Knowledge-Based Systems 174 (2019) 27–42

Contents lists available at ScienceDirect

Knowledge-Based Systems

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

Transportation sentiment analysis using word embedding and ontology-based topic modeling

Farman Ali [a], Daehan Kwak [b], Pervez Khan [c], Shaker El-Sappagh [a,d], Amjad Ali [a,e], Sana Ullah [f,g], Kye Hyun Kim [h], Kyung-Sup Kwak [a,*]

[a] Department of Information and Communication Engineering, Inha University, Incheon, Republic of Korea
[b] Department of Computer Science, Kean University, Union, NJ, USA
[c] Department of Computer Science and Information Technology, University of Malakand, Chakdara, Pakistan
[d] Department of Information Systems, Benha University, Banha, Egypt
[e] Department of Computer Science, COMSATS University Islamabad, Lahore Campus, Lahore, Pakistan
[f] Department of Informatics, Gyeongsang National University, Jinju, Republic of Korea
[g] Department of Computer and Software Technology, University of Swat, Swat, Pakistan
[h] Department of Geoinformatic Engineering, Inha University, Incheon, Republic of Korea

HIGHLIGHTS

- Social networks provide a new approach to collect data regarding transportation.
- Sentiment analysis can make observations of social data to examine transportation.
- Current text mining techniques are unable to generate the topics accurately.
- Document representation is another challenging tasks in sentiment analysis.
- We proposed a new topic modeling and word embedding system for sentiment analysis.

ARTICLE INFO

Article history:
Received 23 July 2018
Received in revised form 2 January 2019
Accepted 24 February 2019
Available online 5 March 2019

Keywords:
Social network analysis
Sentiment analysis
Topic modeling
Mobility users
Word embedding

ABSTRACT

Social networks play a key role in providing a new approach to collecting information regarding mobility and transportation services. To study this information, sentiment analysis can make decent observations to support intelligent transportation systems (ITSs) in examining traffic control and management systems. However, sentiment analysis faces technical challenges: extracting meaningful information from social network platforms, and the transformation of extracted data into valuable information. In addition, accurate topic modeling and document representation are other challenging tasks in sentiment analysis. We propose an ontology and latent Dirichlet allocation (OLDA)-based topic modeling and word embedding approach for sentiment classification. The proposed system retrieves transportation content from social networks, removes irrelevant content to extract meaningful information, and generates topics and features from extracted data using OLDA. It also represents documents using word embedding techniques, and then employs lexicon-based approaches to enhance the accuracy of the word embedding model. The proposed ontology and the intelligent model are developed using Web Ontology Language and Java, respectively. Machine learning classifiers are used to evaluate the proposed word embedding system. The method achieves accuracy of 93%, which shows that the proposed approach is effective for sentiment classification.

1. Introduction

Recent advances in social media and textual resources have allowed realization of information retrieval and sentiment analysis in data mining and natural language processing (NLP) [1].

* Corresponding author.
E-mail addresses: farmankanju@gmail.com (F. Ali), dkwak@kean.edu (D. Kwak), pervaizkanju@hotmail.com (P. Khan), shaker_elsapagh@yahoo.com (S. El-Sappagh), amjad.khu@gmail.com (A. Ali), sanajcs@hotmail.com (S. Ullah), kyehyun@inha.ac.kr (K.H. Kim), kskwak@inha.ac.kr (K.-S. Kwak).

https://doi.org/10.1016/j.knosys.2019.02.033
0950-7051/

However, extracting valuable information from online news articles and social media, such as Twitter, Facebook, and TripAdvisor, has become a new challenge for sentiment analysis. On one hand, the texts on social networks are unstructured and constantly increasing. On the other hand, online texts are short and have a lot of slang, idioms, jargon, and dynamic topics.

Intelligent transportation systems (ITSs) need social network data in order to examine transportation services and support traffic control and management systems. In social media, information about transportation networks, such as traffic jams and accidents, appears regularly with unexpected texts, and it would be a challenging task to extract these data and transform them into valuable information for analysis.

Text mining has gained much more attention among researchers, and has been proposed for automating information extraction from unstructured textual data. The rapid improvement in NLP and machine learning (ML) has developed two frameworks for text mining: one-hot encoding and word embedding. Statistical learning models exhibit good performance in document representation. Bag-of-words (BoW) is the first and the most popular model to represent a document in the field of NLP [2]. This model represents a document as a dictionary, and contains all words that occur in the document. The BoW model is easy to implement, works fast, and achieves good results with very little data. However, the dimensionality of a word vector is high, even for a single sentence, and neglects word order in the BoW model. Since it is not capable of representing large-scale data, the performance of the classifiers could not be improved. Therefore, a probabilistic approach has been proposed to overcome the limitations of BoW, such as latent Dirichlet allocation (LDA), latent semantic indexing (LSI), and principle component analysis (PCA).

Word embedding is a distributed representation approach, which is an alternative to BoW [1,2]. It represents each word with a very low-dimensional vector and semantic meaning. In order to represent a word-vector for corpus data, a word embedding model, such as word2vec, doc2vec, and GloVe, must be trained using a large amount of social media data. However, word embedding models have some limitations. Using a pre-trained word embedding model with high dimensionality for a small amount of data is not the best way. For document representation, the two estimation methods of a word-vector miss the context of documents. In addition, word embedding neglects information on sentiment in any given content.

An LDA statistical model can automatically discover a latent topic from a large volume of transportation data. LDA disregards word order and groups semantically related words into the same topic based on their representation in the documents. However, LDA has three main limitations that affect the classification results. First, the generated topics under LDA comprise irrelevant features when other transportation-related text is in them. Second, it produces very noisy topics from short text, and misses valuable topics because of the limited dataset. Third, it neglects the relation between topic and document when a document has low-probability words. Ontologies are considered the best approach, and can enhance the performance of LDA to find appropriate topics along with features (words) in transportation data.

The goal of the proposed system is to improve the performance of document representation and sentiment classification. However, the accuracy of sentiment classification is dependent on the representation of text in documents. The existing text representation models examine imprecise words, which are not associated with the topics of the document, and neglect information on sentiment in any given content. Therefore, we propose ontology- and LDA-based topic modeling and a word embedding system to precisely represent texts to improve the accuracy of sentiment classification. The proposed model was trained using datasets from different social media networks, and an evaluation is conducted with ML classifiers. The results prove that the proposed approach is capable of correctly representing documents, and improves the accuracy of sentiment classification. The main contributions in this research are the following.

- We propose a novel framework that retrieves the most relevant documents, reviews, and Tweets from social media and news articles.
- We propose ontology- and LDA-based topic modeling called topic2vec that extracts the most appropriate topics and features of document, and neglects irrelevant words to enhance the document representation. The proposed ontology represents semantic knowledge that enriches an LDA model to extract more accurate features from transportation texts.
- We integrate a topic2vec with word2vec and generate a word embedding model that represents each word in the document with semantic meanings and a low-dimensional vector.
- We propose a new fuzzy ontology-based lexicon method, which is used with six other lexicons to enhance the accuracy of the pre-trained word embedding model in sentiment classification tasks.
- We compare the performance of string2vec, word2vec, doc2vec, glove2vec, and lexicon2vec with our proposed model. We use ML algorithms to classify the data from these models and present the results. The comparison results help understand the limitations and advantages of the document representation models.

This paper is structured as follows. Section 2 presents discussions of sentiment analysis, topic modeling, and document representation models. Section 3 illustrates our proposed framework and the procedure of data collection and filtration. Section 4 provides information about topic modeling and word embedding. Section 5 presents the experimental results. Finally, Section 6 concludes our work.

2. Related work

This section looks at sentiment analysis, topic modeling, and word embedding approaches. First, we discuss the general standpoint of sentiment analysis, and then focus on the domain of social data related to transportation. We also present a brief review of topic modeling and deep learning-based word embedding approaches in sentiment classification.

2.1. Sentiment analysis

Sentiment analysis and opinion mining have been studied since the early 2000s, and several methods have been introduced to analyze emotions and opinions from social media [3–5]. Nowadays, people have a special interest in sharing their opinions on social networks like Facebook [6,7], Twitter [8,9], and TripAdvisor [10,11], regarding many topics. These data are analyzed using two main methods: ML methods and lexicon-based methods. ML methods such as the support vector machine (SVM), Naïve Bayes (NB), logistic regression, and multilayer perceptron (MLP) need a training dataset to learn the model from corpus data, and a testing dataset to verify the model built [5,9,12–14]. The lexicon-based method is based on a dictionary of words and phrases with positive and negative values. The most widely used lexicon in the field of sentiment analysis is SentiWordNet [15,16]. However, this approach may not obtain a good result in some domains

due to the different senses of words. To overcome this problem, domain-dependent lexicons are presented for the proposed system.

In the field of text mining, the extraction of relevant information from social media is a challenging task. In the existing research, many approaches have been proposed for information extraction and sentiment analysis, including lexical knowledge, deep learning, neural word embedding, and fuzzy logic [8,17–19].

Recently, the processing of transportation data and the extraction of traffic information from social networks have become hot topics in ITSs [13,20–26]. People share their opinions on social networks regarding various issues related to transportation (e.g., traffic jams, landslides, and accidents), and other users react to the same subject with text or emoticons. However, the opinions of users about transportation are in terms of features like "*traffic police were helpful in Victoria downtown.*" [21]. It is difficult for both system and users to understand the expression of people's opinions. In addition, the transformation of data into meaningful information can be useful for traffic management and control systems, and transportation services. However, to extract information and handle transport-related problems, it is important to filter out irrelevant information, and then identify topics and features. A neutrality proximity function was presented to filter out neutral information before binary classification [27]. This existing system employs proximity function to assign weights to each neutral point, uses the polarity aggregation model to compute polarities, and then filter out neutral information. However, a pre-processing steps of text analysis must be applied before and during binary classification, which is difficult and time consuming. A system based on multiple filters was proposed to enhance the performance of sentiment analysis [28]. In this existing system, each filtering service is assigned to a specific task, such as, misspelled detection and special characters removal. In addition, several systems have been proposed to manually filter the text to obtain meaningful information [29,30]. However, these are cumbersome tasks to examine each word by multiple filters or manually check each sentence to find irrelevant text. Therefore, SVM-based data filtering is an efficient approach to handle the above issues, which classifies the data into two classes: relevant and irrelevant information.

