# Implicit Feature Extraction from Customer Reviews Using Supervised Learning

Atheesan Sornalingam

179306F

M.Sc. in Computer Science

Department of Computer Science and Engineering

University of Moratuwa
Sri Lanka

March 2020

# Implicit Feature Extraction from Customer Reviews Using Supervised Learning

Atheesan Sornalingam

179306F

This dissertation submitted in partial fulfillment of the requirements for the Degree of M.Sc. in Computer Science specializing in Data Science, Engineering, and Analytics

Department of Computer Science and Engineering

University of Moratuwa
Sri Lanka

March 2020

# DECLARATION

I declare that this is my own work and this PG Diploma Project Report does not incorporate without acknowledgment any material previously submitted for a Degree or Diploma in any other University or institute of higher learning and to the best of my knowledge and belief it does not contain any material previously published or written by another person except where the acknowledgment is made in the text. Also, I hereby grant to University of Moratuwa the non-exclusive right to reproduce and distribute my thesis, in whole or in part in print, electronic or other medium. I retain the right to use this content in whole or part in future works.

…….. ……...

Atheesan Sornalingam Date

I certify that the declaration above by the candidate is true to the best of my knowledge and that this project report is acceptable for evaluation for the CS5999 PG Diploma Project.

…….. ……...

Dr. Uthayasanker Thayasivam Date

# ABSTRACT

Every online product selling applications are having review systems for their customers to review the products that they have purchased. Customers' reviews about the product will be either negative or positive and some reviews will give the meaning explicitly and some reviews will have implicit meaning. Nowadays most of the people do purchasing through online as a result there are thousands of reviews for a single product. On the other hand, these reviews will be useful for other customers to decide whether to purchase the product or not by going through the reviews. Mining implicit features from the customer reviews is a fundamental requirement for extracting customers' opinions and summarizing. This research focuses on extracting implicit features from reviews for opinion mining using a word embedding model. It removes noisy words and learn the model parameters automatically and extract the implicit features from customer reviews. Most of the existing researches have focused on implicit feature extraction from Chinese web reviews and only few attempts are made to extract implicit features from English web reviews. Implicit feature extraction was done through supervised, semi-supervised and unsupervised learning approaches. This research focuses on supervised aspect extraction using deep learning. This research proposes a novel and yet simple CNN model employing two types of pre-trained embeddings for aspect extraction: general-purpose embeddings and domain-specific embeddings associated with a Word Embedding based Correlation (WEC) model by integrating advantages of both the translation model and word embedding to extract implicit features. WEC model can score their correlation score for each word in review and feature. Then the CNN is used to identify the feature where the input for CNN is similarity matrix generated using the correlation scores. CNN gives the matching score of the review feature pair as the output and the review’s corresponding feature will be identified from the feature set.

## Acknowledgements

I would like to express profound gratitude to my advisor, Dr. Uthayasanker Thayasivam, for his invaluable support by providing relevant knowledge, materials, advice, supervision and useful suggestions throughout this research work. His expertise and continuous guidance enabled me to complete my work successfully.

I would like to thank all my colleagues for their help in finding relevant research material, sharing knowledge and experience and for their encouragement.

I am as ever, especially indebted to my parents for their love and support throughout my life. I also wish to thank my loving friends, who supported me throughout my work. Finally, I wish to express my gratitude to all my colleagues at my work place, for the support given to me to manage my MSc research work.

# Table of Contents

# LIST OF FIGURES

Page



# LIST OF ALGORITHMS

Page

# LIST OF TABLES

Page

# INTRODUCTION

## 1.1 Overview

Online shopping becomes the leading selling and buying place in the current world. Before purchasing from online we will go through the reviews available for the product to get an opinion about the quality and the features of the product. This is important because buyers are taking a risk of purchasing a new product which they have not used before or a product from a new seller who they do not know. Also, manufacturers can learn on which way they need to improve the product features to meet customer satisfaction by analyzing the reviews. So, it is important and helpful for the buyers as well as for the manufacturers to analyze the customers' reviews.

Online stores are a booming business today as there are many advantages of online shopping. Online shopping includes variety of things to select and purchase for the customers. There are lot of advantages on online shopping like, when doing online purchasing no need to go to the store, just need to select and pay for the goods online and they will be delivered to home. This saves time. Another advantage is, price comparison can be made easily. Search engine allows to easily compare and cross check product prices from different web sites. This allows to determine which online store offers the most affordable item for purchasing. Other advantage is we can purchase the reliable product by going through other customers' reviews about the product and its features.

Even though processing each and every feature is trivial, on identifying the reliable product customer may need to go through thousands of reviews to get an idea about the product's quality. Sometimes the customer may purchase the product because of its particular feature. For example, customer may buy iPhone because of its camera. In this case customer needs to identify the reviews which are about the iPhone camera. It is difficult to go through thousands of reviews from different websites to get an opinion about the product or about its specific feature. The purchaser may criticize or praise the product or its feature directly or indirectly in the review.

The feature of the product may express explicitly or implicitly in a review sentence. Explicit feature is the feature which appears as the noun or noun phrase of the sentence. On the other hand, implicit feature will not appear in the sentence, but it is implied in the sentence [15]. Table 1 shows the sample customer reviews which have explicit features and implicit features in the review sentences [6].

**Table 1:** Sample customer reviews on phone.

| Feature | Explicit Review | Implicit Review |
|---|---|---|
| Size | "I like the size of the phone, it's really small" | "The phone fits nicely into any pocket without falling out" |
| Battery life | "The battery life is excellent" | "You don't need to carry a charger with you anymore" |

Sentiment analysis and opinion mining help to process and analyze a large amount of customer reviews automatically. Opinion mining can be described by three different levels. They are document level, feature level and sentence level. In sentiment analysis; at document level, it

analyzes whether a particular document is positive or negative; emotional orientation of the whole review. In sentence level, it identifies whether a particular sentence is negative or positive or neutral. That differentiates objective sentences from subjective sentence opinions. However, both sentence level and document level analysis fail to identify exactly what do the customers like or dislike about the product and significant details are not discovered. Here the task is to mine implicit features from the customer review which will help to summarize the customers' opinions from the reviews. In order to solve this problem opinion mining at feature level is being used. Feature opinion mining extracts specific features and opinions from customer reviews.

### 1.2 Problem and Motivation

Buying products from online shopping websites is an increasing trend. After purchasing products from online customers post reviews about the products. This gives valuable information to other users and manufacturers as well. But for a famous product the number of reviews may grow rapidly, this makes the customers and the manufacturers difficult to go through each and every reviews to get an opinion about the product and its features. This is a time and effort consuming task. Also might require expertise knowledge in that particular domain. Automating the process of analyzing and summarizing the reviews makes this task easy. To analyze and summarize reviews, extracting the features mentioned in the reviews is a key task. These features may be expressed explicitly or implicitly in a review. Implicit features extraction from a sentence is significantly difficult compared to explicit features extraction. In this research implicit features extraction is focused.

Most of the researches focused on extracting explicit features from the online customer reviews, while only few researches have focused on extracting implicit features. Surprisingly as mentioned by Karmaker et al [7] most of the researches extracted the implicit features from Chinese web reviews and few researches have focused on extracting the implicit features from English online reviews.

Most of these existing methods tend to be depending on measures that are heuristically designed such as association rules and correlation counts, making them hard to generalize [3, 4, 7, 9, 10].

### 1.3 Research Objectives

The main objective is to extract the implicit feature from online customer reviews.

Since most of the existing approaches tend to be depend on measures that are heuristically designed like association rules and correlation counts. The research objective is to develop a supervised aspect extraction using deep learning. This research proposes a novel and yet simple CNN model employing two types of pre-trained embeddings for aspect extraction: general-purpose embeddings and domain-specific embeddings associated with a Word Embedding Correlation (WEC) model which integrates advantages of both the word embedding and translation model to extract implicit features. And to be able to effectively extract the implicit feature from customer reviews with the accuracy of above 80%.

### 1.4 Organization of the Thesis

The thesis is ordered as follows. General introduction of this research presented in chapter 1. Chapter 2 reviews the literature survey on implicit feature extraction. Chapter 3 explains the methodology used in this research to extract implicit features. Chapter 4 presents the experiment details of the work presented in the thesis. Chapter 5 evaluates the work presented in the thesis along with a discussion on the observed results. And Chapter 6 concludes the thesis with a look into future work.

# LITERATURE SURVEY

## 2.1 Overview

There are researches using different type of approaches to identify implicit features and aspects from online customer reviews. Researches used either supervised or semi-supervised or unsupervised learning approaches in implicit feature extraction. Different approaches used by researchers are discussed in detail in the following sub sections.

## 2.2 Unsupervised Learning

Unsupervised learning is a machine learning algorithm that draws conclusions from data sets that consist of input data without labeled responses. Under un-supervised methods co-occurrence association rule mining, hybrid association rule mining, point wise mutual information (PMI) method and statistical learning based on generative feature language models' approaches were used to mine implicit features. Karmaker et al [7] used an unsupervised statistical learning to mine the implicit features based on generative feature language models. Co-occurrence association rule mining approaches [3, 4, 7] and rule-based approaches [9, 10] were used to extract the implicit features from reviews. Based on semantic association analysis, Su et al. [2] introduced Point-wise Mutual Information (PMI) to identify implicit features.

### 2.2.1 Hybrid association rule mining approach

In Wang et al.'s [4] approach many association rules were mined using several complementary methods and this is called hybrid association rule mining. Here association rules were used to identify the implicit features. Feature cluster was used to collect explicit sentences of each feature and POS tagging and word segmentation was used in extraction of candidate features from explicit review sentences. Weight for candidate feature indicator is calculated using five types of collocation extraction algorithms. These algorithms include PMI, frequency, frequency-PMI, v2 (chi-square) test and t-test.

Hybrid association rules mining was used to find:

1. The degree of co-occurrence between the product features and the features indicators.
2. The relation between the candidate features indicators.
3. The dependency word of the product features.
4. The constrained topic model based on product features and previous rule set.

***Approach***

At first customer reviews related to a specific product were taken and feature set was extracted based on the algorithm, frequent item set and then some manual operations were applied after the preprocessing steps like POS tagging and word segmentation. Then based on the synonymy factor the feature words were clustered. As an example, the word "price's" feature cluster is {price, cost, selling price, price position} where all these given words are closely related with the word "price"

and these are explicit features. After that, the sentence which conditions these phrases or words were selected as the explicit feature sentence collection with all its related features. Using the least occurrence and POS tags, the indicators of corresponding candidate feature were taken from the explicit review sentences. Here other than opinion words, some frequent items and other words also selected as feature indicators.

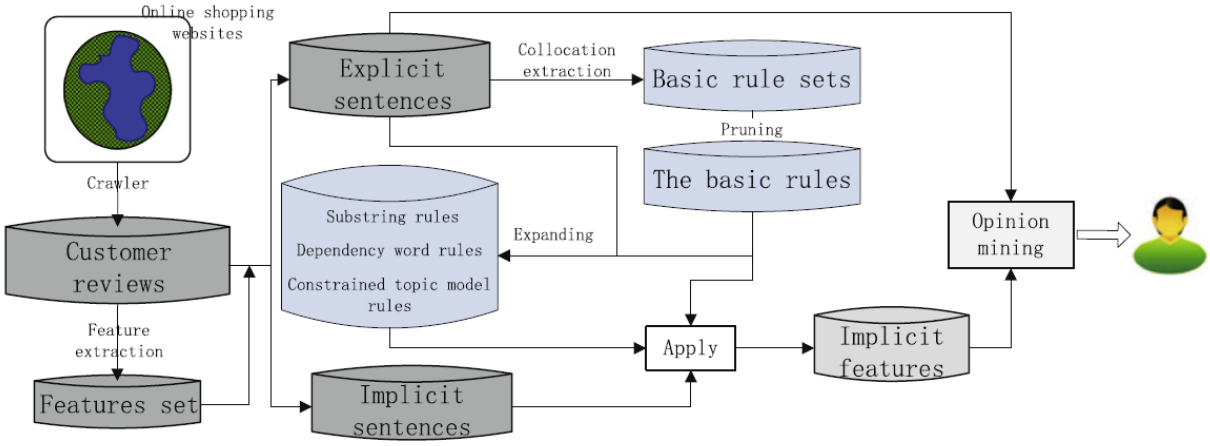



**Figure 1:** Framework of hybrid association rule mining for implicit feature identification.

Two words can be used at most to imply a product feature in Chinese language. So here considered only two-dimensional frequent items. For example, if we consider the review sentence, "The cell phone's screen is very exquisite, but needs to be charged every day", in this example, two dimensional frequent items are "day" and "charge". These two terms talk about the battery capacity when they appear in reviews of the product "cell phone".

A rule set was created by including the feature words whose indicator weight was greater than the mentioned threshold value. Among the five rule collections used, frequency-PMI method gave the best results in identifying implicit features. The best performing rules are made as basic rules which include indicators and the features that co-occur frequently.

After the preprocessing step, some features contain the same feature indicators which results in conflict. To resolve this a pruning algorithm was used to control the indicator that should be removed. Once the pruning is done for the indicators, some important rules ("feature indicator" ? "feature") were identified using a suitable threshold value. For example, the rule ("cheap" ? "price") indicates that if the word "cheap" appears in an implicit review sentence, it implies "price" is the feature.

Due to the shortages of the basic rules, current rules were expended using three approaches. The first method was retrieved from a hypothesis of substring. Then the second method adopted dependency grammar. And the last method was developed using semi-supervised learning which gets each feature's topic word collection using a constrained topic model. After that according to the given conditions, from the topic word set some reasonable rules are selected. Finally, implicit

features were identified by combining the basic rules and expanding rules together to create the final association rules. Most of the implicit features were identified from the basic rules as different collocation extraction algorithm's top five candidate feature indicators are very common and relatively reasonable.

