Tech Science Press

Multi-Level Knowledge Engineering Approach for Mapping Implicit Aspects to Explicit Aspects

Jibran Mir[1], Azhar Mahmood[2,*] and Shaheen Khatoon[3]

[1]Computer Science Department, Shaheed Zulfikar Ali Bhutto Institute of Science and Technology (SZABIST), Faculty of Computing and Engineering Sciences, Islamabad, Pakistan
[2]Department of Computer Science, Capital University of Science & Technology, Faculty of Computing, Islamabad
[3]College of Computer Science and Information Technology, King Faisal University, Saudi Arabia
*Corresponding Author: Azhar Mahmood. Email: azhar.mahmood@cust.edu.pk

Abstract: Aspect's extraction is a critical task in aspect-based sentiment analysis, including explicit and implicit aspects identification. While extensive research has identified explicit aspects, little effort has been put forward on implicit aspects extraction due to the complexity of the problem. Moreover, existing research on implicit aspect identification is widely carried out on product reviews targeting specific aspects while neglecting sentences' dependency problems. Therefore, in this paper, a multi-level knowledge engineering approach for identifying implicit movie aspects is proposed. The proposed method first identifies explicit aspects using a variant of BiLSTM and CRF (Bidirectional Long Short Memory-Conditional Random Field), which serve as a memory to process dependent sentences to infer implicit aspects. It can identify implicit aspects from four types of sentences, including independent and three types of dependent sentences. The study is evaluated on a large movie reviews dataset with 50k examples. The experimental results showed that the explicit aspect identification method achieved 89% F1-score and implicit aspect extraction methods achieved 76% F1-score. In addition, the proposed approach also performs better than the state-of-the-art techniques (NMFIAD and ML-KB+) on the product review dataset, where it achieved 93% precision, 92% recall, and 93% F1-score.

1 Introduction

In the last decade, the advancement of the internet encouraged people to participate in social media networks. Consequently, the number of internet users is constantly increasing. IDC (International Data Corporation) foresees that the volume of data will be increased to 175 zettabytes by 2025. This massive volume of data is an excellent source of knowledge for business, political, social, and medical purposes [1]. Data engineers [2,3] have categorized data into three different

types: structured, semi-structured, and unstructured. The most challenging and favorable content generated by internet users is unstructured data or reviews. It allows the users to comment on a picture, video, or product they like or dislike. Business giants use these reviews to extract sentiments that help them profile their customers and project their future business competition position. Customers, on the other hand, use these reviews to make efficient buying decisions.

The tremendous increase in data volume makes it difficult to read every single review and determine its sentiment polarity [4]. Providentially, data science experts have proposed sentiment analysis techniques. These sentiment techniques can be classified into three categories: document level, sentence level, and aspect level. Among these three levels, the most challenging sentiment analysis technique is aspect level, which identifies a specific aspect or feature from a review and then determines its sentiment polarity [5,6]. The process of aspect identification is further divided into Explicit and Implicit aspects identification [7].

Several Aspect Based Sentiment Analysis Techniques (ABSTs) have been defined [8–14]. Nevertheless, their main contribution is explicit aspect identification. Very few ABSTs focus on implicit aspect mapping [12,15–20]. A survey on implicit aspect identification by Tubishat et al. [21] emphasized that most implicit aspect identification studies are designed for the product application domain. Mapping implicit sentence of product reviews to product-specific aspects is a simple task since the product reviews are short and simple [15–17]. For instance, the sentence "*iPhone is too expensive*" is an example of an implicit sentence that can be mapped to product-specific aspect '*price*.' However, the sentences in the reviews are not often independent. Sometimes, it is essential to connect a sentence to a preceding sentence to infer the opinion target. In this research, the different types of implicit sentences have been explored to map explicit aspects to implicit aspects using the following four types of sentences.

(1) Type I is an independent sentence, where the mapping algorithm simply explores the n-gram opinion words to decide their associated aspects.
(2) Type II is a dependent sentence, where the algorithm consults the preceding sentence memory to maps the sentence to a specific aspect.
(3) Type III is also a dependent sentence; however, it handles pronouns using the preceding sentences.
(4) Type IV sentences are too short or too long for their meaning to be interpreted successfully. The mapping algorithm declares these sentences ambiguous.

The state-of-the-art implicit aspect mapping techniques [22,23] are developed for product application domains. Moreover, the product application domain only uses Type I implicit sentences [24] to identify relevant product aspects. This paper proposed a multi-level knowledge engineering approach to infer movie-related implicit aspects inspired by the product application domain. The proposed approach first identifies the explicit aspects, which serve as a memory to identify implicit aspects. In addition, the proposed approach is capable of identifying the above four different types of dependent and independent sentences to identify movie-specific implicit aspects from the reviews. Finally, the proposed approach is validated on a large movie review dataset and compared with the state-of-the-art techniques on the product review dataset, discussed in the results section.

2 Literature Review

Feng et al. [25] proposed an implicit aspect identification algorithm on product reviews in Chinese. After basic pre-processing, the authors used a convolution neural network (CNN) to

identify aspect, sentiment shifter, sentiment intensity, and sentiment words. In the next step, sequential algorithms are used to annotate the corpus. Implicit aspect identification was achieved by exploring the context of a single sentence or word due to simple and short reviews of users' product reviews. In contrast, movie reviews consist of different types of implicit sentences. For example, Liao et al. [26] classified different sentences into Fact-implied, Metaphorical, Rhetorical-question, and Ironic types. The study claimed that 72% of the reviews consist of the Fact-implied sentence. They used a hybrid approach such as Semantic Dependency Tree-based CNN (SDT-CNN)-based fusion model and applied it to car review and social media (Weibo). However, these two application domains are still less complex than movie application domain. In addition, their approach only identifies explicit aspects.

In contrast, Tubishat et al. [27] proposed a model of explicit and implicit aspect identification for a product application domain. They used Whale Optimization Algorithm, dependency parser relations, and web-based similarity to extract explicit aspects. Whereas, for implicit aspect identification, word co-occurrence, dictionary-based, and web-based similarity were used. The main limitation of this model is its domain dependency and lack of scalability to other domains. Similarly, Afzaal et al. [28] proposed explicit and implicit aspects identification for tourism applications. Explicit aspects were identified using a dependency parser, and WordNet and implicit aspects were determined using a predefined decision tree. However, the limitation of this model is its inefficiency in identifying implicit aspects from long and complex sentences.

Likewise, Gupta et al. [29] proposed an integrated approach to summarize mobile reviews. However, the model is limited to only explicit aspects and opinions associated with those aspects. Moreover, the mobile application domain is a form of a product application domain. Subsequently, the proposed model fails to perform efficiently when applied to the movie application domain. The study [30] is an extension of [28], where an aspect-specific lexicon was built for identifying co-occurrence between sentiment words and explicit aspects. The lexicon is then used to construct an aspect-sentiment tree. However, the proposed implicit aspect identification approach identifies only aspects from independent sentences. Asghar et al. [31] proposed a hybrid model by integrating unsupervised and supervised approaches to extract explicit aspects and aspect-level sentiment determination. However, the model is not robust in extracting implicit aspects. In addition, the use of POS patterns to extract aspects detects irrelevant aspects as well.

Similarly, El Hannach et al. [32] also proposed a hybrid approach to identify implicit aspects from product reviews. Their approach first identifies candidates of implicit aspects and then maps them to any of the explicit aspects. Many semantic tools such as WordNet and a frequency-based approach for implicit aspect identification were used. The limitation of this work is handling dependency in sentences similar to the above approaches. Khalid et al. [33] proposed an unsupervised approach to identify implicit aspects from a Twitter dataset. They asserted that their proposed framework not only mines aspect-level sentiment but also elaborates the reasons for liking and disliking. This framework makes use of the association of linguistic patterns for identifying implicit aspects. The study is limited to explores implicit aspects within the sentence but fails to handle dependent sentences.