2.2. Ontology and LDA-based topic modeling

In the past few years, researchers have used the ontology and LDA in the field of sentiment analysis that is briefly discussed in this section [16,31–33]. Topic-based sentiment analysis was presented to extract topics from online travel reviews [34]. The main objective of the existing research is to find topics deemed the most important by the tourist, and to find the emotion words in each topic. Onan et al. examined the performance of LDA-based feature representation in text classification [35]. They used machine learning classifiers with LDA to find the optimal number of latent topics for sentiment classification. Ren et al. encoded information about topics into word embedding to analyze sentiments expressed on Twitter, and employed a recursive encoder to accomplish the goal [36]. They first generated topics from Tweets by using LDA, and the topic information was then incorporated into the main function. Their system performance was improved 4% by integrating topics into word embedding. An LDA-based ontology learning method was presented to update the domain of a civil aviation system [32]. The system enriched the representation content of the ontology information and provided better support for an emergency management system. The LDA-based approach was presented to find opportunities for a product [37]. In their work, the authors identified product opportunities from social media to monitor changing customer needs. However, LDA-based generated topics contain irrelevant words. In addition, the LDA-based approach avoids the association between topic and document and is unable to learn semantic relationship between words. Therefore, an ontology can be used as a feature representation method with latent Dirichlet allocation to identify a topic [33]. The ontology represents information about a specific domain that can be understood by both humans and systems. LDA analyzes documents to extract topics. Santosh et al. presented an ontology to improve the performance of LDA [16]. They used the ontology to identify appropriate features after clustering, and showed that the accuracy of the feature extraction largely improved. An ontology-based, feature-level sentiment analysis was presented to describe the relationships between concepts in a specific domain [38]. The semantic knowledge of an ontology reduces the effort required to implement an expert system. Katsumi and Fox presented a survey of transportation ontologies [39]. The main aim of this survey is to understand the existing transportation ontologies and to provide research direction for the development of transportation ontologies. A traffic accidents management system was developed using an ontology [40]. This ontology contains meaningful information regarding roads, climate, environments, and pedestrians, which are used in situations of traffic safety. However, traffic management systems are always developed according to the requirements, but semantic knowledge based on fuzzy logic is very limited in retrieving and expressing information. Ali et al. presented two systems for fuzzy ontology-based sentiment analysis of transportation features [41]. The first system calculates the polarity of six features to monitor transportation services. However, these features are not enough to monitor all the traffic and to help the traffic management system. There is no such detail regarding the allocation of values to the corresponding sentiment words. In addition, their system is based on a simple web crawler and keywords-based matching. Mostly, the data are stated in various ways and are disorganized; therefore, their system may not help to collect useful information about a specific topic. To overcome the above-mentioned limitations, Ali et al. presented another fuzzy ontology-based sentiment analysis system to help ITS transportation facilities during traffic monitoring [21].

The previous works are based on traditional approaches, which differs from our research. We propose an ontology and LDA-based topic modeling method for transportation data classification. A classic ontology may not be enough for the transportation domain. The concept of classic ontology is based on crisp logic and unable to deal with uncertain information. In addition, it is hard to represent non crisp data; such as a very horrible accident, traffic is very slow, and driver is very old, within the ontology definition. Therefore, LDA-based extracted information is fed to the ontology to build a fuzzy transportation ontology.

2.3. Document representation models

The texts on social media are in a short format and contain dynamic topics, which create new challenges for sentiment analysis research. Recently, various systems have been proposed to represent social data in the form of feature engineering. For example, fuzzy bag-of-words [42], a neural network [43], augmented LDA [44,45], lexicons [1,15,46,47], and semantics [30,48] were used in text representation. In addition, feature selection [49,50] and text normalization [51,52] were also combined for text representation.

Fuzzy BoW (FBoW) was presented to study more vigorous document representations by programming more semantics [42]. The existing system substitutes for the original hard planning by fuzzy mapping, and improves the FBoW model. FBoW matches vague words and allows words semantically similar to a basis

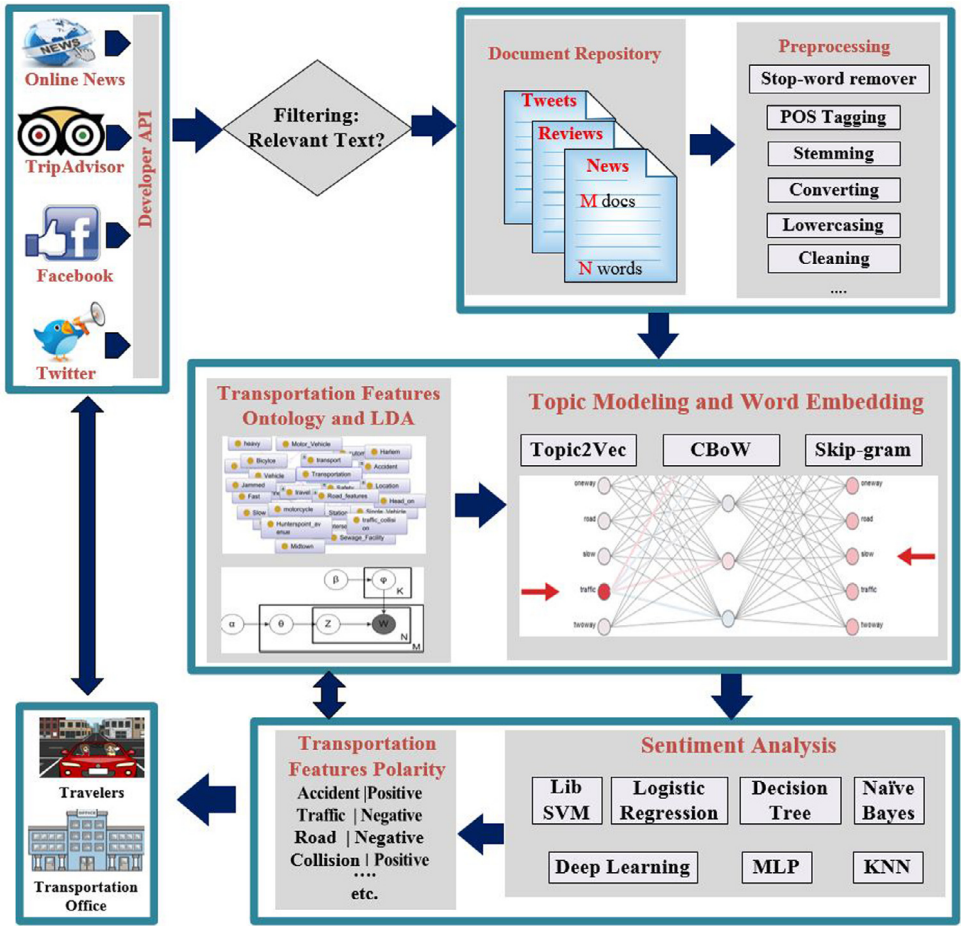

Fig. 1. Architecture of the proposed system for sentiment analysis of transportation.

term. An unsupervised framework was presented to learn the representation of distributed vectors for pieces of text [2]. The vector representation was trained to predict words in a paragraph. Stochastic gradient descent was used to train both word vectors and paragraph vectors. A least squares model was proposed to train global word2word co-occurrence [53]. A word vector space was produced by this model, along with a useful structure. The performance accuracy was 75% on the analogy dataset of words. Content-tree word embedding was proposed to represent the word vector [50]. The risk of word ambiguity is checked, and a local context is injected into pre-trained word vectors. During development, each vector is updated in the content tree, and this step improved the performance of accuracy and F-score. A vector representation for ontology classes was proposed that converts the instances to the vector space [54]. The existing ontology was developed by an expert in the consumer protection domain. They used FindLaw cases to train a word2vec model. For the word embedding vectors, the ontology instances and class labels were obtained by employing the word2vec model.

Recently, a deep learning approach was used to study text representation and to overcome the problem of sentiment classification on a large social network dataset [17,50,55]. A novel approach, Improved Word Vectors (IWV), was proposed for word embedding in the field of sentiment analysis [56]. It is based on three approaches: lexicon-based methods, word2vec and GloVe approaches, and part-of-speech (POS) tagging approaches. IWV was tested with different datasets and various deep learning models, and the results showed that this approach increases the accuracy of word embedding for sentiment analysis.

Most of the above-mentioned research was based on traditional approaches and can relieve the problems of sentiment classification to some extent. These systems are still insufficient to deal with sentiment analysis in social media when the texts are from different social media platforms, when topics are continuously changing, and when text data are constantly increasing.

3. Proposed approach

This section briefly introduces different methods that are applied to develop the proposed OLDA-based topic modeling and word embedding system. The main focus of the proposed approach is to enhance the performance of topic modeling, document representation, and sentiment classification. We used different techniques (namely LDA, the ontology, and deep learning) to represent words along with the most relevant topics for opinion classification. LDA is applied to find the statistical relationships between words, topics, and documents in a large corpus [57]. However, traditional LDA generates topics that conflict with human thinking. It neglects low probability words and selects high probability words as a topic. In transportation text mining, low probability words can also differentiate the topics efficiently. In addition, LDA is unable to learn the semantic relationships among words. Therefore, ontology-based semantic knowledge is applied to enrich the LDA model and extract more accurate topics related to transportation.

Text representation is another challenging task in feature extraction and sentiment classification. We employed deep learning-based word embedding methods called word2vec and

glove2vec. These methods are used to transform each word in the corpus to a low dimensional vector. The word is highly associated with real semantics. However, these techniques have several limitations that are discussed in depth in Section 4. Fig. 1 illustrates the entire architecture of the proposed approach. The proposed framework is composed of six main components: data collection and filtration, preprocessing of data, a transportation feature ontology, LDA and ontology-based topic modeling, deep learning-based word embedding, and sentiment classification and polarity declaration.

3.1. Data collection and filtration

This module collects transportation-related data from online news articles and different social media platforms.

3.1.1. News article data

For news article collection, we used a *New York Times* Developer Network application programming interface (API) to retrieve articles containing related information published between April 2017 and July 2017. The retrieved data contain multilingual and irrelevant text. Therefore, the collected data were filtered to detect only news that is written in English [58]. We split lengthy news articles into paragraphs, and then a keyword-based search technique was applied to detect the most relevant information. This process reduces noisy and irrelevant text.

3.1.2. TripAdvisor data

The proposed system retrieved data from TripAdvisor for three types of entities: location, city name, and city features (e.g., hotels, restaurants, train stations, bus stations, parks, hospitals, and bridges). The city features along with a city name are used as a search query to collect reviews with metadata (title and description of web page). TripAdvisor contains real-time information that may not be about a fixed topic but often discusses various locations and places. However, retrieving information about a specific topic is a more challenging task. We designed predefined search queries for specific events and topics that extract more precise data about transportation. The dataset we retrieved from TripAdvisor contains 1851 reviews about New York and London and the aforementioned city features [21]. The average length of each review was 73 words.

3.1.3. Facebook data

People with the same interests are connected through the Facebook platform. Various organizations, businesses, and transportation departments use it to share information and activities with customers. Users can respond to posts and share with others the posts by these organizations. For Facebook data collection, we used the Graph API along with a Java client, RestFB, to extract data from Facebook pages [7]. The API allows us to fetch data from Facebook and to automate extraction of the information. We first selected those pages (e.g., future transportation, Transport for London, and the New York State Department of Transportation) that contain transportation information about New York and London. We extracted all posts from pages that were published between March 2016 and January 2018. Then, we retrieved responses (reactions and emotions) and comments made in these posts, and stored them for further processing.

3.1.4. Twitter data

We employed the Twitter APIs called REST APIs and Streaming APIs to retrieve Tweets containing transportation data. REST APIs allows users to employ queries for crawling the most recent Tweets [21]. These queries should be more specific in terms of transportation features to fetch the most relevant Tweets. Therefore, we constructed queries that are based on keywords, radius, centroid, and the Boolean operators AND and OR, e.g., car AND (collision OR accident). We employed 500 keywords related to transportation features to construct queries. However, REST APIs permits a single customer to employ only 350 queries per 15 min, and it retrieves 3200 of the most-recent Tweets per query. We fetched 30,000 Tweets about transportation features by using the aforementioned keywords-based queries.

