Received: 16 October 2018 | Revised: 28 July 2019 | Accepted: 21 March 2020

DOI: 10.1111/coin.12327

ORIGINAL ARTICLE

Computational Intelligence WILEY

Aspect category detection using statistical and semantic association

Ashish Kumar[1] | **Mayank Saini**[2] | **Aditi Sharan**[1]

[1]School of Computer and Systems Sciences, Jawaharlal Nehru University, New Delhi, India

[2]AI and Data Practice, Publicis.Sapient, Noida, India

Correspondence
Ashish Kumar, School of Computer and Systems Sciences, Jawaharlal Nehru University, New Delhi, India.
Email: ashishkumar2912@gmail.com

Abstract

Aspect category detection (ACD) is an important subtask of aspect-based sentiment analysis (ABSA). It is a challenging problem due to subjectivity involved in categorization, as well as the existence of overlapping classes. Among various approaches that have been applied to ACD include rule-based approaches along with other machine learning approaches, and most of them are statistical in nature. In this article, we have used an association rule-based approach. To deal with the statistical limitation of association rules, we proposed a hybridized rule-based approach that combines association rules with the semantic association. For semantic associations, we have used the notion of word-embeddings. Experiments were performed on SemEval dataset, a standard benchmark dataset for aspect categorization in the restaurant domain. We observed that semantic associations can complement statistical association and improve the accuracy of classification. The proposed method performs better than several state-of-the-art methods.

KEYWORDS

aspect category detection, association rule, review analysis, semantic association, word-embeddings

1 | INTRODUCTION

Online repositories of opinionated data such as reviews and blogs have become an important platform for consumers to evaluate products, services, and organizations. With the explosive growth of social media such as Twitter, blogs, and reviews hosting sites (Yelp, TripAdvisor), one no longer has to rely only on friends or family for suggestions to make various decisions. Online product reviews are becoming an important source that can assist in making a purchasing decision. In spite of huge availability, it is tough to get useful information from these unstructured reviews. It is not feasible for anyone to go through each of them. So, there is a need for automatically extracting information from these reviews. In most of the cases, one is interested in the sentiment expressed in the review and not in the entire review content. This leads to the well-known problem of sentiment analysis.

Sentiment analysis is generally formulated as a binary classification problem, where the entire review can either be positive or negative. This is, however, a generic formulation. It is obvious that a review can talk about several features (aspects) of an entity/product and can be positive for certain aspects and negative for others. This leads to a more complicated problem of aspect-based sentiment analysis (ABSA).

ABSA is a process of mining and summarizing opinions for each component/aspect of an entity/product discussed in reviews.[1] Although ABSA can be considered as a multiclass classification problem, the framing of the problem itself is a challenge. Subtasks of ABSA involve identifying aspect-terms (mainly explicit) and determining the sentiment associated with these terms. More insights about ABSA can be found in a survey by Schouten et al.[2] An individual aspect-term may not represent a generic aspect and moreover different aspect-terms may correspond to a single aspect. For example, aspect-term *pizza* and *pasta* may correspond to aspect FOOD in restaurant domain. Classifying aspect-terms in a generic aspect is known as aspect categorization. An aspect-term is a fine grain entity, whereas aspect category is a coarse entity. Ideally, we should be able to identify certain aspect categories associated with a domain/product. For example, in the restaurant domain, the identified aspect categories may be FOOD, SERVICE, PRICE, and AMBIENCE. However, a lot of subjectivity is associated with the aspect category, for example, aspect categories may not be well defined and categories may be overlapping. Most of the time aspect categories are implicitly mentioned in the review.

This article is an attempt toward a challenging problem of aspect categorization, which is an important subtask of ABSA. Although some work has been done in this area, the quantum of work is very less compared with the work done in the field of sentiment analysis. In most of the earlier works, people used either statistical or semantic approach to establish the relation between terms used in the review and corresponding category. Both approaches have their pros and cons. In our approach, we used hybridizing these two approaches to exploit more associations. As word-embeddings have become the state-of-the-art for finding context-based semantic associations, we have used association rules along with word-embeddings (context vectors) to capture statistical and semantic associations.

To our knowledge, such an approach has not been tried earlier in the field of aspect categorization. With limited work in this area, our work presents a better take on the challenging problem of aspect categorization. Albeit, our approach is independent of domain, we have tested our approach on standard benchmark data of restaurant domain provided by SemEval2014.* The

*http://alt.qcri.org/semeval2014/task4/index.php?id=data-and-tools

remaining part of this article is organized as follows. Section 2 discusses the background and motivation. In Section 3, we introduce related work on aspect category detection (ACD). In Section 4, we present the details of our proposed method. Experimental settings and results are presented in Section 5. Finally, Section 6 holds the conclusion and future work.

2 | BACKGROUND AND MOTIVATION

Our work is concerned with aspect categorization, which itself is a vague and subjective term. Therefore, in this section, we focus on defining aspect categorization problem more objectively. In order to bring more clarity in the problem statement, Table 1 depicts review sentences and corresponding aspect-terms and aspect categories. Given examples have clearly differentiated the concept of aspect-terms and aspect category.

As presented in Table 1, it is clear that a review sentence is not restricted to only one category. This table also illustrates three different scenarios. First, when aspect category and aspect-term are the same and mentioned explicitly (3rd and 4th). This is the simplest scenario and it is easier to find aspect category for a given review sentence. Second, when the aspect-term is present in the sentence and implying to an aspect category (1st and 6th). Finally, there are cases where neither aspect-term nor aspect category is mentioned explicitly (2nd and 5th). Determining aspect categories, where aspect-term is not explicitly mentioned but is implied in the sentence, is a complicated task. In that case, it requires some other information like contextual information or domain knowledge to extract aspect category.

Much work has been done in the field of sentiment analysis on reviews. However, due to several challenges and subjectivity, very limited work has been done in the field of aspect categorization. Even in SemEval (International Workshop on Semantic Evaluation), the task has been defined for the restaurant domain only. Because of the nature of the problem, the problem has been handled in different ways. As discussed earlier, ACD can be considered as a multiclass classification problem.

Researchers have tried different statistical classification and rule-based methods to extract aspect category and achieved reasonable accuracy. These methods try to establish an association between the review terms and aspect category based on term frequency and other statistical measures. However, most of the methods are incapable to find aspect category in sentences that contain less frequent but category representative terms. In the direction of identifying aspect categories, it is important to find aspect category representative terms. For example, considering 6th review of Table 1, term *Mojito* clearly referring to the FOOD category but may get filtered because of its low frequency in corpora by most of the stats based methods.

TABLE 1 Examples demonstrating aspect-terms and aspect-categories

Review Sentence	Aspect-Terms	Aspect-Categories
1. "*pizza was delicious*."	pizza	FOOD
2. "*delicious but expensive*."	—	FOOD, PRICE
3. "*the food was very cheap*."	food	FOOD, PRICE
4. "*The food was great*."	food	FOOD
5. "*It is very overpriced and not very tasty*."	—	FOOD, PRICE
6. "*Mojito was one of the best item they served*"	mojito	FOOD

In order to deal with the above-mentioned issue, we have used an association rule-based approach combined with the semantic similarity notion using word-embeddings for ACD. In spite of having low frequency but the high semantic similarity with the FOOD category, the term *Mojito* can be used to label the sentence to the FOOD category. Our motivation is to build an aspect category detector by hybridizing statistical and semantic associations between review terms and aspect categories. Semantic approaches, if utilized properly, can overcome some of the limitations of statistical approaches and hence may improve the quality of results.

