DOI: 10.15439/2016F372
ACSIS, Vol. 9. ISSN 2300-5963

Identification of Product's Features Based on Customer Reviews

Katarzyna Smietanka
University of Warsaw
Banacha 2, 02-097 Warszawa, Poland
katarzyna.smietanka@students.mimuw.edu.pl

Abstract—**In recent years an e-commerce has become more and more popular. This fact is mainly related to a low cost of running a business, wide access to a large group of potential customers and ease of advertising. Analysis of products' reviews can lead to valuable insights for both customers and manufacturers. Owing to positive reviews a future customer may be convinced to buy the product. A number of reviews for one product can amount to even hundreds what makes it hard for a potential buyer to read them all. The main aim of this paper is to present a method for mining reviews considering products' features, extracting products' features and preparing a summary of reviews. For that purpose a new promising technique—*Rule-Based Similarity Model* is used. The performance of the algorithm has been verified on online product review articles.**

Keywords*—*opinion mining; data mining; Rule-Based Similarity Model; Natural Language Processing; sentiment mining;

I. Introduction

THE utilization of websites aggregating customer's reviews as a source of information has increased rapidly with the expansion of e-commerce. More and more people decide to buy, even everyday products on the Internet. In order to enhance customer shopping experience and satisfaction, merchants give a possibility to share feedback. Usually, such a form of a review is unlimited and unstructured which encourages customers to express opinion of a product or a service. There are lots of websites on which we can find customers' opinions about products or services. The most popular are *Amazon*, *Google Shopping*, *Google Maps* - reviews about places and *OPiNEO* - a polish web page aggregating products' reviews. As more and more people are eager to write reviews, some popular products or services can get a huge number of reviews. Some of these written reviews can be extremely long, but only a few sentences focus on product's features. Taking into account the unlimited form of a feedback and a huge number of reviews, it may be difficult for a potential customer to analyse all of them. It justifies the need for an automatic system to extract key sentences from reviews considering products' features.

The knowledge about others people's opinion is not only beneficial for a potential buyer, but it is also useful for manufacturers. They can keep track of both positive and negative aspects of a product. In a case of negative opinions, manufacturers can decide to improve the quality of the product to enhance people's satisfaction. Thanks to this, it is possible to decrease the number of future complaints. Moreover, it helps to build a positive image of the products and services.

Usually feature extraction algorithms in customers' reviews are decomposed into three subtasks: (i) identification of features of a product, (ii) identification of opinions expressed about features of a product (iii) determination a sentiment orientation of the opinions [1].

II. Related Works

In [2] K. Kahn, B. Baharudin and A. Khan proposed hybrid patterns technique for features identification. The hybrid pattern is a combination of dependency patterns, which exploits grammatical structure and contextual rules. Most of the patterns derive from observation and empirical analysis. Patterns: *NN, NN NN, JJ NN, NN NN NN, JJ NN NN, JJ JJ NN* (*NN* - noun, *JJ* - adjective) are potentially exploited for candidate selection of product feature. Additionally authors created new groups of patterns: *Definite Base Noun* (definite article *the* before the noun phrase), *Linking Verb Based Noun Phrases* (based on assumption that linking verbs between *JJ* and *NN* provide best clues for opinion expressions) and *Preposition Based Noun Phrases* (noun phrases with *of* preposition).

Hiu and Liu in [3] presented a system which uses association rule mining [4] to extract nouns and noun phrases as feature candidates from product reviews. The main parts of their opinion summary system are: (i) part-of-speech tagging, (ii) frequent feature generation (iii) feature pruning (iv) opinion word extraction (v) infrequent feature identification [1]. In (i) stage, linguistic parser identifies noun and noun phrases. For generation frequent features (ii) authors use found expressions in previous stage. Other components of a sentence than noun/noun phrases are unlikely to be product features. In order to find frequent features, authors run [5] that bases on *association rule mining* technique. The main purpose of (iii) is to remove uninteresting and redundant features generated by [5] algorithm. There are two types of pruning: *compactness pruning* - remove those candidate features whose words do not appear together and *redundancy pruning* - remove redundant features that contain single word (e.g. *life* is not a useful feature while *battery life* is meaningful feature). In (v) is used an idea described in Section IV (2). The last stage of the system is identification of opinion words in sentences and semantic orientation using *bootstrapping* technique and *WordNet*.

In paper [6], authors presented aspect-based summary model, where a summary is built by extracting relevant aspects of a service. Partially that approach bases on ideas from [3]. A novel aspect of created models is that they exploit user-provided labels and domain-specific characteristics of service reviews to increase quality.

Authors A. Ghobadi and M. Rahgozar [7] approached the problem using an ontology to extract the products' information. Ontology is defined as concepts, their relationships and concepts instances of a specific domain. Required information is automatically extracted from dependency patterns.

In [8] authors proposed a method for aspect extraction based on *Double Propagation* with aspect recommendations: semantic similarity-based and aspect association-based. Semantic similarity-based recommendation aims to solve the problem of missing synonymous aspects of DP using word vectors from a large corpus. Aspect association-based recommendation is based on an idea that many aspects are correlated or co-occur across domains.

