Customer Product Review Summarization over time for Competitive Intelligence

Submitted: 13th December 2018; accepted: 28th December 2018

Kamal Amarouche, Houda Benbrahim, Ismail Kassou

DOI: 10.14313/JAMRIS_4-2018/28

Abstract:
Nowadays, Customer's product reviews can be widely found on the Web, be it in personal blogs, forums, or e-commerce websites. They contain important products' information and therefore became a new data source for competitive intelligence. On that account, these reviews need to be analyzed and summarized in order to help the leader of an entity (company, brand, etc.) to make appropriate decisions in an effective way. However, most previous review summarization studies focus on summarizing sentiment distribution toward different product features without taking into account that the real advantages and disadvantages of a product clarify over time. For this reason, in this work we aim to propose a new system for product opinion summarization which depends on the time when reviews are expressed and that covers the sentiments change about product features. The proposed system firstly, generates a summary based on product features in order to give more accurate and efficient information about different features. Secondly, classify the product based on its features in its appropriate class (good, medium or bad product) using a fuzzy logic system. The experimental results demonstrate the effectiveness of the proposed system to generate the real image of a product and its features in reviews.

Keywords: *Feature extraction, Fuzzy logic, Competitive intelligence, Opinion mining, Opinion Summarization, Sentiment analysis, SentiWordNet*

1. Introduction

Competitive Intelligence (CI) is a monitoring process of the competitive environment through which information is gathered, analysed and distributed in order to obtain results that will be carried out gradually for efficient support to business activity and help to make qualified decisions in relation to its competitors (Štefániková and Masàrovà, 2014 [25]). Traditionally, the capability of CI was greatly restricted to the lack of sufficient and reliable information sources about competitors (Xu et al., 2011 [28]). But nowadays, many people share opinions on a variety of topics and precisely on products/services on the Internet (e.g. E-commerce websites, social media..). These opinions present a new information source for CI (Amarouche et al., 2015 [3]). Summarizing these opinions automatically in some concise form could bring enormous benefits to business. Various research works (Hu and Liu, 2004 [13]; Zhuang et al., 2006 [32]; Wang et al., 2013 [26]; Kansal and Toshniwal, 2014 [15]; Asgarian and Kahani, 2014 [5]; Chen et al., 2015 [9]; Kangale et al., 2015 [14]; Zhou et al., 2016 [31]; Yang et al., 2016 [29]; Cho and Kim, 2017 [10]) have been conducted on opinion summarization systems that classify them into those that require a set of features (feature-based-summarization) and those that do not rely on the presence of features (non-feature-based summarization).

Feature-based-Summarization approaches concentrate on the features (called also aspects) of a specific product to produce a summary. They consist of three distinct steps such as, i) Identifying the important features of the product; ii) Identifying the opinion words and predict their polarities; iii) Generating the actual summary. Contrariwise, systems based on Non-feature-based Summarization produce a generalized summary without considering the features. This kind of summarization is useless in CI because it loses some detailed information, which is important to inform the decision maker of an organization regarding the development and marketing of a product/service. However, summaries based on features provide information about different features and show what customers usually try to search while referring to opinions that rend these types of summaries of great demand in many application domains like: recommender system (Ładyżyński and Grzegorzewski, 2014 [17]) and trust reputation (Rahimi and Bakkali, 2015 [23]) with different languages such as English (Kansal and Toshniwal, 2014 [15]), Chinese (Zhou et al., 2016 [31]), Arabic (Abd-Elhamid et al., 2016 [1]) and Italian (Maisto and Pelosi, 2014 [21]). However, the large availability of reviews shared in English allows researchers to focus on this language in their works. Fig. 1 shows an example of a feature-based summary about a cellphone. The 'battery life' and 'camera' are two product features among other features of this product. The positive count opinions of the battery life is 200, and 230 negative ones. While in the case of the 'camera', there are 200 positive and 30 negative opinions. For each product feature, the number of positive and negative opinions is also provided. With such a summary, we can see how many users have liked or disliked a specific feature.

After usage of the product by customers, their opinion may change on the same features. As indicated in reviews *1* and *2* about 'Samsung note 7' phone on gsmarena.com. In *Review 1*, the commentator shows a positive sentiment about feature 'battery'. In contrast to *Review 2*, we can notice that the sentiment about the same feature has changed over time. To put differently, the advantages and disadvantages of the

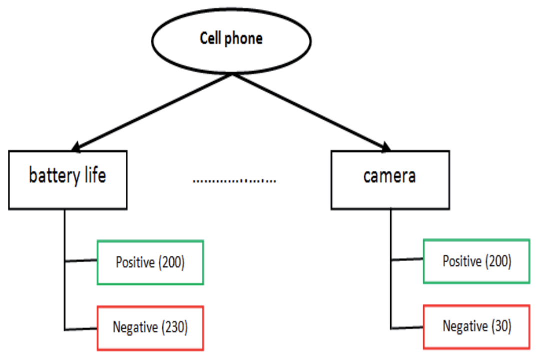


Fig. 1. Feature-based summary using positive and negative scores of smartphone

product around its features will be clarified in the reviews according to time and its use. Another factor that has an influence on the effectiveness of feature-based summary systems is when the product has been improved on one of its features, the results provided using these systems do not reflect the real image of the product after its improvement. All this shows the importance of the time axis to express opinions in order to generate useful summaries. However, the most current efforts produced a summary without taking into account the changes of sentiments about features over time. For this reason, an opinion summarization system that depends on time to express reviews is needed to cover these features changes if they exist.

Review 1 "Slim phone with bigger battery" date: 02 May 2016.

Review 2 "The phone is not problem the important problem is battery, he needs to create a brand new battery for Samsung galaxy note 7" date: 21 Oct 2016.

The objective of this paper is to propose a product review summary system which takes into considiration the time when reviews were expressed in order to produce an effective output reflecting the real image of the product in reviews. All in all, we summarize our contributions of this research as:

- We have introduced a new feature-based-summarization method which generates a summary that depends on time where sentiments are expressed around features.
- We have proposed in this system, a Fuzzy Logic process to assign the product to its corresponding class (good, medium or bad) that is important to being able to compare it with its competitors.
- We have conducted experiments using real data concerning mobile phones and hotels to show that our system does not refer to any domain information.

The rest of this paper is structured in the following sections. Section 2 discusses the related works. Section 3 formally defines the problems that will be solved in this work. The proposed system in detail is presented in Section 4. Section 5 discusses the experimental results. Finally, the conclusion and proposed future works are given in Section 6 of this paper.

2. Related Works

This section presents a review of some important previous works performed in the feature-based summarization field. Their contributions usually focus on the three major tasks which were mentioned above that are (1) feature identification, (2) feature-associated opinion classification, (3) feature-based summaries generation. As such, the different approaches proposed in these previous works are presented.

One of the earliest works was done by Hu and Liu (2004) [13] that proposed a system to generate feature-based summaries of products' customer reviews. First, nouns and noun phrases that frequently appeared in reviews were identified as candidate product features. Next, two types of pruning methods were used to remove unlikely features. Then, they extracted the adjectives as opinion words from the sentences that contained one or more features. Based on the orientation of these opinion words and negation words, the orientations of opinion sentences were generated. Finally, a statistical summary that shows features with their corresponding review sentences and their positive and negative numbers is produced. Liu et al. (2005) [20] proposed a framework called *Opinion Observer* for analyzing and comparing consumer opinions of competing products which gives the strength and weakness of each product in terms of various product features. The product features were extracted by the association miner. They used adjectives as opinion words and assign prior polarity to these by WordNet exploring method. Zhuang et al. (2006) [32] proposed an approach based on multi-knowledge to summarize movie reviews. First, a keyword list was built to find features and opinions. Then, they mined the relations between features and associate opinion words using dependency grammar graph and the building list. Next, the valid feature-opinion pairs were identified by applying these mined relations. Finally, a summary was generated based on the sentences that contained opinions or extracted features. Abulaish et al. (2009) [2] proposed a system that produced a graphical summary about a product features. Frist, they extracted features and opinion words using a semantic and linguistics analysis of reviews. Then, the polarity of opinions was detected using a lexical resource. Finally, they provided a feature-based summary of reviews in a graphical way.