***Algorithm***

```
1:  for each f_i in feature clusters do
2:    ES_i ← extract explicit sentence set from the corpus
3:  end for
4:  IS ← implicit sentence set with no feature
5:  for each f_i in feature clusters do
6:    CFI_i ← CollocationExtraction (ES_i)
7:  end for
8:  PR ← PruneConflictingRules (CFI, cut_threshold_m)
9:  for each f_i in feature clusters do
10:   BR_i ← BasicRulesSelect (CFI_i, cs_threshold_m)
11:   SSR_i ← SubStringRules (CFI_i, BR_i)
12:   DWR_i ← DependencyStructure (ES_i, CFI_i,
      dw_threshold_m)
13:   CTMR_i ← ConstrainedTopicModel (ES_i, CFI_i,
      ctm_threshold_m, PR);
14:   R_i ← BR_i + SSR_i + DWR_i + CTMR_i
15:   for each s in implicit sentence set IS do
16:     if the feature of s is null then
17:       for each r in R_i do
18:         if s satisfies r then
19:           the feature of s ← f_i
20:           break
21:         end if
22:       end for
23:     end if
24:   end for
25: end for
```

**Algorithm 2.1** Hybrid association rule mining

Identification of implicit aspects by using "hybrid association rule mining" is shown in Algorithm 2.1. Different parameters selection generated different experimental results. The word's least occurrence number and POS tag were used to filter all the candidate aspect indicators when choosing the candidate aspect indicators from the explicit aspect dataset. The min support of FP-tree was used to filter when the candidate feature indicator occurs frequently.

After each aspect's candidate aspect indicators was selected, some indicators will have more than one aspect and the algorithm for pruning was executed to prune the unrelated indicators (line 8). After that, several methods were used to identify rules from the selected candidate aspect indicators. Association rules were constituted using the aspect and the candidate aspect indicators which have higher weight than the threshold value. The best rules among the five rulesets were chosen as basic rules (line 10). A greater threshold can be used to filter out the association rules with lower frequency, this will increase the precision, but there are possibilities for recall to be reduced. If decrease the threshold value, then some unfitting feature indicators will appear. So, a greater threshold was used and achieved robust basic rules. Other appropriate rules were identified based on original aspect indicators and basic rules. Line no 11 shows the substring suppose, line no 12 shows dependency structure, and the line no 14 shows the constrained topic model. Finally, implicit features were identified using the state-of-the-art association rules which were combined with these rules (lines 14–24).

***Summary***

The researchers used a hybrid association rule mining to extract implicit aspects. Since the basic candidate rules are very common and reasonable, they have used the method frequency PMI to get good performance. Experiment was done with basic rules, substring rules and basic rules, dependency rules and basic rules, constrained topic model rules and basic rules and using all the rules. Since the basic rules are insufficient, they have used the three approaches of dependency rules, substring rules and constrained topic model rules to find some infrequent but sensible rules. As expected, without decreasing precision, the F-measure has been increased after integrating all of the rules. As a result, they have claimed that these rules of mining are reasonable as the basic rules and the hybrid association rule mining approach is an efficient method in extracting implicit aspects. At the same time more parameters are introduced in hybrid association rule mining approach and it is difficult to handle parameter tuning in practical application and different parameters produce different experimental results.

**Table 2:** Hybrid association rule mining: The highest performance of using all of the rules

| Rule set | Precision | Recall | F-measure |
|---|---|---|---|
| basic rules | 86.33% | 56.88% | 68.58% |
| all rules | 86.10% | 67.25% | 75.51% |

### 2.2.2 Generative Feature Language Models based approach

An unsupervised statistical learning approach was used by Karmaker et. al [6] to mine the implicit features based on generative feature language models. The parameters were optimized using Expectation-Maximization algorithm. Implicit feature mentions were identified through a new probabilistic method. Explicit feature mentions were used as the training data set. The researchers

have used hidden variables for representing sentence feature associations and the generative probabilistic model to model the review data. After defining these an iterative Expectation-Maximization approach was used in model parameters estimation. Implicit mentions were identified through the inferred values of hidden variables and model parameters. These techniques helped to avoid restrictions on opinion words and model parameters learning. At the same time a background model was used to handle the noisy words.

In this generative feature language model, a unigram language model which is known as a word distribution or a feature language model was used for vocabulary occurring modeling in sentences relating some features. This unigram language model will assign high probabilities to the frequently occurring words, in sentences that discuss a particular feature. For example, in a sentence which discusses about the "size" feature, the word "small" will get high probability and the word "service" will get small probability.

A mixture model word distribution was created using the sentences explicitly mentioning phrases describing a feature by mapping each word with a feature. A language model called background language model was created to model the noisy words which will assign high probabilities to the frequently occurring words like "a", "the", etc. This will help to identify noise in a sentence when using in mixture model.

***Approach***

The Generative Feature Language Model (GFLM) was created using generative mixture model with feature language models as components and the review data was created using this model.

This includes the following processes.

1. Generating each sentence by independently producing each of the sentence's words.

2. Decide on whether to generate the word in a sentence using the background model ($\gamma_B$) or a feature language model.

3. The word will be sampled from the distribution $p(w|\gamma_B)$ if the word is chosen from the background language model. Otherwise, need to choose on which k feature language models can be used, with the help of set of parameters $\{\pi_{S,i}\}$. $\pi_{S,i}$ gives the probability of when generating the word choosing feature language model $\gamma_i$. With probability $\pi_{S,i}$, the word using $p(w|\gamma_i)$ can sampled.

4. All the words in a sentence will be generated by repeating this process. Set of topic choice parameters $\pi_{S,i}$ based on sentence specific will be used to generate each sentence.

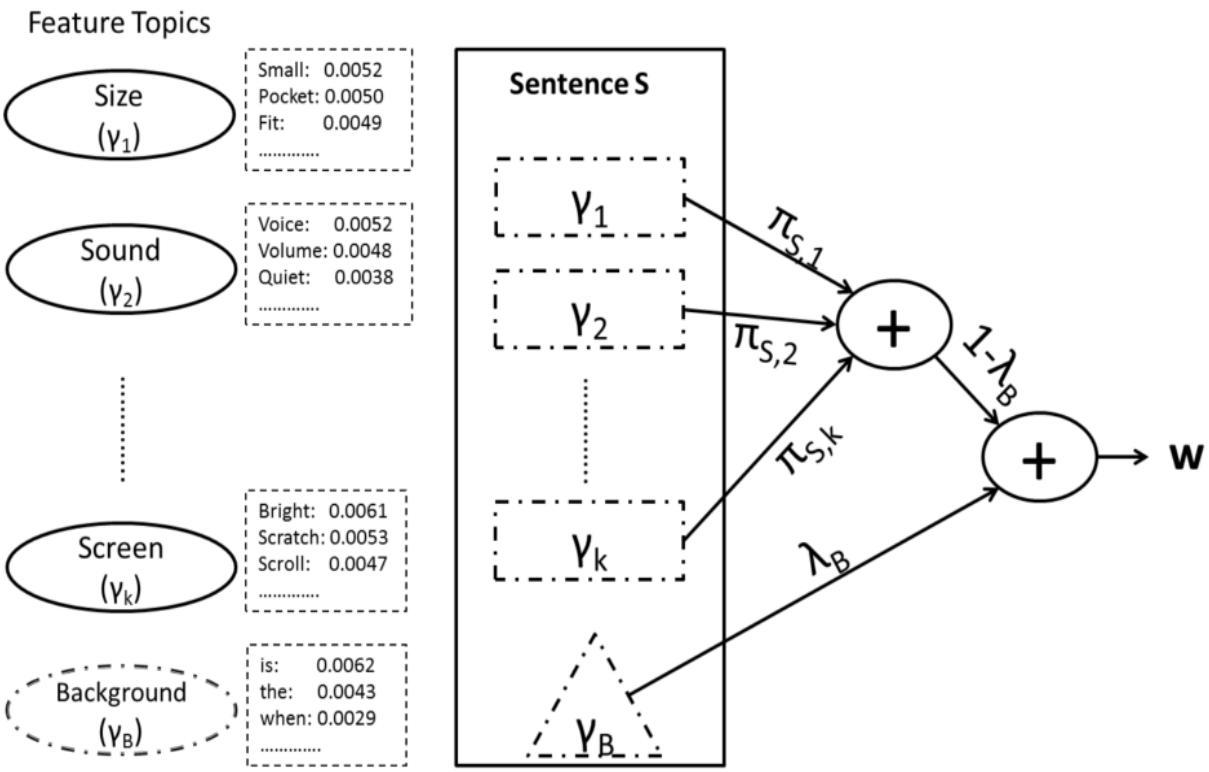



**Figure 2:** Schematic diagram: Generative model for a hypothetical Dataset

To estimate the parameters, introduced a hidden variable Z to estimate $\pi_{S,i}$ and $Z_{S,w}$ represents the identity of each word which is the distribution of feature topics. E.g.: "It's very light and holds charge for the whole day".

A pseudo training data was used to estimate all the unigram language models, it also includes the background model $\gamma_B$ and the feature language models $\gamma_i$. These were kept as constants when calculating the maximum likelihood estimation.

In E-Step, it estimated the hidden variables' distribution, estimated the uniqueness of each of the words, each word was divided into segments and the segments were generated through the feature topics. In maximization step, aggregated fractions to estimate the new $\Lambda$ and the new $\Lambda$ increase the log likelihood of the data.

To predict the implicit features, they have used two different ways of prediction. One is GFLM-Word (FM-W) which looks at feature topic distribution at the word level and the other method is GFLM-Sentence (FM-S) which looks at feature topic distribution at the sentence level.

***Summary***

Implicit feature mentions of online customer reviews are solved using a generative feature language model in a general and unsupervised method. The parameters are optimized automatically through EM algorithm and the research mainly based on statistical learning. At the same time the noisy words are filtered using a background language model.

The researchers have compared their methods, GFLM-Word (FM-W), and GFLM-Sentence (FM-S) with a supervised learning method which is Naive Bayes classifier (NB) to show how effective their method is for exploiting unlabeled data. And used current state of the art approach which is correlation-based method (CR) to compare their method's performance with the current state of the art method. The results are shown in the below table.

**Table 3:** Performance comparison of Generative Feature Language Model with baseline methods

| | F1 | | | | Corresponding Precision | | | | Corresponding Recall | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Dataset | NB | CR | FM-W | FM-S | NB | CR | FM-W | FM-S | NB | CR | FM-W | FM-S |
| Cellular phone1 | 0.2446 | 0.3092 | 0.4840$^{\dagger}$ | **0.4853**$^{\dagger}$ | 0.1818 | 0.2206 | 0.5487 | 0.6169 | 0.3736 | 0.5164 | 0.4329 | 0.4 |
| DVD player | 0.3147 | 0.3420 | **0.5125**$^{\dagger}$ | 0.5106$^{\dagger}$ | 0.2157 | 0.2268 | 0.6543 | 0.5559 | 0.581 | 0.6950 | 0.4212 | 0.4723 |
| Mp3 player1 | 0.2947 | 0.2671 | 0.5570$^{\dagger}$ | **0.5612**$^{\dagger}$ | 0.2406 | 0.2390 | 0.6397 | 0.6383 | 0.3800 | 0.3025 | 0.4933 | 0.5007 |
| Digital camera1 | 0.3312 | 0.3380 | **0.3831**$^{\ddagger}$ | 0.3808$^{\ddagger}$ | 0.3253 | 0.3870 | 0.5705 | 0.5504 | 0.3375 | 0.3 | 0.2886 | 0.2911 |
| Digital camera2 | 0.2177 | 0.2555 | 0.4439$^{\dagger}$ | **0.4559**$^{\dagger}$ | 0.1483 | 0.1619 | 0.5940 | 0.5303 | 0.4090 | 0.6060 | 0.3545 | 0.4 |
| Cellular phone2 | 0.4051 | 0.4134 | **0.597**$^{\dagger}$ | 0.5805$^{\dagger}$ | 0.3775 | 0.4156 | 0.6791 | 0.5795 | 0.4371 | 0.4112 | 0.5325 | 0.5816 |
| Mp3 player2 | 0.4604 | 0.4634 | **0.6666**$^{\dagger}$ | 0.6480$^{\dagger}$ | 0.4477 | 0.6495 | 0.6574 | 0.6195 | 0.4739 | 0.3601 | 0.6761 | 0.6793 |
| Router | 0.4805 | 0.5291 | **0.6686**$^{\dagger}$ | 0.6439$^{\dagger}$ | 0.4213 | 0.8009 | 0.7487 | 0.6339 | 0.5593 | 0.3951 | 0.6040 | 0.6543 |

### 2.2.3 Co-occurrence Association Rule Mining based approach

Zhang et al. [7] focused on product features and associations between the feature words and the notional words in a sentence.

Implicit feature extraction was done in four steps. In the first step a co-occurrence matrix(C) was determined by identifying all the notional words in the text corpus (D) and then co-occurrence frequency at clause level was recorded for each pair of notional words in a square matrix(C).

In the second step word modification matrix (M) was determined by using a bilateral iterative method. This determines the modification matrix (M) which includes the relationship between opinion words and their corresponding feature words in the same clause.

In the third step candidate feature word set Fc was obtained by identifying all the opinion words for a review (R) which did not has any explicit feature and formed the set Or. Then a candidate feature word set Fc was constituted by selecting all the feature words which can be modified using the opinion words available in Or.

In the final step implicit features were extracted. The commented features are related to the opinion words and correlated with the rest notional words in the review.

For example, if we consider the review sentence “No electricity after a few phone calls”, the feature word "battery” co-occurs with the words “no electricity” and “phone call”. In one sentence “battery” and “no electricity” coexist and in another “battery” and “phone call” coexist. The association between candidate feature words and the notional words in the review sentence was used to infer the implicit features. This method was evaluated using Chinese web reviews.

Hai et al. introduced two phase co-occurrence association rule mining approach to extract the implicit features from reviews [3].

In the first phase of rule generation, the researchers defined a form of [opinion word, explicit feature] from a co-occurrence matrix to mine a significant set of association rules for each opinion word occurring in an explicit sentence in the corpus.

In the second phase of rule application, more robust rules were generated for each opinion word mentioned above by clustering the rule consequents (explicit features). Following procedures were performed to identify a new opinion word with implicit feature. A search was performed to find the matched list of robust rules. Among them the rule which has the highest frequency weight feature cluster was selected and the representative word of the selected cluster is marked as the implicit feature. This research used Chinese review data for the evaluation.

