Personalized Meta-Action Mining for NPS Improvement

Jieyan Kuang[1], Zbigniew W. Raś(✉)[1,2], and Albert Daniel[1]

[1] Univ. of North Carolina, College of Computing and Informatics, KDD Laboratory, Charlotte, NC, 28223, USA
[2] Warsaw Univ. of Technology, Inst. of Computer Science, 00-665 Warsaw, Poland
e-mail: jkuang1@uncc.edu, ras@uncc.edu

Abstract. The paper presents one of the main modules of HAMIS recommender system built for 34 business companies (clients) involved in heavy equipment repair in the US and Canada. This module is responsible for meta-actions discovery from a large collection of comments, written as text, collected from customers about their satisfaction with services provided by each client. Meta-actions, when executed, trigger action rules discovered from customers data which are in a table format. We specifically focus on the process of mining meta-actions, which consists of four representative and characteristic steps involving sentiment analysis and text summarization. Arranging these four steps in proposed order distinguishes our work from others and better serves our purpose. Compared to procedures presented in other works, each step in our procedure is adapted accordingly with respect to our own observations and knowledge of the domain. Results obtained from the experiments prove the high effectiveness of the proposed approach for mining meta-actions.

1 Introduction

Improving companies' NPS (Net Promoter System) has become one of the hottest topics nowadays since NPS is the most popular measurement for evaluating the performance of a company's growth engine. Generally speaking, NPS systems categorizes customers into three groups: *Promoter, Passive* and *Detractor*, which describe customers' satisfaction, loyalty and likelihood to recommend the company in a descending order [14].

The dataset we have contains over 42,000 records that are collected from 34 clients dealing with similar businesses crossing US and Canada. Each record represents answers to a questionnaire sent to a randomly selected customer. The questionnaire asks customers' personal information, general information about the service, but more importantly, customers' feeling on the service, such as *"was the job completed correctly"* and *"are you satisfied with the job"*. To answer such questions, customers assign scores ranging from 0 to 10 (higher the score is, more satisfied the customer is), and detailed comments in text format are also recorded if customers left any. Based on the numerical values given by customers, the average score of each customer can be computed and used to determine their NPS status: 9-10 is promoter, 7-8 is passive and 0-6 is detractor. Additionally, the NPS rating of each client can be calculated as the percentage difference between customers that are promoter and customers that are detractor.

In our dataset, NPS rating of individual client ranges from 0.503 to 0.86. Our ultimate goal is to provide proper suggestions for improving NPS rating of every client, in other words, improving customer satisfaction. To achieve this goal, we intend to build a hierarchically structured recommender system driven by action rules and meta-actions. We already built a hierarchical dendrogram by applying agglomerative clustering algorithm to the semantic distance matrix covering 34 clients [6]. Based on the dendrogram, we also proposed a strategy called Hierarchically Agglomerative Method for Improving NPS (HAMIS) to extend every client's dataset by merging it with other clients' datasets which are relatively close in the dendrogram, have better NPS, and classifiers extracted from them have higher precision and recall [7]. After a new maximally enlarged dataset is assigned to a client by HAMIS, action rules and meta-actions have to be extracted from it.

In this paper, we mainly focus on mining meta-actions. Many strategies have been designed for discovering action rules, but the area of mining meta-actions is still blossoming. The concept of action rules and meta-actions will be recalled in the next section. After that, the process of mining meta-actions will be explained thoroughly via four steps: *1*) Identifying opinion sentences and their orientation with localization; *2*) Summarizing each opinion sentence using discovered dependency templates; *3*) Opinion summarizations based on identified feature words; *4*) Generating meta-actions with regard to given suggestions. To test the proposed method for extracting meta-actions, experiments with a sample dataset are made. Evaluation results prove its high accuracy and effectiveness.

2 Action Rules, Meta-Actions and the Process of Generating Them

The concept of an action rule was firstly proposed by Ras and Wieczorkowska in [12] and investigated further in [3] and [16]. It is defined as a term $[(\omega) \wedge (\alpha \rightarrow \beta)] \Rightarrow (\phi \rightarrow \psi)$, where $(\omega \wedge \alpha) \Rightarrow \phi$ and $(\omega \wedge \beta) \Rightarrow \psi$ are classification rules, ω is a conjunction of stable attribute values, $(\alpha \rightarrow \beta)$ shows changes of flexible attribute values, and $(\phi \rightarrow \psi)$ shows desired effect of the action. Now we give an example assuming that a is stable attribute, b is flexible attribute and d is decision attribute. Terms (a, a_2), $(b, b_1 \rightarrow b_2)$, $(d, d_1 \rightarrow d_2)$ are examples of atomic actions. Expression $r = [(a, a_2) \wedge (b, b_1 \rightarrow b_2)] \Rightarrow (d, d_1 \rightarrow d_2)$ is an example of an action rule saying that if value a_2 of a in a given object remains unchanged and its value of b will change from b_1 to b_2, then its value of d is expected to transition from d_1 to d_2.

