V(j)$

Fig. 3. Voted perceptron training and decoding algorithms

4.2 Manually Tagged Data

For our experiments we used data manually[7] tagged with triggers and the corresponding targets. Each trigger is tagged with a trigger type. It refers to the syntactic pattern expected to identify the trigger automatically. In some cases triggers do not match any of the syntactic patterns, and require deep domain knowledge; for example the tweet `i juss got text messages sent more than 30 mins`. In such cases trigger phrases are tagged with a tag `KN`. Table **??** shows tags for different trigger types. It also shows the counts and percentages of each in the manually tagged data. An important thing to note from this table that only 9% of triggers used in communication related problem descriptions use domain dependent trigger phrases. T he other 91% of trigger phrases are domain independent. Because of this reason we believe models trained for our trigger-target approach in one domain -

[7] Space limitations do not allow us to delineate the annotation process. A kappa value of 0.806 for inter-annotator agreement was obtained.

Table 1. Tags for different trigger types (counts are in tagged data)

Trigger Types	Tag	Counts	Perc.
Happening/Soft Verbs	V	478	35%
Not Happening Verbs	NV	392	29%
Problem Noun	NN	331	25%
Phrases	PH	32	2%
Others (domain specific)	KN	117	9%

Input:*Trigger phrase*, *Trigger type* and *parse tree*
Output:*B*; The constituent node
that best matches the *Trigger phrase*
and its *Trigger type*, (Object B in Table 2)

1. *A*=Lowest common ancestor of the *Trigger phrase* in parse tree (Object A in Table 2)
2. if (*Trigger type* != `NN`)
3. *B*=VP constituent closest to *A* and with head verb in *Trigger phrase*.
4. else *B*= *A*

Fig. 4. Mapping manually tagged trigger phrase on to the parse tree

e.g. communications equipment - can be applied to another domain - e.g. consumer electronics - without significant degradation. In future work we plan to formally quantify this degradation experimentally.

4.3 Used Features

To train the rankers we used the feature sets presented in Table 2. These features are similar in spirit to those used for semantic role labeling [19] and are divided into 4 groups labeled in bold font in Table 2. They are extracted for each pair of trigger and target to be ranked. However to extract them from the manually tagged data for training and testing the models, we must first map the tagged trigger and target phrases on to the parse trees. We use the algorithm described in Figure 4 to map the trigger phrases on to the best constituent node of the parse tree. To map target phrases we find out the best NP node covering the entire target phrase. Objects A and B referenced in this algorithm and in Table 2 are intermediate objects that help in extraction of features of the trigger phrase.

Most of the features listed in Table 2 are self explanatory, still some require some explanation. Feature 17 is generated by checking if the target phrase contains any of the words from a list of key words used to refer to objects in the domain of interest. For example for the communication domain such a list may contain words like `network`, `smart phone`, `telephone`, `cell phone`, `internet`, `modem` etc. In the data driven approach we also take advantage of the rule based approach described in [8]. Specifically we let the rule based approach decide if an NP is a target of the given trigger and use the binary outcome as feature 32.

5 Experimental Results

For our experiments we tagged 2110 previously identified problem tweets with triggers and their targets. We divided this data in training and test set as shown in Table 3. In our first experiment we tested the performance of rule based trigger identification. Results of this test on the entire tagged data set are shown in Table 4. For this test we assumed a match between the triggers if their head words were identical. We obtained an acceptable F-measure of 0.74, but a low precision of 0.65 (35% false positive). Clearly target phrase identification must have some ability to reject these false triggers. To test target phrase identification, therefore besides 206 tagged manually trigger-target pairs, we automatically tagged the test data with 73 false positive triggers identified by the trigger phrase identifier.

Table 5 shows the performance of 4 different algorithms to identify the target phrases corresponding to the trigger phrases in the test data. In this table the column labeled “Error” gives the counts of cases where the manually tagged target does not match the hypothesis. To calculate precision and recall this error count is added to both false positives and false negatives. As a baseline, the first row of this table shows the performance of the rule based target phrase identification described in [8].

Table 3. Tagged data (TTP is Trigger-Target pairs)

Data set	Number of tweets	Number of TTP
Training	1806	1144
Test	304	206
Total	2110	1350

The second row shows that the performance of voted perceptron is slightly better than baseline. The main reason for this poor performance is