### 2.2.4 Point wise Mutual Information method

Point wise Mutual Information (PMI) method considers the mutual information of the review to identify the implicit feature. Based on information theory PMI is an ideal measure of word association norms. PMI compares the probabilities of observing two items independently with the probability of observing two items together. As a result, it will identify whether the two words are genuinely associated or observed by chance. In Su et al.’s [2] method, the implicit feature was identified through the identification of opinion-oriented words and then the implicit product features in reviews were mapped by the adjectives.

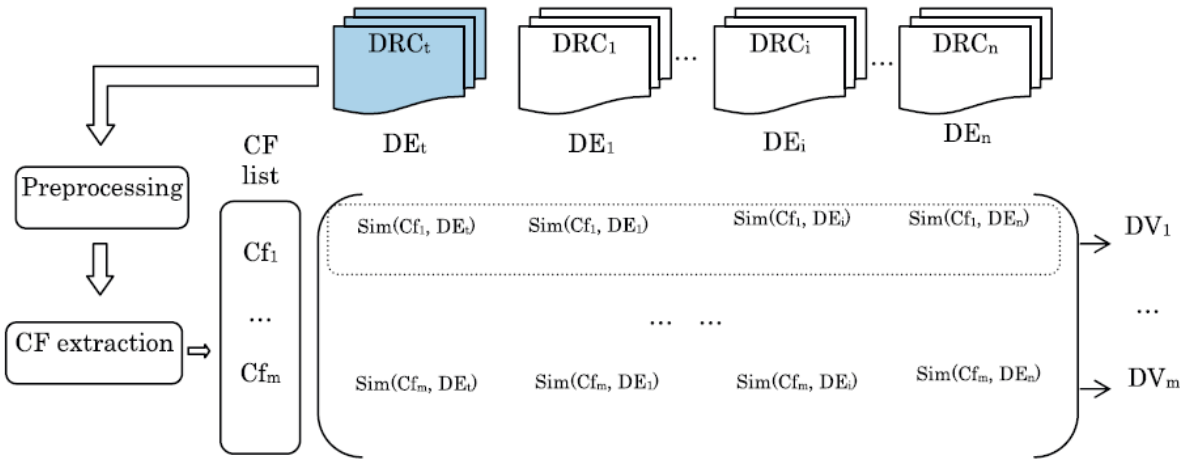
### 2.2.5 Feature-oriented opinion determination using unsupervised product feature extraction

Quan et al. [1] used a comparative domain corpus which consists of several product review sets to develop a feature extraction method. Here the domain-based product features were extracted through their weight’s evaluation in different related domains. Computation of association between features and domains were considered as the key point when extracting the features. The association of candidate features and domain words were evaluated based on a term similarity measure. A domain vector is derived for each candidate feature based on these similarities. Then the distances between the domain entity’s domain vector and the domain vectors of features were measured to extract the domain specific features.

Most of the body in the review domain, a domain he has his own domain feature, compared with the present-day in the actions of the associations with being closer to the body, the body is in another domain, one domain.

In a certain domain review corpus, a domain specific feature will be closely associated with the current corpus's domain entity than a comparative domain corpus's another domain entity. For an example, if we take the feature 'photo quality' from a camera review corpus, its semantic is closely associated with the domain entity 'camera' than 'mp3'.

The deriving process of domain vectors for candidate features is illustrated in Figure 3 and the symbols used in Figure 3 and their descriptions are listed in Table 2.




| Symbol | Descriptions |
|---|---|
| DRC | Domain review corpus: A review set of a certain domain<br>$DRC_t$: Domain review corpus that used for deriving domain-specific features<br>$DRC_1 \cdots DRC_n$: Comparative domain review corpora |
| DE | Domain entity: A term that represents a certain domain (e.g., camera, computer) |
| CF | Candidate feature: Candidate features that derived from *DRC* through the steps of phrase recognition and term extraction |
| DV | Domain vector: Each dimension of a domain vector records the similarity between a candidate feature and a domain entity $Sim(Cf_i, DE_j)$ |

**Figure 3:** The deriving process of domain vectors for candidate features

Multiple nouns string and nouns can be filtered during the pre-processing since domain specific features are mostly noun phrases and nouns. Noun phrase extraction, stop words filtration and name entity recognition were done in candidate feature extraction step. A domain entity was used to represent each domain review corpus. For example, camera for digital camera review corpus and phone for the corpus of cell phone review.

A feature and its domain are associated with each other based on the likeness between the feature and the domain entity term. The likeness was evaluated using PMI–TFIDF measurement which is a combination of point wise mutual information (PMI) and term frequency inverse document frequency (TF-IDF). Based on the similarity between the feature word and the domain entity term, for each candidate feature a domain vector is derived. The similarity represents its close

connection with the comparative domain corpora and the number of comparative domain corpora is represented by the dimension of a domain vector.

***Summary***

Feature oriented opinion is determined unsupervised product feature extraction method. It uses domain specificity of words as a form of domain knowledge. Feature and domain association is incorporated by the likeness between the feature and the domain entity term which is a representation of each domain review corpus. Below table shows the experimental results of domain specific features extraction on ten different classes of review. The results are retrieved by setting comparative domain review corpora while adding the corpus with the comparative domain corpora with same domain.

**Table 4:** Feature-oriented opinion determination: Best results for domain specific features extraction on D1 – D10

| Review domain | Data | Comparative domain corpora (threshold α) | Precision | Recall | F-score |
|---|---|---|---|---|---|
| Digital camera | D1 | D1–D7 (α = 5.0) | 0.755 | 0.906 | 0.824 |
| | D2 | D1–D7 (α = 5.0) | 0.810 | 0.889 | 0.848 |
| | D5 | D1–D2,D5 (α = 5.0) | 0.828 | 0.914 | 0.869 |
| | Avg. | | 0.798 | 0.903 | 0.847 |
| Cell phone | D3 | D1–D10 (α = 2.5) | 0.856 | 0.927 | 0.890 |
| | D10 | D1–D8,D10 (α=2.5) (α=2.5) | 0.878 | 0.933 | 0.905 |
| | Avg. | | 0.867 | 0.930 | 0.898 |
| Mp3 player | D4 | D1, D4 (α = 3.0) | 0.923 | 0.918 | 0.920 |
| | D7 | D1–D10 (α = 1.0) | 0.807 | 0.885 | 0.844 |
| | D9 | D9 (α = 1.0) | 0.853 | 0.895 | 0.873 |
| | Avg. | | 0.861 | 0.899 | 0.879 |
| Router | D6 | D1–D4, D6 (α = 0.3) | 0.848 | 0.691 | 0.761 |
| | D8 | D1–D10 (α = 0.5) | 0.737 | 0.673 | 0.704 |
| | Avg. | | 0.793 | 0.682 | 0.733 |

### 2.2.6 Rule-Based Approach

In Poria et al.'s [9] approach at first, the sentence dependency tree is obtained using the Stanford Dependency Parser3 and then using the means of Stanford Lemmatizer dependency structure elements are processed for each sentence. The dependency tree was built before lemmatization because the lower grammatical accuracy of lemmatized sentences may cause several imprecisions swapping the two steps.

***Approach***

In Implicit aspect lexicon creation first the sentences which are having implicit aspects are extracted and then for each sentence their corresponding labeled categories are considered and

implicit aspect clues (IACs) are extracted. As an example, in the review sentence "The car is expensive", it is labeled by the category price since the implicit aspect clue is expensive.

Two general rules are used to construct the aspect parser, the first rule set is for the sentences with subject verb and the second rule set is for the sentences without subject verb. Even though there are exceptions, most of the times if the active token acts as the head of the relation it will be considered in a relation. Depending on how matches the properties of the tokens with the rules and the dependency relation with the rules, several ways will be used to compute active token's contribution once it is identified as a rule's trigger. The correct way is to consider the combination of both the token's contribution along with the dependency relation of the other elements. At first the dependency parse structure of each sentence was obtained using the Stanford parser. And then to extract aspects the parse trees were used based on the hand-crafted dependency rules.

When a token's syntactic subject is the active token, if a subject noun relationship contains an active token "h" with a word "t" then,

1. If “t” is somehow determined to exist adjectival and adverbial determination in SenticNet, then t will be expressed as an aspect.
2. If the auxiliary verb is not in the sentence (i.e., was, is, could, would, should) then:
    - If the verb "t" is to be modified by the adjective or adverb or adverbial clause in relation with another token, then "t" and "h" will be extracted as features.
    - If there is any direct relation between a token "n" and "t" and if the token's POS is a noun and "n" does not exist in SenticNet, then "n" will be extracted as feature.
    - If there is any direct relation between a token "n" and "t" and if the token's POS is a noun and "n" does not exist in SenticNet, then "n" will be extracted as feature. If some other token "n1" is connected to "n" using any dependency relation in the sentence's dependency parse tree and if POS of "n1" is Noun, then "n1" will be extracted as feature.
    - If a token "t1" and "t" is having the relation of open clausal complement, and if "t-t1" exists in the opinion lexicon then the feature "t-t1" will be extracted. If "t1" and token "t2" are connected and if "t2"'s POS is a noun, then "t2" will be extracted as feature.
3. A copula is defined by the “copular verb’s” relationship with the "complement of a copular verb". If the token "t" and a "copular verb" are in a copula relation, and also if the implicit aspect lexicon contains the copular verb, then "t" will be extracted as feature.
4. If the token "t" and a "copular verb" are in a copula relation and if "h"'s POS is a noun, then "t" will be extracted as an explicit feature.
5. If the token "t" and a "copular verb" are in a copula relation and if any dependency relation exist between "copular verb" and token "t1" and if "t1" is a verb, then "t" and "t1" will be extracted as an implicit feature.

Features will be extracted using the following rules if the sentences are without subject noun relation in the parse trees:

1. If an adverb or adjective "h" is in open clausal complement relation or infinitival relation with the token "t" and the implicit feature lexicon contains "h", then "h" will be extracted as feature.
2. If a noun "t" and token "h" are having a prepositional relation, then "t" and "h" will be extracted as feature.
3. If a token "h" and token "t" are having a direct object relation, then "t" will be extracted as feature.

Additional Rules followed are:

- For all the features which are extracted above, if a token "t" and an aspect "h" are having conjunct relation or coordination relation, then "t" will be extracted as feature.
- Any noun that modifies the head noun is a "noun compound modifier" of an NP. If "t" is identified as a feature and "t" is having noun compound modifier "h", then the feature "h-t" will be extracted and "t" will be removed from the feature set.

***Summary***

This research has extracted both explicit and implicit aspects from reviews using unsupervised method focusing on the dependency structure and the commonsense knowledge of sentences. Developed an aspect knowledge base using SenticNet and WordNet to obtain the aspect categories of implicit aspect clues. Below table shows the results of the experiment carried out on Semeval 2014 aspect-based sentiment analysis data.

**Table 5:** Rule-Based Approach: Results of experiment on aspect-based sentiment analysis data (Semeval 2014)

| Domain | Precision | Recall |
|---|---|---|
| Laptop | 82.15% | 84.32% |
| Restaurants | 85.21% | 88.15% |

### 2.3 Semi-supervised learning approach

Semi-supervised learning is a class of supervised learning tasks and techniques that also make use of unlabeled data for training, typically a small amount of labeled data with a large amount of unlabeled data. Explicit topic mining model with SVM classifiers [5] and constrained k-means clustering with background knowledge [8] methods were used under the semi-supervised learning approach to extract implicit features.

### 2.3.1 Explicit topic mining model method

A semi-supervised learning approach was used by Xu et al. [5] for implicit feature identification on Chinese reviews by considering facts and opinions. For each of the product feature they have generated a classifier. The classification model was established using the sentences which are explicit, and their corresponding features extracted from the review as the training samples. Various vector space models (VSM) were built up using different collections of training attribute based on several traditional feature selection methods and different types of part-of-speech (POS) selection. For each features' training model, relevant explicit sentences are considered as the positive samples and relevant sentences which are non-explicit are as negative samples. After that sentences which are non-explicit were discriminated using some SVM classifiers.

***Approach***

Each sentence in the training data has a product feature and two features which will never co-occur. Therefore, an algorithm for clustering was used to get the relevant terms and the product feature in the same topic. Terms of the same topic were gathered into one group using the topic modeling approach. Same type of product features will be clustered into one cluster and the same cluster terms will have high association. As a result, the training attributes will be selected from these terms with higher priority. Also, clustering algorithms were created using these topic modeling methods. Once the terms are clustered into homogeneous topics and better and more meaningful clusters were produced using the pre-existing knowledge.

Although the basic LDA (Latent Dirichlet Allocation) has some limitations like obtaining the appropriate number of topics, product features co-occurring very often with some common words which are used to describe product features, words like good, nice, bad, etc. This will lead to cluster some product features in the wrong topic model when clustering the explicit sentences using basic LDA. So, it will become difficult to decide the topic cluster and its product feature which is the SVM’s classification category.

However, they have used an LDA based explicit topic model instead of traditional feature selection methods. Each topic will be pre-allocated with a certain product feature using the explicit topic model. Depends on each topic model’s explicit product feature, each topic cluster's corresponding category can be identified using the pre-explicit knowledge obtained from the explicit topic features.

In identifying implicit features following steps are involved. In the first step, from the explicit sentences two different types of constraints will be extracted. In the second step the explicit topic model will be incorporated with some relevance based prior knowledge and identified constraints. In the third step words will be filtered using the explicit topic models and each product feature’s training attributes will be extracted for SVM. In the final step selected attributes will be trained using several SVM classifiers and corresponding implicit features of the sentences will be identified.