But during the last few years, this problem has been an attractive research topic. In practice, much work has been devoted to perform this problem as a system.

Wang et al. (2013) [26] proposed a Web-based review summarization system called *SumView*. They extracted product features and grouped the sentences into feature relevant clusters using a Feature-based weighted Non-Negative Matrix Factorization algorithm. Then, a summary was generated by selecting the sentence that had the highest probability in each cluster.

Bafna and Toshniwal (2013) [6] proposed a dynamic system called *FBS* for feature-based opinion summarization. The product features were identified using a combination of association mining and probabilistic approaches. Next, they extracted the opinions associated to each feature and their polarities were detected based on lexicon dictionaries. Finally, the system generated a feature-based summary by extracting the relevant excerpts according to each feature-opinions pair and placing it into their respective feature-based cluster.

Kansal and Toshniwal (2014) [15] proposed a system called *ASAS* (Aspect Based Sentiment Analysis and Summarization) that took into account the context dependent opinion words. The product features were extracted using an approach like the one proposed by Bafna and Toshniwal (2013) [6]. They mapped each opinion word to its closest products features, then, applied some natural linguistic rules to find the polarity of context dependent words.

Asgarian and Kahani (2014) [5] designed a semantic framework for structured summarization. They extracted the features, analysed the sentiment and then integrated and summarized the opinions using a developed ontology. The framework of their proposed ontology shows the output results of the structured summarization as semantic data.

Chen et al. (2015) [9] proposed a system to handle Cantonese opinion mining. They built an opinion orientation dictionary to identify the orientation of opinion words, then explored some syntax rules to predict the opinion sentences orientation. Finally, they finished by a summarization which categorized all the related opinion sentences into positive or negative categories, they counted the respective numbers.

Kangale et al. (2015) [14] proposed a feature-based review-summary system. In the first step, the data were passed through opinion spam filter, which could detect fake reviews. After that, the features would be identified using association rule mining. Then, feature trimming was used to trim some of the unwanted features using the kinds of pruning that was described in Hu and Liu (2004) and added others called miscellaneous pruning which improved this task. They used database-based algorithm to predict the orientation of every opinion word near to a feature. Finally, this system produced a graphical summary.

A system called *SSPA* (Sentiment Summarization on Product Aspect) was proposed by Li et al. (2015) [19]. SSPA used bootstrapping dependency patterns to extract features and opinion words, then, the system clustered these features into aspects based on word semantic similarities. Next, it disambiguated sentiment orientations of opinion collocations for each aspect. Finally, the system extracted aspect opinion clauses and analysed their sentiment strengths for each aspect.

Zhou et al. (2016) [31] built a feature-based opinion summarization system for Chinese microblogs called *CMiner*. They extracted opinion targets using an unsupervised label propagation algorithm and built a lexicon-based sentiment classifier to classify a message into positive, negative or neutral class.

Yang et al. (2016) [29] proposed an approach helpful to the prediction by leveraging aspect analysis of reviews. They extracted features using a proposed model that exploited product category information. Then, a two-layer regression model was built to predict helpfulness scores on all aspects and then ensembled them into final helpfulness score.

Cho and Kim (2017) [10] proposed a feature network-driven quadrant mapping to extract the most representative opinions from customer reviews. They found all product features and users' opinion scores for each product reviews using a feature-based summary method as in Hu and Liu (2004). Then, feature relationships within the reviews are discovered from the perspective of co-occurrence and semantic similarity. A paired feature-dictionary feature network was proposed to investigate feature relationship between features that were used to construct a feature network. Finally, a graph was generated by mapping the results of the feature network into a quadrant to summarize customer reviews.

Note that, the contributions of the related works usually focus on the standard sub-tasks, such as features extraction, sentiments identification around these features, polarity detection of these sentiments and summarization styles (Yuan et al., 2015 [30]). However, the effectiveness of summaries also focuses on the good exploitation of reviews to produce an effective summary that describes the real image of the product in the market. For this reason, in this paper, a system which summarizes products customers' reviews has been proposed. This system takes into account the time when reviews were expressed to cover the changes of sentiments over time in order to generate a feature-based summarization, and based on it, a product classification will be performed to assign the product to its corresponding class (good, medium or bad).

3. Problem Formulation

Denote $P = \{p_1, p_2, .., p_n\}$ a set of comparative products (e.g. a cellphone), where $p_i (i = 1$ to $n)$ denotes the i^{th} product. $F = \{f_1, f_2, .., f_m\}$ is the set of product features that can be determined by the consumers according to their preferences concerning the product p_i, where f_j $(j = 1$ to $m)$ denotes the j^{th} product feature. $NumRev = \{NumRev_1, NumRev_2, .., NumRev_n\}$ is a vector of number of online reviews about products, where $NumRev_i$ denotes the number of the online reviews concerning product p_i. For each product $p_i \in P$, $R_i = \{r_{i1}, r_{i2}, .., r_{iNumRev_i}\}$ a set of online reviews and a set $T_i = \{t_{i1}, t_{i2}, .., t_{iNumRev_i}\}$ of times when these reviews are expressed is defined.

After defining the set notations that will be used throughout this paper, the problems are defined as follows:

Problem 1 (feature-based summary) Given the sets of online Reviews R_i and time T_i concerning a product $p_i \in P$ described by a set of features F, our goal is to generate a summary that includes two representative values concerning the positive and negative sentiments, about each feature by taking into account the time (identified from T_i) when they were expressed.

We provide more perspectives of Problem 1 using the reviews *1* and *2* that are expressed about the feature 'battery' of Samsung note 7 cellphone. Note that in *Review 1*, a positive sentiment was expressed about this feature in '02 May 2016'. Contrariwise to *Review 2* which contains a negative sentiment for the same feature in '21 Oct 2016'. Since this product has experienced a problem in the battery, it is obvious that the large number of positive sentiments generated before detecting that problem by the consumer, influences this feature's values. Another example is a product that has experienced improvements in some features will still be influenced by the negative sentiments which are expressed before their improvement. To capture the changing of sentiments concerning a specific feature over time, our proposed method to solve Problem 1 is to take into account the polarity of sentiments which are associated to the feature, and the time when they are expressed, to generate a feature-based-summary vector of product p_i. The proposed method at this stage produces a feature-based-value at each t_{ik} (such that $t_{ik} \in T_i$) for product p_i:It can also show all the changing sentiments about a given feature that helps the manufacturer to have an idea about the quality of this product and compare it with his competitors.

Problem 2 (aggregation of feature-based-summary) Consider the output produced by feature-based-summary step, for each product p_i of P, the main goal here is to summarize the values generated in a single value to take a decision about this product (good, medium or bad product) and compare it with competitors (the other products of the set P). This summarization will be realized via an aggregation of the values generated around the features to get a global score value of polarities (positive and negative) about product p_i. The proposed process at this stage, solves the problem of assigning a decision to products that have score values of polarities close together using a fuzzy logic technique.

4. The Proposed System

This section discusses the design of the proposed system (Fig. 2) that takes as input a set of crawled reviews for a particular product p_1 and competing products (p_2,...,p_n). These competing products are selected by a supervisor who takes into account those belonging to the same familly of p_1. After preprocessing these reviews, the system extracts automatically product features in the first step, then, identified their associated sentiments of them, and generates a polarity (positive, negative) to each one in the second step. Afterwards, we produce a summary for each product based on the features. Finally, we classify each product in its appropriate class (bad, medium or good).

