IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received May 9, 2022, accepted June 5, 2022, date of publication June 15, 2022, date of current version June 21, 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3183205

A Survey on Implicit Aspect Detection for Sentiment Analysis: Terminology, Issues, and Scope

PIYUSH KUMAR SONI, (Member, IEEE), AND RADHAKRISHNA RAMBOLA
Mukesh Patel School of Technology Management and Engineering, SVKM's NMIMS, Shirpur 425405, India

Corresponding author: Piyush Kumar Soni (piyushkumar.soni@nmims.edu)

ABSTRACT Sentiment analysis or opinion mining has come forth as an attractive research field in the past few years. Sentiment analysis extracts sentiments from the text for analysis and aggregation at different levels of detail. In aspect-level sentiment analysis, we aggregate sentiment for different aspects of entities. The bulk of the research work executed so far focuses on detecting explicit aspects but ignored implicit aspects, which are insinuated by other existing words and articulates of the sentence. Since a significant percentage of sentences contain implicit aspects, detection of implicit aspects becomes vital for sentiment analysis. This survey concentrates on implicit aspect detection, and a detailed discussion about state of the art is provided. The available methods are categorized depending on the algorithm applied. Quantitative evaluation for different methods as stated by authors is included for comparison purpose. Discussion about terminology, issues, and scope in the detection of implicit aspects is also included. The fine-grained sentiment information collected may be used in many applications in various domains. This survey aims to advocate the need for implicit aspect detection, determine existing efficient solutions, identify complications in implicit aspect detection, and suggest measures to improve performance, which comprise future research trends in implicit aspect detection.

INDEX TERMS Aspect, explicit aspect, implicit aspect, opinion mining, sentiment analysis.

I. INTRODUCTION

In the past few years, people are increasingly sharing their views about different entities like products, persons, or organizations through blogs, discussion forums, social networking platforms, and e-commerce websites. This sharing of views has become possible by the swift growth in web applications and the wide and low-cost availability of the Internet, resulting in enormous data on the Internet. This enormous data contains valuable information which may be utilized for critical decision making, and sentiment analysis is one very significant application.

Even before World Wide Web came into wide use, most of us were used to ask our friends about their experiences/recommendations for decision making. The Web and Internet have now made it possible to read opinions or recommendations of ordinary people from diversified locations/cultures and whom we do not even know. Buying behavior of many customers is influenced by the opinions of other customers that they find on the Web [1]. Sentiment analysis systems provide automatic summary generation of user reviews that may help a customer in decision making.

The main advantages of the sentiment analysis system are scalability as it can summarize large quantities of text, real-time analysis as it can generate results at run time, and consistent criteria as it is automated and free from bias compared to humans.

A sentiment analysis system is also an essential tool for private and government organizations. Sentiment analysis can be used to improve traditional recommendation systems. It is helpful for manufacturers as it gives the sentiment orientation of customers about their products. It can also be used for market research and competitive analysis. Other domains of application include politics, government policy-making, investigation of legal matters [2].

Sentiment analysis may be performed at different levels of detail, aspect level sentiment analysis being the most informative one. Detection of explicit aspects is explored widely by researchers, and a variety of approaches are suggested. On the other hand, due to its complexity, less attention is given to

The associate editor coordinating the review of this manuscript and approving it for publication was M. Shamim Kaiser.

detecting implicit aspects. A significant proportion of the text contains implicit aspects making detection of implicit aspects important for sentiment analysis.

Many survey papers discuss and analyze different sentiment analysis approaches, but a handful of them have discussed aspect-level sentiment analysis. The following table lists these papers and their specifics.

TABLE 1. Survey papers on aspect detection.

Author	Aspect Extraction	Shortcomings
Schouten et al. [3]	All	Absence of separate discussion about implicit aspects
Sabeeh et al. [4]	All	Absence of separate discussion about implicit aspects
Hai et al. [5]	All	Includes only deep learning-based methods without separate discussion about implicit aspects
Rana et al. [6]	All	Includes separate discussion about implicit aspects but only for few articles
Maitama et al. [7]	All	Includes separate discussion about implicit aspects but very short description of methods
Ganganwar et. al. [8]	Implicit Only	Includes discussion about only few articles
Tubishat et al. [9]	Implicit Only	Very short description of methods and too many sub-categories

Schouten *et al.* [3] have discussed methods for aspect-level sentiment analysis in detail, but methods for detecting implicit aspects were not discussed separately. Sabeeh and Dewang [4] have briefly discussed methods for aspect detection in their article, but implicit aspect detection was not discussed separately. Deep learning-based methods for detecting aspects were analyzed by Hai and co-authors [5] without particular discussion of implicit aspect detection.

Rana and Cheah [6] have discussed explicit and implicit aspect detection methods separately, but discussion about implicit aspect detection included a small number of approaches. Similar to [6], Maitama *et al.* [7] have discussed methods for explicit and implicit aspect detection separately, but the description of the methods is too short of understanding.

Although Ganganwar and Rajalakshmi [8] have analyzed methods for only implicit aspect detection, only a handful of methods were discussed. Tubishat *et al.* [9], according to us, is the most formidable survey on implicit aspect detection amongst the article we have studied. However, the description of the methods is not sufficiently detailed; also, they have defined too many sub-categories for supervised and unsupervised methods.

The importance of implicit aspect detection for sentiment analysis and the discussed limitations of the existing survey papers have motivated us to write a paper focusing on only implicit aspect detection. Our paper is different from existing surveys in following aspects:

- Focused on implicit aspect detection
- Breadth of coverage
- Details of individual approaches
- Categorization of these approaches

The remaining sections of the paper are organized as follows: Terminology used in the surveyed field is discussed in section 2. Section 3 includes a detailed discussion about various approaches for the detection of implicit aspects. Performance measures used and a comparison of the performance of surveyed methods is given in section 4. After discussing issues and future research prospects in implicit aspect detection in section 5, we have concluded the survey paper inside section 6.

II. TERMINOLOGY

The surveyed field of research is generally termed sentiment analysis and is also referred to as opinion mining or subjectivity analysis. It is a field within natural language processing. In this research domain, we study the concept of sentiment, opinion, attitude, and emotion [10].

Sentiment analysis is computationally recognizing, categorizing, and aggregating sentiments conveyed in a part of the text. Technically, identifying sentiment may be construed as identifying the quadruple(st, i, h, pt), where st denotes the sentiment, i denotes the item about whom the sentiment is conveyed, h denotes the holder (person conveying the sentiment), and pt denotes the point of time when the sentiment was conveyed. Nevertheless, most attempts focus on identifying (st, i) only. Sentiments are generally classified as +ve, neutral, or -ve.

As shown in figure 1, sentiments can be aggregated at different levels of details, including a document, a sentence, an entity, and different aspects of an entity. The accuracy and usefulness of the generated sentiment information increase with the aggregation of sentiments at finer levels of detail.

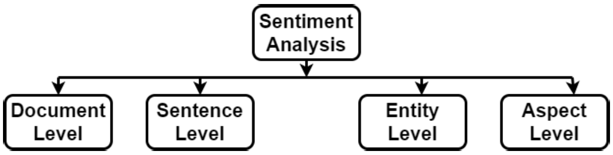

FIGURE 1. Levels of sentiment analysis.

A document may be a review, a post, or an article; an entity may be anything like a product, an organization, an individual, a topic, an event [11]; and an aspect may be a part/component or property/attribute of an entity.

At document-level sentiment analysis, first opinion words from the document are extracted. Then based on the polarity of (majority of) opinion words, a sentiment label is assigned to the whole document. Subjective sentences (sentences with sentiment) are treated as small documents in sentence-level sentiment analysis, and a sentiment label is assigned to each sentence based on opinion words from the sentence. At entity-level sentiment analysis, first entities in the document are identified. Then based on the opinion words in the context of the respective entities, a sentiment label is assigned for each entity.

Terminology about aspect level sentiment analysis may be understood from the given example review:

"The camera quality of Samsung M21 is very good."

Here Samsung M21 is the entity, camera is the aspect of Samsung M21, and good is the opinion word representing positive sentiment about the camera aspect of entity Samsung M21.

As shown in figure 2, aspect level sentiment analysis can be performed in three steps:- aspect detection, determination of sentiment associated with that aspect, and aggregation of sentiment.

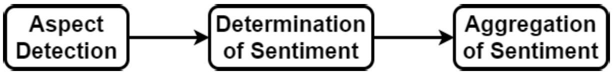

FIGURE 2. Steps in aspect level sentiment analysis.

In the aspect detection step, all the aspect terms are extracted, and similar aspects are grouped into aspect categories. Sentiment label is determined for each occurrence of aspect term based on opinion words in its context. Finally, sentiment label is determined for each aspect category based on (majority of) sentiment labels of occurrences of aspect terms belonging to the aspect category.

An aspect may appear explicitly in a sentence or maybe implied by the words of the sentence. Consider the following two sentences:

"The camera performance is average when it comes to video recording."

"The daylight shots are nothing extraordinary but the low light shots were still better than expectations." [12]

The aspect camera appears explicitly and is termed an explicit aspect in the first sentence, but it is implied and termed an implicit aspect in the second sentence. A sentence with an explicit (implicit) aspect is termed an explicit (implicit) sentence. A dataset of reviews/posts is termed as corpus, and terms feature and aspect are used interchangeably in the field of survey.

The training data is required to pass through many preprocessing and cleaning steps before processing by sentiment analysis algorithms. Like many Natural Language Processing(NLP) applications, the preprocessing steps include lowercasing, tokenization, removing punctuation and stop words, stemming, and lemmatization. In addition, as social networking data is used for sentiment analysis, a few cleaning steps like removing emojis and noise and normalization of words to canonical form are also performed. Finally, the text is encoded to a numeric representation, processed by sentiment analysis algorithms.

III. IMPLICIT ASPECT DETECTION

Implicit aspect detection detects aspects from implicit sentences and is also termed implicit feature identification in the surveyed field. This task may be accomplished using information retrieved from the corpus, concepts of linguistics, and available knowledgebase. The scholarly literature on implicit aspect detection selected through the article retrieval and selection process as shown in figure 3 has been studied, and based on the algorithm used, various approaches are categorized as unsupervised, supervised, and hybrid methods.

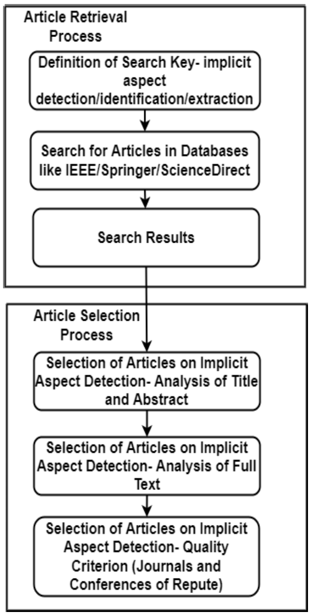
Article Retrieval Process
Definition of Search Key- implicit aspect detection/identification/extraction
Search for Articles in Databases like IEEE/Springer/ScienceDirect
Search Results
Article Selection Process
Selection of Articles on Implicit Aspect Detection- Analysis of Title and Abstract
Selection of Articles on Implicit Aspect Detection- Analysis of Full Text
Selection of Articles on Implicit Aspect Detection- Quality Criterion (Journals and Conferences of Repute)

FIGURE 3. Article retrieval and selection process.

The proportion of surveyed methods belonging to these categories is shown in figure 4.

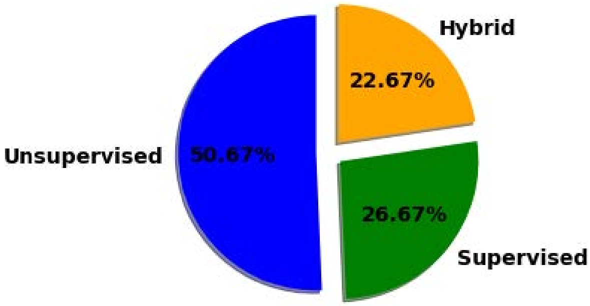

FIGURE 4. Proportion of implicit aspect detection methods belonging to different categories.

Also, the year-wise count of the surveyed methods belonging to these categories is shown in figure 5. We have taken count for two consecutive years for proper and clear presentation.

As represented in figure 6, these categories may be further divided into sub-categories. Unsupervised methods are further divided into co-occurrence-based, topic modeling-based, clustering-based, and other methods, while classification-based, rule-based, and sequence tagging-based methods are sub-categories of supervised methods. Also, hybrid methods are divided into serially and parallelly applied methods.

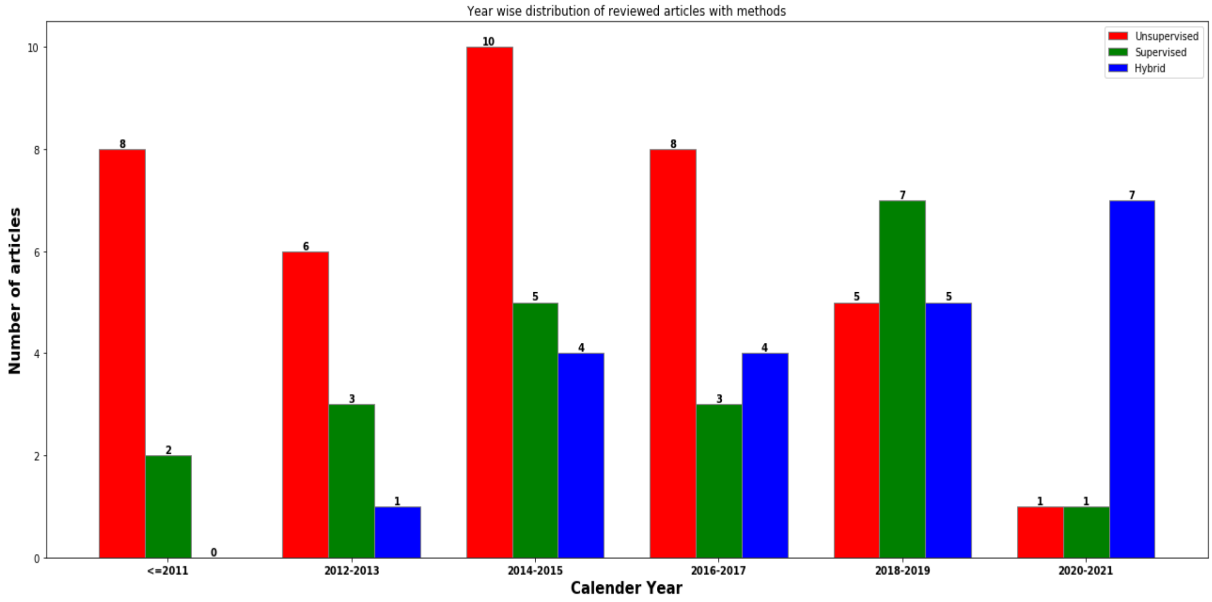

FIGURE 5. Year-wise distribution of reviewed articles with methods.

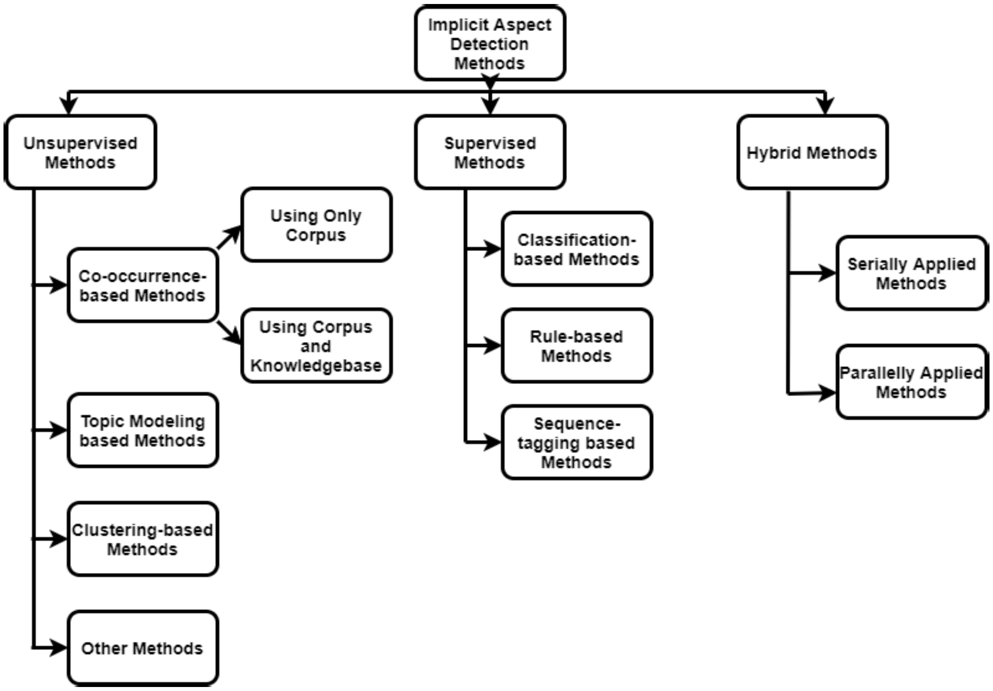
Implicit Aspect Detection Methods
Unsupervised Methods
Supervised Methods
Hybrid Methods
Using Only Corpus
Co-occurrence-based Methods
Using Corpus and Knowledgebase
Topic Modeling based Methods
Clustering-based Methods
Other Methods
Classification-based Methods
Rule-based Methods
Sequence-tagging based Methods
Serially Applied Methods
Parallelly Applied Methods

FIGURE 6. Categorization of methods for implicit aspect detection.

A. UNSUPERVISED METHODS

Training/labelled data (sentences with labelled aspects) is not a requisite in unsupervised methods. Helpful information is extracted from the corpus and applied for the detection of implicit aspects. In addition to information extracted from the corpus, some authors have also used existing linguistic or domain knowledge available on the Internet.