***Algorithm***

Algorithm 2.2 shows the steps involved in extracting the implicit features. POS tagging, word segmentation and feature clustering are used when presenting a specific product and reviews to retrieve explicit sentences and their corresponding features (line 1).

```
1:   ES ← extract explicit sentence set
2:   NES ← non-explicit sentence set
3:   ML ← must-links from ES
4:   CL ← cannot-links from ES
5:   SPK ← syntactic prior knowledge from ES
6:   PPK ← PMI prior knowledge from ES
7:   ETM ← ExplicitTopicModel(T, ES, ML, CL, SPK, PPK)
8:   TA ← select training attributes from ETM
9:   for each f_i in feature clusters do
10:    TD_i ← GenerateTrainingData(TA_i, ES)
11:    C_i ← BuildClassificationModelBySVM(TD_i)
12:    PRS_i ← positive result sentences of Classify(C_i, NES)
13:    for each sentences s_i in NES do
14:      if s_i ∈ PRS_i then
15:        the feature of s_i ← f_i
16:      end if
17:    end for
18:  end for
```

**Algorithm 2.2** Implicit Feature Identification using explicit topic mining model and SVM

After the explicit sentences are identified, others are non-explicit sentences which contains implicit sentences as well as no feature sentences (line 2). From the explicit sentences, the constraint set, which is in the form of must links and cannot links is retrieved in the next step (line 3 and 4).

Furthermore, PMI prior knowledge and syntactic prior knowledge are extracted from the explicit sentences (line 5 and 6). All these pre-existing knowledges are extracted automatically and integrated into the explicit topic model so that it will enable to compare different experimental results (line 7).

Then via the pre-defined topic feature the word clusters are chosen as the training attributes for the classifiers of each product feature (line 8). Finally, according to the explicit sentence and the training attributes of fi, a corresponding SVM classifier is established for each feature fi (line 10

and 11) which is applied to classify the non-explicit sentences (line 12). For a non-explicit sentence si, an assumption is made that the sentence is an implicit sentence and contains the related feature fi if its classification result is positive (line 12 - 17).

***Summary***

The researchers have used an SVM based method to extract implicit features from Chinese customer reviews. An implicit topic model with preexisting knowledge is used to select the training attributes and handle the implicit features. Finally, implicit features are identified using the SVM classifier. They have compared their method with three other methods, which are baseline method which is using SVM and traditional method, co-occurrence association rule mining (CoAR) and point-wise mutual information (PMI). The comparison results show that their method, explicit topic model which incorporating all constraints and prior knowledge gives the best precision and recall.

**Table 6:** Explicit topic mining model: Best performance of different methods

| Methods | Precision (%) | Recall (%) | F-measure (%) |
|---|---|---|---|
| Baseline | 70.03 | 63.97 | 66.87 |
| PMI | 64.8 | 40.21 | 49.62 |
| CoAR | 69.25 | 56.04 | 61.95 |
| ETM + all constraints + all prior knowledge | 87.42 | 70.05 | 77.78 |

### 2.3.2 COP-KMeans Clustering based approach

Liu et al.'s [8] method identifies the implicit features and group the high similarity features into one cluster. Here feature level opinion mining is done to identify implicit features. This includes three steps, first is extracting the features and their corresponding opinion words. Second is clustering the features. Finally orient the features' opinions.

***Approach***

A. Extract Opinions and Features

The corresponding features are extracted using the opinion words which are modifiers. Each modifier is used to modify a feature without considering either the part of the entity or the whole entity. This method considers noun and noun phrases as well as verb and verb phrases. For example, it considers "running" as a feature.

In Chinese reviews mostly the wording pattern is like "the price is little expensive" rather than the pattern "high price". So, the left or right relationship (side of the features in a review relative to the opinion word) of the feature will be considered. If neither of the relationship is used, then it

assumes that there is an implicit feature in the review. Syntax analysis is not used for normative sentences because it is difficult to find the opinions and their corresponding features.

Since using opinions as the feature indicator is not good as it is ambiguous and produce wrong features. To solve this problem the relationship between the opinions and features is used. The words with low frequency may be the noises but they may consist of features and their corresponding opinions, so the noises are filtered mutually. In the repeated noise removal procedure, which is based on reversing the roles of opinions and features, the low confidence score opinions are selected and checks whether their corresponding features are also with low confidence scores. If the confidence score is low, then that opinion word is removed, and co-occurrence matrix will be recalculated.

B. Identify Implicit Features

Since there are two kinds of opinions, one is entity where the feature cannot be identified without the context (vague opinions). In this case the implicit feature is replaced with the entity. Other is feature where the opinions imply specific features which is context-independent (clear opinions).

The opinion words are grouped using the part of speech dictionary which includes the synonyms and the antonyms of words.

Implicit features get the candidate set as the explicit features which are modified by the opinion group. The representative word with the highest importance will be selected as the implicit feature.

C. Cluster Features

K-means algorithm is used to cluster the features with high similarity into groups as different words are used to express the same feature.

Three aspects of the features are considered. Feature's corresponding opinion similarity is one of the aspects where the clusters can be created using the similarity of the opinion words. The similarity of the corresponding opinion words are calculated by utilizing co-occurrence matrix and their type. Figure 2 shows a sample co-occurrence matrix.

Second is the similarity of the features in a text. The features which include the same word is considered in the aspect. For example, "speed" and "running speed" both represent the same feature speed. Set theory is used to calculate the similarity in the aspect.

Third is the structure of the features in a comment. Two indexes are considered in the aspect. Type of the features is one of the index. The feature may be the noun or noun phrase and verb or verb phrase of the review. In this method five types are considered. Which are noun, noun + verb, verb, verb + noun, noun + noun. Location of the features is the other index. Here the similarity is expressed by the cosine distance.

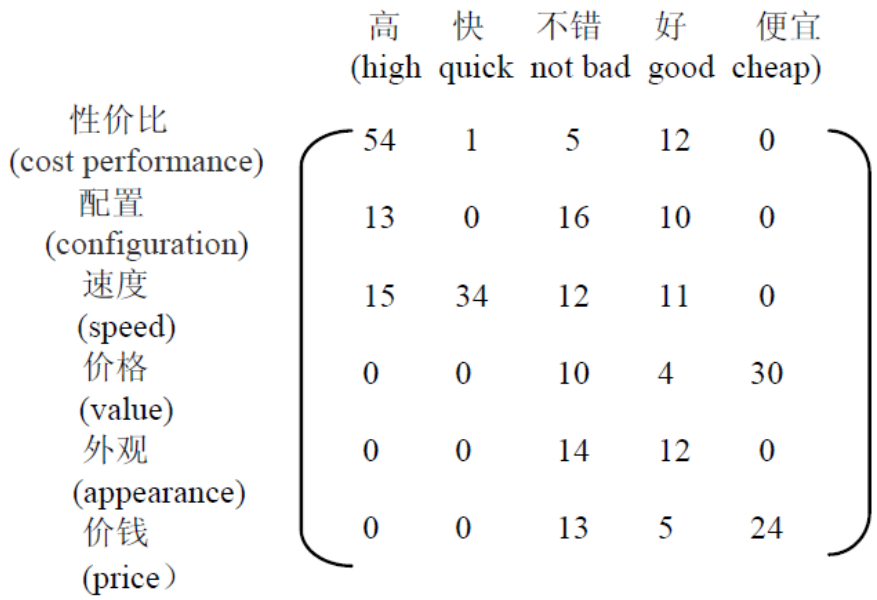



**Figure 4:** Similarity co-occurrence matrix

D. Clustering Enhancement

Clustering process is conducted by utilizing the constructed instance representation. Clustering enhancement is done using the COP-KMeans [12] which is a semi-supervised variant of K-Means.

Clustering process partition is generated using the background knowledge provided in the form of constraints between data objects. One of the constraints here is the incompatibility which is the same cluster cannot have two data objects. Constraints are constructed using the context-dependent information also. And an assumption is made that the same feature will not repeat in one review. Along with these the approach used the incompatibility to enhance the cluster.

***Summary***

This research uses the corresponding opinion words to extract the implicit features and then according to the confidence scores and mutual support scores it filters the noises. After that based on the knowledge of the context dependent information the features are clustered. Clustering enhancement is done using the COP-KMeans which is a semi-supervised variant of K-Means. A limitation in this research is the proposed method will not perform well if it is evaluated using a small-scale corpus. The results of the research are showed in the below table. Precision and recall for identifying the implicit features are calculated based on Manual annotation results. The effectiveness of the enhancement based on the context dependent information is showed in the below table. "K-Means" represent the pure K-Means algorithm and "Enhance" represent the K-Means based on the knowledge which is used in the research. The results show that the K-Means

based on the knowledge performs well. Also, it shows that the context-dependent information is a good indicator for clustering the features.

**Table 7:** Opinion Mining Using Clustering: Results of implicit feature identification

| Date sets | Computer1 | Computer2 | Phone | Camera |
|---|---|---|---|---|
| Precision | 0.65 | 0.72 | 0.79 | 0.74 |
| Recall | 0.56 | 0.67 | 0.70 | 0.65 |

**Table 8:** Opinion Mining Using Clustering: Results of comparison

| Date sets | | Computer1 | Computer2 | Phone | Camera |
|---|---|---|---|---|---|
| K | | **23** | **33** | **25** | **35** |
| Precision | K-Means | 0.54 | 0.64 | 0.63 | 0.72 |
| | Enhance | **0.65** | **0.70** | **0.67** | **0.79** |
| Recall | K-Means | 0.43 | 0.52 | 0.58 | 0.55 |
| | Enhance | **0.53** | **0.62** | **0.65** | **0.64** |

### 2.4 Supervised learning

Supervised learning infers a function from labeled training data using the machine learning task. Conditional Random Field (CRF) [11] is a supervised learning approach used for implicit feature extraction.

#### 2.4.1 A Classification based Approach for Implicit Feature Identification

Implicit features were extracted from product reviews through feature-level opinion mining. The products are represented using a set P = {P1, P2, P3, ..., Pn}. For each product Pi, customer

reviews are available, this is represented by Ri = {r1, r2, r3, ..., rm}. These reviews are considered as text documents. A sequence of sentences rj = {s1, s2, s3, ..., sl} is used to represent each review rj and each sentence sk may have several clauses sk = {c1, c2, c3, ..., ch}.

***Approach***

Zeng et al. [10] used three steps to extract implicit features, first step is extracting explicit opinion feature pair, second is constructing training document using opinion feature pair and third is identifying implicit feature.

1. Explicit feature-opinion pair Extraction

Opinion feature pairs were extracted from reviews using a rule-based approach. The mentioned method made use of grammar of Chinese dependency to identify opinion feature pairs. At first several rules were created using Chinese dependency grammar. Then making use of these rules' candidate opinion feature pairs were extracted. For each product candidate opinion word set CO and candidate feature word set CF was constructed achieve good precision in opinion feature pair extraction. Extracted the nouns and the adjectives from reviews and considered noun words as candidate feature and adjective words as candidate opinion with the assumption of nouns are possible to be feature words and adjectives are expected to be opinion words. From the candidate set CF and CO irrelevant nouns and adjectives are filtered using a stop word list. At the same time some of frequently used the non-noun feature words and non-adjective opinion words are supplemented.

In Chinese review sentences most features are in the structure of either DE (Chinese structural pattern) or subject-predicate (SBV). Therefore, to extract feature opinion pairs two different kinds of dependency relation were used by means of rules. From the observations it is identified that the feature satisfies the relation of SBV with opinion word when the feature appears before the opinion word and there is a DE structure between the opinion and the feature when the feature appears after the opinion word. To extract the explicit opinion feature pairs, three different rules are defined by the authors to handle different sentence structure types based on the above observations.

Rule 1: The dependency structure of a dependency relation SBV is denoted as sbv(w1, w2), where the word w2 is depending on word w1 in SBV, when word w2 fits to the opinion set CO and word w1 fits to feature set CF, then <w1, w2> will be extracted as opinion feature pair [10].

Rule 2: The dependency structure of a dependency relation SBV is denoted as sbv(w1, w2), where the word w2 is depending on word w1 in SBV, when word w2 does not fits to the opinion set CO and word w1 fits to feature set CF, and there is a word w3 which comes after word W2 that fits to the opinion set CO, then <w1, w3> will be identified as opinion feature pair [10].

Rule 3: In DE which is a dependency relation, where a word w2 fits to the feature set CF after a defined Chinese word and the word w1 that fits to the opinion set CO which appears before a defined Chinese word, then <w2, w1> will be identified as opinion feature pair [10].

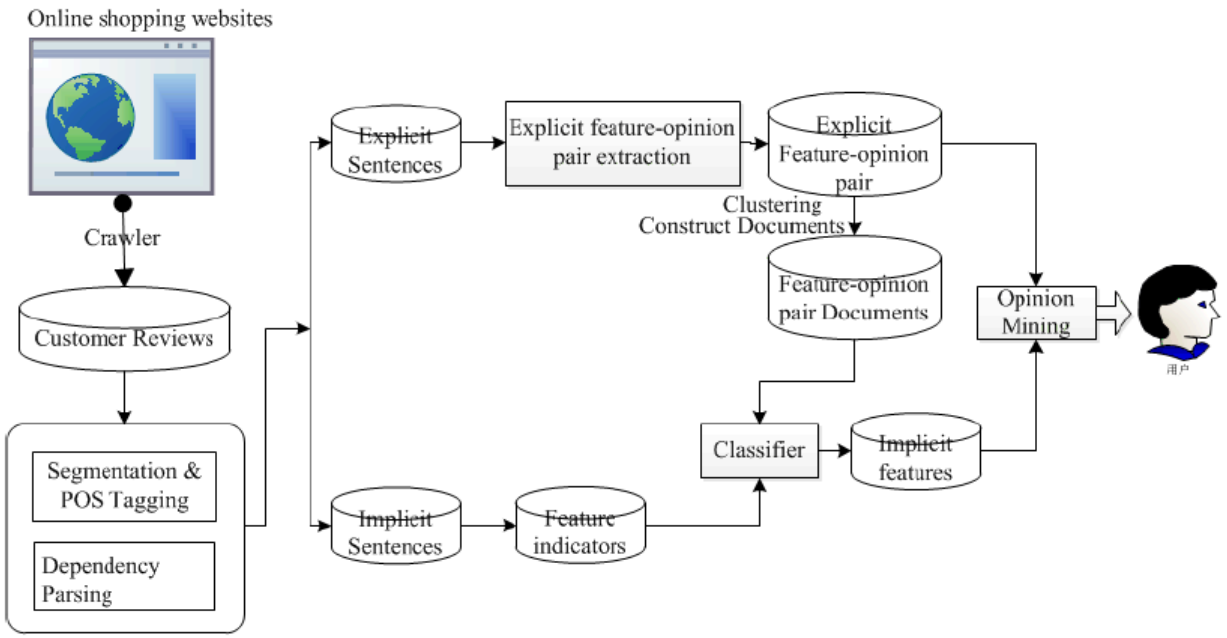



**Figure 5:** Framework of classification-based approach.

Customer reviews are processed using the word segmentation, dependency parsing and part-of-speech (POS) tagging. A sentence's satisfaction of a rule is identified based on the results of the preprocessing.