3 | RELATED WORK

In the related work, we have included the papers specifically focused on ACD. Considering the multi-class nature of the problem, many machine learning classifiers have been used by the researchers for determining aspect category. In the work of Kiritchenko et al,[3] 5-support vector machines (SVM) were built using different features, one for each category (one vs all). They used n-gram, stemmed n-grams, character n-gram, non-contiguous n-gram, word cluster n-gram, and lexicon features to train SVM classifiers. On the other hand, Alghunaim[4] also trained the SVM but represented words as a vector using the Skip-gram model of Word2Vec trained on Google news dataset. Afterward, some features like normalized average vector (NAV), token number (TN), and category similarity (CS) using vector representation of words were obtained. Linear classification models such as SVM are able to use the bag-of-words model to make predictions. But these models do not use the sequential information of words in the sentence even if they are using word-embeddings.

Zhou et al[5] used a semi-supervised word-embeddings algorithm that obtains continuous word representations of reviews with noisy labels. Then a neural network stacked on the word vectors automatically generates deeper and hybrid features. Finally, a logistic regression classifier is trained using hybrid features to predict the aspect category. In the work of Blinov,[6] words were represented by Word2Vec (Skip-gram with 300 dimensions). Each sentence was converted to a vector by averaging its word vectors. Similarly, each category also converted to a vector. For any new sentence, distance is calculated for each category. Category, which produces minimum distance, was finally assigned. Most of these models use Google pre-trained word-embeddings, which are generic in nature. It fails to incorporate sentiments; for example, words vectors of "good" and "bad" have a high similarity score as these words occurred in the same context in news corpus on which they are trained.

Many researchers have tackled the problem of ACD using rule-based approaches. Schouten et al[7] used a normalized frequency co-occurrence matrix of word and category. For each sentence, a score was calculated by summing up the weights of all words occurring in that sentence and normalized it by a total number of words in a sentence. Then each sentence has a score corresponding to each category. Using training data, a separate threshold (that provides best f1-score for training data) was learned for each category. If a sentence score, in testing data, crosses any category's threshold then the sentence was assigned to the corresponding category. Similar to the above methods, Schouten et al[8] further used word category co-occurrence matrix. While calculating sentence score instead of averaging all words weight in the sentence, they took maximum word weight and the remaining process was same as above except for category ANC/MISC, which was assigned to sentences that were not belonging to any of the other categories.

Bornebusch et al[9] used aspect-terms to detect categories by assuming aspect-terms predefined. If aspect-term is a category term, then they assigned the corresponding category; otherwise, for a category like FOOD, if they identify aspect-term is a dish, then the FOOD category is assigned. For all remaining unassigned aspect-terms, a similarity score between the aspect-term and category was calculated using RiTa (WordNet similarity calculation). If the path length is less than 0.4, then the aspect-term is assigned to the corresponding category. If no aspect category is found, then ANC/MISC is assigned.

An ontology-driven approach was applied by Schouten et al[10] for ABSA. They used only 20% of the training data to achieved results comparable to the bag-of-words approach. They trained independent binary SVM classifiers for each aspect-category. Various pre-processing steps (spell correction, tokenization, part-of-speech (POS) tagging, lemmatization, syntactic analysis, word-sense disambiguation, etc.) were performed on sentences to generate features for those classifiers. They also created a domain-specific ontology to extract ontology concepts (sentiment, target, sentiment expression), which helped in determining the correct aspect category label and sentiment label even in sentences having multiple aspects. Using domain-specific ontology always helps to increase the accuracy but at the same time, ontology-based approaches are not scalable and need high manual effort in creating and keep updating the ontology.

A combination of rule-based techniques and a conditional random field (CRF) have been utilized by Patra et al[11] to identify aspect categories. They have used POS, dependency relations, WordNet information, aspect-terms, and sentiment lexicon as features. Garcia-Pablos et al[12] used a topic modeling (LDA) approach combined with continuous word-embeddings and a maximum entropy classifier to output weighted list of aspect-terms, positive-sentiment words, and negative-sentiment words with minimal seeds (each category word and some positive and negative words) as input. They claimed their system as a multi-domain and multilingual unsupervised ABSA system.

Many researchers have used association rule mining in ABSA in the past. Liu et al[13] generated all strong association rules to extract both explicit and implicit features for ABSA. But this method fails in cases when the sentence segment (short phrases or incomplete sentences) has only a single word, for example, "heavy" and "big." Because to generate association rules at least two words are required in the sentence. Hai et al[14] used association rules to identify implicit features and it also discriminates between opinion words and features words by maintaining opinion words only in the rule antecedents and feature words in rule consequents. They used this technique to identify implicit aspects from Chinese reviews.

4 | PROPOSED APPROACH

4.1 | Problem definition

For a given predefined aspect category set $C = \{c_1, c_2, c_3, \ldots, c_k\}$, where C denotes the category label space with k possible categories and a review dataset $R = \{R_1, R_2, R_3, \ldots, R_n\}$ containing n review sentences, the task of ACD can be formulated to learn a function $h : R \rightarrow 2^C$ from a multi category training set $D = (r_i, Y_i)|1 \leq i \leq n, Y_i \subseteq C$ is a set of category labels associated with r_i. For each unseen review $r \in R$, the aspect category prediction function $h(\cdot)$ predicts $h(r) \subseteq C$ as the set of proper category label for r.

4.2 | Design of the proposed approach

Our proposed approach is a rule-based approach, which uses statistical and semantic associations between the review terms (words) and their associated categories in order to generate the rules. Broadly the approach can be divided into the following steps:

1. Finding aspect category representative words by determining the statistical association between review words and aspect category using class-based association rules (CARs).
2. Train word-embeddings on the domain-specific dataset.
3. Finding semantic association between review words and aspect categories using word-embeddings.
4. Aspect rule generation
   a. Generate CARs.
   b. Update rules to incorporate semantic aspect using word-embeddings.
5. Evaluate and analyze the results on test data.

4.2.1 | Class-based association rules

Association rule mining, a popular data mining technique, was originally coined for market basket analysis. It was used to find item-sets that are bought together in a large number of transactions. Association rules are utilized in many other areas such as bio-informatics, web usage mining, and continuous production. Let $I = \{i_1, i_2, \dots, i_d\}$ and $T = \{t_1, t_2, \dots, t_N\}$ represent the set of items and set of transactions, respectively, in a database, where each transaction $t_j; j \in \{1, 2, \dots, N\}$ comprises a subset of items chosen from item-set I. Association rule mainly describes the inferential relation between two disjoint item-sets say X and Y, that is, $X \rightarrow Y$, where, $X \cap Y = \phi$. To determine the strength of a given association rule, two measures are usually calculated named as support and confidence. Support determines the frequency of items appearing in item-sets X and Y in all transactions. While confidence determines how frequently items in Y appear in transactions that contain X.