The proportion of surveyed unsupervised methods belonging to different sub-categories is shown in figure 7. Each sub-category with literature belonging to it is discussed in the following sub-sections.

1) CO-OCCURRENCE-BASED METHODS

Initial solutions for implicit aspect detection were co-occurrence-based. These methods use count of co-occurrence

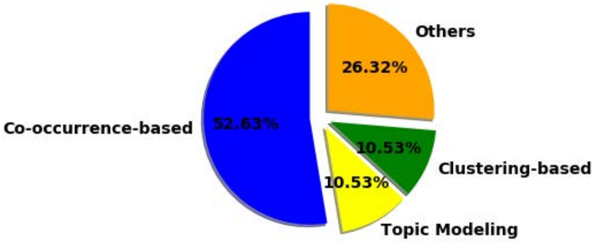

FIGURE 7. Proportion of unsupervised implicit aspect detection methods belonging to different sub-categories.

of terms in given corpus/knowledgebase. Most of the solutions for implicit aspect detection are based on the co-occurrence of terms. Co-occurrence-based solutions generally follow the steps as shown in figure 8.

The text from the corpus and knowledgebase is processed before performing any operation. Preprocessing the text generally involves POS tagging, parsing, and removal of unimportant words. Based on linguistic properties, explicit aspects and opinion words are then extracted. Sentences without any explicit aspect are then identified and termed implicit sentences. Also, using the co-occurrence of words, associations/mappings between words are generated. Finally, implicit aspects are detected using these mappings and notional words from implicit sentences.

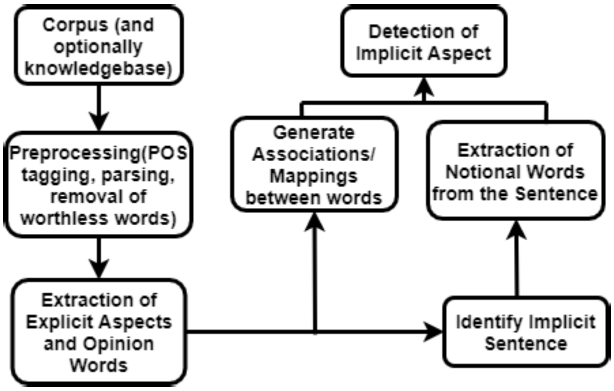
Corpus (and optionally knowledgebase)
Preprocessing(POS tagging, parsing, removal of worthless words)
Extraction of Explicit Aspects and Opinion Words
Generate Associations/ Mappings between words
Identify Implicit Sentence
Extraction of Notional Words from the Sentence
Detection of Implicit Aspect

FIGURE 8. General sequence of steps in co-occurrence-based methods.

First, we will discuss co-occurrence-based methods which utilize only the corpus.

Zhang and Zhu [13] have proposed an approach based on association calculated from co-occurrence. After identifying notional words in the corpus, co-occurrence matrix C is formed based on the co-occurrence frequency of each pair of notional words. Then modification matrix M is created using Qiu's double propagation method to store the modification relationship between opinion and aspect words.

Given an implicit sentence, opinion words are identified first, and then a set of all the aspect words (F_C) that opinion words from the sentence may modify is prepared using M.

For each candidate aspect word f_i in F_C, the average correlation between candidate aspect and notional words of the sentence is determined as in (1):

$$T(f_i) = \sum_{j=1}^{v} \frac{p(\frac{f_i}{w_j})}{v} \tag{1}$$

where v is the count of notional words in the sentence and

$$p(\frac{f_i}{w_j}) = \frac{n_c}{n_b} \tag{2}$$

where n_c is the co-occurrence frequency of f_i and w_j, and n_b is the frequency of w_j.

Aspect with highest $T(f_i)$ is selected as implicit aspect.

Hai and co-authors have suggested a method utilizing association rule mining based on co-occurrence and works in two phases [14]. First, they have created a set of features (aspects) G_F by including noun and noun phrases with a predefined set of dependency relations from explicit sentences. Adjectives and verbs are included in the opinion word set G_O.

Co-occurrence matrix M_{OF} is formed depending on the co-occurrence frequency of pairs of aspect and opinion words. Association rules of the form $O_i \rightarrow F_j$ are then extracted for each opinion word. Features related semantically and conceptually are clustered using the K-means algorithm on the contextual vector representation of aspect words; thus, robust rules are generated for every opinion word.

The opinion word in the implicit sentence is matched with antecedents in rules, and the rule with the aspect cluster having the highest number of aspects is selected, and the representative word for that cluster is assigned as an implicit aspect to the sentence.

Schouten and Frasincar suggested an improvement over [13] and [14] in [15]. Previous works ([13] and [14]) have assumed the same sentential context when the aspect appears explicitly or implicitly, and only aspects that have been found explicitly can be chosen as implicit aspects. They had proposed a method to overcome the drawbacks of previous works; they also distinguished between sentences with implicit aspects and sentences without any aspects.

Data with annotated implicit aspects is used to prepare set F of implicit aspects, set O of all lemmas and their respective frequencies, and co-occurrence matrix C, which stores co-occurrence frequencies of words of the sentence and annotated implicit aspects.

In test data, for each sentence, a score for every candidate implicit aspect f_i is determined as in (3):

$$\text{score}(f_i) = \sum_{j=1}^{v} \frac{(\frac{c_{ij}}{o_j})}{v} \tag{3}$$

where v is total words in the sentence, count of co-occurrence of aspect f_i and lemma j is c_{ij}, and o_j is lemma $j's$ frequency.

Aspect with the maximal score is assigned as implicit aspect if it exceeds the given threshold. A threshold is used to distinguish sentences without any aspect.

A context-based method is proposed by Sun *et al.* [16] for the extraction of implicit aspects. Adjectives are extracted as opinion words; furthermore, noun, noun/noun and verb/verb nearby them are extracted as aspects to build a co-occurrence matrix. A confidence score is calculated for opinion words/aspects with low frequency as in (4):

$$\mathrm{con}(i) = P(i)\sum W \tag{4}$$

$P(i)$ is the probability of opinion word/aspect i, and W is the count of opinion words/aspects with high frequency. Opinion words/aspects with low confidence are removed.

From the training data, a list of candidate aspects for each opinion word is prepared.

Aspects present in the implicit aspect's context are searched, and similarity scores between them and implicit aspect are calculated as in (5):

$$\mathrm{sim}(a, b) = \mathrm{index}(p, q) \times \mathrm{dis}(a, b) \tag{5}$$

where a is the candidate implicit aspect with opinion word p, b is the aspect in its context with opinion word q, and dis(a,b) is the cosine distance depending on co-occurrence matrix. index(p,q) is calculated as in (6):

$$\mathrm{index}(p, q) = \begin{cases} 1, & p \in c, q \in c \\ 0.5, & p \in c, q \in v \text{ or } p \in v, q \in c \\ 0, & p \in v, q \in v \end{cases} \tag{6}$$

where c is a set of clear opinions (which implies specific aspects), v is a set of vague opinions (which implies many aspects).

Given method, selects aspect having maximal score calculated as in (7):

$$\mathrm{score}(a) = \mathrm{con}(a) \times \{\alpha \times \mathrm{sim}(a, b) + \beta \times P(a)\} \tag{7}$$

where $\alpha + \beta = 1$ and determined from training data.

In a method proposed by Liu *et al.* [17], opinion words are used to extract aspects. If a noun/noun or verb/verb phrase is present at either side of the opinion word, it is extracted as an explicit aspect. If a sentence contains only opinion words, it is termed as an implicit sentence.

The co-occurrence matrix is generated from explicit sentences to store the co-occurrence frequency of opinion-aspect pairs. For an opinion word with low confidence, if its associated aspects also have low confidence scores, then the opinion word is deleted, and the co-occurrence matrix is recalculated. The same procedure is repeated, but the roles of opinion words and aspects are changed.

The confidence score is calculated as in (8):

$$\mathrm{con}(x_i) = \frac{p(x_i)}{N} \tag{8}$$

where x_i is either aspect or opinion word, and N is count of aspect/opinion words.

If an implicit sentence contains a vague opinion word, then the whole entity is assigned as an implicit aspect. For clear opinions, opinion word groups are formed using synonyms and antonyms. Explicit aspects modified by words in the opinion group become candidates for implicit aspect and aspect with the highest importance (calculated as in (9)) is assigned as implicit aspect.

$$\mathrm{imp}(x_i) = \mathrm{weight}(x_i) \times \{\sup(x_i) + \mathrm{con}(x_i)\} \tag{9}$$

where $\sup(x_i)$ and weight(x_i) are calculated as in (10) and (11) respectively.

$$\sup(x_i) = \frac{p(x_i)}{N(X)} \tag{10}$$

$$\mathrm{weight}(x_i) = \sum_{f_i \in F(x_i)} \mathrm{con}(f_i) \tag{11}$$

where $N(X)$ is the count of candidate aspects, $F(x_i)$ is a set of opinion words corresponding to aspect x_i.

A graph-based approach is suggested by Bagheri *et al.* [18] to identify implicit aspects using set of explicit aspects and polarity lexicon generated by them in the preceding step. Opinion words and aspects are represented as nodes with edges between opinion words and aspects. Weight of edge (A,O) is assigned using (12)-

$$W_{AO} = \log\left\{\frac{\mathrm{CO}_{AO}}{D_A \times D_O} + \epsilon\right\} \tag{12}$$

where W_{AO} is the weight of edge (A,O), CO_{AO} is the co-occurrence frequency of A and O, D_A is the count of different opinion words which co-occur with aspect A, D_O is the count of different aspects which co-occur with opinion word O and a parameter ϵ is used to avert the fraction from becoming zero. They have defined a gap-threshold to differentiate between weights of aspects for a given opinion word. Depending on the gap-threshold, most probable implicit aspects are extracted for each opinion word.

Su *et al.* [19] have proposed a pointwise mutual information (PMI) based method to identify implicit aspects. Noun and noun phrases are extracted as features/aspects. A set of opinion words is constructed manually by extracting opinion words from review webpages. The set of opinion words is expanded by using synonyms and antonyms from the Chinese Concept Dictionary (CCD).

PMI demonstrates the genuine association between two words and is calculated as in (13):

$$\mathrm{PMI}(w_1, w_2) = \log\left\{\frac{P(w_1\&\&w_2)}{P(w_1) \times P(w_2)}\right\} \tag{13}$$

where w_1 and w_2 are two words, and $P(w_1\&\&w_2)$ is the probability of w_1 and w_2 co-occurring in a sentence, and $P(w_1)$ / $P(w_2)$ is the probability that word w_1 / w_2 will occur.

For each opinion word, its PMI score is calculated with each feature (aspect), and then it is mapped to one or more features (aspects) based on the PMI score.

The opinion word in an implicit sentence is extracted, and the aspect mapped to that opinion word is assigned to the sentence.

Wang and Wang [20] have proposed an iterative and unified process to identify the sets of aspects and opinion words, given a small seed set of opinion words. Aspects are extracted based on identified opinion words; also, opinion words are extracted based on identified aspects.

They have defined RMI (revised mutual information) to measure the association of opinion words and aspects. It is calculated as in (14):

$$\mathrm{RMI}(x, y) = A \times \log \left\{ \frac{(N \times A)}{(A + B) \times (A + C)} \right\} \tag{14}$$

where N is the count of reviews, A is the count of reviews where x and y have co-appeared, B is the count of reviews where x has appeared but y is absent, and C is the count of reviews where y has appeared but x is absent.

For an implicit sentence, the opinion word is extracted, and the aspect word with the highest RMI is assigned as an implicit aspect of that sentence.

Mankar and Ingle [21] have used Part of Speech (POS) tagging to extract nouns as aspects and adjectives/adverbs as opinion words. Aspect-based sentences are then selected by eliminating irrelevant nouns using PMI. Aspect-opinion pairs are then generated using association rules and used for detecting implicit aspects.

In the method suggested by Kama *et al.* [22], first, groups of nouns and sentiments are extracted, and their mapping is generated. A list of aspects is supplied as input, and only noun groups corresponding to the aspect in the given list are kept, and the remaining are discarded. For an opinion word in an implicit sentence, its mapping with aspects is extracted, and aspect with the highest association is considered, and the following threshold criteria are evaluated

a) Frequency count of sentiment word
b) Co-occurrence
c) Difference amongst this mapping and mapping for the second opinion word for the selected aspect

Furthermore, the evaluated aspect is assigned as an implicit aspect to the sentence.

The process given by Makadia *et al.* [23] for identifying implicit aspects involves the prediction of sentiment orientation, generation of aspect-opinion pairs, replacement of synonym words with corresponding aspect word, and counting the frequency of each pair.

In sentiment prediction, the sentence is classified as a positive or negative sentence. After (POS) tagging, nouns and adjectives are extracted and are stored as aspect-opinion pairs. Only the nouns denoting aspect words or their synonyms are considered for aspect-opinion pair construction. The aspect word then replaces its synonyms, and the frequency for each unique pair is counted using RapidMiner.

The opinion word is identified from an implicit sentence, and its frequency with all the aspects is checked. The aspect with the highest frequency is assigned as an implicit aspect to the sentence. If the frequency count with the given opinion word is the same for two different aspect words, then the word's total frequency is considered.

Adjusted Laplace smoothing is used in calculating the weight of relation between opinion word and aspect in the method suggested by Omurca and Ekinci [24].

Set of sentiment words and explicit aspects are represented as $S = \{S_1, S_2, .., S_s\}$ and $E = \{E_1, E_2, ., E_k\}$ respectively. The sentiment words from implicit sentences are represented as $I = \{I_1, I_2, .., I_n\}$. A graph between every element $S_i \in S$ and $E_j \in E$ is drawn, and weight w_{ij} of the edge between S_i and E_j is calculated using Naive Bayes probability as in (15):

$$w_{ij} = P(E_j|S_i) = \left\{ \frac{P(S_i|E_j) \times S_i + 1}{P(E_j) \times \psi_j + \phi_i} \right\} \tag{15}$$

where ϕ_i and ψ_j are incorporated to perform Laplace smoothing. ϕ_i denotes the number of explicit aspects that appear together with S_i, and ψ_j represents the number of sentiment words that appear together with E_j.

The weights between each sentiment word $I_i \in I$ and each explicit aspect $E_j \in E$ are determined and explicit aspect E_j with highest weight w_{ij} is assigned as implicit aspect wherever I_i appears in an implicit sentence.

After explicit aspect-sentiment word matching, implicit aspect extraction is included as a further step in the procedure suggested by Karagoz *et al.* [25].

In the explicit aspect-sentiment word matching step, they had counted the number of times an explicit aspect appeared with a sentiment word. For the explicit aspect-sentiment word pair with maximal count, sentiment word is considered to imply explicit aspect based on a threshold value, which depends on the count for sentiment word and difference of count between sentiment word of selected aspect-sentiment word pair and sentiment word of aspect-sentiment word pair with the second maximal count. This process is applied to every sentiment word. If an implicit sentence contains a sentiment word, the corresponding matched aspect is assigned as an implicit aspect of the sentence.

In [26] Dadhaniya and Dhamecha have generated feature-opinion pairs from training data. If an implicit sentence contains an opinion word x, then the feature-opinion pairs containing x are scanned, and pair with the highest frequency is selected, and the feature from that pair is assigned as an implicit aspect of the given sentence.

In [27], Schouten *et al.* have proposed an approach based on spreading activation algorithm to assign predefined aspect categories to sentences. The proposed approach can also be used for the identification of implicit aspects. To keep the method unsupervised, they have used a seed set of words for every category. An occurrence vector N is prepared with remaining lemmas and their respective frequencies after removing stop words and low-frequency lemmas from data. Then co-occurrence frequency for each lemma pair is stored in matrix X. Co-occurrence digraph is then constructed using X and N. Node for each notional word is created, and an edge $< i, j >$ exists if their co-occurrence frequency is higher than a given threshold. The weight of the edge is calculated as in (16):

$$w_{ij} = \frac{X_{ij}}{N_j} \tag{16}$$

where w_{ij} is the count when i has co-occurred with j, and N_j is count of occurrence of j.

Nodes corresponding to words in the seed set of an aspect category c are assigned an initial activation value of 1, and the remaining nodes are assigned activation value 0. Activation values are spread by firing vertices iteratively, and activation values of adjacent nodes are updated using a decay factor of δ. Updated activation value for node j for category c is calculated as in (17) when node i is fired-

$$A_{c,j} = \min\{A_{c,j} + A_{c,i} \times w_{ij} \times \delta, 1\} \tag{17}$$

For all the nodes A with final activation value greater than threshold τ_c, association rules of the form $A \rightarrow c$ are mined. Aspect categories are assigned to new sentences using the generated set of rules.

Some authors have also used available knowledgebase and corpus, and methods suggested by them are discussed next.

In [28], Schouten *et al.* have developed a method based on the co-occurrence of aspects and synsets from WordNet. The co-occurrence frequency of labelled implicit aspects with synsets from Wordnet is stored in a matrix. The synsets represent the meaning or semantics of a word for a given context.

The score for each aspect is calculated using the co-occurrence of aspects and each synset in the given sentence. A fraction of all the semantically related synsets' co-occurrence frequency is incorporated in the calculation of the score (18):

$$\text{score}(a_i) = \frac{1}{v}\left(\sum_{j=1}^{v}\left(\frac{c_{ij}}{f_j} + \sum_{r \in R}\sum_{k=1}^{K_r(j)} w(r)\frac{c_{ik}}{f_k}\right)\right) \tag{18}$$

where v is the count of the synsets related to the sentence, a_i is the i^{th} candidate aspect, f_j is the j^{th} synset related to the sentence, c_{ij} is the co-occurrence frequency of a_i and f_j, r is the semantic relation related to f_j. R is a set of semantic relations, $K_r(j)$ is the set of the synsets related to f_j and r, $w(r)$ is the weight related to r, c_{ik} is the count of co-occurrence of a_i and synset k in $K_r(j)$, and f_k is the frequency of synset k.