2. Feature opinion pair Training Document Construction

Identified explicit sentences are used as a training data and the category or the topic of each sentence is given the label of the sentence's opinion feature pair. For the explicit sentence "Delivery is very fast, ordered in the morning and received in the afternoon" the opinion feature pair (<fast, delivering>) is the labeled topic of the sentence. If more than one opinion feature pair exist in a sentence, then that sentence will be classified into each opinion feature pair topic of the sentence sk's opinion feature pair set (FOk).

Usually, for all opinion word, the opinion word is contained in more than one opinion-feature pair. If an opinion feature pair is <f, o>, then the feature word f is described by the opinion word o. Several different features can be described by an opinion word. For example, different kind of product features like "screen", "mobile phone" and "quality" can be described using the opinion word "good". Same feature can be expressed using different feature words or phrases in reviews. For example, the feature "vocality" can be expressed using the features "vocality quality", "music" and "sound effect". So, each opinion word o, they have clustered the opinion feature pairs FO(o) = {<f1, o>, <f2, o>... <fn, o>} which covers opinion term based on the semantical and also conceptual relation of the features F(o) = {f1, f2, ..., fn}. When comparing the feature set F(o)'s size with the whole features set F it is relatively small, so it became effective and easier to cluster the opinion feature pairs. After that, the training document for each clustered opinion feature pair

was constructed using the set of clustered opinion feature pair. And the sentences that contain the opinion feature pair were collected into a document and they were labeled using the clustered opinion feature pair for each clustered opinion feature pair.

3. Implicit Feature Identification

Identifying the implicit features is expressed into a classification of text problem by constructing the training set for opinion feature pair. For each implicit sentence Isk their corresponding opinion word set is denoted as Iok = {o1, o2, ..., on}. Identified implicit features by finding the implicit feature “fi” for each opinion term “oi” in “Io”. The clustered opinion feature pair set, which contains opinion term oi are denoted as FOc(oi) = {<fc1, oi>, <fc2, oi>, ..., <fcm, oi>}. The implicit feature “fci” finding problem for opinion term oi in implicit feature sentences Isk is transformed into a classification of text problem since the feature opinion pair is regarded as the sentence's topic or category.

A topic feature centroid classifier is designed to classify the implicit feature sentence with an opinion term oi into the most probable opinion feature pair <fi, oi> topic. The lexicon set is constructed using a small feature set related distinguished words in the training dataset. In training dataset of opinion feature pair only small amount of words contributed to the topic’s feature domain. Other words, like stop words occurs more frequently, and they do not have influence in judgement for the topic. Also, noise can be introduced because of these irrelevant words in the topic representation. Therefore, the nouns, verbs and adjectives in the training dataset is used in the construction of the lexicon set. Irrelevant words and stop words are removed using a filter word set. Denoted the lexicon set as L = {wf1, wf2, ..., wfL}, and the centroid for category <fj, oj> is denoted by a “word vector centroid” j = {wf1j, wf2j, ..., wfLj}, where wfkj(1 <= k <= L) is the word wfk’s weight. The weight calculation for word wfk in the topic feature centroid classifier was derived using a different formulation. The opinion feature topic was produced with more discriminative features using the weight calculation method. A denormalized cosine measure is used to classify implicit feature sentence after obtaining the centroid vector for each category.

$$C' = arg\max_{j}(\overrightarrow{s_i} \bullet \overrightarrow{Centroid_j})$$

Here, for the implicit sentence “Isk”, “si” is the word vector representation. Since, usually the sentence is short, concern was only whether the word is appeared or not appeared in the sentence. The capability of discrimination of opinion feature pair topic's centroid vector is preserved by using the de-normalized cosine measure. The de-normalized measure is more discriminative for the classification since the vector space size is comparatively small.

***Algorithm***

The process for identifying implicit feature is shown in algorithm 2.3.

**Input:** The set of implicit sentences.
**Output:** A set of implicit feature-opinion pair.

1: **for** each implicit sentence $\mathcal{I}s_k$ **do**
2: **for** each opinion word $o_i$ in $\mathcal{I}o$ **do**
3: apply the topic-feature-centroid classifier for $o_i$
4: get the implicit feature $fc_i$
5: **end for**
6: **end for**
7: return the set of implicit feature-opinion pair.

**Algorithm 2.3** Classification based Approach: Implicit Feature Identification

***Summary***

This research uses a classification-based approach to extract implicit features. The training dataset is labeled by using specific opinion feature pair cluster is obtained by constructing a dataset for the opinion feature pair cluster. Then identification of implicit feature task is formulated as a classification of text task. Explicit opinion feature pairs are extracted using a rule-based method from the customer reviews. A feature topic centroid classifier is used to classify the implicit feature. Although, there are some undesirable errors exists in identification of implicit features, which are due to incorrect classification and by wrong identification of implicit feature indicators, comparing coAR with the rule-based approach, the limitations of the rule-based approach were overcame by the proposed approach and achieved a better performance in comparison. The results are shown in the below table.

**Table 9:** Classification based Approach: Results of implicit feature identification

| Data Sets | Our Approach | | | CoAR | | |
|---|---|---|---|---|---|---|
| | Precision | Recall | F-measure | Precision | Recall | F-measure |
| Cell Phone | 82.07% | 68.48% | 74.66% | 67.88% | 52.93% | 59.48% |
| Digital Camera | 85.59% | 72.93% | 78.76% | 79.94% | 66.91% | 72.85% |

### 2.4.2 A Proposed framework for improved identification of implicit aspects using supervised learning technique

Bhatnagar et al. [11] used a supervised machine learning technique CRF (Conditional Random Field) to identify implicit aspects. Conditional random field technique predicts the probable outcome based on some previously given training on some desired data sets.

As a first step review data is collected, noise removed and converted into text document form. Each sentence is tokenized to input each word to the POS-tagging process. For further processing

each word is given a label and this label is its identity. POS-tagging is done to identify nouns and adjectives. Since explicit aspects are the frequently occurring nouns in the document. If the words that are tagged as nouns occur more than a threshold value, then they are counted as explicit aspects. Hidden aspects are found by deploying the trained CRF file on whole document. All the aspects are collected and provided with scores with the help of “SentiWordNet”. Found aspects' positions are marked in the document and then the adjectives associated to each aspect are found. Then all adjectives related to a particular aspect get their scores summed up after which the aspects get its final score of being rated as to what extent they have a positive impact on the review.

### 2.4.3 Using deep learning for implicit aspect extraction

Panchendrarajan et. al [39] have considered the situation where an opinion word is used to describe different aspects. This paper presented a method to recognize multiple implicit aspects in a sentence. Separate models are created for identification of implicit and explicit aspects. These models used manually labelled training data set. First model uses maximum entropy classification technique for explicit aspect identification. Second model finds opinion words and their associated implicit aspects. Double propagation is used to extract opinion target. To improve the accuracy in presence of multiple interrelated aspects, entities and their aspects are modelled as a hierarchy. Since they have focused on domain specific model, which is the restaurant domain, the model can be improved and extended to other domains by identifying relationships between domain specific aspects and modelling them as a hierarchy.

Feng et. al [40] built a feature vector by aggregating words vectors, part of speech vectors, and dependent syntax vectors extracted from the words is given as input for training the deep convolution neural network, and sequential algorithm is then used for finding the sentiment expressed in the sentence. To identify the implicit aspect Feng et. al [40] created an evaluation tuple [Ai, Fi, Ci, Oi] which is comprised by four elements, aspect, sentiment shifter, sentiment intensity and sentiment after they obtain the sentiment label of each word. The aspect is the implicit aspect when they extract the tuple Ai. They have proposed the following algorithm to identify the implicit aspect.

In step one, removed stop words, count the times of the words and the explicit aspects appearing in the explicit aspect sentence and arrange the statistics into a matrix, where the columns correspond to the explicit feature words and the rows correspond to the words in the sentence. In step two, obtained the matrix with number of other words that co-occurrence of the explicit aspect word in explicit aspect sentence.

In step three, calculated the times of the explicit aspect word and the sentiment word appearing simultaneously in the explicit evaluation tuple and build the matrix, where the columns correspond to the aspect word and the rows correspond to the sentiment word. In step four, reformulated matrix as another matrix, where the probability of the aspect word and the sentiment word appearing simultaneously.

In step five, if the sentence is the non-explicit aspect sentence, removed the stop words. The sentence consists of t words, i.e., S = {w1, w2 ⋯ wt}. The sentiment word without corresponding feature word is identified, the probability of which and candidate aspect words, where all the explicit aspect are candidates. All explicit aspect words are in all the explicit aspect sentences, and all the explicit aspect words are formed into a set, which are candidate aspect words. The declaration of other words does not include sentiment words, because the matching of sentiment words and aspect words has been calculated. The score for candidate is the sum of the topic score and the matching score. Then, they selected the candidates such that max as the implicit aspect of sentiment word.

In step six, suppose that the sentence is a continuous aspect sentence, the explicit aspect appearing before the implicit aspect is Ai, and the sentiment word of the implicit aspect is Oj. Oj match with Ai if the frequency of the co-occurrence of explicit aspect word Ai and sentiment word Oj is larger than the threshold β. Otherwise, return to step five for implicit aspect identification. In step seven, if the implicit aspect after the continuous aspect sentence, the recognized implicit aspect is treated as the explicit aspect. Then, they repeated step six to identify the implicit aspect.

Since they have mainly focused on mobile phone reviews; it is difficult to generalize and cannot ensure that their algorithm works well for another domain. And getting the appropriate model parameters also not mentioned clearly. They have obtained precision of 0.8758, recall of 0.7769 and F1 score of 0.8233.

**Table 10:** Summary: Methods used to identify implicit features

| Method | F-Score | Domain | Limitation |
|---|---|---|---|
| Hybrid association rule mining [4] | 75.51% | Chinese reviews | Different and more parameter tuning |
| Generative Feature Language Models [7] | 53.21% | English reviews | Insufficient data |
| Co-occurrence Association Rule Mining [3] | 61.30% | Chinese reviews | Associations between feature words and the rest of the factual/ notional words are ignored |
| Point wise Mutual Information [2] | 49.62% | Chinese reviews | Only the opinion words are considered, and all other factual/ notional words are discarded |
| Comparative domain corpora [1] | 73.30% | Chinese reviews | Insufficient feature-oriented opinion lexicons generation |
| Rule-Based Approach [9] | 84.47% | Chinese reviews | Define rules manually |
| Explicit topic mining model [5] | 77.78% | Chinese reviews | Prior knowledge needed to achieve better results |
| COP-KMeans clustering [8] | 68.71% | Chinese reviews | Low performance for small scale corpora |

| | | | |
|---|---|---|---|
| Classification based approach [10] | 76.56% | Chinese reviews | Errors due to incorrect classification and wrong implicit feature indicators identification |
| Word vector-based approach [40] | 82.33% | Chinese reviews | Focused on mobile phone reviews only.<br>Difficulty in getting the appropriate model parameters. |

### 2.5 Summary

This chapter presented the literature survey for different approaches used for implicit feature extraction. Major techniques discussed in this chapter are hybrid association rule mining, co-occurrence association rule mining, generative feature language model, point wise mutual information, feature-oriented opinion determination, rule-based approach, explicit topic mining model, COP-KMeans clustering, classification based approach and word vector-based approach.

# METHODOLOGY

## 3.1 Overview

This research focuses on supervised aspect extraction using deep learning. This research proposes a novel and yet simple CNN model employing two types of pre-trained embeddings for aspect extraction: general-purpose embeddings and domain-specific embeddings associated with a Word Embedding based Correlation (WEC) model that combines the advantages of translation model and word embedding model to find out the implicit features. For an arbitrary pair of words, co-occurrence probability in review and feature are score by WEC. At the same time, smoothness and continuity of continuous space word representation can also leveraged by WEC which help to deal with unfamiliar word pairs from the training parallel corpora. Better results can be achieved by removing the noisy words properly from the review sentences since many researchers have struggled removing noisy words properly. The system architecture is shown as a block diagram in figure 6.

## 3.2 Background

Implicit features are extracted with the help of word embedding based correlation model, general purpose embeddings, domain specific embeddings and convolutional neural networks. These features are discussed in detail in the following sub sections.

### 3.2.1 Word embedding

In deep learning applications, word embedding is broadly utilized in predictive NLP modeling for feature identification. Sparse vector representation of word can be transformed into dense vector representation with the use of word embedding. Comparability between phrases and words on a huge scale, depend on the context, can be found out using continuous vector space.

Word embedding transforms the words in a vocabulary into dense vectors of real numbers in a continuous embedding space. In classic NLP systems, words are represented as indices in a vocabulary which do not focus on the semantic connection among words. Word embedding is learned by neural networks explicitly encode distributional semantics in learned word vectors. Moreover, through low-dimensional matrix operations, word embeddings can be used to efficiently compute the semantics of larger text units such phrases, sentences and documents.

Similarities among words are not considered in most of the topical NLP works, while they focus words as atomic units just as these are represented as indices. There are several advantages in choosing this option such as robustness, simplicity and the observations which show that simple model trained on mass data outplay than the complex systems which are trained using less data. As an example, we can consider famous n-gram model which is used to build statistical language model.

