Information Sciences 561 (2021) 243–273

Contents lists available at ScienceDirect

Information Sciences

journal homepage: www.elsevier.com/locate/ins

An approach for a decision-making support system based on measuring the user satisfaction level on Twitter

Huyen Trang Phan [a], Ngoc Thanh Nguyen [b,*], Van Cuong Tran [c], Dosam Hwang [a,*]

[a] Department of Computer Engineering, Yeungnam University, Gyeongsan, Republic of Korea
[b] Department of Applied Informatics, Wroclaw University of Science and Technology, Wroclaw, Poland
[c] Faculty of Engineering and Information Technology, Quang Binh University, Dong Hoi, Viet Nam

ARTICLE INFO

Article history:
Received 30 June 2020
Received in revised form 28 December 2020
Accepted 4 January 2021
Available online 26 January 2021

Keywords:
Sentiment analysis
Decision-making
Fuzzy decision tree
User satisfaction level
Fuzzy sentiment phrase
BiLSTM-CRF model

ABSTRACT

Social networks are a very popular channel for people to communicate with, to find, to reference other users before making decisions, especially those concerning purchase. How can users' opinions within social networks be used in making decisions cost-effective and reliable? In this paper, we propose an approach for supporting decision-making based on measuring the user satisfaction level by analyzing the sentiment of aspects and mining the fuzzy decision trees. Our proposal has been proved to overcome some of the disadvantages of previous methods. Specifically, we consider the fuzzy sentiments of users for aspects and the effects of user satisfaction, dissatisfaction, and hesitation for decision-making. The proposed method comprises four main stages. The first stage identifies a topic, which the user is interested. In the second stage, aspects of the topic and their sentiments within tweets are extracted. At the third stage, the user satisfaction level is calculated according to each kind of sentiment identified in the second step. Finally, a decision matrix is constructed, and the fuzzy decision tree is built to generate a set of rules for supporting users in decision-making. The experiments using tweets show that the proposed method achieves promising results regarding the accuracy and gained information.

1. Introduction

Social networks are a popular channel for exchanging views, sharing knowledge, chatting, learning, and consulting among people in daily life. Additionally, social networks are presumed to influence the information technology industry, e-governance, personal information management, business, healthcare, individual privacy rights, etc. [12]. One such popular social network platform is Twitter. The amount of data on Twitter is increasing exponentially; according to statistics, on April 17, 2019,[1] an average of 6,000 tweets were posted on Twitter every second. Large and larger number of users post their comments regarding various topics on Twitter. Many tools, especially those supporting decision-making, now are using the sentiments in these comments as an essential source of data for different applications in life [15]. However, they still are not convincing for the users because of some significant disadvantages. On the other hand, it has been proved that sentiments considerably influence the decision-making process of people. Therefore, users need an effective tool that can assist them in

* Corresponding authors.
E-mail addresses: huyentrangtin@gmail.com (H.T. Phan), Ngoc-Thanh.Nguyen@pwr.edu.pl (N.T. Nguyen), vancuongqbuni@gmail.com (V.C. Tran), dosamhwang@gmail.com, dshwang@yu.ac.kr (D. Hwang).

[1] https://zephoria.com/twitter-statistics-top-ten/.

https://doi.org/10.1016/j.ins.2021.01.008
0020-0255/© 2021 The Author(s). Published by Elsevier Inc.

decision-making using the sentiments within tweets. Owing to it, the user's sentiments will be used more effectually in making a decision.

Furthermore, we can observe that in recent years e-trade on Twitter has been developed, since this social network enables users to easily publish and share their opinions about products or events with other potential consumers who can read, evaluate, and learn from these experiences when buying similar items. Thus, decision-making is made more rational. Users want to make a sensible decision; therefore, they need a support system, or more concretely, a decision-making support system. Decision making is a process of selecting, synthesizing, and evaluating various opinions or alternatives to give one best selection that fits user requirements [9]. Decision support systems are built using computer–human interaction tools to help users identify, solve problems, and make reasonable decisions by using available data and models [11]. Users often need to be supported in decision-making regarding the following two specific cases:

- In the first case, the users have seen the product/service or tried this product/service directly. They have reviewed the aspects of this product/service, but they cannot ascertain whether they should buy the product or use the service based only on these reviews.
- In the second case, the users have not seen the product/service directly. They also do not know the aspects of the product/service. They provide the aspect requirements that they are interested in and, consequently, refer to the opinions of users who have used this product in order to make a purchasing decision.

It often happens that people need to get advice from others to make sensible decisions in many real-life situations. However, it is not easy to be realized. Therefore, users need an automatic tool to support them in decision-making that satisfies their requirements.

Sentiment analysis is a specific case of text analysis that needs to solve such problems as detection, extraction, and opinion-oriented text analysis. It aims to identify positive and negative opinions; it measures how positively or negatively an object is regarded [4]. Entities included in sentiments can refer to people, organizations, events, locations, products, or topics. Sentiment analysis can be implemented in multiple different approaches dependently on the user's goal, specifically, either according to the number of sentiment types—such as binary (positive or negative) or ternary (positive, negative, and neutral) classification [38]; or based on the context of reviews—such as favorable/unfavorable, thumbs up/thumbs down [41] or likely to win/unlikely to win [24]). As Liu has shown in [25], the user's sentiment can be analyzed based on lexicon, machine learning, or hybrid methods (i.e., a combination of lexicon and machine learning). Sentiment analysis is applied at the following levels: document, sentence, and aspect. Document-level sentiment analysis extracts sentiments toward an entity from the entire document. Sentence-level sentiment analysis classifies the sentiment in each sentence of the text as positive, negative, or neutral. *Aspect-based sentiment analysis* determines the sentiment of users regarding a given aspect of an entity in a sentence [25,50]. Such sentiment analysis has widespread application, such as in prediction of the market price, box office, and political elections, in the estimation of movie sales, and in recommendation systems [50]. Some previous researches, among others described in [33,37,46], have used the aspects-level sentiment analysis to identify user satisfaction. In this paper, we determine the satisfaction, dissatisfaction, and hesitation of users based on aspects-level sentiment analysis. Below we present some motivation of our work.

First, we should explain what *sentiments of aspects* mean. An aspect represents a small part or a specific feature or an attribute of an entity that has been mentioned in user opinions (e.g., a product or service) [2,44,49]. By a sentiment of aspects we understand the feeling, attitude, evaluation, or emotion of users toward the given aspects of entities [28,44,50] (e.g., the strong positive, positive, neutral, negative, strong negative). For example, in the opinion "*I have another phone. Its color is not nice, but its style is so modern*", there are two aspects of "*color*" and "*style*", and the user expresses negative and positive sentiments toward them, respectively. The sentiment analysis of aspects has a significant role and has been applied by both industry and academic communities deeply [2,28]. In general, one can state that previous methods have tried to solve the aspect-based sentiment analysis problem with quite good performance. However, their authors have analyzed the sentiment of aspects without considering the impact of fuzzy sentiments shown in opinions. To understand clearly the role of fuzzy sentiments, let's discuss some cases as follows: (i) Consider the opinion "*My shirt has a relatively modern style*". It has a sentiment phrase that is not clear, namely "*relatively modern*". Most of the previous methods treat the sentiment toward aspect "*style*" as positive because they only focus on the word "*modern*" disregarding the impact of the word "*relatively*". (ii) Consider the opinion "*The smell of this kind of coffee is not too special*". This short text has a fuzzy sentiment phrase, namely "*not too special*". Some methods treat the sentiment toward the aspect "*smell*" as positive because they only consider the role of the word "*special*" but ignoring the effects of both words "*not*" and "*too*". A few methods classify the sentiment toward this aspect as negative because they do not consider the effect of the word "*too*". (iii) In opinion "*Yesterday, I threw away my phone, although its camera and style are not so bad*". This short text has a sentiment phrase that is not clear, namely "*not so bad*". Some methods classify the sentiment toward two aspects, "*camera*" and "*style*" as positive, because they only focus on "*not bad*" while most of the others classify it as negative because they only focus on "*so bad*" without considering the impact of all three words simultaneously. From these examples, we can note that fuzzy sentiment has high impacts on the accuracy of sentiment analysis methods. However, the previous methods have handled them as regular emotions, which is not logical and has a not good impact on the performance of the sentiment analysis results.

Next, note that a decision tree is a decision-making support tool which includes learning and reasoning mechanisms. This tool uses a tree-like predictive model by mapping observations about an entity on several levels into a tree until the desired outcome is obtained [45]. A decision tree can express the discovered relationships as a set of rules that are easier to be used for decision-making. In addition, a decision tree uses a recursive divide-and-conquer strategy to group the dataset into partitions, owing to which all records in a partition have the same class label [42]. However, the user's sentiments are not always unequivocal (e.g., "*good*" or "*bad*") [36]. Many users like to use words that express vague feelings to voice their opinions, such as "*quite good*" or "*so bad*". In these examples, "*quite*" and "*too*" are called language uncertainties, and a decision tree without taking this aspect into account may decrease the efficacy of decision-making. Therefore, a different approach has been proposed, which uses fuzzy logic for building so called a fuzzy decision tree. This tree is different from the traditional decision tree in two respects [19]: The splitting criteria and inference process. The first of them is based on fuzzy restrictions and the second is based on fuzzy logic with the ability to model good knowledge details.

Sentiment analysis and fuzzy decision tree algorithms are vital tools to analyze the behavior of users toward products, movies, events, and other entities [13]. However, if these two approaches are used separately and are performed independently for decision-making, the efficiency of the decision support system could not be high. Therefore, we propose a combination of the sentiment analysis method and fuzzy decision tree algorithms to build a more robust support system for decision-making. Many methods supporting users in decision-making in the second component have yielded promising results [1,26]. Nevertheless, only a few methods approach the first component [37,46]. The reason of this fact follows from the difficulties in increasing efficiency of decision-making in the first case with a decision support system that is based on measuring user satisfaction, dissatisfaction, and hesitation in tweets. Specifically, there are no exact scales for measuring the user satisfaction level. The value of the user satisfaction level is estimated by abstract satisfaction indices that are hard to understand and compare. To address this problem, previous methods have been developed to automatically support decision-making based on the analysis of the user's Twitter opinions that are related to products, services, events, and policies. However, they propose some ways to help the users in decision-making without taking into account the influence of fuzzy sentiment phrases in analyzing the sentiment of aspects in tweets. Besides, most methods only identify user satisfaction for aspects without considering the effects of user dissatisfaction and hesitation [33,37,46]. However, in these methods, user satisfaction is only considered in the form of discrete values [37,46]. These disadvantages motivate us to propose a novel approach to help users in decision-making by using the combination of the sentiment analysis method with the fuzzy decision tree algorithm based on determining the degrees of satisfaction, dissatisfaction, and hesitation of users. We accordingly propose three steps. In the first step, the sentiments of tweets are analyzed as strong positive, positive, neutral, negative, and strong negative by calculating the sentiment scores of aspects in the tweets. In the second step, the degrees of user satisfaction, hesitation, and dissatisfaction according to each kind of sentiment are determined in order to build the decision matrix. In the final step, the fuzzy decision tree is constructed based on the decision matrix, and then, this tree is mined to discover a set of rules to support decision-making.

The remainder of this paper is organized as follows. In Section 2, we summarize the works related to the decision making methods. Section 3 presents a formal decision-making model based on the user satisfaction level in tweets. The research problem is listed in Section 4, and the proposed method is described in Section 5. The experimental results and evaluations are shown in Section 6. The conclusion and suggestions for future work are discussed in the last section.

2. Related works

In this section, we discuss the prominent research on decision-making support systems based on sentiment analysis and product rankings. Throughout the years, various approaches have been evolved to improve the performance of sentiment analysis methods. These approaches have been developed from the lexicon-based algorithms to machine learning algorithms and now to deep learning algorithms. As we have known, the natural language often expresses uncertainty in various cases. It is difficult to solve this problem even using the most advanced deep learning algorithms [20]. However, fuzzy logic can deal with this uncertainty by providing us with the ability to make decisions when there is ambiguity [6]. To express sentiment via opinions, many people habit to use uncertainty phrases in terms of the semantics. This is a practical approach due to as Wu [43] stated that the sentiment depends on inter and intrapersonal uncertainty. Therefore, to represent the uncertainty, the author had to use fuzzy sentiment phrases. How can we handle this uncertainty problem by using fuzzy logic systems? There are many methods using fuzzy logic in emotion modeling. Karyotis et al. [23] introduced the modeling approach for emotions aiming to incorporate human emotion into intelligent computer systems. The authors used a genetically optimized adaptive a fuzzy logic-based method to build a predictive framework and track the user's emotional trajectory over time. The fuzzy method has been evaluated in terms of its ability to model emotional states better than other existing machine learning methods. The performance of this method has been tested by implementing a personalized learning system with offline and online experiments. Meanwhile, Jefferson et al. [20] provided a sentiment analysis method based on fuzzy logic. The authors used fuzzy membership functions to transform continuous attributes into linguistic attributes and then extracted a set of fuzzy rules for classification sentiments. The experimental results proved that this method achieves marginally better than the other algorithms and enables defining different degrees of sentiment without a larger number of classes. Additionally, Bedi et al. [6] provided a new method to improve the performance of text sentiment analysis by combining fuzzy logic with the deep learning algorithm. This combination has taken advantage of the learning capabilities of the deep learning method and uncertainty handling abilities of fuzzy logic to give more appropriate sentiment analysis.

Besides fuzzy decision tree-based approaches to help users in decision-making, some other recent techniques supporting decision-making employ different fuzzy-based solutions, such as using fuzzy ontologies to support scene comprehension by evaluating different perspectives [11] and learning fuzzy cognitive maps to help decision-makers to make informed decisions [5], have been developed. This reason motivated Baykasoğlu et al. [5] who provided a new multiple attribute decision-making model by learning fuzzy cognitive maps (FCMs). This model has been built to help users in decision-making by assessing the future performance of alternatives. FCMs have been trained using the Jaya algorithm to pick up historical data patterns. Then, the decision-making matrices of short-term, medium-term, and long-term features have been constructed. Finally, decision-making matrices of future, current, and past have been considered, and the alternatives are ranked using closeness coefficients. The performance of this model is evaluated based on a dataset from the case company. The main limitation of this method is that the performance highly depends on the data set. Besides, it is time-consuming to generalize the obtained results. Meanwhile, Cavalier et al. [11] presented a method to help the multi-unmanned vehicle (UV) system in decision-making regarding situations that have been appeared in scenarios. This method has been built based on the group decision-making (GDM) model with consensus modeling. First, each UV has been made based on high-level situations from the detected events through a fuzzy-based event aggregation modeled with a fuzzy ontology. Then, a collective explanation of cases is attained by assessing each UV explanation. Finally, the sureness of the final group decision is evaluated using consensus and proximity metrics. This model allows the multi-UV system to express a global team judgment about the situations that occur in the scenario. The model can support human operators or an autonomous ground station regarding the system's decision reliability. However, the model has only achieved high performance in straightforward situations. UV teams are required to operate on the conditions from the same scenario.

Some studies have applied aspect-based sentiment analysis in decision-making for manufacturers for gaining valuable information from big data to monitor user satisfaction, for tracking product weaknesses, or for extracting user requirements. Among them, Zhang et al. [48] assessed the online Chinese opinions of body wash products of a cosmetic producer and filtered features, which users are most unsatisfied with, that need to be improved. Jin et al. [22] used user opinions to help designers in market-driven product design; they extracted significant information on aspects based on user requirements. Qi et al. [39] presented an approach to extract helpful comments of items design and analyzed sentiment to evaluate the effect of items aspects on user satisfaction. All three studies [22,39,48] did not consider the impact of the fuzzy sentiment when analyzing sentiments. Yussupova et al. [46] presented a novel domain-independent decision support approach based on user satisfaction. This approach researched user satisfaction by profoundly analyzing the user's comments. It is performed as a prototype of a decision support system. The authors evaluated the performance of this method by two experiments on hotels and bank customer reviews. The results proved the efficacy of this decision support system for quality management using customer satisfaction. Nevertheless, this study failed to consider the effect of user dissatisfaction and hesitation toward products. Besides, this method also ignored the effects of blurred sentiments. Most recently, the author of work [16] showed a decision support system that able to effectively support producers in advertising and marketing on dynamic social networks. This method continuously collected user reviews on promotions, products, and services. Then, by analyzing these comments, the reputation of companies has been estimated and is the basis for the next advertising and marketing actions. The advantage of this method is that it has developed a sentiment analysis tool that can calculate the degree of the sentiment of users, such as positive, negative, or neutral, in a review. This tool has been built by using a machine learning algorithm. Then, the reviews can be assigned labels based on cross emotional analysis and applied in such cases as restaurant and shop online. However, this method also ignored the role of fuzzy sentiments and has not yet considered the user's satisfaction levels in decision-making. The author of paper [31] tried to build a system, namely SentDesk that to be able to support the business by considering sentiments in users' opinions. This method was applied to some businesses in Ghana and was assessed that is better than humans. Despite that, the authors have also ignored the fuzzy sentiment phrases when analyzing the sentiment of the opinions and have not considered the user dissatisfaction and user hesitation when making decisions.

There also exists literature on supporting decision-making by ranking products using sentiment analysis. This topic has recently attracted the concern of researchers; however, the number of studies is still limited. One such study has been done by Zhang et al. [47],who presented a methodology to rank products by considering the aspects of products through online reviews. They classified the sentences in user opinions as positive or negative by analyzing sentiment based on the lexicon. Then, an improved page rank algorithm has been used to build a weighted and directed graph that indicates the quality of products; the products are ranked by mining this graph. Najmi et al. [29] analyzed the sentiment of online reviews regarding products based on the lexicon such as HD television and cameras in order to create ranking lists by computing the sentiment score of opinions, brand score of each product, and helpfulness of each review. While these methods evaluated user satisfaction for products by analyzing sentiment, they did not use multi-criteria decision-making (MCDM), which is a ranking procedure that considers the aspects of products and user satisfaction, dissatisfaction, and hesitation for these aspects. In addition, in [10] a decision support system for items ranking is published; this approach integrates the MCDM and the aspect-based sentiment analysis methods to rank alternative items. This method provided the best-fit alternative to potential users using a set of online item criteria and user opinions thereof. Liu et al. [26] proposed an approach to rank items through use's opinion by analyzing the user's sentiment and then using the interval-valued fuzzy technique for prioritizing of the similarity-valued items. Besides, Liu et al. [27] published initial research to express sentiment polarities into fuzzy numbers for finding out hesitations in comments. The authors classified online comments of automobiles into positive, negative, and neutral classes. Then, they ranked products based on preference-ranking organization methods and identified the condition weights subjectively. There also exist studies on the hotel choosing problem based on MCDM methods [32]. These works extracted sentiments from the opinions on social networks according to the pre-accepted framework on criteria regarding the product. However, none of these studies applied text reviews. They largely focused on supporting decision-making for the second case (presented in Section 1); very few researchers mention solving problems for the first case. A few studies have solved the above problem with many different approaches. Nevertheless, the previous methods do not mine the level of user satisfaction with aspects regarding given topics, nor do they analyze the level of user satisfaction for that topic as a whole. Phan et al. [37] provided a method to handle the mentioned disadvantages using the combination between the object's sentiment analysis and the binary decision tree mining. The results prove the efficiency of this approach. Nonetheless, this method did not consider the effect of user dissatisfaction and hesitation. The authors have not yet evaluated the influence of fuzzy sentiment when analyzing the sentiment of tweets.