Aspect with the maximal score is assigned as an implicit aspect to the sentence if it betters a trained threshold.

In [29], Song *et al.* have divided implicit sentences into sentences with context information $S_{context}$ and sentences without context information $S_{non-context}$.

For sentences in $S_{non-context}$, the association between words is calculated using Wikipedia. Each Wikipedia concept is represented as a word vector with a TF-IDF value as association strength between words and concepts. Similarly, a word may be represented as a series of Wikipedia concepts and association strength between them. The similarity between two words/concepts can be measured using cosine distance between vectors, representing the word/concept.

Set of domain-specific opinion words S_O and set of domain-specific feature words S_T is prepared from explicit sentences. Then a set of related synonym features (aspects) R_{TS} is prepared for each opinion word, and similarly, a set of related synonym opinion words R_{OS} is prepared for each feature word (aspect).

For sentences in $S_{context}$, centering theory and named entities are used to prepare candidate feature (aspect) set $S_{candidate}$. A set t_{FC} is prepared from $S_{candidate}$ with candidates with high TF-IDF value. Similarity between words is calculated using cosine distance.

An opinion tree is then prepared from gathered information with opinion words as nodes and feature (aspect) words as their children. This opinion tree is then used to identify aspects for implicit sentences.

Zhang *et al.* [30] have first generated a set of 8 aspect categories (for cosmetic products) based on suggestions of industry experts. They have expanded the set to include three more, most frequently discussed aspect categories.

Nouns, verbs, and adjectives with a frequency of more than ten are considered candidate features. Then explicit features are grouped for each aspect category using the concept of synonym and antonym, sharing morphemes and similarity based on HowNet.

Implicit features are grouped for each aspect category based on their collocation with explicit features. Multiplication of PMI and frequency is used as a measure of collocation and calculated as in (19):

$$\text{Frequency} \times \text{PMI}(f, w) = p(f, w) \times \log\left\{\frac{p(f, w)}{p(f) \times p(w)}\right\} \tag{19}$$

where $p(f, w)$ is the co-occurrence frequency of explicit feature f and candidate implicit feature w, and $p(f)$ and $p(w)$ are frequencies of f and w, respectively. Candidate implicit feature is assigned to aspect category if it has highest collocation score with feature words of that aspect category.

Prasojo [31] has proposed a method based on adjective to aspect mapping and WordNet lexical database. A set of comments with tagged entity and aspects is used to map entity-adjective pairs with aspects. In this mapping, they have counted the co-occurrence of entity-adjective pair and aspect.

For a sentence with a pair of adjective and entity but without explicit aspect, the aspect having the highest co-occurrence is assigned using the mapping. If more than one aspect has the same (highest) frequency, the lexical database WordNet is used, and the similarity score defined in WordNet is calculated for all the candidate aspects, and the aspect with the highest similarity score is selected.

Nandhini and Pradeep [32] have proposed a co-occurrence and ranking-based algorithm for implicit aspect detection. First, opinionated sentences are separated from non-opinionated sentences, and nouns in the opinionated sentences are extracted as explicit aspects. Adverbs and adjectives are extracted as sentiment words. Then co-occurrence of sentiment words and explicit aspects is calculated, and sentiment words are mapped to explicit aspects with which it had co-occurred the most. This mapping is used for the identification of implicit aspects.

2) TOPIC MODELING BASED METHODS

Topic modeling is a statistical method for determining hidden topics from a given collection of text documents. For implicit aspect detection, every sentence is considered as a document, and topic modeling is applied to detect topics (aspects) for that document (sentence). Latent Dirichlet Allocation (LDA) is the most prevalent algorithm for topic modeling, and its working is illustrated in figure 9. Methods with the application of LDA are discussed next.

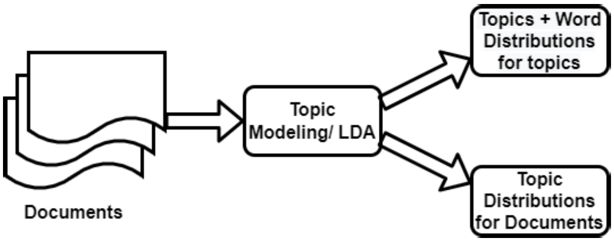
Documents
Topic Modeling/ LDA
Topics + Word Distributions for topics
Topic Distributions for Documents

FIGURE 9. Working of topic modeling/LDA.

Xu *et al.* [33] have suggested a topic model-based method to retrieve aspects and aspect-specific opinion words. The extracted lexicon of aspect-specific opinion words is then used for the detection of implicit aspects.

Topic model LDA was adapted such that all extracted topics correspond to some aspect by assigning all words in a sentence to one topic.

First, an aspect is assigned to a sentence, and for each word in the sentence, its subjectivity label $\zeta_{d,s,n}$ (factual or opinion word) and sentiment label $l_{d,s,n}$ (positive or negative) is determined for n^{th} word of sentence s in the document d.

Using Gibbs sampling, word distributions concerning aspect-specific +ve and -ve sentiments ($\phi^{t,pos} + \phi^{t,neg}$) are approximated, where t is the aspect. High probability words in $\phi^{t,pos}$ and $\phi^{t,neg}$ are chosen as aspect-specific opinion words.

If a non-explicit sentence contains opinion words from the lexicon related to a specific aspect t, then t is assigned as an aspect for that sentence.

Lau and co-authors have given an LDA-based topic modeling approach to extract implicit and explicit aspects [34]. For LDA, each document is characterized by multinomial distribution θ, and a term is generated for the given topic using multinomial distribution ϕ, controlled by Dirichlet prior β. Directly computing θ and ϕ will require very high computation time. They have calibrated Gibbs sampling and Markov chain algorithm to estimate θ and ϕ. The approximation $\bar{\theta}$ and $\bar{\phi}$ are given in (20) and (21) respectively:

$$\bar{\theta} = \frac{C_{np}^{ZD} + \alpha}{\sum_{n\prime \in V} C_{n\prime p}^{ZD} + |Z|\alpha} \tag{20}$$

$$\bar{\phi} = \frac{C_{mn}^{VZ} + \beta}{\sum_{m\prime \in V} C_{m\prime p}^{VZ} + |V|\beta} \tag{21}$$

where C_{mn}^{VZ} is count matrix that stores count when term m is assigned to topic n, V is a set of vocabulary, Z is a set of topics, C_{np}^{ZD} is count matrix that stores count when topic Z is assigned to document D. Gibbs sampling is invoked with different count for topics and smallest count for topics which achieves a good perplexity is used. The top 10 topics are used as aspects.

A knowledge-based topic modeling (KTM) approach was given by Zhang *et al.* [35] to extract implicit aspects.

After removing irrelevant elements and tokenization, nouns, adjectives, verbs, and adverbs are selected as candidates, and other words are removed. Then PMI and χ^2 test are used to filter words that are highly related to emotions. A topic set is initialized by including synonyms and antonyms of emotion words. Then PMI and TF-IDF are used to calculate the similarity of words and enhance the topic set. Constraints in form of indicator function δ are included in the topic updating function of LDA to incorporate existing knowledge. Value of δ is 1 in case the word is present in the topic set and 0 otherwise. In the output of the KTM, words under a topic are highly related to each other.

Rules of the form of (emotion, emotion indicator) are learnt from KTM, and explicit sentences and indicators from implicit sentences are used to identify implicit aspects by applying rules generated in the previous step. They have also developed a four-level hierarchy of emotions.

In [36], Ekinci *et al.* have suggested a method for extracting implicit aspects that incorporate semantic information in LDA to improve its performance. Although they have not entirely implemented the proposed solution, but have suggested using semantic information from Bebelfly, which improves the performance of LDA.

3) CLUSTERING BASED METHODS

Clustering is the process of dividing a set of items into groups so that items in a group (cluster) are analogous to each other and disparate to those in other groups (clusters). Generally, explicit aspects and opinion words are clustered to generate more robust mappings/associations and in turn, robust results.

Su and co-authors [37] have proposed a mutual reinforcement approach for aspect-level sentiment analysis. The outcome may also be used for the detection of implicit aspects. Noun and noun phrases are extracted as aspects, and adjectives are extracted as opinion words. Opinion words and aspects are represented in the form of a vector to perform clustering. The vector consists of PMI between instance and its context, inner word PMI within the phrase, and POS tag of context. A link weight matrix $R = \left[r_{ij}\right]$ is constructed to store pairwise weights between the set of features (F) and opinions (O), where r_{ij} is the co-appearance frequency of f_i and o_j. Objects in F and O are clustered based on the similarity of objects of the same type termed as intra-relationship. Impact of surrounding opinion word (aspect word) on clustering aspects (opinions) is also incorporated as inter-relationship.

Clustering begins from any type of object, and the results update the link information, thus affect the clustering of other types of objects. This process is repeated until clustering results converge for both types of objects. Knowledge in the form of compatibility, incompatibility, and similarity (calculated based on the textual structure) is incorporated in the clustering process to improve the results of the clustering process.

Association set between groups of aspects and opinions is established using the strongest n inter-links. This pre-constructed association set may be used for the identification of implicit aspects.

Extraction of aspect words and their clustering into aspect categories are combined in the solution proposed by Chen *et al.* [38]. A set of candidate aspect words is prepared by extracting nouns and noun phrases as candidates for explicit aspects, and verbs and adjectives as candidates for implicit aspects.

A novel clustering approach was proposed to group the candidate words in different aspect categories. Most frequent candidate words are clustered, and seed clusters are generated, then the remaining candidate words are assigned to the closest seed cluster. Distance between candidates/clusters needs to be calculated for clustering, and the proposed approach saves time by clustering only frequent candidates. Also, the most frequent words are more likely to be aspects.

A similarity measure specific to the domain is proposed, including corpus-based statistical association and the general semantic similarity. UMBC Semantic Similarity Service is used to store general similarity of candidates. An association matrix is also prepared to store normalized pointwise mutual information (NPMI) between candidates representing the statistical association.

Also, two clusters cannot be merged if the distance is greater than the specified threshold or one cluster does not contain any noun or noun phrase, or the total of frequencies of candidates from given two clusters appearing together in the same sentence is higher than frequencies of candidates appearing together in the same document but in different sentences. These constraints are termed problem-specific merging constraints.

Verbs and adjectives from the aspect cluster may be used as an indicator for that aspect category, i.e., if an implicit sentence contains a verb or adjective from a given cluster, the corresponding aspect category may be assigned as an implicit aspect for the sentence.

A new method combining context information and two different opinion types (clear and vague) was proposed by Wu and Liu [39] to retrieve implicit aspects.

First, they have used dependency parsing to extract explicit aspect-opinion pairs. The aspects are clustered based on shared words and the similarity of associated opinion words is calculated from clustered aspect-opinion co-occurrence matrix. The similarity of aspects appearing in the same sentence is considered zero.

A candidate feature context information matrix is constructed to store the co-occurrence of context words and features in a feature cluster from explicit sentences to extract implicit aspects. Implicit sentences are identified using opinion words, and three different cases are handled in different ways.

The product is assigned as an implicit aspect in a sentence with only vague opinions and no verbs and nouns.

Suppose the sentence contains only clear opinion and does not contain verbs and nouns, then clustered aspect-opinion co-occurrence matrix is utilized to extract implicit aspect. A confidence score is calculated as in (22) for each candidate aspect, and the aspect with maximal confidence value is assigned as implicit aspect.

$$\mathrm{con}(f_i) = \frac{n_{f_o}}{n_{f_i}} \tag{22}$$

where n_{f_o} is the weight of candidate aspect f_i and clear opinion in clustered aspect-opinion co-occurrence matrix, and n_{f_i} is the count of opinions co-occurred with candidate aspect f_i.

For remaining cases, strategy based on candidate feature context information matrix as suggested in [13], is followed to extract implicit aspects.

Hai and co-authors [40] have suggested an association-based method to identify explicit aspects, opinion words, and implicit aspects. A seed set of aspects and an empty seed set of opinions are supplied, and all the explicit aspects and opinion words are extracted based on the correlation between aspect and opinion words (AO), aspect words (AA) and opinion words (OO).

Noun and noun phrases that appear as subject/object are considered candidate aspects, and adjectives and verbs are considered candidate opinion words. For each candidate, its correlation is calculated with the elements of the seed set, and using trained thresholds seed sets are expanded to include candidates with a correlation higher than the specified threshold. Two different methods are suggested based on two different measures of correlation, Latent Semantic Analysis (LSA) and Likelihood Ratio Test (LRT). After retrieval of explicit aspects and opinion words, explicit aspects are clustered using k-means algorithm.

For an opinion word from implicit sentence, its correlation is calculated with all the explicit aspects, and the cluster with the highest average correlation with the opinion word is selected. The representative word from the cluster is assigned as an implicit aspect. Even if the new opinion word is absent from the set of opinion words, its synonym/antonym is searched, and if it is present in O, its average correlation is calculated and implicit aspect is assigned accordingly.

4) OTHER METHODS

Authors have developed many different unsupervised approaches for the detection of implicit aspects. It is impossible to plan a category for all of them; hence, approaches that do not fall into any previously discussed categories are discussed here.

Santu *et al.* [41] have proposed a solution incorporating generative feature models to mine implicit aspects from reviews. Given the set of reviews of a product and its aspects of interest, word distributions (feature language models) are proposed for each of the k aspects, denoted by $\gamma 1, \gamma 2, \ldots, \gamma k$. They proposed to fit a mixture model with feature language models as constituents to review data to learn feature language models in an unsupervised fashion analogous to topic models like LDA/PLSA. Special language model γB is given to model the noisy words.

For a given sentence, the probability for every aspect is calculated. The calculation of probability depends on the sentence's words and the word distributions for the aspects. Aspects having probability greater than the specified threshold are assigned as implicit aspects for the sentence.

A rule-based method which uses Normalized Google Distance (NGD), is suggested by Rana and Cheah [42] to extract implicit aspects.

After POS tagging, all the aspect terms are replaced with the word 'aspect', and all the opinion terms are replaced with the word 'opinion'. Then sequential rules among opinions and aspects are generated using extracted sequential patterns.

If an aspect term is associated with some opinion term by some sequential rule, it is extracted as an explicit aspect.

Sentence without aspect word but with opinion word is called an implicit sentence. For an implicit sentence, NGD of the opinion word is calculated with all the aspect terms, and the aspect term with the smallest value of NGD is selected as the implicit aspect.

If two terms have never co-occurred on the same web page, they have infinite NGD, and if the terms always appear together, they have zero NGD.

Galliat *et al.* [43] have proposed supervised and unsupervised methods based on stock-investment taxonomy to extract aspects (both implicit and explicit) from financial microblogs.

A taxonomy with seven classes and 32 subclasses is defined, and the corpus is manually annotated with class and subclass labels. These labels are analogous to aspects.

An unsupervised method named Distributional Semantic Model (DSM), based on word embeddings, was proposed to compute semantic relatedness using Word2Vec. After tokenization and POS tagging, noun phrases and verb phrases with modifiers like adverbs/adjectives are extracted as candidates. The similarity of vectors of candidates with vectors of classes is calculated using intra implementation (Freitas *et al.*) of the cosine similarity measure. Class label with the highest similarity score is assigned.

In [44], Yu and co-authors have organized different aspects of a product in a hierarchy by combining domain knowledge (like product specifications) and customer reviews. Then customer reviews are also organized based on aspect hierarchy.

They have observed that sentiment (opinion) terms are good indicators for implicit aspects. Hence each review is illustrated in the form of a feature vector with sentiment terms as features. For all aspect nodes in the aspect hierarchy, its centroid is calculated as the average of the feature vectors of reviews related to that particular aspect.

For an implicit sentence, its feature vector is first generated, and then its cosine similarity with centroids of all aspects is calculated. Aspect with the highest similarity is selected as an implicit aspect for the sentence.

Qiu proposed a semantic ontology-based method to identify implicit aspects [45]. After POS tagging, noun and adjectives are extracted as aspects and opinion words.

An entity and corresponding ontology are taken as input, and implicit aspects (related to an entity) are identified for opinion words by identifying semantic relations between terms in the ontology and opinion words. The calculation of semantic relatedness is based on PMI.

Meng and Wang [46] have clustered product specifications from various sources to prepare a specification tree, and its nodes are used as aspects. They have used the association of aspects and units of measures to determine implicit aspects. For an implicit sentence with a unit of measure, its associated aspect is assigned as an implicit aspect. A dictionary of units and regular expression is used for the extraction of aspects.

Shi and Chang have proposed a method based on hierarchical product feature model [47]. For each product, a concept model is constructed. Every leaf node has two child nodes, "Name" and "StrongOpinionWord". Node "Name" contains aspects and their synonyms. "StrongOpinionWord" has three children, "Positive", "Negative",and "Neutral", and each of these stores adjectives, verbs,and adverbs related to parent aspect.

After punctuation filtering and elimination of questioning segments, product features (aspects) are identified by applying the concept model. In the case of an implicit sentence, a matching word from "StrongOpinionWord" is searched, and if found, the parent aspect is assigned as the implicit aspect.

Zainuddin and co-authors [48] have used dependency relationships among aspects and opinion words to determine implicit aspects. Direct dependencies (det, amod, aux, dobj, advmod, nsubj, xcomp) and transitive dependencies (a distance of one dependency relation) are used to determine implicit aspects. Stanford's Dependency Parser is used for the extraction of dependency relations. Explicit aspects are extracted using association rule mining.