3.1.5. Data filtering

Vast amounts of data are available on the aforementioned social media platforms. It is predicted that the volume of social media data will double by 2025. However, classic knowledge-based data filtering systems cannot work with large amounts of social media data, and even if they could, it is time consuming. Therefore, we propose a features ontology and a support vector machine-based filtering system. In this proposed system, various queries are constructed to retrieve relevant texts, and then queries with a high percentage of recall are employed. The proposed ontology along with the queries, fetches the most relevant documents, reviews, and Tweets. The SVM is then used to find related texts and remove all content not related to transportation. For example, if the SVM classifier function f (documents, reviews, or Tweets) > 0, then the text is considered positive and associated with transportation; otherwise, the text is considered negative and is filtered out. This process increases the precision rate of information retrieval and the accuracy of sentiment classification.

3.2. Pre-processing

The pre-processing method involves the following steps, which present the corpus data in a more structured form to easily extract transportation features and opinion words. In addition, these steps clean the corpus data and prepare them for word embedding.

3.2.1. Stop-word removing and cleaning

The most common words in a text, such as *is*, *by*, and *the*, decrease the accuracy rate of sentiment classification. The URLs in the corpus do not contain much information about the sentiment of the review, document text, and Tweet. Therefore, before feature extraction, the proposed system eliminates URLs and words that occur frequently. In addition, the proposed system employs a stop-word handler called Raindow to remove content that does not contribute to the sentiment. This includes articles (a, an, the), symbols (@, date, #, etc.), and punctuation.

Some text contains notions of negation and numbers. Negation plays a key role in finding the sentiment of a sentence. The existing work claims that numbers in Tweets and reviews cannot help in text analysis, and hence, discards numbers [59]. However, in transportation text mining, numbers are used to find entities and places (e.g., bus number 908 and street number 23). Thus, the proposed system transforms negations (e.g., *do not* into *do not*) to easily determine the sentiment of a feature, and uses numbers to recognize entities.

3.2.2. Part-of-speech tagging

We split the reviews, Tweets, and texts into sentences and then applied Stanford NLP to assign the parts of speech. After POS tagging, every sentence is confirmed as a full clause with a noun and a verb. This step makes ontology and LDA-based feature extraction much easier.

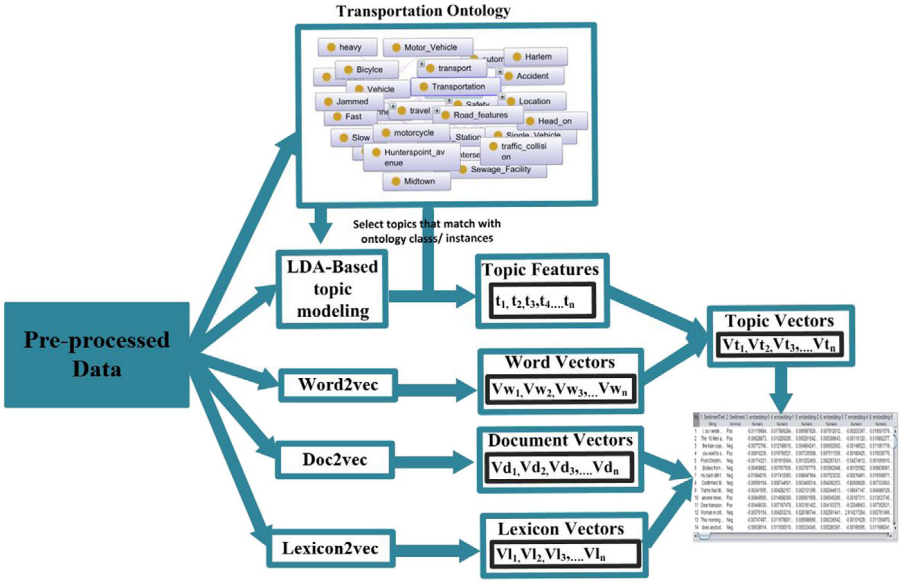

Fig. 2. A scenario of topic modeling and document representation.

Algorithm 1. Extraction of features and opinion words, and construction of the ontology.

Data: A collection of transportation text from different social media
Results: Relevant transportation corpus (TC) and ontology of transportation features
Begin

```
//Extract relevant information
for each doc ϵ corpus do
    for each sentence ϵ doc do
        If SVM classifier (text)>0 then
            Add text in final transportation corpus (TC);
        end if
    end for
end for
// extraction of features and opinion words from TC
For each doc ϵ corpus TC do
    for each sentence ϵ doc do
        Sentence ← pre-process all sentences with NLP
                   approach explained in Section 3;
    end for
    vocabulary ← features and opinion words;
end for
Construct fuzzy ontology of transportation features;
```

3.2.3. Tokenization

The tokenization process separates the composite text in the corpus into small tokens. The composite text may contain a word-space and delimiters. Therefore, the proposed system employs an N-gram tokenizer to eliminate delimiters and word-spaces. We took a sentence from the corpus, such as *"I am stuck in traffic on a bridge in Vienna due to an accident and every time someone goes by on the opposite side"*. After applying the N-gram tokenizer, the system obtains the result in the form of chunks: "stuck"; "in"; "traffic"; "on"; "bridge"; "Vienna"; "an"; "accident". The result is then kept in the form of an array for further analysis [60].

3.2.4. Stemming and lemmatization

In text analysis, stemming is the process of converting a word to its basic form. For example, a sentence about a traffic feature is *"traffic increasing on NH16 near toll plaza. Demand to shift toll plaza. Helped nearby drivers by reporting a stand-still traffic jam on NH16"*. In these sentences, the words *increasing*, *helped*, and *reporting* have the basic forms *increase*, *help*, and *report* as their stems [61]. We applied suffix-dropping algorithms to reduce words to their root forms. Lemmatization plays an important role in sentiment analysis [60]. It employs a lexicon to determine the lemma of the used words in a sentence. After lemmatization, the proposed system obtains lexical information on each word. For example, *accident* is related to *collision*; *congestion* is related to *jam*; and so on. Thus, the proposed system employs the stem and lemma words for further analysis.

3.2.5. Converting characters, and lowercasing

The first three steps above, along with different techniques, are generally used for text analysis, whereas the last two are always avoided, which can affect the results. People use a lot of unusual words on social platforms to express their opinions. For example, *flood* is employed in the form *flooooood*, which indicates the same word: *flood*. However, ML classifiers acknowledge the aforementioned words. To represent words in the form of generic words, the proposed system converts a sequence of characters repeated more than two times (e.g., "floood" becomes "flood"). Lowercasing converts every word to lower case to avoid confusion during processing. The pseudo code for features and opinion extraction, and for ontology construction, is in Algorithm 1.

4. Topic modeling and word embedding

In this section, we employ LDA and ontology-based topic modeling to identify transportation-related topics in preprocessed data. After that, word embedding algorithms (word2vec and glove2vec) along with lexicon2vec are used to convert words in the corpus into a vector format. The whole scenario is shown in Fig. 2.

Table 1
Topics and their words extracted from four different documents related to transportation.

Topics	Keywords
TP1	**Traffic**, **travel**, **bus**, **crash**, reply, relation, application, risk, **rail**, **pedestrian**, drugs, more, male…
Tp2	Closed, one-way, **road, jammed**, open, clean, light, medical, public, management, data, New-York…
TP3	**Jammed**, slow, head, single, **vehicle**, medium, **traffic**, collision, bridge, **car**, **bicycle**, crossing…
TP4	Delay, London, Britain, guardian, police, Wilmington, man, crossing, car, children, school, …

4.1. LDA-based topic modeling

LDA is a well-known unsupervised method in text mining. It can automatically identify the latent topics' "*TP*" from corpus data *D*. Fig. 3 shows a generative process of topic modeling. In LDA, each topic *TP* is made up of different words with probability distribution Φ_{TP}, where each document *d* is made up of different topics with probability distribution Θ_d. Both α and β are hyper-parameters to determine the probability of modeling, where α demonstrates the strength between implicit topics and documents, and β describes the probability distributions of all latent topics themselves. During the learning process, LDA performs two tasks. In the first task, it assigns words/terms to various topics in each document. In the second task, it allocates a high probability to a limited number of words in each topic [34]. But, these tasks create difficulties for each other. The first task may create difficulties if only a few words are assigned to each topic. The second task may cause difficulties if too many documents are assigned to a single topic. To adjust these tasks, a system can identify clusters of tightly co-occurring words. The following steps are used to model topic TP_d for every word W_{d_i} in corpus *d* [23,57].

- Consider *N* ∼Poisson (φ), where N denotes a sequence of words in document D.
- For each document D
  - Consider the topic distribution $\Theta_d \sim$ Dirichlet (α), where Dirichlet (α) is the Dirichlet distributions' parameter for α over implicit classes (e.g., 30% for topic TP1; 40% for topic TP2).
- For each *N* word W_{d_i}.
  - Sample topic $TP_d \sim$ Multinomial (Θ_d), P ($TP_{d,n}|\theta_d$).
  - Sample word W_{d_i} from $p\,(w_n|TP_d)$, a multinomial probability conditioned on the topic $TP_{d.}$

By using Gibbs sampling, we discover the document–topic and topic–word with probability distributions Θ and Φ, respectively. A K-dimensional Dirichlet random variable for probability density function α, and the mutual distribution of θ, TP, and *W* are computed by using the following three equations [34].

$$p\,(\theta|\alpha) = \frac{\Gamma\left(\sum_{i=1}^{k} \propto_i\right)}{\prod_{i=1}^{k} \Gamma(\propto_i)} \theta_1^{\alpha_{1-1}} \theta_2^{\alpha_{2-1}} \ldots . \theta_K^{\alpha_{k-1}} \tag{1}$$

$$p\,(\theta, TP, W|\alpha, \beta) = p\,(\theta|\alpha) \prod_{n=1}^{N} p\,(TP_n|\theta)\, P\,(W_n|TP_n, \beta) \tag{2}$$

$$p\,(D|\alpha, \beta) = \prod_{d=1}^{M} \int p(\theta_d|\alpha) \times \left(\prod_{n=1}^{N_d} \sum_{Z_{dn}} p\,(TP_{dn}|\theta_d)\, p(w_{dn}|TP_{dn}, \beta) \right) \mathrm{d}\theta_d \tag{3}$$

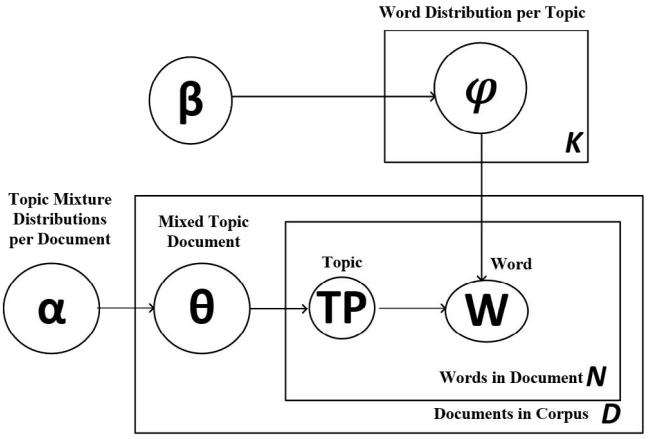

Fig. 3. LDA-based topic modeling.

where α and Γ(x) are the parameters of the k-vector and the gamma function, respectively.

The main aim of LDA is to learn the topic and the word distribution in the collected data. It neglects the order of words, and groups semantically related words into the same topic according to their representations in various documents. In our research, the Machine Learning for Language Toolkit (MALLET) LDA is applied to generate topics about transportation text, and different topic number *K*'s are tested in the topic modeling. Table 1 presents an example of topics and their keywords generated from traffic-related texts by using LDA. In Table 1, keywords in topics 1, 2, and 3 are related to the words used in traffic-related text. Therefore, these texts are highly relevant to the transportation feature *traffic*, whereas topic 4 contains irrelevant words, and it is a tough task to outline the results of the topic as reasonable concepts. However, an LDA-based constructed result comprises the following limitations.