Few studies focus on aspect-based sentiment analysis. For instance, Yang et al. [34] proposed an attention network with LSTM that identifies explicit aspects and determines their sentiments as well. The experiments were conducted on the product and Twitter-based datasets. Similarly, Liu et al. [35] proposed attention model by using local and global attention mechanisms to identify aspect-based sentiments. The local mechanism calculates weights between words in a sentence

while the global one performs this task on the entire review. The aspect identification process was evaluated on SemEval-2015 using restaurant reviews.

Tang et al. [36] used a hybrid technique for aspects and opinion identification and then aspect-wise sentiment determination. The study used topic modeling for aspect and opinion classification and then sentiment determination. The proposed model is implemented on Yelp and Amazon datasets. Apart from the product, service, and movie application domains, Chauhan et al. [37] presented an aspect-based sentiment analysis for the educational domain. They applied linguistic features such as POS tagger, tokenizer, stop word removal, and word case in the pre-processing phase. After that, the most occurring NOUN/NOUN phrases were considered as aspect candidates. Then a pruning technique was applied to filter out irrelevant aspects. The technique is validated on online educational comments. However, this study also ignored the implicit aspects. Shams et al. [38] proposed a language-independent aspect-based sentiment analysis technique using an unsupervised method for English and Persian language. For aspect identification, it used ELDA topic modeling, which is validated on product reviews. In contrast, Jibran et al. [39] proposed an aspects classification model for movie reviews. Several pre-processing techniques were used for multi-class annotation and linguistic patterns to eliminate irrelevant movie aspects. CRF classifier was used to not only identify movie-specific aspects but also person and movie title as movie aspects.

Based on the above literature analysis, we identified two main limitations in the previous studies. The first is that state-of-the-art techniques for implicit aspect identification have been proposed mostly for product and service domains that are relatively simpler than complex domains such as movie reviews. The second is that the proposed solutions only identify implicit aspects from straightforward independent sentences or Type I sentences. To overcome the stated limitations, this study first proposed an implicit aspects identification phase for movie reviews. Secondly, we developed novel techniques for processing four different types of independent sentences to detect movie-specific implicit aspects accurately. The following section provides the technical detail of each technique.

3 Proposed Approach for Mapping Implicit Aspects to Explicit Aspects

An end-to-end architecture of the proposed approach is shown in Fig. 1, which consists of annotation, explicit aspects identification [40], and implicit aspects mapping phases. In the following section, the techniques used in each stage are described.

3.1 Annotation Phase of Movie Reviews

The first phase of the proposed approach is the annotation phase, which annotates data for training BiLSTM-CRF, a supervised learning technique. The linguistic features (Part of speech and chunking information), orthographic features (word casing information), and movie dictionary are used as a feature set. The linguist tool such as OpenNLP[1] is used to transform each unstructured review into a structured review. Therefore, the annotation phase uses a tokenizer, POS tagger, and chunker to annotate the dataset. The orthographic features include punctuation, word casing, suffix, and prefix since writers often use these orthographic features to emphasize a word or a group of words. We used these orthographic features to extract NERC (Named Entity Recognition and Classification) [41] inspired by [42], where orthographic features were used to identify NER (named entity recognition) in a social application domain. Although these linguistic

---

[1] https://opennlp.apache.org/

features are very useful in the annotation process, yet they also annotate irrelevant aspects. For instance, the annotation phase annotates a word as a person's name or a movie title if it has POS (Parts of Speech) tag as NOUN or NOUN PHRASE. However, not every word with NOUN or NOUN PHRASE is a relevant person's name or a movie title. To remove these types of irrelevant aspects, proposed approach used imdbpy[2].
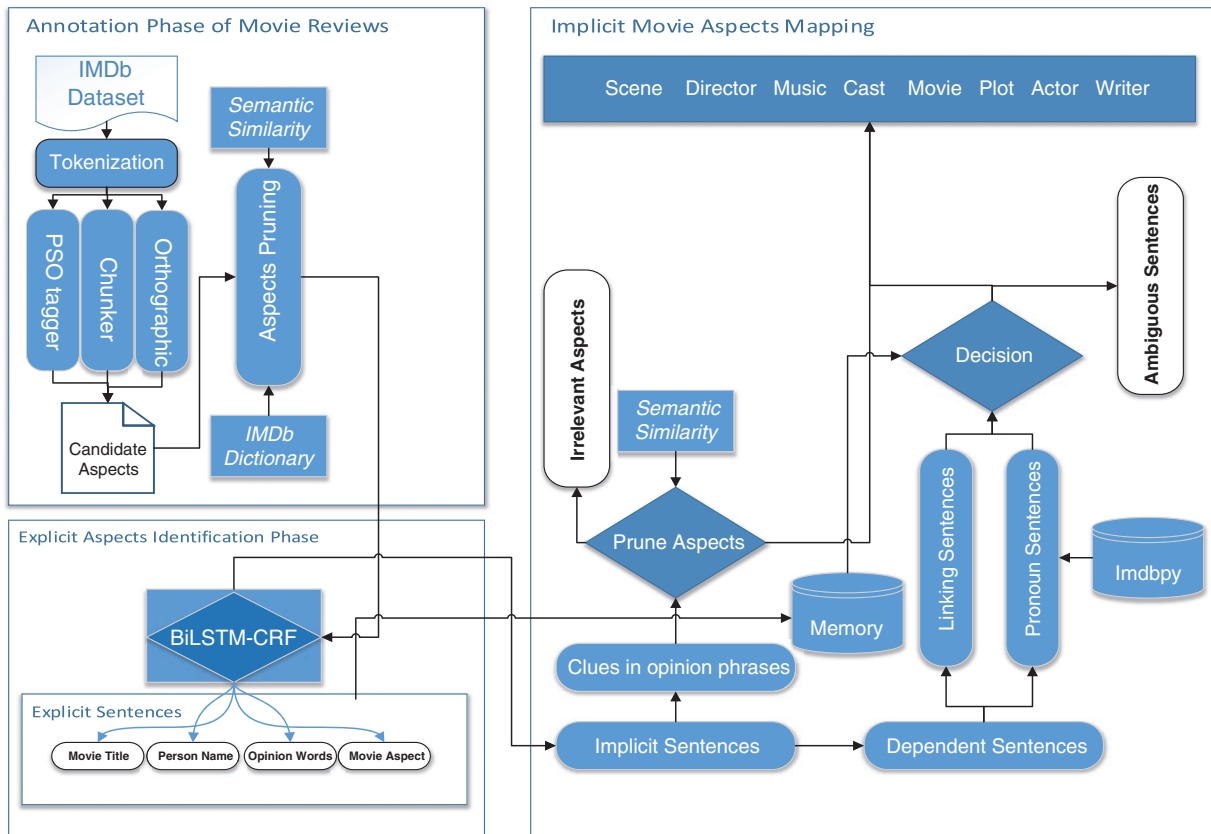
Figure 1: Overall architecture of mapping implicit aspects to explicit aspects

Tab. 1 demonstrates the relevant extraction of a person name entity. If the POS is "NNP," chunker is "B-NP" or "I-NP," the word case is "TC," the relevant IOB (inside outside beginning) annotation is *person* entity, i.e., *B-Person* or *I-Person*.