Wan *et al.* [49] have extracted words, POS tagged as nouns, as explicit aspects. They have grouped aspects into categories by utilizing morphemes. Some POS rules are defined to extract implicit aspects/indicators (e.g. $(> 1)|v$ more than one morpheme and POS tag is v). Indicators are then mapped to aspects by decomposing words and using regular expressions.

In a case study on the use of ontology for aspect-based opinion mining, Cadilhac *et al.* [50] have stated that ontology properties may be used to extract implicit aspects. Ontology properties define the relationship between concepts of ontology, e. g. the property "look at" relates to "customer" and "design" concepts.

B. SUPERVISED METHODS

Methods that require training data, i. e. sentences with labelled aspects, fall into this category. Labelled data is used to train an algorithm, and then the algorithm is used to predict implicit aspects for new sentences.

The proportion of surveyed supervised methods belonging to different sub-categories is shown in figure 10. Each sub-category with literature belonging to it is discussed in the following sub-sections.

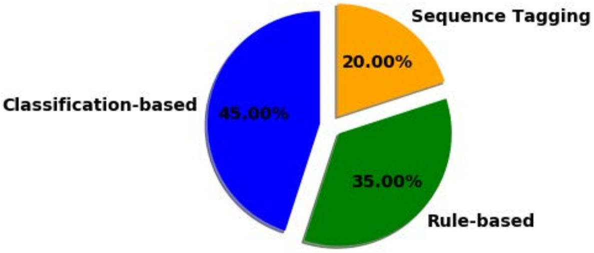

FIGURE 10. Proportion of supervised implicit aspect detection methods belonging to different sub-categories.

1) CLASSIFICATION BASED METHODS

Classification is the task of designating a new observation to a class from the predefined set of classes, depending on a training data set having observations with a known class.

Detection of implicit aspect is generally considered as a multi-class text classification problem with aspects as class labels.

As shown in figure 11, training data (sentences with assigned aspects) are processed to extract features. These features are used to train a classifier. The trained classifier is then used to assign aspects to implicit sentences.

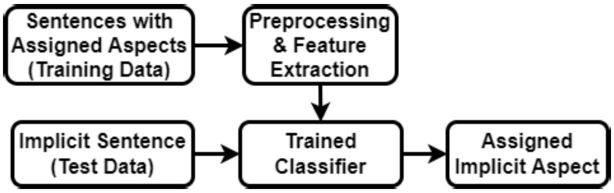
Sentences with Assigned Aspects (Training Data)
Preprocessing & Feature Extraction
Implicit Sentence (Test Data)
Trained Classifier
Assigned Implicit Aspect

FIGURE 11. General sequence of steps in classification based methods.

Zeng and Li suggest an approach based on classification for the identification of implicit aspects [51]. Feature-opinion pairs (f, o) are extracted based on dependency parsing using three rules based on subject-predicate structure and DE structure of Chinese dependency grammar for extraction of feature-opinion pairs. Feature-opinion pairs (f, o) are then clustered for each opinion word o based on sharing words and lexical similarity of aspects.

A topic-feature-centroid classifier is designed to classify implicit sentences into the most probable feature-opinion pair (f, o). A lexicon set is constructed by including only nouns, verbs, and adjectives from training data and denoted as $L = \{wf_1, wf_2, \ldots, wf_L\}$. The centroid for feature-opinion pair (f_i, o_j) is denoted as a word vector $\text{Centroid}_j = \{wf_{1j}, wf_{2j}, \ldots, wf_{Lj}\}$, where wf_{kj} is the weight for word wf_k and is calculated as in (23):

$$wf_{kj} = f_{w_k} \times \log\left(\frac{|C|}{|Cf_{w_k}|}\right) \tag{23}$$

f_{w_k} is the frequency of word w_k in the document for feature-opinion pair (f_i, o_j), C is the count of feature-opinion pairs containing opinion word o_j, and Cf_{w_k} is the count of feature-opinion pairs containing word w_k.

After obtaining the centroid vector for every feature-opinion pair, a cosine measure is used to classify implicit sentences as shown in (24):

$$C = \operatorname{argmax}\left(\overrightarrow{\mathbf{S_i}} \cdot \overrightarrow{\mathbf{Centroid_j}}\right) \tag{24}$$

where $\overrightarrow{\mathbf{S_i}}$ is a word vector representation of sentence S_i. If the pair (f_i, o_j) is identified for sentence S_i then f_i is assigned as an implicit aspect to sentence S_i.

Fei and co-authors [52] have come up with a dictionary-based method to identify aspects indicated by adjectives, and results can be used for implicit aspect identification. Adjectives are extracted from the text to form a set $A = \{A_1, A_2, \ldots, A_r\}$, and online dictionaries are crawled for their glosses. For adjective $A_i \in A$, its glosses are POS tagged, and nouns are retrieved to constitute a set C_i of candidate aspects for the adjective A_i. Using collective classification, they have classified candidate aspect $C_{ij} \in C_i$ as an aspect or not an aspect for adjective A_i.

Data is denoted as a graph having pairs of adjective and one of its candidate aspects $\left(A_i, C_{ij}\right)$ as nodes to perform collective classification. Node $\left(A_i, C_{ij}\right)$ is denoted using a feature vector X_{ij} and assigned with a class label{positive(aspect), negative(no aspect)}.

A collective classification algorithm named iterative classification algorithm is used to perform the task. A classifier h is trained like a traditional supervised method using labelled data. Using h, labels are assigned to each unlabelled node U_{ij}. Then, feature vector X_{ij} is calculated for each U_{ij} as some features depend on adjacent nodes' labels.

Iterations of the classifier are performed until there is no change in the labels for all nodes. To eliminate bias, a random order of nodes is generated for each iteration. The identified aspects for the given adjective may be assigned as implicit aspect if the adjective is present in an implicit sentence.

Three distinct classifiers (Naive Bayes (NB), Random Forest, and Support Vector Machine (SVM)) are tried by Hu *et al.* [53] for identification of aspects (both explicit and implicit) from the context related to aspects from free-form reviews.

After data preprocessing, n-grams in the context of aspect word are extracted and used as features to train classifiers. A list of aspect words and a contextual window size W (3 in this case) is given as input with a set of sentences. If a sentence S contains an aspect a_i, all the n-grams (uni, bi, and tri) within the specified window from a_i's position are extracted and stored into set G_i, representing n-grams related to aspect a_i. All other n-grams which do not relate to any aspect are stored into set G_{other}.

For identification of aspect, first, we apply a classifier that classifies the sentences in two classes- with aspect and without aspect. A multi-class classifier is then applied to sentences with aspects that classify the sentence into one aspect. A vector is generated for each sentence having n-gram information to train the classifiers.

NB, SVM, and Random Forest are implemented, and their performance is compared, and SVM performed best.

Mashkin suggests incorporating Conditional Random Fields (CRF) and NB classifiers to detect implicit aspects [54]. For extraction of explicit aspects, a pipeline of three CRF models is provided. Each CRF performs a subtask-estimation of BIO (beginning of input-output) label, category, and sub-category label estimation, respectively.

Extraction of implicit aspects is performed with the help of two NB classifiers. The sequence of the sentences in the form of bag-of-words is supplied as input. The first NB classifier predicts whether the sentence contains implicit aspects or not, and the second NB classifier predicts the category and sub-category of the implicit aspect word by picking the most probable of 12 categories.

In [55] Hajar and co-author have proposed using definition and synonym relation from WordNet (WN) to enhance training data for NB classifier, which is used to identify implicit aspects.

After POS tagging, adjectives and verbs are extracted as implicit aspect terms (IAT). Synonyms from WN are extracted for all IAT and represented by set S. Also, nouns are extracted from the phrases defining IAT from WN and denoted by set D. Five different enhancement policies, S, D, $S \cap D$, S-D, and D-S, are tried and set which have the best impact on the performance of the NB classifier is selected to enhance training data.

A method to identify aspect and polarity from the sentences not having opinion and aspect words was proposed by Chen *et al.* [56]. Sentence segments in training data are divided into four categories-T1(with opinion word and aspect word), T2(with opinion word but without aspect word), T3(with aspect word but without opinion word and T4(without opinion and aspect word). T1-T4 and T4-T1 pairs are extracted from the training data, and segments of type T4 are assigned opinion words and aspect words from paired T1 segments.

The opinion dictionary is constructed from the training data, and then the aspect dictionary is constructed by including words with POS tag NN and nsubj dependency with some opinion word. Ambiguous segment sequences X-T4-Y, where X/Y can be T1/T2/T3, are discarded from training data. The aspect words are divided into ten classes to avoid sparseness.

A binary polarity classifier and a 10-class aspect classifier are trained and used to assign polarity and aspects to the segments of type T4.

Yu *et al.* [57] have proposed using product aspect hierarchy and hierarchical classification to extract implicit aspects. They have observed that some specific sentiment words usually modify implicit aspects (e.g., long for size). These associations are learnt from hierarchy and used to identify implicit aspects.

The hierarchical classifier identifies implicit aspects for a given input (question) by greedily searching in the product hierarchy. The search starts from the root and stops when the relevance score is lower than the threshold (or at the leaf node). The SVM classifier calculates the relevance score.

In [58], Afzaal and co-authors have presented a decision tree-based method to identify implicit aspects. First, noun and noun phrases are extracted as explicit aspects, and then aspects with similar meanings are grouped. Frequent aspects and explicit sentences are used to create decision trees for individual aspects, having words as decision conditions and aspects as a class. Implicit sentences are divided into words and supplied as input to all decision trees, and all the assigned aspects are returned.

Galliat *et al.* [43] have proposed supervised and unsupervised methods based on stock-investment taxonomy to extract aspects (both implicit and explicit) from financial microblogs.

A taxonomy with seven classes and 32 subclasses is defined, and the corpus is manually annotated with class and sub-class labels. These labels are analogous to aspects.

A feature vector including Bag of Words (BoW), POS, numerical, and predicted sentiments is generated for each message. Machine learning algorithms like decision tree with XGboost, random forest, SVM, and CRF were trained and tested to assign class labels. Particle Swarm Optimization was used to find the best hyperparameters. Out of these methods, decision trees performed the best.

Wang *et al.* [59] have proposed a BERT-based classification method to detect implicit aspects. Embeddings for input text are generated using a 12-layer BERT model and then fed into a classification model to identify the implicit aspect. In addition to the simple BERT model, four different classification models CNN, BiLSTM, RCNN, and attention model were tested, and significant performance improvement is observed.

2) RULE-BASED METHODS

In approaches falling in this category, either association rules are extracted from the corpus, or rules, based on linguistic properties are identified and applied.

Liu *et al.* [60] have suggested a language pattern mining-based technique to extract aspects from a specific type of review that includes pros, cons, and detailed review.

A training data set is constructed by manually labelling the reviews. First, POS tagging is performed, which helps in generating general language patterns.

Actual aspect words are then replaced by word [feature] to find general patterns. In the case of implicit aspects, indicator words are replaced by word [feature]. Long segments are then reduced to multiple short segments using 3-grams, as long segments may generate spurious rules.

If a POS tag appears multiple times in a segment, they are assigned sequence numbers. Association rules are then generated from the training data, keeping 1% as the minimum support and without using minimum confidence. Rules in the following forms are generated:

$< \mathit{Noun}1 >, < \mathit{Noun}2 > \rightarrow [\mathit{feature}]$

$< \mathit{Verb} >$, easy to $\rightarrow [\mathit{feature}]$

$< \mathit{Noun}1 > \rightarrow [\mathit{feature}], < \mathit{Noun}2 >$

$< \mathit{Noun}1 >, [\mathit{feature}] \rightarrow < \mathit{Noun}2 >$

In the post-processing phase, rules that do not have [feature] are deleted, and rules are arranged in the proper sequence to generate language patterns.

$< \mathit{Noun}1 >, < \mathit{Noun}2 > \rightarrow [\mathit{feature}]$

easy to, $< \mathit{Verb} > \rightarrow [\mathit{feature}]$

These language patterns are used to identify explicit aspects. While preparing training data, mapping of indicator words is also performed with implicit aspects where indicator words are replaced by word [feature]. Then the mined rules may be used for the extraction of implicit aspects.

A method based on dependency tree and common-sense knowledge was introduced by Poria *et al.* [61] for extraction of implicit aspects. First, a sentence dependency tree is generated using a dependency parser, and then elements of the dependency structure are processed by the lemmatizer.

The corpus with indicated Implicit Aspect Clues (IAC), labelled with aspect category, is expanded to include synonyms and antonyms of IAC from WordNet. A set of conceptually related IACs is enlarged using semantics extracted from SenticNet. SenticNet3 is used as an opinion lexicon.

Two different sets of hand-crafted rules based on subject verbs are specified for identifying implicit aspects. For example, If a token s has a subject noun relationship with a word, and s has an adjective/ adverbial modifier present in SenticNet then s is identified as an aspect. Dependency parse structure is generated for each sentence, and then rules are applied on the parse tree to extract implicit aspects.

Hu and Liu have suggested a method [62] to extract implicit aspects similar to [60]. It is identical to [60] in pre-processing of training data but generates class sequential rules instead of language patterns as generated in [60].

An algorithm, Class Prefix-Span, based on the pattern growth method (Pei *et al.* 2004) was devised to mine class sequential rules from the training data. Rules in the following form are generated:

(1) $< \mathit{NN} > x < \mathit{NN} > x \rightarrow < \mathit{NN} > [\mathit{feature}] < \mathit{NN} > [\mathit{feature}]$

(2) $< \mathit{JJ} >$ easy to $< \mathit{VB} > x \rightarrow < \mathit{JJ} >$ easy to $< \mathit{VB} > [\mathit{feature}]$

To remove ambiguity as in rule 2 (whether $< \mathit{JJ} >$ is the POS tag for "easy" or other words before "easy"), rules are reassembled in the following form where each word has its POS tag in front of it (e.g., Rule 2):

(1) $< \mathit{JJ} > -1, -1$ easy, -1 to, $< \mathit{VB} > x \rightarrow < \mathit{JJ} > -1, -1$ easy, -1 to, $< \mathit{VB} > [\mathit{feature}]$

(2) $< \mathit{JJ} >$ easy, -1 to, $< \mathit{VB} > x \rightarrow < \mathit{JJ} >$ easy, -1 to, $< \mathit{VB} > [\mathit{feature}]$

(3) $< \mathit{JJ} > -1, -1$ easy, -1 to, $< \mathit{VB} > -1, -1x \rightarrow < \mathit{JJ} > -1, -1$ easy, -1 to, $< \mathit{VB} > -1, -1[\mathit{feature}]$

(4) $< \mathit{JJ} >$ easy, -1 to, $< \mathit{VB} > -1, -1x \rightarrow < \mathit{JJ} >$ easy, -1 to, $< \mathit{VB} > -1, -1[\mathit{feature}]$

-1 represents do not care situations when only word type is essential or the word does not have a POS tag. Rules are applied on new reviews, and aspects (explicit + implicit) are identified.

In [63] Poria and Gelbukh have proposed a method based on implicit aspect lexicon and hand-crafted rules. From a product review data-set, implicit aspect term and its category is manually extracted. Then synonyms of the terms are extracted from WordNet to expand the lexicon for implicit aspects. First, a dependency tree is generated for sentences, and then a set of hand-crafted dependency rules are applied to extract aspects and aspect terms for implicit aspects. Two separate classes of rules are defined for trees with subject noun relation and without subject noun relation.

For implicit sentences, an implicit aspect term is extracted using above mentioned rules, and the aspect category is assigned using the implicit aspect lexicon.

Lazhar has suggested a method incorporating association rule mining (ARM) and classification [64]. After preprocessing, opinion words and their targets (aspects) are identified from the extracted dependency relations and stored as tuples in the transaction database. Then association rules are mined from the transaction database.

Based on association rules, a classifier is built to predict the target aspect for a set of opinion words. For a given set of opinion words O, all the rules containing O as antecedent are extracted, and all the consequent aspects are considered candidate aspects. Candidate aspect f with highest average confidence (calculated as the average of confidence of rules containing f as consequent considering confidence 0 if f is not the consequent) is selected as target aspect. For an implicit sentence, opinion words are extracted, and using the classifier aspect is assigned.

In the method suggested by Sindhuja *et al.* [65], semantic aspect extraction is performed after preprocessing of data, and implicit aspects are extracted using rule-based classifiers. IF condition THEN conclusion, rules are extracted from the class-labelled input data set with attributes. Conditions are formed based on values of one or more attributes, and the consequent part consists of aspect category/class. The generated set of rules are applied to assign aspects to implicit sentences.

Schouten *et al.* [27] have also proposed a supervised approach called probabilistic activation algorithm to assign predefined aspect categories to sentences. The proposed approach can also be used for the identification of implicit aspects. They have used co-occurrence between lemmas/grammatical dependencies and annotated aspect categories from the training data to generate association rules. A dependency is a triplet having three parts: relation type, governor word, and dependent word. As the frequency of dependency triplets is usually very low, they have also used frequencies of two variants of dependency-1) relation type and governor word, 2) relation type and dependent word.

After removing stop words and low-frequency lemmas from data, an occurrence vector Y is prepared, having remaining lemmas/dependency forms (three forms-dependency relation and two variants) and their respective frequencies. Then co-occurrence frequency for each pair lemmas/dependency form and annotated category is stored in matrix X. A weight matrix is calculated using X and Y as in (25)-

$$W_{c,j} = \frac{X_{c,j}}{Y_j} \tag{25}$$

where c is the category and j is the lemma/dependency form. For an unseen sentence, maximum weights are calculated for all its lemmas and dependency forms. If the weight for a lemma/dependency form and category c is more than the threshold, category c is assigned to the sentence.

3) SEQUENCE TAGGING BASED METHODS

In machine learning, sequence tagging is algorithmically assigning a tag to each element of a sequence. In the methods falling in this category, Conditional Random Fields (CRF) is generally used to tag a sequence of words (generally sentences).

