User Friendly NPS-based Recommender System for driving Business Revenue

Zbigniew W. Ras[1,2,3], Katarzyna A. Tarnowska[1], Jieyan Kuang[1], Lynn Daniel[4], and Doug Fowler[4]

[1] University of North Carolina, Department of Computer Science, Charlotte NC 28223, USA,
ras@uncc.edu, ktarnows@uncc.edu, jkuang1@uncc.edu,
[2] Warsaw University of Technology, Institute of Computer Science, 00-665 Warsaw, Poland
[3] Polish-Japanese Academy of IT, Warsaw, Poland
[4] The Daniel Group, Charlotte NC, USA
LynnDaniel@thedanielgroup.com, DougFowler@thedanielgroup.com

Abstract. This paper provides an overview of a user-friendly NPS-based Recommender System for driving business revenue. This hierarchically designed recommender system for improving NPS of clients is driven mainly by action rules and meta-actions. The paper presents main techniques used to build the data-driven system, including data mining and machine learning techniques, such as hierarchical clustering, action rules and meta actions, as well as visualization design. The system implements domain-specific sentiment analysis performed on comments collected within telephone surveys with end customers. Advanced natural language processing techniques are used including text parsing, dependency analysis, aspect-based sentiment analysis, text summarization and visualization.

Keywords: NPS, recommender system, actionable knowledge mining, semantic similarity, sentiment analysis, visualization

1 Introduction

The main idea behind this system is based on today's standard metric for measuring customer satisfaction called Net Promoter Score (NPS) [5]. It was designed to evaluate and improve the performance of a company's growth engine. The NPS metric is a concept based on the assumption that each customer can be labeled as either Promoter, Passive or Detractor. Promoters are loyal enthusiasts who are buying from a company and recommend others to do so. Passives are satisfied but unenthusiastic customers who are open to offers from competitors, while detractors are the least loyal customers who may urge others to avoid that company. The total Net Promoter Score is computed as %Promoters -%Detractors,

[5] NPS®, Net Promoter®and Net Promoter®Score are registered trademarks of Satmetrix Systems, Inc., Bain and Company and Fred Reichheld

where percentage is understood as the total number of promoters/detractors divided by the total number of surveys. The goal here is to maximize NPS, which in practice is a difficult task to achieve especially when a company has already quite high NPS. Nowadays most businesses, whether small, medium-sized or enterprise-level organizations with hundreds or thousands of locations collect their customers' feedback on products or services.
The data we worked on was collected by telephone surveys on customers satisfaction. There are about 400,000 records in the dataset collected in the years 2011-2016, and the data is continued to be collected. The dataset represents questionnaires sent to a randomly chosen group of customers and consists of features related to customers details (localization, type of work done, invoice, etc.), survey details (date, survey type, etc.), and benchmark questions on which service is being evaluated. Benchmarks include numerical scores (0-10) on different aspects of service. For example, if the job is done correctly, are you satisfied with the job, likelihood to refer, etc. All the responses from customers are saved into a database with each question (benchmark) as one feature in the dataset. The entire dataset consists of 38 companies, located in different sites across the United States as well as several parts of Canada. Based on overall benchmark scores, the Net Promoter Status (Promoter, Passive or Detractor) is determined for a client, which is a decision attribute in the dataset.

2 Semantic similarity

The dataset was divided into single-client subsets (38 in total). Additional attributes were developed, including spacial and temporal attributes. More detailed description of data pre-processing techniques is provided in Kuang et al. ([1]).

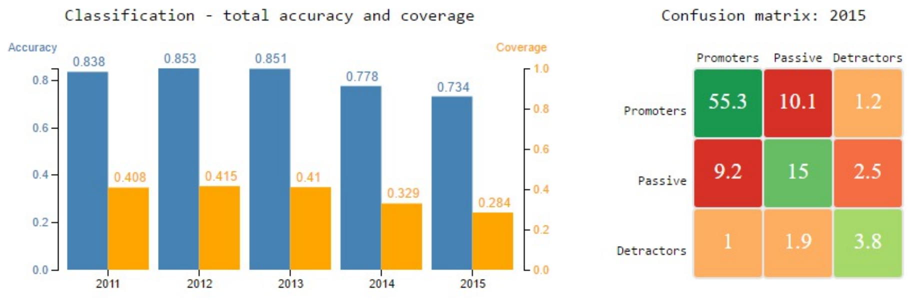


Fig. 1. Javascript-based visualization for depicting the results (accuracy, coverage and confusion matrix) of classification experiments on service data.

In the first place, classification experiments were conducted for each single dataset in order to determine the predictive capability of standard classifier

model and the same ability to discern and recognize different types of customers (promoters, passives and detractors). It was discovered that the classifier's accuracy/coverage was high for the category "Promoters", but low for the two other categories "Passives" and "Detractors".
We have used *RSES* (Rough Set Exploration System) to conduct initial experiments. The results of the classification experiments - accuracy, coverage and confusion matrix, for Service data for each client were implemented into a visualization system. The view for a sample client is shown in Figure 1. The confusion matrix updates for a chosen year after placing a mouse over the corresponding bar on the first chart.
Following the classification experiments, the notion of semantic similarity was defined ([1]). Assuming that RC[1] and RC[2] are the sets of classification rules extracted from the single-client datasets (of clients *C1* and *C2*), and also:
$RC[1] = RC[1, Promoter] \cup RC[1, Passive] \cup RC[1, Detractor]$, where the above three sets are collections of classification rules defining correspondingly: "Promoter", "Passive" and "Detractor":
$RC[1, Promoter] = \{r[1, Promoter, i] : i \in I_{Pr}\}$
$RC[1, Passive] = \{r[1, Passive, i] : i \in I_{Ps}\}$
$RC[1, Detractor] = \{r[1, Detractor, i] : i \in I_{Dr}\}$