Thus, the literature cited herein indicates the importance of the decision support system in research as well as its broad-based applications. To make them more effective, realistic, and relevant to user psychology is necessary. Generally, the previous methods are marked by impressive accuracy in applying approaches to data. In other words, the previous methods tried to support users in decision-making by using many different approaches applying in various data with quite an accuracy. However, most focus on helping users by assuming that users do not have specific information about their entity of interest. That is, they build the support system based on the second case (mentioned in Section 1) to rank alternatives and look for the best choice. In this study, we provide a decision support system in term of the first case. We assume that users are aware of the desired entity; they had some reviews related to its aspects, e.g., a user in a phone store, after looking at this phone, he/she can give some comments as follows: *"the color is gorgeous, the style is quite modern, but it only has 16 GB*". When deciding, they seek a support system that can help in decision-making—one that refers to opinions of other users who are using or have used a similar phone.

3. Formal decision-making model based on user satisfaction level in tweets

In this section, a new mathematical model for decision-making is presented. First, the tweet's generalized definition based on the main factors, such as topic, aspects of the topic, and sentiment of aspects, is given in Section 3.1. As the proposed method is based on the user's satisfaction level, the equations related to determining user satisfaction, hesitation, and dissatisfaction are introduced in Section 3.2. Finally, the significant result of the presented study is to help the user making a sensible decision related to a given topic; therefore, the definition of decision-making from the user's satisfaction level is introduced in Section 3.3.

3.1. Tweet

Consider a set of tweets $\mathcal{T} = \{t_1, t_2, \cdot\cdot, t_n\}$ where t_i $(i = 1, 2, \cdot\cdot, n)$ is a tweet. A tweet is a sequence of tokens $t = \langle w_1, w_2, \cdot\cdot, w_m \rangle$. Feature words are extracted from the tweets, where the feature words w_j $(j = 1, 2, \cdot\cdot, m)$ in a tweet t can belong to one of three key categories: topic, aspects of the topic, and the sentiment regarding aspects.

A topic can be any entity such as product, service, organization, event, or person. The topics are often nouns or noun phrases, and they are usually described in more detail by clauses or sentences regarding aspects of these topics.

A topic can have aspects that are mentioned in the tweets. An aspect is a specific detail, a subpart, or an attribute of the topic. An aspect is usually represented a noun or a noun phrase that is complemented by a phrase, clause, or sentence containing a user's emotion.

In a tweet given by a user, a sentiment regarding an aspect in a topic is an emotion of this user toward this aspect. We assume that the emotion can be one of five states: *Strong positive* (SP), *Positive* (P), *Neutral* (Ne), *Negative* (N), *Strong negative* (SN).

Let $\mathcal{P}(t)$ be the set of topics appearing in a tweet t.

Let $\mathcal{A}(t)$ be the set of aspects of a tweet t.

Let $\mathcal{S}(t)$ be the set of sentiments regarding the aspects of a tweet t.

Definition 1. A tweet is represented by a triple

$$t = \langle \mathcal{P}(t), \mathcal{A}(t), \mathcal{S}(t) \rangle \tag{1}$$

where

- For $p \in \mathcal{P}(t) : A(t, p)$ is the set of aspects of topic p in tweet t.
- For $a \in \mathcal{A}(t, p) : \mathcal{S}(t, p, a) \in \mathcal{S}(t)$ is the sentiment regarding aspect a of topic p in tweet t.
- $\mathcal{S}(t, p, a) \subseteq \{SP, P, Ne, N, SN\}$.

$S(t)$ is a set with repetitions (called multiset) defined by Lipski and Marek [30].

Thus,

$$\mathcal{A}(t) = \bigcup_{p \in \mathcal{P}(t)}^{\cdot} \mathcal{A}(t,p) \tag{2}$$

$$\mathcal{S}(t) = \bigcup_{p \in \mathcal{P}(t), a \in \mathcal{A}(t,p)}^{\cdot} \{\mathcal{S}(t,p,a)\} \tag{3}$$

where the symbol $\bigcup^{\cdot}$ expresses the sum of multisets.

A multiset is presented as the following example. For example, $S(t) = \{positive, positive, negative, negative, negative, neutral\}$ is called a multiset because, in this set element, *positive* appears twice, whereas *negative* and *neutral* appear three times and one time, respectively.

Example 1. Given a tweet "*This phone has the loose charging port. I got that soldered it. But it has a new battery. I changed it last month*". In this tweet, as the above presentation, the topic is "*Phone*". This topic has aspects such as "*charging port*" and "*battery*". And the sentiment regarding aspect "*charging port*" is *negative*, and the sentiment regarding aspect "*battery*" is *positive*.

3.2. User's satisfaction level

The user's satisfaction level, such as user satisfaction, hesitation, and dissatisfaction, can be described as follows. For $p \in \mathcal{P}(t)$ and $a \in \mathcal{A}(t)$:

- Let $\mathcal{S}^{++}(p,a)$ be the multiset of strong positive sentiments for aspect a of topic p in tweets from $\mathcal{T}$.
- Let $\mathcal{S}^{+}(p,a)$ be the multiset of positive sentiments for aspect a of topic p in tweets from $\mathcal{T}$.
- Let $\mathcal{S}^{\pm}(p,a)$ be the multiset of neutral sentiments for aspect a of topic p in tweets from $\mathcal{T}$.
- Let $\mathcal{S}^{-}(p,a)$ be the multiset of negative sentiments for aspect a of topic p in tweets from $\mathcal{T}$.
- Let $\mathcal{S}^{--}(p,a)$ be the multiset of strong negative sentiments for aspect a of topic p in tweets from $\mathcal{T}$.
- Let $\mathcal{S}^{*} = |\mathcal{S}^{++}(p,a)| + |\mathcal{S}^{+}(p,a)| + |\mathcal{S}^{\pm}(p,a)| + |\mathcal{S}^{-}(p,a)| + |\mathcal{S}^{--}(p,a)|$.
- Let $\mathcal{T}(p,a) \in \mathcal{T}$ be the set of tweets containing aspect a of topic p.

Definition 2. The user satisfaction, dissatisfaction, and hesitation for the aspect a of topic p in tweet t are determined respectively as follows:

$$\mathcal{F}(t,p,a) = \frac{|\mathcal{S}^{++}(p,a)| + |\mathcal{S}^{+}(p,a)|}{\mathcal{S}^{*} \times |\mathcal{T}(p,a)|} \tag{4}$$

$$\mathcal{D}(t,p,a) = \frac{|\mathcal{S}^{--}(p,a)| + |\mathcal{S}^{-}(p,a)|}{\mathcal{S}^{*} \times |\mathcal{T}(p,a)|} \tag{5}$$

$$\mathcal{H}(t,p,a) = \frac{|\mathcal{S}^{\pm}(p,a)|}{\mathcal{S}^{*} \times |\mathcal{T}(p,a)|} \tag{6}$$

where $\mathcal{F}(t,p,a)$ is the user satisfaction for aspect a of topic p in tweet t; $\mathcal{D}(t,p,a)$ is the user dissatisfaction for aspect a of topic p in tweet t; $\mathcal{H}(t,p,a)$ is the user hesitation for aspect a of topic p in tweet t.

Definition 3. The user satisfaction, dissatisfaction, and hesitation for topic p in tweet t are calculated as follows:

$$\mathcal{F}(t,p) = \sum_{a \in \mathcal{A}(t,p), t \in \mathcal{T}} \mathcal{F}(t,p,a) \tag{7}$$

$$\mathcal{D}(t,p) = \sum_{a \in \mathcal{A}(t,p), t \in \mathcal{T}} \mathcal{D}(t,p,a) \tag{8}$$

$$\mathcal{H}(t,p) = \sum_{a \in \mathcal{A}(t,p), t \in \mathcal{T}} \mathcal{H}(t,p,a) \tag{9}$$

where $\mathcal{F}(t,p)$ is the user satisfaction for topic p in tweet t; $\mathcal{D}(t,p)$ is the user dissatisfaction for topic p in tweet t; $\mathcal{H}(t,p)$ is the user hesitation for topic p in tweet t.

3.3. Decision-making

The sensible decision-making to support the user is the final result of the presented method. Deriving a decision from the user's satisfaction level is produced as follows.

Definition 4. For a set of tweets $\mathcal{T}$ and a set of topics $\mathcal{P}(t)$, a decision regarding a topic expressed via a user's opinion in a tweet $(m(t,p))$ is predicted based on the satisfaction, dissatisfaction, and hesitation of other users for this topic. This decision is determined as follows

$$m(t,p) = \begin{cases} Yes, & if \quad \mathcal{S}\rfloor(t,p) > \alpha, \\ No, & if \quad \mathcal{S}\rfloor(t,p) \leqslant \alpha. \end{cases} \tag{10}$$

where

$$\mathcal{S}\rfloor(t,p) = \begin{cases} \mathcal{F}(t,p) + \mathcal{H}(t,p), & if \quad \mathcal{H}(t,p) < \beta, \\ \mathcal{D}(t,p) + \mathcal{H}(t,p), & if \quad \mathcal{H}(t,p) \geqslant \beta. \end{cases} \tag{11}$$

α and β are thresholds, and they are set manually following the experiments.

4. Research problems

In this section, from the basic concepts and definitions related to determining the user's satisfaction level for a specific topic from tweets and helping the user in decision-making presented in the previous section, the research problems are introduced in Section 4.1. The research questions are stated at the end of this section.

4.1. Basic notions

Definition 5. Given a set of tweets $\mathcal{T}$, let $\mathcal{P}(\mathcal{T})$ be a set of topics from all tweets $\mathcal{T}$. The set $\mathcal{P}(\mathcal{T})$ is defined as follows

$$\mathcal{P}(\mathcal{T}) = \bigcup_{t\in\mathcal{T}} \mathcal{P}(t) \tag{12}$$

Definition 6. Given a set of tweets $\mathcal{T}$ and a set of topics $\mathcal{P}(\mathcal{T})$, let $\mathcal{A}(p)$ be a set of aspects of topic p. The set $\mathcal{A}(p)$ is defined as follows

$$\mathcal{A}(p) = \bigcup_{t\in\mathcal{T}, p\in\mathcal{P}(\mathcal{T})} \mathcal{A}(t,p) \tag{13}$$

Definition 7. Given a set of tweets $\mathcal{T}$ and a set of topics $\mathcal{P}(\mathcal{T})$ and a set of aspects $\mathcal{A}(p)$, let $\mathcal{S}(p,a)$ be a set of sentiments toward aspect a of topic p. The set $\mathcal{S}(p,a)$ is defined as follows

$$\mathcal{S}(p,a) = \dot{\bigcup_{t\in\mathcal{T}, p\in\mathcal{P}(\mathcal{T}), a\in\mathcal{A}(p)}} \mathcal{S}(t,p,a) \tag{14}$$

Definition 8. Given a set of tweets $\mathcal{T}$ and a set of topics $\mathcal{P}(\mathcal{T})$, a set of decisions related to aspects of topic p is denoted by $\mathcal{M}(p)$. The set $\mathcal{M}(p)$ is defined as follows

$$\mathcal{M}(p) = \dot{\bigcup_{t\in\mathcal{T}, p\in\mathcal{P}(\mathcal{T})}} m(t,p) \tag{15}$$

4.2. Research questions

Consider that a user is interested in a topic such as a *product*, *service*, or *event*. Each user has some reviews on this topic. However, the user wonders whether this topic is also used by other people. Thus, the user needs to refer to the opinions of the others who have been using this topic in order to make a sensible decision. In this study, we answer the following question: *How to support a user making a decision by measuring user satisfaction, dissatisfaction, and hesitation (called user satisfaction level) for each aspect of a given topic as well as for the topic in a tweet collection?* This question can be further divided into the four following sub-questions:

1. *Given a set of tweets describing a specific topic, how to identify the aspects and the sentiments of aspects for a given topic?*
2. *How to determine the user satisfaction level for the aspects of the topic by using the other users' sentiments for these aspects?*
3. *How to determine the user satisfaction level for a topic by using other users' sentiment for the topic?*
4. *How to build a system to help users in decision-making by using the user satisfaction level for a given topic?*

Table 1
Table with the definitions of symbols.

Symbols	Description	Symbols	Description
t	A tweet	$\mathcal{F}(t,p,a)$	The user satisfaction for aspect a of topic p in t
p	A topic	$\mathcal{D}(t,p,a)$	The user dissatisfaction for aspect a of topic p in t
a	An aspect	$\mathcal{H}(t,p,a)$	The user hesitation for aspect a of topic p in t
w	A word	$\mathcal{F}(t,p)$	The user satisfaction for topic p in tweet t
$\mathcal{F}$	A set of tweets	$\mathcal{D}(p,a)$	The user dissatisfaction for topic p in tweet t
$\mathcal{W}$	A set of words	$\mathcal{H}(p,a)$	The user hesitation for topic p in tweet t
$\mathcal{P}(t)$	A set of topics in tweet t	$m(t,p)$	A decision toward topic p via a user's opinion in t
$\mathcal{A}(t)$	A set of aspects in tweet t	$\mathcal{F}$	A set of fundamental sentiment words
$\mathcal{A}(t,p)$	A set of aspects of topic p in tweet t	$\mathcal{F}\int$	A set of fuzzy semantic words
$\mathcal{S}(t,p,a)$	A set of sentiments toward the aspects in t	$\mathcal{F}_{\surd}$	A set of fuzzy sentiment phrases
$\mathcal{T}(p,a)$	A set of tweets containing aspect a of topic p	$\mathcal{C}_{\surd}$	A set of clear sentiment phrases
$\mathcal{P}(T)$	A set of topics in tweets $\mathcal{T}$	$l2v$	A lexicon vector
$\mathcal{A}(p)$	A set of aspects of topic p	$se2v$	A semantic vector
$\mathcal{S}(p,a)$	A set of sentiments toward aspect a of topic p	$sy2v$	A word-type vector
$\mathcal{M}(p)$	A set of decisions related to aspects of topic p	$pl2v$	A sentiment score vector
m	The number of tokens/words	$ps2v$	A position vector
n	The number of tweets	$\mathcal{WV}$	A vector of a word

We can define the formulation of the research problems by using the above definitions as follows:

- For $t \in \mathcal{T}$, it is needed to determine $\mathcal{P}(\mathcal{T})$.
- For $p \in \mathcal{P}(\mathcal{T})$, it is needed to determine $\mathcal{A}(p)$.
- For $a \in \mathcal{A}(p)$, it is needed to determine $\mathcal{S}(p,a)$.
- For $\mathcal{S}(t,p,a) \in \mathcal{S}(p,a)$, it is needed to determine $\mathcal{S}(t,p,a), \mathcal{D}(t,p,a), \mathcal{H}(t,p,a), \mathcal{S}(t,p), \mathcal{D}(t,p), \mathcal{H}(t,p)$, and $\mathcal{M}(p)$.

5. Proposed method

This section presents an approach to answer the questions formulated in Section 4. This approach provides a method to support users in decision-making by using the user satisfaction level measured based on analyzing the sentiment of aspects and mining the fuzzy decision tree. The proposed method includes four main stages: (i) Extracting the topic of interest; (ii) Determining the aspects and sentiments of aspects in tweets based on their sentiment scores; (iii) Calculating the user satisfaction degree according to each kind of sentiment identified in the first step; (iv) Constructing a decision matrix, which then helps build the fuzzy decision tree. This tree is next mined to provide a set of rules aiming to assist users' decision-making. The symbols used in the following sections are described in Table 1. The workflow of this approach is shown in Fig. 1. Its components are described as follows:

- Topic modeling: The topic modeling step aims to determine the topic that a user is interested in and collects from Twitter all tweets regarding this topic based on the user's keywords. This stage is implemented by using the latent dirichlet allocation (LDA) model.
- Data cleaning: The tweets are first cleaned. E.g., removing punctuation marks, retweet symbols, URL symbols, hashtags, query terms. Next, one emoji icon is replaced by one describing text by using the Python emoji package.[2] Besides, tweets are often written with the unofficial style, which can contain many acronyms and spelling errors. These can decrease the efficiency of the proposal. Therefore, the Python-based Aspell library[3] is used for spelling corrections.
- Pre-processing: The tweets are preprocessed by implementing the tokenization and part of speech (POS) tagging.
- Word embeddings: This stage aims to convert tweets into the vectors of real numbers by using the feature ensemble model. The feature ensemble model is created by extracting the five feature vectors, such as the lexicon vector, the word-type vector, the sentiment score vector, the position vector, and the semantic vector.
- Extraction of aspects and their sentiments: This step is to determine aspects and their sentiments of the given topic using the combination of the bidirectional long short-term memory (BiLSTM) and conditional random fields (CRF) models (called the BiLSTM-CRF model).
- Prediction of the decision of users emphasized in tweets: In this step, the user satisfaction, dissatisfaction, and hesitation are first calculated based on the sentiment of aspects. Then, the fuzzy decision tree is constructed based on the user satisfaction levels. Finally, a set of rules is extracted from the fuzzy decision tree.
- Supporting the user in decision-making: The decision related to the given topic is made based on the extracted rules.

[2] https://pypi.org/project/emoji/.
[3] https://pypi.org/project/aspell-python-py2/

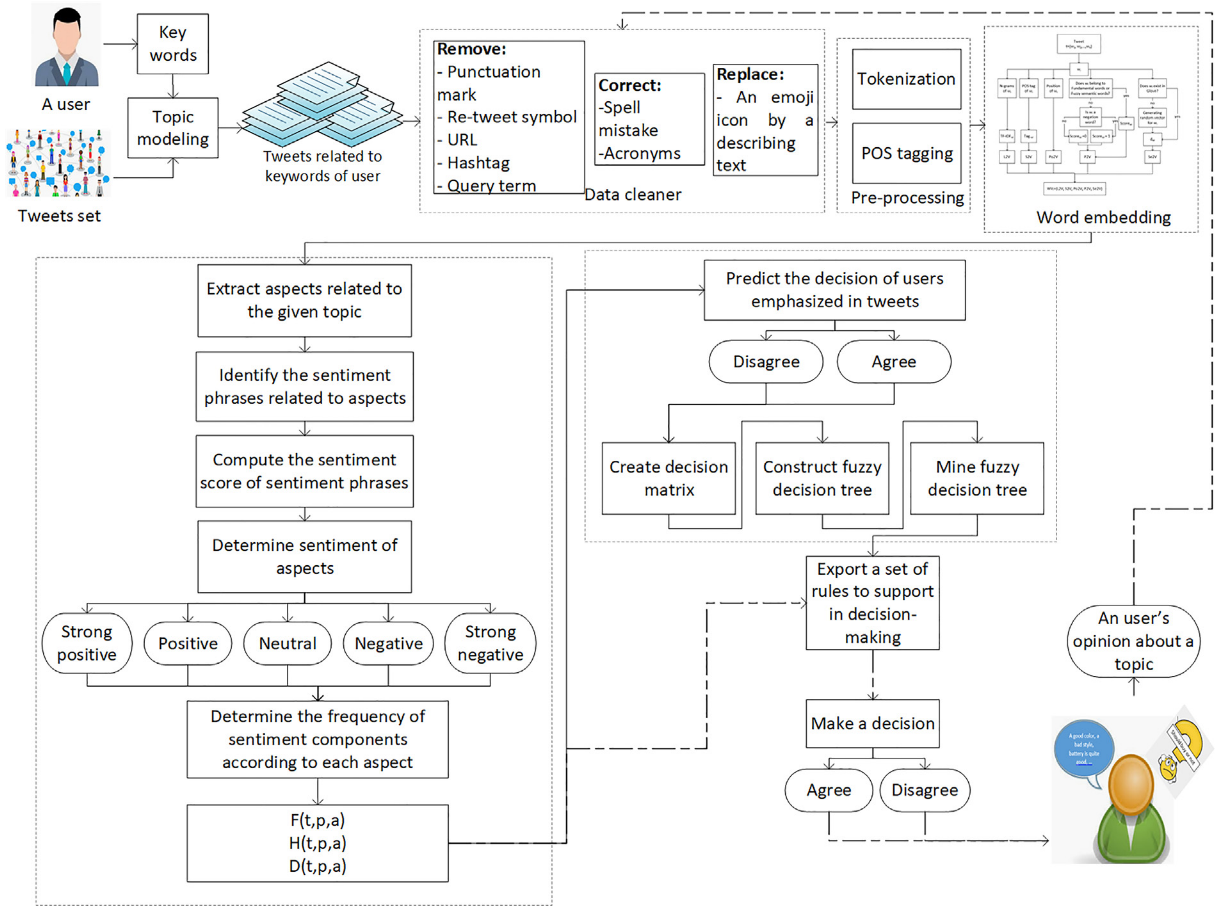

Fig. 1. Workflow of the proposed model for each topic.

