Extracting and Normalizing Entity-Actions from Users' Comments

Swapna Gottipati, *Jing Jiang*
School of Information Systems, Singapore Management University, Singapore
`swapnag.2010@smu.edu.sg`, `jingjiang@smu.edu.sg`

ABSTRACT
With the growing popularity of opinion-rich resources on the Web, new opportunities and challenges arise and aid people in actively using such information to understand the opinions of others. Opinion mining process currently focuses on extracting the sentiments of the users on products, social, political and economical issues. In many instances, users not only express their sentiments but also contribute their ideas, requests and suggestions through comments. Such comments are useful for domain experts and are referred to as actionable content. Extracting actionable knowledge from online social media has attracted a growing interest from both academia and the industry. We define a new problem in this line which is extracting entity-actionable knowledge from the users' comments. The problem aims at extracting and normalizing the entity-action pairs. We propose a principled approach to solve this problem and detect exactly matched entities with 75.1% F-score and exactly matched actions with 76.43% F-score. We could achieve an average precision of 81.15% for entity-action normalization.

KEYWORDS: Information Extraction, Normalization, Clustering, Conditional Random Fields.

Proceedings of COLING 2012: Posters, pages 421–430,
COLING 2012, Mumbai, December 2012.

1 Introduction

Opinion mining generally refers to extracting, classifying, understanding, and assessing the opinions expressed in various forums, online news sources, review sites, social media comments, and other user-generated content. Many different aspects of opinions, such as opinion targets (Ma and Wan, 2010), opinion polarity (Pang and Lee, 2008), and opinion holders (Kim and Hovy, 2005, 2006), have been studied.

In general, 90% of user's intention to write product reviews is to talk about the quality of the product and help others in decision making to buy the products[1]. Different from product reviews, user's intention to write comments on non-product issues like social, economical and political issues is to express sentiments or suggestions to the issue. In this work, we focus on comments that contain suggestions. Following the work by (Whittle et al., 2010; Ferrario et al., 2012), we define *actionable comments* as expressions that contain the requests or suggestions that can be acted upon. While motivating our task based on the previous work, we further extend the definition of *an actionable comment as an expression with an entity such as person or organization and a suggestion that can be acted upon*. More formally, an *actionable comment* is an expression with an entity and action expression. For example, in the comment "the government should tighten immigration rules," "the government" is the entity and "tighten immigration rules" is the action expression.

Detecting actionable comments is an important subtask for various problems. First, actionable knowledge detection opens a new perspective to opinion mining such that it taps into the aspect of suggestion generation process currently missed by traditional content analysis approaches. Second, this task aids in finding the public's actionable sentiment towards the entity by exploiting the individual value of an opinion and aids domain experts (Ferrario et al., 2012). Third, when users intend to get the gist of the comments, this task aids in generating such well-structured entity-based summaries on suggestions.

Finding a piece of actionable knowledge in social media typically involves extensive human inspection, which is labor-intensive and time-consuming. To illustrate the nature of the task, let us examine the following examples:

[C1]*The government should lift diplomatic immunity of the ambassador.*

[C2]*Govt must inform the romanian government of what happened immediately.*

[C3]*SG government needs to cooperate closely with romania in persecuting this case.*

[C4]*Hope the government help the victims by at least paying the legal fees.*

[C5]*I believe that goverment will help the victims for legal expenses.*

The above comments are in response to the news about a car accident. First, all sentences consist of an action and the corresponding entity who should take the action. Second, users tend to express the actions in various sentence structures and hence extracting entities and actions is desired and challenging as well. Third, we observe that entities in all the above sentences refer to the same entity, Government, but expressed in various forms. This drives the need for normalizing the entities. Finally, similar actions are expressed differently which drives the need for normalizing the actions. We treat all the above expressions as actionable comments and here we study how to extract and normalize entities and actions from users' comments. Table 1 gives an example output of our task.