In a similar way we define:
$RC[2] = RC[2, Promoter] \cup RC[2, Passive] \cup RC[2, Detractor]$.
$RC[2, Promoter] = \{r[2, Promoter, i] : i \in J_{Pr}\}$
$RC[2, Passive] = \{r[2, Passive, i] : i \in J_{Ps}\}$
$RC[2, Detractor] = \{r[2, Detractor, i] : i \in J_{Dr}\}$

By $C1[1, Promoter, i], C1[1, Passive, i], C1[1, Detractor, i]$ we mean confidences of corresponding rules in a dataset for client $C1$.
We define $C2[1, Promoter, i], C2[1, Passive, i], C2[1, Detractor, i]$ as confidences of rules extracted from $C1$ calculated for $C2$.
Analogously, $C2[2, Promoter, i], C2[2, Passive, i], C2[2, Detractor, i]$ are confidences of rules extracted from $C2$, and $C1[2, Promoter, i], C1[2, Passive, i]$, $C1[2, Detractor, i]$ are confidences of rules extracted from client $C2$ calculated for client $C1$.
Based on the above, the concept of semantic similarity between clients $C1$, $C2$, denoted by $SemSim(C1, C2)$ was defined as follows:

$$SemSim(C1, C2) =$$
$$\frac{\sum\{C1[1,Promoter,k]-C2[1,Promoter,k]|k\in I_{Pr}\}}{card(I_{Pr})}+\frac{\sum\{C1[1,Passive,k]-C2[1,Passive,k]|k\in I_{Ps}\}}{card(I_{Ps})}+$$
$$\frac{\sum\{C1[1,Detractor,k]-C2[1,Detractor,k]|k\in I_{Dr}\}}{card(I_{Dr})}+\frac{\sum\{C2[2,Promoter,k]-C1[2,Promoter,k]|k\in I_{Pr}\}}{card(J_{Pr})}+$$
$$\frac{\sum\{C2[2,Passive,k]-C1[2,Passive,k]|k\in I_{Ps}\}}{card(J_{Ps})}+\frac{\sum\{C2[2,Detractor,k]-C1[2,Detractor,k]|k\in J_{Dr}\}}{card(J_{Dr})}$$

The metric is used to find clients similar to a current client in semantic terms. It calculates the distance between each pair of clients–the smaller the distance is, the more similar the clients are. The resulting distance matrix serves as an input to the hierarchical clustering algorithm. The output of the algorithm is a structure, called dendrogram.

3 Hierarchical Agglomerative Method for Improving NPS

Hierarchical Agglomerative Method for Improving NPS (HAMIS) was proposed in Kuang at al ([2]) as a strategy for improving NPS of a company based on its local knowledge and knowledge collected from other semantically similar companies operating in the same type of industry. The strategy is based on the definition of semantic similarity introduced in the previous section. HAMIS is a dendrogram built by using agglomerative clustering strategy and semantic distance between clients.
The dendrogram was visualized in our web-based system by means of a node-link diagram that places leaf nodes of the tree at the same depth (see Figure 2). The clients (leaf nodes) are aligned on the right edge, with the clusters (internal

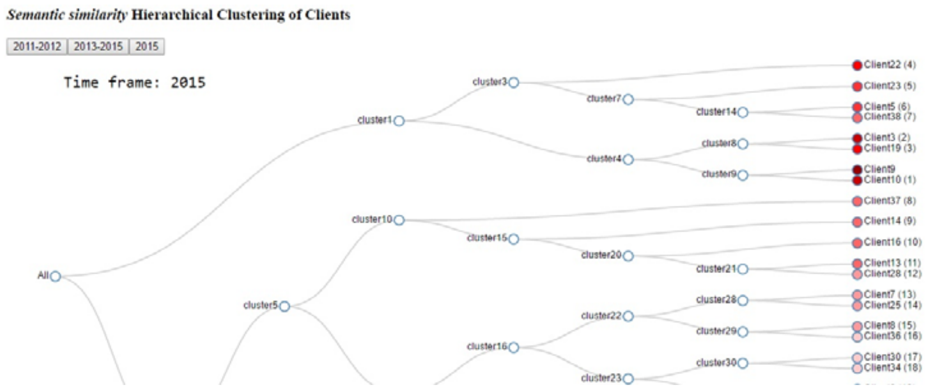


Fig. 2. Javascript-based visualization of the dendrogram showing semantic similarity between clients in 2015: chosen Client9 with highlighted semantically similar clients ordered by numbers.

nodes) - to the left. The data shows the hierarchy of client clusters, with the root node being “All” clients. The visualization facilitates comparing the clients by means of similarity. The nodes that are semantically closest to the chosen client are the leaf nodes on the sibling side. The diagram is interactive: after clicking on the client node, all the semantically similar clients are highlighted with numbers in parentheses denoting sequence of the most similar clients (with 1 - denoting the first most similar client, 2 - the second similar, etc.), and the color strength corresponding to the similarity.