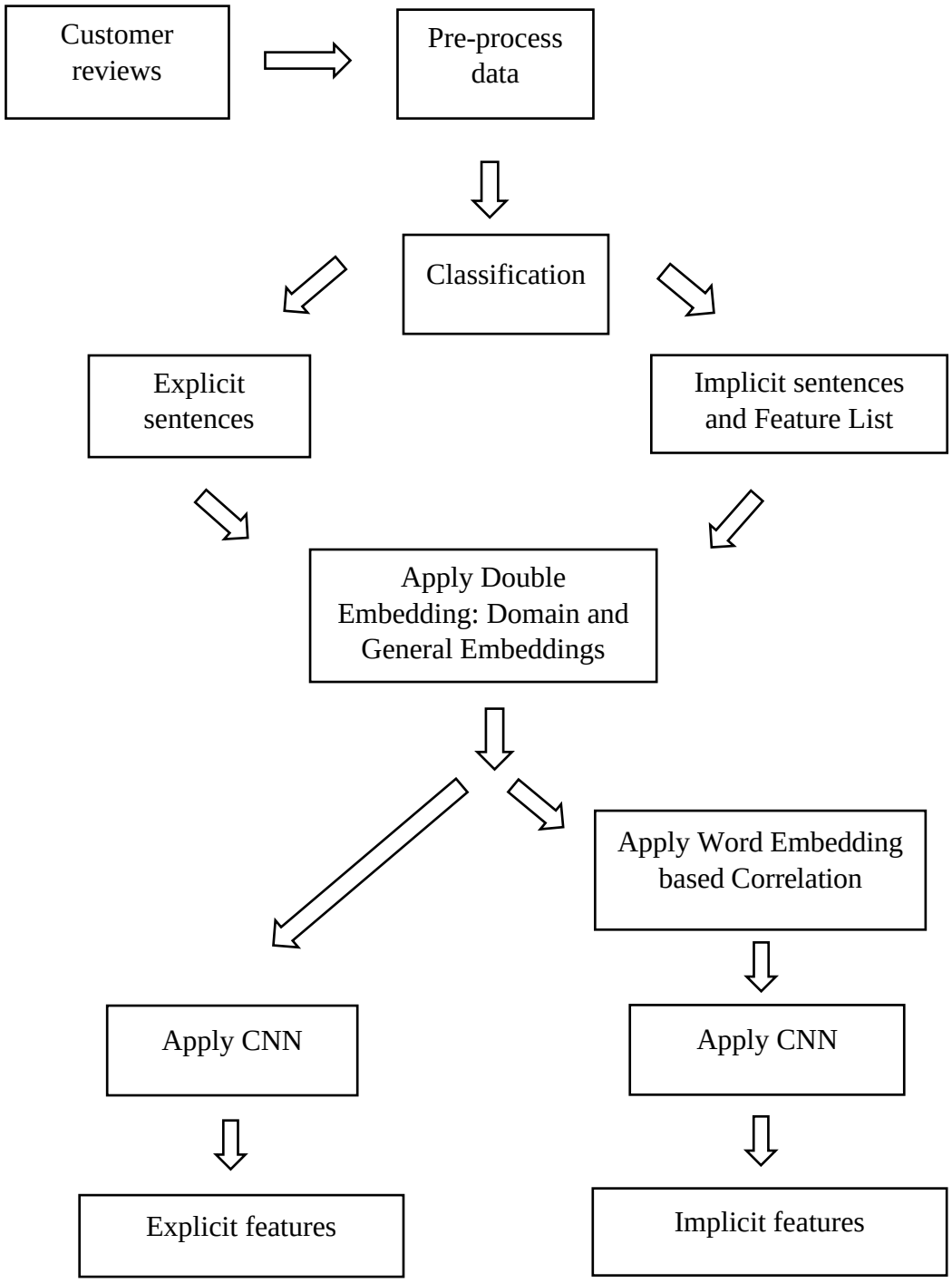



**Figure 6:** Conceptual block diagram of the proposed system architecture.

Anyhow, simple techniques could not achieve well due to the own limitations. As an example, speech recognition system is limited by the volume of relevant in-domain data. The volume of the good quality data determines the excellency of the speech recognition system. Current corpora in machine translation systems have only a petty amount (few billions) of words for many languages. So, elementary scaling up of the fundamental techniques does not give notable progress. Thus, there are situations where simple scaling up of the fundamental techniques will not give notable progress, so we have to focus on new techniques. Growth of machine learning techniques in current era helps us to train complex models on massive data. A distributed representation of words is the most fruitful concept. For instance, neural network-based language models expressively overtake N-gram models.

Word2vec method can be utilized to learn good quality word vectors from massive datasets with billions of words, and with millions of words in the vocabulary. Resulting vector representation is measured by this word2vec method. Tendency of similar words togetherness and multiple degrees of similarity are measured in this technique. Previously this technique is observed in the context of inflectional languages. For instance, there are multiple endings for a noun; and if we seek for correspondent words in a subspace of the original vector space, it is potential to identify words that have similar endings.

Similarity between word representations goes ahead simple syntactic regularities. Word offset technique is used to perform simple algebraic operations on the word vectors. For instance, Vector ("Queen") –Vector ("Woman") + Vector ("Man") outputs in a vector which is nearest to the vector representation of "King" [35].

Continuous space language models have recently demonstrated outstanding results across a variety of tasks. Input layer weights are used to learn implicitly of vector- space word representations. Semantic and syntactic regularities of language can be obtained using these representations. And that each relationship is represented by a relation-specific vector offset. Vector oriented reasoning rest on offsets between words is allowed by this technique. For instance, the male/female relationship is automatically studied, and with the induced vector representations, "Queen - Woman + Man" results in a vector very close to "King." The word vectors capture syntactic regularities by means of syntactic analogy questions.

Representing the words as high dimensional real valued vectors is a defining feature of neural network language models. In these models, words are converted via a learned lookup table into real valued vectors which are used as the inputs to a neural network. The major advantage of this model is that distributed representation attains a stage of generalization which cannot achieve using traditional n-gram language models. An n-gram model focuses in terms of discrete units that don't share inherent relationship to one another; a continuous space model focuses in terms of word vectors where similar words are likely to have similar vectors. So, according to the particular word or word sequence, model parameters are adjusted, the advancements will take over to occurrences of similar words and sequences. Model and learned word representations can be achieved by training neural network language model. Forasmuch as both the semantic and syntactic tasks have been devised as analogy questions; cosine distance based simple vector offset method can be effectively used for solving these questions. Vector offsets are used to present

relationships. Hence in the embedding space, all pairs of words sharing a particular relation are presented by the same constant offset.

Although, there are various types of proposed models; Latent Semantic Analysis (LSA) and Latent Dirichlet Allocation (LDA) available for estimating continuous representations of words many different types of models were proposed for estimating continuous representations of words, the distributed representations of words learned by neural networks outplay significantly better than LSA for preserving linear regularities among words [20, 31]; LDA moreover turns into computationally expensive on massive data sets.

Each word is considered as an atomic unit without focusing the relationship between other words in bag of words model. For instance, “book” and “reading” words will be got different unique ids although they both are frequently appeared words within the context or sentence. Sparse word vectors are mapped into continuous space based on the surrounding context in word embedding. This is called as the process of embedding a high-dimensional vector word representation into a lower dimensional space.

The best property in vector representation is comparing words or phrases. For instance, “book” and “reading” word can be indicated conceptually correlated, if those words appear in same context so many times.

Word embedding representations can be constructed using various existing models. Google’s word2vec method leads ahead due to its performance and training speed. Word2vec is trained to predict the target word from the context of surrounding words without utilizing word count; so it is called as predictive model. These are the steps in word2vec model; one-hot encoding is used to encode each word and matrix of weights is used to feed the encoded words into a hidden layer. The output of the process is the target word. The word embedding vectors are actually the weights of this fitted model.

Continuous bag of words (CBOW) and skip gram are the two types of word embedding models in word2vec. CBOW is implemented using the concept of sliding window which means it looks at sliding window of n around words of the target to make a prediction. Despite to this, skip-gram model predicts the surrounding context for a given target word. The applications of word embedding are syntactic parsing, sentiment analysis, name entity recognition (NER) and more. They can also cater a more refined step to present words in numerical space by conserving word to word similarities based on context, give a measure of similarity between words or phrases, can be used as features in classification tasks and increase model achievement.

Since the non-linear hidden layer in neural network model causes computational complexity in learning distributed representations of words, log-linear model was introduced to reduce complexity. The neural network language model was successfully trained in two steps: first, continuous word vectors are learned using simple model, and then the N-gram NNLM is trained on top of these distributed representations of words.

**Continuous Bag-of-Words Model**

The non-linear hidden layer is taken out and the projection layer is shared for all words (not just the projection matrix); thus, all words get projected into the same position (their vectors are averaged). In bag-of-words model the order of words in the history does not influence the projection. Furthermore, it uses words from the future. A log-linear classifier was built with four future and four history words at the input, where the training criterion is to correctly classify the current (middle) word. Continuous distributed representation of the context is used in CBOW in contrast to standard bag of words model. The weight matrix between the input and the projection layer is shared for all word positions in the same way as in the NNLM.

**Continuous Skip-gram Model**

Despite of predicting the target word based on the context, it attempts to maximize classification of a word based on another word in the same sentence. Log-linear classifier with continuous projection layer is inputted with each current word, words are predicted within a certain range before and after the current word. We can obtain good quality by increasing the range, but it rises the computational complexity. Since the more distant words are usually less related to the current word than those close to it, less weight is given to the distant words by sampling less from those words.

We could see different types of similarities among words. For instance, the words “big” and “bigger” are similar in the same sense of the words “small” and “smaller”. The word pair “big-biggest” and “small-smallest” is similar. A similar word of “small” can be identified in the same sense as “biggest” is similar to “big”, via computing simple vector X = vector("biggest") - vector("big") + vector("small"). After that the closest word to X is searched in the vector space using cosine distance. The correct answer can be found if we trained the system properly. We can answer very subtle sematic relationship question, if we train the model using high dimensional word vectors on a huge amount of data. For example, India is to Delhi as Sri Lanka is to Colombo. Word vectors with such semantic relationships could be utilized to enhance many classic NLP systems, such as information retrieval, machine translation and question answering systems.

### 3.2.2 In domain embedding

Learning high-quality domain word embeddings is important for achieving good performance in many NLP tasks. General-purpose embeddings trained on large-scale corpora are often sub-optimal for domain-specific applications. However, domain specific tasks often do not have large in-domain corpora for training high-quality domain embeddings.

The key to the success of word embeddings is that a largescale corpus can be turned into a huge number (e.g., billions) of training examples. Two implicit assumptions are often made about the effectiveness of embeddings to down-stream tasks:

1) The training corpus for embedding is available and much larger than the training data of the down-stream task

2) The topic (domain) of the embedding corpus is closely aligned with the topic of the down-stream task.

However, many real-life applications do not meet both assumptions. In most cases, the in-domain corpus is of limited size, which is insufficient for training good embeddings. In applications, researchers and practitioners often simply use some general-purpose embeddings trained using a very large general-purpose corpus (which satisfies the first assumption) covering almost all possible topics, e.g., the GloVe embeddings [36] trained using 840 billion tokens covering almost all topics/domains on the Web. Such embeddings have been shown to work reasonably well in many domain-specific tasks. This is not surprising as the meanings of a word are largely shared across domains and tasks. However, this solution violates the second assumption, which often leads to sub-optimal results for domain-specific tasks, as shown in our experiments. One obvious explanation for this is that the general-purpose embeddings do provide some useful information for many words in the domain task, but their embedding representations may not be ideal for the domain and in some cases they may even conflict with the meanings of the words in the task domain because words often have multiple senses or meanings. For example, we have a task in the programming domain, which has the word "Java". A large-scale general-purpose corpus, which is very likely to include texts about coffee shops, supermarkets, the Java island of Indonesia, etc., can easily squeeze the room for representing "Java"' context words like "function", "variable" or "Python" in the programming domain. This results in a poor representation of the word "Java" for the programming task. To solve this problem and also the limited in-domain corpus size problem, cross-domain embeddings have been investigated [21] via transfer learning [22]. These methods allow some in-domain words to leverage the general-purpose embeddings in the hope that the meanings of these words in the general-purpose embeddings do not deviate much from the in-domain meanings of these words. The embeddings of these words can thus be improved. However, these methods cannot improve the embeddings of many other words with domain-specific meanings (e.g., "Java"). Further, some words in the general-purpose embeddings may carry meanings that are different from those in the task domain.

### 3.2.3 Convolutional Neural Networks

Image recognition and speech recognition tasks can get advantages from multilayer back propagation networks by learning complex high dimensional nonlinear mappings. Despite to CNN, classic models of pattern recognition gather relevant information from the input and discards irrelevant variabilities using a hand designed feature extractor.

Then the resulting feature vectors or strings are categorized by a trainable classifier into classes. Standard multilayer networks are used in this scheme as classifiers. Eliminating the feature extractor which feeds the network with raw inputs is an interesting scheme. And it is depending on back propagation to turn the first few layers into an appropriate feature extractor. Even though,

it can be done with an ordinary fully connected feed forward network, there are some issues as well.

Initially, there are several hundred variables are in typical images or spectral representation of spoken words. There are several weights in a fully connected first layer with few hidden units. If we train the system with scares resource, over fitting problems may happen. In supplement, many weights may require memory requirement which can be rule out certain hardware requirement. But the important issue of unstructured nets for speech or image systems is that they have no built-in invariance with respect to translations or local distortions of the inputs.

Neural net character images, spoken word spectra and 2D or 1D signal should be approximately size normalized and centered before sending fixed size input layer. Unfortunately, no such preprocessing can be accurate handwriting is often normalized at the word level which can be a reason for size slant and position variations for individual characters. Words can be spoken at varying speed pitch and intonation. This will be the reason for variations in the position of distinctive features in input objects.

Next issue of fully connected architecture is mitting the topology of the input. Outcome of the training will not be affected by changing the order of input variables. Despite of, images or spectral representations of speech have a strong 2D local structure and time series have strong 2D structure variables or pixels that are spatially or temporally nearby are highly correlated. Local correlations are the causes for the famous advantages of extracting and combining local features before recognizing spatial or temporal objects. CNN force the extraction of local features by restraining the receptive fields of hidden units to be local.