III. Analysis of the problem

A. Difficulties in analysing product reviews

The main problem connected with identifying sentences containing product's features is the unstructured form of reviews. Usually, reviews are rather long, but they only partially cover the topics regarding real aspects of a service or a product. Moreover, most of the opinions are written in an informal way, it often happens that it is not consistent with language grammar. Authors of reviews use different types of abbreviations (*g8* - meaning great, or *imo* - in my opinion) and make a digression, what significantly complicates a task of researching reviews. Another problem is connected with mapping implicit aspect expressions to aspects of a product. It is very hard to properly extract and identify that aspect, e.g. *fit in pockets* in the sentence *"This phone will not easily fit in pockets."* [9].

The problem of aspect and entity coreference resolution is also present in the task of feature identification. It has been extensively researched recent years. A coreference determines which mentioned entities or aspects refer to the same entity. For example in sentences: *"The Sony camera is better than the Canon camera. It is cheap too."*, *it* refers to *Sony*, because of the way how people express their opinions [9].

B. Review formats

There are several types of review formats - collecting customers' opinions about the product. The most popular are: (a) *pros and cons* - a user separately describes positive and negative sides of the product, (b) *pros, cons and review* - despite of pros and cons opinion, user is asked to write whole review (c) *unlimited format* - only text of the review without explicit division into positive and negative aspects (d) *questionnaires* - a structured opinion, the user expresses his opinion by while choosing one of available option. As it is suggested in [10], the review format should be taken into consideration choosing a proper algorithm for extracting features. In (a) and (c) reviewers usually use full sentences, but section *pros and cons* of (b) contains concise and short expressions. The precise and accurate format of (d) makes it easy to analyse and aggregate users' opinions.

C. Main types of opinions

In [11], *Jindal* and *Liu* proposed two types of opinions: (i) a regular opinion, (ii) a comparative opinion . In (i) sentiment expressions refer to some target entities. For example the sentence *"The touch screen is really cool"* presents the direct regular opinion about *the touch screen*, whereas the sentence *"After taking the drug, my pain has gone"* is an example of an indirect regular opinion. In comparative opinion (ii) a part of product is commented in relation to other one, e.g. *"iPhone is better than Blackberry"*.

D. Assumptions

This research is mainly focused on the problem of mining reviews containing features of a product, extracting those features and preparing a structured summary based on customers' reviews. The summary looks like the following:

```
Product Name: DVD_player
  1. Feature: size
    Number of extracted sentences: ...
    Individual reviews: ...
  2. Feature: ...
    Number of extracted sentences: ...
    Individual reviews: ...
```

where `Individual reviews` points to the specific sentences from reviews that comment about the feature. We are interested in all features of the product mentioned by customers, independently of the opinion polarity. In this paper, only unlimited format of reviews is considered and the focus is put on regular opinions.

IV. Aspect Extraction - Approaches

The problem of identification of products' features is strictly connected with aspect extraction, which is a part of an information extraction task. Focusing on the explicit aspect extraction, in [11] authors proposed four main approaches:

1) Extraction based on frequent nouns and noun phrases.
2) Extraction by exploiting opinion and target relations.
3) Extraction using supervised learning.
4) Extraction using topic modeling.

The (1) approach finds aspects that are nouns and noun phrases from a large number of reviews in a given domain. Nouns and noun phrases are identified by a part-of-speech tagger and their occurrence frequencies are counted. The way of solving the problem is justified by the observation that people commenting on products use similar vocabulary. Moreover, it is based on the assumption that frequently talked nouns are important and genuine features of the product. In (2), the relationships between targets of opinion are exploited to extract aspects. It is based on an idea that the same sentiment word can be used to describe or modify different aspects [3]. Supervised learning algorithms (3) need manually labeled data for training. The most dominant methods are *Hidden Markov Models* and *Conditional Random Fields*. Additionally, supervised learning methods usually use a set of domain-independent characteristics such as word distance, syntactic dependency, tokens or part-of-speech tagging. (4) is an unsupervised learning method, which uses statistical topic models. It assumes that each document consists of a mixture of topics, each topic is a probability distribution over words. Two main models are *Probabilistic Latent Semantic Analysis* and *Latent Dirichlet Allocation*.

V. Introduction to Rough Sets

Rule-Based Similarity Model that is presented in Section VI derives from the theory of rough sets (definitions presented in this section comes from [12]). The theory of rough sets, proposed by Zdzislaw Pawlak in 1981 [13], provides a mathematical formalism for reasoning about imperfect data and knowledge. In the rough set theory, available knowledge about object $u \in U$ (U is a finite non-empty set of objects) is represented as a vector of information about values of its *attributes*. An attribute can be treated as a function $a : U \rightarrow V_a$ that assigns values from a set V_a to objects from U. All available information about objects from U can be stored in a structure called an *information system* - $\mathbb{S}$, which is defined as a tuple $\mathbb{S} = (U, A)$, where A is a finite non-empty set of attributes.