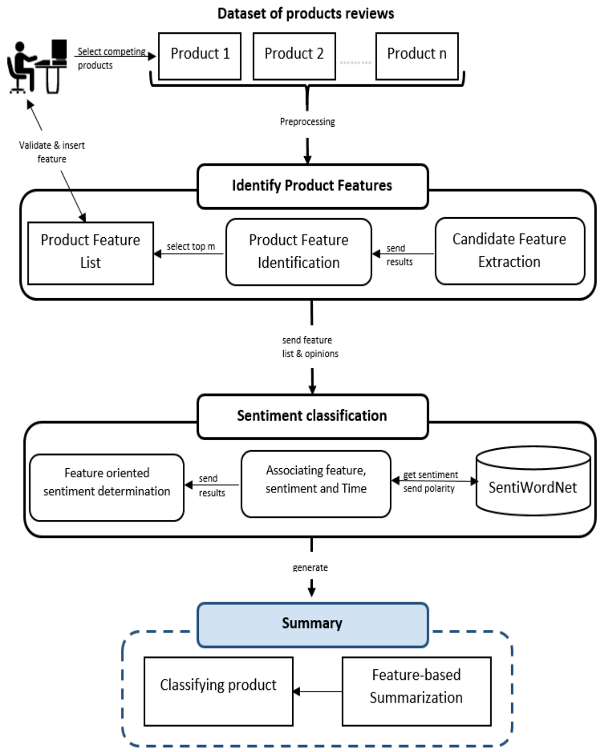


Fig. 2. Proposed summarization system

4.1. Preprocessing

After collecting reviews from different sources, they must be pre-processed. Frequently, these reviews contain several syntactic features that might not be useful, that's why, we will clean them from the data (delete @, username, digits....). After that, the reviews written in English will be selected after detecting the language (Shuyo, 2014 [24]) for each one, because our system performs only on English opinions. Then, a web service[1] will be used to detect and correct the spelling errors in these reviews. Finally, Part-Of-Speech Tagger (POS tagger[2]) will be applied in order to associate each word with its grammatical function. The main goal of using this tagger is to extract nouns and convert plural to singular nouns that are important to increase their weight in order to facilitate the product feature identification step (e.g. batteries to battery in phone reviews), also to produce a useful summary about products.

4.2. Product feature identification

After the preprocessing of reviews, we, then, extract candidate product features from reviews using an algorithm proposed by (Amarouche et al., 2016 [4]) which builds a list that contains nouns composed of one or multiple words extracted from reviews. The main goal of this step is to identify product features from this list. This identification is based on Time Weighting Term Frequency Inverse Document Frequency (TW-TFIDF) method (Amarouche et al., 2016 [4]). This method combines Time Weighting (TW) that is based on the time when opinions are expressed and

Term Frequency Inverse Document Frequency (TF-IDF) that exploits the frequency of candidate features appearing in reviews. Then, we will select the top n of these features in order to generate a list constituting them. Finally, a supervisor verifies the product features list in order to validate them.

4.3. Associating product feature, sentiment and time

To identify feature-sentiment-time triple for each product, reviews are collected from data sources and product features are used as input in this step. Then, we will select features and their associated sentiment terms that appear in reviews. Finally, we will build feature-sentiment-time triple where time is when a review is expressed.

Finding out how sentiments are expressed about these features is an important task in this step, we will analyze dependency relations in reviews using the Stanford CoreNLP[3] that identifies a term's role and dependency relations in a review. We may find many different dependency relations in reviews, but only some of them are helpful to this task. We will consider four major typed dependencies as in Wu et al. study (2009) [27]. It includes *nsubj* (nominal subject), *amod* (adjectival modifier), *rcmod* (relative clause modifier), *dobj* (direct object) in the feature-sentiment-time extraction step. The four dependencies (mentioned above) are used to identify the feature-sentiment pairs and we will add *neg* (negation) to the list of dependencies which is important to check because the meaning will change if a negation is associated to a sentiment.

To associate sentiments to features in sentences that contain just one feature, we have simply to respect these four dependency relations. But in reality, we may find some sentences that contain more than one feature. For instance, the following review contains two features: "*Signal strength will affect the battery life*". After applying the parsing to this sentence, we can identify two important types of dependence *nsubj* (Signal strength, affect) and *dobj* (battery life, affect) which rely on two features with one sentiment. This poses a problem for associating sentiments to features in this case. For this reason, we will use some rules (Chatterji et al., 2017 [8]) to associate sentiment with its corresponding feature in a sentence. Then, we will extract a negation that associates to the sentiment if it exists that is identified with *neg* (negation) type.

Algorithm 1 summarizes the main steps of extracting feature-polarity-time triplet. It takes as input the set of features and reviews about product p_i, then, returns a set of polarity-time pair corresponding to each feature f_j. When each review r_{ik} has been split to a set of sentences and parsed thereafter, we will be extracting sentiments associated to each feature f_j using the rules (Chatterji et al., 2017 [8]). Then, the polarity is generated based on this sentiment and negation if it exists.

Algorithm 1: feature-polarity-time extraction

Input:

$F = \{f_1, \dots, f_j, \dots, f_m\}$ // a set of product features

$RT_i = \{(r_{i1}, t_{i1}), \dots, (r_{ik}, t_{ik}), \dots, (r_{iNumRev_i}, t_{iNumRev_i})\}$ // a set of reviews and time when they are axpressed on the product p_i

Output:

A set of dependencies associated to each feautre $\{rel_1, .., rel_j, .., rel_m\}$ // where $rel_j[f_j \rightarrow \{(polarity, t_{i1}), (polarity, t_{i2}), ...\}\ ]$ is a set of polarity-time associate to feature f_i

for *each* $f_j \in F$ **do**
  for *each* $(r_{ik}, t_{ik}) \in RT_i$ **do**
    split rik into a set of sentence $\{s_1, .., s_p, .., s_q\}$
    for *each* $s_p \in r_{ik}$ **do**
      if $s_i contains f_j$ **then**
        parsing s_p
        extract sentiment with applying rules
        generate polarity of sentiment using SWN
        add (polarity,t_{ik}) into rel_j
  return rel_j

To generate the polarity for each sentiment, a SentiWordNet[4] (SWN) is used to assign sentiment scores for English synsets (Hassan Khan et al., 2016 [16]). Three sentiment scores (positivity, negativity and objectivity) are assigned to each synset in SWN. These scores are assigned by a committee of classifiers (Esuli and Sebastiani, 2006 [12]) for classification of each sysnet that has a range from 0.0 to 1.0 and they always sum up to 1. The following tab. 1 describes SWN structure and sentiment scores associated to their entries.

Headers of the table columns have the following meanings:

- POS: This can take four possible values: a ↦ adjective; v ↦ verb; r ↦ adverb; n ↦ noun.
- Offset: Numerical ID which is associated with part of speech uniquely identifies a synset in the database.
- PosScore: Positive score.
- NegScore: Negative score.
- SysnsetTerms: List of all terms included in the synset.

The polarity classification of the sentiment is based on the difference value between its PosScore and NegScore. So, if this value is greater than zero, it will be classified as positive, whereas if it will be less than zero, it is classified as negative.

4.4. Feature-based summarization

After generating the set of polarity and associated time $\{(polarity, t_{i1}),\ (polarity, t_{i2})\ ,...,\ (polarity, t_{iNumRev_i})\}$ for each product feature f_j (where the couple $polarity - t_{ik}$ is composed of polarity which is positive or negative, and time t_{ik} when this polarity is identified) on a product p_i, the main goal of this step is to generate a percentage for each polarity (positive and negative) around each feature f_j.

Table 1. SentiWordNet fragment

POS	Offset	PosScore	NegScore	Synset-Terms#rank	Gloss
a	01150475	0	0.625	sorry#1 regretful#1 bad#5	feeling or expressing regret or sorrow or a sense of loss over something done or undone
a	00005839	0.5	0.125	living#3	(informal) absolute
n	03931044	0	0	picture#1 image#3 ikon#1 icon#2	a visual representation (of an object or scene or person or abstraction) produced on a surface
v	01824736	0.125	0	wish#2 like#1 care#3	prefer or wish to do something

Formula 1 calculates this percentage for the set of time T_i which depends, at the same time, on the number of polarities and time t_{ik} when they are identified. On the other hand, it produces two values that are: $per_{f_j}^{T_i}(positive)$ and $per_{f_j}^{T_i}(negative)$ which represent percentage values that summarize respectively the positive and negative polarities expressed around feature f_j for the set of time T_i on a product p_i. This percentage introduces a weighting function $\rho_{t_{ik}}$ (formula 2) to give the importance to the polarities that are identified at t_{ik} compared to others that are expressed before this time. To put differently, the latest polarities that are identified around each f_j have a great importance compared to the oldest.