CAR is a slightly different version of traditional association rules. In CARs, the consequent is fixed that generally corresponds to class labels and the antecedent represents the items that frequently occur in that class; this subset of rules is referred to as the CARs. It makes association rules applicable to classification tasks by building a classifier based on the generated CARs. Assume dataset D, which contains all items of item-set I. Let Y represent the set of class labels and a data case $d \in D$, which contains a subset of items X from I. In the context of CAR, $X \rightarrow y$ represents an association rule where $X \subseteq I$, and $y \in Y$. In this case, the confidence c represents the percentage of cases in D containing X with label y. The support is the percentage of cases in D contains X and are labeled with y.[15] In brevity, this process involves the generation of a complete set of CARs with given minimum support and confidence constraints.

CARs have been used widely for document classification. For this, a document is considered as a collection of words (bag-of-words). Each document represents a transaction and words represent items. This method is ideally applicable in our case, where each aspect category corresponds to a class, the documents (sentences) present in that class are the transactions, and words are the items. The rules will be in the form of *word* $\rightarrow$ *category*. Thus with the help of CARs, we are able

to identify the frequently occurring words in each category. With proper tuning of support and confidence, we are able to filter the irrelevant words and get a list of words that are considered to be representative words for the specified aspect category. These words may be considered as implicit features for an aspect category.

4.2.2 | Word-embeddings

Different ways of generating semantic associations are Linked Statistical Data (LSD), WordNet, word-embeddings, and so on. WordNet is an ontology representation of relationships of words, which is constructed manually. WordNet is a symbolic representation, computing the similarity between words is limited to its hierarchical representation. Whereas word-embeddings represent word meaning from its surrounding context words which are learned from large corpora. Word2Vec (word-embeddings model) represents words in multidimensional vector space, and this enables similarity calculation in terms of vector distance. Hence in terms of similarity calculation, Word2Vec is more effective than WordNet. Word-embeddings are distributed representations of words in a vector space. In this technique, words and phrases are mapped to vectors of real numbers. To obtain word-embeddings, we used distributed representations of words suggested by Mikolov et al.[16,17] In this architecture, neural network language model first learns word vectors and then n-grams neural network language model is trained on top of these distributed representations of words.

Out of two models continuous bag-of-words (CBOW) and Skip-gram proposed by them, we have used the Skip-gram model for Word2Vec representation. This model predicts the context based on the current word by maximizing the classification of a word based on another word in the same sentence. The following mathematically formulation needs to be maximized as an objective of a Skip-gram model for a given sequence of words $w_1, w_2, w_3, \ldots, w_T$.

$$\frac{1}{T}\sum_{t=1}^{T}\sum_{-c\leq j\leq c, j\neq 0}\log p(w_{t+j}|w_t), \tag{1}$$

where w_t is the center word and c is the context window size. A larger value of c provides more samples for training, which leads to high accuracy at the cost of training time. Probability $p(w_{t+j}|w_t)$ is calculated using the softmax function:

$$p(w_o|w_I) = \frac{\exp({v'_{w_o}}^T v_{w_I})}{\sum_{w=1}^{W}\exp({v'_w}^T v_{w_I})}, \tag{2}$$

where v_w and v'_w are the "input" and "output" vector representations of w, and W is the vocabulary size. The cost of computing $\bigtriangledown \log p(w_O|w_I)$ is proportional to W, which is often large ($10^5 - 10^7$ terms). So, one of the following two approximations are used, hierarchical softmax and negative sampling. We have used negative sampling to generate word-embeddings.

In order to use the notion of semantic similarity, we represented each word and all categories by word-embeddings. This similarity information between words and aspect categories is used for improving the efficiency of association rules captured for detecting aspect categories. In the next subsection, we present the hybridization of word-embeddings with association rules.

4.2.3 | Aspect categorization

We first start by generating statistical association rules as explained in Section 4.2.1, we generate rules of the form *word* → *aspect category* for each category. With proper tuning of support and confidence, we expect to identify category representative words. However, a major concern with association rule is that it filters out rules based on confidence and support. As the labeled dataset is small, many features (representative words) having high confidence but low support may get the filter out as they may not have a vast presence in the category. This problem may be due to two different reasons. First, the word does not occur frequently due to the limited availability of tagged data, which is a typical problem with statistical approaches. Second, the word may be rare by nature, and it is not expecting to occur frequently. Example: "*Poke is always fresh and hot-ready to eat*." In this review sentence, *poke* is a dish that is very rare. So, it is not expected to have high support for word *poke* in the review corpus. In spite of being rare such words are category representative words. However, these words are missed by association rule due to less support. These rare but relevant words may not have appropriate support but generally have very high confidence.

One option to include these words can be the lowering of minimum support. But in such cases, many irrelevant words may occur. We can tackle this limitation of association rules by using semantic associations. Our assumption is that by including words with high confidence and high semantic similarity with aspect categories, we can include the relevant words left out by association rules. Now generating semantic associations is not as trivial as generating statistical associations. Our objective is to find a semantic association between words and aspect category. A recent development in the field of semantics is associated with word-embeddings. A brief introduction about word-embeddings was given in Section 4.2.2.

Thus, we used the notion of word-embeddings in order to handle the problem associated with association rules. For this, we started by learning word-embeddings from a large text corpus. Using this we get a word vector of specific dimension for each word of the vocabulary. Each aspect category is represented by a number of words. Once Word2Vec (Skip-gram) model is trained, it is used to calculate the similarity of word and aspect category using the cosine similarity of word vectors. Based on high similarity values, we retain low-frequency words to participate in the rule generation process.

In addition to the calculation of support and confidence, we calculate the similarity between words and categories also. We augment the association rules by introducing new rules that may not have ample support but have ample semantic similarity between the antecedent (words) and the consequent (aspect-category). Since rules are independent of different categories, many rules can be applied to the same review. Thus, a single review can be categorized into many categories.

4.3 | Framework of proposed approach

In this section, we present the framework of our proposed approach, along with its workflow and description of algorithms used. The proposed approach has the following components:

- Data preparation.
- Generating word-embedding model.
- Workflow.

TABLE 2 Examples to demonstrate the effect of generic embeddings and domain-specific embeddings

	Cosine similarity value	
word pair	**Generic Embeddings (GloVe)**	**Domain Embeddings (yelp)**
service, waiter	0.194154	0.329492
ambience, relax	0.292993	0.547767
atmosphere, romantic	0.395026	0.575451
service, vehicle	0.459042	0.238897

We are defining some notations that are used in the algorithms:

1. Review corpus $R = \{R_1, R_2, R_3, \dots, R_n\}$ contains n reviews,
2. Vocabulary $V = \{w_1, w_2, w_3, \dots, w_m\}$ contains m unique words from review corpus,
3. Category Set $C = \{c_1, c_2, c_3, \dots, c_k\}$ containing k different aspect categories,
4. Document Term Matrix $DTM_{n,m}$ and Category Term Matrix $CTM_{k,m}$.

Data preparation

A basic pre-processing has been done on each review R_i in the review corpus R. All stop words are removed as these words introduce noise and do not pose the property of category representation. Furthermore, review sentences are broken down to word-tokens and each word-token is lemmatized using Natural Language Toolkit.[†] After that, pre-processed reviews were divided into training and testing data. We generate a vocabulary set V that contains unique lemmatized-words from training data.