The dendrogram was used to construct new “merged” datasets for further data mining (in particular, action rule mining, described in the next section). The merged datasets replace a current client’s dataset expanded by adding datasets of better performing clients who are semantically similar to it. So, besides semantic similarity, NPS efficiency rating is another primary measure considered when “merging” two semantically similar clients ([2]). As a result of this strategy, the NPS rating of the newly merged dataset will be higher than, or at least equal to, the dataset before its extension. This way, we can offer recommendations to the company with a lower NPS based on the data collected by companies with a higher NPS assuming that these two are semantically similar (that is, their customers understand the concepts of promoter, passive and detractor in a similar way). The second factor considered in the merging operation, besides the NPS, is the quality and consistency of the newly merged data. It is checked by means of F-score calculated for a classifier extracted from the newly merged dataset. The F-score was chosen for keeping track of datasets quality as it combines two other important metrics: accuracy and coverage of the classifier. In summary, three conditions have to be met for the two datasets to be merged:

- merged clients have to be semantically similar within defined threshold;
- NPS of the newly merged dataset must be equal or higher than the original dataset’s score;
- F-score of the newly merged dataset must be equal or higher then the currently considered dataset’s score.

If these three conditions are met, the datasets are being merged, and correspondingly the current NPS and F-score are updated as well. Then, the merging operation check with the next candidate datasets is continued, until the merging conditions fail or the root of hierarchical dendrogram is reached. By using dendrogram terminology, the current node is being replaced by the newly updated resulting node by “climbing up” the dendrogram. The HAMIS keeps expanding a current client by unionizing it with all the clients satisfying the conditions. The candidates are checked in a top down order based on their depth in the dendrogram: the smaller the depth of a candidate is, the earlier the candidate will be checked. The detailed algorithm for HAMIS and experiments on example runs with it are described in [2] and [1]. An example of expanding datasets of 36 clients based on Service 2016 data is shown in Figure 4. Half of the clients were extended by applying the HAMIS procedure, and a client was extended on average by about 3 other datasets. It can be observed that generally clients with lower NPS were extended by a larger number of datasets. It shows that their NPS can be improved more by using additional knowledge from semantically similar, better performing clients. For example, in Figure 4, Client20 with the worst NPS (of 63%) was extended by 10 other datasets and Client33 with the second worst NPS (69%) was extended by 11 other datasets. Expanding the original, single-client datasets was followed by action rule mining–the action rules mined from the extended datasets are expected to be better in quality and quantity. Recommender system based on action rules extracted from the extended datasets can give more promising suggestions for improving clients’ NPS score. The more

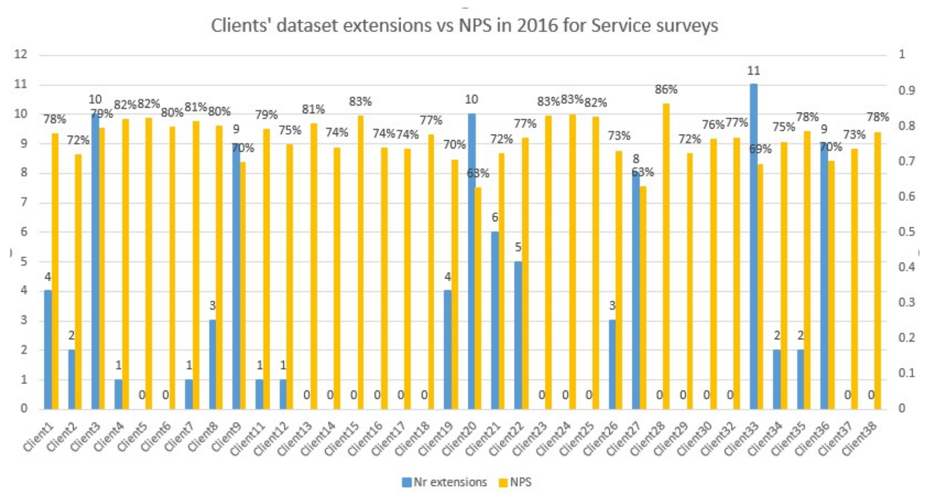


Fig. 3. Results of running HAMIS procedure on 38 datasets representing clients for Service survey data from 2016: the number of clients by which a client was extended and its original NPS.

extended the datasets, the better recommendations for improving NPS can be given by the system.

4 Action rules

The whole system is built from the knowledge extracted from the preprocessed dataset in the form of action rules. The knowledge is in actionable format and collected not only from the customers using certain business, but also from customers using semantically similar businesses having a higher NPS score.
Action rule concept was firstly proposed by Ras and Wieczorkowska in [11], and since then investigated further in application areas such as business, healthcare, music automatic indexing and retrieval. Action rules present a new way in machine learning domain that solve problems that traditional methods, such as classification or association rules cannot handle. The purpose is to analyze data to improve understanding of it and seek specific actions (recommendations) to enhance the decision-making process. An *action* shows a way of controlling or changing some of the attribute values for a given set of objects to achieve desired results [5]. An *action rule* is defined ([11]) as a rule that describes a transition that may occur within objects from one state to another, with respect to decision attribute, as defined by the user. Decision attribute is a distinguished attribute ([11]), while the rest of the attributes are partitioned into stable and flexible attributes.
In nomenclature, action rule is defined as a term: $[(\omega) \wedge (\alpha \rightarrow \beta) \Rightarrow (\Phi \rightarrow \Psi)]$, where ω denotes conjunction of fixed condition attributes often called the header