Knowing that the set T_i is in order from the oldest t_{i1} to the latest $t_{iNumRev_i}$ (some elements of T_i are repeated because we can find several reviews that are expressed at the same time), so before calculating this function for each element of T_i, the first step is to quantify each one from it to give it as input to this function. This quantification is based on a chosen step (day, $week$, $month$..) which depends on the $product$ type. Otherwise, the sentiments expressed around the features can change daily for a specific type of products unlike for others that might change weekly or monthly.

An advantage which can be seen from Fig. 3 of this function is the γ value that is affected by the speed convergence of this function. In other words, its speed increases when the value of γ gets closer to 0 therefore it is important to assign good weights to sentiments polarities.

$$per_{f_j}^{T_i}(polarity) = \frac{\sum_{t_{ik}\in T_i} \rho_{t_{ik}} \times n_{t_{ik}}(polarity)}{\sum_{t_{ik}\in T_i} \rho_{t_{ik}} \times N_{t_{ik}}} \quad (1)$$

$$\rho_{t_{ik}} = \gamma^{-t'_{ik}} \quad (2)$$

where:

- t'_{ik}: is the quantified value of the time t_{ik}.
- γ: is a number in the range $0 < \gamma \leq 1$.
- $n_{t_{ik}}$: is the total number of polarities identified around f_j that has same polarity (positive or negative) at t_{ik}.
- $N_{t_{ik}}$: is the total number of polarities identified around f_j at t_{ik}.

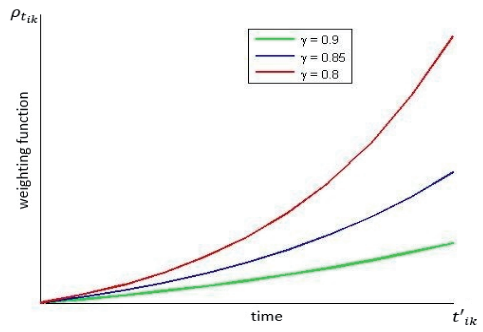


Fig. 3. The influence of γ value on speed convergence of the weighting function $\rho_{t_{ik}}$

Note that, the set of time T_i will be divided into ranges of times t_{ik} so that we can have in each range ($range(l)$), a set of t_{ik} (Fig. 4). The set of t_{ik} can be defined based on a step duration. For example, if we choose the $month$ as a step, automatically each t_{ik} that belongs to the same month will be grouped into the same range.

However, to show the impact of time in the variation results, we must calculate the percentage from the start to each range, For example, to calculate the percentage in the range l, we need to use in the calculation process all t_{ik} of the previous ranges including the range l as a new defined set mentioned as $T_i^{(l)}$ (Fig. 4).

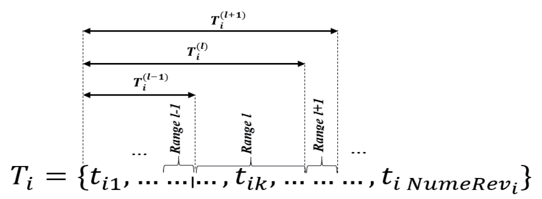


Fig. 4. The division of the set into ranges

4.5. Classifying product

After obtaining the positive and negative percentage scores for each product feature f_j on a product

p_i over time (for each set of time $T_i^{(l)}$) as demonstrated in the above section, the main goal of this step is to take a decision that the product p_i is *bad*, *medium* or *good*. To put differently, we assign the product to its appropriate class (good, medium or bad) for each set of time $T_i^{(l)}$ to show its variation over time and compare it with competitors (the other products of the set P).

For this reason, the first step is to aggregate the percentage scores of the features selected by a supervisor of the system from the set of feature F using formula 3. In the normal case, we give the same weight to all product features ($w_j = 1/m$) to calculate this aggregation. Also, the proposed system allows the supervisor to modify these weights in order to give more priority to some features compared to others. Then, we calculate the difference between the positive and negative values of this aggregation ($aggr_{p_i}^{T_i^{(l)}}$ $(positive)$ and $aggr_{p_i}^{T_i^{(l)}}$ $(negative)$) for each set of time $T_i^{(l)}$ using formula 4 to classify the product p_i (good, medium or bad). let us explain this with an example. When calculating this aggregation for both positive and negative polarities. If $aggr_{p_i}^{T_i^{(l)}}$ $(positive) = 0.9$ and $aggr_{p_i}^{T_i^{(l)}}$ $(negative) = 0.1$, the product p_i can be classified in *good* product class for the set of time $T_i^{(l)}$).

$$aggr_{p_i}^{T_i^{(l)}}(polarity) = \sum_{j=1}^{m} w_j \times per_{f_j}^{T_i^{(l)}}(polarity) \quad (3)$$

$$diff_{aggr_{p_i}}^{T_i^{(l)}} = aggr_{p_i}^{T_i^{(l)}}(positive) - aggr_{p_i}^{T_i^{(l)}}(negative) \quad (4)$$

However, a problem will arise when the values of $aggr_{p_i}^{T_i^{(l)}}$ $(positive)$ and $aggr_{p_i}^{T_i^{(l)}}$ $(negative)$ are close together,in this case, it is not always right to assign the product to the medium product class (e.g. the product p_i was bad for the set of time $T_i^{(l-1)}$). To put differently, the difference between the aggregation of the same polarity for both sets $T_i^{(l)}$ and $T_i^{(l-1)}$ (formula 5) is important to classify each product that have this problem. So, we need a system that takes into account these parameters as input, consequently a 'bad', 'medium' or 'good' product will be generated as output.

$$diff_{aggr_{p_i}}^{T_i^{(l)}\&T_i^{(l-1)}}(polarity) = aggr_{p_i}^{T_i^{(l)}}(polarity) - aggr_{p_i}^{T_i^{(l-1)}}(polarity) \quad (5)$$

In reality, we generally do not use crisp numeric values to assign a product to its appropriate class for the set of time $T_i^{(l)}$, but we use linguistic terms like good or bad. Therefore, we convert the numeric values which are $diff_{aggr_{p_i}}^{T_i^{(l)}}$, and $diff_{aggr_{p_i}}^{T_i^{(l)}\&T_i^{(l-1)}}$ $(positive)$ (or $diff_{aggr_{p_i}}^{T_i^{(l)}\&T_i^{(l-1)}}$ $(negative)$)to linguistic terms and use them in this classification.

The proposed process employs fuzzy logic to do these conversions and calculate a comprehensive classifying product score for every product p_i. This process illustrated in Fig. 5 consists of several parts that will be explained in the following subsections:

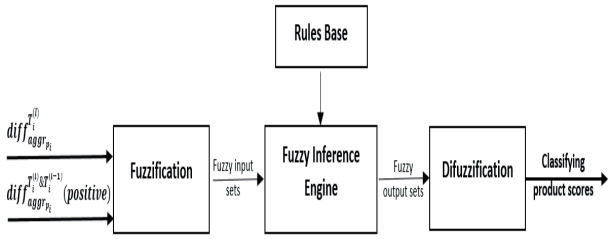


Fig. 5. Fuzzy logic process used for calculating product scores

Fuzzification In a fuzzy inference system, we have a set of crisp input variables which should be processed that are $diff_{aggr_{p_i}}^{T_i^{(l)}}$, and $diff_{aggr_{p_i}}^{T_i^{(l)}\&T_i^{(l-1)}}$ $(positive)$ in our system. Corresponding to each input value, we usually define a linguistic variable with the same name as input variables. Each linguistic variable, that is represented by a function called membership function (denoted by μ), contains a set of linguistic values corresponding to each input variable. The output of a membership function is a real number in the range [0, 1] which reflects the level of membership of an input variable to a linguistic variable. The fuzzifier uses these membership functions to convert crisp input variables to fuzzy linguistic variables. In the proposed process, we use a trapezoidal fuzzy set that demonstrates its effectiveness in many works in the literature like (Boltzheim et al., 2001 [7]; Daneshvar, 2011 [11]).