Generating word-embeddings model

We trained a separate domain-specific word-embeddings model (Word2Vec) on yelp restaurants review dataset[‡], which is used to calculate the similarity between words and categories. This model provides word-embeddings for each word in the vocabulary set V. A brief description of word-embeddings is provided in Section 4.2.2.

As shown in Table 2, domain-specific words such as *waiter*, *relax*, *romantic* get higher similarity compared to out-of-domain words like *vehicle* in domain embeddings. It is always easier to filter out irrelevant category terms based on a minimum threshold. On the other hand, in generic embeddings, to retain category-specific terms such as *waiter*, we have to keep a minimum similarity threshold on the lower side which will lead to retaining irrelevant terms as well.

Workflow

After data preparation and learning word-embeddings model (Word2Vec model), we are ready to generate association rules. Figure 1 shows the abstract workflow of our model and Figure 2

[†]https://www.nltk.org/index.html
[‡]https://www.yelp.com/dataset/challenge

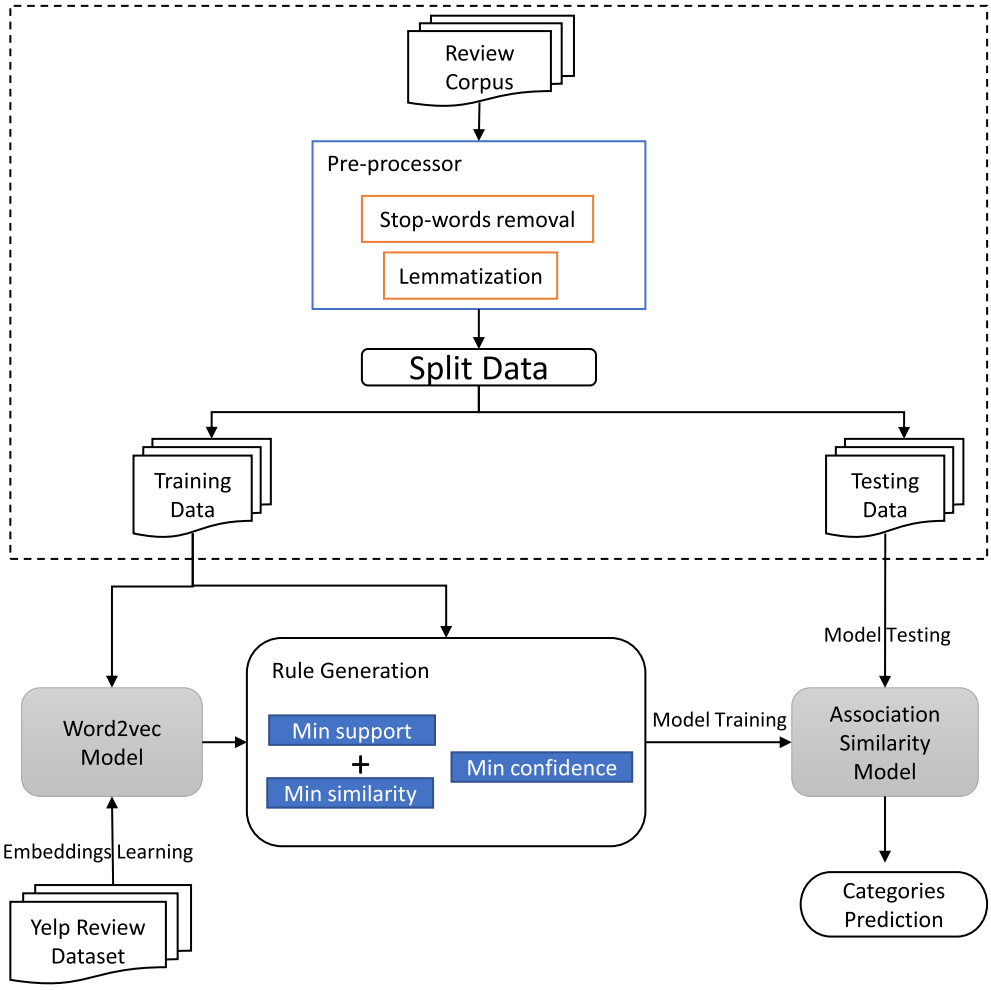

FIGURE 1 Flow chart of the proposed model [Color figure can be viewed at wileyonlinelibrary.com]

illustrates how our method works on a simple dataset. Rule generation process contains the following steps:

Matrices creation

All given aspect categories are identified and stored in the category set C. A document-term-matrix DTM is constructed using review corpus R and vocabulary set V, which contains the frequency of terms in each document (review). So each row represents review R_i and column represents word w_j(term). Cell-value $DTM_{i,j}$ will depict the frequency count of w_j in R_i. Now a category-term-matrix CTM is constructed, which demonstrates term count in each category c_k. For this, reviews belonging to the same category are identified using labeled information and grouped together into k groups (one group for each category). Reviews rows corresponding to the same group are summed up from DTM and stored in CTM. So cell-value $CTM_{k,j}$ will show frequency count of w_j in c_k.

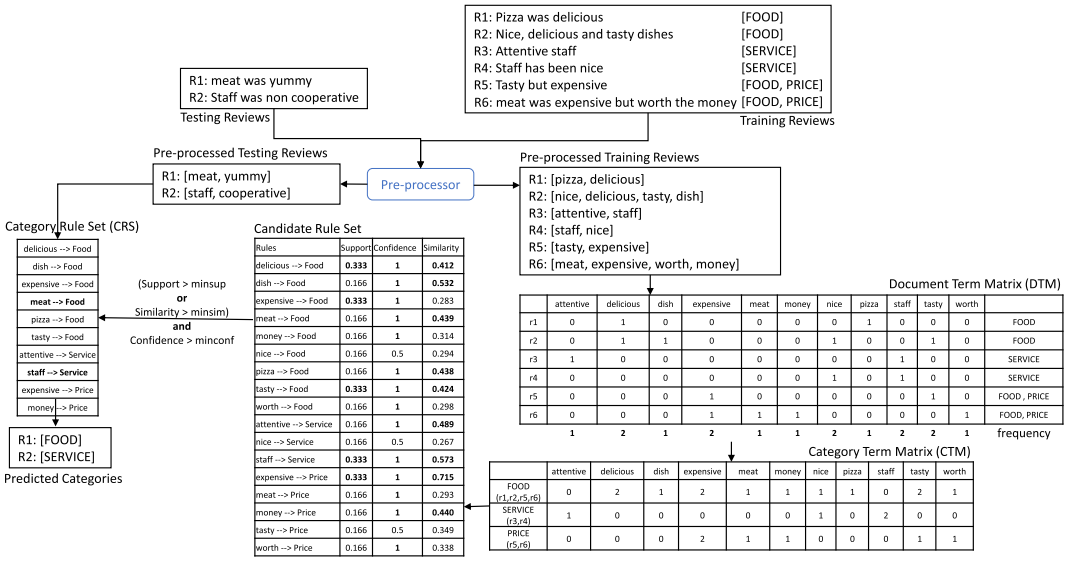

FIGURE 2 Working example of the proposed model [Color figure can be viewed at wileyonlinelibrary.com]

Generating rules using support and confidence

For each word w_i and category c_j, if there exists a non-zero entry in $CTM_{j,i}$ then the following candidate rule can be formed $w_i \rightarrow c_j$. For each association rule $w_i \rightarrow c_j$, support $sup_{i,j}$ and confidence $conf_{i,j}$ scores are calculated using Equations (3) and (4).

$$sup_{i,j} = (w_i \wedge c_j).count/n \tag{3}$$

$$conf_{i,j} = (w_i \wedge c_j).count/w_i.count, \tag{4}$$

where,

1. $(w_i \wedge c_j).count$ is the total number of time word w_i occurs in category c_j and act as a value of cell $CTM_{j,i}$,
2. $w_i.count$ is the total number of time word w_i occurs in review corpus R and act as ith column sum of DTM,
3. n is a total number of reviews in R.