5.1. Types of sentiment phrases

In this paper, we use the types of lexicons, such as fundamental sentiment words, negation words, and fuzzy semantic words, presented in [36] to construct the types of sentiment phrases. In which, fundamental sentiment words are words indicating basic emotions such as *"good"*, *"bad"*, *"excellent"*. These words consist of positive and negative words. Negation words, such as *"not"*, *"-n't"* are used to reverse the sentiment of fundamental sentiment words. For example, *"not"* + positive word => negative sentiment and vice versa. Fuzzy semantic words are words used to decrease or increase the sentiment degree of fundamental sentiment words. These words consist of the intensifier (such as *"too"*, *"so"*), diminisher words (such as *"fairly"*, *"quite"*). There are two types of sentiment phrases that are clear sentiment phrases and fuzzy sentiment phrases. Next, we explain how to create types of sentiment phrases as follows:

- Clear sentiment phrases (CSPs) usually express emotions clearly. These CSPs include at least one word as one fundamental sentiment word or the combination between one fundamental sentiment word and one negation word. The CSPs can be classified into two main types, such as CSP type I: These CSPs contain only one fundamental sentiment word, such as *"bad"* or *"good"*; and CSP type II: These CSPs contain one fundamental sentiment word and one negation word, and they are grouped into the two sub-types: (i) Negation word and negative word, such as *"not bad"*; (ii) Negation word and positive word, such as *"not good"*.
- Fuzzy sentiment phrases (FSPs) do not show emotion clearly. These phrases have more than one word, of which at least one is a fundamental sentiment word and the other word(s) can be either a fuzzy semantic word or a combination of a negation word and a fuzzy semantic word. The FSPs are divided into two main types, such as FSP type I (for example: *"too bad"*, *"fairly bad"*, *"so good"*, and *"slightly good"*) and FSP type II (for example: *"not too bad"*, *"not fairly bad"*, *"not so good"*, and *"not slightly good"*). The detail of these types of FSPs is presented in [36].

5.2. Topic extraction

Given a set of tweets denoted by $\mathcal{T} = \{t_1, \cdot\cdot, t_n\}$, topic modeling creates a set of topics denoted by $\mathcal{P} = \{p_1, \ldots, p_k\}$. In this study, the latent dirichlet allocation (LDA) method is used for topic modeling. LDA was first introduced by [8] and has been widely applied in much research because of its ease of understanding and versatility [13,21]. In the LDA model, a tweet within the tweets set is shown as a probability distribution over hidden topics, and the distribution of topics shares a common Dirichlet prior [8]. A hidden topic is represented as a probability distribution over the words in the vocabulary, and the distribution of words shares another common Dirichlet prior [21]. A topic model can be generated using the following process [21]:

Let m denotes the size of the vocabulary, n the total number of tweets $\mathcal{T}$. A topic model shows the words in a set of tweets as mixtures of k topics, words within tweets, denoted by $w_{e,q}, (e = 1, 2, \ldots, m; q = 1, 2, \ldots, n)$, are observed variables while the probability distribution over words of each hidden topic $\varphi_l(l = 1, 2, \ldots, k)$ with hyperparameter γ, the topic distribution per tweet $\theta_e, (e = 1, 2, \ldots, m)$ with hyperparameter δ and the assignment of the per-word topic $z_{e,q}$ are hidden variables. For each tweet, the words are created by the following steps:

- First, a distribution over topics is randomly selected. A topic is randomly selected for each word in the tweet based on the distribution over topics.
- Second, the hidden random variables (φ_l and θ_e) are not observed that could be learned through Gibbs sampling method[4] via maximizing the probability $p(T|\delta, \gamma)$ [14] as the following equation:

$$p(\mathcal{T}|\delta, \gamma) = \prod_{e=1}^{m} \int p(\theta_e|\delta)(\prod_{q=1}^{n} \sum_{z_{e,q}} p(z_{e,q}|\theta_e) p(w_{e,q}|z_{e,q}, \gamma)) d\theta_e \tag{16}$$

After implementing the LDA model, we obtain two main outputs: the word distribution for each topic $\varphi_l, (l = 1, 2, \ldots, k)$ and the topic distribution for each tweet $\theta_e, (e = 1, 2, \ldots, m)$. The topic extraction step is illustrated in Fig. 2.

5.3. Word representation

The tweets are almost written in natural language. So, to be able to use tweets as the input of algorithms, we need an intermediate step to convert tweets into vectors of real numbers. This step is essential and directly affects the performance of the algorithms. If it is well implemented, the accuracy of the algorithm will be significantly increased. In this work, each word is converted into a vector, this process is called word embedding. Word embedding is executed by concatenating the feature vectors of each word, that is, lexicon vector ($l2v$), word-type vector ($sy2v$), sentiment score vector ($pl2v$), position vector ($ps2v$), and semantic vector ($se2v$) [38]; in which: Table 1.

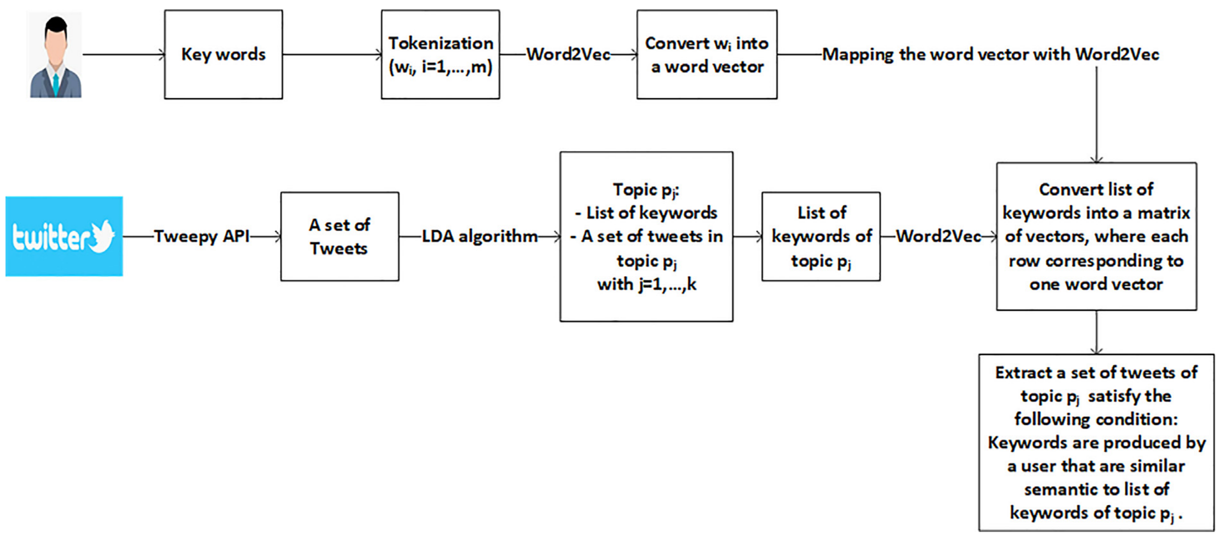

Fig. 2. Illustration of topic extraction.

[4] https://gist.github.com/mblondel/542786#file-lda_gibbs-py.

Table 2
Creating lexical vector of a word.

Words	1-gram	2-grams				3-grams			
w_i	w_i	w_iw_1	w_iw_2	...	w_iw_n	$w_iw_1w_2$	$w_iw_1w_3$	...	$w_iw_{n-1}w_n$
TF-IDF(w_i)	t_i	t_{i1}	t_{i2}	...	t_{in}	t_{i12}	t_{i13}	...	t_{in-1n}
$l2v(w_i)$	$[t_i,$	$t_{i1},$	$t_{i1},$	...,	$t_{in},$	$t_{i12},$	$t_{i13},$		$t_{in-1n}]$

Table 3
Creating semantic vector of a word.

Word	GloVe embeddings
w_i	Cannot find w_i
w_j	[−0.54403, 0.60274, −0.14543, −0.023398,. . ., 0.022125]
$se2v(w_i)$	random()
$se2v(w_j)$	[−0.54403, 0.60274, −0.14543, −0.023398,. . .,0.022125]

Table 4
Creating word-type vector of a word.

Words	CD	DT	IN	JJ	TO	MD	NN	PRP	SYM	TO	VB	...	WRB
w_i	0	0	0	0	0	0	0	1	0	0	0	...	0
$sy2v(w_i)$						[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, . . ., 0]							

Table 5
Creating sentiment score vector of a word.

Word	Negation word	Fundamental word	Fuzzy semantic word
w_i	Score $(w_i) = \mathcal{S}\rfloor(n)$	Score$(w_i) = \mathcal{S}\rfloor(f,p)$	Score $(w_i) = \mathcal{S}\rfloor(fs)$
$pl2v$ (w_i)		[Score (w_i)]	

Parameter $l2v$ of a word is created by using its n-grams. First, we extract n-grams, such as 1-gram, 2-grams, and 3-grams, stated from this word. Then, we calculate the term frequency-inverse document frequency (TF-IDF) value of these n-grams. And finally, we concatenate the TF-IDF value of n-grams into one vector. The process to build parameter $l2v$ is illustrated in Table 2.

Parameter $se2v$ of a word is built based on the GloVe embeddings[5] [34]. First, we find in the GloVe dataset; if this word exists in GloVe, we will extract the corresponding word vector of this word and assign it to the semantic vector. Otherwise, if this word is not in GloVe, we will create a random vector of this word and set it to the semantic vector. Table 3 illustrates the steps to create parameter $se2v$.

Parameter $sy2v$ of a word is built like the following steps: First, we identify the POS tag of this word using the Natural Language Toolkit.[6] And then, we convert this word into a one-hot encoding vector based on the position of the corresponding POS tag. The steps to convert a word into parameter $sy2v$ is shown in Table 4.

Parameter $pl2v$ of a word is built based on the sentiment score of kinds of words, such as the score of fundamental sentiment words (positive, negative), negation words, fuzzy semantic words (intensifier, and diminisher). First, we determine the kind of word of this word. Then the sentiment score of this word is calculated. Finally, the parameter $pl2v$ of this word is created based on the value of its sentiment score. The way to create parameter $pl2v$ is presented in Table 5.

Parameter $ps2v$ of a word is created by computing the relative distances of this word to the remaining words in a tweet (denoted by $d_{ij}(j, i = 1, 2, \ldots, n)$) and shown in Table 6.

In which, d_{ij} is calculated as the Eq. 17.

$$d_{ij} = \frac{position(w_j) - position(w_i)}{|max_{t\in\mathcal{T}}length(t)|} \tag{17}$$

where *position* is a function determining the order of word in a tweet. $|max_{t\in\mathcal{T}}length(t)|$ is a function to give the number of words of the longest tweet.

[5] http://nlp.stanford.edu/projects/glove/..
[6] https://www.nltk.org/.

Table 6
Creating position vector of a word.

Words	Distance to				$ps2v(w_i)$
	w_0	w_1	...	w_n	
w_i	d_{i0}	d_{i1}	...	d_{in}	$[d_{i0}, d_{i1}, \ldots, d_{in}]$

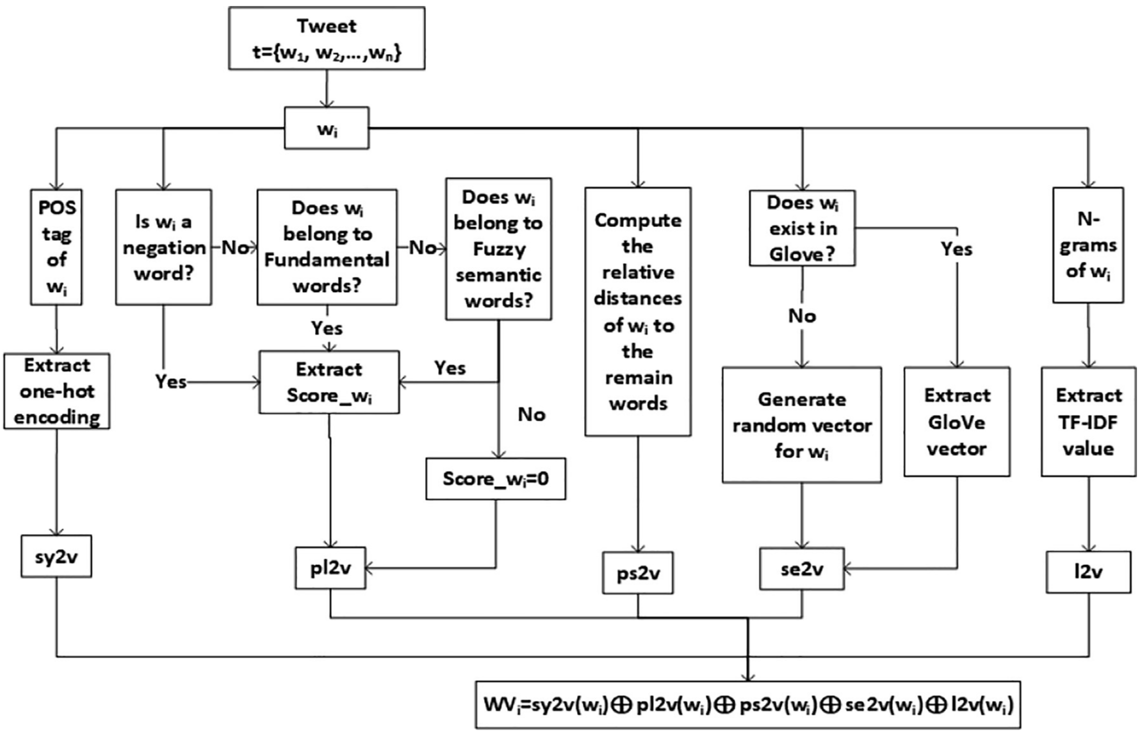

Fig. 3. Workflow of word representation.

The model used to create the word embedding in each tweet is shown in Fig. 3.

5.4. Determination of aspects

Let $\mathcal{C}_{\surd}$ be the set of CSPs. Let $\mathcal{F}_{\surd}$ be the set of FSPs.

Definition 9. The sentiment relation between token w_k and w_h is defined by function Θ as follows:

$$\Theta(w_h, w_k) = \begin{cases} 1, & \text{if} \quad (w_k \text{ referring to } w_h \wedge w_h \in \mathcal{C}_{\surd} \cup \mathcal{F}_{\surd}) \\ 0, & \text{otherwise.} \end{cases} \tag{18}$$

Definition 10. An aspect *a* of topic *p* existing in tweet *t* is a token w_k that satisfies two criteria at the same time: w_k must be a noun or noun phrase and have a relation with at least one sentimental word in this tweet. The aspect *a* is defined as:

$$a = \{w_k | tag(w_k) = \text{'}NOUN\text{'}, \exists w_h \in t : \Theta(w_k, w_h) = 1\}. \tag{19}$$

In this study, we use a combination of the bidirectional long short-term memory (BiLSTM) and conditional random fields (CRF) models (called the BiLSTM-CRF model) [18] to identify aspects and sentiment of them in each tweet. This combination takes advantage of both models [18]. The BiLSTM model is demonstrated to be highly effective in sequence labeling because this model enables access to both past and future contexts of words. That means it allows the hidden states to capture the features of sequences to both the front and back directions and enables it to extract the contextual order information of sequences better and then labels a token. The BiLSTM always considers the constraints between the current label to the front and back labels of the words. However, if we simply predict the labels of words by only providing the above-mentioned hidden states independently to a Softmax layer (It implies that we only use the BiLSTM model), then such correlations will be less likely to be broken. Whereas, the CRF model feeds the most popular method for controlling the structure prediction because it uses a series of potential functions to approximate the conditional probability of the output label sequence given the input word sequence.

The BiLSTM-CRF model is built according to the following phases:

(1) Word embedding layer: Each word is described by a vector $\mathcal{WV}$ based on concatenating five sub-vectors such as $l2v, sy2v, ps2v, pl2v$, and $se2v$. The vector $\mathcal{WV}$ is expressed as follows:

$$\mathcal{WV} \in R^{d\times 1} \wedge \mathcal{WV} = v_1 \oplus v_2 \oplus v_3 \oplus \ldots \oplus v_n \tag{20}$$

where $\oplus$ is the concatenation operator, d is the dimension of $v_i, v_i = l2v(w_i) \oplus sy2v(w_i) \oplus pl2v(w_i)$ $\oplus ps2v(w_i) \oplus se2v(w_i)(v_i = v(w_i))$. Then, we have $t = (\mathcal{WV}_1, \mathcal{WV}_2, \ldots, \mathcal{WV}_n)$.

(2) BiLSTM Layer: This layer extracts the significant features from the word embedding layer to use in the next layer. The output of this BiLSTM layer is a sequence of the hidden states for each input word vector, denoted as $(h_1, h_2, \ldots, h_n)$. Each final hidden state is the concatenation of the forward $\overleftarrow{h_i}$ and backward $\overrightarrow{h_i}$ hidden states. The output of this layer is generated as follows:

$$\overleftarrow{h_i} = lstm(\mathcal{WV}_i, \overleftarrow{h_{i-1}}), \overrightarrow{h_i} = lstm(\mathcal{WV}_i, \overrightarrow{h_{i+1}}) \tag{21}$$

$$\langle_i = [\overleftarrow{h_i}, \overrightarrow{h_i}] \tag{22}$$

(3) CRF layer: This layer predicts the final label sequence $y = (y_1, y_2, \ldots, y_n)$ from the hidden states $h = (h_1, h_2, \ldots, h_n)$, where y_i is in the set of all possible labels. Let $\mathcal{Q} \in \mathcal{R}^{n\times k}$ be an output matrix from BiLSTM layer, where k is the number of labels. Let $\mathcal{K}$ be the transition matrix of probability $\mathcal{Q}$. Thus, the label probability matrix can be determined as follows:

$$s(t, y) = \sum_{i=0}^{n} \mathcal{K}_{y_{i,i+1}} + \sum_{i=1}^{n} \mathcal{Q}_{i,y_i} \tag{23}$$

For each y, a softmax function is used to identify the probability over all possible label sequences as follows:

$$p(y|t) = \frac{e^{(t,y)}}{\sum e^{(t,y)}} \tag{24}$$

Subsequently, we achieve the maximal log probability of the correct label sequence based on the following equation:

$$log(p(y|t)) = s(t, y) - log(\sum e^{(t,y))} \tag{25}$$

Finally, the probability is maximized as y^* to determine potential label sequences. The value of y^* is determined as follows:

$$y^* = argmaxs(t, y) \tag{26}$$

The parameters of the BiLSTM-CRF model are continuously updated in the training process on the labeled tweets through the back-propagation algorithm with stochastic gradient descent. Additionally, during the training process, the dropout technique is used to avoid overfitting. Also, to further improve the efficacy of determining aspects and their sentiments in each tweet, we provide a set of rules to label the training data, as presented in our previous study [35].