Table 1: The annotation of person entity using orthographic features

Token	Parts of speech	Chunker	Word	Inside outside beginning
Sally	NNP	B-NP	TC	B-Person
Kirkland	NNP	I-NP	TC	I-Person

---

[2] https://imdbpy.github.io/

These features also accept irrelevant entities as a personal name, as shown in Tab. 2. In this case, the "Bermuda Triangle" is not a person entity, but the conditions are satisfied, and entities are extracted as a personal name. We used the imdbpy library to removes irrelevant movie-related person entities. The movie titles are extracted by following the same procedure except for person names Bi-gram feature are extracted, whereas for title N-gram features are extracted.

Table 2: The irrelevant extraction of a person entity

Token	Parts of speech	Chunker	Word case	Inside outside beginning
Bermuda	NNP	B-NP	TC	B-Person
Triangle	NNP	I-NP	TC	I-Person

Apart from movie NEs (Named Entities), movie-specific aspects are extracted using nouns or noun phrases as candidate's movie-specific aspects. We used the SpaCy[3] similarity matrix to filter out irrelevant movie-specific aspects. A list of seed words such as "scene," "director," "music," "cast," "movie," "plot," "actor" and "writer" are defined to extract movie-specific aspects. The candidate word for movie-specific aspects and seed words are provided as input to the SpaCy similarity matrix to identify similarity. In addition, a predefined threshold, i.e., 45%, is defined if the similarity between a candidate word and seed word is greater than or equal to the threshold, then the candidate word is a relevant movie-specific aspect.

Tab. 3 shows an example of candidate a word and seed words semantic similarity. In this case, the candidate word *screen* is a relevant movie-specific aspect since its semantic similarity and the semantic similarity of seed words "movie" are the highest among the other entries.

Table 3: An example of candidate word and seed words semantic similarity

Candidate word	Seed words	Semantic similarity
Screen	Scene	0.37093803
Screen	Director	0.20464844
Screen	Music	0.26128027
Screen	Cast	0.35902005
Screen	Movie	0.46251473
Screen	Plot	0.28155455
Screen	Actor	0.31339672
Screen	Writer	0.16731432

3.2 Explicit Aspects Identification Phase

The annotated dataset from the previous phase is then used to train Bidirectional LSTM-CRF for explicit aspects identification. Previous studies show that CRF performs efficiently on sequential data and named entity recognition [43–45]. However, the recent studies [46–48] have shown that CRF performs less efficiently when used stand-alone, therefore, these studies have

---

[3] https://spacy.io/usage/vectors-similarity

integrated CRF with variants of neural networks. We also exploit CRF capabilities and variants of neural networks, i.e., BiLSTM-CRF, in the explicit aspects' identification phase. Nevertheless, the BiLSTM-CRF requires annotated dataset; therefore, we annotated the dataset using IOB (Inside outside beginning) tagging scheme, which identifies five entities, include person (B-Person, I-Person), movie title (B-Movie, I-Movie), movie-specific-aspects (B-Feature, I-Feature), opinion chunks (B-Feature, I-Feature), and others (O).

Fig. 2 shows a five-layered Bidirectional LSTM-CRF framework. The explicit aspects identification phase takes a sequence of token words to train the model to identify person and movie names, movie-specific aspects, and opinion words. The bottom layer is the word embedding of the token words. The next two layers in the middle are Backward and Forward LSTM, which receives the word embedding and connects with a dense layer. Finally, the top layer is CRF, which tags the sentences with movie-specific entities using the proposed IOB tagging scheme.
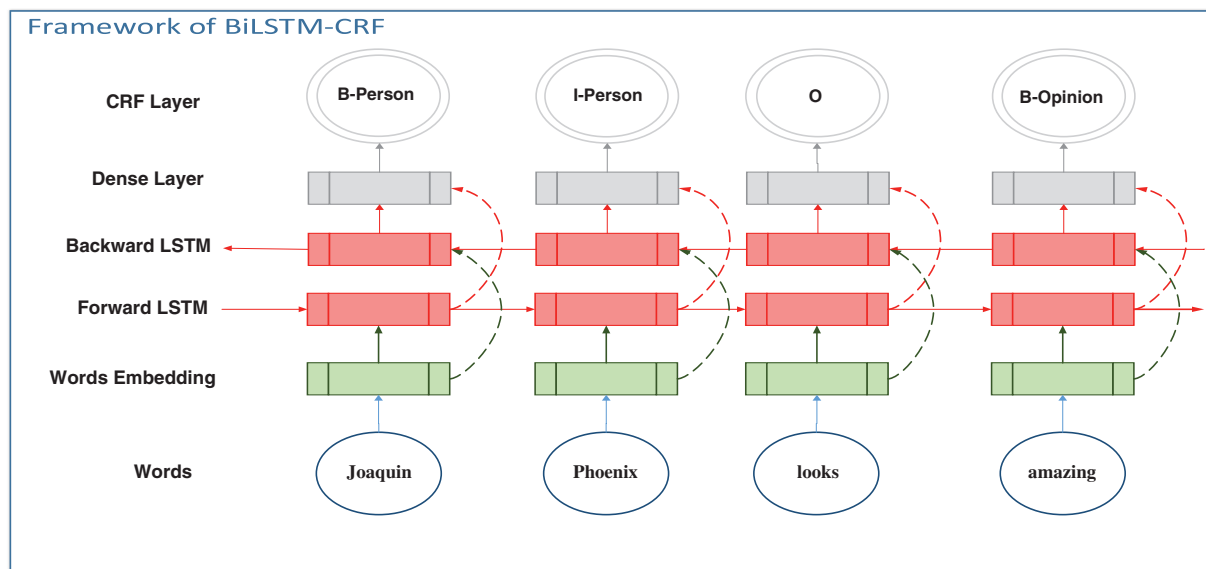

Figure 2: A five-layered BiLSTM-CRF, LSTM-CRF, and CRF for explicit aspects identification

3.3 Implicit Aspects Mapping

In the Explicit Movie Aspect identifier, four named entities, namely Person Name, Movie Title, Feature, and Opinion, are identified. In the next step, these entities are used as a memory for implicit aspect mapping. Tab. 4 shows the parts of two sentences. "Sentence-1" is identified with Feature and Opinion entities, whereas Sentence-2 is identified with Person and Opinion entities. These aspects are called explicit aspects, and the corresponding sentences are called explicit sentences. The implicit aspects are not mentioned explicitly; thus, they need to be interpreted [49] as one of the explicit aspects. Sentences that contain opinion words but no explicit aspects are implicit.

For example, the sentence, "*The cryptic dialogue would make Shakespeare seem easy to a third grader*," refers to the implicit movie aspect *script*. To map implicit aspects, we adopt the divide and conquered strategy. In the first step, the sentences with explicit aspects and opinion words or phrases or entities are separated from a review. Tab. 5 illustrates explicit and implicit sentences.

These sentences are tokenized and labeled with a sentence number, POS, and IOB tag. In addition, the explicit and implicit sentences are connected by a sentence number. For instance, the explicit sentence is a sentence-2 of review-1. Similarly, the implicit sentence is sentence-6 of review-1. Among different types of implicit sentences, Tab. 5 shows only a Type-I implicit sentence where an aspect can be hidden in consecutive opinion words or chunks [50]. However, this cannot always be true since different implicit sentences require different treatment, as described in the following section.