- LDA-based generated topics contain feature words and opinion words. However, they also contain words that are not features of transportation when other traffic-related reviews include them.
- LDA avoids the association between topic and document when a document contains low-probability words.
- LDA is not able to learn semantic relationships among words. Moreover, it allocates a limited number of words to each topic, and allocates a limited number of topics to each document. Therefore, the vector of a document is inadequate.
- LDA generates very noisy topics when a text is very short. It also misses meaningful topics due to the inadequate dataset.

To overcome the above limitations, we propose an ontology of transportation features that improves the results of LDA and finds appropriate topics and features (words).

4.2. Transportation features ontology

An ontology is used to share domain knowledge among different systems. It is constructed in OWL and is widely used in text

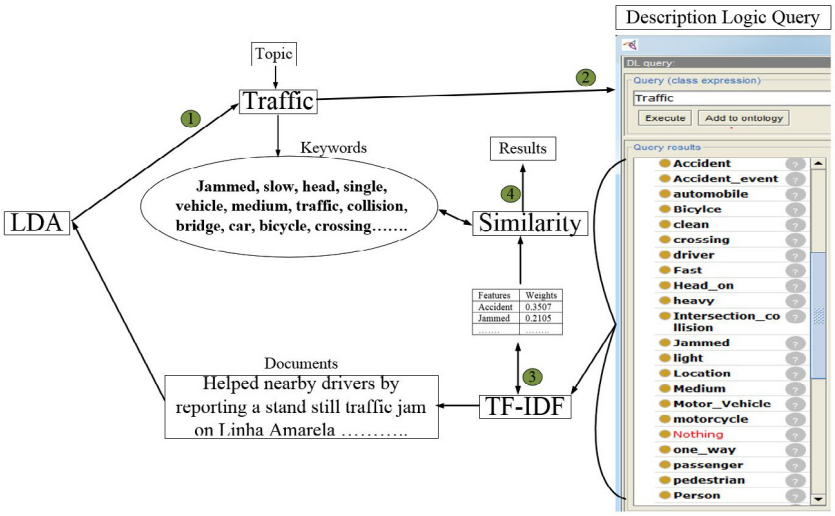

Fig. 4. Ontology classes and topic keywords.

classification studies [16,22,32,62]. The ontology is composed of five elements as follows:

$$\widetilde{\text{Ontology}} = (\text{C},\ \text{R},\ \text{A}, \text{V}_\text{c}, I) \tag{4}$$

where the notations C, R, A, V_c, and I stand for the set of domain concepts, relationships of domain concepts, rules and axioms, constraint values of properties, and instances, respectively. In ontology-based sentiment classification, a domain ontology is manually developed before feature extraction [21]. However, expert knowledge is important to confirm the semantic relationships among various concepts. Otherwise, the developed ontology would be vague and useless. In a transportation data context, crisp or classic ontologies become very large and difficult to use for feature extraction. Therefore, to develop the proposed transportation feature ontology, we constructed a classic ontology by employing Protégé OWL. A plug-in for fuzzy OWL was then applied to extend the classic ontology to a fuzzy ontology [63]. We collected valuable transportation knowledge from different transport-related websites about vehicles, climate, traffic, accidents, roads, and other features, and delivered it to the ontology. In our research, a fuzzy ontology is used for the following three main purposes.

- The fuzzy ontology is used to inspect the topic-level completeness of a transportation feature's description.
- The ontology contains a set of transportation-related entities and their relations, which can be used to extract entities from unstructured text.
- The fuzzy ontology can also be used to efficiently classify reviews and Tweets, and to compute the polarity of transportation features.

4.2.1. Ontology-based topic modeling and feature extraction

The proposed fuzzy ontology was applied to extract the appropriate transportation features from LDA-based generated clusters, and to identify a topic-level explanation. We used LDA to extract topics from transport-related data and applied a perplexity-based approach to identify the number of topics in the corpus data [62]. Description Logic (DL) query is used to extract features from the ontology to inspect the topic-level completeness. Each LDA-based generated topic is searched in the ontology using DL query to extract its semantically related features. The output of DL query contains various concepts or features, and each features has multiple sub-features and instances, which are stored in a database for further processing. To investigate the importance of these features, we utilized a numerical statistic approach called Term Frequency and Inverse Document Frequency (TF-IDF) [64,65]. In this work, a term is a feature of the ontology, and a document is the corpus data related to generated topic. We assigned weight to each feature using the following equations of term frequency (TP) and inverse document frequency (IDF);

$$TF = \frac{n_{fi,d}}{\sum_k n_{fk,d}} \tag{5}$$

$$IDF = \log \frac{|D|}{N_{fi}} \tag{6}$$

$$Weight_{fi} = TF * IDF \tag{7}$$

In the above equations, $n_{fi,d}$ indicates the number of existences of the ontology feature *fi* in the document *d*, $\sum_k n_{fk,d}$ indicates the sum of the existences of all the ontology features found in the document *d*, $|D|$ indicates the size of the set of all the documents in the corpus, and N_{fi} indicates the number of all documents associated with *fi*. The weight shows how relevant the ontology feature is for the document. After computing weight, the top X important features based on the weights of TF-IDF are considered for similarity measurement (e.g., X = 15). The Similarity function is defined as follows.

$$Similarity\,(OntoF, TK) = \sum_{i=1}^{n} f\,(ontoF_i, TK_i) \times weight_i \tag{8}$$

where *OntoF, TK, and* $weight_i$ represent features of the ontology, keywords of the generated topic, and importance of the feature *i*, respectively. When a similarity is detected and the score exceeded 4, the LDA-based generated topic is considered as a semantically associated with the document. The keywords are extracted from the topic cluster and stored with the obtained

transportation features. If the similarity score is zero or less than 4, then the second word in ranking based on LDA result is considered as a new topic of the generated cluster. The above procedure is repeated to investigate the topic-level completeness for the new topic. The whole scenario is shown in Fig. 4.

The domain ontology was difficult to construct. Therefore, we constructed the ontology in the form of dependent modules. For example, we took "traffic" as a sub-domain and created an ontology for it. The "traffic" topic description is then compared with the collected data in the corpus. After matching a large number of topic words with the ontology and corpus data, we use the topic along with keywords for further processing. Fig. 5 shows the topic modeling results in transportation features-related contents. Each cluster of topic includes a set of five closely co-occurring terms. The cluster of topic 1 includes the terms 'Crash,' 'Accident,' 'Traffic,' 'Speed,' and 'Event' and clearly shows the accident event data. Therefore, the proposed system declares that topic1 is about accident and topic-level completes regarding features description. Topic 2, 3, 4, 5, and 6 include terms, which are related to traffic congestion, driverless vehicle, traveling, road information, and airport passenger, respectively. The system labels these topics and confirms their topic-level completeness, as shown in Fig. 5. However, the similarity score between topic 7 terms and ontology features is below the threshold value. This shows that topic 7 is not a topic-level completeness, and there are no semantically relationships among terms. Therefore, the proposed system avoids topic 7 for further processing. A complete overview of the topic modeling procedure using LDA and the ontology is given in Algorithm 2.

Another main aim of the proposed system is to automate the text processing that needs a large number of event contexts. For example, in order to know that the sentence "*B20 was stopped due to some problem*" is related to transportation, it is important to infer that "B20" is a bus and the name is employed purposely in the context of transportation. The most efficient way is to provide background context for this kind of situation. Our proposed ontology contains a set of transportation-related entities and the relations that can be used for entity recognition during text processing. The term *B20* is represented in the ontology as an entity and linked with the concept of *bus*, whereas *bus* is connected with the feature *vehicle*. This creates relations of a text with transport features. The transportation features ontology is deeply discussed in the published paper [21]. The proposed ontology is also used with opinion lexicons for text analysis and polarity computation of transport features, which is explained in depth in Section 4.3.

Algorithm 2. Topic modeling using LDA and an ontology.

Data: Transportation corpus (TC) and ontology of transportation features
Results: Topics with relevant features
Begin
// topic modeling using LDA and ontology
Consider N sequence of words in document D
for each topic tp ϵ {1,2,..TP} **do**
  Select a word distribution $\phi_{tp} \sim Dirichlet(\beta)$;
end for
for each *doc* ϵ *corpus TC* **do**
  Consider the topic *distribution* $\Theta_d \sim Dirichlet(\alpha)$;
  for each N word of $doc(W_{d_i})$ **do**
    Sample a topic $TP_d \sim$ Multinomial (Θ_d);
    Sample a word W_{d_i} from $p(w_n \mid TP_d)$;
  end for
end for
// select topic that matches ontology classes and instances
for each topic tp
  relevant =0;
  for each word W_{d_i}
    if W_{d_i} in *text of TC* **then**
      relevant =relevant+1;
    end if
  end for
  if (relevant $\geq$ SVM (text)) **then**
    topic tp $\geq$ cluster of feature words;
  end if
end for
if topic tp ϵ ontology class && relevant ϵ ontology instances && relevant $\geq$ cluster of feature words **then**
  remove irrelevant words from cluster;
  tp $\leftarrow$ relevant features
end if

4.3. Deep learning and lexicon-based word embedding models

Generally, two methods are applied to convert document data into numerical data for advanced analysis: one-hot word representation and word embedding. In this section, we discuss five different types of document representation models and compare them with the proposed approach. These models are string2vec, word2vec, doc2vec, glove2vec, and embedded feature vectors that are presented by GloVe and Google [2,53,66].

4.3.1. string2word model

string2word enhances the one-hot representation method, called the bag-of-words model. It combines one-hot vectors in the text to represent the document. In this model, the corpus data is considered as a collection of disordered terms. Let us suppose a corpus C contains document $D = \{d_i, i = 1, 2, \ldots, n\}$, and m is a unique term existed in each document. Mathematically, the documents can be defined by $m \times n$ matrix S, where $S \in R^{m \times n}$. Each document and word are represented by column vector and row vector, respectively. If a term exists in the document, weight '1' is assigned to its corresponding place. Otherwise, the term weight is '0'. However, the string2word model has two limitations. The dimensionality of the document vector is dependent upon the existence of words in the document. Therefore, the dimensionality of string2word is high, which decreases the precision rate and is also time consuming. The second drawback is that it only symbolizes the count of the word and neglects the

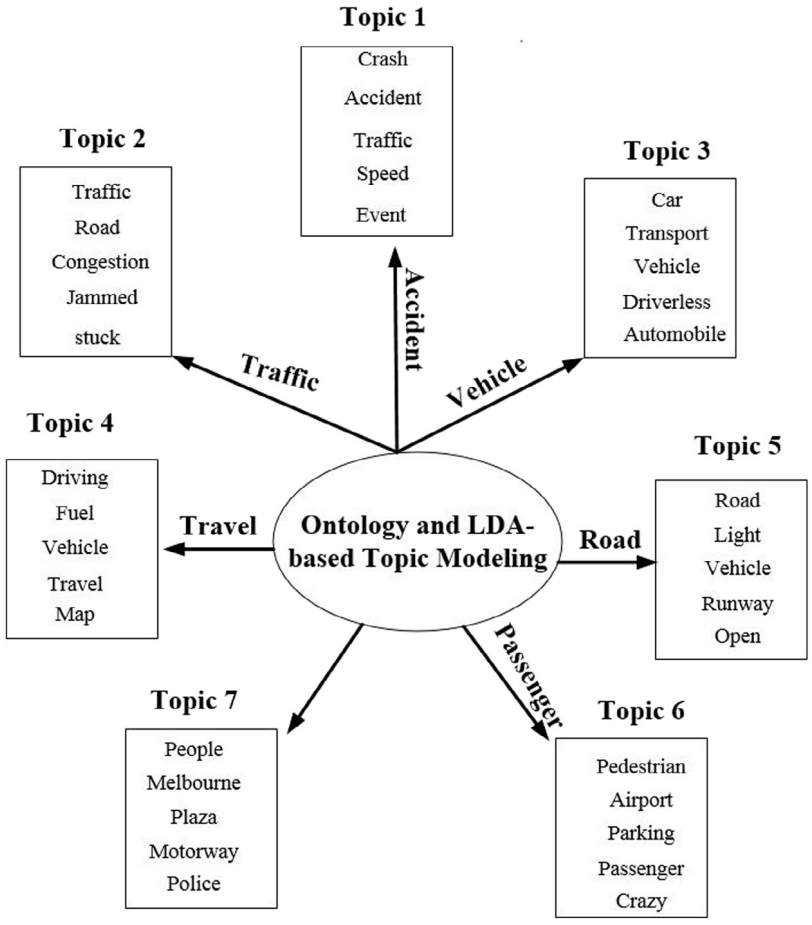

Fig. 5. Topic modeling results in transportation features-related contents.