[1]http://www.bazaarvoice.com/about/press-room/keller-fay-group-and-bazaarvoice-study-finds-altruism-drives-online-reviewers

Entity	Action
government	lift diplomatic immunity of the ambassador and get him to face..
government	inform the romanian government of what happened immediately..
government	cooperate closely with romania in persecuting..
government	help victims by at least paying the legal fees

Table 1: Sample output of actionable comments extraction and normalization task.

2 Nature of actionable comments

How are actionable comments expressed in English sentences? In this section, we study the language aspects of actionable comments at sentence level and at phrase level. This study is important for motivating and designing our solution.

2.1 Sentence level study

First, to understand how frequently a user writes an actionable comment, we randomly selected 500 sentences from AsiaOne.com[2], a news forum site. These sentences are from users' comments and each comment contains one or more sentences. We manually labeled these sentences as actionable comments or non-actionable comments. Our first observation is that 13.6% of the sentences are actionable comments. This is a very small set of candidates and hence justifies the need for detecting actionable comments. Second, to understand how to filter the comments that are non-actionable using some patterns, we further analyzed actionable comments at sentence level and our second observation is that, 88.3% of the actionable comments use the keywords listed in the Table 2. These findings are very similar to (Ferrario et al., 2012).

Keyword	Frequency	Keyword	Frequency	Keyword	Frequency
should	54.24%	hope	8.47%	believe	3.39%
may be	5.08%	have to	5.08%	ought	1.69%
to be	3.39%	suggest	3.39%	suppose to	1.69%
need to	3.39%	must	3.39%	advise	3.39%
needs to	1.69%	request	1.69%		

Table 2: Keywords and their relative frequencies in actionable comments.

Using the above keywords we now study the accuracy of identifying the actionable comments. We randomly extracted 550 sentences with the actionable keywords defined in Table 2 and traced for actionable comments. We identified that 83.41% of the comments are actionable and others are non-actionable comments. This observation justifies the need for filtering the user comments using the keywords and generating the candidate set of sentences. For our solution, we rely on data pre-processing by leveraging on these language dynamics.

2.2 Phrase level study

Intuitively, given an actionable comment, the entities can be treated as noun phrases and actions as verb phrases. We observe the following challenges in extracting actionable comments:
Entity extraction: Users tend to express the suggestions either in active voice or passive voice. The first challenge is to identify the correct entity in the actionable comment.

[2]www.asiaone.com

Normalization: People may refer to the same entity using different expressions and ideally we should normalize them. The second challenge is to normalize the entity mentions to their canonical form.
Redundancy: Very similar actions can be expressed differently. The third challenge is to normalize similar actions to aid in redundancy elimination.
Overall, the first challenge motivates us to detect entity-action pairs as an information extraction task and the last two challenges motivate normalizing the entity-action pairs as a normalization task.

3 Task Definition

The goal of our task is to extract and normalize actionable comments from user generated content in response to a news article. The actionable comments will be represented as an entity-action pair. Our problem of detecting normalized actionable comment is defined as follows: Given a news article A and corresponding candidate comments $C = \{c_1, c_2, \ldots, c_n\}$ extracted using the keywords, our goal is to detect pairs of $\{ne_i, na_i\}$ where ne_i is a normalized entity and na_i is a normalized action.

4 Solution Method

In this section, we first describe our solution for entity-action extraction based on CRF model (Lafferty et al., 2001) and then our normalization model based on the clustering techniques for entity and action normalization.

4.1 Entity-action extraction

The entity-action extraction problem can be treated as a sequence labeling task. Let $x = (x_1, x_2, \ldots, x_n)$ denote a comment sentence where each x_i is a single token. We need to assign a sequence of labels or tags $y = (y_1, y_2, \ldots, y_n)$ to x. We define our tag set as {BE, IE, BA, IA, O}, following the commonly used BIO notation (Ramshaw and Marcus, 1995), where E stands for entity and A stands for action.