Table 4: An excerpt for named entity recognition

Part of sentence-1				Part of sentence-2			
12	Starts	VB	O	115	And	CC	O
13	Out	RP	O	116	Frederic	NNP	B-person
14	With	IN	O	117	Forrest	NNP	I-person
15	A	DT	O	118	Can	MD	B-opinion
16	Opening	NN	O	119	Be	VB	I-opinion
17	Scene	NN	B-feature	120	Seen	VBN	I-opinion
18	That	WDT	O	121	Briefly	NN	O
19	Is	VBZ	B-opinion	122	–	–	O

Table 5: An example of explicit *vs.* implicit sentences

Explicit sentence	Implicit sentence
2 starts VB O	6 the DT O
2 out RP O	6 cryptic JJ B-opinion
2 with IN O	6 dialogue JJ B-opinion
2 a DT O	6 would MD B-opinion
2 opening NN O	6 make VB I-opinion
2 scene NN B-Feature	6 Shakespeare NNP O
2 that WDT O	6 seem VB B-opinion
2 is VBZ B-opinion	6 easy JJ B-opinion
2 terrific JJ B-opinion	6 to TO O
2 example NN I-opinion	6 a DT O
2 of IN O	6 third JJ B-opinion
2 absurd JJ B-opinion	6 grader NN I-opinion
2 comedy NN I-opinion	6 - - - O

3.3.1 Type I Implicit Sentences

In Type I implicit sentences, consecutive words are inspected if those words are identified as opinion entities. For example, if the current word's POS is JJ and the next word's POS is NN, then the word that has NOUN POS is a potential candidate for implicit aspect mapping. In the previous example, two consecutive word patterns, '*Cryptic*' **(JJ)**, '*Dialogue*' **(NN)**, and '*Third*' **(JJ)** '*Grader*' **(NN)**, will be extracted. Therefore, the potential candidates are the '*Dialogue*' and '*the Grader*.' The next step calculates the similarity score between potential candidates and movie-

specific aspects. The candidate words are considered a movie aspect if their score is more or less than the specified threshold. In this scenario, '*Dialogue*' is mapped to aspect 'Script,' where '*Grader*' is ignored since it is semantically not closer to movie-specific aspects.

Tab. 6 shows the algorithm for Type I implicit sentences. The sentences are tokenized, and each token's POS label and IOB label are tagged. Therefore, SN_i, $Word_i$, POS_i, and $IOBtag_i$ is given as an input, where SN_i denotes an array of sentence number, $Word_i$ denotes an array of words, POS_i denotes an array of POS and $IOBtag_i$ denotes an array of IOB tags. In addition, the seed words "scene," "director," "music," "cast," "movie," "plot," "actor" are also given as an input. and the output is relevant movie specific aspects.

Table 6: Algorithm for extracting movie-specific aspects from type I sentences

---

Input: a set of implicit sentences having adjectives and nouns $SN = \{s_1, s_2, s_3, \ldots, s_n\}$
Output: $IS_{Map\ with\ movie\ aspects} = \{is^1_{aspect}, is^2_{aspect}, is^3_{aspect}, \ldots, is^n_{aspect}\}$
1: **for** $i \leftarrow 0$ *to n total number of SN* (*sentence number*)
2:　　**if** POS_i **is** *JJ* **and** POS_{i+1} **is** *NN* **or** *NNS* **or** *NNP*
3:　　　　$Candidate \leftarrow Word_i$
4:　　**for** $j \leftarrow 0$ *to* **n** *seed words*
5:　　　　**if** $semantic\_similarity(Candidate, s_j)$ **is** $\geq$ 45 *percent*
6:　　　　　$write\_to\_file(SN_i, i, Word_i, POS_i, IOBtag_i)$
7:　　　　　　$AI_j \leftarrow writeFile(TK_j, POS_j, CK_j, CL, iobT, similarity\_per)$
8:　　　　**endif**
9:　　**endfor**
10:　**endif**
11: **endfor**

3.3.2 Type II Implicit Sentences

Type II implicit sentences are those in which linking words such as "instead," "in addition," "therefore," etc., are used. Fig. 3 shows a sample of implicit sentences, where opinion words are not helpful to be mapped to the movie-specific aspects; therefore, the sentence is considered as a dependent sentence.

But beyond that, I simply had no connection with or sympathy for any of the characters. Instead I felt only contempt for this parade of shuffling, whining, nicotine stained, martyrs in a perpetual quest for identity.

Figure 3: An example of explicit and implicit sentences connected with linking word

An implicit sentence is considered a dependent sentence if it starts with transition words, providing detail, or explaining a preceding sentence. For example, in Fig. 4, it is very difficult to understand what the sentence is talking about without consulting the preceding sentence. In this case, the prior sentence is sentence-11, an explicit sentence referring to the movie aspect '*character*.' After that, the implicit sentence follows the linking word '*instead*,' which expresses more sentiments about the aspect '*character*.' Fig. 4a represents a memory structure for the implicit sentence in Fig. 4b, which assists the algorithm in deciding the exact aspect. The memory is formed from an

explicit aspect identification which is identified by the BiLSTM-CRF framework in the previous phase. The memory contains the sentence number, POS, IOB tagging, and word index of explicit sentences.

```
9 Bergman NNP I-Person 141
10 film NN B-Feature 144
11 characters NNS B-Feature 167
13 story NN B-Feature 180
14 Everyone NN B-Feature 189
15 Everyone NN B-Feature 200
```

(a)

```
12  Instead RB  O   O
12  I    PRP O   O
12  felt     VBD B-Opinion   O
12  only     RB  O   O
12  contempt     NN  O   O
12  for IN  O   O
12  this     DT  O   O
12  parade  NN  O   O
12  of  IN  O   O
12  shuffling    VBG O   O
12  ,    ,   O   O
12  whining VBG B-Opinion   O
12  ,    ,   O   O
12  nicotine     JJ  B-Opinion   O
12  stained JJ  I-Opinion   O
12  ,    ,   O   O
12  martyrs NNS O   O
12  in  IN  O   O
12  a    DT  O   O
12  perpetual    JJ  B-Opinion   O
12  quest   NN  I-Opinion   O
12  for IN  O   O
12  identity     NN  O   O
12  .    .   O   O
```

(b)

Figure 4: An example of type II implicit sentence (a) Memory (b) Implicit sentence

Tab. 7 shows an algorithm that uses preceding explicit sentences from memory to map an implicit sentence. The algorithm takes classes $Review_{List}$ and $Review_{Columns}$ with five properties, namely $SN(sentence\ number)$, $Word$, POS, $IOBtag$ and $Status$. The output is implicit sentences mapped to specific movie aspects. The algorithm is also capable of processing Type II and IV implicit sentences.

Table 7: Algorithm to maps independent and dependent sentences

Input: $Review_{List}\langle Review_{Columns}\rangle$
Output: Movie aspects $IS_{Map\ with\ movie\ aspects} = \{is^1_{aspect},\ is^2_{aspect},\ is^3_{aspect},\ \ldots,\ is^n_{aspect}\}$
1: **foreach** $Current_{Sentence}$ **in** $Review_{List}$
2:   **if** $Current_{Sentence}.Length < 30$ **or** $Current_{Sentence}.Length < 3$
3:     **if** $Current_{sentence}.POS$ **is** "RB"
4:       **if** $Memory_{preceding Sentence}(Current_{sentence}.SN)$ //check if there is any explicit aspect of preceding sentence
5:           $Current_{Sentence}.Status \leftarrow Preceding_{Sentence}.Status$
6:       **endif**
7:     **endif**
8:   **endif**
9: **forend**

3.3.3 Type III Implicit Sentences

While Type II implicit sentences are based on linking words, Type III sentences are dependent on personal pronouns such as he, she, and it. Fig. 5 shows a dependent sentence that is based on a personal pronoun. The first sentence explicitly talks about the named entity '*Kareena Kapoor*', while a personal pronoun is used in the second sentence.

Kareena Kapoor has never looked so beautiful. She looks stunning and leaves you, all stand up.