Conditional attributes are those attributes about which information about values of all attributes from A can be obtained for any object, including those which are not present in U. A *decision attribute* is an attribute, which can be used to define a partitioning of U into disjoint sets. An information system with a defined decision attributes is called a *decision system* and is denoted by $\mathbb{S}_d = (U, A \cup \{d\})$, where $A \cap \{d\} = \emptyset$.

The rough set theory is often utilized to provide description of concepts from the considered universe. Any concept can generally be associated with a subset of objects from U which belong or match to it. Decision attributes in a decision system can usually be interpreted as expressing the property of belongingness to some concept. Given some information about characteristics (values of attributes) of objects corresponding to considered concept, it can be described using a *decision logic language*. Decision logic language L_A is defined over an alphabet consisting of a set of attribute constants (names of attributes from A) and a set of attribute value constants (symbols representing possible attribute values). Atomic formulas of L_A are attribute-value pairs $a = v$, where $a \in A$ and $v \in V_a$. Each description (a formula) ϕ in a decision logic language L_A can be associated with a set of objects from U that satisfy it.

Knowledge about dependencies between conditional attributes and decision attributes of a decision system are often represented using special formulas called *decision rules*.

Definition V.1 (Decision rules)**.** *Let A and D be conditional and decision attribute sets of some decision system and let $L_{A \cup D}$ be a decision logic language and π be a formula of $L_{A \cup D}$. We will say that π is a decision rule iff the following conditions are met: (i) $\pi = (\phi \rightarrow \psi)$, (ii) ϕ and ψ are conjunctions of descriptors, (iii) ϕ is a formula of L_A and ψ is a formula of L_D.*

The right hand side of a decision rule $\pi = (\phi \rightarrow \psi)$ (i.e. ψ) is called a *successor* of a rule (denoted by $rh(\pi)$) and the left hand side (denoted by $lh(\pi)$) is called a *predecessor* (i.e. ϕ).

There is also a different type of rules - *inhibitory rules*, which is useful for analysing dependencies in data with multiple decision values.

Definition V.2 (Inhibitory rules)**.** *Let A and D be conditional and decision attribute sets of a decision system and let $L_{A \cup D}$ be a decision logic language and π be a formula of $L_{A \cup D}$. We will say that π is an inhibitory rule iff the following conditions are met: (i) $\pi = (\phi \rightarrow \neg\psi)$, (ii) ϕ and ψ are conjunctions of descriptors, (iii) ϕ is a formula of L_A and ψ is a formula of L_D.*

An inhibitory rule tells us that an object which satisfies the predecessor of this rule cannot belong to a pointed decision class. The inhibitory rules can be seen as a complement to decision rules as they often provide means to classify objects which are difficult to cover by the traditional rules.

Usefulness of a rule for prediction of decision classes of new objects can be quantitatively assessed using rule quality measures: *support* and *confidence*. The support of a rule π is defined as

$$supp(\pi) = \frac{|lh(\pi)(U)|}{|U|} \tag{1}$$

and the confidence of π is:

$$conf(\pi) = \frac{|lh(\pi)(U) \cap rh(\pi)(U)|}{|lh(\pi)(U)|}. \tag{2}$$

In the rough set theory any arbitrary set of objects X can be approximated within an information system $\mathbb{S} = (U, A)$ by a pair of definable sets $App(X) = (\underline{X}, \overline{X})$, called a rough set of X in S. The set $\underline{X}$ is the largest and the set $\overline{X}$ is the smallest definable set which contains X. The sets $\underline{X}$ and $\overline{X}$ are called a *lower* and *upper approximation* of X in S, respectively.

VI. Rule-Based Similarity Model

In [12], author proposed a similarity model, called Rule-Based Similarity (RBS). In RBS the similarity is assessed by examining whether two objects share some binary higher-level features. Features which are relevant for a considered similarity context are extracted from data and their importance is assessed based on available data.

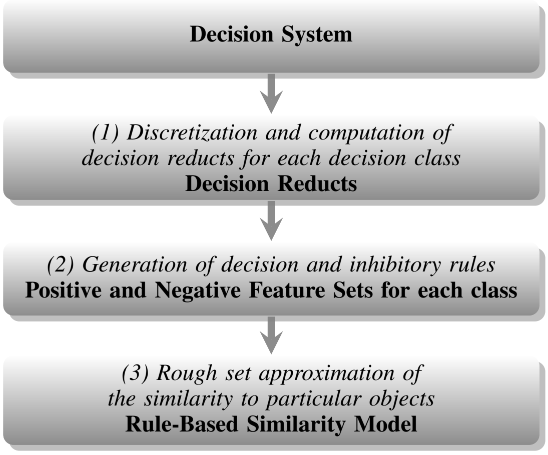
Decision System
(1) Discretization and computation of
decision reducts for each decision class
Decision Reducts
(2) Generation of decision and inhibitory rules
Positive and Negative Feature Sets for each class
(3) Rough set approximation of
the similarity to particular objects
Rule-Based Similarity Model

Fig. 1. A construction schema of the RBS model [12].