5.5. Determining the sentiment of aspects

Let $\mathcal{F}$ be a set of fundamental sentiment words; $\mathcal{F}\int$ be a set of fuzzy semantic words; $\mathcal{F}_{\surd}$ be a set of fuzzy sentiment phrases; and $\mathcal{C}_{\surd}$ be a set of clear sentiment phrases.

- For $fs \in \mathcal{F}\int$: let $\mathcal{S}\rfloor(fs)$ be a sentiment score of fs.
- For $f \in \mathcal{F}$: let $\mathcal{S}\rfloor(f)$ be a sentiment score of f.
- For $fp \in \mathcal{F}_{\surd}$: let $\mathcal{S}\rfloor(fp)$ be a sentiment score of fp.
- For $cp \in \mathcal{C}_{\surd}$: let $\mathcal{S}\rfloor(cp)$ be a sentiment score of cp.

In which, $\mathcal{S}\rfloor(f)$ and $\mathcal{S}\rfloor(fs)$ are computed as proposed by Phan et al. in the paper [38].

Definition 11. The sentiment score of a clear sentiment phrase *cp*, denoted by $\mathcal{S}\rfloor(cp)$ is determined based on the score of the fundamental sentiment words in this *cp*. The value of the $\mathcal{S}\rfloor(cp)$ is computed as:

$$\mathcal{S}\rfloor(cp) = (-1)^i((-1)^j\mathcal{S}\rfloor(f)) \tag{27}$$

where

$$i = \begin{cases} 1, & if \quad (\mathcal{S}\rfloor(f) \leqslant 0) \\ 2, & if \quad (\mathcal{S}\rfloor(f) > 0) \end{cases} \tag{28}$$

$$j = \begin{cases} 1, & if \quad (cp \text{ is a CSP of Type II}) \\ 2, & if \quad (cp \text{ is a CSP of Type I}) \end{cases} \tag{29}$$

Definition 12. The sentiment score of a fuzzy sentiment phrase *fp*, denoted by $\mathcal{S}\rfloor(fp)$, is identified based on the score of the fundamental sentiment and fuzzy semantic words in this *fp*. The value of *fp* is computed as:

$$\mathcal{S}\rfloor(fp) = (-1)^i((-1)^k\mathcal{S}\rfloor(f) + (-1)^j\mathcal{S}\rfloor(fs)) \tag{30}$$

where

$$i = \begin{cases} 1, & if \quad (\mathcal{S}\rfloor(f) \leqslant 0) \\ 2, & if \quad (\mathcal{S}\rfloor(f) > 0) \end{cases} \tag{31}$$

$$k = \begin{cases} 1, & if \quad (fp \text{ is a FSP of Type II}) \\ 2, & if \quad (fp \text{ is a FSP of Type I}) \end{cases} \tag{32}$$

$$j = \begin{cases} 1, & if \quad (fs \text{ is a diminisher word}) \\ 2, & if \quad (fs \text{ is an intensifier word}) \end{cases} \tag{33}$$

From that, for $w \in \mathcal{F}_{\surd} \cup \mathcal{C}_{\surd}$, we have:

$$\mathcal{S}\rfloor(w) = \begin{cases} \mathcal{S}\rfloor(cp), & if \quad (w \in \mathcal{C}_{\surd}) \\ \mathcal{S}\rfloor(fp), & if \quad (w \in \mathcal{F}_{\surd}) \end{cases} \tag{34}$$

Definition 13. For $a \in \mathcal{A}(p)$, the sentiment of aspect *a* in tweet *t* is identified by using the sentiment relation between the sentiment phrase w_k and this aspect as well as the sentiment score of the sentiment phrase. A sentiment of an aspect in a tweet is calculated as:

$$\mathcal{S}(t,p,a) = \begin{cases} SP, & if \quad (\exists w : w \in (\mathcal{F}_{\surd} \cup \mathcal{C}_{\surd}) \wedge \Theta(w,a) = 1) \wedge (0.6 < \mathcal{S}\rfloor(w) \leqslant 1) \\ P, & if \quad (\exists w : w \in (\mathcal{F}_{\surd} \cup \mathcal{C}_{\surd}) \wedge \Theta(w,a) = 1) \wedge (0.2 < \mathcal{S}\rfloor(w) \leqslant 0.6) \\ Ne, & if \quad (\exists w : w \in (\mathcal{F}_{\surd} \cup \mathcal{C}_{\surd}) \wedge \Theta(w,a) = 1) \wedge (-0.2 \leqslant Sc(w) \leqslant 0.2) \\ N, & if \quad (\exists w : w \in (\mathcal{F}_{\surd} \cup \mathcal{C}_{\surd}) \wedge \Theta(w,a) = 1) \wedge (-0.6 \leqslant \mathcal{S}\rfloor(w) < -0.2) \\ SN, & if \quad (\exists w : w \in (\mathcal{F}_{\surd} \cup \mathcal{C}_{\surd}) \wedge \Theta(w,a) = 1) \wedge (-1 \leqslant \mathcal{S}\rfloor(w) < -0.6) \end{cases} \tag{35}$$

5.6. Decision-making support based on the fuzzy decision tree

In this section, methodologies used in decision-making support based on the fuzzy decision tree are overviewed. First, the algorithm to construct a decision support matrix is given in Section 5.6.1. As the proposed decision method employs the fuzzy decision tree algorithm extended based on the fuzzy Iterative Dichotomiser 3 (ID3) algorithm, the fuzzy decision tree algorithm is introduced in Section 5.6.2.

5.6.1. Constructing a decision support matrix

In this step, we construct a matrix (called a decision support matrix) using the user satisfaction ($\mathcal{F}(t,p,a)$), hesitation ($\mathcal{H}(t,p,a)$), and dissatisfaction ($\mathcal{D}(t,p,a)$), where the value of $\mathcal{F}(t,p,a), \mathcal{H}(t,p,a)$ and $\mathcal{D}(t,p,a)$ is calculated based on the sentiment of aspects of the given topic. This matrix is the firs step to build the fuzzy decision tree. Assume that $\mathcal{M}$ is a matrix that expresses the relationship between user satisfaction and the predicted decision existing in tweets. The decision support matrix is shown as follows:

$$\mathcal{M} = \begin{bmatrix} \mathcal{F}(t_1,p,a_1) & \mathcal{H}(t_1,p,a_1) & \mathcal{D}(t_1,p,a_1) & \cdots & \mathcal{F}(t_1,p,a_m) & \mathcal{H}(t_1,p,a_m) & \mathcal{D}(t_1,p,a_m) & m(t_1,p) \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots \\ \mathcal{F}(t_n,p,a_1) & \mathcal{H}(t_n,p,a_1) & \mathcal{D}(t_n,p,a_1) & \cdots & \mathcal{F}(t_n,p,a_m) & \mathcal{H}(t_n,p,a_m) & \mathcal{D}(t_n,p,a_m) & m(t_n,p) \end{bmatrix}$$

The decision support matrix is built based on the following Algorithm 1.

Algorithm 1. Constructing a decision support matrix algorithm

$\mathcal{T}$
$\mathcal{P}(\mathcal{T})$
$\mathcal{A}(p)$
Output: a matrix $\mathcal{M}$
for $i = 0$ to $n-1$ **do**
$\mathcal{F}(t_i,p) = \mathcal{D}(t_i,p) = \mathcal{H}(t_i,p) = 0$
$\mathcal{S}^* = 0$; $|\mathcal{S}^{++}(p,a_j)| = |\mathcal{S}^{+}(p,a_j)| = |\mathcal{S}^{\pm}(p,a_j)| = |\mathcal{S}^{-}(p,a_j)| = |\mathcal{S}^{--}(p,a_j)| = 0$
for $j = 0$ to $m-1$ **do**
if $\mathcal{S}(t_i,p,a_j)$ = "SP" **then**
$|\mathcal{S}^{++}(p,a_j)| = |\mathcal{S}^{++}(p,a_j)| + 1$
else if $\mathcal{S}(t_i,p,a_j)$ = "P" **then**
$|\mathcal{S}^{+}(p,a_j)| = |\mathcal{S}^{+}(p,a_j)| + 1$
else if $\mathcal{S}(t_i,p,a_j)$ = "Ne" **then**
$|\mathcal{S}^{\pm}(p,a_j)| = |\mathcal{S}^{\pm}(p,a_j)| + 1$
else if $\mathcal{S}(t_i,p,a_j)$ = "N" **then**
$|\mathcal{S}^{-}(p,a_j)| = |\mathcal{S}^{-}(p,a_j)| + 1$
else if $\mathcal{S}(t_i,p,a_j)$ = "SN" **then**
$|\mathcal{S}^{--}(p,a_j)| = |\mathcal{S}^{--}(p,a_j)| + 1$
end if
$\mathcal{S}^* = |\mathcal{S}^{++}(p,a_j)| + |\mathcal{S}^{+}(p,a_j)| + |\mathcal{S}^{\pm}(p,a_j)| + |\mathcal{S}^{-}(p,a_j)| + |\mathcal{S}^{--}(p,a_j)| = 0$
$\mathcal{F}(t_i,p,a_j) = \frac{|\mathcal{S}^{++}(p,a_j)|+|\mathcal{S}^{+}(p,a_j)|}{\mathcal{S}^* \times |\mathcal{T}(p,a_j)|}$; $\mathcal{D}(t_i,p,a_j) = \frac{|\mathcal{S}^{--}(p,a_j)|+|\mathcal{S}^{-}(p,a_j)|}{\mathcal{S}^* \times |\mathcal{T}(p,a_j)|}$; $\mathcal{H}(t_i,p,a_j) = \frac{|\mathcal{S}^{\pm}(p,a_j)|}{\mathcal{S}^* \times |\mathcal{T}(p,a_j)|}$
$h = 3 \times j$; $\mathcal{M}[i,h] = \mathcal{F}(t_i,p,a_j)$; $\mathcal{M}[i,h+1] = \mathcal{H}(t_i,p,a_j)$; $\mathcal{M}[i,h+2] = \mathcal{D}(t_i,p,a_j)$
end for
$\mathcal{F}(t_i,p) = \mathcal{F}(t_i,p) + \mathcal{F}(t_i,p,a_j)$; $\mathcal{D}(t_i,p) = \mathcal{D}(t_i,p) + \mathcal{D}(t_i,p,a_j)$; $\mathcal{H}(t_i,p) = \mathcal{H}(t_i,p) + \mathcal{H}(t_i,p,a_j)$
if $\mathcal{H}(t_i,p) < \beta$ **then**
$\mathcal{S}\rfloor(t_i,p) = \mathcal{F}(t_i,p) + \mathcal{H}(t_i,p)$
else
$\mathcal{S}\rfloor(t_i,p) = \mathcal{D}(t_i,p) + \mathcal{H}(t_i,p)$
end if
if $\mathcal{S}\rfloor(t_i,p) > \alpha$ **then**
$\mathcal{M}[i,h+1] =$"Yes"
else
$\mathcal{M}[i,h+1] =$"No"
end if
end for
return $\mathcal{M}$

The thresholds α and β are set manually following the experiments. We carried out an exhaustive search for values of the thresholds α from 0 to 1 and β from 0 to 0.1. In each trial, we adjusted the thresholds with an increment of 0.05 for α and an increment of 0.01 for β. And an evaluation measure was necessary to select the highly reliable instances of α and β in an exhaustive search. Therefore, in this study, the evaluation measure is chosen, which is the value of the $\mathcal{F}_1$ score. The highest $\mathcal{F}_1$ score is 80% with the thresholds $\alpha = 0.2$ and $\beta = 0.05$. Selecting a higher value for alpha will end up in more decisions misclassified to *"Yes"*. A lower value of alpha will end up with most decisions assigned to *"No"* which is equivalent to a not good F_1 score.

5.6.2. Building the fuzzy decision tree

Two significant procedures need to execute to construct the fuzzy decision tree for decision-making: building a fuzzy decision tree and extracting classification rules. In this study, the fuzzy ID3 algorithm [33,7,19] is employed to construct a fuzzy decision tree with the following steps:

- Define the fuzzy database, that is, the fuzzing for the domains of the continuous features by mapping the continuous areas into the discrete domains based on the set of linguistic labels.
- Take the place of the continuous attributes on the decision support matrix by the linguistic labels of the fuzzy sets with the highest similarity.
- Determine the entropy and information gain of each attribute to identify the root nodes of the tree until all mentioned-attributes are classified.

When applying the fuzzy ID3 algorithm to construct the fuzzy decision tree, it is necessary to notice the following issues: the method of partitioning the attribute value; the method of selecting the root node; the method of checking the degree of the branching attribute to decide which the branches follow down of the root node; and the method of labeling the leaf node to determine classes for which the leaf nodes stand.

5.6.2.1. The fuzzy decision tree algorithm. To build the fuzzy decision tree, we introduce a fuzzy decision tree algorithm that is extended based on the fuzzy ID3 algorithm [19]. This algorithm creates a fuzzy decision tree using fuzzy sets defined by a user for all attributes. This tree includes nodes for testing condition attributes, edges for branching by values of fuzzy sets defined by a user, and leaves for expressing decisions (decision-making). Our algorithm is described as follows:

Assume that we have the decision support matrix $\mathcal{M}$ (see Section 5.6.1), where each row of matrix $\mathcal{M}$ has m numerical values for attributes $\mathcal{C} = \{\mathcal{F}(t,p,a_1), \mathcal{H}(t,p,a_1), \mathcal{D}(t,p,a_1) \cdot\cdot, \mathcal{F}(t,p,a_m),\mathcal{H}(t,p,a_m), \mathcal{D}(t,p,a_m)\}, \mathcal{C}_i \in [0,1], (i = 1,2,\ldots,m)$; and one classified class $m(t,p) \in \{m_1, m_2\}$; and fuzzy sets $\mathcal{F}\ddagger_{i1}, \mathcal{F}\ddagger_{i2}, \ldots, \mathcal{F}\ddagger_{ik}, \mathcal{F}\ddagger_{ij} \in \{high, low\}, (j = 1,2,\ldots,k)$, for the attribute $\mathcal{C}_i$. Then the fuzzy decision tree algorithm is implemented in the following steps:

- **Step 1**: Create a fuzzy version of the decision support matrix $\mathcal{M}$, called the fuzzy decision support matrix and denoted by $\mathcal{M}_{\mathcal{F}}$, where the values in $\mathcal{M}$ are converted from the numerical values to the fuzzy values and put into $\mathcal{M}_{\mathcal{F}}$ by using the following membership function as

$$\mu_{\mathcal{F}\ddagger_{ij}}(\mathcal{C}'_i) = \begin{cases} low, & if \quad \mathcal{C}_i = 0, \\ high, & otherwise. \end{cases} \tag{36}$$

We can note that all condition attributes are the same form of continuous values. Therefore, we only need to build a membership function applying for all attributes. A fuzzy subset of the condition attribute $\mathcal{C}_i$ is described by a membership function $\mu_{\mathcal{F}\ddagger_{ij}}(\mathcal{C}'_i) : \mathcal{C}_i \to [0,1]$, which expresses the degree to which $\mathcal{C}_i$ belongs to the set $\mathcal{F}\ddagger_{ij}$. A fuzzy linguistic variable (such as $\mathcal{C}'_i$) is an attribute whose domain contains linguistic values (also called fuzzy terms), which are labels for the fuzzy subsets. In other words, the meaning of a fuzzy term is defined by a fuzzy set, which itself is defined by its membership function. For example, consider the condition attribute $\mathcal{C}_i$, it is a continuous attribute. It becomes a fuzzy variable when two linguistic terms, such as "*low*" and "*high*", are used as domain values. The defining fuzzy sets usually overlap and should cover the whole condition attributes. How has the fuzzy membership been fixed? In this study, the membership function was created using a grid partitioning method for each input. This method is presented by Rajabi et al. in [40]. A combination algorithm between least squares and backpropagation was then used to adjust the parameters.

After fuzzing we have the fuzzy decision support matrix $\mathcal{M}_{\mathcal{F}}$, where each row has m fuzzy values for condition attributes $\mathcal{C}' = \{\mathcal{F}(t,p,a_1), \mathcal{H}(t,p,a_1), \mathcal{D}(t,p,a_1) \cdot\cdot, \mathcal{F}(t,p,a_m),\mathcal{H}(t,p,a_m), \mathcal{D}(t,p,a_m)\}, \mathcal{C}'_i \in \{high, low\}$ and one decision attribute $m(t,p) \in \{m_1, m_2\}$ with $m_1, m_2 \in \{$"*Yes*", "*No*"$\}$. The values of attributes in $\mathcal{M}$ are converted into the fuzzy decision support matrix $\mathcal{M}_{\mathcal{F}}$ as follows:

$$\mathcal{M} = \begin{bmatrix} \mathcal{F}(t,p,a) & \mathcal{H}(t,p,a) & \mathcal{D}(t,p,a) & \cdots & m(t,p) \\ 0 & 0.1 & 0 & \cdots & m_1 \\ 0.3 & 0 & 0.7 & \cdots & m_1 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 0.1 & 0 & 0.1 & \cdots & m_2 \end{bmatrix} \to \mathcal{M}_{\mathcal{F}} = \begin{bmatrix} \mathcal{F}(t,p,a) & \mathcal{H}(t,p,a) & \mathcal{D}(t,p,a) & \cdots & m(t,p) \\ low & high & low & \cdots & m_1 \\ high & low & high & \cdots & m_1 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ high & low & high & \cdots & m_2 \end{bmatrix} \tag{37}$$

From that we build the fuzzy decision tree using the fuzzy decision support matrix $\mathcal{M}_{\mathcal{F}}$.

- **Step 2**: Build the fuzzy decision tree:

+ **Step 2.1**: Generate a root node r: For $\mathcal{C}'_i (i = 1,2,\ldots,m)$: calculate the information gains of the condition attributes $\mathcal{G}(\mathcal{C}'_i, M_F)$ to be described by Eq. 42 and choose the attribute with the maximum information gain (denoted by $\mathcal{C}'_{max}$) $\to r = \mathcal{C}'_{max}$.

+ **Step 2.2**: Divide $\mathcal{M}_{\mathcal{F}}$ into two fuzzy subsets $\mathcal{M}_{\mathcal{F}_1}$ and $\mathcal{M}_{\mathcal{F}_2}$ according to $\mathcal{C}'_{max}$.

+ **Step 2.3**: Generate new nodes r_1, r_2 for two fuzzy subsets $\mathcal{M}_{\mathcal{F}_1}$ and $\mathcal{M}_{\mathcal{F}_2}$, then calculate the information gains of the condition attributes $\mathcal{C}'_i$ that are not used by the root node r in the branching process and select the attribute with the maximum information gain as extended node, and label the fuzzy sets $\mathcal{F}\ddagger_{max,j}$ to edges that connect between the nodes r_j to r.