Figure 5: An example of a type III sentences

Like Type II sentences, we used memory to map whether the dependent sentence talks about Actor, Writer, Director, or Movie. Fig. 6 shows Type III sentence and corresponding memory with mapping information. The first column in the memory is the number of sentences that contain named entities. The second, third, and fourth columns are names of person, POS, and IOB tags. The fifth column is the index number of the word in a sentence. The last column is the status column that indicates the status of a specified name. In this example, the name '*Kareena Kapoor*' is an actress.

```
13   Kareena NNP B-Person      128 Actress        14   She PRP O    Actress
13   Kapoor  NNP I-Person      129 Actress        14   looks   VBZ B-Opinion   O
15   Ali NNP B-Person      137 N/A                14   stunning    JJ  B-Opinion   O
15   Khan    NNP I-Person      138 N/A            14   and CC  O    O
16   Sanjay  NNP B-Person      147 Actor          14   leaves  VBZ B-Opinion   O
16   Mishra  NNP I-Person      148 Actor          14   you PRP O    O
16   Yashpal NNP B-Person      153 Actor          14   ,   ,   O    O
16   Sharma  NNP I-Person      154 Actor          14   all DT  O    O
                                                  14   stand   VBP B-Opinion   O
                (a)                               14   up  RP  O    O
                                                  14   .   .   O    O

                                                                (b)
```

Figure 6: An example of a type III sentence based on personal pronounce and associated memory (a) Memory (b) Type III sentence

3.3.4 Type IV Implicit Sentences

Type IV Implicit sentences are either too lengthy or too short to map to specific movie aspects. Fig. 7 demonstrates three sentences that are exceeding the predefined sentence length (30 words). These sentences also contain several conjunction words (and) and are separated by ','. In addition, merging too many ideas in a single sentence makes them ambiguous.

The ambiguous sentences appearing in Fig. 7 are ignored and not included in the mapping process. Tab. 8 shows an algorithm to map both Type III and IV sentences.

It all comes through in the shopworn visual/aural tricks: the deafening scratching of a pencil on paper, the towering surf that dwarfs the people walking on the beach. etc, etc.

Yes, there the goof-ups were original, like in the beginning a car is shown with number UP**** number and few seconds later it starts falling down a hill with number MH**** .

Throw in a deranged don with an obsession for English, a call center smart Alec, a femme fa tale who can don a ***** and a Saree with the same aplomb, a levitating, gravity defying hit-man and a cop with a hundred (or was it a thousand) black cat commandos on their trail....good ingredients in competent hands.

Figure 7: Example of ambiguous sentences

Table 8: Algorithm III maps pronoun based implicit sentences to actor or movie aspect

---

Input: $Review_{List}\langle Review_{Columns}\rangle$
Output: A set of implicit sentences $IS_{Map\ with\ movie\ aspects} = \{is^1_{aspect}, is^2_{aspect}, is^3_{aspect}, \ldots, is^n_{aspect}\}$
1: *foreach* $Current_{Sentence}$ *in* $Review_{List}$
2:     *foreach* $Word_j$ *in* $Current_{sentence}$
3:         *if* $Word_j$ *is* '*He*' *or* '*She*' *or* '*You*'
4:             *if* $Memory_{precedingSentence}(Current_{sentence}.SN)//check$ *if there is any name entity of*
                 preceding sentence
5:                 $Current_{Sentence}.Status \leftarrow Actor$
6:             *endif*
7:         *endif*
8:         *if* $Word_j$ *is* '*It*'
9:             *if* $Memory_{precedingSentence}(Current_{sentence}.SN)//check$ *if there is any name entity of*
                 preceding sentence
10:                 $Current_{Sentence}.Status \leftarrow Movie$
11:             *endif*
12:         *endif*
13:   *endfor*
14: *endfor*

---

4 Experimental Results

The proposed approach is validated on the Movie Dataset[4] contains 50k examples. The dataset is divided into 25k for training and 25k for testing by the dataset builders. We further partitioned 24k data into 12k negatives and 12k for positives reviews. To train LSTM and BiLSTM, we portioned the dataset into 80 percent for training and 20 percent for testing with default layers' settings of both models. Python programming was used to conduct all experiments. The experimental setup for BiLSTM is the following. The parameters given to the embedding layer are input dimension: the total number of token words, the output dimension: 20, input length: 75, and mask zero: true. For the BiLSTM the parameters are units: 50, return sequences: true and recurrent dropout: 0.1. Next, the parameters for the dense layer are dense dimension: 50 and activation: "relu." Finally, the CRF layer was given the labels (person name, movie title, feature, and opinion labels), which tags the sentence as an output layer.

---

4 https://ai.stanford.edu/~amaas/data/sentiment/

Fig. 8 shows the overall accuracy of the BiLSTM-CRF, LSTM-CRF, and CRF. The BiLSTM-CRF performs better than the LSTM-CRF and CRF. The precision, recall, and F1-score of the proposed BiLSTM-CRF are 89.9%, 88.9%, and 89.4%. In comparison, CRF performs better than LSTM-CRF since it depends on feature engineering. When these features were removed, CRF performance degraded considerably with the precision, recall, and F1-score of 87.1%, 87.8%, and 87.4%, respectively. In contrast, LSTM-CRF performance is lower than CRF in terms of recall and F1-score. Conversely, it performs efficiently, although there is no use of handcraft features. The precision, recall, and F1-score of the LSTM-CRF are 87.5%, 86.2%, and 86.9%.
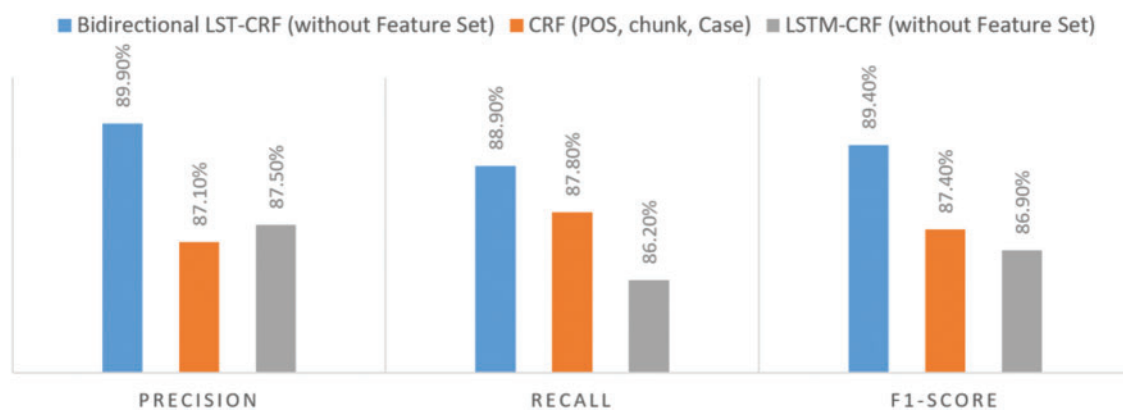