of the rule, $(\alpha \rightarrow \beta)$ are proposed changes in values of flexible features, and $(\Phi \rightarrow \Psi)$ is an expected change of a decision attribute value (action effect).
So, in our domain, decision attribute is *PromoterStatus* (with values *Promoter*, *Passive*, *Detractor*). Let us assume that Φ means 'Detractors' and Ψ means 'Promoters'. The discovered knowledge would indicate how the values of flexible attributes need to be changed under the condition specified by stable attributes so the customers classified as detractors should become promoters. So, an action rule discovery applied to customer data would suggest a change in flexible attribute values, such as different benchmarks to help "reclassify" or "transit" an object (customer) to a different category (Passive or Promoter) and consequently, attain better overall customer satisfaction.

An action rule is built from *atomic action sets.*

Definition 1. *Atomic action term is an expression* $(a, a_1 \rightarrow a_2)$*, where* a *is an attribute, and* $a_1, a_2 \in V_a$*, where* V_a *is a domain of attribute* a.

If $a_1 = a_2$ then attribute a is called stable on a_1.

Definition 2. *By action sets, we mean the smallest collection of sets such that:*

1. *If* t *is an atomic action term, then* t *is an action set.*
2. *If* t_1, t_2 *are action sets, then* $t_1 \wedge t_2$ *is a candidate action set.*
3. *If* t *is a candidate action set and for any two atomic actions* $(a, a_1 \rightarrow a_2)$, $(b, b_1 \rightarrow b_2)$ *contained in* t *we have* $a \neq b$*, then* t *is an action set. Here* b *is another attribute* $(b \in A)$*, and* $b_1, b_2 \in V_b$.

Definition 3. *By an action rule, we mean any expression* $r = [t_1 \Rightarrow t_2]$*, where* t_1 *and* t_2 *are action sets.*

The interpretation of the action rule r is, that by applying the action set t_1, we would get, as a result, the changes of states in action set t_2. So, action rule suggests the smallest set of necessary actions needed for switching from current state to another within the states of the decision attribute. We need to extract these kind of actions, so that we can build an effective recommender system that provides actionable suggestions for improving a client's performance.
The first step to extract action rules from the dataset by our recommender system is to complete the initialization of the mining program by setting up all the variables. This process consists of selecting stable attributes, flexible attributes and the decision attribute. We also need to set up the favorable state and the unfavorable state for the decision attribute, as well as minimum support of the rule and its minimum confidence. *PromoterScore* is set as the decision attribute, with *Promoter* value to be the target state (most favorable one) and *Detractor* the most undesirable state. For the stable attributes, all features related to the general information about clients and customers are considered; the final choice of stable attributes includes:

– *ClientName* - since rules should be client-oriented,
– *Division* - specific department,

– *SurveyType* - type of service: field trips, in-shop, parts, etc.
– *ChannelType*

Initially, as the flexible attributes, all features denoting numerical benchmark questions were chosen, as it is believed that representing them areas of service/parts can be changed by undertaking certain actions. This set of benchmarks has been reduced to smaller set of benchmarks, which we can call critical. We used them for mining action rules. The choice of critical benchmarks was preceded by an analysis of decision reducts, which are visualized in a user-friendly interface built for our recommender system.
According to the definition, *reducts* are minimal subsets of attributes that keep

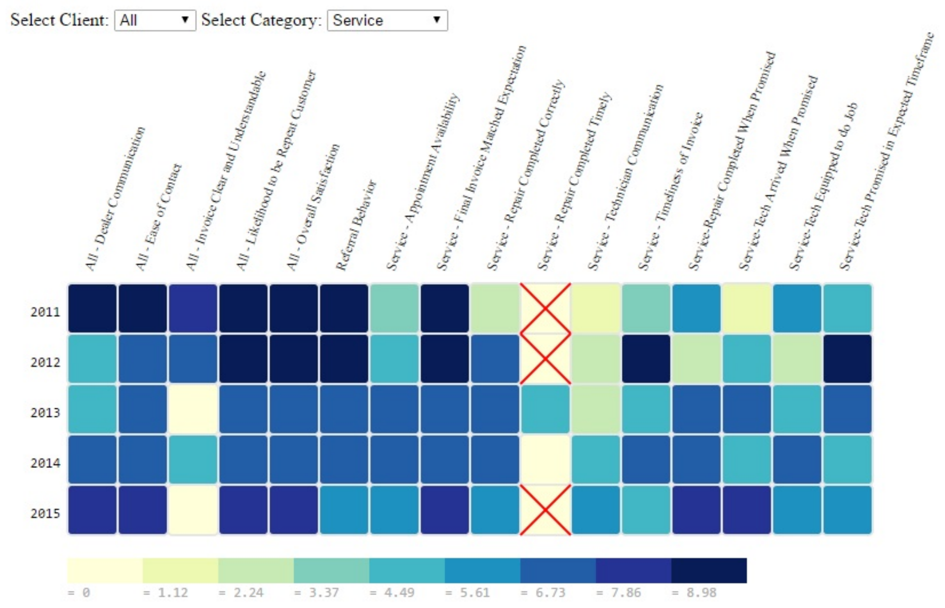

Select Client: All Select Category: Service

Fig. 4. Javascript-based visualization supporting an analysis of features related to survey benchmarks. The color of the cell corresponds to an occurrence of the associated benchmark in reducts of a corresponding dataset (for a client in a year).

the characteristics of the full dataset. In the context of action rule discovery, an *action reduct* is a minimal set of attribute values distinguishing a favorable object from another.