+ **Step 2.4**: Replace $\mathcal{M}_{\mathcal{F}}$ by $\mathcal{M}_{\mathcal{F}_1}, \mathcal{M}_{\mathcal{F}_2}$ and repeat from step 2 recursively until it is unavailable to branch.

5.6.2.2. The value of entropy and information gain is computed as follows. For the fuzzy decision support matrix $\mathcal{M}_{\mathcal{F}}$; $\mathcal{C}' = \{\mathcal{F}(t,p,a_1),\mathcal{H}(t,p,a_1),\mathcal{D}(t,p,a_1)\ldots,\mathcal{F}(t,p,a_m),\mathcal{H}(t,p,a_m),\mathcal{D}(t,p,a_m)\}$ be a set of condition attributes, where $\mathcal{C}'_i$ has fuzzy sets $\mathcal{F}\ddagger_{i1},\mathcal{F}\ddagger_{i2},\ldots,\mathcal{F}\ddagger_{ik},\mathcal{F}\ddagger_{ij}\in\{high,low\}, j=[1,k]$ (called the range of fuzzy sets); $\mathcal{D}=\{m_1,m_2\}$ be a set of decision attributes, in which m_1="*Yes*" and m_2="*No*" are the range of the decision attributes. Let the information gain or entropy need to classify a given tweet is calculated by:

$$\mathcal{I}(\mathcal{C}'_1,\ldots,\mathcal{C}'_n) = -\sum_{i=1}^{n} p_i \times log_2 p_i \tag{38}$$

where p_i is the probability that an arbitrary tweet belongs to class m_i and is estimated by summing the tweets' entropy. Let attribute $\mathcal{C}'_i$ have v distinct values $\{\mathcal{A}_1,\mathcal{A}_2,\ldots,\mathcal{A}_v\};\mathcal{C}'_i$ can be used to partition $\mathcal{T}$ into v subsets $\{\mathcal{T}_1,\mathcal{T}_2,..,\mathcal{T}_v\}$, where $\mathcal{T}_j(j=1,2,..,v)$ contains those tweets in $\mathcal{T}$ that have a value $\mathcal{A}_j$ of $\mathcal{C}'_i$. The entropy of attribute $\mathcal{C}'_i$ is given by

$$\mathcal{E}(\mathcal{C}'_i) = \sum_{j=1}^{v} p_{ij} \times \mathcal{I}(\mathcal{C}'_{1j},..,\mathcal{C}'_{mj}) \tag{39}$$

For a given subset $\mathcal{T}_j$, let $|\mathcal{T}_j|$ be the number of samples in $\mathcal{T}_j$, and $|\mathcal{C}'_{ij}|$ be the number of tweets of class m_i in subset $\mathcal{T}_j$, the information gain is expressed as

$$\mathcal{I}(\mathcal{C}'_{1j},..,\mathcal{C}'_{mj}) = -\sum_{i=1}^{m} p_{ij} \times log_2 p_{ij} \tag{40}$$

where

$$p_{ij} = \frac{|\mathcal{C}'_{ij}|}{|\mathcal{T}_j|} \tag{41}$$

Therefore, the information gain of attribute $\mathcal{C}'_i$ is given by

$$\mathcal{G}(\mathcal{C}'_i) = \mathcal{I}(\mathcal{C}'_{1j},..,\mathcal{C}'_{mj}) - \mathcal{E}(\mathcal{C}'_i) \tag{42}$$

5.6.2.3. Extracting classification rules from the fuzzy decision tree. The built fuzzy decision tree enables us to indicate the user's satisfaction with not only aspects that exist in the tweets but also with different aspects not exist in the tweets by using certain rules. These rules can help users in decision-making. Each rule (denoted by A→B) is featured by the support and confidence measures. The support measure is the percentage between tweets containing both A and B to the total of tweets. The confidence is the percentage between tweets including both A and B to the tweets containing only A. Rules with minimum support and maximum confidence are called "strong" or "good" rules.

5.7. Case study

To easier visualize the problem, the following examples are implemented and explained to illustrate one cycle of the presented approach. Given a set of tweets, $\mathcal{T}=\{t_1,t_2,t_3,t_4,t_5,t_6,t_7,t_8,t_9\}$, where.

t_1: The phone has a good color and I quite like its style.
t_2: The screen is so clearly but fairly small.
t_3: The battery is good, the screen is good, but the color is quite bad.
t_4: My father does not like the color of this phone.
t_5: I like the screen and battery, but the camera, style, and color are relatively familiar.
t_6: I like the battery and the camera.
t_7: The screen is so big, but the camera is quite clear.
t_8: The phone has a beautiful color, but the style is not lovely.
t_9: The color, camera, and style are good.

Example 2. From $\mathcal{T}$, aspects and type of phrases related to objects are determined as follows:

t_1 : a_1= color, cp_1= good; a_2= style, fp_1= quite like $\Rightarrow \mathcal{S}\rfloor(cp_1)$= 0.625, $\mathcal{S}\rfloor(fp_1)$ =0.375 + 0.25 = 0.625.
t_2 : a_1= screen, fp_1= so clearly; a_2= screen, fp_2= fairly small.
$\Rightarrow \mathcal{S}\rfloor(fp_1)$= 0.167 + 0.25 = 0.417, $\mathcal{S}\rfloor(fp_2)$=-0.396–0.25=-0.646.
t_3 : a_1= battery, cp_1= good; a_2= screen, cp_2= good; a_3= color, fp_1= quite bad.
$\Rightarrow \mathcal{S}\rfloor(cp_1)$= 0.625, $\mathcal{S}\rfloor(cp_2)$= 0.625,$\mathcal{S}\rfloor(fp_1)$=-0.542–0.25=-0.792.
t_4 : a_1= color, cp_1= not like $\Rightarrow \mathcal{S}\rfloor(cp_1)$= -0.375.
t_5 : a_1= screen, cp_1= like; a_2= battery, cp_2= like; a_3= camera, fp_1= relatively familiar; a_4= style, fp_2= relatively familiar; a_5= color, fp_3= relatively familiar.
$\Rightarrow \mathcal{S}\rfloor(cp_1) = \mathcal{S}\rfloor(cp_2)$=0.375, $\mathcal{S}\rfloor(fp_1) = \mathcal{S}\rfloor(fp_2) = \mathcal{S}\rfloor(fp_3)$ = 0.375 - 0.25 = 0.125.

$t_6 : a_1$= battery, cp_1= like; a_2= camera, cp_2= like $\Rightarrow \mathcal{S}\rfloor(cp_1)$=0.375, $\mathcal{S}\rfloor(cp_2)$=0.375.
$t_7 : a_1$= screen, fp_1= so big; a_2= camera, fp_2= quite clearly $\Rightarrow \mathcal{S}\rfloor(fp_1)$=-0.025–0.25=-0.275, $\mathcal{S}\rfloor(fp_2)$= 0.167 + 0.25 = 0.417.
$t_8 : a_1$= color, cp_1= beautiful; a_2= style, cp_2= not lovely $\Rightarrow \mathcal{S}\rfloor(cp_1)$=0.688, $\mathcal{S}\rfloor(cp_2)$=-0.458.
$t_9 : a_1$= color, cp_1= good; a_2= camera, cp_2= good; a_3= style, cp_3= good $\Rightarrow \mathcal{S}\rfloor(cp_1) = \mathcal{S}\rfloor(cp_2)$= $\mathcal{S}\rfloor(cp_3)$=0.625.

Example 3. Based on the result of Example 2, aspects in $\mathcal{T}$ including: *color*, *style*, *battery*, *screen*, *camera* are extracted. The sentiment of aspects in tweets is then identified as follows:

$t_1 : a_1$= color, cp_1= good $\in \mathcal{C}_{\surd}, \mathcal{S}\rfloor(cp_1)$=0.625 $\Rightarrow \mathcal{S}(t_1, p, a_1)$ = SP; and a_2= style, fp_1= quite like $\in \mathcal{F}_{\surd}, \mathcal{S}\rfloor(fp_1)$ =0.625 $\Rightarrow \mathcal{S}(t_1, p, a_2)$ = SP.

$t_2 : a_1$= screen, fp_1= so clearly $\in \mathcal{F}_{\surd}, \mathcal{S}\rfloor(fp_1)$=0.417 $\Rightarrow \mathcal{S}(t_2, p, a_1)$ = P; and a_2= screen, fp_2= too small $\in \mathcal{F}_{\surd}, \mathcal{S}\rfloor(fp_2)$=-0.646 $\Rightarrow \mathcal{S}(t_2, p, a_2)$ = SN.

$t_3 : a_1$= battery, cp_1= good $\in \mathcal{C}_{\surd}, \mathcal{S}\rfloor(cp_1)$= 0.625 $\Rightarrow \mathcal{S}(t_3, p, a_1)$ = SP; and a_2= screen, cp_2= good $\in \mathcal{C}_{\surd}, \mathcal{S}\rfloor(cp_2)$= 0.625 $\Rightarrow \mathcal{S}(t_3, p, a_2)$ = SP; and a_3= color, fp_1= quite bad $\in \mathcal{F}_{\surd}, \mathcal{S}\rfloor(fp_1)$=-0.792 $\Rightarrow \mathcal{S}(t_3, p, a_3)$ = SN.

$t_4 : a_1$= color, cp_1= do not like $\in \mathcal{C}_{\surd}, \mathcal{S}\rfloor(cp_1)$= $-0.375 \Rightarrow \mathcal{S}(t_4, p, a_1)$ = N.

$t_5 : a_1$= screen, cp_1= like $\in \mathcal{C}_{\surd}, \mathcal{S}\rfloor(cp_1)$= 0.375 $\Rightarrow \mathcal{S}(t_5, p, a_1)$ = P; and a_2= battery, cp_2= like $\in \mathcal{C}_{\surd}, \mathcal{S}\rfloor(cp_2)$= 0.375 $\Rightarrow \mathcal{S}(t_5, p, a_2)$ = P; and a_3= camera, fp_1= relatively familiar $\in \mathcal{F}_{\surd}, \mathcal{S}\rfloor(fp_1)$= 0.125 $\Rightarrow \mathcal{S}(t_5, p, a_3)$ = Ne; and a_4= style, fp_2= relatively familiar $\in \mathcal{F}_{\surd}, \mathcal{S}\rfloor(fp_2)$= 0.125 $\Rightarrow \mathcal{S}(t_5, p, a_4)$ = Ne; and a_5= color, fp_3= relatively familiar $\in \mathcal{F}_{\surd}, \mathcal{S}\rfloor(fp_3)$= 0.125 $\Rightarrow \mathcal{S}(t_5, p, a_5)$ = Ne.

$t_6 : a_1$= battery, cp_1= like $\in \mathcal{C}_{\surd}, \mathcal{S}\rfloor(cp_1)$=0.375 $\Rightarrow \mathcal{S}(t_6, p, a_1)$ = P; and a_2= camera, cp_2= like$\in \mathcal{C}_{\surd}, \mathcal{S}\rfloor(cp_2)$=0.375 $\Rightarrow \mathcal{S}(t_6, p, a_2)$ = P.

$t_7 : a_1$= screen, fp_1= so big $\in \mathcal{F}_{\surd}, \mathcal{S}\rfloor(fp_1)$= -0.025–0.25=-0.275 $\Rightarrow \mathcal{S}(t_7, p, a_1)$ = N; and a_2= camera, fp_2= quite clearly $\in \mathcal{F}_{\surd}, \mathcal{S}\rfloor(fp_2)$= 0.167 + 0.25 = 0.417 $\Rightarrow \mathcal{S}(t_7, p, a_2)$ = P.

$t_8 : a_1$= color, cp_1= beautiful $\in \mathcal{C}_{\surd}, \mathcal{S}\rfloor(cp_1)$=0.688 $\Rightarrow \mathcal{S}(t_8, p, a_1)$ = SP; and a_2= style, cp_2= not lovely $\in \mathcal{C}_{\surd}, \mathcal{S}\rfloor(cp_2)$=-0.458 $\Rightarrow \mathcal{S}(t_8, p, a_2)$ = N.

$t_9 : a_1$= color, cp_1= good $\in \mathcal{C}_{\surd}, \mathcal{S}\rfloor(cp_1)$= 0.625 $\Rightarrow \mathcal{S}(t_9, p, a_1)$ = SP; and a_2= camera, cp_2= good $\in \mathcal{C}_{\surd}, \mathcal{S}\rfloor(cp_2)$= 0.625 $\Rightarrow \mathcal{S}(t_9, p, a_2)$ = SP; and a_3= style, cp_3= good $\in \mathcal{C}_{\surd}, \mathcal{S}\rfloor(cp_3)$=0.625 $\Rightarrow \mathcal{S}(t_9, p, a_3)$ = SP.

Example 4. Using aspects and their sentiment extracted in Examples 2 and 3, the decision support matrix is built as the following steps:

- Step 1: The relation between aspect-based sentiment and user satisfaction in tweets is determined in Table 7.
- Step 2: The relation between user satisfaction and decision-making in tweets is identified as shown in Table 8.
- Step 3: The decision support matrix based on the user satisfaction level is shown in Table 9.

Example 5. Based on Example 4, the fuzzy decision tree is built as follows:

Let h and l corresponding to *"high"* and *"low"*, respectively. After fuzzing, the decision support matrix has the form as Table 10.

In the second step, from Table 10, there are five samples indicating the user's dissatisfaction with the phone (classified *"No"*) and four samples indicating the user's satisfaction (classified *"Yes"*). The values of entropy and information gain, for each attribute, are determined in Table 11.

From Table 11, because $\mathcal{D}(t, p, style)$ has the highest value of information gain among the attributes, $\mathcal{D}(t, p, style)$ is chosen as the root node of the tree. The same process is then done again for the other attributes (see Tables 12 and 13) until the nodes of the tree are clustered completely.

Table 12 shows the decision support matrix corresponding to $\mathcal{D}(t, p, style) = l$. After the $\mathcal{D}(t, p, style)$ is selected as the root node of the fuzzy decision tree, the samples containing $\mathcal{D}(t, p, style) = h$ is classified as *"No"*, and, thus, removed from the decision support matrix. Also, the leaf of node $\mathcal{D}(t, p, style)$ is node *"No"*.

From Table 13, because $\mathcal{H}(t, p, color)$ has the highest value of information gain among the other attributes, $\mathcal{H}(t, p, color)$ becomes the child node of node $\mathcal{D}(t, p, style)$. The same step is then done again for the other attributes (see Tables 14 and 15) until the nodes on the tree are clustered fully.

Table 14 shows the decision support matrix corresponding to $\mathcal{H}(t, p, color) = l$. After $\mathcal{H}(t, p, color)$ becomes the child of node $\mathcal{D}(t, p, style)$, samples containing $\mathcal{H}(t, p, color) = h$ is classified as *"No"*, and, thus, removed from the decision support matrix. Also, the leaf of node $\mathcal{H}(t, p, color)$ is node *"No"*.

Table 7
The relation between aspect-based sentiment and user satisfaction in tweets.

	a_1=color					a_2=screen					a_3=battery					a_4=style					a_5=camera				
	SP	P	Ne	N	SN	SP	P	Ne	N	SN	SP	P	Ne	N	SN	SP	P	Ne	N	SN	SP	P	Ne	N	SN
t_1	1	–	–	–	–	–	–	–	–	–	–	–	–	–	–	1	–	–	–	–	–	–	–	–	–
t_2	–	–	–	–	–	–	1	–	–	1	–	–	–	–	–	–	–	–	–	–	–	–	–	–	1
t_3	–	–	–	–	1	1	–	–	–	–	1	–	–	–	–	–	–	–	–	–	–	–	–	–	–
t_4	–	–	–	1	–	–	–	–	–	–	–	–	–	–	1	–	–	–	1	–	–	–	–	–	–
t_5	–	–	1	–	–	–	1	–	–	–	–	1	–	–	–	–	–	1	–	–	–	–	1	–	–
t_6	–	1	–	–	–	–	–	1	–	–	–	1	–	1	–	–	–	–	–	–	–	1	–	–	–
t_7	–	–	–	–	–	–	–	–	1	–	–	–	–	–	–	–	–	–	–	1	–	1	–	–	–
t_8	1	–	–	–	–	–	–	–	–	–	–	–	1	–	1	–	–	–	1	–	–	–	–	–	–
t_9	1	–	–	–	–	–	–	–	–	–	–	–	–	–	–	1	–	–	–	–	1	–	–	–	–
$\|S\|$	3	1	1	1	1	1	2	1	1	1	1	2	1	1	2	2	0	1	2	1	1	2	1	0	1
F(t,p,a)	0.081632653					0.1					0.085714286					0.055555556					0.12				
D(t,p,a)	0.042735043					0.066666667					0.085714286					0.083333333					0.04				
H(t,p,a)	0.020408163					0.033333333					0.028571429					0.027777778					0.04				

Table 8
The relation between the user satisfaction and the decision-making in tweets.

	$a_1 = color$			$a_2 = screen$			$a_3 = battery$			$a_4 = style$			$a_5 = camera$			$\mathcal{F}\ (p,t)$	$\mathcal{D}\ (p,t)$	$\mathcal{H}\ (p,t)$	$\mathcal{S}\rfloor\ (p,t)$
	$\mathcal{F}$	$\mathcal{H}$	$\mathcal{D}$	$\mathcal{F}$	$\mathcal{H}$	$\mathcal{D}$	$\mathcal{F}$	$\mathcal{H}$	$\mathcal{D}$	$\mathcal{F}$	$\mathcal{H}$	$\mathcal{D}$	$\mathcal{F}$	$\mathcal{H}$	$\mathcal{D}$				
t_1	.08	0	0	0	0	0	0	0	0	.06	0	0	0	0	0	.14	0	0	.14
t_2	0	0	0	.1	0	.07	.09	0	0	0	0	0	0	0	.04	.19	.11	0	.19
t_3	0	0	.04	.1	0	0	.09	0	0	0	0	0	0	0	0	.19	.04	0	.19
t_4	0	0	.04	0	0	0	0	0	.09	0	0	.08	0	0	0	0	.21	0	0
t_5	0	.02	0	.1	0	0	.09	0	0	0	.12	0	0	.04	0	.19	0	.09	.09
t_6	.08	0	0	0	.03	0	.09	0	.09	0	0	0	.12	0	0	.29	.09	.03	.32
t_7	0	0	0	0	0	.07	0	0	0	0	0	.08	.12	0	0	.12	.15	0	.12
t_8	.08	0	0	0	0	0	0	0	.09	0	0	.08	0	0	0	.08	.17	.03	.11
t_9	.08	0	0	0	0	0	0	0	0	.06	0	0	.12	0	0	.26	0	0	.26

Table 9
The decision support matrix.