### 3.3 Approach

This research focuses on supervised aspect extraction using deep learning. The research proposes a CNN model employing two types of pre-trained embeddings: general-purpose embeddings and domain-specific embeddings [18] with a Word Embedding based Correlation (WEC) model [19] for aspect extraction.

Aspect extraction is done using a double embeddings mechanism. All the information about each word is encoded in the embedding layer which is the very first layer. Later layers (e.g., LSTM, CNN or attention) can decode useful information based on the quality of the embeddings. Either a pre-trained general-purpose embedding, e.g., GloVe [36], or a general review embedding [31] was used in existing deep learning models for aspect extraction.

Since aspect extraction is a complex task it requires fine-grained domain embeddings for better results. For example, in the review sentence "Its speed is awesome" to detect the aspect “speed”, embeddings of both “Its” and “screen” is required. However, the embedding of “Its” and “speed” can be totally different in some criteria. Since “Its” is a general word, the general embedding which was trained from a large corpus will have a better representation for the word “Its”. But, “speed” is a very fine-grained meaning (e.g., number of instructions per second) in the laptop

domain, whereas in general embeddings or general review embeddings “speed" can be referred to number of kilometers per hour.

So, even though the in-domain embedding corpus is not large enough, using in-domain embeddings is important. The network will decide which embeddings have more useful information based on the general embeddings and domain embeddings.

A Word Embedding based Correlation (WEC) model is proposed for extracting implicit features. WEC integrates the advantages of both the word embeddings and the translation model. WEC can calculate the co-occurrence probability for a given random pair of words in review sentence and feature set pairs, while it takes advantage of the smoothness and continuity of continuous space word representation to deal with new pairs of words that are rare in the training parallel text.

Since a fundamental task is to properly extract potential candidate features from the reviews to make better use of information available in customer reviews. In extracting implicit feature, the lexical chasm or lexical gap between the review sentence and candidate features is one of the challenges [24]. Lexical gap describes the distance between dissimilar but potentially related words in review and feature. In implicit reviews the exact feature word is not mentioned in the review, but they are associated by hyponyms, synonyms or other semantic associations [22, 24].

Employing translation model is a possible approach for the lexical gap problem, which will learn the semantically related words from the review and feature pairs [23, 24, 28, 29]. By representing words in a discrete space, relationship between words (or phrases) can be established through word-to-word (or phrase-to-phrase) translation probabilities with the basic assumption of review sentences and feature pairs are "parallel text".

Discrete space representation has two major disadvantages in spite of its wide use in many natural language processing tasks,

1) The curse of dimensionality: need to learn at most NxN word-to-word translation probabilities for a natural language with a vocabulary V of size N [24].

2) The generalization structure is not obvious: if the feature word is rare in the training parallel text, it is difficult to estimate the probability of exact word [28].

Semantic-based model is an alternative method is to use. The lexical gap problem was resolved by using the vector representation of words by using similarity of word vector to represent the word-to-word relation which is using the word embeddings. This method calculates the matching probability of the review and feature based on semantic similarities between words. Because local smoothness properties of continuous space word representations, generalization can be obtained more easily [29]. In some aspects semantic similarities can be weak between review and feature, because sometimes reviews and features are heterogeneous [33].

In this research we propose a Word Embedding Correlation (WEC) model which is inspired by the pros and cons of the translation model and semantic model. WEC integrates the advantages of both the translation model [33, 34] and word embedding [35, 36, 37]. The word-to-word relation is captured using a word level correlations function C(qi, aj). This function calculates words co-

occurrence probability in parallel text which is similar to traditional translation probability (words from review and feature pairs).

The co-occurrence relationship of words is captured into a low dimension dense translation matrix M by mapping input words ri and aj into vectors. This avoids the problem of maintaining a big and sparse translation probability matrix when using word's discrete representation. Because of the local smoothness properties of continuous space word representations, C(ri, aj) can also estimate their correlations strength if co-occurrences of exact words are rare in the training parallel text [37]. A sentence-level correlations functions is proposed based on the word-level correlations function, C(r, a) to calculate the relevance between review words and feature. In sentence-level correlation function also the translation matrix M is learnt directly from parallel corpus.

A Convolutional Neural Network (CNN) [24] model is used for sequence labeling. CNN is also successful in NLP related tasks [22, 26] although most of the existing models have used LSTM [30] to model sequences [29, 32]. LSTM cells are sequentially dependent which is one of the major drawbacks of LSTM.

The training/testing process becomes slower since the back propagation and forward pass must serially go through the whole sequence. Since max-pooling and convolution operations are usually used for sequential inputs summarization it is a challenging task to apply CNN on sequence labeling. Also, the outputs are not well-aligned with the inputs in CNN. To estimate the matching probability, both lexical and syntactical information stored in review words and features are integrated by combining our model with convolution neural network (CNN) [24, 34].

### 3.4 Model

Implicit features are extracted based on the methods that are used for extracting explicit features. Both explicit features and implicit features are extracted with the help of general embeddings, domain embeddings and CNN approach.

#### 3.4.1 Explicit feature extraction

The proposed model has 2 embedding layers at first, 4 CNN layers next and a fully connected layer which is shared across all positions of words. Also, a SoftMax layer over the labeling space Y = {B, I, O} for each position of inputs. Since aspect can be a phrase in the review, here B indicates the beginning word and I indicates non-beginning word of an aspect phrase and O indicates non-aspect words available in the review. Assumed that the input is a sequence of word indexes, can be represented as x = (x1, . . ., xn). Here two separate embedding layers called (or embedding matrices) Wg which is general embedding and Wd which is domain embedding are used and the word sequences get there two corresponding continuous representations xg and xd via Wg and Wd. The proposed model for extracting explicit features is shown in Figure 7.

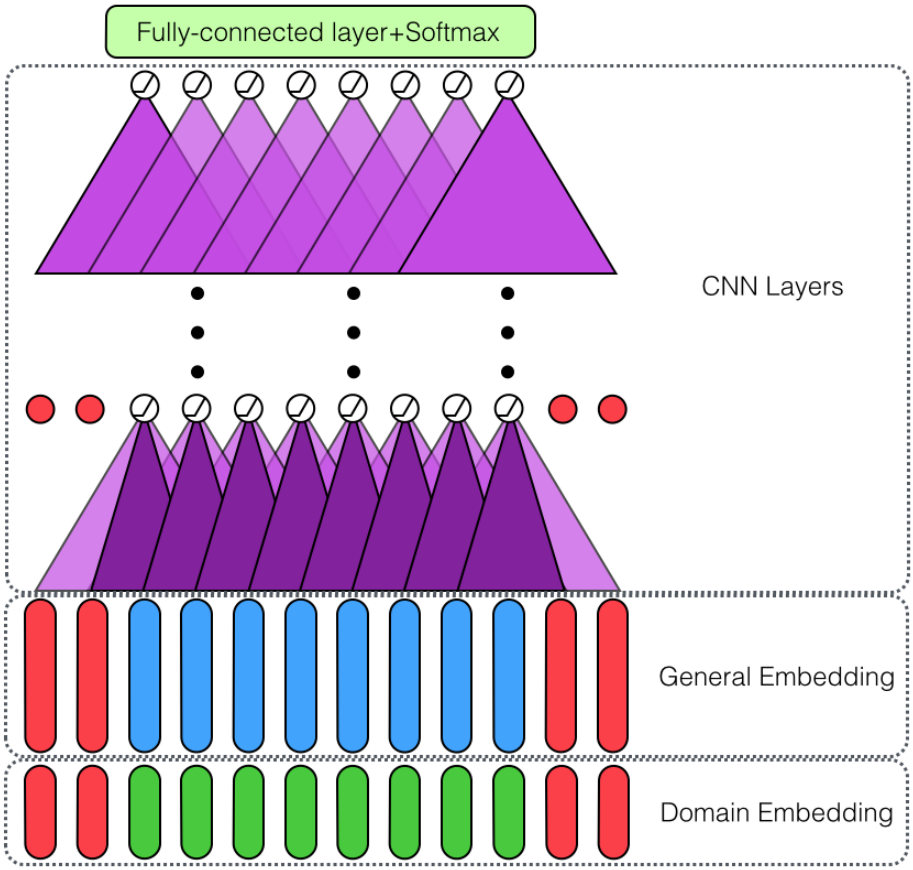



**Figure 7:** The proposed model for extracting explicit features

The first embedding matrix is pretrained from a very large general-purpose corpus which is the general embeddings (Wg). The second embedding matrix is pretrained from a small in-domain corpus which is domain embeddings (Wd). The scope of the domain embedding is exactly the domain to which the testing/training data belongs to. For example, the electronics domain will be considered as out of domain embeddings if the testing or the training data is in the laptop domain (e.g., the word “adapter” electronics domain may refer to different types of adapters where laptop domain will exactly refer to a laptop adapter).

This means only laptop reviews will be considered as in domain. If these two embedding layers trainable, because of the small training examples this may lead to many unseen words in test data. So, it is not allowed to train these two embedding layers. The features for seen words' embeddings can be adjusted if embeddings are tunable. For example, new features which are related to the labels of the training examples can be infused while ignoring useless features. And the CNN filters will be get adjusted to the new features accordingly. But there is the possibility for old features available in test data that may be mistakenly extracted by CNN from the embeddings of unseen words.

Then two embeddings x(1) = xg ⊕ xd are concatenated and the result is fed into a stack of 4 CNN layers. Many 1D-convolution filters are available in a CNN layer and each filter (r-th) has a fixed kernel size k = 2c+1. The following convolution operation and ReLU activation are performed:

$$x_{i,r}^{(l+1)} = \max\left(0, \left(\sum_{j=-c}^{c} w_{j,r}^{(l)} x_{i+j}^{(l)}\right) + b_r^{(l)}\right) \quad (3.1)$$

where l is the l th CNN layer. Each filter is applied to all positions i = 1: n. So, the representation for the i th word along with 2c nearby words in its context is computed by each filter. The kernel size k is forced to be an odd number and the stride step is set to be 1. Furthermore, pad the right c and left c positions with all zeros. In this way, for sequence labeling purposes the original input x is well aligned with the output of each layer. Two different filter sizes are employed for the first (l = 1) CNN layer. And only use one filter size is used for the rest 3 CNN (l ∈ {2, 3, 4}) layers. Finally, a fully connected layer is applied with weights shared across all positions and the label distribution for each word is computed using a softmax layer. The fully connected layer's output size is |Y| = 3. After the embedding layer and each ReLU activation dropout is applied. Since a good representation is needed for every position in a sequence labeling model, max-pooling layer is not applied after convolution layers because the representations of different positions will be mixed by max-pooling operation.

### 3.4.2 Implicit feature extraction

Given a review r = r1...rn, where ri is the i-th term in the review, and a candidate feature set "A = $\{a^1, a^2, ..., a^n\}$", where $a^j = a^j_1 ... a^j_m$ and $a^j_k$ is the "k-th" word in "j-th" candidate feature, the goal is to extract the relevant feature from the feature set.

To identify the implicit feature the matching probability between review sentence and each feature is calculated and then the candidate features are ranked based on their identical probabilities. Matching probabilities are calculated using these three steps:

1. Review words and features are signified as vectors using continuous space.

2. A word level correlation function is used to calculate the score of word-to-word correlation.

3. A phrase-level correlation function is used to obtain the review and feature matching probability.

Furthermore, to achieve a better matching precision, convolution neural network (CNN) is used along with WEC model.

**Word Embedding**

Bengio et al.'s [21] neural language model is used to study the word embeddings in an n dimensional vector space and predict how likely the vectors given its context for a word to represent words correctly in a continuous space. A widely used method for computing such embeddings is Skip gram model [35]. Skip gram networks can be optimized through ascent of gradient and the word embedding matrix "$L \in R^{(n\times|V|)}$" is modified using derivatives, here the vocabulary size is denoted by |V|. Word vectors which are available in the embedding matrix are used to identify the semantic information and distributional syntactic via Bengio et al. [19] and Mikolov et al.'s [35] word co-occurrence statistics. Each word's vector ($v_w$ - one column in L) can be used to represent that particular word in subsequent tasks once the matrix is learned on an unlabeled corpus.

**Word Embedding based Correlation (WEC) Model**

Word Level Correlation Function: A correlation scoring function is created using the word embeddings as input and at the same time this function can model the co-occurrence of words. A translation matrix (M) is used for transformation of feature words into words of the review to achieve this goal. The WEC scoring function is defined as: Given a pair of words ($r_i$, $a_j$),

$$C(\mathrm{r}_i, a_j) = \cos < v_{\mathrm{r}_i}, \mathbf{M}v_{a_j} >= \frac{v_{\mathrm{r}_i}^{\mathrm{T}}}{||v_{\mathrm{r}_i}||} \frac{\mathbf{M}v_{a_j}}{||\mathbf{M}v_{a_j}||} \tag{3.2}$$

Here $r_i$ and $a_j$ 's d-dimensional word embedding vectors is represented by $v_{ri}$ and $v_{aj}$; Euclidean norm is denoted by $\| \cdot \|$; matrix for correlations is "$M \in R^{d\times d}$". Here "M" is translation matrix, this translation matrix does the mapping for feature word into a possible correlated word in the review. The similarity of semantic between the mapped and origin words in the review are captured using the cosine function. When identity matrix is set with M, a special case of WEC scoring function is the previous cosine similarity. In this model $C(a_j, r_i)$ is not necessarily equal to $C(r_i, a_j)$, as the probability of $a_j$ presents in review and $r_i$ presents in feature might not have equal probability of $r_i$ presents in review and $a_j$ presents in feature.

**Phrase-level Correlation Function**

Using the word level correlation function, a phrase level correlation function is created to identify the review and feature pair correlation score by integrating word-to-word correlation scores. The correlation score for a review and feature pair (r, a) is defined as:

$$C(\mathrm{r}, a) = \frac{1}{|a|} \sum_j \max_i C(\mathrm{r}_i, a_j) \tag{3.3}$$

Here the length of feature a is denoted by |a|. Score of the correlations between the review's $i^{th}$ and feature's $j^{th}$ word is represented by $C(r_i, a_j)$. For each word in the feature phrase one most related word from the review is mapped using the max operator. By averaging the selected word-level scores the phrase level correlation score is calculated. The max-average function is used to maximize the correlation score rather than just averaging the word level correlation score, this performs well and efficiently.