For our domain, decision reducts were extracted using Rough Set Exploration System (RSES). We also kept track of how the importance (or criticality) of particular benchmarks changed year by year and by client. The resulting visualization is depicted as a heatmap with colors denoting the importance of a benchmark (relative frequency of occurrence in decision reducts), rows representing years (2011-2015) and columns representing benchmarks occurring in decision reducts.

5 Meta actions and triggering mechanism

Our recommender system is driven by action rules and meta-actions to provide proper suggestions to improve the revenue of companies. Action rules, described in the previous section, show minimum changes needed for a client to be made in order to improve its ratings so it can move to the Promoter's group. Action rules are extracted from the client's dataset expanded by HAMIS procedure, explained in the previous sections.

Meta-actions are the triggers used for activating action rules [4] and making them effective. The concept of *meta-action* was initially proposed in Wang et al.([9]) and later defined in Ras et al. ([6]). Meta-actions are understood as higher-level actions. While an action rule is understood as a set of atomic actions that need to be made for achieving the expected result, meta-actions are the actions that need to be executed in order to trigger corresponding atomic actions.
For example, the temperature of a patient cannot be lowered if he does not take a drug used for this purpose - taking the drug would be an example of a higher-level action which should trigger such a change. The relations between meta-actions and changes of the attribute values they trigger can be modeled using either an influence matrix or ontology.
An example of an influence matrix is shown in Table 1 ([1]). It describes the relations between the meta-actions and atomic actions associated with them. Attribute a denotes stable attribute, b - flexible attribute, and d - decision attribute. $\{M_1, M_2, M_3, M_4, M_5, M_6\}$ is a set of meta-actions which hypothetically triggers action rules. Each row denotes atomic actions that can be invoked by the set of meta-actions listed in the first column. For example, in the first row, atomic actions $(b_1 \rightarrow b_2)$ and $(d_1 \rightarrow d_2)$ can be activated by executing meta-actions M_1, M_2 and M_3 together.

Table 1. Sample meta-actions influence matrix

	a	b	d
$\{M_1, M_2, M_3\}$		$b_1 \rightarrow b_2$	$d_1 \rightarrow d_2$
$\{M_1, M_3, M_4\}$	a_2	$b_2 \rightarrow b_3$	
$\{M_5\}$	a_1	$b_2 \rightarrow b_1$	$d_2 \rightarrow d_1$
$\{M_2, M_4\}$		$b_2 \rightarrow b_3$	$d_1 \rightarrow d_2$
$\{M_1, M_5, M_6\}$		$b_1 \rightarrow b_3$	$d_1 \rightarrow d_2$

In our domain, we assume that one atomic action can be invoked by more than one meta-action. A set of meta-actions (can be only one) triggers an action rule that consists of atomic actions covered by these meta-actions. Also, some action rules can be invoked by more than one set of meta-actions.
If the action rule $r = [\{(a, a_2), (b, b_1 \to b_2)\} \Rightarrow \{(d, d_1 \to d_2)\}]$ is to be triggered, we consider the rule r to be the composition of two association rules r_1 and r_2, where $r_1 = [\{(a, a_2), (b, b_1)\} \Rightarrow \{(d, d_1)\}]$ and $r_2 = [\{(a, a_2), (b, b_2)\} \Rightarrow \{(d, d_2)\}]$. The rule r can be triggered by the combination of meta-actions listed in the first and second row in Table 1, as meta-actions $\{M_1, M_2, M_3, M_4\}$ cover all required atomic actions: (a, a_2), $(b, b_1 \to b_2)$, and $(d, d_1 \to d_2)$ in r. Also, one set of meta-actions can potentially trigger multiple action rules. For example, the mentioned meta-action set $\{M_1, M_2, M_3, M_4\}$ triggers not only rule r, but also another rule, such as $[\{(a, a_2), (b, b_2 \to b_3)\} \Rightarrow \{(d, d_1 \to d_2)\}]$, according to the second and fourth row in Table 1, if such rule was extracted.
The goal is to select such a set of meta-actions which would trigger a larger number of actions and the same bring greater effect in terms of NPS improvement. The effect is quantified as following ([1]): supposing a set of meta-actions $M = \{M_1, M_2, ..., M_n : n > 0\}$ triggers a set of action rules $\{r_1, r_2, ..., r_m : m > 0\}$ that covers objects in a dataset with no overlap. We defined the coverage (support) of M as the summation of the support of all covered action rules. That is, the total number of objects that are affected by M in a dataset. The confidence of M is calculated by averaging the confidence of all covered action rules:

$$sup(M) = \sum_{i=1}^{m} sup(r_i)$$
$$conf(M) = \frac{\sum_{i=1}^{m} sup(r_i) \cdot conf(r_i)}{\sum_{i=1}^{m} sup(r_i)}$$

The effect of applying M is defined as the product of its support and confidence: $(sup(M) \cdot conf(M))$, which is a base for calculating the increment of NPS rating.