Rubtsova and Koshelnikov have given a method based on CRF to identify implicit aspects [66]. CRF is an undirected sequence model that selects a hidden sequence Y for sequence X, which maximizes $P(X|Y)$.

Sequential labels s-e/s-i/s-f for start of explicit aspect/implicit aspect/fact, c-e/c-i/c-f for continuation of explicit aspect/implicit aspect/fact and o for others are used for labelling.

Word, its POS tag, and lemma are the features used for labelling tasks. Two separate CRF are used for extraction of explicit aspect and implicit aspect/fact.

A FrameNet-based method was proposed by Chatterji and co-authors [67] for the identification of implicit aspects. FrameNet is a network of Frames, and a Frame stores descriptions of an event, its participants, sub-events, and relation between sub-event and event. They have developed Frames for each implicit aspect named AspectFrame, containing explicit aspects (Frame Elements), implicit aspect clues, relation with parent Frames, and a unique ID.

The CRF-based technique is used to tag implicit aspects to sentences using the following feature function.

$$\mathrm{FE}(S) = f(W, L, P, N, G, D, H, \mathrm{EA})$$

where W is the word, L is its lemma, P is its POS tag, N is its Named Entity tag, D is dependency tag, G is dependency tag when the word is used as governor, H is head of the dependency relation, and EA is explicit aspect. AspectFrameNet is used to correct mistakes in output generated by CRF tool.

Mamatha and co-authors [68] have suggested a CRF-based method for aspect category detection, which includes detection of both explicit and implicit aspects. After preprocessing, the notional words and dependency relations with two variations (governor word + dependency type and dependent word + dependency type) are used as features. Synonyms of category words are used to prepare a seed set for each category. After POS tagging, if noun/adjective matches with any of the seed words, rules of the form noun/adjective → aspect category are generated and stored. The word's suffix and prefix are considered in the absence of a match, and the aspect category is assigned.

A novel method depending on CRF was suggested by Cruz *et al.* [69] to extract implicit aspect indicators (IAI): words that indicate the presence of implicit aspects.

The task of extraction of IAI is cast as a sequence labelling task with labels IAI and O (others). The sequence labelling task is performed using CRF (Linear Chain). The features used for training are word, character n-grams, POS tag, context, class sequence.

Implicit aspects corresponding to the extracted indicators are assigned to the sentences.

C. HYBRID METHODS

Many authors have applied a combination of methods for the task of implicit aspect detection. Constituent methods may be applied in serial or in parallel. These approaches are termed hybrid, and discussion about methods combined in serial is followed by a discussion about methods combined in parallel. The proportion of surveyed hybrid methods belonging to different sub-categories is shown in figure 12.

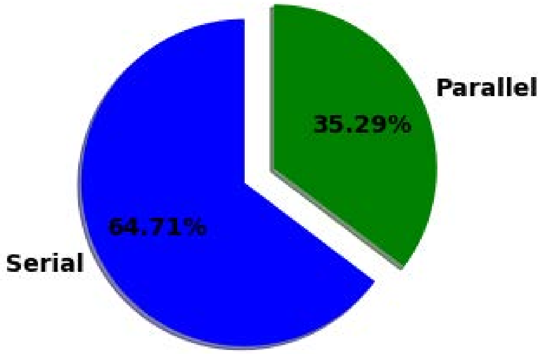

FIGURE 12. Proportion of hybrid implicit aspect detection methods belonging to different sub-categories.

Feng and co-authors [70] have proposed integrating sequential algorithm and deep convolution neural network to label sentiment and identify implicit aspects. Feature vectors are fed into a deep convolution neural network to generate scores for the sentiment tag of words. After that, the sequential algorithm is used to train for assigning tags for the whole sentence.

A quadruple (A_i, F_i, C_i, O_i) is generated for each clause using the output of the previous step, and "no" is used if any of the values are not present. A clause is assumed to have an implicit aspect if its associated tuple has value "no" for A_i. If a sentence has multiple clauses and an explicit aspect precedes the implicit aspect, it is termed a continuous aspect sentence otherwise an implicit sentence.

Co-occurrence matrix C is generated with explicit aspects as columns and words in a sentence as rows. Then matrix D is generated to store the probability of co-appearance of aspect word and words in a sentence, where D_{ij} represents probability of co-appearance of aspect i and word j. Also, a matrix F is generated to store the probability of co-appearance of aspect word and opinion word, where F_{ij} represents probability of co-appearance of aspect i and opinion word j.

For a given implicit sentence with t words, a score for each candidate aspect is determined as in (26):

$$A_i = \lambda_1 \sum_{j=1}^{t} D_{ij} + \lambda_2 F_{ij} \tag{26}$$

where $\lambda_1 + \lambda_2 = 1$ and aspect with the maximal score is assigned as an implicit aspect to the given sentence.

If the given sentence is a continuous aspect sentence with the opinion word O_j and preceding explicit aspect A_h, then g_{hj} is retrieved. If g_{hj} is the probability of co-appearance of aspect h and opinion word j and $g_{hj} > \beta$ (threshold) then A_h is assigned as an implicit aspect; otherwise, the preceding step identifies implicit aspect.

Panchendrarajan *et al.* [71] have used a method to extract multiple implicit aspects for a sentence. They have extended the work of [15] for the task of identification of implicit aspects.

Training data is prepared to store a list of sentences with annotated aspects for each opinion word. For every opinion word in a sentence, a listing of candidate aspects is obtained from the training data. A score for every candidate aspect is determined, and the aspect with the maximal score is assigned as an implicit aspect for that opinion word if it exceeds the given threshold. They have modified the equation (given in [15]) to calculate score for aspect A_i to include the distance from the opinion word as shown in (27):

$$\text{Score}\ (A_i) = \frac{1}{n} \sum \frac{c_{ij}}{f_j d_j} \tag{27}$$

where n is the count of words in the sentence, count of co-occurrence of aspect A_i and j^{th} word in the sentence is c_{ij}, f_j is the frequency of j^{th} word and distance of j^{th} word in the sentence with opinion word is d_j.

They have developed a hierarchy of aspects for restaurant reviews. They have also stated three rules to validate the predicted implicit aspects. First, the target for opinion word is retrieved utilizing grammar rules and then following rules are applied.

Rule 1: If the target is the parent entity in aspect hierarchy, then the prediction is correct. Otherwise, the extracted target is used to extract additional targets, and rule 2 is applied.

Rule 2: If the further target is the parent entity in aspect hierarchy, then the prediction is correct; otherwise, extract further target for given opinion word, and then rule 3 is applied.

Rule 3: If the extracted opinion word is a sibling in the hierarchy, then the prediction is correct; otherwise predicted aspect is discarded.

In [72] (Xu *et al.*) a topic mining model with some prior knowledge is used to extract features based on explicit aspects and sentences, and then SVM classifiers are established for each aspect to classify non-explicit sentences. A topic model LDA is extended by incorporating prior knowledge in the form of must links, cannot links, syntactic, and PMI-based prior knowledge derived from explicit aspects.

Must links specify pairs of words that must be assigned to the very topic and cannot links specify pairs of words that cannot be assigned to the very topic. The knowledge induced from must links and cannot links, is incorporated in topic updating process of LDA.

The association between a word and aspect is taken into consideration for the determination of topic word distribution. The association is measured based on the dependency relation and PMI score of the pair. SVM classifiers for distinct aspects are then trained on features extracted through a topic model, and used to identify aspects for non-explicit sentences.

Hajar and Mohammed proposed a hybrid method combining corpus and WordNet (WN) dictionary to extract adjectives related to implicit aspects [73]. The mapping of adjectives to implicit aspects may be used for the extraction of implicit aspects.

A list is prepared to have all adjectives present in the corpus. For adjective a_i, frequencies of its related words (synonyms, antonyms and derived words) in WN are represented in a vector, like $V_{r_i} = (fw_1, fw_2, \ldots, fw_n)$, where fw_i is the frequency of i^{th} related word.

From training data, a vector is constructed to store the frequency of adjectives for aspect A_j.

$VA_j = \left(ft_1, ft_2, \ldots, ft_{NA_j}\right)$, where ft_i is the frequency of i^{th} adjective and NA_j is the number of adjectives for aspect A_j.

A matrix Ma ($M \times N$) is constructed from VA_j vectors for all aspects, where M is the count of adjectives and N is the count of aspects. Global frequency vector is calculated for every adjective to bind the impact of WN frequency with the corpus frequency as in (28):

$$\text{Vrg}_i = ft_i \times Vr_i \tag{28}$$

A matrix Mr ($L \times N$) is constructed from Vrg_i vectors for all aspects, where L is the count of WN-related words and N is the count of aspects. Finally, a matrix Mt is prepared by combining Ma and Mr.

Mt is used to train an NB classifier, assigning each pair (adjective, aspect) a probability. A threshold is experimentally determined to generate a set of adjectives that best relate to an aspect. In an implicit sentence, the adjectives present in the sentence may be used to assign corresponding implicit aspects.

Jiang *et al.* [74] suggested an association rule mining method that incorporates improved collocation extraction and topic modeling to extract implicit aspects. First, an improved collocation extraction algorithm is used to generate a basic rule set (BR) of the form aspect indicator → aspect. Then semi-supervised LDA is used to generate new rules, and BR is extended to include these new rules, and the extended rule set is termed as MBR (Model rules + Basic rules).

The product aspects are extracted using a frequent itemset algorithm and few manual operations, and synonyms are also grouped. Sentences containing aspect words are considered explicit, and candidate aspect indicators are obtained using POS tags and a minimum frequency. Verbs, adjectives, adverbs, nouns, pronouns and quantifiers are considered indicators; combining two words with the tag mentioned above is considered an indicator. The weight of the indicator is determined depending on the degree of co-occurrence.

Candidate indicators are extracted for each aspect (feature) using some threshold value, and redundant indicators are then pruned. Then association rules (Basic rules) are generated of the form aspect indicator → aspect.

These rules may not include some brand words and abbreviations of aspects; also, they may suffer from an imbalance of data. Hence constrained-LDA (Zhai *et al.*) is adopted for the given number of topics and topic words to generate additional rules (Model rules). MBR (Model rules + Basic rules) extracts implicit aspects for sentences with only indicator words.

In [75], Chen and co-authors have combined context information and a topic model to identify implicit aspects.

Two probability distributions are generated in phase one based on the topic model (LDA) and improved co-occurrence matrix. In phase two, scores for candidate aspects are calculated considering context weight and cosine similarity.

A set of candidate implicit aspects A is prepared based on the opinion word's context probability. n_i is included in A, if $P_{\text{opn}}(n_i) > p$, where $P_{\text{opn}}(n_i)$ is the context probability of opinion word opn and p is the threshold. A formula is given in (29) to calculate a score for each candidate aspect:

$$\text{score}(n_i) = \text{weight}^{\text{opn}} \times \psi^{\text{opn}} + \left(1 - \text{weight}^{\text{opn}}\right) \times \Omega^{\text{opn}} \tag{29}$$

where ψ^{opn} is the opinion word's context distribution, Ω^{opn} is the topic probability distribution for the opinion word, and $\text{weight}^{\text{opn}}$ is the weight of the context. The candidate aspect with the maximal score is assigned as the implicit aspect.

Dosoula *et al.* [76] have given a method to detect multiple implicit aspects in a given sentence. They have extended the work of [15] by including a classifier that determines whether a given sentence contains multiple implicit aspects or not.

Calculating a score for every candidate aspect, based on the co-occurrence frequency of the aspect and other words of the sentence is suggested in [15]. The aspect with the maximal score is selected if its score is greater than a trained threshold.

If the classifier indicates that the sentence contains multiple aspects, all the candidate aspects with a score greater than the threshold are assigned. Otherwise, the approach given in [15] is followed.

A score is calculated for each sentence s by using (30):

$$\begin{aligned}\text{Score}_{K_s} = {} & \beta_0 + \beta_1\#\text{NN}_s + \beta_2\#\text{JJ}_s + \beta_3\#\text{Comma}_s \\ & + \beta_4\#\text{And}_s\end{aligned} \tag{30}$$

where, $\#\text{NN}_s$ is the count of nouns, $\#\text{JJ}_s$ is the count of adjectives, $\#\text{Comma}_s$ is the count of commas, and $\#And_s$ is the count of 'and' in sentence s. β_i's are generated from training data using logistic regression and maximum likelihood.

A novel approach utilizing non-negative matrix factorization was suggested by Xu *et al.* [77] to identify implicit aspects. Aspects are clustered by combining the co-occurrence of aspects and opinion words and intra-relation information of aspects and opinion words. Then context information is used to predict implicit aspects for a sentence. Explicit aspects and opinion words are extracted using the double propagation method as suggested by Qiu *et al.* Set of opinion words, and aspects are represented as $O = \{o_1, o_2, \ldots, o_m\}$ and $A = \{a_1, a_2, \ldots, a_n\}$ respectively. A weight matrix X $(m \times n)$ is constructed to store the co-occurrence of aspects and opinion words. As the sparsity of matrix X may affect the identification of implicit aspects, they have clustered aspects into categories and opinion words into clusters. They have clustered rows of X (aspects) and columns of X (opinion words) by decomposing matrix X into two non-negative matrices as in (31):

$$\min J_1 = \|X - UV^T\|_F^2 \; s.t. V \in R^{n\times k}, U \in R_+^{m\times k} \tag{31}$$

U and V are cluster indicator matrices for opinion words and aspects, respectively, and k is the number of clusters.

After removing irrelevant words and stop words, a lexicon set $L = \{w_1, w_2, \ldots, w_L\}$ is prepared from the training data. Every aspect category k is represented as a vector $w^k = \{w_1^k, w_2^k, \ldots, w_L^k\}$ where w_i^k is calculated as in (32):

$$w_i^k = f_i^k \times \log \frac{|c_k|}{n_i} \tag{32}$$

where f_i^k is the frequency of w_i in category k, $|c_k|$ is the count of aspects in category k and n_i is the count of categories that include w_i.

A feature-centroid classifier is constructed to identify implicit aspects for a sentence S. First sentence S is transformed into a word vector $S = \{S_1, S_2, \ldots, S_L\}$ and its cosine similarity is determined with each aspect category vectors. Aspect category with the highest similarity is selected if the similarity is more than the given threshold, and a representative word of that category is assigned as an implicit aspect to the sentence S.

Liu and co-authors [78] have suggested a bipartile graph model for the extraction of implicit aspects. Aspect-opinion pairs are extracted using CRF from explicit sentences, and then implicit aspects are identified based on a random walk on bipartile graph.

After preprocessing, aspect-opinion pairs are extracted using CRF by considering word, POS tag, position, and interdependent syntactic relation as features. Then aspect-opinion pairs are clustered for every opinion word, eg. $\{\langle f_1, o\rangle, \langle f_2, o\rangle, \ldots, \langle f_n, o\rangle\}$ for opinion word O.

A bipartile graph $G = (V_1, V_2, E)$ is then constructed from gathered information (aspects and opinion words) where V_1 and V_2 are disjoint sets of vertices, and E is a subset of $V_1 \times V_2$. Matrix W stores weights of edges where w_{ij} is the weight of edge (i, j). Given F is a set of aspects, O is a set of opinion words, and F_S is a seed set of features. The proposed algorithm calculates the probability of the candidate implicit feature from $F - F_S$ assigned to the opinion word b_j. Assume that $X(t)$ represents a state matrix with $X(0)$ as the initial state of the candidate aspect set. X is updated iteratively as given in (33):

$$X(t) = \lambda HX(t-1) + (1-\lambda)X(0), \lambda \in (0, 1) \qquad (33)$$

where $H = D^{-1/2}RD^{-1/2}$, $R = WW^T$, and D is a diagonal matrix with d_{ii} is the sum of elements of R. The probability that aspect f_i belongs to opinion word b_j is calculated as in (34):

$$P\left(b_j|f_i\right) = \frac{X_{i,j}}{X_{i,0} + X_{i,1}} \qquad (34)$$

and aspect f_i with the maximal probability is assigned as an implicit aspect wherever the opinion word b_j appears.

In the solution suggested by Khalid and co-authors [79], LDA generates raw topics from the given set of reviews. Generated topics are sets of words with high contextual correlation.

In the second step, POS tagging of words in the topic sets is performed. Nouns and noun phrases are treated as candidate aspects and stored in set cAspectTerms$_{\mathrm{i}}$. Words in cAspectTerms$_{\mathrm{i}}$ represent explicit aspect terms. Remaining words with POS tags verb/adverb/adjective are stored in the set cReasonTerms$_{\mathrm{i}}$ and represent implicit aspect indicators.

In the next step, the paradigmatic association between words of cAspectTerms$_{\mathrm{i}}$ is calculated based on contextual similarity (calculated as in (35)), and words with low paradigmatic association with all other words in cAspectTerms$_{\mathrm{i}}$ are discarded. The resultant set is termed as Aspect$_{\mathrm{i}}$-containing words representing aspect i.

$$\text{ContextSim}\left(w_1, w_2\right) = \sim \left(\text{Context}\left(w_1\right), \text{Context}\left(w_2\right)\right) \qquad (35)$$

Similarity of general context is calculated as in (36):

$$J\left(w_1, w_2\right) = |w_1 \cap w_2||w_1 \cup w_2| \qquad (36)$$

It represents overlap of general context.

Terms present in cReasonTerms$_{\mathrm{i}}$ act as indicator words for aspect I represented by words in Aspect$_{\mathrm{i}}$. Hence, the term from cReasonTerms$_{\mathrm{i}}$ in an implicit sentence indicates that the implicit aspect is aspect I and may be represented by representative word from Aspect$_{\mathrm{i}}$.

Maylawati *et al.* [80] have proposed a hybrid method for implicit aspect detection, which incorporates feature selection, clustering, and association rule mining in serial.