In the presented RBS model similarity is learnt in a classification context, which is defined by a decision attribute in a data set. In the (1) step of the model, raw attributes values are transformed into a symbolic representation that resembles basic qualitative characteristics of objects. This approach is based on a way in which people assess similarity. Tversky [14] suggested that people rarely think in terms of exact numbers but rather tend to operate on binary characteristics (e.g. *an object is large* or *an object is round*). In a practical data analysis, it is enough to divide numerical attribute values into intervals representing qualitative symbols. This process is done by applying a heuristic discretization technique. For each decision class are computed *decision reducts*. Reducts are an important tool for selecting informative features in the rough set theory, they help to discover redundancy and dependencies between attributes. A *decision reduct* is a subset of attributes that discriminates objects from different decision classes and is minimal in a sense that no attribute can be removed from a reduct without losing the properties of discernibility.

In (2), higher-level features that are important for a notion of similarity are created using a rule mining algorithm. Each of those features is defined by the left-hand side of a rule - $lh(\pi)$. There are two types of rules: *decision rules* and *inhibitory rules*, described in Section V. Decision rules aim is to provide partial descriptions of concepts indicated by the decision attribute. They are used to predict decision classes of new objects. Inhibitory rules specify what kind of objects cannot belong to a pointed decision class. Depending on a type of a rule, the corresponding feature can be useful either as an argument for or against the similarity to a matching object.

The induction of rules in RBS may be treated as a process of learning aggregations of local similarities from data. Features defined by predecessor of the rules express higher-level properties of objects.

The last stage includes rough set approximation of the similarity to particular objects. The problem of learning the similarity relation in RBS is closely related to searching for a relevant approximation space. Formally, let $F^+_{(i)}$ and $F^-_{(i)}$ be the sets of binary features derived from the decision and the inhibitory rules, generated for i-th decision class:

$$F^+_{(i)} = \{\phi : (\phi \rightarrow (d = i)) \in RuleSet_i\} \tag{3}$$

$$F^-_{(i)} = \{\phi : (\phi \rightarrow \neg(d = i)) \in RuleSet_i\} \tag{4}$$

where $RuleSet_i$ is a set derived from a reduct associated with the i-th decision class. The rule set may be generated using any rule mining algorithm. For efficiency in practical applications of the model it is necessary to require that the generated sets of rules $RuleSet_i$ be minimal. It means that there is no rule $\pi \in RuleSet_i$ that could be removed without reducing the set of covered objects.

A feature ϕ is a decision logic formula, we will say that an object u, described in a decision system $\mathbb{S} = (\mathbb{U}, \mathbb{A})$, has a feature ϕ iff $u \models \phi$. A set of all object from U that have the feature ϕ will be denoted by $\phi(U)$. In RBS a *similarity relation* is approximated by means of approximating multiple concepts of being similar to a specific object. It consists of those objects from U which share with u at least one feature from the set F^+_i, where i is a decision class of u ($d(u) = i$):

$$SIM_i(u) = \bigcup_{\phi \in F^+_i \wedge u \models \phi} \phi(U) \tag{5}$$

the approximation of the dissimilarity to u is a set of objects from U which have at least one feature from $F^-_{(i)}$ that is *not in common* with u:

$$DIS^0_i(u) = \bigcup_{\phi \in F^-_i \wedge u \not\models \phi} \phi(U) \tag{6}$$

The set of objects that have at least one feature from $F^-_{(i)}$ that *in common* with u will be denoted by

$$DIS^1_i(u) = \bigcup_{\phi \in F^-_i \wedge u \models \phi} \phi(U) \tag{7}$$

The functions SIM and DIS are used for the approximation of the similarity and the dissimilarity to objects from U. The assessment of degree in which an object u_1 is similar and dissimilar to u_2 is done using two functions (abbreviations $SIM(u) = SIM_{(d(u))}(u)$; $DIS(u) = DIS^0_{(d(u))}(u)$ are written when the decision for an object u is known):

$$Similarity(u_1, u_2) = \frac{|SIM(u_1) \cap SIM_{d(u_1)}(u_2)|}{|SIM(u_1)| + C_{sim}} \tag{8}$$

$$Dissimilarity(u_1, u_2) = \frac{|DIS(u_1) \cap DIS^1_{d(u_1)}(u_2)|}{|DIS(u_1)| + C_{dis}} \tag{9}$$

In the above formulas C_{sim} and C_{dis} are positive constants which can be treated as parameters of the model. The similarity function of the RBS model combines values of $Similarity$ (Sim) and $Dissimilarity$ (Dis) for a given pair of objects:

$$Sim_{RBS}(u_1, u_2) = F(Sim(u_1, u_2), Dis(u_1, u_2)) \tag{10}$$

where $F : \mathbb{R} \times \mathbb{R} \rightarrow \mathbb{R}$ can be any function that is monotonically increasing with regard to its first argument and monotonically decreasing with regard to its second argument. Detailed description of the similarity function is provided in [12].