6 Text mining

Triggers aiming at different action rules are extracted from respectively relevant comments left by customers in our domain ([4]). Text comments are a complementary part of structured surveys. For example, for a rule described by: $r = [(a, a_2) \wedge (b, b_1 \to b_2)] \Rightarrow (d, d_1 \to d_2)]$, where a is a stable attribute, and b is a flexible attribute, the clues for generating meta-actions are in the comments of records matching the description: $[(a; a2) \wedge (b; b1) \wedge (d; d1)] \vee [(a; a2) \wedge (b; b2) \wedge (d; d2)]$.

Mining meta-actions consists of four characteristic steps involving sentiment analysis and text summarization ([3]):

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.

The whole process of mining customers comments uses sentiment analysis, text summarization and feature identification based on guided folksonomy (domain specific dictionaries are built). It also generates appropriate suggestions, such as meta-actions, which is important for the purpose of recommender system.
The schema of the presented aspect-based sentiment mining was inspired by a process described in [10]. *Sentiment analysis* is generally defined as analyzing people's opinions, sentiments, evaluations, attitudes, and emotions from written language. *Aspect-based sentiment analysis* is based on the idea that an opinion consists of a sentiment (positive or negative) and a *target* of the opinion, that is, a specific aspect or feature of the object. It offers more detailed and fine-grained analysis than document-level or sentence-level sentiment analysis.
Consequently, the first step in text mining consists of identifying an opinion sentence, based on the occurrence of an opinion word. A dictionary (list) of positive and negative words (adjectives) were used for that purpose. Context (localization) was also taken into account. For example, a comment *"the charge was too high"*, "high" is recognized according to the adjective lists as neither positive nor negative. However, the comment still presents an insightful opinion about discontent when it comes to pricing. Therefore, "high" was added to the list as a negative in the context of pricing.
In the next step, sentences with opinion words identified are shortened into segments. Feature-opinion pairs are generated based on grammatical dependency relationships between features and opinion words. The foundation of this step is the grammatical relations defined by Stanford Typed Dependencies Manual ([8]) and generated by Stanford Parser. A dependency relationship describes a grammatical relation between a governor word and a dependent word in a sentence. Given the wide definition of dependency templates (about 50 defined dependencies in [8]), all the necessary relations associated with opinion words can be identified. On top of it, negation and 'but'-clauses are identified.
Having extracted segments, feature words are identified using the supervised pattern mining method (similarly as described in [7]). The Parts-of-Speech tags (POS) help in the process of recognizing the features.
Opinion summarizations are used in many sentiment analysis works (first in [10]) to generate a final review summary about the discovery results on feature and opinions mining and also rank them according to their appearances in the reviews. In our work, we also focused on removing the redundancy of extracted segments and clustering segments into different classes. The feature clustering was based on the pre-defined list of seed words or phrases. To cluster a segment into the corresponding class, its feature word or the base form of its feature is checked whether it exists in any list of the seed words.
For the purpose of generating meta-actions, each feature class has been divided into several subclasses. Each subclass is related to the specific aspect of that feature. The aspects have been defined based on the domain knowledge.
The last step is generating meta-actions and providing them to the end business user along with the comments from which they were mined. The recommendations are divided into positive and negative recommendations. Negative opinions

show the undesirable behaviors that should be fixed, while the positive segments indicate which areas should be continued.

7 Visualization

For review summarization purposes, often a variety of visualization methods are deployed in the literature. We have developed an interactive user-friendly web-based interface for the recommender system. The interaction was divided into three basic steps:

1. Selecting the entity (client) the business user would like to analyze (see Figure 5);
2. Rating feasibility of improvements (drop-down lists in Figure 6);
3. Exploring the recommended improvement options (bubble chart in Figure 6) and comments from raw data related to the chosen option (data table in Figure 7).