After preprocessing of sentences in the input data set, TF-IDF value is used for feature extraction. Particle Swarm Optimization (PSO) is used for feature selection. The explicit words generated as output of the feature selection process are used as input for clustering. K-means clustering is used to generate seven explicit clusters of sentences. Each cluster is assigned a label (aspect category), and each sentence in the cluster is assigned the label. Results of explicit sentence clustering are used for implicit aspect extraction.

FIN algorithm is used to mine association rules which are based on nodeset data structure. From frequent 2−itemsets, POC tree (pre-order coding) is used to generate nodesets. Value of support is calculated for each item, then the support values for words in each category are sorted, and words with support less than the minimum support are discarded, and the rest of the words are inserted into POC tree. The output rules are used to extract implicit aspects.

Few authors have also applied multiple methods in parallel; methods suggested by them are discussed here.

A hybrid method to mine association rules (indicator → aspect) for implicit aspect detection is given by Wang *et al.* in [81]. Their idea was to mine an extensive set of association rules using multiple algorithms. Segmentation and POS tagging were performed on given reviews about a specific product, and aspects are extracted using the frequent itemset method, and synonymous aspects are clustered. From explicit sentences, candidate indicators are extracted using POS tags and the least occurrence. The weight of an indicator is calculated based on the degree of co-occurrence between indicator and aspect. Five different collocation extraction methods- frequency, PMI, PMI*frequency, t-test, and χ^2 test are used to measure the degree of co-occurrence. A pruning method based on a threshold (also used to generate rules) is used to prune conflicting indicators that occur with multiple aspects. After pruning, a set of basic rules is generated using a threshold.

The basic ruleset is expanded to include reasonable rules from non-indicators and lower-weight indicators using three additional methods- substring hypothesis, dependency structure, and constrained topic model. If a word's substring is part of the basic ruleset, it constitutes a reasonable rule. If a word is an adjective and is in subject relation with an aspect, they form a reasonable rule. LDA is expanded to include some prior knowledge in the constrained topic model and then used to extract additional rules. This expanded ruleset is used for the detection of implicit aspects.

Sun at el. [82] have proposed a method that is very similar to [75] for the identification of implicit aspects. They have presented a joint topic-opinion model that considers both the topics and opinion word's context.

After POS tagging, nouns and adjectives are extracted to form a new phrase. Vocabulary or all distinct words are denoted by V while V_{opn} and V_{con} denote the number of opinion words and noun words, respectively. For n^{th} word $w_{d,n}$ in document d, the value of an an indicator variable called POS label $l_{d,n}$ with possible values opn (for opinion word), and con (for context word) is determined.

A word in comment d is generated by first generating a topic z_d from topic distribution and then POS label $l_{d,n}$ is drawn from Bernoulli distribution over POS labels (opn or con). If $l_{d,n}$ is opn then $w_{d,n}$ is generated from opinion distribution for topics $\left(\Phi^{Z}\right)$ specific to topic z_d. In case of $l_{d,n}$ is con; first, the opinion label is generated from opinion distributions for topic $\left(\Phi^{Z}\right)$ and then $w_{d,n}$ is drawn from context distribution for both opinions and topics $\left(\Phi^{Z,opn}\right)$.

Opinion word from an implicit sentence is identified, and for general opinion words like good, the score for each candidate aspect is determined as in (37):

$$\text{Score}_i\,(\text{opn}) = a \times \Psi^{\text{opn}} + b \times \Omega^{\text{opn}} \tag{37}$$

where Ω^{opn} is context distribution for opinions, $a+b = 1$ and Ψ^{opn} is calculated as in (38):

$$\left(\Psi^{\text{opn}}\right) = \frac{\phi^{Z,\text{opn}} \times V}{T \times V_{\text{adj}}} \tag{38}$$

where T is the number of topics.

For special opinion words like heavy, score for each candidate aspect is determined as in (39):

$$\text{Score}_i\,(\text{opn}) = \Psi^{\text{opn}} \tag{39}$$

The candidate aspect with the maximal score is assigned as the implicit aspect.

A WordNet (WN) based method is proposed by Hajar and Benkhalifa [83] to identify implicit aspects. Adjectives and verbs are considered as indicators for implicit aspects, and a hybrid model incorporating WN semantic relations and Term-Weighing is used to enhance training data. Three different classifiers Multinomial NB, SVM, and Random Forest, are trained and tested.

After preprocessing, adjectives and verbs (called terms) are extracted, and a list of terms is prepared $\{Ta_1, Ta_2, \ldots, Ta_{na}, Tv_{na+1}, Tv_{na+2}, \ldots, Tv_n\}$. Where $+-$ Ta_i denotes adjectives, Tv_i denotes verbs, and n is the total number of terms. V_{T_i} vectors are constructed for terms T_i using WN semantic relations. A document term vector V_{dt_j} is generated to represent each document j. Then document term frequency vector V_{dtf_j} is prepared for each document j.

Instead of using Inverse Document Frequency (IDF), they have used Inverse Class Frequency (ICF) which effectively deals with imbalanced data. ICF for each T_i is calculated as in (40):

$$\text{ICF}\,(T_i) = \log\left(\frac{N_c}{\sum_i^{N_c} \alpha}\right) \tag{40}$$

where, N_c is the count of class, $\alpha = 0$ if T_i does not occur in class C_k and $\alpha = 1$ otherwise. Finally, matrix $M_{TF-ICF}\,(N_c, N)$ is generated as in (41) to store the association of classes and terms.

$$M_{TF-ICF} = M_{TF} \times M_{ICF} \tag{41}$$

TABLE 2. Contingency matrix.

	Predicted +ve	Predicted Negative
Actual +ve	A++	A+-
Actual -ve	A-+	A–

where M_{TF} is the diagonal matrix of ICF. W-training data splits are used to improve/enhance training data. Then M_{TF-ICF} is calculated from the enhanced training data, reflecting the association strength of terms and classes. M_{TF-ICF} may be used to identify which term (indicator) represents which class (implicit aspect).

Tubishat and Idris [84] have suggested Whale Optimization Algorithm (WOA) to extract explicit aspects. A hybrid method incorporating corpus co-occurrence, web-based similarity, and dictionary is proposed to extract implicit aspects.

After preprocessing and generation of dependency relations, WOA is applied on training data to select rules. Selected rules are employed on test data to retrieve explicit aspects. From these aspects, infrequent aspects are discarded. Aspects from product specifications are also included in to the set of aspects. Synonyms and meronyms of aspects are found from WordNet (WN), and included in a set of aspects. The similarity between discarded aspects and synonyms of domain entities is found using Normalized Google Distance (NGD), and filtered aspects using a threshold are included in a set of aspects.

Matrix M is created to store the co-occurrence frequency of extracted explicit aspects and corresponding opinion words. Co-occurrence of opinion words with other notional words in a given sentence is added to M. Then the co-occurrence of notional words is added to M.

Synonyms and antonyms of each opinion word from M are extracted using WN. Glosses of words from an opinion lexicon are searched, and nouns are extracted to prepare a dictionary D which stores nouns for opinion words. For opinion word from an implicit sentence (OIA), OIA and its synonym and antonym are searched from M, and co-occurred aspects are considered candidate implicit aspect (CIA). Also, D is searched, and co-occurred nouns are extracted as CIA.

For each CIA, its NGD is calculated with all notional words in the implicit sentence, and CIA with the smallest NGD value is assigned as implicit aspect. For sentences without any notional word, notional words from the same OIA sentence are used to calculate NGD.

In [85] Rana and co-authors have suggested a multilevel method based on co-occurrence and similarity calculations for implicit aspect detection. First, rules are devised to extract clues for implicit aspects, and then, using a multilevel approach, aspects are assigned based on extracted clues.

Rules, based on sequential patterns are complemented by some manually crafted rules for identifying clues. If the clue is an entity and opinion is a concept, then the explicit aspect

TABLE 3. Performance comparison.

Author	Dataset Source	Domain	Precision%	Recall%	F-measure%
Unsupervised Methods					
1. Zhang et al. [13]	Reviews from taobao.com	Mobile Phone	81.34	79.51	80.41
		Clothes	80.17	77.38	78.75
2. Hai et al. [14]	Reviews from 3 Chinese forums	Cell Phone	76.29	72.71	74.46
3. Schouten et al. [15]	Reviews from Citysearch New York	Restaurant	-	-	63.3
4. Sun et al. [16]	Reviews from 360buy.com	Computer	66	58	61.74
		Mobile Phone	71	62	66.20
5. Liu et al. [17]	Reviews from 360buy.com	Phone	79	70	74.23
6. Kama et al. [22]	Reviews from donanimhaber.com	News	90.36	80.89	85.36
7. Omurca et al. [24]	Reviews from otelpuan.com	Product	100	61	77
8. Karagoz et al. [25]	Reviews from donanimhaber.com	Product	90.36	80.89	85.36
9. Schouten et al. [28]	Reviews from Citysearch New York	Restaurant	-	-	79.4
		Product	-	-	70.4
10. Song et al. [29]	Comments from e-commerce websites	Cosmetic Products	91	74	82
11. Prasojo [31]	Al Jazeera and CNN websites	News	72.42	64.62	68.30
12. Wu et al. [39]	Reviews from jd.com	Computer	81	76	78
13. Hai et al. [40]	Reviews from product.mobile.163.com	Cell Phone	72.99	69.93	71.43
	Reviews from lvping.com	Hotel	58.09	62.20	60.08
14. Santu et al. [41]	Reviews from cnet.com and amazon.com	Router	74.87	60.4	66.86
15. Yu et al. [44]	Reviews from cnet.com, viewpoints.com,reevoo.com and gsmarena.com	Product	-	-	86
Supervised Methods					
1. Zeng et al. [51]	Reviews from amazon.cn	Cell Phone	82.07	68.48	74.66
		Digital Camera	85.59	72.93	78.76
2. Hajar et al. [55]	SemEval 2014 ABSA Dataset	Mobile Phone	-	-	82.7
3. Lazhar[64]	TripAdviser Dataset	Hotel	-	-	82.1
Hybrid Methods					
1. Feng et al. [70]	Reviews from amazon.com and jd.com	Mobile Phone	87.58	77.69	82.33
2. Panchendrarajan et al. [71]	Reviews from yelp.com	Restaurant	88.6	69.4	77.9
3. Xu et al. [72]	Reviews from 360buy.com	Cell Phone	87.42	70.05	77.78
4. Hajar et al. [73]	SemEval 2015 ABSA Dataset	Restaurant	97.6	94.3	95.92
		Digital Camera	96	92	93.96
5. Jiang et al. [74]	Reviews from amazon.com	Mobile Phone	82.57	73.65	77.86
6. Chen et al. [75]	Reviews from amazon.com	Canon G3	72	68	69.94
7. Dosoula et al. [76]	Reviews from Citysearch New York	Restaurant	-	-	69.3
8. Xu et al. [77]	SemEval 2015 ABSA Dataset	Restaurant	76.61	98.87	86.33
9. Liu et al. [78]	Chinese e-commerce websites	Phone	90.33	85.62	87.91
		Computer	86.49	80.42	83.34
10. Wang et al. [81]	Reviews from 360buy.com	Cell Phone	86.10	67.25	75.51
11. Sun et al. [82]	Reviews from 360buy.com	Computer	69	64	66.40
		Mobile Phone	82	68	74.35
12. Hajar et al. [83]	Twitter	Crime	-	-	90
13. Rana et al. [85]	Reviews from cnet.com and amazon.com	Products	77	79	78
14. Eldin et al. [86]	Reviews from souq.com and amazon.com	Digital Camera	-	82.2	-

with the highest co-occurrence with the opinion word is assigned as the implicit aspect. The explicit aspect as a part of the clue's opinion is looked at; if co-occurrence is found, it is then assigned as an explicit aspect for the rest of the opinion. The entity itself is assigned as a clue if no clue is present in the sentence. If the clue is an entity and there is no co-occurrence between the clue's and explicit aspect's opinion, then NGD between each explicit aspect and opinion is calculated, and the aspect with the smallest NGD is assigned as implicit aspect.

Eldin *et al.* [86] have proposed a metaheuristic optimization approach incorporating multiple similarity measures to identify implicit features. They have categorized implicit features into context-based features and features with indicators. For implicit features with indicators, all the explicit features frequently co-occurred with the indicator are considered candidate features. For context-based features, all explicit features are considered candidate features. They have proposed a cuckoo search algorithm for the selection of optimal features. The proposed algorithm incorporates a fitness

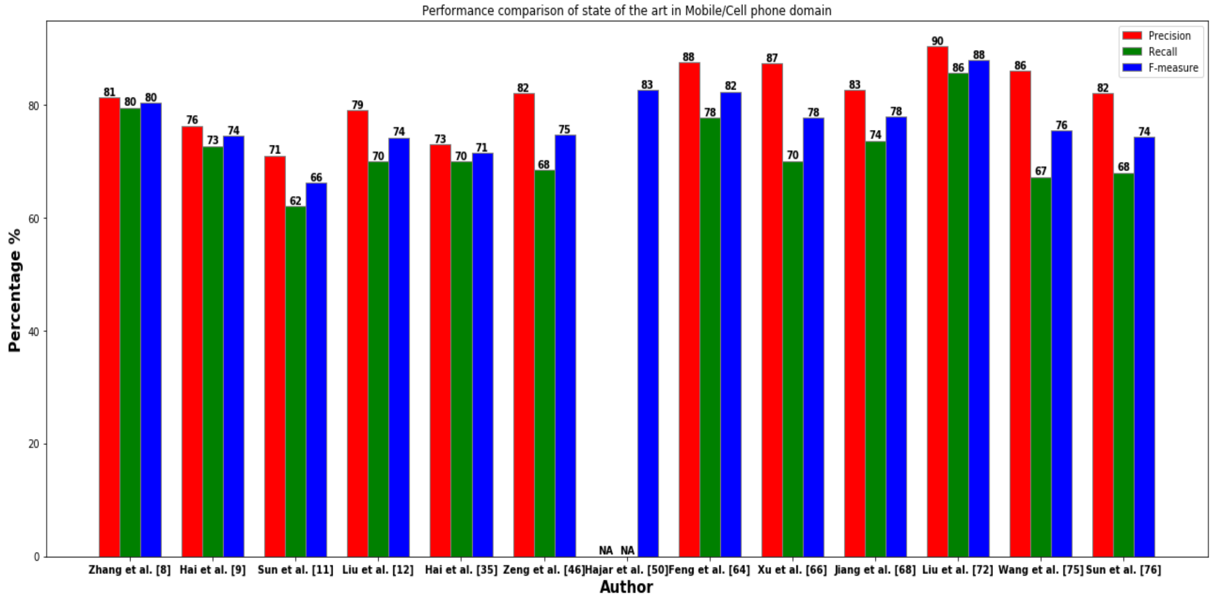

FIGURE 13. Performance comparison of methods in mobile/cell phone domain.

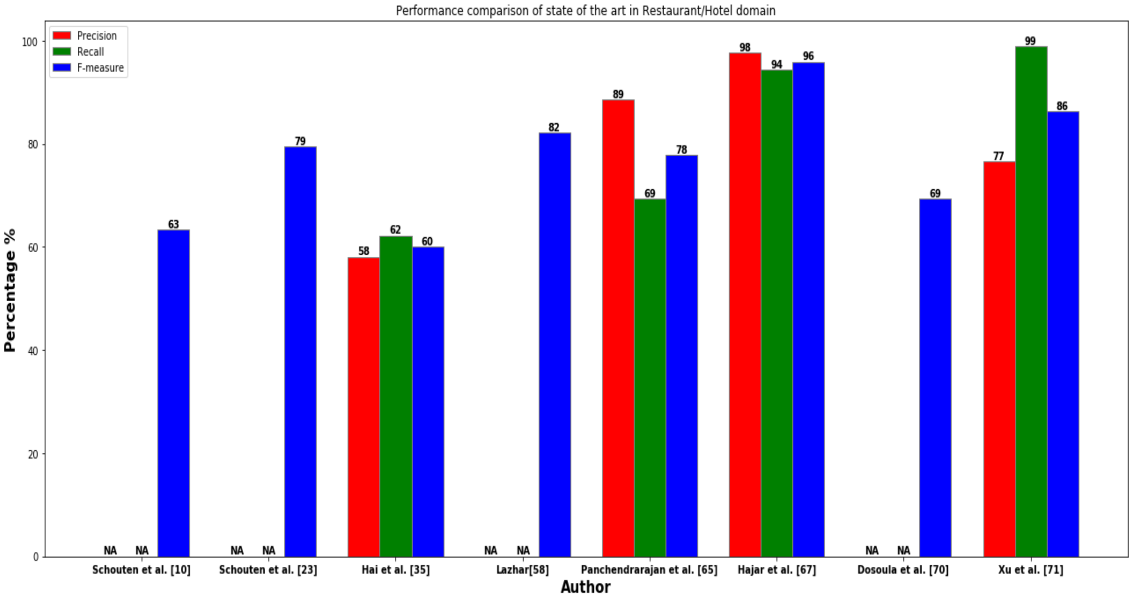

FIGURE 14. Performance comparison of methods in restaurant/hotel domain.

function based on Jaccard similarity and Normalized Google Distance (NGD) to rank candidate features. The candidate feature with the highest average similarity (among various iterations) in cuckoo search is assigned as an implicit feature.

In [87], authors have suggested a deep learning (LSTM) based method that incorporates information from WordNet and spaCy. An LSTM model is trained after preprocessing of the data. Also, the similarity of words of the sentences with all aspect categories is calculated using WordNet and spaCy. A score for each aspect category is calculated for a given sentence based on the trained LSTM model and similarity from WordNet and spaCy. Based on training, weights are assigned to each method (LSTM, WordNet, and spaCy), and aspect category is assigned to the given sentence using a weighted sum of scores calculated from these methods.