	$a_1 = color$			$a_2 = screen$			$a_3 = battery$			$a_4 = style$			$a_5 = camera$			m
	$\mathcal{F}$	$\mathcal{H}$	$\mathcal{D}$	$\mathcal{F}$	$\mathcal{H}$	$\mathcal{D}$	$\mathcal{F}$	$\mathcal{H}$	$\mathcal{D}$	$\mathcal{F}$	$\mathcal{H}$	$\mathcal{D}$	$\mathcal{F}$	$\mathcal{H}$	$\mathcal{D}$	
t_1	.08	0	0	0	0	0	0	0	0	.06	0	0	0	0	0	No
t_2	0	0	0	.1	0	.07	.09	0	0	0	0	0	0	0	.04	Yes
t_3	0	0	.04	.1	0	0	.09	0	0	0	0	0	0	0	0	Yes
t_4	0	0	.04	0	0	0	0	0	.09	0	0	.08	0	0	0	No
t_5	0	.02	0	.1	0	0	.09	0	0	0	.12	0	0	.04	0	No
t_6	.08	0	0	0	.03	0	.09	0	.09	0	0	0	.12	0	0	Yes
t_7	0	0	0	0	0	.07	0	0	0	0	0	.08	.12	0	0	No
t_8	.08	0	0	0	0	0	0	0	.09	0	0	.08	0	0	0	No
t_9	.08	0	0	0	0	0	0	0	0	.06	0	0	.12	0	0	Yes

Table 10
The fuzzified version of the decision support matrix.

	a_1=color			a_2=screen			a_3=battery			a_4=style			a_5=camera			m
	$\mathcal{F}$	$\mathcal{H}$	$\mathcal{D}$	$\mathcal{F}$	$\mathcal{H}$	$\mathcal{D}$	$\mathcal{F}$	$\mathcal{H}$	$\mathcal{D}$	$\mathcal{F}$	$\mathcal{H}$	$\mathcal{D}$	$\mathcal{F}$	$\mathcal{H}$	$\mathcal{D}$	
t_1	h	l	l	l	l	l	l	l	l	h	l	l	l	l	l	No
t_2	l	l	l	h	l	h	h	l	l	l	l	l	l	l	h	Yes
t_3	l	l	h	h	l	l	h	l	l	l	l	l	l	l	l	Yes
t_4	l	l	h	l	l	l	l	l	h	l	l	h	l	l	l	No
t_5	l	h	l	h	l	l	h	l	l	l	h	l	l	h	l	No
t_6	h	l	l	l	h	l	h	l	h	l	l	l	h	l	l	Yes
t_7	l	l	l	l	l	h	l	l	l	l	l	h	h	l	l	No
t_8	h	l	l	l	l	l	l	l	h	l	l	h	l	l	l	No
t_9	h	l	l	l	l	l	l	l	l	h	l	l	h	l	l	Yes

Table 11
The information to choose the root node.

	a_1=color						a_2=screen						a_3=battery						a_4=style						a_5=camera					
$\mathcal{E}$ =.99	$\mathcal{F}$		$\mathcal{H}$		$\mathcal{D}$		$\mathcal{F}$		$\mathcal{H}$		$\mathcal{D}$		$\mathcal{F}$		$\mathcal{H}$		$\mathcal{D}$		$\mathcal{F}$		$\mathcal{H}$		$\mathcal{D}$		$\mathcal{F}$		$\mathcal{H}$		$\mathcal{D}$	
	h	l	h	l	h	l	h	l	h	l	h	l	h	l	h	l	h	l	h	l	h	l	h	l	h	l	h	l	h	l
Yes	2	2	0	4	1	3	2	2	1	3	1	3	3	1	0	4	1	3	1	3	0	4	0	4	2	2	0	4	1	3
No	2	3	1	4	1	4	1	4	0	5	1	4	1	4	1	4	2	3	1	4	1	4	3	2	1	4	1	4	0	5
I	1	.97	0	1	1	.99	.92	.92	0	.95	1	.99	.81	.72	0	1	.92	1	1	.99	0	1	0	.92	.92	.92	0	1	0	.95
$\mathcal{E}$	.98		.89		.99		.92		.84		.99		.76		.89		.97		.99		.89		.61		.92		.89		.84	
G	.01		.1		0		.07		.15		0		.23		.1		.02		0		.1		.38		.07		.1		.15	

Table 12
The decision support matrix after choosing the root node.

	a_1=color			a_2=screen			a_3=battery			a_4=style			a_5=camera			m
	$\mathcal{F}$	$\mathcal{H}$	$\mathcal{D}$	$\mathcal{F}$	$\mathcal{H}$	$\mathcal{D}$	$\mathcal{F}$	$\mathcal{H}$	$\mathcal{D}$	$\mathcal{F}$	$\mathcal{H}$	$\mathcal{D}$	$\mathcal{F}$	$\mathcal{H}$	$\mathcal{D}$	
t_1	h	l	l	l	l	l	l	l	l	h	l	l	l	l	l	No
t_2	l	l	l	h	l	h	h	l	l	l	l	l	l	l	h	Yes
t_3	l	l	h	h	l	l	h	l	l	l	l	l	l	l	l	Yes
t_5	l	h	l	h	l	l	h	l	l	l	h	l	l	h	l	No
t_6	h	l	l	l	h	l	h	l	h	l	l	l	h	l	l	Yes
t_9	h	l	l	l	l	l	l	l	l	h	l	l	h	l	l	Yes

From Table 15, because $\mathcal{F}(t,p,battery)$ has the highest value of information gain among the attributes, $\mathcal{F}(t,p,battery)$ becomes the child node of node $\mathcal{H}(t,p,color)$. The same process is then done again for the other attributes (see Tables 14 and 15) until the nodes of the tree are grouped completely.

Table 16 shows the decision support matrix corresponding to $\mathcal{F}(t,p,battery) = l$. After the $\mathcal{F}(t,p,battery)$ becomes the child of node $\mathcal{H}(t,p,color)$, samples containing $\mathcal{F}(t,p,battery) = h$ is classified as *"Yes"*, and, thus, removed from the decision support matrix. Also, the leaf of node $\mathcal{F}(t,p,battery)$ is node *"No"*.

From Table 17, because $\mathcal{F}(t,p,camera)$ has the highest value of information gain among the attributes, $\mathcal{F}(t,p,camera)$ is selected the child node of node $\mathcal{F}(t,p,battery)$. At node $\mathcal{F}(t,p,camera)$, we do not need for further division, because its two child nodes are grouped fully. The final fuzzy decision tree is shown in Fig. 4.

Example 6. Based on the fuzzy decision tree in Fig. 4, five rules are extracted from this tree as follows:

Rule #1: $\mathcal{D}(p,t,style) = h \rightarrow No.$

Rule #2: $\mathcal{D}(p,t,style) = l \wedge \mathcal{H}(p,t,color) = h \rightarrow No.$

Rule #3: $\mathcal{D}(p,t,style) = l \wedge \mathcal{H}(p,t,color) = l \wedge \mathcal{F}(p,t,battery) = h \rightarrow Yes.$

Rule #4: $\mathcal{D}(p,t,style) = l \wedge \mathcal{H}(p,t,color) = l \wedge \mathcal{F}(p,t,battery) = l \wedge \mathcal{F}(p,t,camera) = h \rightarrow Yes.$

Rule #5: $\mathcal{D}(p,t,style) = l \wedge \mathcal{H}(p,t,color) = l \wedge \mathcal{F}(p,t,battery) = l \wedge \mathcal{F}(p,t,camera) = l \rightarrow No.$

From these rules, when a user needs a new phone, this user needs only comment on the aspects of the phone. Our method can help the user who makes a more reasonable purchasing decision. The problem is how to apply these rules to help users in deciding. Assume a user (called Mr. A) saw a new phone, and Mr. A gave the following opinion: *"The screen is beautiful, but the style and color are not good*". How should Mr. A decision-making? The five rules created from the fuzzy decision tree are used to help Mr. A here. In this example, three aspects are given in the user's opinion as follows:

$t : a_1$= screen, cp_1= beautiful $\Rightarrow \mathcal{S}\rfloor(cp_1) = 0.688 \Rightarrow \mathcal{S}(t,p.a_1)$ = SP;

a_2= style, cp_2= not good $\Rightarrow \mathcal{S}\rfloor(cp_2)$=-0.625 $\Rightarrow \mathcal{S}(t,p.a_2)$ = SN;

a_3= color, cp_3= not good $\Rightarrow \mathcal{S}\rfloor(cp_3)$=-0.625 $\Rightarrow \mathcal{S}(t,p.a_3)$ = SN.

Thus, we have $\mathcal{F}(p,t,screen) = h, \mathcal{D}(p,t,style) = h$, and $\mathcal{D}(p,t,color) = h$. Therefore, Rule#1 is applied, whose result is *"No"* with the support value of 3 and the confidence value of 3/3. Consequence, the user should not buy this phone.

6. Experiment

6.1. Data acquisition

For the need of the experiments, we have collected 26,750 English tweets regarding Phone topic, which contain at least one sentiment word, by using the most popular Twitter streaming API,[7] namely the Python package Tweepy.[8] These tweets were gathered by hashtags related to phones, such as *#phone, #samsung, #iphone, #nokia, #huawei, #galaxy*. Then, the non-necessary factors in tweets, such as punctuation, retweet marks, URLs, hashtags, and query terms were discarded. The Python emoji package[9] was used to replace each emoji with a descriptive text, respectively. Tweets often include acronyms, spelling errors, symbols. It was then necessary to correct them. We fixed these spellings by using the Python-based Aspell library.[10] Then, the tweets were grouped into two different files, called training and testing datasets by using the 5-fold cross-validation technique. That means the tweets were merged and shuffled (using the same random seed for all runs), and we used 5-fold cross-validation to evaluate the method's performance. In 5-fold cross-validation, the model was implemented five times, and the tweets were divided into five distinct partitions. For each time, one part in five distinct partitions (20% of tweets) was selected as the test set, and the four remaining parts were used as the training set; the evaluation continues until each of the five distinct parts was used as the test set. Therefore, for each fold, the training dataset contained 21,400 tweets, whereas the testing dataset contained 5350 tweets. The reported performance was calculated as the average result when applying the model to five

7 https://developer.twitter.com/en/docs/twitter-api/v1.
8 https://pypi.org/project/tweepy/.
9 https://pypi.org/project/emoji/
10 https://pypi.org/project/aspell-python-py2/

Table 13
The information to choose the child of root node.

	a_1=color						a_2=screen						a_3=battery						a_4=style						a_5=camera					
$\mathcal{E}$ =.61	$\mathcal{F}$		$\mathcal{H}$		$\mathcal{D}$		$\mathcal{F}$		$\mathcal{H}$		$\mathcal{D}$		$\mathcal{F}$		$\mathcal{H}$		$\mathcal{D}$		$\mathcal{F}$		$\mathcal{H}$		$\mathcal{D}$		$\mathcal{F}$		$\mathcal{H}$		$\mathcal{D}$	
	h	l	h	l	h	l	h	l	h	l	h	l	h	l	h	l	h	l	h	l	h	l	h	l	h	l	h	l	h	l
Yes	2	2	0	4	1	3	2	2	1	3	1	3	3	1	0	4	1	3	1	3	0	4	–	–	2	2	0	4	1	3
No	1	1	1	1	0	2	1	1	0	2	0	2	1	1	0	2	0	1	1	1	1	1	–	–	0	0	1	1	0	2
I	.92	.92	0	.72	0	.97	.92	.92	0	.97	0	.97	.81	1	–	.92	0	.81	1	.81	0	.72	–	–	0	1	0	.72	0	.97
E	.92		0.6		.81		.92		.81		.81		.87		.92		0.65		.87		.6		–		.67		.6		.81	
G	−.31		.01		−.2		−.31		−.2		−.2		−.26		−.31		−.04		−.26		.01		–		−.06		.01		−.2	

Table 14
The decision support matrix after choosing the child of root node.

	a_1=color			a_2=screen			a_3=battery			a_4=style			a_5=camera			m
	$\mathcal{F}$	$\mathcal{H}$	$\mathcal{D}$	$\mathcal{F}$	$\mathcal{H}$	$\mathcal{D}$	$\mathcal{F}$	$\mathcal{H}$	$\mathcal{D}$	$\mathcal{F}$	$\mathcal{H}$	$\mathcal{D}$	$\mathcal{F}$	$\mathcal{H}$	$\mathcal{D}$	
t_1	h	l	l	l	l	l	l	l	l	h	l	l	l	l	l	No
t_2	l	l	l	h	l	h	h	l	l	l	l	l	l	l	h	Yes
t_3	l	l	h	h	l	l	h	l	l	l	l	l	l	l	l	Yes
t_6	h	l	l	l	h	l	h	l	h	l	l	l	h	l	l	Yes
t_9	h	l	l	l	l	l	l	l	l	h	l	l	h	l	l	Yes

Table 15
The information to choose the continuous node.

	a_1=color						a_2=screen						a_3=battery						a_4=style						a_5=camera					
$\mathcal{E}$ =.6	$\mathcal{F}$		$\mathcal{H}$		$\mathcal{D}$		$\mathcal{F}$		$\mathcal{H}$		$\mathcal{D}$		$\mathcal{F}$		$\mathcal{H}$		$\mathcal{D}$		$\mathcal{F}$		$\mathcal{H}$		$\mathcal{D}$		$\mathcal{F}$		$\mathcal{H}$		$\mathcal{D}$	
	h	l	h	l	h	l	h	l	h	l	h	l	h	l	h	l	h	l	h	l	h	l	h	l	h	l	h	l	h	l
Yes	2	2	–	–	1	3	2	2	1	3	1	3	3	1	0	4	1	3	1	3	0	4	–	–	2	2	0	4	1	3
No	1	0	–	–	0	1	0	1	0	1	0	1	0	1	0	1	0	1	1	0	0	1	–	–	0	1	0	1	0	1
I	.92	0	–	–	0	.81	0	.92	0	.81	0	.81	0	1		.72	0	.81	1	0	–	.72	–	–	0	.92	–	.72	0	.81
E	.55		–		.65		.55		.65		.65		.4		.72		.65		.4		.72		–		.55		.72		.65	
G	.05		–		–.05		.05		–.05		–.05		.2		–.12		–.05		.2		–.12		–		.05		–.12		–.05	

Table 16
The decision support matrix after choosing the child of root node.

	a_1=color			a_2=screen			a_3=battery			a_4=style			a_5=camera			m
	$\mathcal{F}$	$\mathcal{H}$	$\mathcal{D}$	$\mathcal{F}$	$\mathcal{H}$	$\mathcal{D}$	$\mathcal{F}$	$\mathcal{H}$	$\mathcal{D}$	$\mathcal{F}$	$\mathcal{H}$	$\mathcal{D}$	$\mathcal{F}$	$\mathcal{H}$	$\mathcal{D}$	
t_1	h	l	l	l	l	l	l	l	l	h	l	l	l	l	l	No
t_9	h	l	l	l	l	l	l	l	l	h	l	l	h	l	l	Yes

Table 17
The information to choose the continuous node.

	a_1=color						a_2=screen						a_3=battery						a_4=style						a_5=camera					
$\mathcal{E}$ =.4	$\mathcal{F}$		$\mathcal{H}$		$\mathcal{D}$		$\mathcal{F}$		$\mathcal{H}$		$\mathcal{D}$		$\mathcal{F}$		$\mathcal{H}$		$\mathcal{D}$		$\mathcal{F}$		$\mathcal{H}$		$\mathcal{D}$		$\mathcal{F}$		$\mathcal{H}$		$\mathcal{D}$	
	h	l	h	l	h	l	h	l	h	l	h	l	h	l	h	l	h	l	h	l	h	l	h	l	h	l	h	l	h	l
Yes	1	0	–	–	0	1	0	1	0	1	0	1	–	–	0	1	0	1	1	0	0	1	–	–	1	0	0	1	0	1
No	1	0	–	–	0	1	0	1	0	1	0	1	–	–	0	1	0	1	1	0	0	1	–	–	0	1	0	1	0	1
I	1	–	–	–	–	1	–	1	–	1	–	1	–	–	–	1	–	1	1	–	–	1	–	–	0	0	–	1	–	1
E	1		–		1		1		1		1		–		1		1		1		1		–		0		1		1	
G	–.6		–		–.6		–.6		–.6		–.6		–		–.6		–.6		–.6		–.6		–		.4		–.6		–.6	

distinct partitions. The training file included two tab-separated columns, with each token on a separate line. On each line, a token is expressed in the first item, whereas its label is shown in the second item. Each token in tweets were assigned one of seven labels (*Strong positive, Positive, Neutral, Negative, Strong negative, Aspect*, and *Other*) by five manual annotators. The tokens that did not belong to factors of interest were annotated as *"Other*". We also labeled the test set as the gold standard to evaluate the efficiency. The details of these datasets are shown in Table 18.

6.2. Evaluation results

Our proposal is evaluated based on metrics, namely *precision* ($\mathcal{P}$), *recall* ($\mathcal{R}$), *and* $\mathcal{F}_1$. The values of $\mathcal{P}, \mathcal{R}$, and $\mathcal{F}_1$ are computed as follows:

$$\mathcal{P} = \frac{TP}{TP + FP} \tag{43}$$

$$\mathcal{R} = \frac{TP}{TP + FN} \tag{44}$$

$$\mathcal{F}_1 = 2 \times \frac{\mathcal{P} \times \mathcal{R}}{\mathcal{P} + \mathcal{R}} \tag{45}$$

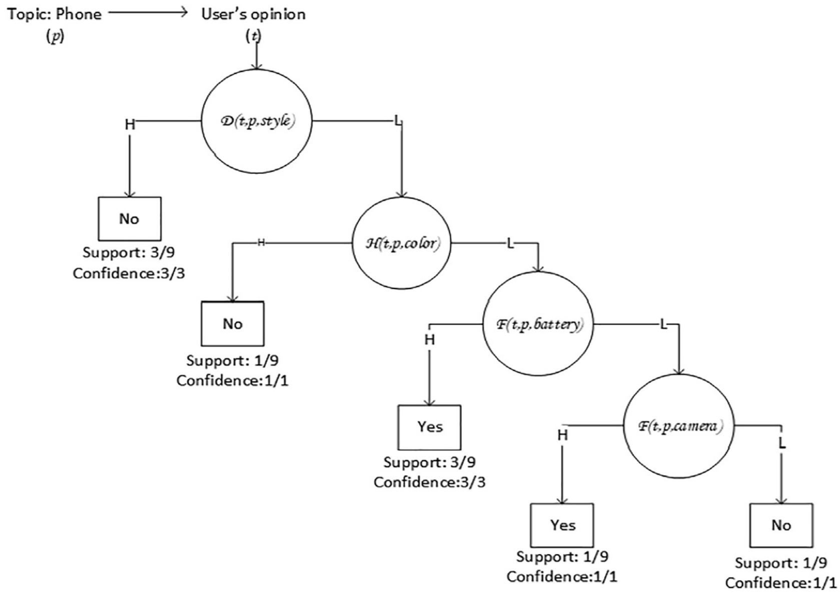

Fig. 4. Example of the fuzzy decision tree. (H is *high* (h), L is *low* (l))

Table 18
Statistics of datasets.

	Training set	Testing set
#Tweets classified *"Yes"*	11,224	3283
#Tweets classified *"No"*	10,176	2067
#Words indicating Aspects	47,780	8560
#Aspects assigned SP sentiment	6534	1181
#Aspects assigned P sentiment	12,638	2276
#Aspects assigned Ne sentiment	12,181	1964
#Aspects assigned N sentiment	10,169	2183
#Aspects assigned SN sentiment	6858	956

where SP = Strong positive, SN = Strong negative,
P = Positive, N = Negative, Ne = Neutral.

Table 19
The confusion matrix of detecting aspects.