Table 2. List of features used for rankers

S.No.	**Description**	**Type**
Features of the Sentence		
1-5	Syntactic parse contains VP, S, SBAR, SQ, FRAG constituent nodes	binary
Features of the Trigger Phrase		
	Object A: Lowest common ancestor of the trigger phrase in parse tree **Object B:** Closest VP node to object A with head one of the verbs in trigger	
6	Syntactic pattern that generated the Trigger	categorical
7	Head word of object B	categorical
8	POS String of the trigger phrase	ngram
9	Head word is in passive form	binary
10	Negation (not, never, none) in trigger vicinity	binary
11	Is object A and B the same	binary
12	xcomp(Head word, ??) or ccomp(Head word, ??) exists	binary
13	xcomp(??, Head word) or ccomp(??, Head word) exists	binary
14	Path from object A to B in Syntactic Parse tree	ngram
Features of the possible targets (NP nodes)		
15	POS String of the target phrase	ngram
16	Negation (not, never, none) in target vicinity	binary
17	Contains words from a predefined list of domain objects	binary
18-23	POS string has a verb, preposition, adverb, adjective ,determiner, pronoun	binary
24	Target contains more than 4 words	binary
25	Appos(Head of target NP, Head of another NP) exists	binary
26	conj_and(Head of target NP, Head of another NP) exists	binary
27	Dominates another NP with the same head	binary
28	Dominated by another NP with the same head	binary
Features of the Trigger and Target phrases		
29	Path from target NP node to object A in Syntactic Parse tree	ngram
30	Path from head of target NP node to head word of object B in dependency tree	ngram
31	Does the path in feature 32 contain S or SBAR	binary
32	Outcome of rule based approach described in [8].	binary

Table 4. Performance of rule based trigger identification

Total Identified	True Pos.	False Pos.	False Neg.	Prec.	Rec.	F-Mes.
1774	1152	622	192	0.65	0.85	0.74

Table 5. Performance of different target identification algorithms on test data

Method	TP	FP	Error	FN	Rec.	Pr.	F1
Rule Based	147	59	41	18	0.71	0.59	0.64
Voted Perceptron	168	73	38	0	0.82	0.60	0.69
Maximum Entropy without threshold	172	73	34	0	0.83	0.62	0.71
Maximum Entropy with threshold	153	29	21	32	0.74	0.75	**0.75**

that the voted perceptron provides the top ranking NP which is assumed to be the target of the corresponding trigger. This algorithm is not able to reject false positive triggers. Voted perceptron adjusts the parameter by jointly considering the positive and negative candidates (examples) for each trigger. Since false positive triggers have no positive candidate, there is no obvious way to use them during the training. Furthermore since magnitudes of $\mathbf{w}^i \cdot h(x_{i,j})$ (line 2 in Figure 3) are only suitable to rank various candidates of a single trigger, their values cannot be used against a common threshold to reject false positives.

The maximum entropy based algorithm considers each positive and negative candidate (example) independently and not as group corresponding to a trigger. Therefore, it can use false positive triggers to generate negative examples for training. Furthermore, the probability $p(\textit{target} = j|\textit{trigger} = x^i)$ output by the classifier can be used against a common threshold to reject highest ranked candidate NPs. To compare the performance of the maximum entropy based algorithm with the voted perceptron, the third row of Table 5 shows the performance of the maximum entropy classifier based ranking without thresholding, i.e. without the ability to reject false positive triggers. In this single result we find that in spite of the several desirable properties of voted perceptron listed in Section 4, it did not perform better than the maximum entropy based algorithm. Last row of Table 5 shows a 4% absolute improvement in the performance of the maximum entropy based ranker with rejection threshold. In this ranker the top ranked NP is rejected if the probability output by the classifier is below a certain threshold (established through a validation set).

6 Conclusion

Social media contains many types of useful information for businesses. In this paper we discussed the extraction from Twitter data the descriptions of problems consumers experience with products/services. There are many efforts towards extracting consumer's subjective opinions from social media, but to best of our knowledge there has been no attempt to extract objective descriptions of problems invaluable to product/service providers. We presented a novel trigger-target approach to extract different types of information. Triggers are the phrasal evidence that a tweet contains a description of a problem. Once triggers are identified their targets, which refer to the objects that are causing the problem, are identified. For trigger identification we presented a rule based approach with an F-measure of 0.74. For target identification we presented a ranking approach with an F-measure of 0.75 which is a significant improvement over a previously published rule based approach [8]. This results in a combined trigger-target identification F-measure of 0.555. Even though further improvements in the performance are desirable, this level of performance is quite encouraging considering the noisy language used in Twitter data and considering that we used NLP tools trained on standard text corpora to process them.

References

1. **Collins, M., Duffy, N., & Park, F.** (**2002**). New ranking algorithms for parsing and tagging: Kernels over discrete structures, and the voted perceptron. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics (ACL2002)*. 263–270.
2. **Cook, P. & Stevenson, S.** (**2009**). An unsupervised model for text message normalization. In *Proceedings of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (ACL-HLT2009)*

Workshop on Computational Approaches to Linguistic Creativity. Boulder, Colorado, USA, 71–78.

3. **Dave, K., Lawrence, S., & Pennock, D. M.** (**2003**). Mining the peanut gallery: Opinion extraction and semantic classification of product reviews. In *Proceedings of the 12th International Conference on the World Wide Web (WWW2003)*. 519–528.

4. **de Marneffe, M.-C., MacCartney, B., & Manning, C. D.** (**2006**). Generating Typed Dependency Parses from Phrase Structure Parses. In *Proceedings of the IEEE / ACL 2006 Workshop on Spoken Language Technology*. The Stanford Natural Language Processing Group.