Meta-actions are the actions that need to be executed in order to trigger corresponding atomic actions. The concept of *meta-action* was initially proposed in [17]. But unlike the traditional understanding of meta-actions in [16], in our domain one atomic action can be invoked by more than one meta-action. So a set of meta-actions will trigger an action rule which consists of atomic actions covered by these meta-actions. Also some action rules can be invoked by more than one set of meta-actions. By selecting a proper set of meta-actions we could benefit in triggering larger number of action rules.

As meta-actions are the actual tools to trigger action rules and ulteriorly improve NPS ratings, the process of discovering them is what we need and have accomplished in this paper. Triggers aiming at different action rules should be extracted from respectively relevant comments left by customers in our domain. Let's assume that action rule r mentioned earlier is our target, and two classification rules r_1 and r_2 have been used to construct r [13], so the clues for generating meta-actions are in the comments stored in records satisfying the description $(a, a_2) \wedge (b, b_1) \wedge (d, d_1)$ or $(a, a_2) \wedge (b, b_2) \wedge (d, d_2)$.

To generate meta-actions from a determined set of comments, four steps mentioned earlier are designed to accomplish this task. The whole process involves not only the sentiment analysis and text summarization, but also generation of appropriate suggestions as meta-actions, which is more characteristic for our purpose. Before going into details of each step, adjusting the order of steps 1-3 is another uniqueness of our approach. This adjustment is made because of two reasons: *a*) Data is uncleaned. Unlike reviews on particular products or experience mentioned in other research, some comments in our domain are useless due to lack of opinion orientation; *b*) Former steps can benefit latter steps. After comparing the methods of extracting aspects (features) from different formatted comments, it turns out that dealing with reviews in short segments is more efficient without compromising their effectiveness than dealing with them in long segments. Therefore, ordering the first three steps in a way we proposed accelerates the process by eliminating useless information and makes preparing the data easier to handle for the next step. Most crucially, the accuracy will get improved.

2.1 Identification of Opinion Sentences and the Orientation with Localization

To identify an opinion sentence which expresses customers' sentiment, the presence of opinion words is considered as a standard sign. Initially adjectives are usually used as the main opinion words, like Hu and Liu have used only adjectives in [4]. Two sets of opinion words expressing positive and negative feelings are generated. Although these sets of opinion words are still growing continually, using them as the only references to detect opinion words and their orientation is not sufficient due to its generality. In some local scenarios, the lists of opinion words can be expanded more broadly by considering some neutral words with implicit polarity. For example, a comment *"the charge was too high"* can not be associated with given lists because the adjective *"high"* is not recognized as a positive nor negative word. However it definitely presents a useful message reflecting customers' negative opinion about the price, so the word *"high"* can be treated as a negative opinion word in this case. Similarly, other special neutral words or phrases can be added as opinion words if they reflect oriented meanings under certain circumstances without confusion. Such addition strongly relies on designers' knowledge about the domain.

Hence, based on our own experience, some neutral adjectives and verbs are added into our library of opinion words with clarified orientations. Four types of words that could have orientations are used to filter the appearance of opinion words and they are: *verb, adjective, adverb* and *noun*. As long as a word in a sentence tagged as any one of the four types exists in an extended list of opinion words, this sentence is an opinion sentence and the orientation of a tagged opinion word depends on its ascription to which list. The orientation of a sentence is determined by following the basic principles summarized in [9], when there is only one opinion word. Otherwise, the orientation of a sentence is a collection of the orientations of all subsentences associating with corresponding opinion words.

2.2 Summarization of Opinion Sentence based on Dependency Relationships

With opinion sentences identified, shortening them into segments is an important procedure. Relevant research like [18] and [15] constructs feature-opinion pairs with grammatical rules describing the relationships between features and opinion words. Without pre-identified features in opinion sentences, extracting summaries from every sentence by following certain grammatical relations associated with opinion words solely is also applicable and sufficient for two reasons. Firstly, unlike other relevant works, there is no need of Part-of-Speech (POS) [10] tagging, as the grammatical structure of a sentence is the only factor that we depend on. Secondly, the grammatical relations of the expected portion most closely connecting to the opinion words in a sentence can be summarized based on the knowledge of linguistics and used to extract a short but meaningful segment from a complete sentence.