VII. Proposed Algorithm

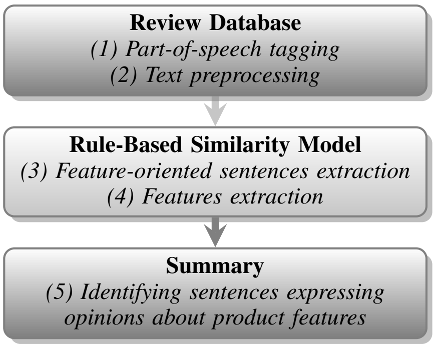
Review Database
(1) Part-of-speech tagging
(2) Text preprocessing
Rule-Based Similarity Model
(3) Feature-oriented sentences extraction
(4) Features extraction
Summary
(5) Identifying sentences expressing opinions about product features

Fig. 2. A construction schema of extraction feature summary system.

Figure 2 provides a general overview of the construction of a summary system for the feature extraction. Product names and reviews associated with a product are an input of the system. Reviews for a single product are split into sentences and tagged with decision class - *positive* if a sentence contains feature otherwise *negative*. This represents the true reference classification. As an output, for each product system creates a summary of the reviews in the following format: product feature and list of sentences from reviews correlated with that feature.

In (2) preprocessing of reviews is performed, what plays an important role in text mining discipline. Text preprocessing stage includes stop words and punctuation removal, stemming. The feature extraction task is focused on finding features that appear explicitly as noun or noun phrases in the reviews. To enhance the quality of feature-oriented sentences extraction, adjectives and adverbs are extracted. It derives from observation, that usually when people express their opinion about an aspect of a product, they use opinion words such as adjectives and adverbs. To identify noun/noun phrases, adjectives and adverbs (1) from the reviews, part-of-speech tagger - NLTK parser (*Natural Language Toolkit*) is used. For each sentence, identified and preprocessed parts of speech are saved in the review database. The following shows a sentence *"I am very pleased with its quality"* with the POS tags. For instance, a tag NN indicates a noun and JJ indicates an adjective.

$$[(I, NN), (am, VBP), (very, RB), (pleased, JJ), \\ (with, IN), (its, PRP), (quality, NN)] \quad (11)$$

In next step of the process, Rule-Based Similarity Model, which is described in Section VI, is built. Documents, sentences from reviews, are transformed into a document-term matrix which represents the frequency of terms occurring in a collection of documents. This is an input to RBS model. In a document-term matrix, rows correspond to sentences in the collection and columns correspond to terms. To determine the value that each entry in the matrix, we use a *term-frequency* scheme, defined as: $f_{t,d}$ of term t in document (sentence) d is the number of times that t occurs in d. The similarity function of RBS model is used to perform a classification of previously unseen objects of textual documents into groups: *positive* - sentences containing product features otherwise *negative*.

The attributes from the left hand side of *decision rules* - $lh(\pi)$ $(\pi = (\phi \rightarrow \psi))$ calculated in RBS model, represents a set of candidates for product features - arguments for the similarity to a matching objects. The formula ϕ from $lh(\pi)$ is expressed by a compound formula - the attribute value pairs. The value from atomic formula (attribute-value pair) is an interval that represents a required frequency of an attribute (term).

The taken approach is justified by the reviews representation as an input to RBS model, association mining algorithm for creating rules and empirical observation of human expressions. On the other hand, *inhibitory rules* are useful as an argument against the similarity to matching objects. In (5) algorithm identifies sentences from reviews expressing opinions about product features and creates a summary.

The proposed algorithm partly depends on opinion identification - sentiment words (adjectives and adverbs) are an input to RBS model. Moreover, the algorithm uses empirically defined threshold for a minimum support and a maximum length of rules, which are created in an association mining process.

VIII. Experiments

The approach was tested on dataset of customer reviews for five products collected from *Amazon.com* and *Cnet.com* as described in [3]. Dataset contains customer reviews focusing on electronic products: a DVD player - *Apex AD2600 DVD player*, digital cameras - *Canon G3* and *Nikon coolpix*, a cell phone - *Nokia 6610* and a MP3 player - *Creative Labs*.

TABLE I. Product review dataset.

Product Name	No. of reviews	No. of sentences
Digital camera1	45	597
Digital camera2	34	346
Cell phone	41	546
MP3 player	95	1716
DVD player	99	739

Originally dataset was annotated by *Hu* and *Liu* [3], they define a product feature as a characteristic of the product which customers have expressed an opinion about. Opinion is a statement which explicitly defines an attitude towards feature, which is positive or negative. The dataset was reannotated by authors of [1], in order to put the main focus on feature extraction. Annotated terms as features satisfy one of the criteria: (i) part of relationship between product and a feature (ii) attribute of product, e.g. *design* - attribute of camera (iii) attribute of a known feature, e.g. *battery life* - it is an attribute of *battery*. Experiments were conducted on dataset tagged by [1].

TABLE II. Detailed summary of dataset.