In our process, two variables have been used as input which are $diff_{aggr_{p_i}}^{T_i^{(l)}}$, and $diff_{aggr_{p_i}}^{T_i^{(l)}\&T_i^{(l-1)}}$ $(positive)$, and one output variable which is classifying product that should be calculated for each set of time $T_i^{(l)}$. The membership functions for both input variables (normalized between 0 and 1) is depicted in Fig. 6 and 7, and the membership function of classifying product as the output function is defined in Fig. 8.

Inference engine The main goal of the inference engine is to convert fuzzy inputs to fuzzy output. This conversion is done using a set of IF-THEN type rules called fuzzy rules. A fuzzy rule specifies the condition in which a set of fuzzy inputs can be mapped to an output fuzzy variable. Since we have 2 input variables and each of which can have 3 different values. Each combination can potentially represent a particular level of product ranking.

In order to arrive to classifying a product, an inference engine should evaluate all fuzzy rules and then

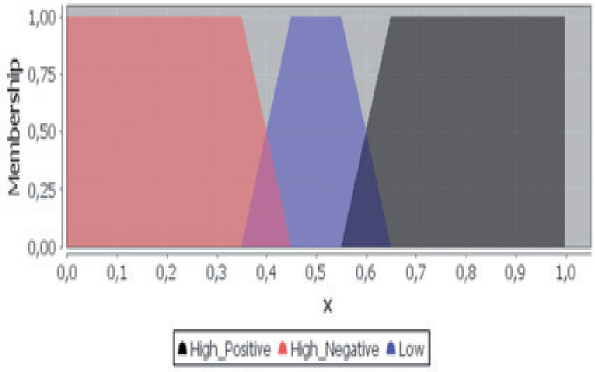


Fig. 6. $diff_{aggr_{p_i}}^{T_i^{(l)}}$ ***membership function***

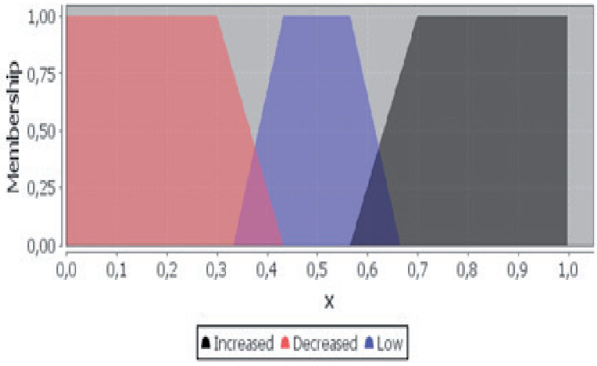


Fig. 7. $diff_{aggr_{p_i}}^{T_i^{(l)} \& T_i^{(l-1)}}(positive)$ ***membership function***

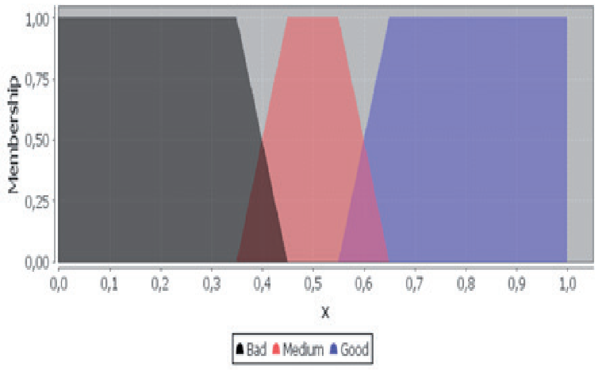


Fig. 8. Classifying product output membership function

compose the results of these evaluations to define output zones for the output variable. An example of some rules used in this step:

- **RULE 1: IF** $diff_{aggr_{p_i}}^{T_i^{(l)}}$ **IS Low AND** $diff_{aggr_{p_i}}^{T_i^{(l)} \& T_i^{(l-1)}}(positive)$ **is Decreased THEN Product IS Bad;**
- **RULE 2: IF** $diff_{aggr_{p_i}}^{T_i^{(l)}}$ **IS Low AND** $diff_{aggr_{p_i}}^{T_i^{(l)} \& T_i^{(l-1)}}(positive)$ **is Increased THEN Product IS Good;**
- **RULE 3: IF** $diff_{aggr_{p_i}}^{T_i^{(l)}}$ **IS Low AND** $diff_{aggr_{p_i}}^{T_i^{(l)} \& T_i^{(l-1)}}(positive)$ **is Low THEN Product IS Medium;**

Defuzzification Based on different values of input fuzzy variables, several fuzzy rules might be activated in the same time and the result is a set of linguistic output values which are included in output with different levels of membership. In this step, we employ the Center-of-Gravity (CoG) method that is the most popular method and is also quick and accurate in computations (Leekwijck et al., 1999 [18]). The final value of classifying product is calculated using this COG approach that produces a real number between 0 and 1. The values close to 1 represent a good product based on our designed model. Low values for classifying product show that it is a bad product.

5. Experiment

This section evaluates the proposed system which generates summaries from a data set that contains reviews about the same type of products. The procedure for our implementation is as follows: After crawling reviews from the source, they must be pre-processed by selecting only the reviews written in English, then, the system detects and corrects the spelling errors. After that, POS tagger tags all the words in each review to their appropriate part of speech tag in order to identify word nouns that are important to the next step. After that, a list containing candidate features is being built, then, potential product features will be identified from this list using TW-TFIDF (Amarouche et al., 2016 [4]). Lastly, in this step, a list of product features is built which is validated by a supervisor. Next step is, the sentiment classification that contains two substeps: 1) associate product features with their corresponding sentiments and time when each one is expressed, 2) determinate the polarity for each sentiment using sentiWordNet. Finally, our system produces a summary based on the features for each product p_i, then, assign each one to its corresponding class (bad, medium or good product) over time.

5.1. Data sets description

The proposed system is experimented with user reviews on two data sets. The first is about four cellphone products that were collected from gsmarena[5]. The second data set contains four hotels reviews that were crawled from Tripadvisor[6]. The details of these data sets are shown in Tab. 2 and 3.

5.2. Experiments on feature-based opinion summaries

In the literature, all proposed works ignore the evolution of feature-based summary over time, however, to compare our method with the others we don't have a direct comparison criterion. So, to show the performance of the proposed method, we need to show the impact of taking into account time axis when reviews are expressed compared with the standards methods.

Fig. 9 (a and b) shows the experimental results of feature-based summaries using the proposed approach on the four cellphones. For each cellphone, five

Table 2. Data description of cellphone reviews

Product (P)	Product Description	Number of reviews	Interval of extraction
p_1	Sony Xperia XA Ultra	1210	17 May 2016 to 4 Jan 2017
p_2	OnePlus 3	1681	25 Mar 2016 to 5 Jan 2017
p_3	Samsung Galaxy Note7	1679	2 May 2016 to 15 Dec 2016
p_4	Apple iPhone 7	904	26 Apr 2016 to 5 Jan 2017

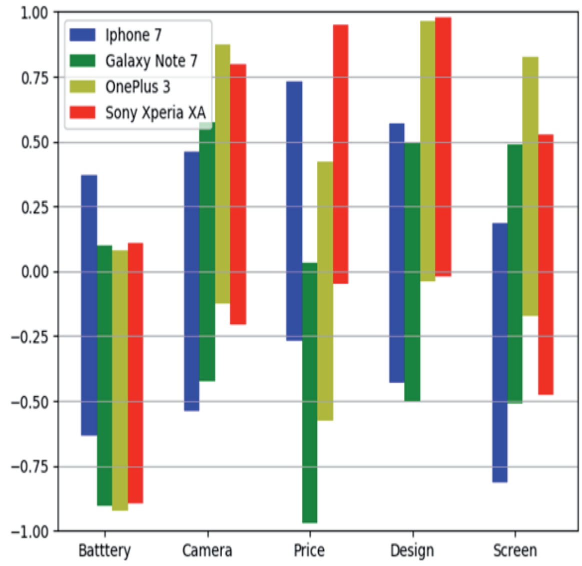
Table 3. Data description of hotel reviews

Hotel (H)	Location	Number of reviews	Interval of extraction
h_1	San Francisco, California	324	21 Nov 2006 to 7 Jan 2009
h_2	Barcelona, Catalonia	244	4 Nov 2004 to 5 Jan 2009
h_3	Hong Kong	249	5 Jun 2005 to 7 Jan 2009
h_4	New York	263	16 Feb 2006 to 6 Jan 2009