Here, support represents the weight of the term in overall review corpus and confidence represents the significance of a term to a category. Greater support value indicates a higher frequency for a term in the overall corpus. The greater value of confidence will show the higher significance of that term in the corresponding category. Both support and confidence values can be easily calculated with the help of document-term-matrix and category-term-matrix. We have counted the occurrence of a term in the review corpus only once.

Incorporating semantic similarity in association rules

To calculate the similarity between word and category, we first trained domain-specific word-embeddings using the Word2Vec model on the yelp restaurants dataset. Once we have word-embeddings for our vocabulary, then we need a set of words to represent each category. A simple way to represent category is to use the single word which in our case is the category name itself. For example, aspect category SERVICE can be represented using the word-embeddings of

service. But this representation has some problem, for example, *service* has high similarity with *staff* but low with other terms like *waiter*, *manager*, and so on. A single word cannot cover all facets of a category. So rather than using a single word we have used a set of words to represent a category. If *staff* is also chosen to represent SERVICE category along with word *service*, then other words (like *waiter*, *chef*) that are indirectly similar with *service* can also be retained based on semantic similarity.

For this purpose, we collected the top K most similar words from the vocab set that has the highest cosine similarity with the aspect category name, where the category name itself is represented using word-embedding. Now, these top K words acted as aspect category. To measure the similarity between a word w_i and category c_j, cosine similarity is calculated between all K words of the category c_j and word w_i. The maximum similarity value among them is retained as a similarity value of w_i and c_j.

$$sim_{i,j} = \max_{k \in \{1,K\}} (cosine(w_i, c_{j_k})). \tag{5}$$

Finally, rules satisfying (*minsup* or *minsim*) and *minconf* for category c_j are stored in category-rule-set *CRS*(j). The steps for generating category rules are presented in Algorithm 1. Once we have a category-rule-set, it can be used to predict the category of a new review sentence from testing data (using Algorithm 2).

Algorithm 1. Generate Category Rule Set

procedure GETCATEGORYRULESET($R_{Training}, C, V, minsup, minsim, minconf, n$)
  for review $r_i \in R_{Training}$ **do**
    for word $w_j \in V$ **do**
      $DTM_{i,j} = freq(w_j \; in \; r_i)$
    end for
  end for
  for category $c_j \in C$ **do**
    $CTM_j = columnwise\_sum(DTM_k) \; \forall \; r_k \in c_j$
  end for
  for word $w_i \in V$ **do**
    for category $c_j \in C$ **do**
      $sup(w_i, c_j) = (wi \wedge c_j).count/n$
      $conf(w_i, c_j) = (wi \wedge c_j).count/w_i.count$
      $sim(w_i, c_j) = 0$
      for word $w_k \in c_j$ **do**
        $sim(w_i, w_k) = word2vec\_similarity(w_i, w_k)$
        $sim(w_i, c_j) = max(sim(w_i, c_j), sim(w_i, w_k))$
      end for
      if $(sup(w_i, c_j) < minsup(c_j)$ **OR** $sim(w_i, c_j) < minsim(c_j))$ **AND** $(conf(w_i, c_j) < minconf(c_j))$ **then**
        $add(w_i \rightarrow c_j) \; to \; CRS(j)$
      end if
    end for
  end for
  return *CRS* ▷ Category Rule Set *CRS*
end procedure

Algorithm 2. Estimate Category for Test Data

```
procedure PREDICTCATEGORY(R, C, minsup, minsim, minconf)        ▷ Review Data R
    R ← Pre_processing(R)
    R_Training, R_Testing ← SplitData(R)
    V ← GenerateVocabulary(R)
    CRS ← GETCATEGORYRULESET(R_Training, C, V, minsup, minsim, minconf, n)
    for review r ∈ R_Testing do
        for word w_i ∈ r do
            for category c ∈ C do
                if w_i ∈ CRS(c) then
                    assign category c to review r
                end if
            end for
        end for
    end for
end procedure
```

5 | EXPERIMENTAL SETUP

5.1 | About the dataset

Research problem in this article is a subtask of the SemEval-2014 challenge.[§] We used the same restaurant domain SemEval2014 task4 dataset. It contains 3041 training instances and 800 testing instances. The task is to classify customer review sentences into different aspect categories based on their overall meaning. These categories are often predefined.

In the dataset, there are five categories: AMBIENCE, FOOD, PRICE, SERVICE, and ANECDOTES/MISCELLANEOUS. Each review has at least one category assigned. There may be some reviews assigned to more than one category, which makes it a multi-label classification task. Figure 3A shows the distribution of the number of categories per review sentence in training and testing data. Furthermore, there is no uniform distribution of categories in the review dataset, and Figure 3B clearly shows that two categories FOOD and ANECDOTES/MISCELLANEOUS dominate in both training and testing data while remaining three categories have almost equal distribution. Out of these categories, the ANECDOTES/MISCELLANEOUS category is assigned to those sentences that do not belong to any other four categories. Our model is trained to learn only four categories. Sentences that do not have any of these four categories will be assigned to the ANECDOTES/MISCELLANEOUS category in the post-processing step.

5.2 | Evaluation metric

For the performance evaluation purpose, three metrics are used: *precision*, *recall*, and *f1-score*. The *precision*, *recall*, and *f1-score* are computed as follows.

$$Precision = \frac{TP}{TP + FP} \tag{6}$$

[§]http://alt.qcri.org/semeval2014/task4/

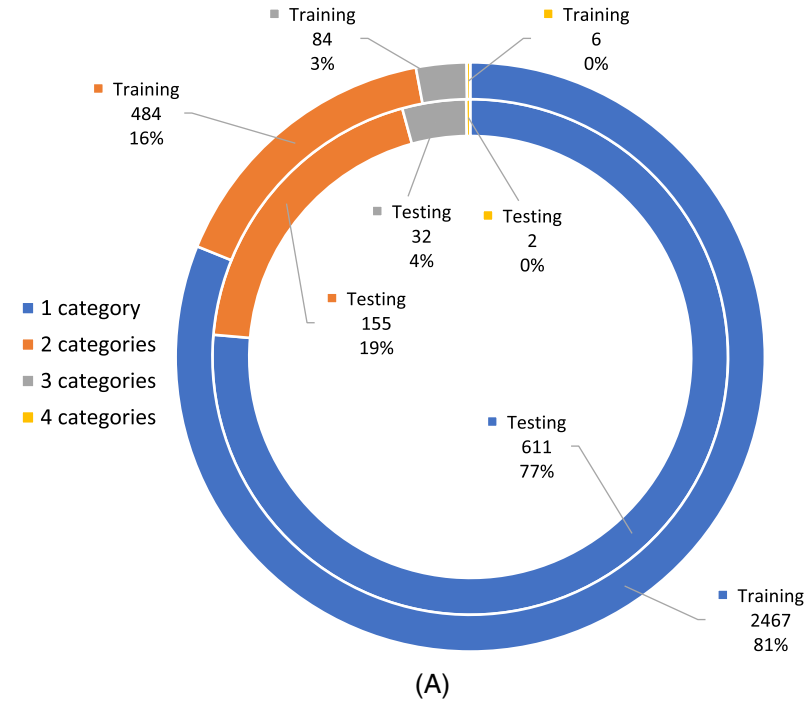

FIGURE 3 A, Distribution of the number of categories per review sentence. B, Number of reviews belonging to each category in training and testing data [Color figure can be viewed at wileyonlinelibrary.com]