Features: To develop features, we consider three main properties of actionable comments. First, the entities of the actionable pairs are mostly nouns or pronouns. Second, the entities display the positional properties with respect to the keywords. Third, the entities should be grammatically related to the actions. For example the verb in the action phrase is related to the subject which is an entity of the actionable comment.

a. Parts-of-speech features: To capture the first property, we classify each word x_i into one of the POS tags using the Stanford POS tagger[3]. We combine this feature with the POS features of neighboring words in [-2, +2] window.

b. Positional features: To capture the second property, we find the position of each word, x_i with respect to the keyword in the given sentence. The feature is represented as positive numbers for words preceding the keyword and negative numbers for words succeeding the keyword in the sentence. We do the same for neighboring words in [-2, +2] window.

c. Dependency tree features: To capture the third property, for each word x_i, we check if it is nominal subject in the sentence and represent it by *nsubj*. The dependency tree features can be extracted using Stanford dependencies tool[4].

[3] http://nlp.stanford.edu/software/tagger.shtml
[4] http://nlp.stanford.edu/software/stanford-dependencies.shtml

The output of this task is $S = \{e_i, a_i\}$, a set of entity-action pairs. The next task is to normalize S which is described below.

4.2 Entity-action normalization

Given $S = \{e_i, a_i\}$, a set of entity-action pairs, the goal is to generate $NS = \{ne_i, na_i\}$, a set of normalized entity-action pairs.

4.2.1 Entity normalization

We use agglomerative clustering which is a hierarchical clustering method which works bottom-up (Olson, 1995) together with expanding the entity with the features from Google and Semantic-Similarity Sieves adopted from Stanford coreference algorithm (Raghunathan et al., 2010).

Features: Two types of features are used to expand an entity mention: first from Google and second from the parse tree structure. The representative of a cluster, n_e is chosen to be the entity mention which has the largest average similarity distance from the other entity mentions in the cluster.

a. Alias features: This sieve addresses name aliases, which are detected as follows: Given an entity mention, it is first expanded with the title of the news article and this query is fed to the Google API. Google outputs the ranked matching outputs. One option is to use the entire snippet as the features. Another option is use partial snippet. Google returns snippets that has bolded aliases. We use them as alias features for a given entity mention. For example, alias features for "Ionescu + *title*" are *Dr.Ionescu, Silvia Ionescu, Romanian Diplomat Ionescu* etc. This sieve also aids in solving the spell problems.

b. Semantic-similarity features: We follow the following steps from the relaxation algorithm from Stanford coreference resolution tool for both named and unnamed entities: (a) remove the text following the mention head word; (b) select the lowest noun phrase (NP) in the parse tree that includes the mention head word; (c) use the longest proper noun (NNP*) sequence that ends with the head word; (d) select the head word.

4.2.2 Action normalization

The main objective of normalizing the actions is to remove the redundant actions. We choose clustering same as above to normalize the actions associated with same normalized entity. The feature set for this task is simply bag-of-words with stop word removal. The representative action is also chosen similar to the above.

5 Dataset

Since the task of actionable comment extraction is new, we gathered and annotated our own dataset for evaluation. Our dataset consists of 5 contentious news articles and the corresponding comments from Asiaone.com, an online forum.

5.1 Pre-processing

For the dataset preparation, we use the keywords listed in Table 2 to extract the candidate sentences from all the comments (each comment has 1 or more sentences) in 5 news articles for the task. We use random 110 candidate sentences from each article and in total 550 candidates for experiments. We calculated the inter-annotator agreement level using Cohen's kappa. Cohen's kappa on actionable comments is 0.7679 which displays a strong agreement between the annotators.

6 Experiments

6.1 Experiments on entity-action extraction

To evaluate the entity-action extraction, we prepare the ground truth using the dataset described in Section 5. We first answer (Q1), how well the model performs in identifying actionable comments. We then evaluate the entity and action extraction from the actionable comments to answer (Q2). We experimented various combinations of features (not reported here) for CRF model and combined feature set gives the best results. We perform 10-fold cross validation for all our experiments. We use this pattern matching technique as a baseline.