The foundation of this step is based on the grammatical relations defined by Stanford Typed Dependencies Manual [2] and generated by Stanford Parser. A dependency relationship describes a grammatical relation between a governor word and a dependent word in a sentence and it is represented as *d(G, D)*, where d is one type of dependency among approximately 50 defined dependencies in [2], G and D are the governor and dependent respectively. With the comprehensive representation of dependencies, the nearest necessary relations associated with opinion words can be identified. Moreover, the types of dependencies that could link to opinion words straightforwardly rely on the tags of opinion words. Table 1 demonstrates all the discovered dependency templates for four types of opinion words respectively.

In Table 1, for each type of opinion words, all the other words linked with opinion words directly or indirectly are labeled as D_* regarding the dependency type. In one template, there could be more than one dependency, and the segment result is the combination of all involved words in a sequence as shown in the third column. When there are two or more templates involved with one opinion word, words from all detected templates will be combined sequentially as the words appear in the sentence. For example, if only dependency $nsubj$ is discovered in a sentence associated with a $noun$ opinion word, then the final segment is "D_{nsubj} *op*" as the first row in Table 1 shows. If additional dependencies $prep$ and $pobj$ are discovered and they are involved with the same $noun$ opinion word in a sentence, then the final segment result becomes "D_{nsubj} *op* D_{prep} D_{pobj}" by combining the results from both templates and ordering the words according to their locations in the sentence. During the process of exploring the dependencies in a sentence, it is necessary to detect the existence of a negation relation linked to a opinion word, if there is, then the opinion word *op* will be changed to *not op* before being used in the final segment.

2.3 Opinion Summarization based on Identified Feature Words

Identifying feature words from opinion summarizations is a simpler case now, because there is at most one feature in each segment with one opinion word, sometimes there is

Table 1. Dependency templates for extracting sentence segments

Type of opinion words	Dependency template	Segment result
Noun	nsubj(op , D_{nsubj})	D_{nsubj} + op
	prep(op, D_{prep}) + pobj(D_{prep}, D_{pobj})	op + D_{prep} + D_{pobj}
	dobj(D_{dobj}, op)	D_{dobj} + op
Adjective	nsubj(op , D_{nsubj})	D_{nsubj} + op
	amod(op, D_{amod}) + vmod(D_{amod}, D_{vmod}) + dobj(D_{vmod}, D_{dobj})	op + D_{amod}+ D_{vmod}+ D_{dobj}
	xcomp(op, D_{xcomp}) + dobj(D_{xcomp}, D_{dobj})	op + D_{xcomp} + D_{dobj}
	prep(op, D_{prep}) + pobj(D_{prep}, D_{pobj})	op + D_{prep} + D_{pobj}
	pcomp(op, D_{pcomp}) + dobj(D_{pcomp}, D_{dobj})	op + D_{pcomp} + D_{dobj}
	vmod(op, D_{vmod}) + dobj(D_{vmod}, D_{dobj})	op + D_{vmod} + D_{dobj}
Adverb	advmod(op, D_{advmod})	D_{advmod} + op
Verb	dobj(op, D)	op + D
	prep(op, D_{prep}) + pobj(D_{prep}, D_{pobj})	op + D_{prep} + D_{pobj}
	xcomp(op, D_{xcomp}) + dobj(D_{xcomp}, D_{dobj})	op + D_{xcomp} + D_{dobj}
	advcl(op, D_{advcl}) + dobj(D_{advcl}, D_{dobj})	op + D_{advcl} + D_{dobj}

[1] *op* denotes opinion word.

no valid feature existing in invalid segments. As the supervised pattern mining method - label sequential rule mining in [8] and [9] is proposed to handle reviews formated similarly as our segments, the similar idea is borrowed but broadened with our own observations to fulfill this step.