Figure 8: Overall accuracy of the explicit aspects identifier and conventional approaches
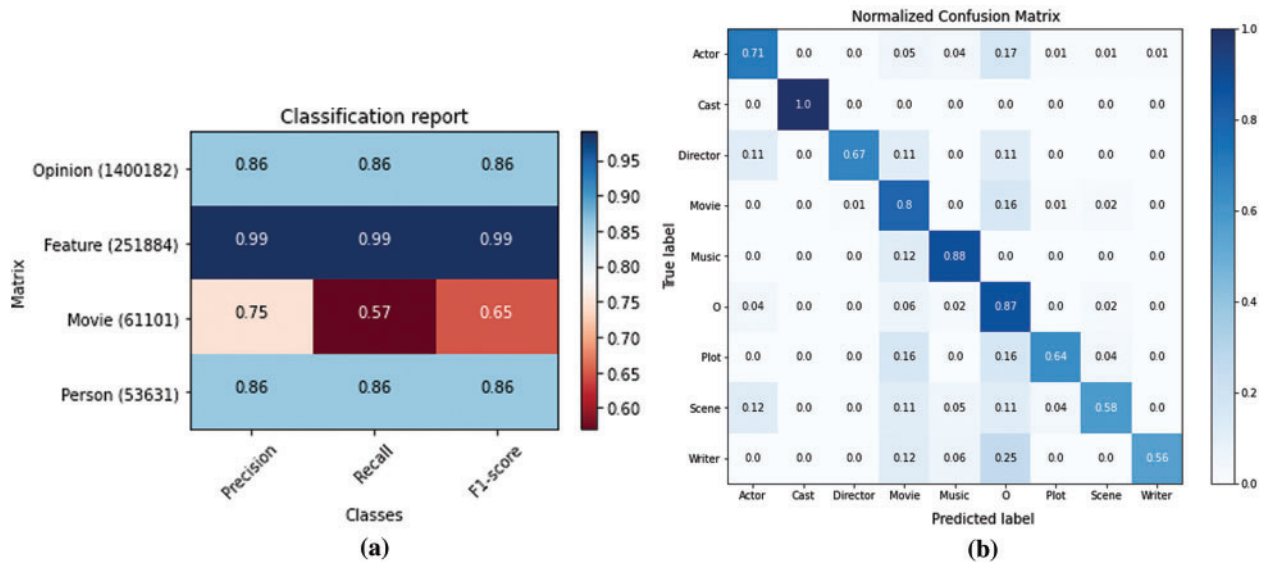
The entity-wise performance of BiLSMT-CRF and LSTM-CRF is shown in Fig. 9. Fig. 9a shows the performance BiLSMT-CRF, where Feature Entity shows the most magnificent performance. The EMAI (Explicit Movie Aspects Identifier) performs medium on entities such as Person and Opinion. Some movie titles are too long or too short to be identified as a title entity. Thus, for the classifier, it becomes hard to classify the correct movie title. Fig. 9b demonstrates the entity-wise performance for LSTM-CRF. Like EMAI, it also performed efficiently on feature entities. It showed more than 80% performance on Opinion and Person entities. The worst performance was recorded on the title entity.

Fig. 10a depicts the entity-wise performance of CRF. It also performs better on feature entities, whereas on title entity its F1-score is less than LSTM-CRF and BiLSTM-CRF. However, it has more than 80% F1-score for opinion and person entities. Likewise, a large Movie Dataset[5] was used to evaluate the performance of the implicit aspects mapping. Fig. 10b represents a confusion matrix for analyzing nine different classes (Actor, Cast, Director, Movie, Music, O (others), Plot, Scene, and Writer). Cast, Movie, Music, and O received 80% or above True Positive (TP) percentage. The class Cast showed exceptional results since it appeared only four times in the implicit sentences and was identified as TP. In contrast, the Actor class has 71 percent TP result, and Director, Plot, Scene, and Writer classes showed less than 70 percent TP.

[5] https://ai.stanford.edu/~amaas/data/sentiment/
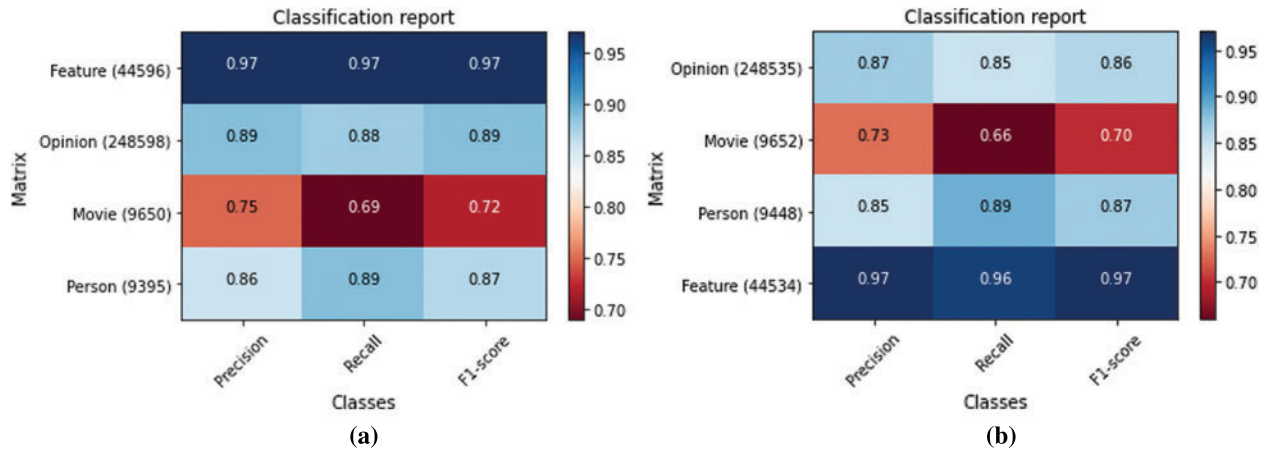
Figure 9: Entity wise performance of the BiLSMT-CRF and LSTM-CRF (a) Entity wise performance of the BiLSMT-CRF (b) Entity wise performance of LSTM-CRF

Figure 10: Entity wise performance of CRF and classification error analysis (a) Entity wise performance of CRF (b) Confusion matrix

The evaluation measures of the aspect mapping algorithm are shown in Fig. 11. It can be noticed from Fig. 11a. that the aspect *Cast* precision, recall, and F1-scores are very high since the support is less in number (four only). The aspect *Director* achieved 86% precision, 67% recall, and 75% F1-score. It showed low recall due to insufficient support (nine only). The aspect support is 83, and its precision, recall, and F1-score are 69%, 80%, and 74%. In contrast, the Movie aspect has better recall but low precision. Therefore, it showed more false positives. Similarly, the aspect *Music* achieved low precision of 58%, recall of 88%, and F1-score of 70%. O (others) is not any

aspect of a movie. Moreover, it represents a sentence that does not match any specific aspects. However, *Plot, Scene,* and *Writer* showed high precision of 80%, 82%, and 90%, respectively. In comparison, they are low in recall that is 64%, 58%, 56%, respectively. The aspects *Director, Plot, Scene,* and *Writer* received more false negatives and less false positives, although *Movie and Music* are predicted more as False Positive. The overall accuracy of the movie mapping algorithm was 77%. Fig. 11b shows the average macro precision, recall, and F1-score of the proposed algorithm 80%, 74%, and 76%. Similarly, here recall is also lower than precision for movie aspect mapping algorithm.
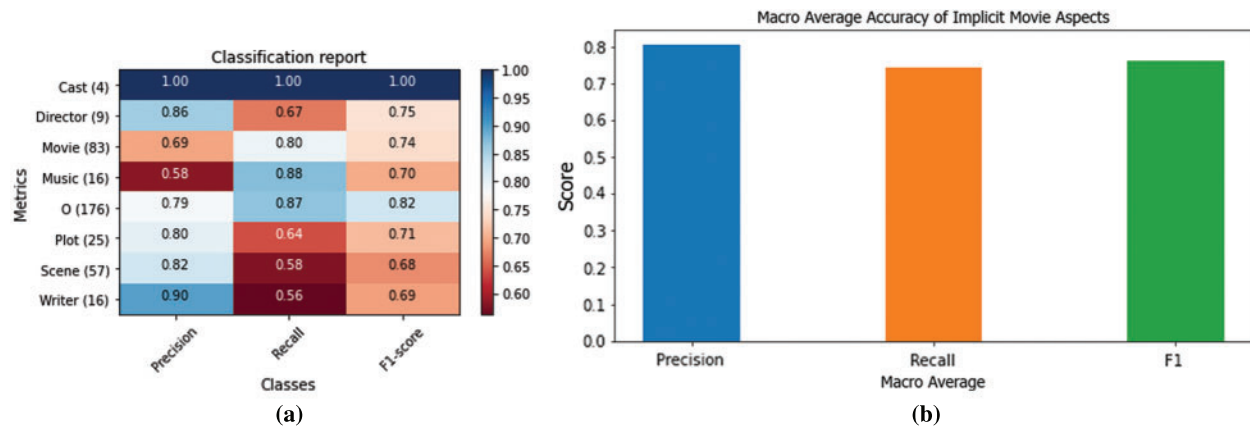
Figure 11: Overall evaluation measures of the aspects mapping algorithm (a) Specific aspect-based Precision, Recall, and F1-score (b) Macro average precision, recall, and F1-score