Product Name	Dataset tagged by [3]		Dataset tagged by [1]	
	Distinct	Total	Distinct	Total
Digital camera1	100	257	161	594
Digital camera2	74	185	120	340
Cell phone	109	310	140	470
MP3 player	180	736	231	1031
DVD player	110	347	166	519

The effectiveness of proposed algorithm is evaluated by standard evaluation methods: *precision*, *recall* and *f1-score*. Precision answers the question *"How many selected items are relevant?"*, while recall expresses an idea about *"how many relevant items are selected?"*. Precision is the fraction of retrieved instances that are relevant, while recall is the fraction of relevant instances that are retrieved. F1-Score is a measure of a test's accuracy, it can be interpreted as a weighted average of the precision and recall. This measure reaches its best value at 1 and its worsts at 0.

TABLE III. Summary of rule sets from RBS model for decision classes.

Product Name	No. of decision rules for positive class	No. of decision rules for negative class
Digital camera1	44	5
Digital camera2	32	9
Cell phone	31	13
MP3 player	98	27
DVD player	38	1
Average	48	11

In Table III is presented a summary of unique rules sets which were generated by the RBS model. It is easy to notice that a number of decision rules for a positive class are several times greater than a number of decision rules for a negative class. It is mainly connected with empirically set thresholds for a minimum negative and positive confidence for association mining rules. In order to increase a recall measure, minimum confidence for the positive is smaller than for the negative decision class.

A. The effectiveness of feature-oriented sentence extraction

In the task of mining reviews containing product's features (*feature-oriented sentence extraction*), received results from proposed algorithm (RBS similarity from Section VII) are compared with *Support Vector Machine* (SVM) [15] *with hidden concepts* (SVM with concepts). The idea of SVM algorithm is to find boundaries in the input feature space. In order to obtain good performance of SVM in the classification process, before training a classifier, dimension reduction is performed and documents are represented in a space of higher-level terms. For that purpose is used the typical approach as Singular Value Decomposition (SVD), which goal is to find a representation of document term matrix as a product of lower-rank matrices. Using calculated SVD matrix and selected hidden concepts, which are two-terms phrases, documents are represented in a space of that concepts. The name of that technique is *Latent semantic analysis* (LSA) [16], it is a method for extracting and representing contextual-usage of words by statistical computations applied to a large corpus of text. In the following experiment, SVM algorithm is applied using linear kernel. In [17] author justified a good performance of SVM in a text classification context. Combining LSA with SVM is a common method for text classification [17].

As in [12], the quality of the compared models was assessed using two different measures - *mean accuracy* (Mean) and *balanced accuracy* (Balanced). The mean classification accuracy, defined as:

$$Mean = \frac{|\{t \in DataSet : p(t) = d(t)\}|}{|DataSet|} \tag{12}$$

where $DataSet$ is a set of test objects; $p(t)$ is a predication of a decision class for an object t and $d(t)$ is an expected decision class for an object t, was estimated using 3-fold cross-validation technique [18]. The balanced accuracy is calculated by computing standard classification accuracies $Mean_i$ for each decision class and then averaging the result over classes: $i \in \{positive,\ negative\}$ [12].

$$Mean_i = \frac{|\{t \in DataSet : p(t) = d(t) = i\}|}{|\{t \in DataSet : d(t) = i\}|} \tag{13}$$

$$Balanced = \frac{Mean_{positive} + Mean_{negative}}{2} \tag{14}$$

TABLE IV. COMPARISON OF MEAN AND BALANCED ACCURACY FOR FEATURE-ORIENTED SENTENCE EXTRACTION.

Product Name	Proposed algorithm from Section VII		SVM with hidden concepts	
	Mean	Balanced	Mean	Balanced
Digital camera1	0.724	0.704	0.427	0.494
Digital camera2	0.699	0.675	0.396	0.517
Cell phone	0.679	0.685	0.460	0.488
MP3 player	0.715	0.714	0.536	0.511
DVD player	0.743	0.744	0.539	0.526
Average	0.712	0.704	0.472	0.507

For all the product reviews in the classification problem, RBS model achieved better results than SVM with concepts. Both an average *mean accuracy* and *balanced accuracy* measures are higher for RBS model. The statistical significance of mean differences in results between classification algorithms was verified using *a paired t-test*. A null hypothesis was tested that obtained mean measurements for data sets have equal means. Difference in means is considered significant if *p-value* of the test is lower than 0.01. For presented results, the null hypothesis is rejected, the performance of RBS similarity model is better than for SVM model with hidden concepts.

B. The effectiveness of feature extraction

In order to illustrate the effectiveness of proposed feature extraction technique, gained results are compared with widely available Content Term Extraction (CTE) algorithm [19] (similarly as in [9]). As proposed algorithm disregards the original ordering of the terms in sentences, it is impossible to directly evaluate gained results. Evaluation of extracted features is performed on single-word features, multi-word features from annotation are divided into separate terms.

The proposed approach to problem from Section VII is assessed by two different measures: *global quality criteria* and *feature-oriented quality criteria*. The global quality criteria examine the algorithm's performance on the task of extracting features from the collection of reviews. This relates to the main task of creating the summary of features, which involves: identifying features of the product that customers expressed their opinion on and finding review sentences corresponding to extracted features. The feature-oriented quality criteria are focused on an evaluation of how many unique features defined by an expert was found by the algorithm. For that measure, it is not important from which sentence and document an algorithm extracted a certain feature.