$$Recall = \frac{TP}{TP + FN} \tag{7}$$

$$F1 - score = \frac{2 \times Precision \times Recall}{Precision + Recall}, \tag{8}$$

where *TP*, *TN*, *FP*, and *FN* are the total number of *True Positive*, *True Negative*, *False Positive*, and *False Negative*, respectively.

	True Label	False Label
True Prediction	TP	FP
False Prediction	FN	TN

5.3 | Experimental results and analysis

The experiments were performed on the SemEval dataset mentioned in Section 5.1. The training part generates the rules in the form of *word* → *aspect category* as mentioned in Algorithm 1, and thereafter in the testing phase, these rules were applied for determining aspect category using Algorithm 2. Algorithm 1 requires inputs in form of review sentences (from training dataset) along with three hyper-parameters *minsup*, *minsim*, and *minconf* and generates category-rule-set (CRS) for each aspect category as an output. The output can also be interpreted as a set of relevant words for each aspect category.

To configure hyper-parameters, we have used grid-search. Since grid-search is computationally expensive, we restraint on search space. Search space for *minsim* and *minconf* is kept between 0.50 and 1.00 in step of 0.01 and for *minsup*, it is between 0.001 and 0.010 in step of 0.0001. To tune optimal values for these hyper-parameters, we kept 20% of the training data for validation. Grid-search is applied to learn and tune hyper-parameters on training and validation data, respectively. Grid-search works by trying every combination of values in search spaces. Hyper-parameters combination that gives best *f1-score* on validation data is finally chosen as models' hyper-parameters. In this way, these hyper-parameters were set empirically 0.002, 0.60, 0.74 for *minsup*, *minsim*, and *minconf*, respectively. Figure 4 demonstrates the impact of *minconf* on *f1-score* for different values in a range of 0.50 to 1.

Words having a frequency greater than the threshold specified by minimum support are chosen to participate in rule formation. As shown is Equation (3), support involves both word and associated aspect category, a word may have different support values in different categories.

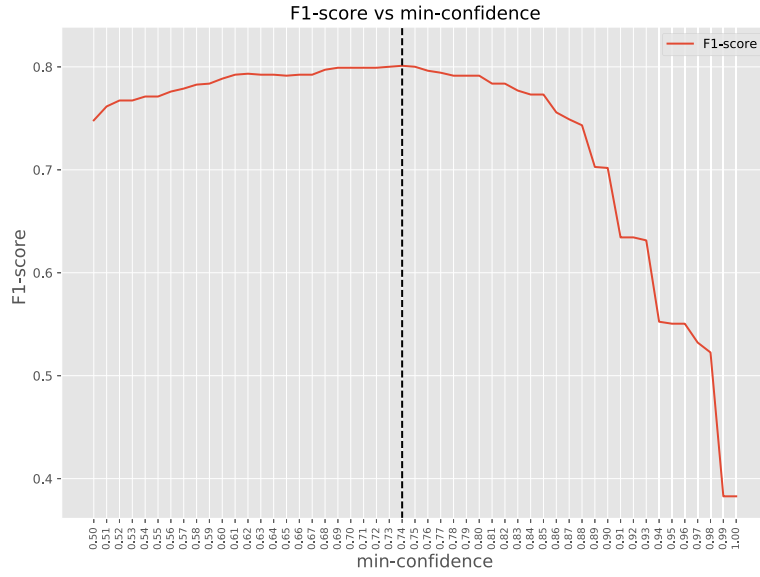

FIGURE 4 F1-score vs minimum confidence [Color figure can be viewed at wileyonlinelibrary.com]

TABLE 3 Results using support and confidence (sup AND conf)

	AMBIENCE	FOOD	PRICE	SERVICE	ALL
Precision	85.19%	89.73%	98.53%	93.13%	90.81%
Recall	58.47%	87.80%	80.72%	86.63%	82.43%
F1-score	69.35%	88.75%	88.74%	89.76%	86.41%

TABLE 4 Results using support or similarity, and confidence [(sup OR sim) AND conf]

	AMBIENCE	FOOD	PRICE	SERVICE	ALL
Precision	83.70%	89.79%	98.59%	92.59%	90.48%
Recall	65.25%	90.43%	84.34%	87.21%	85.34%
F1-score	73.33%	90.11%	90.91%	89.82%	87.83%

For our corpus size (total number of reviews) of 3041 and *minsup* of 0.002, the minimum frequency will become 3041*0.002 = 6.082 in each category. So only those words having frequency 7 or greater will participate in final rule formation and remaining rules will be discarded. This may lead to the removal of good category representative words that are having low frequencies. Schouten et al[7] suggest that a method dealing with very low-frequency words by incorporating semantic similarity could improve results significantly.

In our experiments, we used association rule mining to generate all possible rules. Then one way to filter out the relevant rules is to use support and confidence, and the results have been shown in Table 3. In our proposed method apart from support and confidence, we also utilized semantic similarity, if any rule having *minconf* and either *minsup* or *minsim*, then that rule is retained and rest are discarded. Results have been presented in Table 4. Experimental results (Tables 3 and 4) show that while incorporating similarity with support and confidence there is an increase in the performance of the model. If we compare these two tables, we can see the improvement in the recall in the range of 2%-6% for each category while maintaining the precision at the same time.

Our proposed approach is independent of domain, but on incorporating semantic similarity using domain-specific word-embeddings (trained on specific domain dataset such as yelp), the approach has shown better accuracy over generic word-embeddings (trained on the generic domain such as Wikipedia,[¶] Google news[#]). A variation on the performance of the model using GloVe-embeddings[18] (trained on Wikipedia) of different dimensions (50, 100, 200, 300)[‖] and yelp-embeddings (trained on restaurant domain) is shown in Table 5.

As GloVe embeddings over 100 dimensions did not reflect any significant increase in performance, it can be implied that all information under 100 dimensions is sufficient, computationally efficient, and well exploited by our proposed model. It clearly depicts that the best results are obtained using the domain (restaurant) word-embeddings.