The product review dataset [24] was used for the experiment to compare the proposed approach with the state-of-the-art techniques. The dataset includes five different electronic products. The lengthy names are D1 for Apex AD2600 Progressive-scan, D2 for Canon G3, D3 for Creative Labs Nomad Jukebox Zen Xtra 40 GB, D4 for Nikon coolpix 4300, and D5 for Nokia 6610. In D1, the total number of sentences are 740, the total number of implicit sentences are 38, and the implicit aspects are 38. D2 contains 597 sentences, 18 implicit sentences, and 20 implicit aspects. D3 has 1716 sentences, 56 implicit sentences, and 60 implicit aspects. D4 contains 346 sentences, 14 implicit sentences, and 14 implicit aspects. Finally, D5 contains 546 sentences, 23 implicit sentences, and 24 implicit aspects. The dataset is already annotated for Type I implicit sentences. Therefore, the implicit aspects can be identified in the sentences. However, the dataset is not annotated for Type II and Type III.

The product-specific aspects are *performance, functionality, price, weight, size, behavior, appearance,* and *quality*. The challenge is to find the clues in the implicit sentence, therefore, it can be mapped to one of the product-specific aspects. To solve this problem, we built word embedding for each of the product-specific aspects. For instance, the aspect *performance* has a vocabulary that can identify implicit aspects by finding similarities between VB, NN, and JJ in an implicit sentence. Similarly, for other aspects, separate vocabularies are built and validated previously on implicit aspects identification models such as ML-KB+ [22] and NMFIAD [23]. Tab. 9 shows the performance comparison of these two previous models in precision, recall, and F1-score with the proposed approach, where the best performance is shown in bold.

Table 9: Comparison of the proposed approach with the state-of-the-art techniques

Data	NMFIAD			ML-KB+			Proposed approach		
	Precision (%)	Recall (%)	F1-score (%)	Precision (%)	Recall (%)	F1-score (%)	Precision (%)	Recall (%)	F1-score (%)
D1	50	**98**	66	70	74	72	**88**	86	**87**
D2	50	**98**	66	**93**	87	90	**93**	93	**93**
D3	49	**99**	65	71	74	72	**99**	99	**99**
D4	49	**97**	65	70	75	72	**94**	91	**93**
D5	50	**99**	67	82	84	83	**93**	93	**93**
AVG	50	**98**	66	77	79	78	**93**	92	**93**

NMFIAD has less precision and F1-score than the proposed approach, which achieved 42% more precision and 27% more F1-score. However, the proposed approach achieved 6% less recall than NMFIAD. ML-KB+ on average achieved 16% lower precision, 13% lower recall, and 15% lower F1-score than the proposed model.

5 Conclusion

In this paper, an aspect-based sentiment analysis technique is proposed for the movie application domain. The main focus is to develop an approach for mapping implicit aspects to explicit aspects. In addition, the proposed technique not only investigates implicit aspects within a sentence but also explores preceding and following sentences for the identification of implicit aspects using a multi-level knowledge engineering approach. The first step in implicit aspects identification is to identify explicit aspects using BiLSTM-CRF. The identified explicit aspects serve as input (memory) for the aspect mapping algorithm. Previously, state-of-the-art implicit aspect identification techniques have been developed for simple sentences. Those are based on the assumption that the implicit aspects are hidden between consecutive opinion words. However, while investigating the movie review dataset, four different types of implicit sentences are found. Exploring the implicit aspects of these different types of sentences is very challenging. Therefore, an implicit aspect mapping approach that can map the clues from different types of sentences to movie-specific aspects is proposed. The experimental results showed that the proposed mapping approach performs efficiently on the movie reviews dataset. Moreover, to validate the proposed approach, it is tested on a popular product reviews dataset and compared with state-of-the-art techniques. The experimental results show that the proposed approach outperforms the state-of-the-art methods.

Our ongoing work aims to improve the proposed model's results by better understanding its strengths and weaknesses. We also plan to evaluate the model in related tasks where noise and emerging NEs are prevalent.

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

[1] M. Niemimaa, J. Järveläinen, M. Heikkilä and J. Heikkilä, "Business continuity of business models: Evaluating the resilience of business models for contingencies," *International Journal of Information Management*, vol. 49, no. 268–4012, pp. 208–216, 2019.

[2] J. Kokina, D. Pachamanova and A. Corbett, "The role of data visualization and analytics in performance management: Guiding entrepreneurial growth decisions," *Journal of Accounting Education*, vol. 38, no. 748–5751, pp. 50–62, 2017.

[3] K. A. I. Hammad, M. Fakharaldien, J. Zain and M. Majid, "Big data analysis and storage," in *Int. Conf. on Operations Excellence and Service Engineering*, Orlando, Florida, USA, 2015.

[4] S. Khatoon, L. A. Romman and M. M. Hasan, "Domain independent automatic labeling system for large-scale social data using lexicon and web-based augmentation," *Information Technology and Control*, vol. 49, no. 1, pp. 36–54, 2020.

[5] J. Mir and M. Usman, "An effective model for aspect based opinion mining for social reviews," in *2015 Tenth Int. Conf. on Digital Information Management*, Jeju, Korea (South), 2015.

[6] H. H. Lek and D. C. C. Poo, "Automatic generation of an aspect and domain sensitive sentiment lexicon," *International Journal on Artificial Intelligence Tools*, vol. 23, no. 4, pp. 1460019, 2014.

[7] B. Liu, "Sentiment analysis and opinion mining," *Synthesis Lectures on Human Language Technologies*, vol. 5, no. 1, pp. 1–167, 2012.

[8] E. Marrese-Taylor, J. D. Velásquez and F. Bravo-Marquez, "A novel deterministic approach for aspect-based opinion mining in tourism products reviews," *Expert Systems with Applications*, vol. 41, no. 17, pp. 7764–7775, 2014.

[9] D. Anand and D. Naorem, "Semi-supervised aspect based sentiment analysis for movies using review filtering," *Procedia Computer Science*, vol. 84, pp. 86–93, 2016.

[10] V. Parkhe and B. Biswas, "Sentiment analysis of movie reviews: Finding most important movie aspects using driving factors," *Soft Computing*, vol. 20, no. 9, pp. 3373–3379, 2016.

[11] T. Chinsha and S. Joseph, "A syntactic approach for aspect based opinion mining," in *2015 IEEE Int. Conf. on Semantic Computing*, Anaheim, CA, USA, 2015.

[12] Y. Zhao, B. Qin and T. Liu, "Creating a fine-grained corpus for Chinese sentiment analysis," *IEEE Intelligent Systems*, vol. 30, no. 1, pp. 36–43, 2015.

[13] T. Ahmad and M. N. Doja, "Ranking system for opinion mining of features from review documents," *International Journal of Computer Science Issues*, vol. 9, no. 4, pp. 440–447, 2012.

[14] M. Dragoni, "A three-phase approach for exploiting opinion mining in computational advertising," *IEEE Intelligent Systems*, vol. 32, no. 3, pp. 21–27, 2017.