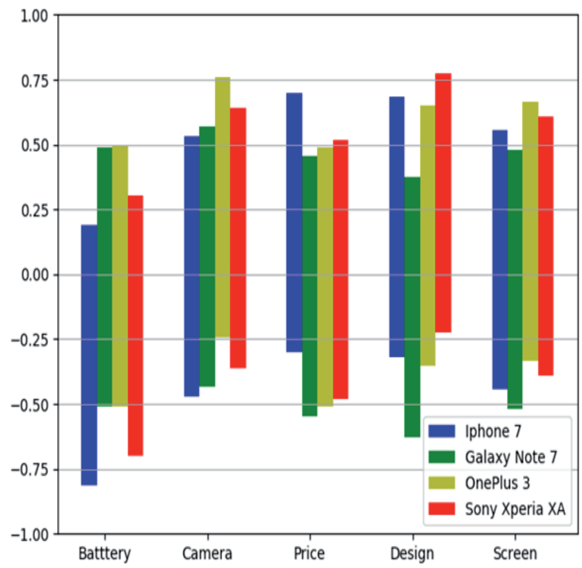
product features are used as important fields to describe the products (such as $\gamma = 0.5$).

The comparison between the proposed method in Fig. 9 (a and b) shows that the results generated by the one where the chosen step is 1 week gives the real image of product features in the reviews compared to the other (Fig. 9 (b)) that exploits the step (*1 month*). To illustrate this, we take the example of galaxy note 7 cellphone which is produced by Samsung that encounters a big problem in the feature 'battery'. So, this problem was easy detected in the proposed approach where the step chosen is *1 week* ($per_{battery}(postive) = 0.0987..$ and $per_{battery}(negative) = 0.9012..$), contrariwise, it is not detected by the proposed method using the step *1 month* ($per_{battery}(postive) = 0.4909..$ and $per_{battery}(negative) = 0.5090..$), which makes that the choice of the step (*day*, *week*, *month* ..) is very important to produce an effective summary depending on the domain.

Using the same steps of the proposed method to generate feature-based summaries and without implementing the times where reviews are expressed, Fig. 10 shows the results obtained about the same five product features. The way to generate this summary showed in Fig. 10 is like the approaches used in several works such as (Hu and Liu, 2004 [13]), (Liu et al., 2005 [20]) and (Abulaish et al., 2009 [2]) after its normalization between -1 and 1 in order to compare it with the proposed method. From Fig. 10, we can find that the results generated by the standard method



(a) step chosen is *1 week*



(b) step chosen is *1 month*

Fig. 9. Bar chart showing the proposed feature-based summaries corresponds to the cellphone corpus

about feature 'battery' of Samsung galaxy note 7 is still a good feature (the positive score is *0.6408..* and negative score is *0.3591..*) even if the product has experienced a problem in this feature. However, the large number of positive sentiments about it which were shared at the beginning influence on the results generated by the standard approach after the detection of the problem by consumers about this feature.

To demonstrate the effectiveness of the proposed approach to detect this problem in the feature 'battery' of Samsung galaxy note 7, Fig. 11 shows a comparison of positive scores between the proposed approach

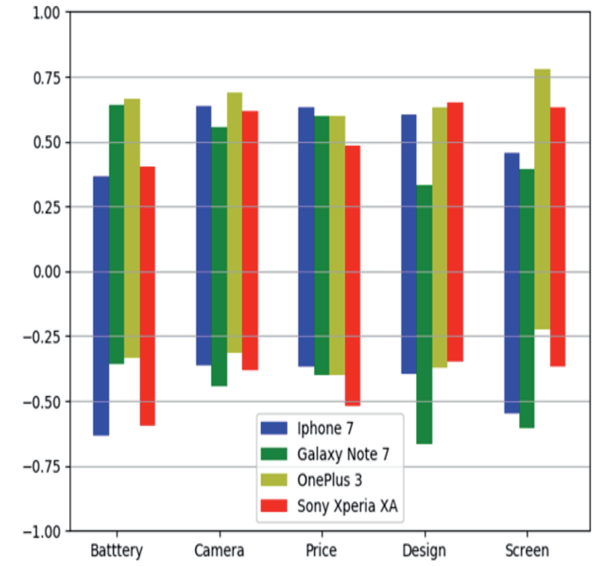


Fig. 10. Bar chart showing the feature-based summaries approach without integrating time when reviews are expressed, corresponds to the cellphone corpus

and standard approach over time. By choosing *week* as a step in the weighting function of the proposed approach, we can detect the problem of feature 'battery' effectively (after October 2016) due to this function which gives the importance to the reviews week after the other. On the other hand, the curve of the proposed method such as the chosen step is *1 month* is close to the standard method which always makes the choice of a step as an important role for the product domain where we are operating.

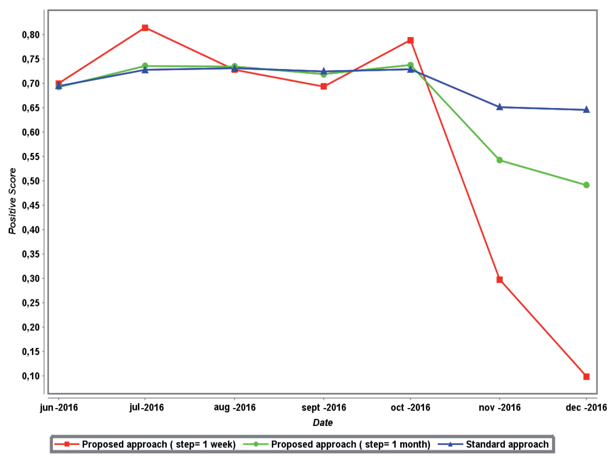


Fig. 11. A comparison between the variation of positive score of the proposed and standard methods about feature 'battery' of 'Samsung galaxy note 7' over time

Fig. 12 shows an example of feature-based summaries about five features of hotels data set using the proposed approach (such as *step=1 month* and $\gamma = 0.5$), which shows that the proposed system is domain independent. By comparing the experimental results in Fig. 12 and 13, we find that the standard summary that does not exploit times axis produces a summary close to reality just for features that do not experience changes in sentiments over time. On the other hand, the other features which have experienced changes after detecting a problem or improvement, it is difficult for this type of summaries to produce one which reflects the real state of these features after changing the sentiments. However, the proposed approach generates a summary that takes into account the change of sentiments about features over time if it exists.

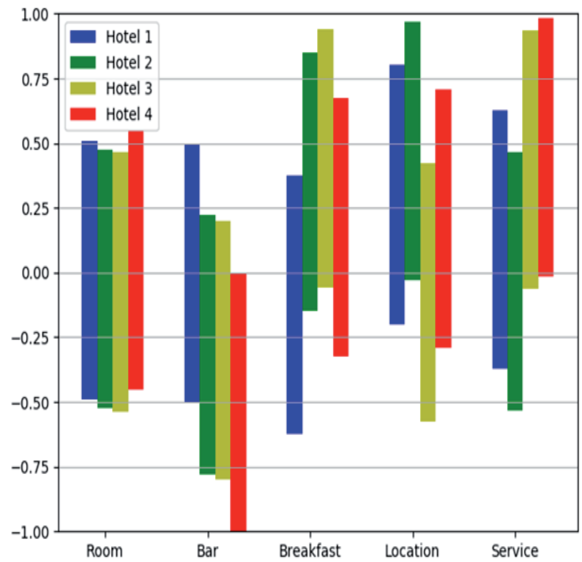


Fig. 12. Bar chart showing the proposed feature-based summaries corresponds to the hotel corpus