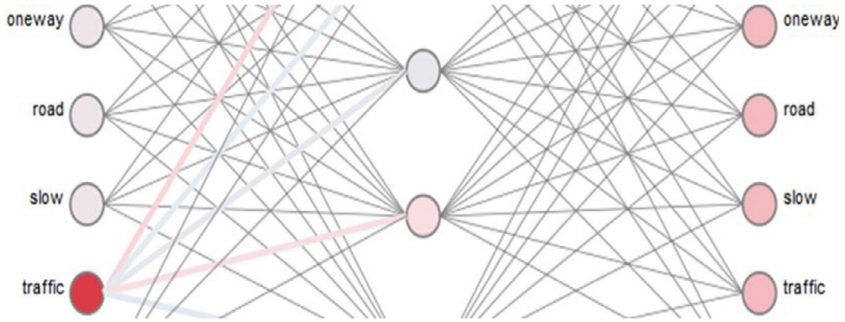

Fig. 6. Architecture of word2vec.

semantic relationships among words. We propose distribution representation approaches that represent each word with its semantic meanings and a very low-dimensional vector.

4.3.2. word2vec model

word2vec is a neural network-based model that studies word embedding from a large corpus of data. It generates a vector for each word in a high-dimensional space. word2vec includes two architectures to generate vector representation: continuous bag-of-words and skip-gram. The CBoW model predicts words using their surrounding context. The skip-gram model employs the current word to predict its surrounding context, as shown in Fig. 6. In this study, we trained and used skip-gram, because it works better for regular words. In skip-gram model, a sentence is represented in the form of the word: $W = \{w_{t-2}, w_{t-1}, w_t, w_{t+1}, w_{t+2}\} \in R^m$. The current word $(v(w_t))$ is an input vector to skip-gram, and the surrounding words $(\{v(w_{t-2}), v(w_{t-1}), v(w_{t+1}), v(w_{t+2})\} \in R^m)$ are the output of skip-gram. A contextual vector $v(x_w)$ is calculated by the following equation.

$$v(x_w) = \sum_{i=t-2}^{t+2} context\,(v(w_i)) \tag{9}$$

Compared with string2vec, the word2vec model can find the semantic relationship between two words; for example, *jammed* is very similar to the context *congestion*. Recently, word2vec was extended to calculate the average of word vectors, and was used as a document representation vector for each document.

4.3.3. Doc2vec model

The doc2vec model is an extension of word2vec. In doc2vec, paragraph vectors are combined at the input layer, which helps with the word vectors to examine the relationships of words and labels. It is an unsupervised method and represents a large corpus of data in the form of paragraph vectors. The paragraph vectors come in two types: a paragraph vector with distributed memory (PV-DM) and a paragraph vector with distributed bag-of-words (PV-DBoW) [50]. In PV-DM model, a paragraph id is added with word vectors during training. This paragraph id acts as a memory, which contains information that is missing from the current context. In PV-DBoW model, the paragraph id is input, which predicts randomly sampled words in the current context. However, both these types have limitations. PV-DM is complex but works better, and PV-DBoW is simple but neglects the order of words.

4.3.4. Global vector model

This is an unsupervised word embedding model introduced by Stanford University [53]. This model employs the statistics of a word-to-word relationship in the corpus to yield the vector representations for the corpus words. This model studies the exact vector representation for words that exist in similar contexts. The GloVe model builds a co-occurrence matrix from the corpus data by minimizing the average log-likelihood function as follows.

$$J = \sum_{i,j=1}^{V} f_0\left(X_{ij}\right)\left(W_i^T \tilde{W}_j + b_i + \tilde{b}_j - \log\left(X_{ij}\right)\right)^2. \tag{10}$$

where X_{ij} is the frequency of the word W_i with the word W_j, V is the size of the vocabulary in the corpus, W_i^T and $\tilde{W}_j$ are semantic vectors, b_i and $\tilde{b}_j$ are bias terms, $\log\left(X_{ij}\right)$ is the co-occurrence count of term i and j. To avoid log 0 errors, the GloVe model presents a weighting into the co-occurrence function of the model as follows [67].

$$f_0\left(X\right) = \begin{cases} \left(\left(x/x_{max}\right)^{\alpha}\right), & if\ x < x_{max} \\ 1, & otherwise \end{cases} \tag{11}$$

where x and α is the count of co-occurrence and the weighting for counts between 0 and x_{max}, respectively. The GloVe model is mostly used for similarity and feature recognition. We trained the GloVe model using transportation-related data in a corpus and more than 0.5 million words created with a 100-dimensional vector space. However, in each document, the average of the word vectors is computed in the word2vec model and employed for document representation vectors.

4.3.5. lexicon2vec model

The lexicon2vec model is based on emotion and sentiment lexicons. These lexicons are words and phrases with a polarity score that can be used for text analysis. Different sentiment and emotion lexicons have been proposed for sentiment analysis. However, the selection of the most proper lexicon is very important. We used five different lexicons along with our proposed fuzzy ontology-based lexicon system. These lexicons are as follows: the Multi-perspective Question Answering (MPQA) Opinion Corpus [68], the feature-based summary of customer reviews by Hu and Liu [69], AFINN lexicons [70], the National Research Council Canada (NRC) hashtag sentiment lexicon [47], the NRC emoticon lexicon [71], and our proposed fuzzy ontology-based lexicon [11,21,72]. The combination of efficient lexicons is difficult and plays a key role in sentiment classification. Therefore, different lexicons were tested, and we selected the best combinations from among them. For each word, these lexicons allocate six-dimensional vectors.

However, the above deep learning–based word embedding also examines imprecise words, which are not associated with the topic of the document. Our proposed LDA and ontology-based system removes these irrelevant words to enhance the document representation. Algorithm 3 describes the procedure of document representation in detail. We discussed the limitations of both LDA and word embedding in depth in the previous sections. In this work, we integrate LDA and word embedding models to overcome those limitations. LDA generates a universal correlation of each document to all topics, and the word embedding model studies the target words from their surroundings to capture the correlations. In addition, words for the topic are linked into the neural probability language model called topic2vec, which studies the topic representation in LDA by combining the context information from the word embedding model. We used the skip-gram model, which predicts the most likely surrounding words (w^{t-2}, w^{t-1}, w^{t+1}, w^{t+2}) given topic Tp^t and current word w^t. A sequence of words and the topic of a document, $D = \{w^1 : Tp^1, \ldots . w^M : Tp^M\}$, are given during training, and the learning function is maximized by using the following likelihood function [57]:

$$LF_{skip-gram}\left(D\right) = \frac{1}{M}\sum_{1}^{M}\sum_{-k\le c\le k, c\neq 0}\left(\log p\left(w^{i+Context}|Tp^i\right) + \log p\left(w^{i+Context}|w^i\right)\right). \tag{12}$$

To improve the accuracy of the proposed document representation vectors, we combined word embedding and lexicon-based approaches, which are discussed in Section 4.3.

Algorithm 3. Document representation using the word embedding approach and topic modeling results.

Data: Pre-processed transportation text corpus and topic modeling results
Results: Document representation and transportation polarity classification
Begin

```
for each doc ϵ corpus TC do
    for each ( W_di )ϵ doc do
        vocabulary ← words;
    end for
    for each word ϵ vocabulary do
        Vector_word ← word2vec_word;
        (Use TC corpus to train word2vec and
        vectorize word) ({V(w_1), V(w_2)…. V(w_n);
    end for
    for each topic tp ϵ LDA and ontology outputs
                {t_1,t_2….t_T,} do
        w_i = topic distribution θ_i / Σ_{n=1}^{h} topic distribution θ_i ;
            (select high-probability word)
        Sum the product of each V(w) and its weight
        W_i to calculate topic2vec ;
    end for
    for each v(d_i) ϵ document vector do
        doc_vector ← vector_word/total words in doc;
    end for
end for
for each doc ϵ corpus TC do
    doc_lexicon ← lexicon2vec;
end for
Combine doc_vector and topic2vec with doc_lexicon to represent documents;
Apply ML algorithms to classify sentiment;
```

Table 2
Results of ML classifiers applied to string2word vector.

Classifiers	Avg. precision (%)	Avg. recall (%)	Avg. function measure (%)	Accuracy (%)	RMSE	MAE
Lib SVM	74	84	78	**75**	0.50	0.25
Logistic regression	77	75	76	74	0.49	0.25
MLP	72	67	70	68	0.52	0.32
KNN	61	71	66	60	0.62	0.39
Naïve Bayes	70	70	70	67	0.51	0.34
Decision tree	66	75	70	66	0.52	0.36
Deep learning	69	76	72	68	0.46	0.35

Table 3
Results of ML classifiers applied to word2vec.

Classifiers	Avg. precision (%)	Avg. recall (%)	Avg. function measure (%)	Accuracy (%)	RMSE	MAE
Lib SVM	74	77	76	**73**	0.42	0.36
Logistic regression	72	74	73	71	0.48	0.30
MLP	73	75	74	71	0.51	0.30
KNN	65	60	63	61	0.62	0.39
Naïve Bayes	69	70	70	67	0.47	0.38
Decision tree	61	74	67	61	0.52	0.44
Deep learning	74	73	74	72	0.44	0.32

5. Experiments

The dataset used in the evaluation process was discussed in Section 3. The proposed approach was presented in Section 4. Here, the validation procedure is defined and the obtained results are discussed.

5.1. Performance evaluation

We used Waikato Environment for Knowledge Analysis (Weka) [73,74] for the evaluation of word embedding approaches using different classifiers. Weka is a very popular API and ML software that is used to train word embedding models. We compare the performance of string2vec, word2vec, doc2vec, glove2vec, and lexicon2vec with our proposed model. We used ML algorithms to classify the data of these models and reported the results. The input data for ML classifiers include numerical word vectors extracted from the corpus data by using a word embedding approach, topical words extracted with LDA and ontology-based methods, and lexicon vectors extracted from six sentiment lexicons. We developed this system by using Weka and the Protégé OWL tool with Java, and we trained the model by randomly dividing the dataset into training and testing sets, at 60% and 40%, respectively. These above-mentioned models were evaluated with the following baselines.