Cai and co-authors [88] have proposed extracting a quadruple including aspect term, aspect category, opinion term, and sentiment polarity, using four different methods. Extraction of all the quadruple also includes extraction of implicit aspects.

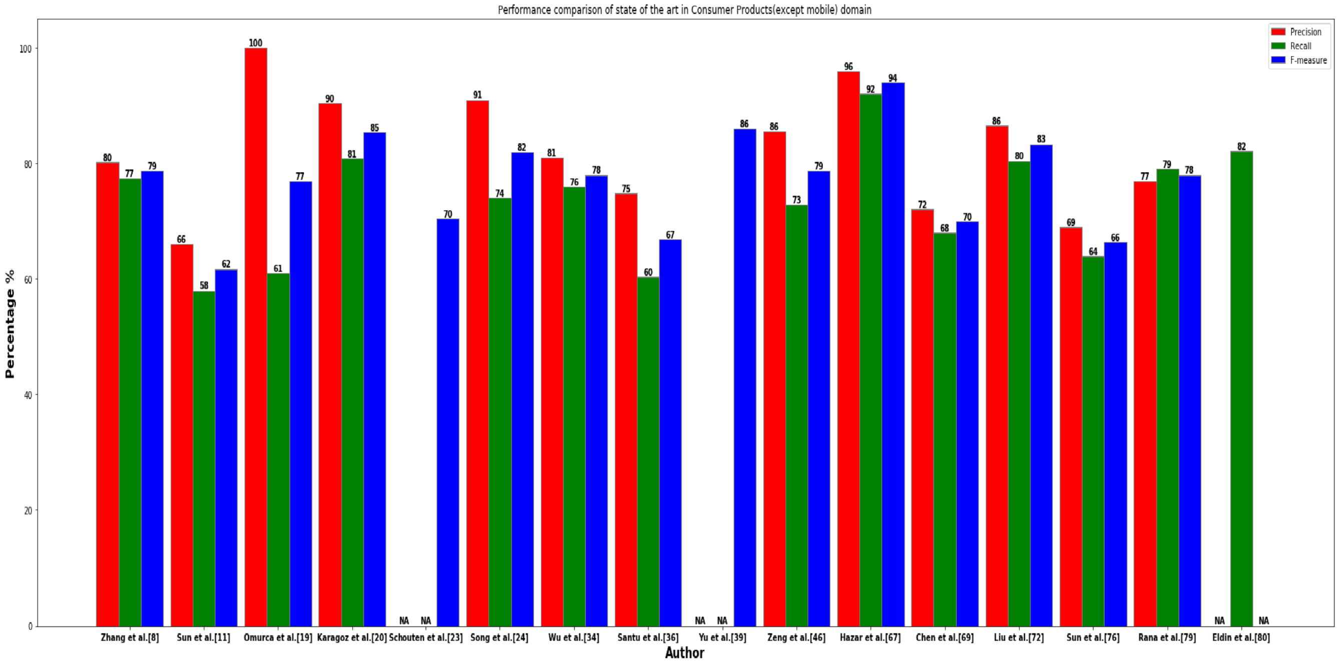

FIGURE 15. Performance comparison of methods in consumer products (except mobile phone) domain.

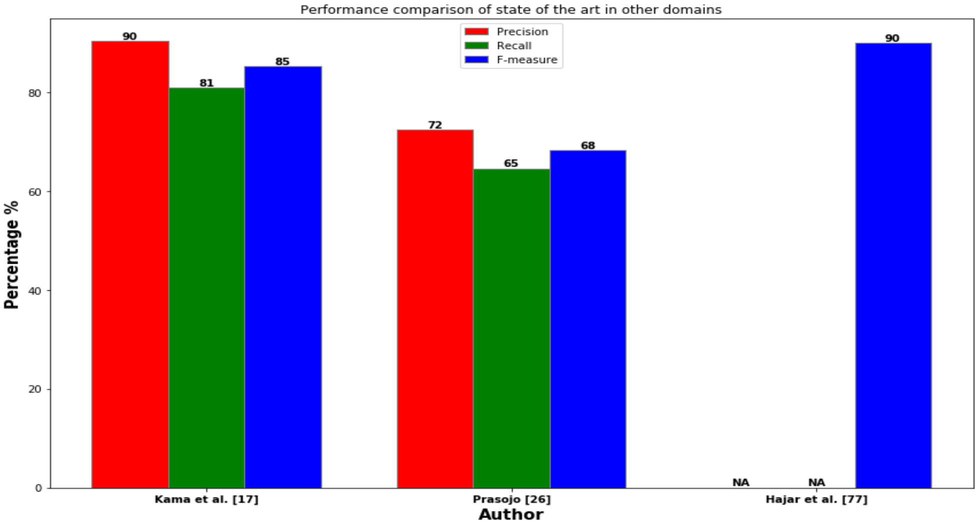

FIGURE 16. Performance comparison of methods in other domains.

In the first method, they have extended the Double Propagation method suggested by Qui *et al.* to extract quadruples. First aspect term-opinion term-sentiment polarity triplets are extracted using Double Propagation, then using co-occurrence from training dataset aspect category is assigned to each triplet. An approach suggested by Xu *et al.* was used to extract aspect term-opinion term-sentiment polarity triplets, and then BERT based method is used to assign aspect category to each triplet. A method suggested by Wan *et al.* is adopted by using the input transformation strategy to extract quadruples followed by removing invalid aspect-opinion pairs.

The authors have proposed a two-step method to extract quadruples. In the first step, they have extracted aspect-opinion pair followed by extraction of category-sentiment pair in the second step.

Tian and White [89] have suggested utilizing the verbs/adjectives rendering information about the implicit aspects to identify them. Then semantic similarity and hierarchical agglomerative clustering are used to merge similar aspects. Finally, synonyms/antonyms from WordNet are used to map the implicit aspect cluster to one explicit aspect.

Wu *et al.* [90] have proposed a method to extract (target, aspect, sentiment) triplets, which also includes implicit aspects. The initial embedding vector for the aspect sentence pair is generated using BERT and bidirectional LSTM is used to generate the representation for aspect and sentence. The dependency between the sentence and aspect is captured using a graph convolutional network including an attention mechanism.

IV. PERFORMANCE COMPARISON

Precision, Recall, and F-measure are generally used as quantitative measures to assess the performance of aspect detection techniques.

Precision is the ratio of actual positive cases which are predicted positive, and total cases which are predicted positive.

Recall is the ratio of actual positive cases which are predicted positive, and total actual positive cases [91].

Precision= $A{+}{+} \,/\, (A{+}{+} + A{-}{+})$

Recall= $A{+}{+} \,/\, (A{+}{+} + A{+}{-})$

Mostly we have a tradeoff between recall and precision. If we try to increase the precision of our method, the recall may decrease and vice versa. F-measure is a balanced measure and is measured as the harmonic mean of precision and recall as in (42):

$$\text{F-measure} = \frac{2 \times \text{precision} \times \text{recall}}{\text{precision} + \text{recall}} \tag{42}$$

In the literature reviewed, some authors have not performed a quantitative analysis of the performance of methods suggested by them. Many of them have suggested solutions to extract both explicit and implicit aspects together and have not performed quantitative analysis of performance separately to detect implicit aspects. The performance of methods for implicit aspect detection as suggested by their authors is included in table 3.

For comparison purposes, we have divided the literature included in table 3 into four domains- 1. Mobile/Cell Phone 2. Restaurant/Hotel 3. Consumer Products(except mobile phones) 4. Other domains

Figures 13 through 16 pictorially depict the comparison of the performance of the methods in each domain as mentioned above. Values for precision, recall or F-measure, not specified by the authors are represented as NA.

As the analysis is performed on different data sets, performance of methods is not directly comparable. For implicit aspect detection, some standard datasets must be developed to reduce the effect of bias (in the dataset) on the results. Also, the datasets should be prepared by using text from multiple different sources, including different sections of society and demographic locations. Although dataset for implicit aspect detection is rare, but following datasets for aspect level sentiment analysis may be modified and used (table 4).

TABLE 4. Datasets for aspect-level sentiment analysis.

Dataset	Author	Link
corpus for implicit aspect extraction and implicit aspect indicator extraction	Ivan Omar Cruz-Garcia, Alexander Gelbukh, Grigori Sidorov	https://www.gelbukh.com/resources/implicit-aspect-extraction-corpus/
SemEval 2014- Task 4- ABSA	Not Available	https://alt.qcri.org/semeval2014/task4/index.php?id=data-and-tools
Restaurant-ACOS dataset	Hongjie Cai et al.	https://github.com/NUSTM/ACOS/tree/main/data/Restaurant-ACOS
Laptop-ACOS dataset	Hongjie Cai et al.	https://github.com/NUSTM/ACOS/tree/main/data/Laptop-ACOS
MAMS (Multi Aspect Multi-Sentiment)	Jiang et al.	https://github.com/siat-nlp/MAMS-for-ABSA/tree/master/data

V. ISSUES AND FUTURE SCOPE

It is more than a decade since the term implicit aspect was first coined. Since then, quite a few authors have attempted to perform the task of implicit aspect detection, as we had discussed in section 3. Nevertheless, this problem is not rigorously researched, and many issues are yet to be resolved.

The language used on social networking platforms is a big issue as it generally does not follow grammar and has abbreviations, slangs, and incorrect spellings, which is a big hurdle in applying concepts related to syntax and linguistics. Emojis are frequently used to represent opinion/sentiment and need to be handled separately. Sarcastic or fake reviews are challenging to identify and adversely affect the evaluation of performance.

Existing approaches are having issues like imbalanced data for training, the high computing time requirement for manual tuning of different parameters, scalability, and performance on small-scale corpora.

Implicit aspect detection is difficult for sentences with little context or without opinion words. Most of the suggested solutions suffer from performance issues if applied to some other domain or language. Also, no standard dataset is available, hence comparison of performance is not possible, as mentioned in section 4.

Significant performance improvement is observed when some existing knowledgebase is incorporated. Hence development and enhancement of knowledgebases for linguistics like lexicons, semantic networks, dictionaries, and domain-specific knowledgebases like ontologies, aspect hierarchies are required. Also, these knowledgebases should be scalable and with minimum noise.

A shift from syntax-based methods to semantics-based methods is expected. Also, the model should be dynamic to accommodate new entities/aspects. A domain-independent approach is required to be developed, which performs equally well for all domains. Finally, efforts should be put in to improve the performance of surveyed approaches.

Application of deep learning models like RNN and Transformer have shown promising results and indeed are part of future research direction for all NLP tasks.

VI. CONCLUSION

Detection of implicit aspects is a challenging task in aspect-level sentiment analysis. Applications and terminology for implicit aspect detection are discussed, followed by a detailed discussion of state of the art in this paper. Existing literature is categorized as supervised, unsupervised, and hybrid methods based on the algorithm applied, with unsupervised methods being the most prevalent. Performance of suggested solutions, as stated by authors, is also included for comparison purposes.

Various issues in detecting implicit aspects are discussed, and suggestions for performance improvement and future scope are also provided.

Based on our survey, we can conclude that unsupervised methods are prevalent as they do not require training data, but supervised methods are more efficient in terms of performance. Hybrid solutions also perform at par with supervised solutions as multiple methods complement each other.

We have also found that using knowledgebase may significantly improve the performance, and a shift from syntax-based methods to semantics-based methods is evident. The development of standard data sets is indispensable for implicit aspect detection. Development of solutions that are independent of language and domain are required for implicit aspect detection.

REFERENCES

[1] B. Bickart and R. M. Schindler, "Internet forums as influential sources of consumer information," *J. Interact. Marketing*, vol. 15, no. 3, pp. 31–40, 2001.

[2] B. Pang and L. Lee, "Opinion mining and sentiment analysis," *Found. Trends Inf. Retr.*, vol. 2, nos. 1–2, pp. 1–135, 2008.

[3] K. Schouten and F. Frasincar, "Survey on aspect-level sentiment analysis," *IEEE Trans. Knowl. Data Eng.*, vol. 28, no. 3, pp. 813–830, Mar. 2016.

[4] A. Sabeeh and R. K. Dewang, "Comparison, classification and survey of aspect based sentiment analysis," *Proc. Int. Conf. Adv. Informat. Comput. Res.*, 2018, pp. 612–629.

[5] H. H. Dohaiha, P. W. C. Prasad, A. Maag, and A. Alsadoon, "Deep learning for aspect-based sentiment analysis: A comparative review," *Expert Syst. Appl.*, vol. 118, pp. 272–299, Mar. 2019.

[6] T. A. Rana and Y.-N. Cheah, "Aspect extraction in sentiment analysis: Comparative analysis and survey," *Artif. Intell. Rev.*, vol. 46, no. 4, pp. 459–483, Dec. 2016.

[7] J. Z. Maitama, N. Idris, A. Abdi, L. Shuib, and R. Fauzi, "A systematic review on implicit and explicit aspect extraction in sentiment analysis," *IEEE Access*, vol. 8, pp. 194166–194191, 2020.

[8] V. Ganganwar and R. Rajalakshmi, "Implicit aspect extraction for sentiment analysis: A survey of recent approaches," *Proc. Comput. Sci.*, vol. 165, pp. 485–491, Jan. 2019.

[9] M. Tubishat, N. Idris, and M. A. M. Abushariah, "Implicit aspect extraction in sentiment analysis: Review, taxonomy, oppportunities, and open challenges," *Inf. Process. Manage.*, vol. 54, no. 4, pp. 545–563, Jul. 2018.

[10] B. Liu, *Sentiment Analysis and Opinion Mining* (Synthesis Lectures on Human Language Technologies), vol. 16. San Rafael, CA, USA: Morgan & Claypool, 2012.

[11] X. Ding, B. Liu, and P. S. Yu, "A holistic lexicon-based approach to opinion mining," in *Proc. Int. Conf. Web Search Web Data Mining (WSDM)*, 2008, pp. 231–240.

[12] *Samsung Galaxy On7 Prime Review*. Accessed: Jan. 10, 2021. [Online]. Available: https://www.gadgetsnow.com/reviews/samsung-galaxy-on7-primereview/articleshow/62426144.cms

[13] Y. Zhang and W. Zhu, "Extracting implicit features in online customer reviews for opinion mining," in *Proc. 22nd Int. Conf. World Wide Web*, Rio de Janeiro, Brazil, May 2013, pp. 103–104.

[14] Z. Hai, K. Chang, and J. J. Kim, "Implicit feature identification via co-occurrence association rule mining," in *Proc. 12th Int. Conf. Comput. Linguistics Intell. Text Process*, vol. 6608, 2011, pp. 393–404.

[15] K. Schouten and F. Frasincar, "Finding implicit features in consumer reviews for sentiment analysis," in *Proc. Int. Conf. Web Eng.*, 2014, pp. 130–144.

[16] L. Sun, S. Li, J. Li, and J. Lv, "A novel context-based implicit feature extracting method," in *Proc. Int. Conf. Data Sci. Adv. Anal. (DSAA)*, Shanghai, China, Oct. 2014, pp. 420–424.

[17] L. Liu, Z. Lv, and H. Wang, "Extract product features in Chinese web for opinion mining," *J. Softw.*, vol. 8, no. 3, pp. 627–632, Mar. 2013.

[18] A. Bagheri, M. Saraee, and F. D. Jong, "Care more about customers: Unsupervised domain-independent aspect detection for sentiment analysis of customer reviews," *Knowl.-Based Syst.*, vol. 52, pp. 201–213, Nov. 2013.

[19] Q. Su, K. Xiang, H. Wang, B. Sun, and S. Yu, "Using pointwise mutual information to identify implicit features in customer reviews," in *Computer Processing of Oriental Languages. Beyond the Orient: The Research Challenges Ahead*, vol. 4285. Berlin, Germany: Springer, 2006, pp. 22–30.

[20] B. Wang and H. Wang, "Bootstrapping both product features and opinion words from Chinese customer reviews with cross-inducing," in *Proc. 3rd Int. Joint Conf. Natural Lang. Process.*, vol. 1, 2008, pp. 289–295.

[21] S. A. Mankar and M. D. Ingle, "Implicit sentiment identification using aspect based opinion mining," *Int. J. Sci. Res.*, vol. 4, no. 11, pp. 1731–1735, Nov. 2015.

[22] B. Kama, M. Ozturk, P. Karagoz, I. H. Toroslu, and M. Kalender, "Analyzing implicit aspects and aspect dependent sentiment polarity for aspect-based sentiment analysis on informal Turkish texts," in *Proc. 9th Int. Conf. Manage. Digit. EcoSyst.*, Nov. 2017, pp. 134–141.

[23] N. Makadia, A. Choudhuri, and S. Vohra, "Aspect-based opinion summarization for disparate features," *Int. J. Adv. Res. Innov. Ideas Educ.* vol. 2, no. 3, pp. 3732–3739, 2016.

[24] S. I. Omurca and E. Ekinci, "Using adjusted Laplace smoothing to extract implicit aspects from Turkish hotel reviews," in *Proc. Innov. Intell. Syst. Appl. (INISTA)*, Thessaloniki, Greece, 2018, pp. 1–6.

[25] P. Karagoz, B. Kama, M. Ozturk, I. H. Toroslu, and D. Canturk, "A framework for aspect based sentiment analysis on Turkish informal texts," *J. Intell. Inf. Syst.*, vol. 53, no. 3, pp. 431–451, Dec. 2019.

[26] B. Dadhaniya and M. Dhamecha, "A novel approach for multiple aspect based opinion summarization using implicit features," *Int. J. Eng. Res.*, vol. 6, no. 5, pp. 68–71, May 2017.

[27] K. Schouten, O. Van Der Weijde, F. Frasincar, and R. Dekker, "Supervised and unsupervised aspect category detection for sentiment analysis with co-occurrence data," *IEEE Trans. Cybern.*, vol. 48, no. 4, pp. 1263–1275, Apr. 2018.