5. **Gildea, D. & Jurafsky, D.** (**2002**). Automatic labeling of semantic roles. *Computational Linguistics*, 28(3), 245–288.

6. **Gimpel, K., Schneider, N., O'Connor, B., Das, D., Mills, D., Eisenstein, J., Heilman, M., Yogatama, D., Flanigan, J., & Smith, N. A.** (**2011**). Part-of-speech tagging for twitter: Annotation, features, and experiments. In *Proceedings of the 49th Annual Meeting on Association for Computational Linguistics (ACL2011)*. 42–47.

7. **Glance, N., Hurst, M., Nigam, K., Siegler, M., Stockton, R., & Tomokiyo, T.** (**2005**). Deriving marketing intelligence from online discussion. In *Preceedings of the eleventh ACM SIGKDD international conference on Knowledge discovery in data mining (KDD2005)*. 419–428.

8. **Gupta, N. K.** (**2011**). Extracting descriptions of problems with product and services from twitter data. In *Proceedings of the 3rd Workshop on Social Web Search and Mining (SWSM2011)*. Beijing, China.

9. **Han, B. & Baldwin, T.** (**2011**). Lexical normalisation of short text messages: Makn sens a #twitter. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies (ACL-HLT2011)*. Association for Computational Linguistics, Portland, Oregon, USA, 368–378.

10. **Hu, M. & Liu, B.** (**2004**). Mining opinion features in customer reviews. In *Proceedings of the 19th national conference on Artifical intelligence (AAAI2004)*.

11. **Jiang, L., Yu, M., Zhou, M., Liu, X., & Zhao, T.** (**2011**). Target-dependent twitter sentiment classification. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies (ACL-HLT2011)*. Association for Computational Linguistics, Portland, Oregon, USA, 151–160.

12. **Kaufmann, J. & Kalita, J.** (**2010**). Syntactic normalization of twitter messages. In *Proceedings of International Conference on Natural Language Processing, Kharagpur, India*.

13. **Kim, S.-M. & Hovy, E.** (**2004**). Determining the sentiment of opinions. In *Proceedings of the International Conference on Computational Linguistics (COLING2004)*.

14. **Liu, J. & Seneff, S.** (**2009**). Review sentiment scoring via a parse-and-paraphrase paradigm. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP2009)*. Singapore, 161–169.

15. **Liu, X., Zhang, S., Wei, F., & Zhou, M.** (**2011**). Recognizing named entities in tweets. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies (ACL-HLT2011)*. 359–367.

16. **Nigam, K. & Hurst, M.** (**2004**). Towards a robust metric of opinion. In *Proceedings of the AAAI Spring Symposium on Exploring Attitude and Affect in Text: Theories and Applications*. Standford, CA.

17. **Nigam, K., Lafferty, J., & Mccallum, A.** (**1999**). Using maximum entropy for text classification. In *Proceedings of International Joint Conference on Artificail Intelligence (IJCAI-99) Workshop on Machine Learning for Information Filtering*. 61–67.

18. **Pang, B., Lee, L., & Vaithyanathan, S.** (**2002**). Thumbs up? Sentiment classification using machine learning techniques. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP2002)*. Philadelphia, PA, USA, 79–86.

19. **Pradhan, S. S., Ward, W., & Martin, J. H.** (**2008**). Towards robust semantic role labeling. *Computational Linguistics*, 34(2), 289–310.

20. **Ritter, A., Clark, S., Mausam, & Etzioni, O.** (**2011**). Named entity recognition in tweets: An experimental study. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP2011)*. 1524–1534.

21. **Saeger, S. D., Torisawa, K., & Kazama, J.** (**2008**). Looking for trouble. In *Proceedings of the 22nd International Conference on Computation Linguistics (COLING2008)*. 185–192.

22. **Toutanova, K., Haghighi, A., & Manning, C. D.** (**2005**). Joint learning improves semantic role labeling. In *Proceedings of the Annual Meeting of the Association of Computational Lingusitics(ACL2005)*. 589–596.

Narendra Gupta received his B.Eng. degree form University of Roorkee (now IIT Roorkee), India and M.A.Sc. degree from University of Waterloo, Canada. From 1972 to 1983 he worked in various capacities with Siemens India Ltd. and with Siemens A.G. in Germany. From 1985 to 1995 he worked as Senior Member of Technical Staff with Siemens Corporate Research in Princeton, N.J. During this period Mr. Gupta worked and published on topics like Design Automation, Expert Systems, Equipment Diagnosis Systems and Information Retrieval. In 1995 Mr. Gupta Joined AT&T Bell Labs, and worked in the area of Decision Support Systems. Mr. Gupta currently works with AT&T research. His main interests are in Natural Language Understanding and Information extaction. Mr. Gupta is the principal contributor to the spoken language understanding module of the AT&T VoiceTone® Service and has also published work in opinion mining and information extraction from social media data.

Article received on 03/12/2012; accepted on 11/01/2013.