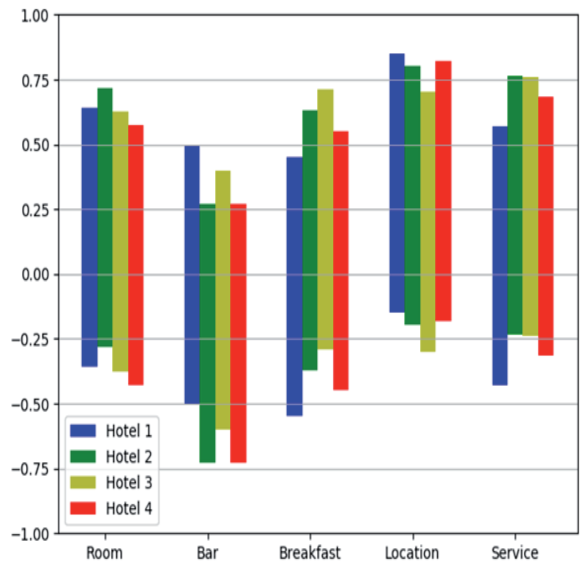


Fig. 13. Bar chart showing the feature-based summaries approach without integrating time when reviews are expressed, corresponds to the hotel corpu

By comparing this variation of the feature 'service' which has experienced a change over time using both approaches, we can notice, the proposed one takes into consideration this change to produce its positive score ((1) an increase between *June 2007* and *January 2008* and between *June 2008* and *January 2009*; and (2) a decrease between *January* and *June 2008*). Contrariwise, this change is not taken into account in the standard approach. For the feature 'room' that has an output (negative or positive score) close together using both methods as indicated in Fig. 12 and 13, we find that the proposed method detects a change of sentiments in its time interval (Fig. 14) that focuses on an increase of positive sentiment between *June 2006* and *January 2008*, by cons in the first six months of *2008* and the beginning of the year *2009*, this proposed method knows an increase of negative scores that the standard approach do not do it. This means that, even if the output of both methods is close, the proposed method detects all the changes over time by exploiting this time axis to produce feature-based summary. However, the sentiments expressed about the features always have an influence on the standard approach output with the same way which does not allow detecting a change of feelings if exist.

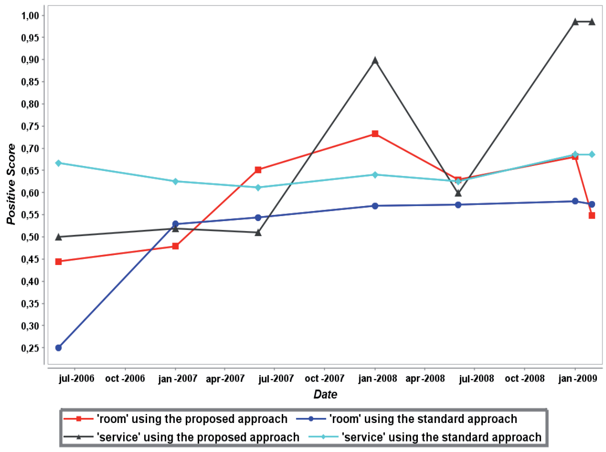


Fig. 14. A comparison between the variation of positive score of the proposed and standard method about two features 'room' and 'service' of hotel 4 reviews over time

5.3. Experiments on classifying product

Fig. 15 and 16 show the variation of classifying cellphones and hotels data sets over time using the fuzzy logic system. This variation is to know the position of these data sets (good, medium or bad) over time. For example, Eig 15 shows that, the position of product 'note 7' which faces a problem in its feature 'battery' will be positioned in the 'bad' and 'medium' class over time. Based on its variation, we can take a decision that it is a 'bad' product. Also, the score produced by the system allows to compare a specific product with its competitors that is an important step in CI.

Fig. 16, shows the variation of each hotel position over time that allows to know the performance of each one based on several competitor attributes which are aggregated including *room*, *service*, *breakfast*, *bar* and *location*. According to the findings of Mohammed et al. (2014) [22], several competitor attributes are considered by hotel managers, including proximity of location, room rate similarity, product and service offerings, service quality delivery, and sales channels. For this reason, our system gives to the supervisor the permission to modify the list of these attributes in order to classify the hotel according to its vision of the attributes that influence the performance of the hotel.

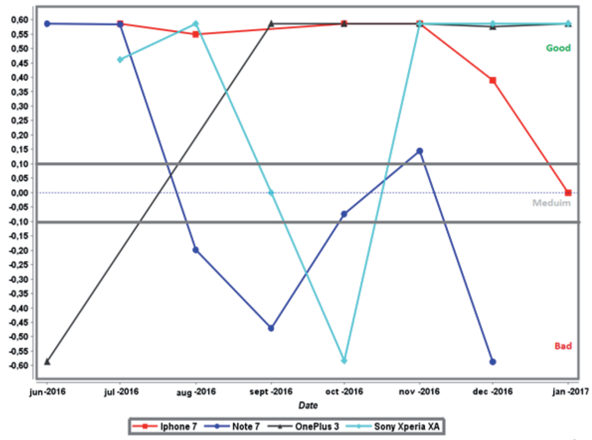


Fig. 15. The variation of classifying cellphone over time

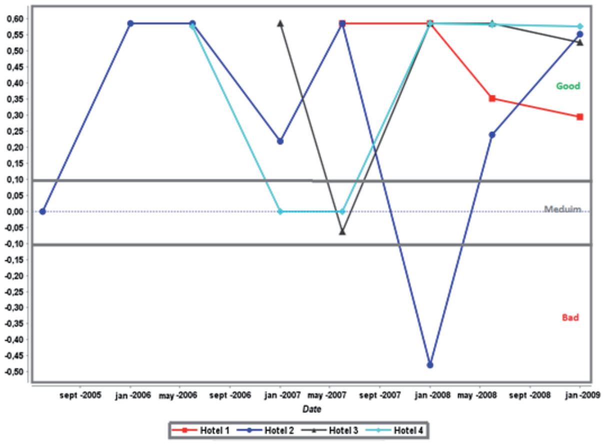


Fig. 16. The variation of classifying hotels over time

6. Conclusion

In this study, we proposed a product opinion summarization framework, comprising two main components: feature-based opinion summary, and product classification. These components have an important role to product manufacturers according to their need like improving their product or launching a new one. After extracting product features in a list and select the *top m* from it that is validated by a supervisor, the proposed feature-based summary approach produces a summary based on the times where the reviews are expressed. Existing related methods, such as (Hu and Liu, 2004 [13]), (Liu et al., 2005 [20]) and (Abulaish et al., 2009 [2]), generate summaries without introducing the time axis in their methods. However, the experimental part demonstrates the robustness of the proposed approach to take into account the change of sentiments about features over time if it exists. For the product classification, we employ the fuzzy logic technique to rank each product to its appropriate class (bad, medium or good). Then, we compare the products with their competitors based on this classification.

For future work, the stage of selecting the competing products in the proposed system that belongs to the same range of a particular product is performed by a supervisor (manually). For this reason, a study will be conducted to detect these competing products automatically. On the other hand, we have seen that our data sets contain an important number of sentences that have no opinions. These sentences need to be filtered since they introduce noise to the proposed system process.

Notes

[1] http://wsf.cdyne.com/SpellChecker/check.asmx
[2] http://nlp.stanford.edu/software/tagger.shtml
[3] http://nlp.stanford.edu/software/corenlp.shtml
[4] http://sentiwordnet.isti.cnr.it/
[5] www.gsmarena.com
[6] www.tripadvisor.com

AUTHORS

Kamal Amarouche* – ALBIRONI Research Team, ENSIAS, Mohammed 5 University, Rabat, Morocco, e-mail: amarouchekamal@gmail.com.
Houda Benbrahim – ALBIRONI Research Team, ENSIAS, Mohammed 5 University, Rabat, Morocco, e-mail: houda.benbrahim@um5.ac.ma.
Ismail Kassou – ALBIRONI Research Team, ENSIAS, Mohammed 5 University, Rabat, Morocco, e-mail: ismail.kassou@um5.ac.ma.

*Corresponding author

REFERENCES

[1] L. Abd-Elhamid, D. Elzanfaly, and A. S. Eldin, "Feature-based sentiment analysis in online Arabic reviews". In: *2016 11th International Conference on Computer Engineering Systems (ICCES)*, 2016, 260–265, 10.1109/ICCES.2016.7822011.

[2] M. Abulaish, Jahiruddin, M. N. Doja, and T. Ahmad, "Feature and Opinion Mining for Customer Review Summarization". In: S. Chaudhury, S. Mitra, C. A. Murthy, P. S. Sastry, and S. K. Pal, eds., *Pattern Recognition and Machine Intelligence*, 2009, 219–224.