Annotation: To prepare the ground truth, we engaged two annotators to label 550 candidate sentences for suggestion, entity and action. For this annotation task the judge should do the following:

1. Look for the person(s) or organization(s) who should execute the suggestion and label the entity with BE (beginning of an entity) and IE (inside an entity).
2. Look out for the action that should be performed by the entity and label it as an action: BA (beginning of an action), IA (inside an action). The others are labeled as O (other).
3. If both entity and action are found, sentence is a valid suggestion. Label it as 1. Otherwise, label it as 0.

6.1.1 Actionable knowledge detection results

Our model achieved precision of 88.26%, recall of 93.12% and F-score of 90.63% in classifying actionable comments and that answers our (Q1). In our analysis, we observed that the model failed in detecting the actionable comments when the sentences have poor grammatical structure. For example, "*Dont need to call the helpline..*", has a poor grammatical structure.

6.1.2 Entity extraction results

In Table 3, the baseline outperformed the CRF model on the overlap F-score and this is due to the relax mode of the overlap. But, for the exact match CRF has high F-score of 75.09% which is relatively 6.67% higher than the baseline. This answers our (Q2) for entity extraction evaluation.

	Exact Match		**Overlap Match**	
Metrics	**Baseline**	**CRF**	**Baseline**	**CRF**
Recall	**0.8799**	0.8352	0.9032	**0.9306**
Precision	0.5866	**0.6849**	**0.9597**	0.8578
F-score	0.7039	**0.7509**	**0.9306**	0.8927

Table 3: Entity Extraction Results

6.1.3 Action extraction results

From Table 4, we see that the baseline, which is the pattern matching technique, has high recall for both exact match and head match. But, for both exact match and head match CRF has high

F-score of 76.43% and 82.7%, respectively, which is relatively 11.9% and 0.03% higher than the baseline. Head match has generally high performance for both due to the property that an action is expressed as a verb. This answers our (Q2) for action extraction evaluation.

	Exact Match		Head Match	
Metrics	**Baseline**	**CRF**	**Baseline**	**CRF**
Recall	**0.8947**	0.8944	**0.9200**	0.9169
Precision	0.5519	**0.6741**	0.7468	**0.7544**
F-score	0.6827	**0.7643**	0.8244	**0.8270**

Table 4: Action Extraction Results

6.2 Experiments on entity-action normalization

We first answer (Q3), between single link and complete link, which technique is more suitable for this problem? We then answer (Q4), how does the clustering-based solution perform in normalizing the entity-action pairs?

6.2.1 Single Link Vs Complete Link

Annotation: The human annotator is given a set of entities from each article and asked to first group the similar entities together and then assign a label to each group.

	Single Link			Complete Link		
Article	**Precision**	**Recall**	**F-Score**	**Precision**	**Recall**	**F-Score**
A1	0.5161	0.5039	0.5100	**0.8462**	**0.6929**	**0.7619**
A2	**1.0000**	0.3333	0.5000	0.7143	**0.5238**	**0.6044**
A3	**0.7368**	0.3218	0.4480	0.5664	**0.7356**	**0.6400**
A4	**0.6258**	0.4567	0.5280	0.5328	**0.6689**	**0.5931**
A5	**0.9661**	0.4560	0.6196	0.7282	**0.6000**	**0.6579**

Table 5: F-score results comparison between single link and complete link

As shown in Table 5, even though the precision for single link is high, complete link out performs single link on recall and F-score and answers our Q3. For example, "*the ceo*" and "*ceo, smrt ceo ms saw*" are grouped into single cluster using complete link. Where as, for single link cluster, "*smrt ceo ms saw*" is a false negative.