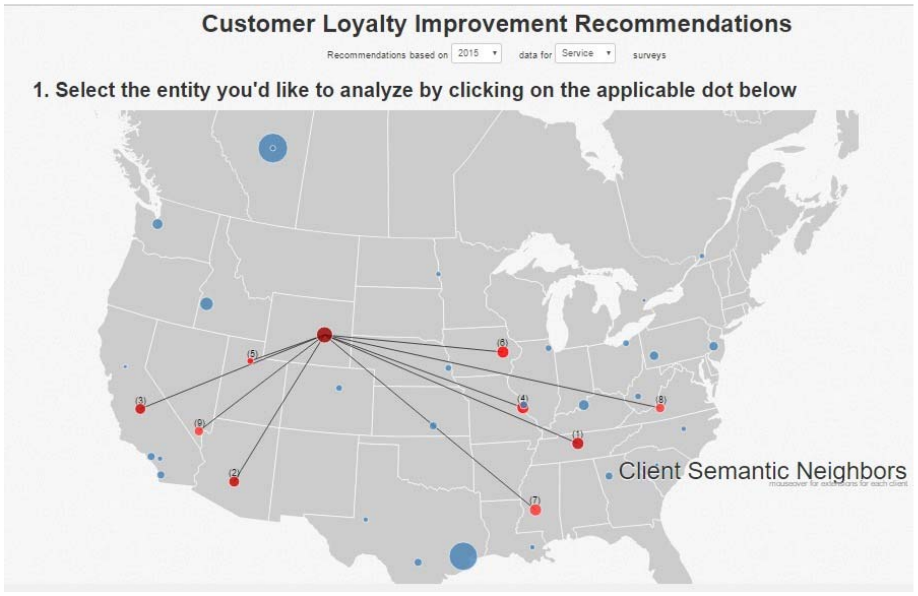
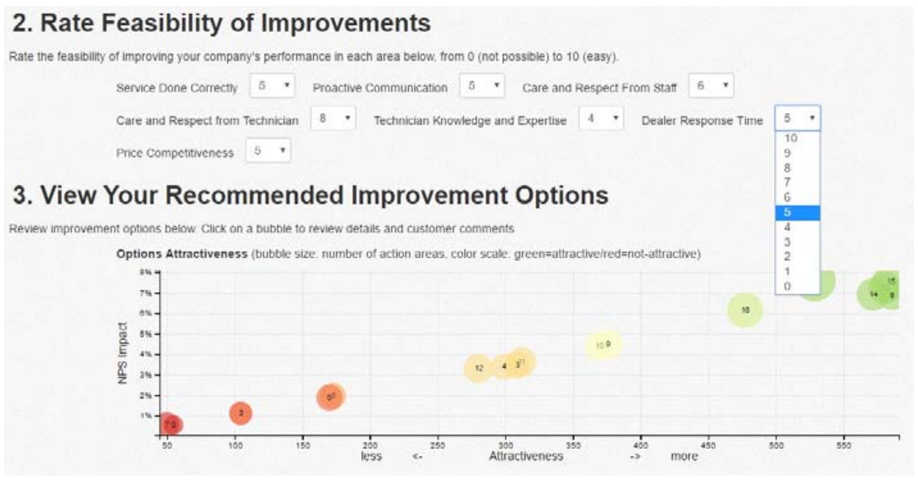

Customer Loyalty Improvement Recommendations
Recommendations based on 2015 data for Service surveys
1. Select the entity you'd like to analyze by clicking on the applicable dot below
(1)
(2)
(3)
(4)
(5)
(6)
(7)
(8)
(9)
Client Semantic Neighbors
mouseover for explanations for each client

Fig. 5. Javascript-based visualization for depicting clients' locations and their semantic neighbors. Also, serves as an interface for further analysis of a chosen entity.

The map in Figure 5 serves as an interface for further analysis of the chosen client (amongst 38 in total). The current version of the interface allows for choosing recommendations based on the datasets from the years 2016 or 2015



2. Rate Feasibility of Improvements

Rate the feasibility of improving your company's performance in each area below, from 0 (not possible) to 10 (easy).

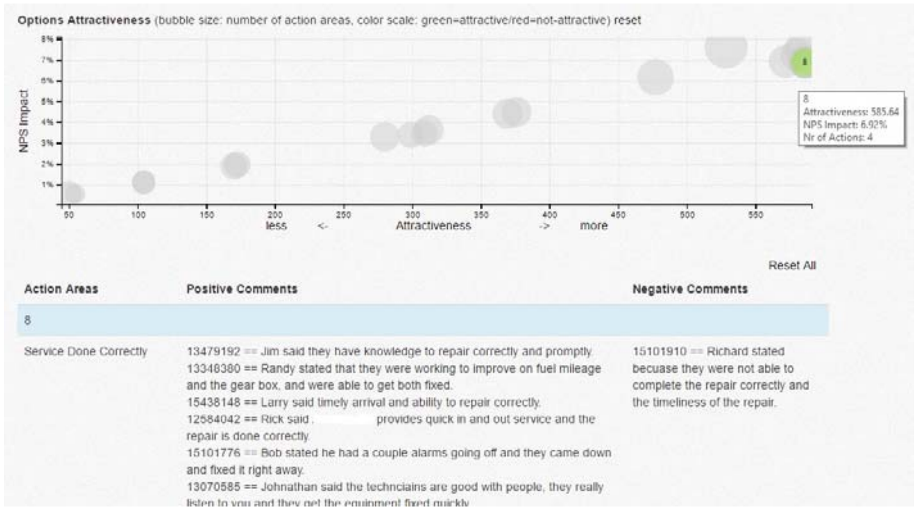
Service Done Correctly: 5
Proactive Communication: 5
Care and Respect From Staff: 6
Care and Respect from Technician: 8
Technician Knowledge and Expertise: 4
Dealer Response Time: 5
Price Competitiveness: 5

3. View Your Recommended Improvement Options

Review improvement options below. Click on a bubble to review details and customer comments

Options Attractiveness (bubble size: number of action areas, color scale: green=attractive/red=not-attractive)

Fig. 6. Javascript-based interactive visualization for exploring recommendations options and their attractiveness based on chosen feasibility.



Options Attractiveness (bubble size: number of action areas, color scale: green=attractive/red=not-attractive) reset

8
Attractiveness: 585.64
NPS Impact: 6.92%
Nr of Actions: 4

Action Areas	Positive Comments	Negative Comments
8		
Service Done Correctly	13479192 == Jim said they have knowledge to repair correctly and promptly. 13348380 == Randy stated that they were working to improve on fuel mileage and the gear box, and were able to get both fixed. 15438148 == Larry said timely arrival and ability to repair correctly. 12584042 == Rick said [redacted] provides quick in and out service and the repair is done correctly. 15101776 == Bob stated he had a couple alarms going off and they came down and fixed it right away. 13070585 == Johnathan said the techncians are good with people, they really listen to you and they get the equipment fixed quickly	15101910 == Richard stated becuase they were not able to complete the repair correctly and the timeliness of the repair.