- Support vector machine: We utilized Libsvm, which classifies data with a linear kernel. It permits sparse training data so the training dataset comprises non-zero values. It is adjustable for sparse data but took a longer time during training. We applied Libsvm with a training parameter kernel-type radial basis function.
- Logistic regression: Based on certain predictors, categorical outcomes can be predicted by regression analysis. It uses a link function that converts the probability of limited range [0,1] into $(-\infty, +\infty)$. We used a multinomial logistic regression model with the training parameter ridge estimator.
- Multilayer perceptron: The Weka library was used to implement MLP and uses a sigmoid activation function and a squared error function for the evaluation of word-embedding approaches.
- K-Nearest Neighbors (KNN): The KNN algorithm stores all cases and is based on similarity measures; it classifies new cases. We used KNN at $k = 3$ and with a Euclidean distance function.
- Naïve Bayes: This is a probabilistic classifier based on Bayes theorem. We use multinomial NB for text data, and both bigrams and unigrams are measured as terms in categorization.
- Decision tree: C4.5 is the advanced version of the ID3 algorithm. It uses the concept of information entropy to build a decision tree from the training dataset. We used it with confidence factor 0.25 and seed 1.
- Deep learning: The Deeplearning4j library of Weka and its default options were used in our work to perform classification with deep neural networks.

For each classifier, we measured the average precision, recall, function measure, residual mean square error (RMSE), and mean absolute error (MAE) over the testing datasets. The same classifier and matrices were used to evaluate each word-embedding model against our proposed model.

5.2. Results

Here, we present results based on the experiments described in Section 5.1. Table 2 shows a summary of the results obtained from six baselines for different word-embedding models in the sentiment classification task. The first column is the name of the classifiers. The next three columns show the average for precision, recall, and function measure, while the fifth column contains the accuracy percentages calculated overall for the embedding models. Finally, the sixth and seven columns present the RMSE and MAE, respectively, calculated via the testing datasets. We compared the accuracies obtained by the classifiers to evaluate the performance of the embedding models.

In Table 2, different ML classifiers are compared against each other to examine the performance of word embedding models. Table 2 presents the results obtained by classifiers applied to the traditional string2word model. SVM and logistic regression gained higher accuracy at 75% and 74%, respectively. Five other classifiers achieved lower accuracy in comparison to SVM and logistic regression. However, RMSE with deep learning is lower than other classifiers. Based on our experiments, we observed that string2word is the most time-consuming for all classifiers

Table 4
Results of ML classifiers applied to doc2vec.

Classifiers	Avg. precision (%)	Avg. recall (%)	Avg. function measure (%)	Accuracy (%)	RMSE	MAE
Lib SVM	72	78	74	71	0.44	0.39
Logistic regression	74	79	76	**74**	0.45	0.29
MLP	71	78	74	70	0.50	0.30
KNN	66	79	72	66	0.55	0.35
Naïve Bayes	63	80	70	64	0.56	0.36
Decision tree	63	79	70	63	0.51	0.42
Deep learning	73	80	76	73	0.44	0.33

Table 5
Results of ML classifiers applied to glove2vec.

Classifiers	Avg. precision (%)	Avg. recall (%)	Avg. function measure (%)	Accuracy (%)	RMSE	MAE
Lib SVM	61	78	68	61	0.48	0.46
Logistic regression	68	69	68	**66**	0.51	0.37
MLP	63	66	64	61	0.59	0.39
KNN	60	72	65	59	0.54	0.44
Naïve Bayes	56	82	66	55	0.50	0.49
Decision tree	58	55	56	54	0.60	0.47
Deep learning	67	73	70	**66**	0.48	0.42

Table 6
Results of ML classifiers applied to lexicon-based features.

Classifiers	Avg. precision (%)	Avg. recall (%)	Avg. function measure (%)	Accuracy (%)	RMSE	MAE
Lib SVM	60	92	73	63	0.47	0.44
Logistic regression	68	87	77	72	0.44	0.38
MLP	71	82	76	73	0.43	0.36
KNN	67	82	74	68	0.52	0.35
Naïve Bayes	66	90	76	69	0.50	0.31
Decision tree	67	84	74	68	0.46	0.40
Deep learning	69	88	77	**72**	0.44	0.38

Table 7
Sentiment analysis results of the public transportation dataset using our proposed approach.

Classifiers	Avg. precision (%)	Avg. recall (%)	Avg. function measure (%)	Accuracy (%)	RMSE	MAE
Lib SVM	80	87	84	82	0.35	0.26
Logistic regression	82	86	84	**83**	0.36	0.22
MLP	80	82	81	80	0.43	0.20
KNN	74	76	75	72	0.52	0.27
Naïve Bayes	73	87	79	75	0.46	0.24
Decision tree	74	78	76	74	0.48	0.26
Deep learning	91	84	88	**90**	0.30	0.17

due to its high dimensionality. We applied word2vec to the transportation dataset and report the results obtained by the classifiers in Table 3. We observed that SVM and deep learning outperform in measured accuracy at 73% and 72%, respectively. In comparison with the results of the string2word model, the accuracy of SVM and logistic regression decreased 2% and 3%, respectively, whereas deep learning accuracy increased, and RMSE decreased. As shown in Tables 4 and 5, the same classifiers are used for the doc2vec and glove2vec models to handle dataset classification. With doc2vec, the measured accuracy of logistic regression and deep learning was 74% and 73%, respectively, which is high with respect to other classifiers, whereas RMSE from deep learning is the same as that measured by using word2vec. The glove2vec model, performed surprisingly poorly; deep learning achieved the highest accuracy (66%) among all the classifiers. However, the RMSE of all classifiers increased highly, compared with the results of string2vec, word2vec, and doc2vec. Before using lexicon2vec with word2vec, we applied lexicon2vec individually to the dataset, and the results obtained by the baselines are presented in Table 6. Note that the results produced from lexicion2vec are better than string2word and glove2vec in terms of function measure and accuracy, but lower than the doc2vec-based word embedding model. The dimensionality of lexicon2vec is very low, and it does not affect the time of execution when combined with another embedding model. Based on the experiment, the function measure of lexicon2vec is higher than the previous embedding model.

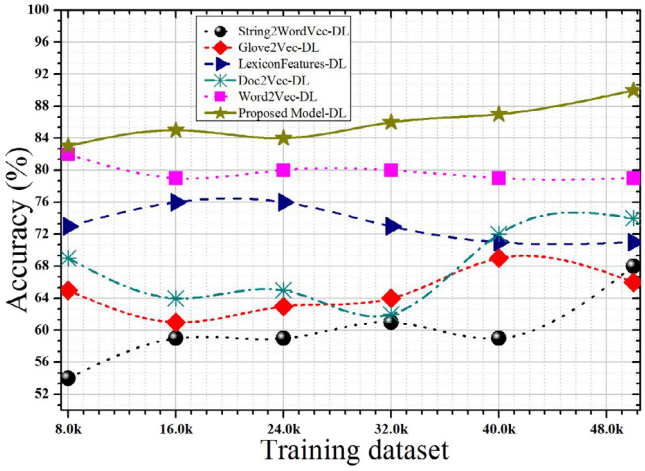

Fig. 7. Accuracy of different embedding models compared with the proposed model.

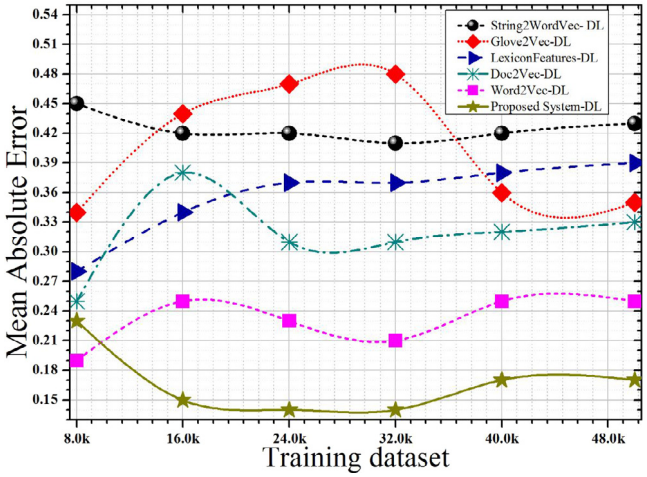

Fig. 8. MAE comparisons of different embedding models with the proposed model.

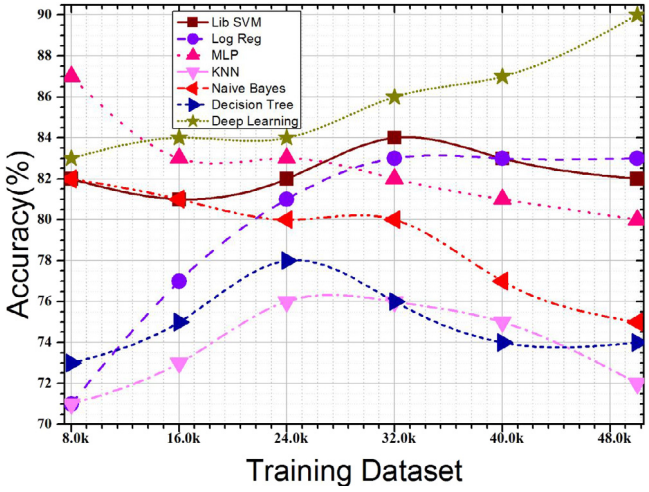

Fig. 9. Accuracy comparison of various classifiers on the proposed model for a transportation dataset.

We integrated the results of topic modeling and lexicon2vec into the word2vec model based on transportation data, and applied the classifiers to categorize its data. According to Table 7, the accuracy of our proposed approach is higher than the previously mentioned models. In particular, the proposed approach increased the accuracy of sentiment classification. For sentiment classification of the transportation data, the best result was achieved by the deep learning classifier in our proposed approach-based word embedding model, as it can be seen in Table 7. The obtained accuracy and function measure are 90% and 88%, respectively. By examining the precision and recall values, we see that the precision rate increased with the proposed method, and the recall rate decreased. It is also noted that the proposed approach-based word embedding registered a higher RMSE with KNN, decision tree, and Naïve Bayes.

5.3. Accuracy and error analysis of deep learning in classification of word embedding models data

Fig. 7 shows the classification accuracy of deep learning with the pre-trained word embedding models. It is a dataset totaling 50,000 items, including reviews, Tweets, and sentences from documents. As can be seen, the proposed approach has the highest accuracy, and glove2vec has the lowest accuracy among all embedding models. The proposed method's accuracy is higher by 24% with respect to glove2vec, by 22% with respect to string2word, and by more than 16% to 18% with respect to the other embedding models. With the dataset increasing in size, the proposed model provides an accuracy rate as good as 90%, while the rates of other models declined or increased just a little. Based on this experiment, it is noted that the proposed approach performs better with document representation and deep learning for sentiment classification.

Fig. 8 shows the details from error analysis by the word embedding models. We may perceive how deep learning tends to provide false classifications for the mentioned embedding models. The figure shows an excellent performance index for the proposed model. With an increased size for the training dataset, the proposed model shows a much lower MAE (0.17), compared to the other models; the MAE of lexicon2vec, doc2vec, and word2vec increased suddenly, while the MAE for string2vec-DL and glove2vec-DL changed little.