[28] K. Schouten, N. de Boer, T. Lam, M. van Leeuwen, R. van Luijk, and F. Frasincar, "Semantics-driven implicit aspect detection in consumer reviews," in *Proc. 24th Int. Conf. World Wide Web*, Florence, Italy, May 2015, pp. 109–110.

[29] H. Song, J. Chu, Y. Hu, and X. Liu, "Semantic analysis and implicit target extraction of comments from E-commerce websites," in *Proc. 4th World Congr. Softw. Eng.*, Hong Kong, Dec. 2013, pp. 331–335.

[30] W. Zhang, H. Xu, and W. Wan, "Weakness finder: Find product weakness from Chinese reviews by using aspects based sentiment analysis," *Expert Syst. Appl.*, vol. 39, no. 11, pp. 10283–10291, Sep. 2012.

[31] R. E. Prasojo, M. Kacimi, and W. Nutt, "Entity and aspect extraction for organizing news comments," in *Proc. 24th ACM Int. Conf. Inf. Knowl. Manage. (CIKM)*, New York, NY, USA, Oct. 2015, pp. 233–242.

[32] M. D. S. Nandhini and G. Pradeep, "A hybrid co-occurrence and ranking-based approach for detection of implicit aspects in aspect-based sentiment analysis," *Social Netw. Comput. Sci.*, vol. 1, no. 3, p. 128, May 2020.

[33] X. Xu, X. Cheng, S. Tan, and Y. Liu, "Aspect-level opinion mining of online customer reviews," *China Commun.*, vol. 10, no. 3, pp. 25–41, Mar. 2013.

[34] R. Y. K. Lau, C. Li, and S. S. Y. Liao, "Social analytics: Learning fuzzy product ontologies for aspect-oriented sentiment analysis," *Decis. Support Syst.*, vol. 65, pp. 80–94, Sep. 2014.

[35] F. Zhang, H. Xu, J. Wang, X. Sun, and J. Deng, "Grasp the implicit features: Hierarchical emotion classification based on topic model and SVM," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, Vancouver, BC, Canada, Jul. 2016, pp. 3592–3599.

[36] E. Ekinci and S. Omurca, ''Extracting implicit aspects based on latent Dirichlet allocation,'' in *Proc. Doctoral Consortium (DCAART)*, 2017, pp. 17–23.

[37] Q. Su, X. Xu, H. Guo, Z. Guo, X. Wu, X. Zhang, B. Swen, and Z. Su, ''Hidden sentiment association in Chinese web opinion mining,'' in *Proc. 17th Int. Conf. World Wide Web (WWW)*, 2008, pp. 959–968.

[38] L. Chen, J. Martineau, D. Cheng, and A. Sheth, ''Clustering for simultaneous extraction of aspects and features from reviews,'' in *Proc. NAACL-HLT*, San Diego, CA, USA, 2016, pp. 789–799.

[39] H. Wu, T. Liu, and J. Xie, ''Fine-grained product feature extraction in Chinese reviews,'' in *Proc. Int. Conf. Comput. Intell. Inf. Syst. (CIIS)*, Nanjing, China, Apr. 2017, pp. 327–331.

[40] Z. Hai, K. Chang, G. Cong, and C. C. Yang, ''An association-based unified framework for mining features and opinion words,'' *ACM Trans. Intell. Syst. Technol.*, vol. 6, no. 2, May 2015, Art. no. 26.

[41] S. K. K. Santu, P. Sondhi, and C. Zhai, ''Generative feature language models for mining implicit features from customer reviews,'' in *Proc. 25th ACM Int. Conf. Inf. Knowl. Manage.*, Oct. 2016, pp. 929–938.

[42] T. A. Rana and Y.-N. Cheah, ''Hybrid rule-based approach for aspect extraction and categorization from customer reviews,'' in *Proc. 9th Int. Conf. IT Asia (CITA)*, Kota Samarahan, Malaysia, Aug. 2015, pp. 1–5.

[43] T. Gaillat, B. Stearns, G. Sridhar, R. McDermott, M. Zarrouk, and B. Davis, ''Implicit and explicit aspect extraction in financial microblogs,'' in *Proc. 1st Workshop Econ. Natural Lang. Process.*, 2018, pp. 55–61.

[44] J. Yu, Z. Zha, M. Wang, K. Wang, and T. Chua, ''Domain-assisted product aspect hierarchy generation: Towards hierarchical organization of unstructured consumer reviews,'' in *Proc. Conf. Empirical Methods Natural Lang. Process.*, Jul. 2011, pp. 140–150.

[45] L. Qiu, ''An opinion analysis model for implicit aspect expressions based on semantic ontology,'' *Int. J. Grid Distrib. Comput.*, vol. 8, no. 5, pp. 165–172, Oct. 2015.

[46] X. Meng and H. Wang, ''Mining user reviews: From specification to summarization,'' in *Proc. ACL-IJCNLP Conf. Short Papers*, Aug. 2009, pp. 177–180.

[47] B. Shi and K. Chang, ''Mining Chinese reviews,'' in *Proc. 6th IEEE Int. Conf. Data Mining Workshops (ICDMW)*, Hong Kong, Dec. 2006, pp. 585–589.

[48] N. Zainuddin, A. Selamat, and R. Ibrahim, ''Hybrid sentiment classification on Twitter aspect-based sentiment analysis,'' *Int. J. Speech Technol.*, pp. 1218–1232, Dec. 2017.

[49] Y. Wan, H. Nie, T. Lan, and Z. Wang, ''Fine-grained sentiment analysis of online reviews,'' in *Proc. 12th Int. Conf. Fuzzy Syst. Knowl. Discovery (FSKD)*, Zhangjiajie, China, Aug. 2015, pp. 1406–1411.

[50] A. Cadilhac, F. Benamara, and N. A. Gilles, ''Ontolexical resources for feature-based opinion mining: A case-study,'' in *Proc. 6th Workshop Ontol. Lexical Resour.*, Beijing, China, Aug. 2010, pp. 77–86.

[51] L. Zeng and F. Li, ''A classification-based approach for implicit feature identification,'' in *Chinese Computational Linguistics and Natural Language Processing Based on Naturally Annotated Big Data* (Lecture Notes in Computer Science), vol. 8202. Berlin, Germany: Springer, 2013, pp. 190–202.

[52] G. Fei, B. Liu, M. Hsu, M. Castellanos, and R. Ghosh, ''A dictionary-based approach to identifying aspects implied by adjectives for opinion mining,'' in *Proc. Int. Conf. Comput. Linguistics*. Mumbai, India: IIT Bombay, 2012, pp. 309–318.

[53] X. Hu, S. Manna, and B. N. Truong, ''Product aspect identification: Analyzing role of different classifiers,'' in *Proc. IEEE Symp. Comput. Intell. Data Mining (CIDM)*, Orlando, FL, USA, Dec. 2014, pp. 202–209.

[54] D. O. Mashkin, ''Extracting aspects, category and sentiment of aspects in Russian user reviews in restaurants domain,'' in *Proc. Int. Conf. Comput. Linguistics Intellectual Technol.*, 2016, pp. 1–8.

[55] E. H. Hajar and B. Mohammed, ''Using synonym and definition WordNet semantic relations for implicit aspect identification in sentiment analysis,'' in *Proc. 2nd Int. Conf. Netw., Inf. Syst. Secur.*, 2019, pp. 1–5.

[56] H.-Y. Chen and H.-H. Chen, ''Implicit polarity and implicit aspect recognition in opinion mining,'' in *Proc. 54th Annu. Meeting Assoc. Comput. Linguistics*, Berlin, Germany, 2016, pp. 20–25.

[57] J. Yu, Z. J. Zha, and T. S. Chua, ''Answering opinion questions on products by exploiting hierarchical organization of consumer reviews,'' in *Proc. Joint Conf. Empirical Methods Natural Lang. Process. Comput. Natural Lang. Learn.*, Jul. 2012, pp. 391–401.

[58] M. Afzaal, M. Usman, and A. Fong, ''Tourism mobile app with aspect-based sentiment classification framework for tourist reviews,'' *IEEE Trans. Consum. Electron.*, vol. 65, no. 2, pp. 233–242, May 2019.

[59] L. Wang, C. Yao, X. Li, and X. Yu, ''BERT-based implicit aspect extraction,'' in *Proc. IEEE 3rd Int. Conf. Civil Aviation Saf. Inf. Technol. (ICCASIT)*, Oct. 2021, pp. 758–761.

[60] B. Liu, M. Hu, and J. Cheng, ''Opinion observer: Analyzing and comparing opinions on the web,'' in *Proc. 14th Int. Conf. World Wide Web*, 2005, pp. 342–351.

[61] S. Poria, E. Cambria, L.-W. Ku, C. Gui, and A. Gelbukh, ''A rule-based approach to aspect extraction from product reviews,'' in *Proc. 2nd Workshop Natural Lang. Process. Social Media (SocialNLP)*, Dublin, Ireland, Aug. 2014, pp. 28–37.

[62] M. Hu and B. Liu, ''Opinion feature extraction using class sequential rules,'' in *Proc. AAAI Spring Symp., Comput. Approaches Analyzing Weblogs*, 2006, pp. 61–66.

[63] S. Poria, A. Gelbukh, B. Agarwal, E. Cambria, and N. Howard, ''Sentic Demo: A hybrid-concept level-aspect based sentiment analysis toolkit,'' in *Proc. ESWC*, 2014, pp. 1–5.

[64] F. Lazhar, ''Implicit feature identification for opinion mining,'' *Int. J. Bus. Inf. Syst.*, vol. 30, no. 1, pp. 13–30, 2019.

[65] R. Sindhuja and M. Nandhini, ''An enhanced sentiment analysis model for incorporating implicit aspects,'' *Int. J. Innov. Res. Sci., Eng. Technol.*, vol. 8, no. 2, pp. 855–861, Feb. 2019.

[66] Y. V. Rubtsova and S. A. Koshelnikov, ''Aspect extraction from reviews using conditional random fields,'' in *Proc. Int. Conf. Knowl. Eng. Semantic Web*, 2015, pp. 158–167.

[67] S. Chatterji, N. Varshney, and R. K. Rahul, ''AspectFrameNet: A FrameNet extension for analysis of sentiments around product aspects,'' *J. Supercomput.*, vol. 73, no. 3, pp. 961–972, Mar. 2017.

[68] M. Mamatha, R. G. Thejeshwini, J. Thriveni, and K. R. Venugopal, ''Supervised aspect category detection of co-occurrence data using conditional random fields,'' in *Proc. IEEE Int. WIE Conf. Electr. Comput. Eng. (WIECON-ECE)*, Bengaluru, India, Nov. 2019, pp. 1–4.

[69] I. Cruz, A. Gelbukh, and G. Sidorov, ''Implicit aspect indicator extraction for aspect-based opinion mining,'' *Int. J. Comput. Linguistics Appl.*, vol. 5, no. 2, pp. 135–152, 2014.

[70] J. Feng, S. Cai, and X. Ma, ''Enhanced sentiment labeling and implicit aspect identification by integration of deep convolution neural network and sequential algorithm,'' *Cluster Comput.*, vol. 22, pp. 5839–5857, May 2019.

[71] R. Panchendrarajan, M. N. Nazick Ahamed, B. Murugaiah, S. Prakhash, S. Ranathunga, and A. Pemasiri, ''Implicit aspect detection in restaurant reviews,'' in *Proc. NAACL-HLT*, San Diego, CA, USA, 2016, pp. 128–136.

[72] H. Xu, F. Zhang, and W. Wang, ''Implicit feature identification in Chinese reviews using explicit topic mining model,'' *Knowl.-Based Syst.*, vol. 76, pp. 166–175, Mar. 2015.

[73] E. H. Hajar and B. Mohammed, ''Hybrid approach to extract adjectives for implicit aspect identification in opinion mining,'' in *Proc. 11th Int. Conf. Intell. Syst., Theories Appl. (SITA)*, Mohammedia, Morocco, Oct. 2016, pp. 1–5.

[74] W. Jiang, H. Pan, and Q. Ye, ''An improved association rule mining approach to identification of implicit product aspects,'' *Open Cybern. Syst. J.*, vol. 8, no. 1, pp. 924–930, Dec. 2014.

[75] J. Chen, L. Sun, Y. Peng, and Y. Huang, ''Context weight considered for implicit feature extracting,'' in *Proc. IEEE Int. Conf. Data Sci. Adv. Anal. (DSAA)*, Paris, France, Oct. 2015, pp. 1–5.

[76] N. Dosoula, R. Griep, R. Ridder, R. Slangen, K. Schouten, and F. Frasincar, ''Detection of multiple implicit features per sentence in consumer review data,'' in *Proc. Int. Baltic Conf. Databases Inf. Syst.*, 2016, pp. 289–303.

[77] Q. Xu, L. Zhu, T. Dai, L. Guo, and S. Cao, ''Non-negative matrix factorization for implicit aspect identification,'' *J. Ambient Intell. Hum. Comput.*, vol. 11, no. 7, pp. 2683–2699, May 2019.

[78] L. Liu, W. Du, H. Wang, and W. Song, ''A bipartite graph model for implicit feature extraction in Chinese reviews,'' in *Proc. IEEE 2nd Int. Conf. Big Data Anal. (ICBDA)*, Beijing, China, Mar. 2017, pp. 915–920.

[79] S. Khalid, M. H. Aslam, and M. T. Khan, ''Opinion reason mining: Implicit aspects beyond implying aspects,'' in *Proc. 21st Saudi Comput. Soc. Nat. Comput. Conf. (NCC)*, Riyadh, Saudi Arabia, Apr. 2018, pp. 1–5.

[80] D. H. Maylawati, W. Maharani, and I. Asror, ''Implicit aspect extraction in product reviews using FIN algorithm,'' in *Proc. 8th Int. Conf. Inf. Commun. Technol. (ICoICT)*, Yogyakarta, Indonesia, Jun. 2020, pp. 1–5.

[81] W. Wang, H. Xu, and W. Wan, ''Implicit feature identification via hybrid association rule mining,'' *Expert Syst. Appl.*, vol. 40, no. 9, pp. 3518–3531, Jul. 2013.

[82] L. Sun, J. Chen, J. Li, and Y. Peng, "Joint topic-opinion model for implicit feature extracting," in *Proc. 10th Int. Conf. Intell. Syst. Knowl. Eng. (ISKE)*, Taipei, Taiwan, Nov. 2015, pp. 208–213.

[83] H. E. Hannach and M. Benkhalifa, "WordNet based implicit aspect sentiment analysis for crime identification from Twitter," *Int. J. Adv. Comput. Sci. Appl.*, vol. 9, no. 12, pp. 150–159, 2018.

[84] M. Tubishat and N. Idris, "Explicit and implicit aspect extraction using whale optimization algorithm and hybrid approach," in *Proc. Int. Conf. Ind. Enterprise Syst. Eng. (IcoIESE)*, 2019, pp. 208–213.

[85] T. A. Rana, Y.-N. Cheah, and T. Rana, "Multi-level knowledge-based approach for implicit aspect identification," *Appl. Intell.*, vol. 50, no. 12, pp. 4616–4630, Dec. 2020.

[86] S. S. Eldin, A. Mohammed, A. S. Eldin, and H. Hefny, "An enhanced opinion retrieval approach via implicit feature identification," *J. Intell. Inf. Syst.*, vol. 57, no. 1, pp. 101–126, Aug. 2021.

[87] P. K. Soni and R. Rambola, "Deep learning, WordNet, and spaCy based hybrid method for detection of implicit aspects for sentiment analysis," in *Proc. Int. Conf. Intell. Technol. (CONIT)*, Jun. 2021, pp. 1–6.

[88] H. Cai, R. Xia, and J. Yu, "Aspect-category-opinion-sentiment quadruple extraction with implicit aspects and opinions," in *Proc. Int. Joint Conf. Natural Lang. Process.*, 2021, pp. 340–350.

[89] H. Tian and M. White, "A pipeline of aspect detection and sentiment analysis for E-commerce customer reviews," in *Proc. SIGIR eCom*, Xi'an, China, Jul. 2020, pp. 1–9.

[90] C. Wu, Q. Xiong, H. Yi, Y. Yu, Q. Zhu, M. Gao, and J. Chen, "Multiple-element joint detection for aspect-based sentiment analysis," *Knowl.-Based Syst.*, vol. 223, Jul. 2021, Art. no. 107073.

[91] D. M. Powers, "Evaluation: From precision, recall and F-measure to ROC, informedness, markedness and correlation," *J. Mach. Learn. Technol.*, vol. 2, no. 1, pp. 37–63, 2011.

PIYUSH KUMAR SONI (Member, IEEE) received the B.E. degree from Rajeev Gandhi Technical University, Bhopal, India, in 2005, and the M.E. degree from Devi Ahilya Vishwavidhyalaya, Indore, India, in 2008. He is currently pursuing the Ph.D. degree with the Narsee Monjee Institute of Management Studies, Mumbai, India.

He is currently an Assistant Professor with the Mukesh Patel School of Technology Management and Engineering, SVKM'S NMIMS, Shirpur, India. He has teaching experience of over 13 years as an Assistant Professor/Lecturer. He is an Aspiring Researcher of sentiment analysis/opinion mining. He is also a member of ISTE.

RADHAKRISHNA RAMBOLA received the B.E. degree from the University of Madras, India, the M.Tech. degree from the Allahabad Agricultural Institute, Allahabad, India, and the Ph.D. degree from T. M. B. University, Bhagalpur, India. He is currently working as an Associate Professor with the Mukesh Patel School of Technology Management and Engineering, SVKM'S NMIMS, Shirpur, India. He has experience of over 22 years in teaching, industry and research. He is an Experienced Researcher of data mining and machine learning.