[3] K. Amarouche, H. Benbrahim, and I. Kassou, "Product Opinion Mining for Competitive Intelligence", *Procedia Computer Science*, vol. 73, 2015, 358–365, 10.1016/j.procs.2015.12.004.

[4] K. Amarouche, H. Benbrahim, and I. Kassou, "Product Features Extraction From Opinions According To Time", 2016, 10.5281/zenodo.1125349.

[5] E. Asgarian and M. Kahani, "Designing an Integrated Semantic Framework for Structured Opinion Summarization". In: V. Presutti, C. d'Amato, F. Gandon, M. d'Aquin, S. Staab, and A. Tordai, eds., *The Semantic Web: Trends and Challenges*, 2014, 885–894.

[6] K. Bafna and D. Toshniwal, "Feature based Summarization of Customers' Reviews of Online Products", *Procedia Computer Science*, vol. 22, 2013, 142–151, 10.1016/j.procs.2013.09.090.

[7] P. Boltzheim, B. Hamori, and L. T. Kóczy, "Optimization of trapezoidal membership functions in a fuzzy rule system by the bacterial algorithm approach", *Budapest University of Telecommunications and Economics, Department of Telecommunications and Telematics, Budapest*, 2001.

[8] S. Chatterji, N. Varshney, and R. K. Rahul, "AspectFrameNet: a frameNet extension for analysis of sentiments around product aspects", *The Journal of Supercomputing*, vol. 73, no. 3, 2017, 961–972, 10.1007/s11227-016-1808-6.

[9] J. Chen, D. P. Huang, S. Hu, Y. Liu, Y. Cai, and H. Min, "An opinion mining framework for Cantonese reviews", *Journal of Ambient Intelligence and Humanized Computing*, vol. 6, no. 5, 2015, 541–547, 10.1007/s12652-014-0237-8.

[10] S. G. Cho and S. B. Kim, "Feature network-driven quadrant mapping for summarizing customer reviews", *Journal of Systems Science and Systems Engineering*, vol. 26, no. 5, 2017, 646–664, 10.1007/s11518-017-5329-5.

[11] M. Daneshvar, "Programmable Trapezoidal and Gaussian Membership Function Generator", 2012.

[12] A. Esuli and F. Sebastiani, "Determining Term Subjectivity and Term Orientation for Opinion Mining", 2006.

[13] M. Hu and B. Liu, "Mining and Summarizing Customer Reviews". In: *Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, New York, NY, USA, 2004, 168–177, 10.1145/1014052.1014073, event-place: Seattle, WA, USA.

[14] A. Kangale, S. K. Kumar, M. A. Naeem, M. Williams, and M. K. Tiwari, "Mining consumer reviews to generate ratings of different product attributes while producing feature-based review-summary", *International Journal of Systems Science*, vol. 47, no. 13, 2016, 3272–3286, 10.1080/00207721.2015.1116640.

[15] H. Kansal and D. Toshniwal, "Aspect based Summarization of Context Dependent Opinion Words", *Procedia Computer Science*, vol. 35, 2014, 166–175, 10.1016/j.procs.2014.08.096.

[16] F. H. Khan, U. Qamar, and S. Bashir, "SentiMI: Introducing point-wise mutual information with SentiWordNet to improve sentiment polarity detection", *Applied Soft Computing*, vol. 39, 2016, 140–153, 10.1016/j.asoc.2015.11.016.

[17] P. P. Ładyżyński and P. Grzegorzewski, "A Recommender System Based on Customer Reviews Mi-

ning". In: L. Rutkowski, M. Korytkowski, R. Scherer, R. Tadeusiewicz, L. A. Zadeh, and J. M. Zurada, eds., *Artificial Intelligence and Soft Computing*, 2014, 512–523.

[18] W. V. Leekwijck and E. E. Kerre, "Defuzzification: criteria and classification", *Fuzzy Sets and Systems*, vol. 108, no. 2, 1999, 159–178, 10.1016/S0165-0114(97)00337-0.

[19] Y. Li, Z. Qin, W. Xu, and J. Guo, "A holistic model of mining product aspects and associated sentiments from online reviews", *Multimedia Tools and Applications*, vol. 74, no. 23, 2015, 10177–10194, 10.1007/s11042-014-2158-0.

[20] B. Liu, M. Hu, and J. Cheng, "Opinion Observer: Analyzing and Comparing Opinions on the Web". In: *Proceedings of the 14th International Conference on World Wide Web*, New York, NY, USA, 2005, 342–351, 10.1145/1060745.1060797, event-place: Chiba, Japan.

[21] A. Maisto and S. Pelosi, "Feature-Based Customer Review Summarization". In: R. Meersman, H. Panetto, A. Mishra, R. Valencia-García, A. L. Soares, I. Ciuciu, F. Ferri, G. Weichhart, T. Moser, M. Bezzi, and H. Chan, eds., *On the Move to Meaningful Internet Systems: OTM 2014 Workshops*, 2014, 299–308.

[22] I. Mohammed, B. D. Guillet, and R. Law, "Competitor set identification in the hotel industry: A case study of a full-service hotel in Hong Kong", *International Journal of Hospitality Management*, vol. 39, 2014, 29–40, 10.1016/j.ijhm.2014.02.002.

[23] H. Rahimi and H. E. Bakkali, "CIOSOS: Combined Idiomatic-Ontology Based Sentiment Orientation System for Trust Reputation in E-commerce". In: Á. Herrero, B. Baruque, J. Sedano, H. Quintián, and E. Corchado, eds., *International Joint Conference*, 2015, 189–200.

[24] N. Shuyo. "Language Detection with Infinity-gram. Contribute to shuyo/ldig development by creating an account on GitHub", April 2019. original-date: 2011-11-18T01:18:37Z.

[25] Ľ. Štefániková and G. Masárová, "The Need of Complex Competitive Intelligence", *Procedia - Social and Behavioral Sciences*, vol. 110, 2014, 669–677, 10.1016/j.sbspro.2013.12.911.

[26] D. Wang, S. Zhu, and T. Li, "SumView: A Web-based engine for summarizing product reviews and customer opinions", *Expert Systems with Applications*, vol. 40, no. 1, 2013, 27–33, 10.1016/j.eswa.2012.05.070.

[27] Y. Wu, Q. Zhang, X. Huang, and L. Wu, "Phrase Dependency Parsing for Opinion Mining". In: *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 3 - Volume 3*, Stroudsburg, PA, USA, 2009, 1533–1541, event-place: Singapore.

[28] K. Xu, S. S. Liao, J. Li, and Y. Song, "Mining comparative opinions from customer reviews for Competitive Intelligence", *Decision Support Systems*, vol. 50, no. 4, 2011, 743–754, 10.1016/j.dss.2010.08.021.

[29] Y. Yang, C. Chen, and F. S. Bao, "Aspect-Based Helpfulness Prediction for Online Product Reviews". In: *2016 IEEE 28th International Conference on Tools with Artificial Intelligence (ICTAI)*, 2016, 836–843, 10.1109/ICTAI.2016.0130.

[30] X. Yuan, N. Sa, G. Begany, and H. Yang, "What Users Prefer and Why: A User Study on Effective Presentation Styles of Opinion Summarization". In: J. Abascal, S. Barbosa, M. Fetter, T. Gross, P. Palanque, and M. Winckler, eds., *Human-Computer Interaction – INTERACT 2015*, 2015, 249–264.

[31] X. Zhou, X. Wan, and J. Xiao, "CMiner: Opinion Extraction and Summarization for Chinese Microblogs", *IEEE Transactions on Knowledge and Data Engineering*, vol. 28, no. 7, 2016, 1650–1663, 10.1109/TKDE.2016.2541148.

[32] L. Zhuang, F. Jing, and X.-Y. Zhu, "Movie Review Mining and Summarization". In: *Proceedings of the 15th ACM International Conference on Information and Knowledge Management*, New York, NY, USA, 2006, 43–50, 10.1145/1183614.1183625, event-place: Arlington, Virginia, USA.