In the training dataset, the column is used to mark sequence of words in each segment. For example, the longest segment contains 5 words, then there are five columns in training dataset and each of them has assigned name ”$word\#$” to indicate the position of values in segments. In each row, every word in one segment is put in its corresponding column from the beginning, along with their POS tags. Last but not the least, every feature word in each segment will be identified and replaced with label [feature] manually, so a segment like ”pleased with attitude” will be represented as ”pleased_VB with_IN [feature]_NN” in our training dataset, where tags VB, IN and NN denote for verb, preposition or conjunction and noun respectively. Then association rules are mined from the training set with assistance of WEKA, and only the ones with label [feature] at the right hand side are kept and transformed into patterns. Following the example given above, the association rule generated for it is: word1=pleased_VB ==> word3=[feature]_NN and the pattern transformed from it becomes: <pleased_VB> <>< [feature]_NN>. Inspired by summarizing quite a lot of valid association rules and patterns retrieved from them, we cannot help thinking that the tags actually help to generalize the recognition of features, especially there are limited kinds of tags appearing in our segments. For instance, <excellent_JJ> <[feature]_NN> and <hard_JJ><[feature]_NN> (JJ denotes adjective) are two patterns which form a more general one <JJ><[feature]_NN> indicating that the noun appearing right after the adjective could be the feature under certain possibility. With regards to such observations, we are more inclined to build general patterns with only tags to predict features, which remarkably decreases the number of useful patterns and increases the efficiency.

Table 2. Feature classes and relevant feature words

Feature Class	Feature Words
Service	service, job, work, part, done, completed
Communication	communication, communicate, contact, reply, hear back
Staff	staff, dealer, mechanic, manager, guy, attitude, knowledgeable
Invoice/Price	invoice, price, charge, amount

In many sentiment analysis works, it is necessary to generate a final review summary for all discovered information about features and opinions, and also rank them by their appearances in the reviews. Besides that, more attention in our strategy is put on avoiding the redundancy of features words. To remove the redundant features words, feature classes are defined with a list of seed words or provided phrases based on our knowledge about the domain. Table 2 shows the examples of four representative feature classes and their feature words in our domain. Thus a segment will be clustered into a feature class if its feature word belongs to that class. As learning process continues, larger set of feature words for each class can be retrieved to enlarge its coverage.

2.4 Generation of Personalized Meta-Actions

The positive opinions indicate the satisfying behaviors that should remain, so the meta-actions for them are called *keeping* actions. Negative opinions show the undesirable behaviors that should be fixed, so their solutions are referred as *fixing* actions. Sometimes it is not hard to create *keeping* actions, since the positive segments can be used directly and they are explicit enough for users to understand and adopt. However, for negative segments, reversing them literally or removing the negation is not always right. To provide the most suitable fixing actions, consulting with company members who have expertise in this field is necessary and useful. In our case, a list of *fixing* actions to commonly discovered problems is collected and labeled with feature classes and subclasses. By subclasses, we mean the more specific aspects that could be designated by segments. For example, staff's attitude and staff's expertise are subclasses in class *Staff* . To map a segment to its meta-actions, we check if its opinion word is a synonym or antonym to the subclasses clarified in the list. If yes, then the meta-action for this segment is found; otherwise, the meta-action is not successfully matched.

3 Experiments

To implement our system for mining meta-actions, several existing tools from other projects are used. Stanford NLP part-of-speech tagger and lexicalized parser are used for generating POS tags [10] and identifying the dependency relations [2]. The lists containing positive and negative words respectively from Liu [8] are applied to detect opinion words and their polarities, ulteriorly the orientation of the segments. WordNet [11] is used to find the set of synonyms or antonyms. The system is built on JAVA and the sample used to test our approach contains 116 sentences which are manually labeled with relevant information including all expected results for each step, such as opinion sentence (or not) and opinion words orientation for the first step, and so on.

Table 3. Experiment results of major steps

	Precision	Recall	F-score
Opinion Sentence Identifier	0. 833	0.696	0.758
Opinion Sentence Summarizer	0.883	0.8	0.839
Feature Words Identifier	0..81	0.71	0.757
Feature Aggregator	0.78	0.733	0.753
Meta-Action Generator	0.78	0.75	0.764

After the sample data is processed with the proposed procedures, precision, recall and F-score are computed and shown in Table 3. Firstly, although there is no other comparable results for *Opinion Sentence Identifier*, its performance is very satisfying, its F-score is over 0.75 and the precision is over 0.8. Secondly, if comparing the performance of *Opinion Sentence Summarizer* and *Feature Words Identifier* in our work to the average results of feature-opinion pair mining and feature mining using approaches from [4] and [18] respectively, our approach achieves much better results in all three measurements. The accuracy of *Feature Aggregator* is very optimistic. For *Meta-Action Generator*, there are 30 *fixing* actions provided, and the performance of mapping them to specific segments is very acceptable, and its F-score is 0.764. Thus, the experiments confirm our expectation in the proposed method.