Fig. 7. Javascript-based dynamic data table for exploring raw data comments associated with the analyzed recommendation option.

and surveys on Service or Parts. The clients are represented as points (dots) placed in their headquarters' locations. The size of the dot informs about how many other clients were added to the original client's dataset to mine for actionable knowledge (see section on semantic similarity and HAMIS procedure). The connecting lines show the semantic neighbors. After clicking the client's dot, it changes color from blue to red and the corresponding semantic neighbors are highlighted in a red scale as well. The color scale corresponds to the strength of semantic similarity. Additionally, the number in parentheses denotes the sequence of semantic similarity to the current client. We have hidden client labels (text next to the dots) on the grounds of data confidentiality.
The next step of interaction with the business user is exploring the recommendation options. The displayed options correspond to the extracted meta-actions (see the previous section) mined from text comments and summarized into aspect categories. The user (business consultant) can assign a feasibility score to each option based on dialogue with the currently analyzed client. For some clients some options might be more feasible that the others. For example, it might be quite difficult to change pricing, while it might relatively easier to change technician knowledge (for example, by introducing appropriate training).
The option's attractiveness depends on both factors: NPS improvement (calculated as described in the previous section based on the action rule and meta action mining) and feasibility chosen by the user. Each bubble (identified by an ordering number) corresponds to a different set of improvement options and they are ordered on the X-axis and the Y-axis according to their attractiveness (see Figure 6). The most attractive options lie in the top right corner of the chart. The attractiveness is also denoted by the color scale - from red scale(unattractive) to green scale (attractive).
The user can choose the option and analyze it further (see Figure 7): the highlighted bubble shows details on:

- number of actions included in the option;
- the quantified attractiveness (calculated as combination of feasibilities and NPS impact of single actions in the option);
- the combined overall NPS impact.

Furthermore, the data table shows raw text comments from customers associated with the particular areas (aspects), divided into negative and positive columns (see Figure 7). Each comment can be investigated further by analyzing the whole survey and context in which it was expressed, as each comment is identified with Work Order ID.

8 Future work

Currently, the system is driven by the knowledge extracted from questionnaires. Our plan is to make it adaptive to text-only data, as the structured forms of surveys will be replaced by open-ended questionnaires in the future. Another

challenge lies in the system's efficiency of mining processes and we need to develop new methods of optimizing them by using distributed environment.
In general, there are many industrial solutions developed in recent years that are based on aspect-based sentiment analysis and text analytics. However, we recognized that although a lot of work has been done in the research community in this area, the problem is still far from being solved. Also, the research focus has been mainly on electronic products, hotels and restaurants. There are still novel ideas needed to study different ranges of domains. Domain and context-dependent sentiments remain to be highly challenging. There is a need to build integrated systems that try to deal with different problems together in an interactive way. As of now, a completely automated and accurate solution is nowhere in sight. On the other hand, there is still a huge and real demand in industry for such systems because every business wants to know how customers perceive their services or products.

References

1. Kuang, J. *Hierarchically structured recommender system for improving NPS.* The University of North Carolina at Charlotte, ProQuest Dissertations Publishing, 2016.
2. Kuang, J., Ras, Z.W., Daniel, A. *Hierarchical Agglomerative Method for Improving NPS.* Proceedings of the International Conference on Pattern Recognition and Machine Intelligence, LNCS, Vol. 9124, Springer, 2015, 54-64
3. Kuang, J., Ras, Z.W., Daniel, A. *Personalized Meta-Action Mining for NPS Improvement.* Foundations of Intelligent Systems, Proceedings of ISMIS 2015 in Lyon, France, LNAI, Vol. 9384, Springer, 2015, 73-80
4. Kuang, J., Ras, Z.W. *In Search for Best Meta-Actions to Boost Businesses Revenue.* Proceedings of the Conference on Flexible Query Answering Systems 2015, in Krakow, Poland, Advances in Intelligent Systems and Computing, Vol. 400, Springer, 2015, 431-443
5. Im, S., Ras, Z., Tsay, L. *Action reducts.* In: Foundations of Intelligent Systems. Proceedings of ISMIS 2011 Symposium, LNAI, Vol. 6804, Springer, 2011, 62-69
6. Tzacheva A., Ras Z.W. *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
7. Liu B. *Sentiment analysis and subjectivity.* Handbook of natural language processing, Vol. 2, 2010, 627-666
8. Marneffe M. D., Manning C. *Stanford typed dependencies manual.* Technical report, Stanford University, 2008.
9. Wang K., Jiang Y., Tuzhilin A. *Mining actionable patterns by role models.* In Proceedings of the 22nd International Conference on Data Engineering, IEEE Computer Society, 2006, 16-25
10. Hu, M., and Liu, B. *Mining and summarizing customer reviews.* Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD 04), (pp. 168-177). New York.
11. Ras Z. W., Wieczorkowska A. *Action-rules: How to increase profit of a company.* In "Principles of Data Mining and Knowledge Discovery", Proceedings of PKDD'00, Lyon, France, LNAI, No. 1910, Springer, 2000, 587-592