↓ →	Screen	Battery	Ram	Rom	Camera	Sound	Style	Color	Application	Software
Screen	396	16	16	14	25	24	11	13	14	11
Battery	12	874	11	42	9	7	12	9	16	8
Ram	10	35	582	31	10	10	11	13	11	11
Rom	17	15	41	871	6	11	12	8	13	10
Camera	49	6	11	7	803	9	9	3	10	6
Sound	9	12	17	10	8	638	8	6	34	25
Style	10	11	7	7	9	6	892	20	11	9
Color	40	16	14	12	11	14	13	950	9	11
Application	14	15	11	7	16	16	11	10	597	20
Software	10	21	16	6	6	17	9	8	39	691

↓: Actual →: Predicted.

where, *TP* is the number of correctly identified elements; *FP* is the number of not correctly identified elements; *FN* is the number of misclassified non-elements.

6.3. Result and discussion

To prove that the performance of our proposal is better than other methods, we have implemented two more baseline methods to compare fairly. The first baseline (called the Baseline 1 method) is described in [37]. The method we propose

Table 20
The performance of aspects determination compare to the baseline methods.

Aspect	Baseline 1 method			Baseline 2 method			Proposed method		
	$\mathcal{P}$	$\mathcal{R}$	$\mathcal{F}_1$	$\mathcal{P}$	$\mathcal{R}$	$\mathcal{F}_1$	$\mathcal{P}$	$\mathcal{R}$	$\mathcal{F}_1$
Screen	0.75	0.78	0.76	0.68	0.55	0.61	0.70	0.73	0.72
Battery	0.82	0.78	0.80	0.76	0.78	0.77	0.86	0.87	0.87
Ram	0.75	0.78	0.76	0.64	0.72	0.68	0.80	0.80	0.80
Rom	0.79	0.81	0.80	0.68	0.69	0.68	0.86	0.87	0.87
Camera	0.83	0.82	0.82	0.78	0.80	0.79	0.89	0.88	0.88
Sound	0.73	0.76	0.74	0.76	0.77	0.76	0.85	0.83	0.84
Style	0.87	0.85	0.86	0.83	0.84	0.83	0.90	0.91	0.91
Color	0.82	0.79	0.80	0.70	0.72	0.71	0.91	0.87	0.89
Application	0.75	0.76	0.75	0.69	0.71	0.70	0.79	0.83	0.81
Software	0.81	0.83	0.82	0.82	0.80	0.81	0.86	0.84	0.85

in this paper is the improved version of this one presented in our conference paper [37]. The second baseline (called the Baseline 2 method) is introduced in [46]. We present the advantages and disadvantages of these methods in Section 2. The results of the aspects detection step of our proposal are shown in Tables 19 and 20. The ten high-frequency aspects are extracted including *screen, battery, ram, rom, camera, sound, style, color, application, software.*

Table 19 shows the confusion matrix of the proposed method. We can see that the distribution of tweets containing aspects of the "*phone*" topic in the dataset is not balanced. The confusion appears in the labeling of aspects in the entire dataset. For instance, with "*screen*", there are 16 cases misassigned to "*battery*", 16 cases misassigned to "*ram*", 14 cases misassigned to "*rom*", and 25 cases misassigned to "*camera*" and so on. Generally, the percentage of cases with misclassified aspects is 14.8%. According to our assessment, this performance could be achieved because the BiLSTM-CRF model takes advantages of both BiLSTM and CRF models in detecting aspects and the sentiment of aspects. Also, the word presentation method also helps more accurately identify the position of words that specify aspects and words expressing the sentiment of aspects.

Table 20 shows the performance of aspects detection of our method in comparison with the baseline methods. Specifically, to determine aspects, our method uses the BiLSTM-CRF model (combined with the improved word representation model), the Baseline 1 method also uses the BiLSTM-CRF model (not combined with the improved word representation model), and the Baseline 2 method does not use both the BiLSTM-CRF model and the improved word representation model. As we can note, our approach improves the efficiency of the Baseline 1 method by up to 10% for "*sound*" and by least at 3% for "*software*". However, the performance of our method is worse than the Baseline 1 method for "*screen*" (by 4%). In general, although both approaches (the Baseline 1 method and our method) use the BiLSTM-CRF model but our proposal provides better results than the Baseline 1 method (by 5.3%). This result shows that it succeeds in overcoming the disadvantages of the mentioned methods. This is because the word representation step of our method is better implemented. This also proves that using much useful information regarding words when implementing the word representation process should help in achieving better performance. In comparison to the Baseline 2 method, our method can improve the average F_1 score by 11%. According to our assessment, one of the main reason of this fact is related to combining of the BiLSTM-CRF method and the improved word representation model.

Let SP, P, Ne, N, and SN be the abbreviations of *Strong positive, Positive, Neutral, Negative*, and *Strong negative*, respectively. In Table 21, the distribution of sentiments in tweets is not balanced. Confusion often appears among two groups of labels such as (SP, P, and Ne) or (SN, N, and Ne). For example, 142 cases of aspects are misassigned from P to SP and 177 cases of aspects are misclassified from Ne to SP. Generally, 16.2% of cases misclassified their sentiment. There is no misclassified in the remaining cases. This is because some tweets in the training data are not assigned with correct labels or features have an unclear difference. Besides, the marks to distinguish among these sentiments may not be clear.

On the basis of Table 21 we have generated the data in Table 22, it is observed that the P and N classes have been implemented better than the remaining ones. Intuitively, the performance is low because training data contains very few words that indicate strong negative and negative sentiments. We believe that with the construction of a large dataset and the number of words indicating relevant factors, such as sentiment and aspects, is more balanced, then the performance should be notably increased.

Due to space limitations, the aspects are denoted as $a_j, (j = 1, \ldots, 10)$, respectively. The confusion matrix of aspect-based sentiment detection in specific cases is illustrated in Table 23.

Looking at the results in Table 23, it can be seen that the distribution among sentiments of aspects in the dataset is quite balanced. The confusion can still occur among all the sentiment aspects. For instance, with strong positive sentiment, there are 14 cases of the screen's sentiment misassigned to the sentiment of other aspects, in which there are two cases of battery, one case of "*ram*", three cases of "*rom*", three cases of "*camera*", and two cases of "*software*", and so on. Also, determining the sentiment of the specific aspects generally yields a good result for classifying aspects such as "*battery*", "*style*", "*color*", and "*screen*". For instance, there are only 1279 cases of misassigned sentiment labels (by 14.94%). However, the classification performance of the sentiment of the remaining aspects is not as desirable.

Table 21
The confusion matrix of detecting sentiments of aspects.

↓ →	SP	P	Ne	N	SN
SP	22	142	177	0	0
P	164	1994	118	0	0
Ne	65	64	1708	70	57
N	0	0	125	1962	96
SN	0	0	81	94	781

↓: Actual →: Predicted.

Table 22
The general performance of aspect-based sentiment analysis.

	SP	P	Ne	N	SN
Actual	1181	2276	1965	2183	956
Predicted	1091	2200	2209	2126	934
TP	862	1994	1708	1962	781
FP	229	206	501	164	153
FN	319	282	256	221	175
P	0.79	0.91	0.77	0.92	0.84
R	0.73	0.88	0.87	0.90	0.82
F_1	0.76	0.89	0.82	0.91	0.83

Table 23
The confusion matrices of aspect-based sentiment detection in specific cases. ↓: actual, →: predicted.

					SP											P					
↓ →	a_1	a_2	a_3	a_4	a_5	a_6	a_7	a_8	a_9	a_{10}	↓ →	a_1	a_2	a_3	a_4	a_5	a_6	a_7	a_8	a_9	a_{10}
a_1	123	2	1	3	3	2	2	1	0	2	a_1	125	2	2	3	7	3	2	2	3	2
a_3	0	121	1	3	0	0	4	0	2	3	a_3	2	221	5	2	2	3	4	3	5	3
a_3	2	0	74	10	4	6	2	3	4	0	a_3	7	5	199	18	5	4	4	5	2	3
a_4	2	2	12	82	4	2	3	1	3	4	a_4	8	3	15	142	5	5	5	7	7	2
a_5	2	1	0	1	119	2	2	3	4	0	a_5	6	5	3	1	254	7	3	3	4	5
a_6	2	0	1	1	2	85	0	5	3	2	a_6	2	2	3	4	3	238	3	3	3	2
a_7	0	3	1	2	1	1	127	1	0	1	a_7	3	4	5	4	4	1	302	4	3	3
a_8	3	2	1	4	3	3	2	105	3	0	a_8	3	2	1	4	3	3	2	211	3	3
a_9	0	0	4	3	2	3	1	3	89	2	a_9	1	3	3	2	3	3	1	2	189	2
a_{10}	2	1	3	3	1	2	0	3	3	65	a_{10}	2	2	4	1	2	2	1	3	1	75
					SN											N					
↓ →	a_1	a_2	a_3	a_4	a_5	a_6	a_7	a_8	a_9	a_{10}	↓ →	a_1	a_2	a_3	a_4	a_5	a_6	a_7	a_8	a_9	a_{10}
a_1	104	3	4	4	3	3	2	2	3	4	a_1	198	4	2	3	5	2	3	1	5	3
a_3	2	91	1	0	1	1	2	1	2	1	a_3	4	215	2	2	2	2	1	6	3	1
a_3	1	1	76	2	2	3	1	3	2	1	a_3	9	5	157	2	11	6	3	9	8	7
a_4	3	2	0	85	2	5	4	1	1	2	a_4	2	9	10	178	3	3	4	4	11	3
a_5	0	0	3	1	93	1	1	2	0	1	a_5	5	2	2	6	218	3	2	3	5	5
a_6	2	3	3	1	1	87	1	4	0	0	a_6	4	4	3	5	3	173	3	2	3	3
a_7	1	1	3	1	1	1	60	1	1	3	a_7	1	3	4	2	7	3	201	1	2	2
a_8	2	3	2	1	2	2	1	79	2	1	a_8	2	3	3	2	3	3	3	196	6	3
a_9	2	1	0	1	1	3	1	1	62	5	a_9	2	6	7	2	5	7	7	3	139	2
a_{10}	0	1	4	1	1	1	2	2	1	63	a_{10}	1	2	5	9	4	4	3	2	7	154

Ne										
↓ →	a_1	a_2	a_3	a_4	a_5	a_6	a_7	a_8	a_9	a_{10}
a_1	212	5	3	5	4	3	3	5	5	3
a_2	2	140	2	2	3	5	2	3	4	2
a_3	5	2	169	4	6	3	5	4	2	2
a_4	3	1	3	172	3	3	3	2	1	3
a_5	4	5	2	3	251	3	3	3	4	1
a_6	1	3	2	2	1	175	1	2	3	2
a_7	3	3	5	4	3	4	146	4	1	1
a_8	3	5	2	1	2	2	4	171	1	1
a_9	3	2	3	3	7	3	2	3	131	3
a_{10}	1	3	4	2	1	2	4	1	5	139

Using the confusion matrices in Table 23, the performance of aspect's sentiment detection in specific cases is calculated (see Table 24). Here we can note that the F_1 score of "*application*", "*rom*", "*ram*", "*sound*", and "*software*" is lower than the F_1 score of the remaining ones by up to 6%. Intuitively, the efficiency is low because the training data contains a few words that

Table 24
The performance of aspect-based sentiment detection in specific cases of the proposed method.

	SP			P			Ne			N			SN		
	$\mathcal{P}$	$\mathcal{R}$	$\mathcal{F}_1$	$\mathcal{P}$	$\mathcal{R}$	$\mathcal{F}_1$	$\mathcal{P}$	$\mathcal{R}$	$\mathcal{F}_1$	$\mathcal{P}$	$\mathcal{R}$	$\mathcal{F}_1$	$\mathcal{P}$	$\mathcal{R}$	$\mathcal{F}_1$
Screen	0.90	0.89	0.89	0.79	0.83	0.81	0.90	0.86	0.88	0.87	0.88	0.87	0.89	0.79	0.84
Battery	0.93	0.90	0.91	0.88	0.88	0.88	0.83	0.85	0.84	0.85	0.90	0.87	0.86	0.89	0.87
Ram	0.75	0.71	0.73	0.83	0.79	0.81	0.87	0.85	0.86	0.81	0.74	0.77	0.79	0.83	0.81
Rom	0.73	0.71	0.72	0.79	0.71	0.75	0.87	0.89	0.88	0.84	0.78	0.81	0.88	0.81	0.84
Camera	0.86	0.89	0.87	0.88	0.87	0.87	0.89	0.90	0.89	0.84	0.86	0.85	0.87	0.91	0.89
Sound	0.80	0.84	0.82	0.89	0.91	0.90	0.88	0.91	0.89	0.84	0.86	0.85	0.81	0.85	0.83
Style	0.89	0.93	0.91	0.92	0.91	0.91	0.85	0.84	0.84	0.87	0.89	0.88	0.80	0.82	0.81
Color	0.84	0.83	0.83	0.87	0.90	0.88	0.86	0.90	0.88	0.86	0.88	0.87	0.82	0.83	0.82
Application	0.80	0.83	0.81	0.86	0.90	0.88	0.83	0.82	0.82	0.74	0.77	0.75	0.84	0.81	0.82
Software	0.82	0.78	0.80	0.75	0.81	0.78	0.89	0.86	0.87	0.84	0.81	0.82	0.78	0.83	0.80

Table 25
The performance of aspect-based sentiment detection in specific cases of the baseline 1 method.

	SP			P			Ne			N			SN		
	$\mathcal{P}$	$\mathcal{R}$	$\mathcal{F}_1$	$\mathcal{P}$	$\mathcal{R}$	$\mathcal{F}_1$	$\mathcal{P}$	$\mathcal{R}$	$\mathcal{F}_1$	$\mathcal{P}$	$\mathcal{R}$	$\mathcal{F}_1$	$\mathcal{P}$	$\mathcal{R}$	$\mathcal{F}_1$
Screen	0.81	0.76	0.78	0.87	0.88	0.87	0.87	0.79	0.77	0.76	0.79	0.77	0.83	0.85	0.84
Battery	0.82	0.78	0.80	0.79	0.82	0.80	0.87	0.85	0.86	0.82	0.85	0.83	0.79	0.75	0.77
Ram	0.79	0.80	0.79	0.83	0.85	0.84	0.87	0.83	0.82	0.77	0.75	0.76	0.82	0.83	0.82
Rom	0.78	0.82	0.80	0.84	0.81	0.82	0.89	0.85	0.83	0.78	0.81	0.79	0.78	0.80	0.79
Camera	0.85	0.86	0.85	0.89	0.91	0.90	0.85	0.84	0.87	0.84	0.86	0.85	0.83	0.82	0.82
Sound	0.83	0.81	0.82	0.82	0.82	0.82	0.88	0.86	0.84	0.90	0.91	0.90	0.83	0.79	0.81
Style	0.81	0.80	0.80	0.84	0.83	0.83	0.87	0.82	0.83	0.89	0.88	0.88	0.87	0.88	0.87
Color	0.86	0.85	0.85	0.87	0.88	0.87	0.90	0.90	0.88	0.91	0.92	0.91	0.88	0.90	0.89
Application	0.82	0.81	0.81	0.83	0.81	0.82	0.82	0.81	0.79	0.84	0.78	0.81	0.78	0.81	0.79
Software	0.85	0.82	0.83	0.82	0.84	0.83	0.86	0.84	0.81	0.76	0.75	0.75	0.77	0.80	0.78

Table 26
The performance of aspect-based sentiment detection in specific cases of the baseline 2 method.

	SP			P			Ne			N			SN		
	$\mathcal{P}$	$\mathcal{R}$	$\mathcal{F}_1$	$\mathcal{P}$	$\mathcal{R}$	$\mathcal{F}_1$	$\mathcal{P}$	$\mathcal{R}$	$\mathcal{F}_1$	$\mathcal{P}$	$\mathcal{R}$	$\mathcal{F}_1$	$\mathcal{P}$	$\mathcal{R}$	$\mathcal{F}_1$
Screen	0.72	0.70	0.71	0.82	0.80	0.81	0.77	0.78	0.77	0.70	0.79	0.74	0.70	0.68	0.69
Battery	0.81	0.79	0.80	0.80	0.79	0.79	0.76	0.74	0.75	0.72	0.85	0.78	0.69	0.72	0.70
Ram	0.76	0.75	0.75	0.72	0.75	0.73	0.73	0.75	0.82	0.75	0.75	0.75	0.73	0.76	0.74
Rom	0.75	0.77	0.76	0.78	0.79	0.78	0.79	0.83	0.83	0.73	0.81	0.77	0.75	0.71	0.73
Camera	0.83	0.82	0.82	0.83	0.80	0.81	0.81	0.79	0.87	0.78	0.86	0.82	0.71	0.70	0.70
Sound	0.79	0.81	0.80	0.79	0.8	0.79	0.83	0.81	0.84	0.81	0.91	0.86	0.75	0.79	0.77
Style	0.78	0.81	0.79	0.8	0.77	0.78	0.74	0.75	0.83	0.82	0.88	0.85	0.78	0.83	0.80
Color	0.81	0.83	0.82	0.85	0.81	0.83	0.75	0.74	0.88	0.79	0.92	0.85	0.79	0.81	0.80
Application	0.74	0.75	0.74	0.77	0.79	0.78	0.70	0.72	0.79	0.82	0.78	0.80	0.74	0.79	0.76
Software	0.76	0.77	0.76	0.75	0.71	0.73	0.69	0.74	0.81	0.77	0.75	0.76	0.72	0.71	0.71

Table 27
The performance of aspect-based sentiment detection compares to the baseline methods.

Method	Baseline 1	Baseline 2	Proposed
$\mathcal{P}$	0.84	0.77	0.83
$\mathcal{R}$	0.83	0.78	0.85
$\mathcal{F}_1$	0.83	0.78	0.84

indicate these aspects. This limitation will be significantly solved by constructing a large data warehouse with a balance between factors. Additionally, the performance of aspect's sentiment detection in specific cases of the baseline methods is also shown in Tables 25 and 26.

Table 27 shows that our method gets better effectiveness than the baseline methods. Although the difference in efficacy is not very high, it demonstrates that our method increased the F_1 score of aspect-based sentiment analysis by up to 1% for the Baseline 1 method, and by up to 6% for the Baseline 2 method. Why does our approach achieve better performance than the baseline methods? This is because the word representation step can fully capture the useful information regarding words in

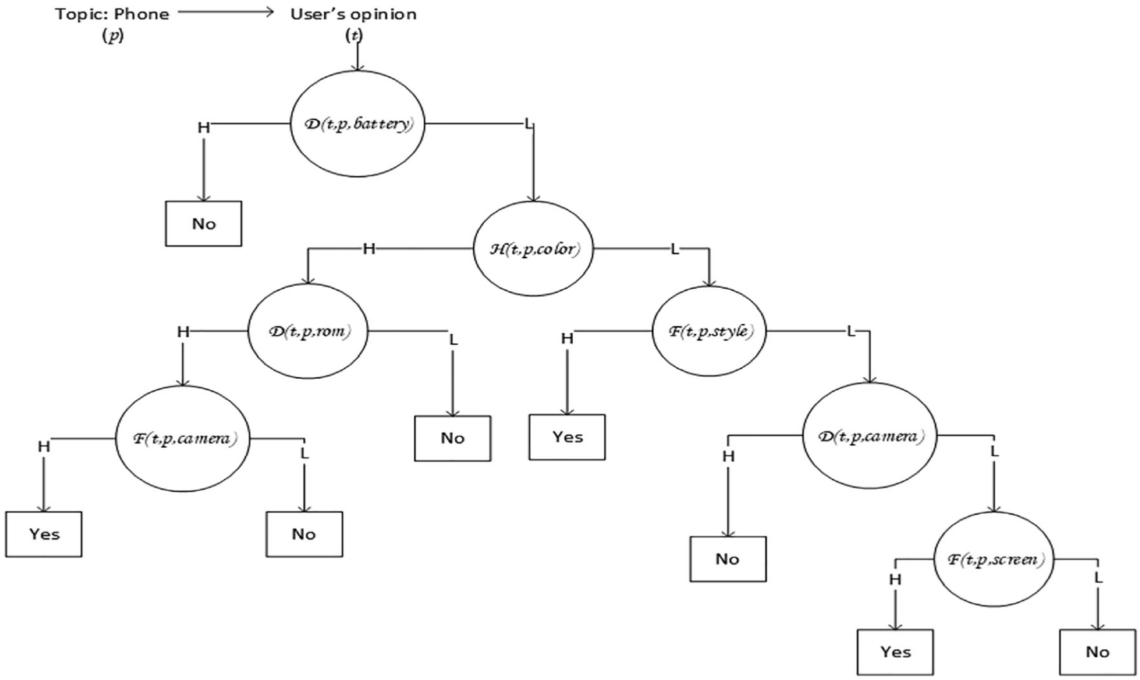

Fig. 5. The fuzzy decision tree of Phone topic (H is *high* (h), L is *low* (l)).