Fig. 9 shows the accuracy of ML algorithms in classifying our proposed model-based text representation. The performance of KNN, Naïve Bayes, and decision tree are surprisingly poor. The result of KNN is affected by the value of K. It misclassifies the test sample if the value of K is too large. This is due to some data points that are far away from the test sample neighborhood. The Naïve Bayes is unable to study the relations between features, and has numerical instabilities due to zero conditional probability problem, which worse the results. Decision tree is complex in representation for some concepts, and has overfitting problem when exists noise in the dataset [75]. In the testing data, KNN, Naïve Bayes, and decision tree registered lower accuracy compared to the proposed model. As seen in Fig. 9, with an increased size for the training dataset, the average accuracy for deep learning and logistic regression, respectively, increased drastically and gradually. But the accuracy of the other classifiers declined rapidly. After in-depth analysis of the results of these three classifiers, it is found that these ML algorithms failed to understand and classify lengthy comments containing more than one sentiment-based sentence. According to Fig. 9, we can see how the efficiency obtained by logistic regression is very close to that obtained by deep learning. Indeed, the accuracy of logistic regression is less than 7%. PCA along with the classifiers reduced the dimensionality of the dataset, as shown in Fig. 10. The accuracy with deep learning and MLP increased by 3% and 1%, respectively. The accuracy of other classifiers was not changed.

The proposed method overcomes almost all others in terms of function measure and accuracy, in comparison with string2word, word2vec, doc2vec, glove2vec, and lexicon2vec. The complexity of the proposed system is very low, compared with string2word and glove2vec. However, in comparison with doc2vec and word2vec, the proposed system requires two extra steps: ontology- and LDA-based topic modeling and lexicon2vec model.

6. Conclusion

In this paper, we presented an ontology and LDA-based topic modeling and word embedding system to enhance the performance of document representation and sentiment classification, and to facilitate mobility users and ITSs. Various sensible issues are discussed, including valuable-information extraction, transformation of extracted data into useful knowledge, generation of topics and features using an ontology and LDA, representation of documents under different approaches, and integration of lexicons into word-embedding models. The proposed approach

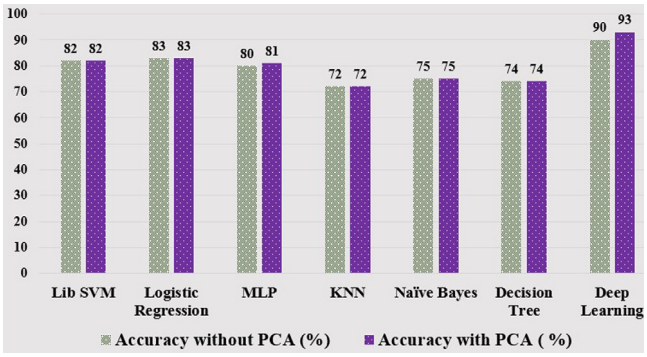

Fig. 10. Proposed model accuracy with PCA and without PCA.

offers a text classification system that identifies the most relevant transportation texts in social media and analyzes them to examine traffic control management and transportation services. Indeed, our proposed system efficiently classifies extremely ambiguous text and determines transportation polarity. This new approach not only improves the performance of LDA but also outperforms topic2vec document representation methods with transportation datasets. It integrates lexicons into a pre-trained word embedding model that increases the accuracy of sentiment classification. It can also represent each word in the corpus with a low-dimensional vector and with semantic meaning. Furthermore, this approach can be linked to different information extraction, text mining, and sentiment analysis systems, since it can generate topics, features, and opinions from unclear text, and represents these extracted data accurately in order to improve the performance of sentiment classification. This approach can be helpful for medical application. For example, the increase of chronic patients in hospitals is due to adverse drug reaction. The LDA and biomedical ontologies are highly valuable to extract drug reaction information from online contents and can be utilized to identify the most common cause. This method can also be used in social robotics application. For example, social robotic technology is used as an interface to provide a recommendation for hotels, movies, books, or restaurants. Such a recommendation depends on the available information in the system. The proposed approach can help the social robotic extract efficient information using ontology, and inspect the correctness of items by identifying their popularity on the Internet.

In future directions for our research, we will enhance classification performance by analyzing different traffic- and travel-related content. We will make it possible to solve traffic congestion issues in big cities and to improve transportation facilities using social network content. The existing research considers irrelevant words as sentiment words, which decreases the precision rate of a classification task. We will preprocess the data in a more sophisticated way to provide the required precision rate for successful sentiment classification.

Acknowledgment

This research was supported by the Ministry of Science, ICT and Future Planning (MSIP), South Korea, under the ITRC support program (IITP-2017-2014-0-00729) supervised by the Institute for Information & communications Technology Promotion (IITP).

References

[1] X. Dai, M. Bikdash, B. Meyer, From social media to public health surveillance: Word embedding based clustering method for twitter classification, in: SoutheastCon 2017, 2017, pp. 1–7, http://dx.doi.org/10.1109/SECON.2017.7925400.

[2] Q.V. Le, T. Mikolov, Distributed Representations of Sentences and Documents, Vol. 32, 2014, http://dx.doi.org/10.1145/2740908.2742760.

[3] M.D.P. Salas-Zárate, J. Medina-Moreira, K. Lagos-Ortiz, H. Luna-Aveiga, M.Á. Rodríguez-García, R. Valencia-García, Sentiment analysis on tweets about diabetes: An aspect-level approach, Comput. Math. Methods Med. 2017 (2017) http://dx.doi.org/10.1155/2017/5140631.

[4] C. Clavel, Z. Callejas, Sentiment analysis: From opinion mining to human-agent interaction, IEEE Trans. Affect. Comput. 7 (2016) 74–93, http://dx.doi.org/10.1109/TAFFC.2015.2444846.

[5] A. Krouska, C. Troussas, M. Virvou, Comparative evaluation of algorithms for sentiment analysis over social networking services, J. UCS 23 (2017) 755–768.

[6] Y. Shibuya, Public Sentiment and Demand for Used Cars after A Large-Scale Disaster : Social Media Sentiment Analysis with Facebook Pages, 2018.

[7] A. Teixeira, Data extraction and preparation to perform a The example of a Facebook fashion brand page, (n.d.).

[8] F.B. Marquez, Acquiring and Exploiting Lexical Knowledge for Twitter Sentiment Analysis, Vol. 1994, 2017.

[9] J. Song, K.T. Kim, B. Lee, S. Kim, H.Y. Youn, A novel classification approach based on Naïve Bayes for Twitter sentiment analysis, Vol. 11, 2017, pp. 2996–3012, http://dx.doi.org/10.3837/tiis.2017.06.011.

[10] R.Y.K. Lau, C. Li, S.S.Y. Liao, Social analytics: Learning fuzzy product ontologies for aspect-oriented sentiment analysis, Decis. Support Syst. 65 (2014) 80–94, http://dx.doi.org/10.1016/j.dss.2014.05.005.

[11] F. Ali, D. Kwak, P. Khan, S.H.A. Ei-sappagh, S.M.R. Islam, D. Park, Merged Ontology and SVM-Based Information Extraction and Recommendation System for Social Robots, Vol. 5, 2017, pp. 1–16.

[12] C. Chang, C. Lin, LIBSVM : A library for support vector machines, ACM Trans. Intell. Syst. Technol. (TIST) 2 (2013) 1–39, http://dx.doi.org/10.1145/1961189.1961199.

[13] V. Effendy, A. Novantirani, M.K. Sabariah, Sentiment Analysis on Twitter about the Use of City Public Transportation Using Support Vector Machine Method, 2011.

[14] S. Agarwal, P. Kachroo, E. Regentova, A hybrid model using logistic regression and wavelet transformation to detect traffic incidents, IATSS Res. 40 (2016) 56–63, http://dx.doi.org/10.1016/j.iatssr.2016.06.001.

[15] L. Gatti, M. Guerini, M. Turchi, SentiWords: Deriving a high precision and high coverage lexicon for sentiment analysis, IEEE Trans. Affect. Comput. 7 (2016) 409–421, http://dx.doi.org/10.1109/TAFFC.2015.2476456.

[16] D.T. Santosh, K.S. Babu, S.D.V. Prasad, A. Vivekananda, Opinion mining of online product reviews from traditional LDA topic clusters using feature ontology tree and sentiwordnet, Int. J. Educ. Manag. Eng. 6 (2016) 34–44, http://dx.doi.org/10.5815/ijeme.2016.06.04.

[17] W. Zhao, Z. Guan, L. Chen, X. He, D. Cai, B. Wang, Q. Wang, Weakly-supervised deep embedding for product review sentiment analysis, IEEE Trans. Knowl. Data Eng. 30 (2017) 185–197, http://dx.doi.org/10.1109/TKDE.2017.2756658.

[18] M. Dragoni, G. Petrucci, A neural word embeddings approach for multi-domain sentiment analysis, IEEE Trans. Affective Comput. 8 (2017) 457–470, http://dx.doi.org/10.1109/TAFFC.2017.2717879.

[19] M. Dragoni, G. Petrucci, A fuzzy-based strategy for multi-domain sentiment analysis, Int. J. Approx. Reason. 93 (2018) 59–73, http://dx.doi.org/10.1016/j.ijar.2017.10.021.

[20] F.C. Pereira, F. Rodrigues, E. Polisciuc, M. Ben-akiva, Transport overcrowding with internet data, IEEE Trans. Intell. Transp. Syst. 16 (2015) 1–10, http://dx.doi.org/10.1109/TITS.2014.2368119.

[21] F. Ali, D. Kwak, P. Khan, S.M.R. Islam, K. Hyun, K.S. Kwak, Fuzzy ontology-based sentiment analysis of transportation and city feature reviews for safe traveling q, Transp. Res. Part C 77 (2017) 33–48, http://dx.doi.org/10.1016/j.trc.2017.01.014.

[22] S.M. Grant-muller, A. Gal-tzur, E. Minkov, S. Nocera, T. Ku, I. Shoor, Enhancing Transport Data Collection Through Social Media Sources: Methods, Challenges and Opportunities for Textual Data, 2014, pp. 1–11, http://dx.doi.org/10.1049/iet-its.2013.0214.

[23] S. Das, X. Sun, A. Dutta, Text mining and topic modeling of compendiums of papers from transportation research board annual meetings, Transp. Res. Rec.: J. Transp. Res. Board 2552 (2016) 48–56, http://dx.doi.org/10.3141/2552-07.

[24] L. Abberley, N. Gould, K. Crockett, J. Cheng, Modelling road congestion using ontologies for big data analytics in smart cities, in: 2017 International Smart Cities Conference, ISC2 2017, 2017, http://dx.doi.org/10.1109/ISC2.2017.8090795.

[25] J.F.F. Pereira, Social Media Text Processing and Semantic Analysis for Smart Cities, 2017.

[26] S.M. Riazul Islam, M. Nazim Uddin, K.S. Kwak, The IoT: Exciting possibilities for bettering lives: Special application scenarios, IEEE Consum. Electron. Mag. 5 (2016) 49–57, http://dx.doi.org/10.1109/MCE.2016.2516079.

[27] A. Valdivia, M.V. Luzón, E. Cambria, F. Herrera, Consensus vote models for detecting and filtering neutrality in sentiment analysis, Inf. Fusion 44 (2018) 126–135, http://dx.doi.org/10.1016/j.inffus.2018.03.007.