TABLE V. COMPARISON OF RECALL AND PRECISION FOR GLOBAL QUALITY CRITERIA IN DATASET TAGGED BY [1].

Product Name	Proposed algorithm from Section VII		Content Term Extraction	
	Precision	Recall	Precision	Recall
Digital camera1	0.610	0.545	0.240	0.485
Digital camera2	0.586	0.545	0.292	0.586
Cell phone	0.562	0.528	0.277	0.531
MP3 player	0.633	0.631	0.195	0.390
DVD player	0.685	0.644	0.209	0.388
Average	0.615	0.579	0.243	0.476

Table V shows the comparison of recall and precision for *global quality criteria* for two different algorithms: proposed algorithm based on RBS model (Section VII) and Content Term Extraction. It can be observed that both of these measures are lower for CTE algorithm. The major reason of poor precision is the fact that CTE generates a large number of terms, which are not product features.

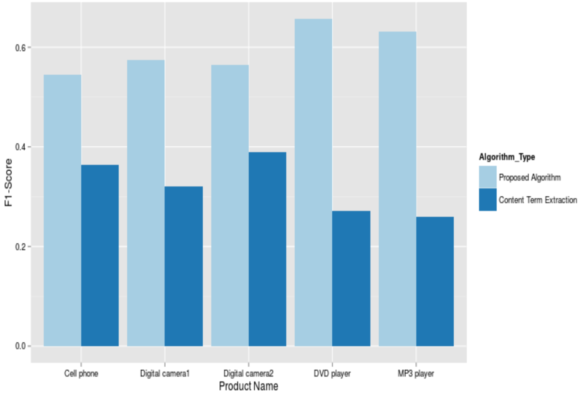

Fig. 3. Comparison of F1-Score for algorithms.

Comparing results with *Likelihood Test approach* presented in [1] (*Table 8. Feature extraction results on instance level*), the proposed algorithm receives better average recall, but the average precision is slightly lower. In contrast to *Association Mining approach* [1], the average precision is improved for the algorithm based on RBS model. The proposed algorithm's performance on the task of identifying features from the collection of reviews turned out to be quite promising.

TABLE VI. COMPARISON OF RECALL AND PRECISION FOR FEATURE-ORIENTED QUALITY CRITERIA IN DATASET TAGGED BY [1].

Product Name	**Proposed algorithm from Section VII**		**Content Term Extraction**	
	Precision	Recall	Precision	Recall
Digital camera1	0.811	0.088	0.135	0.750
Digital camera2	0.806	0.107	0.164	0.805
Cell phone	0.673	0.083	0.149	0.808
MP3 player	0.703	0.110	0.115	0.775
DVD player	0.723	0.112	0.145	0.730
Average	0.743	0.100	0.142	0.774

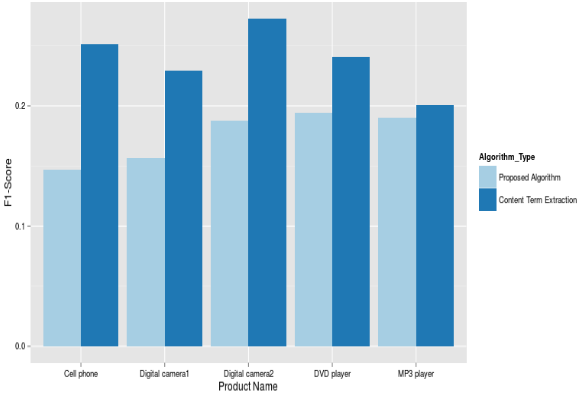

Fig. 4. Comparison of F1-Score for algorithms.

Considering *feature-oriented quality criteria* it can be noticed that the *F1-Score* measure for product reviews is better for CTE algorithm. However, the difference between that measure for the proposed and CTE algorithm is not so meaningful. The low recall for the proposed algorithm is a result of a construction of a decision reduct, which is a minimal subset of attributes that discriminates objects from different decision classes. The high recall for CTE approach derives from the fact that CTE generates a lot of terms, but it results in a low precision. The average precision for the proposed algorithm is about 0.743, whereas for CTE algorithm is 0.142. *Likelihood Test Approach* [1] (LTA) has a similar property - a high precision and a low recall as the algorithm based on RBS. The LTA average recall is about 0.104 and the precision amounts to 0.804.

IX. Conclusions

In this paper, the problem of identification of product's features based on customer reviews and the new method which based on Rule-Based Similarity Model were discussed. Moreover, difficulties in analysing product reviews, review formats and methods for extracting product features were described. In particular, there were presented results of conducted experiments in comparison to other algorithms and evaluated gained results with two measures: *global quality criteria* and *feature-oriented quality criteria.*

Additionally, this paper shows that algorithm based on Rule-Based Similarity Model could be successfully used for the problem of feature extraction and feature-oriented sentence extraction problem. Proposed algorithm does not require any prior knowledge despite labelled data to train the RBS model.