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

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

[17] W. Wang, H. Xu and W. Wan, "Implicit feature identification via hybrid association rule mining," *Expert Systems with Applications*, vol. 40, no. 9, pp. 3518–3531, 2013.

[18] 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*, Berlin, Heidelberg: Springer, 2013.

[19] F. L. Cruz, J. A. Troyano, F. EnrÃquez, F. J. Ortega and C. G. Vallejo, "'Long autonomy or long delay?' The importance of domain in opinion mining," *Expert Systems with Applications*, vol. 40, no. 8, pp. 3174–3184, 2013.

[20] X. Zhou, X. Wan and J. Xiao, "CMiner: Opinion extraction and summarization for Chinese microblogs," *IEEE Transactions on Knowledge and Data Engineering*, vol. 28, no. 7, pp. 1650–1663, 2016.

[21] M. Tubishat, N. Idris and M. A. Abushariah, "Implicit aspect extraction in sentiment analysis: Review, taxonomy, oppportunities, and open challenges," *Information Processing & Management*, vol. 54, no. 4, pp. 545–563, 2018.

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

[23] Q. Xu, L. Zhu, T. Dai, L. Guo and S. Cao, "Non-negative matrix factorization for implicit aspect identification," *Journal of Ambient Intelligence and Humanized Computing*, vol. 11, pp. 1–17, 2019.

[24] M. Hu and B. Liu, "Mining and summarizing customer reviews," in *Presented at the Proc. of the Tenth ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining*, Washington, USA, 2004.

[25] 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 Computing*, vol. 22, no. 3, pp. 5839–5857, 2019.

[26] J. Liao, S. Wang and D. Li, "Identification of fact-implied implicit sentiment based on multi-level semantic fused representation," *Knowledge-Based Systems*, vol. 165, pp. 197–207, 2019.

[27] M. Tubishat and N. Idris, "Explicit and implicit aspect extraction using whale optimization algorithm and hybrid approach," in *Presented at the Int. Conf. on Industrial Enterprise and System Engineering*, Indonesia, 2019.

[28] M. Afzaal, M. Usman and A. Fong, "Tourism mobile App with aspect-based sentiment classification framework for tourist reviews," *IEEE Transactions on Consumer Electronics*, vol. 65, no. 2, pp. 233–242, 2019.

[29] V. Gupta, V. K. Singh, P. Mukhija and U. Ghose, "Aspect-based sentiment analysis of mobile reviews," *Journal of Intelligent & Fuzzy Systems*, vol. 36, no. 5, pp. 4721–4730, 2019.

[30] M. Afzaal, M. Usman, A. C. Fong and S. Fong, "Multiaspect-based opinion classification model for tourist reviews," *Expert Systems*, vol. 36, no. 2, pp. e12371, 2019.

[31] M. Z. Asghar, A. Khan, S. R. Zahra, S. Ahmad and F. M. Kundi, "Aspect-based opinion mining framework using heuristic patterns," *Cluster Computing*, vol. 22, no. 3, pp. 7181–7199, 2019.

[32] H. El Hannach and M. Benkhalifa, "A New semantic relations-based hybrid approach for implicit aspect identification in sentiment analysis," *Journal of Information & Knowledge Management*, vol. 19, no. 3, pp. 2050019, 2020.

[33] S. Khalid, M. H. Aslam and M. T. Khan, "Opinion reason mining: Implicit aspects beyond implying aspects," in *Presented at the 21st Saudi Computer Society National Computer Conf. (NCC)*, Riyadh, Saudi Arabia, 2018.

[34] C. Yang, H. Zhang, B. Jiang and K. Li, "Aspect-based sentiment analysis with alternating coattention networks," *Information Processing & Management*, vol. 56, no. 3, pp. 463–478, 2019.

[35] N. Liu, B. Shen, Z. Zhang, Z. Zhang and K. Mi, "Attention-based sentiment reasoner for aspect-based sentiment analysis," *Human-Centric Computing and Information Sciences*, vol. 9, no. 1, pp. 35, 2019.

[36] F. Tang, L. Fu, B. Yao and W. Xu, "Aspect based fine-grained sentiment analysis for online reviews," *Information Sciences*, vol. 488, pp. 190–204, 2019.

[37] G. S. Chauhan, P. Agrawal and Y. K. Meena, "Aspect-based sentiment analysis of students' feedback to improve teaching–learning process," in *Information and Communication Technology for Intelligent Systems. Smart Innovation, Systems and Technologies*. vol. 107. Singapore: Springer, 2019.

[38] M. Shams, N. Khoshavi and A. Baraani-Dastjerdi, "LISA: Language-independent method for aspect-based sentiment analysis," *IEEE Access*, vol. 8, pp. 31034–31044, 2020.

[39] A. M. Jibran Mir, S. Khatoon, "Aspect Based classification model for social," *Engineering, Technology & Applied Science Research*, vol. 7, no. 6, pp. 2296–2302, 2017.

[40] J. Mir and A. Mahmood, "Movie aspects identification model for aspect based sentiment analysis," *Information Technology and Control*, vol. 49, no. 4, pp. 564–582, 2020.

[41] E. F. Sang and F. De Meulder, "Introduction to the CoNLL-2003 shared task: Language-independent named entity recognition," in *CONLL'03: Proc. of the Seventh Conf. on Natural Language Learning at HLT-NAACL*, Stroudsburg, PA, United States, Association for Computational Linguistics, vol. 4, 2003.

[42] G. Aguilar, S. Maharjan, A. P. López-Monroy and T. Solorio, "A multi-task approach for named entity recognition in social media data," in *Proc. of the 3rd Workshop on Noisy User-generated Text*, Denmark, Association for Computational Linguistics, pp. 148–153, 2019.

[43] J. Lafferty, A. McCallum and F. C. Pereira, "Conditional random fields: probabilistic models for segmenting and labeling sequence data," in *Proc. of the 18th Int. Conf. on Machine Learning*, San Francisco, USA, 2001.

[44] L. Chen, L. Qi and F. Wang, "Comparison of feature-level learning methods for mining online consumer reviews," *Expert Systems with Applications*, vol. 39, no. 10, pp. 9588–9601, 2012.

[45] W. Etaiwi, A. Awajan and D. Suleiman, "Statistical arabic name entity recognition approaches: A survey," *Procedia Computer Science*, vol. 113, no. 1877-0509, pp. 57–64, 2017.

[46] G. Lample, M. Ballesteros, S. Subramanian, K. Kawakami and C. Dyer, "Neural architectures for named entity recognition," in *Conf. of the North American Chapter of the Association for Computational Linguistics*, San Diego, 2016.

[47] F. Dernoncourt, J. Y. Lee and P. Szolovits, "NeuroNER: An easy-to-use program for named-entity recognition based on neural networks," in *Proc. of the EMNLP System Demonstrations*, Denmark, Association for Computational Linguistics, pp. 97–102, 2017.

[48] H. Do, K. Than and P. Larmande, "Evaluating named-entity recognition approaches in plant molecular biology," in *Int. Conf. on Multi-Disciplinary Trends in Artificial Intelligence*, Hanoi, Vietnam, 2018.

[49] K. Schouten, N. De Boer, T. Lam, M. Van Leeuwen, R. Van Luijk *et al.*, "Semantics-driven implicit aspect detection in consumer reviews," in *Proc. of the 24th Int. Conf. on World Wide Web*, lorence, Italy, 2015.

[50] R. Panchendrarajan, N. Ahamed, B. Murugaiah, P. Sivakumar, S. Ranathunga *et al.*, "Implicit aspect detection in restaurant reviews using cooccurence of words," in *Proc. of the 7th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis*, San Diego, California, USA, 2016.