In CAR mining, there is a concept of multiple minimum support. Where instead of having the same minimum support threshold value for all category classes, each category class can have its own minimum support values.[19] To illustrate the effect of multiple minimum support, we also

[¶]https://dumps.wikimedia.org/enwiki/
[#]https://drive.google.com/file/d/0B7XkCwpI5KDYNlNUTTlSS21pQmM/edit?usp=sharing
[‖]https://nlp.stanford.edu/projects/glove/

TABLE 5 F1-score of different word-embeddings

Embedding	AMBIENCE	FOOD	PRICE	SERVICE	ALL
glove.50d	68.54%	88.44%	86.79%	85.63%	84.95%
glove.100d	70.65%	88.17%	87.90%	88.05%	85.83%
glove.200d	70.00%	88.57%	89.03%	90.15%	86.52%
glove.300d	70.00%	88.67%	89.61%	90.09%	86.62%
yelp.100d	73.33%	90.11%	90.91%	89.82%	87.83%

TABLE 6 Results of model for multiple minimum support values in different categories

	AMBIENCE	FOOD	PRICE	SERVICE	ALL
Precision	83.70%	90.75%	97.22%	95.39%	91.47%
Recall	65.25%	89.23%	84.34%	84.30%	84.07%
F1-score	73.33%	89.99%	90.32%	89.51%	87.62%

TABLE 7 Results of model for multiple minimum support values and multiple minimum confidence values in different categories

	AMBIENCE	FOOD	PRICE	SERVICE	ALL
Precision	73.11%	90.75%	97.53%	90.06%	88.62%
Recall	73.73%	89.23%	95.18%	89.53%	87.61%
F1-score	73.42%	89.99%	96.34%	89.80%	88.11%

conducted experiments and the results are presented in Table 6. Minimum support threshold values 0.002, 0.004, 0.001, and 0.009 corresponding to AMBIENCE, FOOD, PRICE, and SERVICE categories are set empirically. No significant difference has been observed with a single threshold (Table 4) and multiple threshold (Table 6) results on a given dataset.

Instead of setting multiple minimum support alone, we have learned all three hyper-parameters, that is, minimum support, minimum confidence, and minimum similarity separately for each category. Minimum confidence values are set to [0.57, 0.74, 0.61, 0.63] and minimum support values are set to [0.002, 0.004, 0.001, 0.009] for AMBIENCE, FOOD, PRICE, and SERVICE categories, respectively. While a minimum similarity value is set to 0.60 for each category. Results with these settings are presented in Table 7. Tuning all three hyper-parameters has shown more promising results (Table 7) over methodologies that use a single threshold (Table 4) or just multiple minimum supports (Table 6).

Even though we get slightly better results, multiple-threshold model overfits on the given dataset. Some of the classes have as low as 321 examples to learn the threshold. The threshold calculated for small classes is overfitted rather than being generalized due to the small size of the dataset. In this article, we have focused more on having a single threshold across classes. Final results for all categories including the ANECDOTES/MISCELLANEOUS category using restaurant domain word-embeddings (yelp) are depicted in Table 8. It can be analyzed from Table 8 that categories that can be represented by a limited set of category represented words (eg, PRICE, SERVICE) will have high precision than the vague categories (eg, AMBIENCE).

TABLE 8 Evaluation metrics of our method (yelp.skipgram.100d) on test data

Category	TP	FP	FN	TN	Precision	Recall	F1-score
AMBIENCE	77	15	41	667	83.70%	65.25%	73.33%
FOOD	378	43	40	339	89.79%	90.43%	90.11%
PRICE	70	1	13	716	98.59%	84.34%	90.91%
SERVICE	150	12	22	616	92.59%	87.21%	89.82%
ANEC./MISC.	168	58	66	508	74.34%	71.79%	73.04%
all	843	129	182	2846	86.73%	82.24%	84.43%

TABLE 9 Comparison of results with different models

Model	Precision	Recall	F1-score
NRC-Canada	91.04%	86.24%	88.58%
UNITOR_U	84.98%	85.56%	85.27%
Our Method	**86.73%**	**82.24%**	**84.43%**
Schouten et al[8]	84.40%	83.10%	83.80%
XRCE	83.23%	81.37%	82.29%
UWB_U	84.36%	78.93%	81.55%
UWB	85.09%	77.37%	81.04%
UNITOR	83.68%	78.05%	80.77%
SAP_RI	82.57%	75.80%	79.04%
SNAP	79.20%	77.27%	78.22%
Blinov_U	76.64%	73.95%	75.27%
UBham_U	82.82%	68.20%	74.80%
UBham	81.88%	67.90%	74.24%
EBDG_U	71.56%	76.59%	73.99%
SeemGo	84.13%	65.66%	73.75%
SINAI	66.59%	82.44%	73.67%
JU_CSE	68.03%	73.07%	70.46%
lsis_lif	77.88%	60.78%	68.27%
ECNU	65.26%	69.46%	67.30%
UFAL	57.21%	73.95%	64.51%
Baseline	—	—	63.89%

We compare our method with all the methods presented in SemEval-2014 competition[20] along with some latest state-of-the-art methods used for ACD (in Table 9). Only NRC-Canada and UNITOR_U are outperforming our method. But as shown in Table 9, our method has achieved higher precision score compared with UNITOR_U.

However, we can observe that these methods are lexicon-based methods that are using multiple lexicon features, named entity recognition, etc.; therefore, these methods are computationally

TABLE 10 Precision of 5-fold cross validation

	AMBIENCE	FOOD	PRICE	SERVICE	ALL
Fold_0	89.36%	82.54%	93.10%	92.91%	86.70%
Fold_1	90.99%	82.59%	91.13%	91.62%	86.39%
Fold_2	91.05%	81.75%	91.56%	92.70%	86.16%
Fold_3	90.27%	82.51%	91.12%	93.09%	86.57%
Fold_4	90.34%	81.05%	91.55%	92.98%	85.71%
Fold_Avg	90.40%	82.09%	91.69%	92.66%	86.31%
Test	83.70%	89.79%	98.59%	92.59%	90.48%

TABLE 11 Recall of 5-fold cross validation

	AMBIENCE	FOOD	PRICE	SERVICE	ALL
Fold_0	72.01%	92.35%	80.60%	79.17%	84.57%
Fold_1	68.86%	94.69%	82.11%	82.32%	86.17%
Fold_2	69.41%	92.80%	87.85%	79.33%	85.46%
Fold_3	65.85%	93.14%	86.02%	80.30%	84.83%
Fold_4	66.36%	93.93%	83.38%	80.59%	85.10%
Fold_Avg	68.50%	93.38%	83.99%	80.34%	85.23%
Test	65.25%	90.43%	84.34%	87.21%	85.34%

TABLE 12 F1-score of 5-fold cross validation

	AMBIENCE	FOOD	PRICE	SERVICE	ALL
Fold_0	79.75%	87.17%	86.40%	85.49%	85.62%
Fold_1	78.40%	88.23%	86.39%	86.72%	86.28%
Fold_2	78.77%	86.93%	89.67%	85.49%	85.81%
Fold_3	76.15%	87.50%	88.50%	86.22%	85.69%
Fold_4	76.52%	87.02%	87.28%	86.34%	85.41%
Fold_Avg	77.92%	87.37%	87.65%	86.06%	85.76%
Test	73.33%	90.11%	90.91%	89.82%	87.83%

expensive. On the other hand, our method is based on statistical and semantic associations only and it is computationally very efficient.