In the future, this work will be extended with evaluation of the performance of the proposed algorithm on other data sets. The main focus will be put on improving the recall of the described method with regard to the feature-oriented quality criteria.

References

[1] L. Ferreira, N. Jakob, and I. Gurevych, "A comparative study of feature extraction algorithms in customer reviews," in *Proceedings of the 2th IEEE International Conference on Semantic Computing (ICSC 2008), August 4-7, 2008, Santa Clara, California, USA*, 2008, pp. 144–151. [Online]. Available: http://dx.doi.org/10.1109/ICSC.2008.40

[2] K. Khan, B. Baharudin, and A. Khan, "Identifying product features from customer reviews using hybrid patterns," *Int. Arab J. Inf. Technol.*, vol. 11, no. 3, pp. 281–286, 2014. [Online]. Available: http://www.ccis2k.org/iajit/index.php?option=com_content&task=blogcategory&id=92&Itemid=353

[3] M. Hu and B. Liu, "Mining opinion features in customer reviews," in *Proceedings of the Nineteenth National Conference on Artificial Intelligence, Sixteenth Conference on Innovative Applications of Artificial Intelligence, July 25-29, 2004, San Jose, California, USA*, 2004, pp. 755–760. [Online]. Available: http://www.aaai.org/Library/AAAI/2004/aaai04-119.php

[4] R. Agrawal and R. Srikant, "Fast algorithms for mining association rules in large databases," in *VLDB'94, Proceedings of 20th International Conference on Very Large Data Bases, September 12-15, 1994, Santiago de Chile, Chile*, 1994, pp. 487–499. [Online]. Available: http://www.vldb.org/conf/1994/P487.PDF

[5] B. Liu, W. Hsu, and Y. Ma, "Integrating classification and association rule mining," 1998, pp. 80–86.

[6] S. Blair-goldensohn, T. Neylon, K. Hannan, G. A. Reis, R. Mcdonald, and J. Reynar, "Building a sentiment summarizer for local service reviews," in *In NLP in the Information Explosion Era*, 2008.

[7] A. Ghobadi and M. Rahgozar, "An ontology-based semantic extraction approach for b2c ecommerce."

[8] Q. Liu, B. Liu, Y. Zhang, D. S. Kim, and Z. Gao, "Improving opinion aspect extraction using semantic similarity and aspect associations," in *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence, February 12-17, 2016, Phoenix, Arizona, USA.*, 2016, pp. 2986–2992. [Online]. Available: http://www.aaai.org/ocs/index.php/AAAI/AAAI16/paper/view/11973

[9] C. C. Aggarwal and C. X. Zhai, *Mining Text Data*. Springer Publishing Company, Incorporated, 2012.

[10] M. Hu and B. Liu, "Opinion feature extraction using class sequential rules," in *Computational Approaches to Analyzing Weblogs, Papers from the 2006 AAAI Spring Symposium, Technical Report SS-06-03, Stanford, California, USA, March 27-29, 2006*, 2006, pp. 61–66. [Online]. Available: http://www.aaai.org/Library/Symposia/Spring/2006/ss06-03-013.php

[11] B. Liu, *Sentiment Analysis and Opinion Mining*, ser. Synthesis digital library of engineering and computer science. Morgan & Claypool, 2012. [Online]. Available: https://books.google.pl/books?id=Gt8g72e6MuEC

[12] A. Janusz, "Algorithms for similarity relation learning from high dimensional data," *Trans. Rough Sets*, vol. 17, pp. 174–292, 2014. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-54756-0_7

[13] Z. Pawlak, "Information systems theoretical foundations," *Inf. Syst.*, vol. 6, no. 3, pp. 205–218, 1981. [Online]. Available: http://dx.doi.org/10.1016/0306-4379(81)90023-5

[14] A. Tversky, "Features of similarity," *Psychological Review*, vol. 84, no. 4, pp. 327–352, 1977.

[15] "Support vector machines," https://cran.r-project.org/web/packages/e1071/vignettes/svmdoc.pdf, accessed: 2016-05-04.

[16] T. K. Landauer and S. T. Dumais, "Latent semantic analysis," *Scholarpedia*, vol. 3, no. 11, p. 4356, 2008. [Online]. Available: http://www.scholarpedia.org/article/Latent_semantic_analysis

[17] T. Joachims, "Text categorization with suport vector machines: Learning with many relevant features," in *Proceedings of the 10th European Conference on Machine Learning*, ser. ECML '98. London, UK, UK: Springer-Verlag, 1998, pp. 137–142. [Online]. Available: http://dl.acm.org/citation.cfm?id=645326.649721

[18] J. Demsar, "Statistical comparisons of classifiers over multiple data sets," *Journal of Machine Learning Research*, vol. 7, pp. 1–30, 2006. [Online]. Available: http://www.jmlr.org/papers/v7/demsar06a.html

[19] "Content term extraction using pos tagging," https://pypi.python.org/pypi/topia.termextract/, accessed: 2016-05-04.