6.2.2 Entity-Action Normalization Results

Annotation: We asked a human judge to validate the normalized entity-action pairs. Only if both entity and action are normalized (entity should be in canonical form and action should be non-redundant), the pair is labeled as valid. If we obtain (e1, a1), (e2, a2), and a1 and a2 refer to the same action, we label one of them as invalid.

From Figure 1, we notice that on all articles the precision is high for complete link measure. This can be justified due to high F-score from complete link measure.

We observed that for single link, the entities like *he*, *they* are not normalized into the correct clusters resulting in the lower precision. Complete link measure outperforms single link measure for all articles in normalizing task with an average precision of 81.15% and that answers

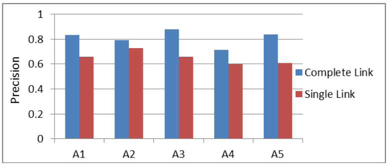

Figure 1: Entity-action normalization results

our Q4. We further analyzed the results for complete link. For article, A4 and the normalized entity *Ionescu*, the actionable comments have entity mentions like *asshole*, *dog* etc., which could not be normalized due to non-distinctive feature set.

7 Related Work

Opinion Mining: Opinion mining is a well studied research for the past ten years mainly focussing on the sentiment extraction and classification tasks (Turney, 2002; Pang et al., 2002). However, according to (Hu and Liu, 2004), fine-grained opinion mining and analysis is highly effective like feature identification by (Popescu and Etzioni, 2005), linking opinions to features by (Lin and He, 2009), and polarity classification by (Liu et al., 2005). Assessing the usefulness and quality of text has been well studied in natural language processing as quality plays a key role in online content (Agichtein et al., 2008) like helpfulness of reviews (Ghose and Ipeirotis, 2011), detecting low quality reviews (Liu et al., 2007) and detecting spam reviews (Lim et al., 2010).

Actionable content: (Zhang et al., 2009) attempted to discover the diagnostic knowledge and defined diagnostic data mining as, "a task to understand the data and/or to find causes of problems and actionable knowledge in order to solve the problems". Their work is more focussed towards manufacturing applications in which the problems are identified to aid the designers in the product design improvements. (Simm et al., 2010) analysed actionable knowledge in on-line social media conversation and the concept of actionability is defined as request or suggestion. (Ferrario et al., 2012) work aims at discovering aspects of actionable knowledge in the social media. To the best of our knowledge, our problem of extracting and normalizing entity-action pairs from users' comments is not studied.

Conclusion and perspectives

Actionable content extraction is a new direction in opinion mining process with many opportunities and challenges. With the increasing user generated content in micro blogs, detecting actionable knowledge in such media will be an interesting problem. For example, during Obama's state union address, apart from political and news forums, the public was asked to express opinions on Twitter using specific hashtags. This triggers the need for gathering actionable content in micro blogs. In the same line, diagnostic opinion detection that talks about what could have happened, who should be blamed, etc., is also an interesting problem.

Acknowledgments

This research/project is supported by the Singapore National Research Foundation under its International Research Centre@Singapore Funding Initiative and administered by the IDM Programme Office.

References

Agichtein, E., Castillo, C., Donato, D., Gionis, A., and Mishne, G. (2008). Finding high-quality content in social media. In *WSDM '08: Proceedings of the international conference on Web search and web data mining*, pages 183–194, New York, NY, USA. ACM.

Ferrario, M. A., Simm, W., Whittle, J., Rayson, P., Terzi, M., and Binner, J. (2012). Understanding actionable knowledge in social media: Bbc question time and twitter, a case study. In *ICWSM*.

Ghose, A. and Ipeirotis, P. G. (2011). Estimating the helpfulness and economic impact of product reviews: Mining text and reviewer characteristics. *IEEE Trans. Knowl. Data Eng.*, 23(10):1498–1512.

Hu, M. and Liu, B. (2004). Mining opinion features in customer reviews. In *Proceedings of the 19th national conference on Artifical intelligence*, AAAI'04, pages 755–760. AAAI Press.