We have also checked the stability of our method by using cross-validation. For this, we divided our training dataset into 5-folds to perform cross-validation. Cross-validation have shown consistency in results (presented in Tables 10,11, and 12). Table 13 is a snapshot of some extracted category representative words using our method. Highlighted words are those words that do not have support greater than minimum support but having similarity greater than minimum similarity. From Table 13, it can be observed that many categories represented words not captured by standard association rules have been included when word-embeddings are used.

TABLE 13 Extracted category representative words

AMBIENCE	FOOD	PRICE	SERVICE
ambiance	**alcohol**	affordable	accomodating
ambience	american	**bargain**	**amiable**
ambient	**americanized**	beat	**apologetic**
architecture	**apppetizers**	buck	**apologize**
artsy	**aunthentic**	charge	**attendant**
atmoshere	authentic	cheap	attentive
atmoshpere	**authentically**	cost	**berate**
atmosphere	**bacon**	dollar	**busboy**
atomosphere	**barbecued**	**economical**	**clueless**
attractive	basil	**exorbitant**	**competent**
backyard	**basmati**	expensive	**cordial**
beautiful	**bbqed**	inexpensive	**courteous**
bistrotype	**bean**	**moderate**	**crew**
breezy	**beancurd**	**outragous**	**curtious**
bustling	beat	**overcharged**	delivery
calm	beef	**overprice**	**efficient**
candlelight	beer	overpriced	**employee**
chill	**biscuit**	**penny**	**engaging**
clean	bite	price	five
comfortable	bland	pricey	**freindly**
conversation	**braised**	**pricy**	friendly
cool	bread	**ratio**	**gabriela**
cozy	**brulee**	reasonable	**gentleman**
cramped	bun	reasonably	**genuine**
cute	burger	**spending**	**gracious**
decor	**buttah**	**steep**	greet
decoration	**butternut**	**wallet**	helpful

6 | CONCLUSION

In this article, we presented a rule-based approach for ACD, which is a subtask of aspect-based sentiment analysis. Separate rule sets were generated for each aspect category. The rules for determining categories were generated using the notion of statistical and semantic association between the words and aspect category. The rules were generated using a class-based association approach hybridized with word-embeddings. Our approach is domain-independent and can be applied across domains without changing the algorithm. At the same time, incorporating domain knowledge can further enhance accuracy.

In the future, instead of using unigrams to predict aspect category, we will try to incorporate n-grams as category identifier. We can also learn word-embeddings for compound

words to disambiguate the context. For example, "*I don't like the hot dog they served*." In the given review sentence, *hot* and *dog* separately cannot indicate FOOD category but *hot dog* combined can. At the same time, word-embeddings for *hot dog* will have high similarity with the FOOD category. This approach is ideal for small to medium size datasets. On the SemEval-2014 dataset, we start getting around 80% F1-score on as low as 1000 training instances. Due to the hierarchical category structure, we cannot apply our model on the SemEval-2016 dataset with our present model setting. But we are working to accommodate all these ideas and enhancement in our future research.

ORCID

Ashish Kumar https://orcid.org/0000-0002-2156-2104

REFERENCES

1. Wang Bo, Liu Min. Deep learning for aspect-based sentiment analysis. Stanford University Report. 2015.
2. Schouten K, Frasincar F. Survey on aspect-level sentiment analysis. *IEEE Trans Knowl Data Eng*. 2016;28(3):813-830.
3. Kiritchenko S, Zhu X, Cherry C, Mohammad S. NRC-Canada-2014: detecting aspects and sentiment in customer reviews. Paper presented at: Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014); 2014:437-442; The Association for Computer Linguistics.
4. Alghunaim A, Mohtarami M, Cyphers S, Glass J. A vector space approach for aspect based sentiment analysis. Paper presented at: Proceedings of the VS@HLT-NAACL: The Association for Computational Linguistics; 2015:116-122.
5. Zhou X, Wan X, Xiao J. Representation learning for aspect category detection in online reviews. Paper presented at: Proceedings of the 29th AAAI Conference on Artificial Intelligence; 2015:417-424; AAAI Press.
6. Blinov P, Kotelnikov EV.. Blinov: Distributed representations of words for aspect-based sentiment analysis at SemEval 2014. Paper presented at: Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014); 2014:140-144; The Association for Computer Linguistics.
7. Schouten K, Frasincar F, Jong F. COMMIT-P1WP3: a co-occurrence based approach to aspect-level sentiment analysis. Paper presented at: Proceedings of the 8th international workshop on semantic evaluation (SemEval 2014); 2014:203-207; The Association for Computer Linguistics.
8. Schouten K, Weijde O, Frasincar F, Dekker R. Supervised and unsupervised aspect category detection for sentiment analysis with co-occurrence data. *IEEE Trans Cybern*. 2018;48(4):1263-1275.
9. Bornebusch F, Cancino G, Diepenbeck M, et al. iTac: aspect based sentiment analysis using sentiment trees and dictionaries. Paper presented at: Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval2014); 2014:351-355; The Association for Computer Linguistics.
10. Schouten K, Frasincar F, Jong F. Ontology-enhanced aspect-based sentiment analysis. Paper presented at: Proceedings of the International Conference on Web Engineering; vol. 10360; 2017:302-320; Springer.
11. Patra BG, Mandal S, Das D, Bandyopadhyay S. JU_CSE: A conditional random field (crf) based approach to aspect based sentiment analysis. Paper presented at: Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014); 2014:370-374; The Association for Computer Linguistics.
12. García-Pablos A, Cuadros M, Rigau G. W2VLDA: almost unsupervised system for aspect based sentiment analysis. *Expert Syst Appl*. 2018;91:127-137.
13. Liu B, Hu M, Cheng J. Opinion observer: analyzing and comparing opinions on the web. Paper presented at: Proceedings of the 14th International Conference on World Wide Web; 2005:342-351; ACM.
14. Hai Z, Chang K, Kim J-J. Implicit feature identification via co-occurrence association rule mining. Paper presented at: Proceedings of the International Conference on Intelligent Text Processing and Computational Linguistics; vol. 6608, 2011:393-404; Springer.
15. Liu B, Hsu W, Ma Y. Integrating classification and association rule mining. Paper presented at: Proceedings of the 4th International Conference on Knowledge Discovery and Data mining; 1998:80-86; AAAI Press.

16. Mikolov T, Sutskever I, Chen K, Corrado GS, Dean J. Distributed representations of words and phrases and their compositionality. Paper presented at: Proceedings of the Advances in Neural Information Processing Systems; 2013:3111-3119.
17. Mikolov T, Chen K, Corrado G, Dean J. Efficient Estimation of Word Representations in Vector Space; 2013.
18. Pennington J, Socher R, Manning CD. Glove: global vectors for word representation. Paper presented at: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP); 2014:1532-1543; ACL.
19. Liu B, Hsu W, Ma Y. Mining association rules with multiple minimum supports. Paper presented at: Proceedings of the 5th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining; 1999:337-341; ACM.
20. Pontiki M, Galanis D, Pavlopoulos J, Papageorgiou H, Androutsopoulos I, Manandhar S. SemEval-2014 task 4: aspect based sentiment analysis. Paper presented at: Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014); 2014:27-35; The Association for Computer Linguistics.

How to cite this article: Kumar A, Saini M, Sharan A. Aspect category detection using statistical and semantic association. *Computational Intelligence*. 2020;36:1161–1182. https://doi.org/10.1111/coin.12327