[28] K. Ali, H. Dong, A. Bouguettaya, A. Erradi, R. Hadjidj, Sentiment analysis as a service: A social media based sentiment analysis framework, in: Proceedings - 2017 IEEE 24th International Conference on Web Services, ICWS 2017, 2017, pp. 660–667, http://dx.doi.org/10.1109/ICWS.2017.79.
[29] C. Lin, P. Chao, Opinion Target Identification Focusing on the Tourist Attractions, 2010, pp. 3–16.
[30] P. Nakov, A. Ritter, S. Rosenthal, F. Sebastiani, V. Stoyanov, SemEval-2016 Task 4: Sentiment analysis in twitter, in: Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016), 2016, pp. 1–18, http://dx.doi.org/10.18653/v1/S16-1001.
[31] S. De Kok, K. Schouten, R. Van Den Puttelaar, Review-Level Aspect-Based Sentiment Analysis Using an Ontology, 2018.
[32] W. Hong, Z. Hao, S. Jinchuan, Research and application on domain ontology learning method based on LDA, J. Softw. 12 (2017) 265–273, http://dx.doi.org/10.17706/jsw.12.4.265-273.
[33] Y. Zhang, J. Ma, Z. Wang, Semi supervised classification of scientific and technical literature based on semi supervised hierarchical description of improved latent dirichlet allocation (LDA), Cluster Comput. (2018) http://dx.doi.org/10.1007/s10586-017-1674-x.
[34] G. Ren, T. Hong, Investigating online destination images using a topic-based sentiment analysis approach, Sustainability (Switzerland). 9 (2017) http://dx.doi.org/10.3390/su9101765.
[35] A. Onan, S. Korukoglu, H. Bulut, LDA-based Topic modelling in text sentiment classification: An empirical analysis, Int. J. Comput. Linguist. Appl. 7 (2016) 101–119.
[36] Y. Ren, R. Wang, D. Ji, A topic-enhanced word embedding for twitter sentiment classification, Inform. Sci. 369 (2016) 188–198, http://dx.doi.org/10.1016/j.ins.2016.06.040.
[37] N. Ko, B. Jeong, S. Choi, J. Yoon, Identifying product opportunities using social media mining: Application of topic modeling and chance discovery theory, IEEE Access XX (2017) http://dx.doi.org/10.1109/ACCESS.2017.2780046.
[38] F. Ali, D. Kwak, P. Khan, S.H.A. Ei-Sappagh, S.M.R. Islam, D. Park, K.S. Kwak, Merged ontology and SVM-Based information extraction and recommendation system for social robots, IEEE Access 5 (2017) 12364–12379, http://dx.doi.org/10.1109/ACCESS.2017.2718038.
[39] M. Katsumi, M. Fox, Ontologies for transportation research: A survey, Transp. Res. Part C 89 (2018) 53–82, http://dx.doi.org/10.1016/j.trc.2018.01.023.
[40] D. Yue, S. Wang, A. Zhao, Traffic accidents knowledge management based on ontology, in: 2009 Sixth International Conference on Fuzzy Systems and Knowledge Discovery, 2009, pp. 447–449, http://dx.doi.org/10.1109/FSKD.2009.134.
[41] C.F. Map, 교통망 관찰과 도시 특징지도를 위한 퍼지영역 온톨로지 기반 오피니언 마이닝, 15 (2016) 109–118.
[42] R. Zhao, K. Mao, Fuzzy Bag-of-Words model for document representation, IEEE Trans. Fuzzy Syst. 14 (2017) http://dx.doi.org/10.1109/TFUZZ.2017.2690222, 1–1.
[43] A. Kakade, K. Dhumal, A Neural Network Approach for Text Document Classification and Semantic Text Analytics, Vol. 2, 2017, pp. 1–6.
[44] M. Shi, J. Liu, D. Zhou, M. Tang, B. Cao, WE-LDA: A word embeddings augmented LDA model for web services clustering, in: 2017 IEEE International Conference on Web Services (ICWS), 2017, pp. 9–16, http://dx.doi.org/10.1109/ICWS.2017.9.
[45] A. García-Pablos, M. Cuadros, G. Rigau, W2VLDA: Almost unsupervised system for aspect based sentiment analysis, Expert Syst. Appl. 91 (2018) 127–137, http://dx.doi.org/10.1016/j.eswa.2017.08.049.
[46] X. Wang, H. Zhang, Z. Xu, Public sentiments analysis based on fuzzy logic for text, Int. J. Softw. Eng. Knowl. Eng. 26 (2016) 1341–1360, http://dx.doi.org/10.1142/S0218194016400076.
[47] S.M. Mohammad, S. Kiritchenko, X. Zhu, NRC-Canada: Building the State-of-the-Art in Sentiment Analysis of Tweets, 2013.
[48] S. Ruder, P. Ghaffari, J.G. Breslin, INSIGHT-1 at SemEval-2016 Task 4: Convolutional Neural Networks for Sentiment Classification and Quantification, 2016, pp. 178–182.
[49] L. MA, A Multi-label Text Classification Framework : Using Supervised and Unsupervised Feature Selection Strategy, 2017.
[50] M. Kamkarhaghighi, M. Makrehchi, Content tree word embedding for document representation, Expert Syst. Appl. 90 (2017) 241–249, http://dx.doi.org/10.1016/j.eswa.2017.08.021.
[51] D. Sarkar, R. Bali, T. Sharma, Analyzing Movie Reviews Sentiment, 2018.
[52] A. Mohasseb, M. Bader-El-Den, H. Liu, M. Cocea, Domain specific syntax based approach for text classification in machine learning context, in: 2017 International Conference on Machine Learning and Cybernetics (ICMLC), Vol. 2, 2017, pp. 658–663, http://dx.doi.org/10.1109/ICMLC.2017.8108983.
[53] 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, http://dx.doi.org/10.3115/v1/D14-1162.
[54] V. Jayawardana, D. Lakmal, N. de Silva, A.S. Perera, K. Sugathadasa, B. Ayesha, Deriving a Representative Vector for Ontology Classes with Instance Word Vector Embeddings, 2017, http://dx.doi.org/10.1109/INTECH.2017.8102426.
[55] Z. Hu, J. Hu, W. Ding, X. Zheng, Review sentiment analysis based on deep learning, in: 2015 IEEE 12th International Conference on E-Business Engineering, 2015, pp. 87–94, http://dx.doi.org/10.1109/ICEBE.2015.24.
[56] S.M. Rezaeinia, A. Ghodsi, R. Rahmani, Improving the Accuracy of Pre-trained Word Embeddings for Sentiment Analysis, (n.d.).
[57] L. Niu, X. Dai, J. Zhang, J. Chen, Topic2Vec: Learning Distributed Representations of Topics, 2015, pp. 193–196.
[58] E.H.J. Kim, Y.K. Jeong, Y. Kim, K.Y. Kang, M. Song, Topic-based content and sentiment analysis of Ebola virus on Twitter and in the news, J. Inf. Sci. 42 (2015) 763–781, http://dx.doi.org/10.1177/0165551515608733.
[59] S. Ahmed, S. Haman, E. Atwell, F. Ahmed, Aspect based sentiment analysis framework using data from social media network, IJCSNS Int. J. Comput. Sci. Netw. Secur. 17 (2017) 100–105.
[60] F. Ali, E.K. Kim, Y.G. Kim, Type-2 fuzzy ontology-based opinion mining and information extraction: A proposal to automate the hotel reservation system, Appl. Intell. 42 (2015) 481–500, http://dx.doi.org/10.1007/s10489-014-0609-y.
[61] F. Ali, K.-S. Kwak, Y.-G. Kim, Opinion mining based on fuzzy domain ontology and support vector machine: A proposal to automate online review classification, Appl. Soft Comput. 47 (2016) 235–250.
[62] R. Chen, Y. Zheng, W. Xu, M. Liu, J. Wang, Secondhand seller reputation in online markets: A text analytics framework, Decis. Support Syst. (2018) http://dx.doi.org/10.1016/j.dss.2018.02.008, #pagerange#.
[63] F. Bobillo, U. Straccia, Fuzzy ontology representation using OWL 2, Internat. J. Approx. Reason. 52 (2011) 1073–1094, http://dx.doi.org/10.1016/j.ijar.2011.05.003.
[64] M. Chandrashekar, R. Nagulapati, Y. Lee, Ontology mapping framework with feature extraction and semantic embeddings, in: Proceedings - 2018 IEEE International Conference on Healthcare Informatics Workshops, ICHI-W 2018, 2018, pp. 34–42, http://dx.doi.org/10.1109/ICHI-W.2018.00012.
[65] M.Á. Rodríguez-García, R. Valencia-García, F. García-Sánchez, J.J. Samper-Zapater, Ontology-based annotation and retrieval of services in the cloud, Knowl.-Based Syst. 56 (2014) 15–25, http://dx.doi.org/10.1016/j.knosys.2013.10.006.
[66] T. Mikolov, K. Chen, G. Corrado, J. Dean, Efficient Estimation of Word Representations in Vector Space, 2013, pp. 1–12, http://dx.doi.org/10.1162/153244303322533223.
[67] H. Peng, M. Bao, J. Li, M.Z.A. Bhuiyan, Y. Liu, Y. He, E. Yang, Incremental term representation learning for social network analysis, Future Gener. Comput. Syst. 86 (2018) 1503–1512, http://dx.doi.org/10.1016/j.future.2017.05.020.
[68] T. Wilson, J. Wiebe, P. Hoffmann, Recognizing contextual polarity in phrase-level sentiment analysis, in: Proceedings of the Conference on Human Language Technology and Empirical Methods in Natural Language Processing - HLT'05, 2005, pp. 347–354, http://dx.doi.org/10.3115/1220575.1220619.
[69] M. Hu, B. Liu, Mining and summarizing customer reviews, in: Proceedings of the 2004 ACM SIGKDD International Conference on Knowledge Discovery and Data Mining - KDD'04, 2004, p. 168, http://dx.doi.org/10.1145/1014052.1014073.
[70] G. Gaikwad, Lexicon Dictionary and Machine Learning, (n.d.).
[71] S. Kiritchenko, X. Zhu, S.M. Mohammad, Sentiment analysis of short informal text, J. Artif. Intell. Res. 50 (2014) 723–762, http://dx.doi.org/10.1613/jair.4272.
[72] F. Ali, K.S. Kwak, Y.G. Kim, Opinion mining based on fuzzy domain ontology and support vector machine: A proposal to automate online review classification, Appl. Soft Comput. J. 47 (2016) 235–250, http://dx.doi.org/10.1016/j.asoc.2016.06.003.
[73] M. Hall, E. Frank, G. Holmes, P. Bernhard, P. Reutemann, I. Witten, The WEKA Data Mining Software: An Update, Vol. 11, 2009, pp. 10–18.
[74] M. Ahmad, S. Aftab, Analyzing the performance of svm for polarity detection with different datasets, Int. J. Modern Educ. Comput. Sci. 9 (2017) 29–36, http://dx.doi.org/10.5815/ijmecs.2017.10.04.
[75] S.D. Jadhav, H.P. Channe, Comparative study of K-NN, Naive Bayes and decision tree classification techniques, Int. J. Sci. Res. (IJSR) 14611 (2013) 2319–7064, http://dx.doi.org/10.1016/j.cplett.2013.06.065.