Kim, S.-M. and Hovy, E. (2005). Identifying opinion holders for question answering in opinion texts. In *Proceedings of the AAAI Workshop on Question Answering in Restricted Domains*.

Kim, S.-M. and Hovy, E. (2006). Extracting opinions, opinion holders, and topics expressed in online news media text. In *Proceedings of ACL/COLING Workshop on Sentiment and Subjectivity in Text*, Sidney, AUS.

Lafferty, J. D., McCallum, A., and Pereira, F. C. N. (2001). Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the Eighteenth International Conference on Machine Learning*, ICML '01, pages 282–289, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

Lim, E.-P., Nguyen, V.-A., Jindal, N., Liu, B., and Lauw, H. W. (2010). Detecting product review spammers using rating behaviors. In *CIKM*, pages 939–948.

Lin, C. and He, Y. (2009). Joint sentiment/topic model for sentiment analysis. In *Proceedings of the 18th ACM conference on Information and knowledge management*, CIKM '09, pages 375–384, New York, NY, USA. ACM.

Liu, B., Hu, M., and Cheng, J. (2005). Opinion observer: analyzing and comparing opinions on the web. In *Proceedings of the 14th international conference on World Wide Web*, WWW '05, pages 342–351, New York, NY, USA. ACM.

Liu, J., Cao, Y., Lin, C.-Y., Huang, Y., and Zhou, M. (2007). Low-quality product review detection in opinion summarization. In *EMNLP-CoNLL*, pages 334–342.

Ma, T. and Wan, X. (2010). Opinion target extraction in chinese news comments. In *Coling 2010: Posters*, pages 782–790, Beijing, China. Coling 2010 Organizing Committee.

Olson, C. F. (1995). Parallel algorithms for hierarchical clustering. *Parallel Computing*, 21:1313–1325.

Pang, B. and Lee, L. (2008). Opinion mining and sentiment analysis. *Foundations and Trends in Information Retrieval*, 2(1-2):1–135.

Pang, B., Lee, L., and Vaithyanathan, S. (2002). Thumbs up?: sentiment classification using machine learning techniques. In *Proceedings of the ACL-02 conference on Empirical methods in natural language processing - Volume 10*, EMNLP '02, pages 79–86, Stroudsburg, PA, USA. Association for Computational Linguistics.

Popescu, A.-M. and Etzioni, O. (2005). Extracting product features and opinions from reviews. In *Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing*, HLT '05, pages 339–346, Stroudsburg, PA, USA. Association for Computational Linguistics.

Raghunathan, K., Lee, H., Rangarajan, S., Chambers, N., Surdeanu, M., Jurafsky, D., and Manning, C. (2010). A multi-pass sieve for coreference resolution. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, EMNLP '10, pages 492–501, Stroudsburg, PA, USA. Association for Computational Linguistics.

Ramshaw, L. A. and Marcus, M. P. (1995). Text chunking using transformation-based learning. *CoRR*, cmp-lg/9505040.

Simm, W., Ferrario, M. A., Piao, S. S., Whittle, J., and Rayson, P. (2010). Classification of short text comments by sentiment and actionability for voiceyourview. In *SocialCom/PASSAT'10*, pages 552–557.

Turney, P. D. (2002). Thumbs up or thumbs down?: semantic orientation applied to unsupervised classification of reviews. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, ACL '02, pages 417–424, Stroudsburg, PA, USA. Association for Computational Linguistics.

Whittle, J., Simm, W., Ferrario, M. A., Frankova, K., Garton, L., Woodcock, A., Nasa, B., Binner, J., and Ariyatum, A. (2010). Voiceyourview: collecting real-time feedback on the design of public spaces. In *UbiComp*, pages 41–50.

Zhang, L., Liu, B., Benkler, J., and Zhou, C. (2009). Finding actionable knowledge via automated comparison. In *ICDE*, pages 1419–1430.