4 Conclusion

Generally speaking, the typical procedure of feature-based sentiment analysis in [4], [5] and [1] proceeds without opinion sentence summarization. Later, although [18] and [15] have completed some work on opinion summarization by mining feature-opinion pairs, our approach accomplishes opinion summarization by following the discovered templates of dependency relations involving expanded opinion words solely, and features in the summarized opinion segments are recognized by applying tag-dominated patterns transformed from association rules. Compared with other relevant work, our *Sentence Summarizer* and *Feature Words Identifier* achieve higher accuracy in the experiments, which proves the effectiveness of the atypically ordered and accordingly adjusted procedures. Besides adapting the traditional sentiment analysis into our project, designing the unique procedure - generation of meta-actions - resolves the demands for providing proper solutions to exposed problems, and the experiments also demonstrate its very positive effect. Moreover, we believe this process can be applied to other areas for solving the discovered problems with their personalizations.

References

1. S. Blair-Goldensohn, K. Hannan, R. McDonald, T. Neylon, G. A. Reis, J. Reynar (2008) Building a sentiment summarizer for local service reviews. In WWW Workshop on NLP in the Information Explosion Era, Vol. 14
2. M.C. De Marneffe, C.D. Manning (2008) Stanford typed dependencies manual. URL http://nlp. stanford. edu/software/dependencies manual. pdf.
3. Z. He, X. Xu, S. Deng, R. Ma (2005) Mining action rules from scratch, in Expert Systems with Applications, Vol. 29, No. 3, Elsevier, 691-699
4. M. Hu, B. Liu (2004) Mining and summarizing customer reviews. In Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining, pp. 168-177, ACM.
5. M. Hu, B. Liu (2004) Mining opinion features in customer reviews. In AAAI, Vol. 4, No. 4, pp. 755-760
6. J. Kuang, A. Daniel, J. Johnston, Z.W. Ras (2014) Hierarchically structured recommender system for improving NPS for a company. Proceedings of 9th International Conference, RSCTC 2014, Granada and Madrid, Spain, LNAI, vol. 8526, Springer, pp. 347-357
7. J. Kuang, Z.W. Ras, A. Daniel (2015) Hierarchical agglomerative method for improving NPS. Proceedings of the International Conference on Pattern Recognition and Machine Intelligence, LNCS, Vol. 9124, Springer, 2015
8. B. Liu, M Hu, J. Cheng (2005) Opinion observer: analyzing and comparing opinions on the web. In Proceedings of the 14th international conference on World Wide Web, pp. 342-351. ACM.
9. B. Liu (2010) Sentiment analysis and subjectivity. Handbook of natural language processing, 2, pp. 627-666
10. M. P. Marcus, B. Santorini, M. A. Marcinkiewicz (1993) Building a large annotated corpus of english: The Penn Treebank. Computational Linguistics.
11. G. A. Miller, R. Beckwith, C. Fellbaum, D. Gross, K. J. Miller (1990) Introduction to WordNet: An on-line lexical database*. International journal of lexicography, 3(4), pp. 235-244
12. Z.W. Ras, A. Wieczorkowska (2000) Action-rules: how to increase profit of a company, in Principles of Data Mining and Knowledge Discovery, (Eds. D.A. Zighed, J. Komorowski, J. Zytkow), Proceedings of PKDD'00, Lyon, France, LNAI, No. 1910, Springer, 587-592
13. Z. Ras, A. Dardzinska (2011) From data to classification rules and actions, in Proceedings of the Joint Rough Sets Symposium (JRS07), LNAI, Vol. 4482, Springer, pp. 322-329
14. F.F. Reichheld (2003) The one number you need to grow, in Harvard Business Review, December 2003, 1-8
15. G. Somprasertsri, P. Lalitrojwong (2010) Mining feature-opinion in online customer reviews for opinion summarization. J. UCS, 16(6), 938-955
16. A. Tzacheva, Z.W. Ras (2010) Association action rules and action paths triggered by meta-actions, in Proceedings of 2010 IEEE Conference on Granular Computing, Silicon Valley, CA, IEEE Computer Society, pp. 772-776
17. K. Wang, Y. Jiang, A. Tuzhilin (2006). Mining actionable patterns by role models. In Proceedings of the 22nd International Conference on Data Engineering, IEEE Computer Society, pp. 16-25
18. L. Zhuang, F. Jing, X. Y. Zhu (2006) Movie review mining and summarization. In Proceedings of the 15th ACM international conference on Information and knowledge management, pp. 43-50. ACM.