Table 28
Rules extraction.

#	Rules
1	$\mathcal{D}(t,p,style) = l \wedge \mathcal{D}(t,p,battery) = l \wedge \mathcal{F}(t,p,color) = l \rightarrow No$
2	$\mathcal{D}(t,p,style) = l \wedge \mathcal{D}(t,p,battery) = l \wedge \mathcal{F}(t,p,color) = h \rightarrow Yes$
3	$\mathcal{D}(t,p,style) = l \wedge \mathcal{D}(t,p,battery) = h \rightarrow No$
4	$\mathcal{D}(t,p,style) = l \wedge \mathcal{F}(t,p,screen) = l \wedge \mathcal{D}(t,p,rom) = l \rightarrow No$
5	$\mathcal{D}(t,p,style) = l \wedge \mathcal{F}(t,p,screen) = l \wedge \mathcal{D}(t,p,rom) = h \wedge F(t,p,software) = l \rightarrow No$
6	$\mathcal{D}(t,p,style) = l \wedge \mathcal{F}(t,p,screen) = l \wedge \mathcal{D}(t,p,rom) = h \wedge \mathcal{F}(t,p,software) = h \rightarrow Yes$
7	$\mathcal{D}(t,p,style) = l \wedge \mathcal{F}(t,p,screen) = h \wedge \mathcal{F}(t,p,camera) = l \wedge \mathcal{H}(t,p,battery) = l \rightarrow Yes$
8	$\mathcal{D}(t,p,style) = l \wedge \mathcal{F}(t,p,screen) = h \wedge \mathcal{F}(t,p,camera) = l \wedge \mathcal{H}(t,p,battery) = h \rightarrow No$
9	$\mathcal{D}(t,p,style) = l \wedge \mathcal{F}(t,p,screen) = h \wedge \mathcal{F}(t,p,camera) = h \rightarrow Yes$

Table 29
The confusion matrix decision-making between the proposed approach and the baseline methods.

Baseline 1 method			Baseline 2 method			Proposed method		
↓ →	Yes	No	↓ →	Yes	No	↓ →	Yes	No
Yes	2712	861	Yes	2371	849	Yes	2826	755
No	571	1206	No	912	1218	No	457	1312

↓: Actual →: Predicted.

tweets. Further, the sentiment score of FSPs and CSPs are computed more correctly. In addition, the BiLSTM?CRF algorithm is currently one of the best methods for aspect-based sentiment analysis. The results again prove that considering the role of FSPs has a significant impact on the performance of the sentiment analysis methods.

Using the results from the detection of the aspects and the sentiment of them, the fuzzy decision tree for the given topic is built and illustrated in Fig. 5; eight rules are developed as shown in Table 28.

Further, Table 29 shows that, by using the fuzzy decision tree and eight rules created from Table 28, we have 4138 users' opinions supporting decision-making (greater than the Baseline 1 and Baseline 2 methods with 220,549 decisions, respectively), wherein the "*Yes*" class has 2826 tweets (greater than the baseline methods with 114 and 455 tweets, respectively) and the "*No*" class has 1312 tweets (greater than the baseline methods with 94 and 106 tweets, respectively). Thus, our approach can effectively help users in decision-making regarding *phone* topic with an accuracy of 76% (greater than the Baseline 1 method in 4.2%). Thus, the error percentage of our method is approximately 24%. Therefore, our method is better in terms of supporting users in decision-making by analyzing the sentiment toward aspects of tweets. However, our method

Table 30
Comparison the performance in decision-making between the proposed approach and the baseline methods.

Performance	Baseline 1 method	Baseline 2 method	Proposed method
P	0.83	0.72	0.86
R	0.76	0.74	0.79
F_1	0.79	0.73	0.82

generally implemented better with the "*No*" class than with the "*Yes*" class. This may take place because of the uneven distribution among factors in tweets. Nevertheless, our proposal yields good decision-making results regarding the accuracy and gained information.

Table 30 shows the performance of the baseline methods in comparison with our proposal. As we can note, the proposed method has improved the F_1 score of the Baseline 2 method by up to 9% and the Baseline 1 method by 3% for supporting users in decision-making. Why can our method improve the accuracy? The reason is that the tweets are converted by using the information related to the lexicon, word-type, semantic, position, and polarity sentiment of words. Furthermore, the fuzzy structure of sentiments is used. In addition, the combined BiLSTM-CRF model used to detect aspects and their sentiments is one of the algorithms that achieves good accuracy for the items detection at the moment.

7. Conclusion and future work

In this study, a method supporting users in decision-making is proposed by means of combining of sentiment analysis for aspects with rules mined from a fuzzy decision tree. To implement this method, we consider four primary steps. First, the topic of user interest was extracted. Second, aspects and the sentiments of these aspects were identified in each tweet. Next, the user satisfaction degree for each aspect, as well as for the given topic, was determined. Finally, a fuzzy decision tree was constructed and mined. The results of experiments prove the efficiency of our proposal regarding accuracy and gained information. Our approach has several distinct characteristics. First, in order to indicate the sentiment orientations of aspects in tweets, we consider not only clear phrases but also fuzzy sentiment phrases. This is a proper attempt at increasing performance for analyzing the sentiment of aspects. Second, the main strength of the proposed method is that we consider user satisfaction, dissatisfaction, and hesitation when evaluating the aspects of a given topic, and use them to build a fuzzy decision tree. Considering the FSPs as well as user dissatisfaction and hesitation helps us overcome the limitations of the previous studies. More importantly, this method develops and enriches extant theories and techniques for decision-making based on the sentiment within tweets.

The main limitation of the presented study is that it has only been applied to tweets with a specific topic, namely "*phone*" without implementing other topics simultaneously. In some instances, the effect is slightly low, mostly because the distribution among the labels in tweets is not balanced. The problem is that our approach can usefully support the contextual analysis of tweets but not always. If the contextual analysis is understood like the point of view of the authors in the paper [3], this method is useful to support the contextual analysis of tweets. However, if the contextual analysis is understood like the point of view introduced in the paper [17], our proposal does not always support regarding this aspect. According to [17], contextual analysis is the process of identification, sorting, organization, interpretation, consolidation, and communication of the data gathered in the contextual inquiry. This process is used to build applications regarding business, marketing, and social phenomenon analyses that can understand the work context. In other words, to satisfy the requirements of the contextual analysis described in the paper [17], our method has to be implemented on many different topics simultaneously. Therefore, in the future, we need to make reasonable improvements so that our approach can usefully support the contextual analysis of tweets. It is essential that our proposal has to be applied for a large set of tweets regarding many topics, and the distribution between labels of tweets is more balanced. Nevertheless, the fuzzy decision tree depends much on data. Even with a small change in the data set, its structure could change significantly. Therefore, to construct a more comprehensive fuzzy decision tree to decision-making support concurrently on many different topics, we could extend the built fuzzy decision tree by adding many branches, where each branch of the tree is a fuzzy decision tree representing a particular topic. However, we need to deal with some new issues when building the tree with many branches: First, using an expansive tree could lead to higher complexity related with training the model, labeling cost, and differences among labels. Besides, under certain conditions, if the imbalance between labels in the data is high, then the model's performance could decrease. Finally, large number of branches can cause the process for building the fuzzy decision tree more time-consuming. That could lead to scaling issues, such as the fuzzy decision tree being more in-depth or broader, the decision-making time could increase, and as the consequence, the quality of the decisions is likely to decrease. In the future studies, these considerations will be the most urgent challenges for us.

CRediT authorship contribution statement

Huyen Trang Phan: Conceptualization, Methodology, Formal analysis, Software, Writing - original draft, Writing - review & editing. **Ngoc Thanh Nguyen:** Supervision, Conceptualization, Validation, Methodology, Writing - original draft, Writing - review & editing. **Van Cuong Tran:** Data curation, Conceptualization, Validation, Writing - review & editing. **Dosam Hwang:** Supervision, Methodology, Writing - original draft, Project administration, Funding acquisition.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

[1] A.M. Abirami, A. Askarunisa, Sentiment analysis model to emphasize the impact of online reviews in healthcare industry, Online Inf. Rev. 41 (2017) 471–486.
[2] M.S. Akhtar, T. Garg, A. Ekbal, Multi-task learning for aspect term extraction and aspect sentiment classification, Neurocomputing (2020).
[3] AL-Sharuee, M.T., Liu, F., Pratama, M., 2018. Sentiment analysis: an automatic contextual analysis and ensemble clustering approach and comparison. Data & Knowledge Engineering 115, 194–213..
[4] A.S.M. Alharbi, E. de Doncker, Twitter sentiment analysis with a deep neural network: an enhanced approach using user behavioral information, Cogn. Syst. Res. 54 (2019) 50–61, URL:https://doi.org/10.1016/j.cogsys.2018.10.001, 10.1016/j.cogsys.2018.10.001.
[5] A. Baykasoğlu, İ. Gölcük, A dynamic multiple attribute decision making model with learning of fuzzy cognitive maps, Comput. Ind. Eng. 135 (2019) 1063–1076.
[6] P. Bedi, P. Khurana, Sentiment Analysis Using Fuzzy-Deep Learning, 2019, pp. 246–257. doi: 10.1007/978-3-030-30577-2_21..
[7] S. Begenova, T. Avdeenko, Building of fuzzy decision trees using id3 algorithm, in: Journal of Physics: Conference Series, IOP Publishing, 2018, p. 022002.
[8] D.M. Blei, A.Y. Ng, M.I. Jordan, Latent dirichlet allocation, J. Mach. Learn. Res. 3 (2003) 993–1022.
[9] M. Bohanec, Decision making: a computer-science and information-technology viewpoint, Interdisc. Descript. Complex Syst. INDECS 7 (2009) 22–37.
[10] S. Çali, S.Y. Balaman, Improved decisions for marketing, supply and purchasing: mining big data through an integration of sentiment analysis and intuitionistic fuzzy multi criteria assessment, Comput. Ind. Eng. 129 (2019) 315–332, URL:https://doi.org/10.1016/j.cie.2019.01.051, 10.1016/j.cie.2019.01.051.
[11] D. Cavaliere, J.A. Morente-Molinera, V. Loia, S. Senatore, E. Herrera-Viedma, Collective scenario understanding in a multi-vehicle system by consensus decision making, IEEE Trans. Fuzzy Syst. (2019).
[12] D. Chandramohan, T. Vengattaraman, D. Rajaguru, P. Dhavachelvan, A new privacy preserving technique for cloud service user endorsement using multi-agents, J. King Saud Univ. Comput. Inf. Sci. 28 (2016) 37–54.
[13] I. Chaturvedi, E. Cambria, R.E. Welsch, F. Herrera, Distinguishing between facts and opinions for sentiment analysis: survey and challenges, Inf. Fusion 44 (2018) 65–77.
[14] Y. Chen, H. Zhang, R. Liu, Z. Ye, J. Lin, Experimental explorations on short text topic mining between lda and nmf based schemes, Knowl.-Based Syst. 163 (2019) 1–13.
[15] M. Colhon, C. Bădică, A. Şendre, Relating the opinion holder and the review accuracy in sentiment analysis of tourist reviews, in: International Conference on Knowledge Science, Engineering and Management, Springer, 2014, pp. 246–257.
[16] P. Ducange, M. Fazzolari, M. Petrocchi, M. Vecchio, An effective decision support system for social media listening based on cross-source sentiment analysis models, Eng. Appl. Artif. Intell. 78 (2019) 71–85.
[17] R. Hartson, P.S. Pyla, The UX Book: Process and Guidelines for Ensuring a Quality User Experience, Elsevier, 2012.
[18] Z. Huang, W. Xu, K. Yu, Bidirectional lstm-crf models for sequence tagging, 2015. arXiv preprint arXiv:1508.01991..
[19] C.Z. Janikow, Fuzzy decision trees: issues and methods, IEEE Trans. Syst., Man Cybern. B 28 (1998) 1–14.
[20] C. Jefferson, H. Liu, M. Cocea, Fuzzy approach for sentiment analysis, in: 2017 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE), IEEE, 2017, pp. 1–6..
[21] H. Jelodar, Y. Wang, C. Yuan, X. Feng, X. Jiang, Y. Li, L. Zhao, Latent dirichlet allocation (lda) and topic modeling: models, applications, a survey, Multimedia Tools Appl. 78 (2019) 15169–15211.
[22] J. Jin, Y. Liu, P. Ji, H. Liu, Understanding big consumer opinion data for market-driven product design, Int. J. Prod. Res. 54 (2016) 3019–3041.
[23] C. Karyotis, F. Doctor, R. Iqbal, A. James, V. Chang, A fuzzy computational model of emotion for cloud based sentiment analysis, Inf. Sci. 433 (2018) 448–463.
[24] S.M. Kim, E. Hovy, Crystal: Analyzing predictive opinions on the web, in: Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL), 2007, pp. 1056–1064.
[25] B. Liu, Sentiment Analysis: Mining Opinions, Sentiments, and Emotions, Cambridge University Press, 2020.
[26] Y. Liu, J.W. Bi, Z.P. Fan, A method for ranking products through online reviews based on sentiment classification and interval-valued intuitionistic fuzzy topsis, Int. J. Inf. Technol. Decis. Mak. 16 (2017) 1497–1522.
[27] Y. Liu, J.W. Bi, Z.P. Fan, Ranking products through online reviews: a method based on sentiment analysis technique and intuitionistic fuzzy set theory, Inf. Fusion 36 (2017) 149–161.
[28] M.E. Mowlaei, M.S. Abadeh, H. Keshavarz, Aspect-based sentiment analysis using adaptive aspect-based lexicons, Expert Syst. Appl. 148 (2020) 113234.
[29] E. Najmi, K. Hashmi, Z. Malik, A. Rezgui, H.U. Khan, Capra: a comprehensive approach to product ranking using customer reviews, Computing 97 (2015) 843–867.
[30] N.T. Nguyen, Consensus system for solving conflicts in distributed systems, Inf. Sci. 147 (2002) 91–122.
[31] S.O. Oppong, D. Asamoah, E.O. Oppong, D. Lamptey, Business decision support system based on sentiment analysis, Int. J. Inf. Eng. Electron. Business 12 (2019) 36.
[32] H.G. Peng, H.Y. Zhang, J.Q. Wang, Cloud decision support model for selecting hotels on tripadvisor. com with probabilistic linguistic information, Int. J. Hospital. Manage. 68 (2018) 124–138..
[33] W. Peng, J. Chen, H. Zhou, An implementation of id3-decision tree learning algorithm, 2009. From web. arch. usyd. edu. au/wpeng/DecisionTree2. pdf Retrieved date: May 13..
[34] J. Pennington, R. Socher, C. Manning, Glove: global vectors for word representation, in: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), 2014, pp. 1532–1543.
[35] H.T. Phan, N.T. Nguyen, V.C. Tran, D. Hwang, A sentiment analysis method of objects by integrating sentiments from tweets, J. Intell. Fuzzy Syst. 1–13..
[36] H.T. Phan, N.T. Nguyen, T. Van Cuong, D. Hwang, A method for detecting and analyzing the sentiment of tweets containing fuzzy sentiment phrases, in: 2019 IEEE International Symposium on INnovations in Intelligent SysTems and Applications (INISTA), IEEE, 2019, pp. 1–6..
[37] H.T. Phan, V.C. Tran, N.T. Nguyen, D. Hwang, Decision-making support method based on sentiment analysis of objects and binary decision tree mining, in: International Conference on Industrial, Engineering and Other Applications of Applied Intelligent Systems, Springer, 2019, pp. 753–767.
[38] H.T. Phan, V.C. Tran, N.T. Nguyen, D. Hwang, Improving the performance of sentiment analysis of tweets containing fuzzy sentiment using the feature ensemble model, IEEE Access 8 (2020) 14630–14641.
[39] J. Qi, Z. Zhang, S. Jeon, Y. Zhou, Mining customer requirements from online reviews: a product improvement perspective, Inf. Manage. 53 (2016) 951–963.
[40] M. Rajabi, B. Bohloli, E.G. Ahangar, Intelligent approaches for prediction of compressional, shear and stoneley wave velocities from conventional well log data: a case study from the sarvak carbonate reservoir in the abadan plain (southwestern Iran), Comput. Geosci. 36 (2010) 647–664.

[41] P.D. Turney, Thumbs up or thumbs down?: Semantic orientation applied to unsupervised classification of reviews, in: Proceedings of the 40th Annual Meeting on Association for Computational Linguistics, Association for Computational Linguistics, 2002, pp. 417–424.
[42] T.C. Wang, H.D. Lee, et al, Constructing a fuzzy decision tree by integrating fuzzy sets and entropy, WSEAS Trans. Inf. Sci. Appl. (2006).
[43] D. Wu, Fuzzy sets and systems in building closed-loop affective computing systems for human-computer interaction: advances and new research directions, in: 2012 IEEE International Conference on Fuzzy Systems, IEEE, 2012, pp. 1–8.
[44] S. Wu, Y. Xu, F. Wu, Z. Yuan, Y. Huang, X. Li, Aspect-based sentiment analysis via fusing multiple sources of textual knowledge, Knowl.-Based Syst. 183 (2019) 104868.
[45] Y. Yuan, M.J. Shaw, Induction of fuzzy decision trees, Fuzzy Sets Syst. 69 (1995) 125–139.
[46] N. Yussupova, M. Boyko, D. Bogdanova, A. Hilbert, A decision support approach based on sentiment analysis combined with data mining for customer satisfaction research, Int. J. Adv. Intell. Syst. 8 (2015) 145–158.
[47] K. Zhang, R. Narayanan, A.N. Choudhary, Voice of the customers: mining online customer reviews for product feature-based ranking, WOSN 10 (2010) 11.
[48] W. Zhang, H. Xu, W. Wan, Weakness finder: find product weakness from chinese reviews by using aspects based sentiment analysis, Expert Syst. Appl. 39 (2012) 10283–10291.
[49] J. Zhou, Q. Chen, J.X. Huang, Q.V. Hu, L. He, Position-aware hierarchical transfer model for aspect-level sentiment classification, Inf. Sci. 513 (2020) 1–16.
[50] J. Zhou, J.X. Huang, Q.V. Hu, L. He, Sk-gcn: Modeling syntax and knowledge via graph convolutional network for aspect-level sentiment classification, Knowl.-Based Syst. 205 (2020) 106292.