**WEC combined with Convolution Neural Networks (CNN)**

WEC is a bag of word-based method and it puts the syntactical information aside, which means it does not consider the word sequence information. If we consider a case where two phrases have the same bag of words model while they have the completely opposite real meaning [19]. To overcome this problem, can use the convolution neural network (CNN) model (Mou et al.; He, Gimpel, and Lin). According to Kalchbrenner et al., in CNN model dynamic pooling and convolutional layer can relate input sentence's far apart phrases. The S+CNN model which is proposed by (Shen et al.), it estimates the matching probability by integrating both lexical and syntactical information for review-feature matching. They have used the following function to transform the input review-feature pair into a similarity matrix S.

$$\mathbf{S}_{ij} = cos(\mathrm{r}_{i \bmod |\mathrm{r}|}, a_{j \bmod |a|}) \qquad (3.4)$$

Here |r| is the lengths of review and |a| is the lengths of feature, S is a fixed size matrix of $n_f \times m_f$, and $n_f$ is the number of rows and $m_f$ are the number of columns. Thus, the maximum length for reviews should be less than $n_f$ and features should be less than $m_f$. Then the CNN is used to identify the feature where the input for CNN is similarity matrix (LeCun et al.). Then CNN gives the matching score of the review-feature pair as the output. Figure 8 shows the architecture of the WEC + CNN.

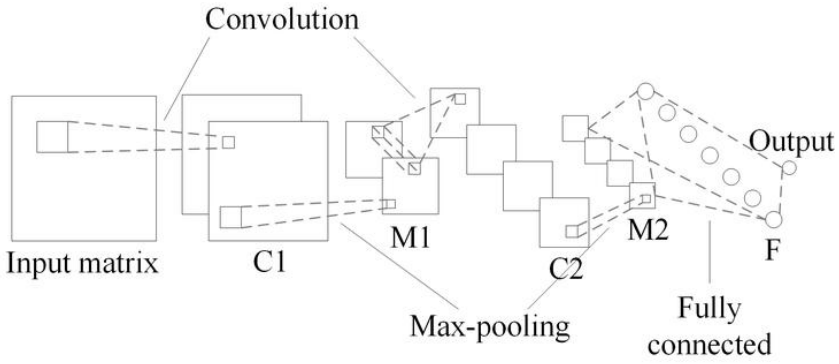



**Figure 8:** Architecture of WEC + CNN

The CNN model contains double convolution layers, "C1" and "C2", then a max pooling layer "P1" and "P2" is added after each convolution layer, and finally a fully connected layer "F". Input matrix "S" is a fixed size matrix of "nf × mf".

The input matrix for CNN is created using the word level correlation scores produced by WEC. The correlations matrix C is generated using the following formula:

$$\mathbf{C}_{ij} = C(q_{i \bmod |q|}, a_{j \bmod |a|}) \quad (3.5)$$

Here, "c" is a fixed size matrix of "nf × mf" which is used as the input for CNN. Through these steps a combination model called WEC + CNN is obtained. Two supervised pretraining and a fine tuning step is used in the training process. The WEC function's output margin is maximized to pretrain M in the first pretraining step. Then CNN's output margin is maximized while keeping M as fixed to train CNN model in the second pretraining step. Finally, CNN's output margin is maximized in order to fine tune WEC and CNN's parameters in the fine-tuning step. CNN gives the matching score of the review feature pair as the output and the review's corresponding feature will be identified from the feature set.

### 3.5 Summary

This chapter discussed the methodology used for extracting implicit features. Discussed how general-purpose embeddings, domain-specific embeddings and CNN helped to extract the explicit features and then it is associated with a Word Embedding based Correlation (WEC) model to extract the implicit features.

# EXPERIMENTS

## 4.1 Overview

Experiments are done on different datasets to show that our proposed method outperforms the baseline methods. Different datasets are collected from SemEval, Liu et al. [15] and Karmaker et al. [6] for the experiment. The baseline methods are also discussed in detail.

## 4.2 Performance Measures

The performance of the proposed approach is measured using the three standard measures available in the literature: Recall, Precision and the F1 measure. For each review sentence we have recorded the implicit features identified by our approach along with the review sentence and the result is compared with the gold implicit features. Then we compute the true positive, true negative, false positive and false negative counts by comparing each sentence's identified features. After Precision, Recall and F1 measure are calculated using the true positive, true negative, false positive and false negative counts for each dataset.

## 4.3 Datasets

Experiments for explicit feature extraction is conducted on two benchmark datasets from SemEval challenges [14]. The data set is shown in Table 11. The first dataset is from the laptop domain on subtask 1 of SemEval-2014 Task 4. The second dataset is from the restaurant domain on subtask 1 (slot 2) of SemEval-2016 Task 5. These two datasets consist of review sentences with aspect terms labeled as spans of characters. Used NLTK to tokenize each sentence into a sequence of words. For the general-purpose embeddings, used the glove.840B.300d embeddings [26], which are pre-trained from a corpus of 840 billion tokens that cover almost all web pages. These embeddings have 300 dimensions. For domain-specific embeddings, collected a laptop review corpus and a restaurant review corpus and used fastText [30] to train domain embeddings. The laptop review corpus contains all laptop reviews from the Amazon Review Dataset [36]. The restaurant review corpus is from the Yelp Review Dataset Challenge 4. We only use reviews from restaurant categories that the second dataset is selected from. We set the embedding dimensions to 100 and the number of iterations to 30 (for a small embedding corpus, embeddings tend to be under-fitted), and keep the rest hyper-parameters as the defaults in fastText. We further use fastText to compose out-of-vocabulary word embeddings via subword N-gram embeddings.

For implicit feature extraction evaluation, Karmaker et al. [6] created eight new data sets, which includes datasets annoted by humans as well as datasets annotated automatically by using computational methods. The review data made available by Hu et al. was used to create the implicit feature extraction evaluation dataset. Five different electronic products' reviews are included in these five different datasets. Totally these datasets contain 314 reviews which have 4259 review sentences. For each sentence, the features that are mentioned in that sentence was tagged in the dataset. Since implicit feature mentions were required for evaluation, as the dataset

produced by Hu et al. [15] contains only the explicit feature mentions, the implicit feature mentions were tagged by Karmaker et al. [6] and Hu et al. [15] tagged the explicit feature mentions and Karmaker et al. tagged the implicit feature mentions in the same dataset and these details are shown in Table 12 and 13.

**Table 11:** SemEval 14 and 16 datasets

| Dataset | Number of sentences | Number of aspect terms |
|---|---|---|
| SemEval-14 Laptop | 800 | 654 |
| SemEval-16 Restaurant | 676 | 622 |

**Table 12:** Human annotated dataset from Liu et al. [15]

| Dataset | Total no. of reviews | Total no. of sentences |
|---|---|---|
| DVD player | 99 | 839 |
| Cellular phone1 | 41 | 587 |
| Digital camera1 | 45 | 642 |
| Digital camera2 | 34 | 380 |
| Mp3 player1 | 95 | 1811 |
| Total | 314 | 4259 |

**Table 13:** Automatically annotated dataset from Karmaker et al. [6]

| Dataset | Total no. of reviews | Total no. of sentences |
|---|---|---|
| Mp3 player2 | 1495 | 10347 |
| Cellular phone2 | 966 | 9856 |
| Router | 2238 | 17853 |
| Total | 4699 | 38056 |

### 4.4 Hyper-parameter

We hold out 150 training examples as validation data to decide the hyper-parameters. The first CNN layer has 128 filters with kernel sizes k = 3 (where c = 1 is the number of words on the left (or right) context) and 128 filters with kernel sizes k = 5 (c = 2). The rest 3 CNN layers have 256 filters with kernel sizes k = 5 (c = 2) per layer. The dropout rate is 0.55 and the learning rate of Adam optimizer (Kingma and Ba) is 0.0001 because CNN training tends to be unstable.

### 4.5 Baseline Methods

We perform a comparison of DE-CNN with three groups of baselines using the standard evaluation of the datasets. The results of the first two groups are copied from Li and Lam [16].

The first group uses single-task approaches.

- CRF is conditional random fields with basic features and GloVe word embedding [26].
- IHS RD [21] and NLANGP [22] are best systems in the original challenges.
- WDEmb [23] enhanced CRF with word embeddings, linear context embeddings and dependency path embeddings as input.
- LSTM [24] is a vanilla BiLSTM.
- BiLSTM-CNN-CRF [25] is the state-of-the-art from the Named Entity Recognition (NER) community. We use this baseline to demonstrate that a NER model may need further adaptation for aspect extraction.

The second group uses multi-task learning and also take advantage of gold-standard opinion terms/sentiment lexicon.

- RNCRF [27] is a joint model with a dependency tree based recursive neural network and CRF for aspect and opinion terms co-extraction. Besides opinion annotations, it also uses handcrafted features.
- CMLA [28] is a multi-layer coupled-attention network that also performs aspect and opinion terms co-extraction. It uses gold standard opinion labels in the training data.
- MIN [23] is a multi-task learning framework that has two LSTMs for jointly extraction of aspects and opinions, and a third LSTM for discriminating sentimental and non-sentimental sentences. A sentiment lexicon and high precision dependency rules are employed to find opinion terms.

The third group is the variations of DE-CNN.

- GloVe-CNN only uses glove.840B.300d to show that domain embeddings are important.
- Domain-CNN does not use the general embeddings to show that domain embeddings alone are not good enough as the domain corpus is limited for training good general words embeddings.
- MaxPool-DE-CNN adds max-pooling in the last CNN layer. We use this baseline to show that the max-pooling operation used in the traditional CNN architecture is harmful to sequence labeling.
- DE-OOD-CNN replaces the domain embeddings with out-of-domain embeddings to show that a large out-of-domain corpus is not a good replacement for a small in-domain corpus for domain embeddings. We use all electronics reviews as the out-of-domain corpus for the laptop and all the Yelp reviews for restaurant.
- DE-Google-CNN replaces the glove embeddings with GoogleNews embeddings, which are pre-trained from a smaller corpus (100 billion tokens). We use this baseline to demonstrate that general embeddings that are pre-trained from a larger corpus performs better.

- DE-CNN-CRF replaces the SoftMax activation with a CRF layer. We use this baseline to demonstrate that CRF may not further improve the challenging performance of aspect extraction.

This research shows how our proposed method is efficiently exploiting the unlabeled data in evaluating the implicit feature extraction also shows how efficient our method is when comparing with the state-of-the-art methods. To show the efficiency of our method when comparing with the state-of-the-art methods, we have used Naive Bayes classifier which is a supervised learning method that performs well in text classification tasks. And a correlation count method that was proposed by Zhang et al. which is the current state of the art that identifies the implicit features by counting the correlation between opinion words and feature words.

### 4.5.1 Naive Bayes Classifier Based Ranking

In this approach, to identify the implicit feature, the sentences which are mentioning the features explicitly are taken as the training samples. Then a classifier is created using the words that are available in the training set so that the sentences that do not mention the features explicitly can get hint from the training data to identify the implicit feature. The words in the review sentence S is taken as the input for the ranking function. Then the ranking function produces a score based on how much likely that f' is appeared implicitly in the review sentence for each of the candidate implicit feature f'. The following formula is used to calculate the score for the implicit feature identification:

$$\begin{aligned} score(f'|S) &= P(w_1|f') \times P(w_2|f') \times .... \times P(w_n|f') \times P(f') \\ &= P(f') \times \Pi_{i=1}^{n} P(w_i|f') \end{aligned} \tag{4.1}$$

Here, S = {w1, w2, …., wn}. A threshold value θ is used to select the implicit feature. If score(f'|S) > θ for a candidate feature f' then f' will be assigned to sentence S.

### 4.5.2 Correlation Based Ranking

The following equation shows the ranking function which is used to calculate the correlation measurements between candidate feature (f') and context words ($w_i$):

$$score(f'|S) = \frac{\sum_{i=1}^{n} \frac{M(f', w_i)}{count(w_i)}}{n} \tag{4.2}$$

Here, number of times word wi is mentioned in the entire dataset is given by count($w_i$) and the number of times both f' and wi appear together in a sentence is given by M(f', $w_i$). And a threshold θ is used to identify the implicit feature. If score(f'|S) > θ for a candidate feature f' then f' will be assigned to sentence S.

### 4.5.3 Generative Feature Language Model

The Generative Feature Language Model (GFLM) was created using generative mixture model with feature language models as components and the review data was created using this model. This includes the following processes.

1. Generating each sentence by independently generating each of the words in the sentence.

2. Decide on whether to generate the word in a sentence using the background model ($\gamma_B$) or a feature language model.

3. The word will be sampled from the distribution $p(w|\gamma_B)$ if the word is chosen from the background language model. Otherwise, need to decide on which of the k feature language models to use, with the help of another set of parameters $\{\pi_{S,i}\}$. $\pi_{S,i}$ gives the probability of choosing feature language model $\gamma_i$ to generate the word. With probability $\pi_{S,i}$, can sample the word using $p(w|\gamma_i)$.

4. All the words in a sentence will be generated by repeating this process. Set of sentence specific topic choice parameters $\pi_{S,i}$ will be used to generate each sentence.

## 4.6 Summary

This chapter explained the experimental setup and different types of dataset used for the feature extraction. Baseline methods are chose based on the current state of the art method which is the Generative Feature Language Model. Our proposed method is evaluated with the baseline methods Naive Bayes Classifier Based Ranking, Correlation Based Ranking and Generative Feature Language Model.

# RESULTS AND DISCUSSION

## 5.1 Overview

To complete this study properly, it is necessary to analyze the results obtained in order to check the objective of this research has accomplished. The present study was an attempt to extract the implicit features from the customer reviews. This section presents the results that are evaluated on the SemEval datasets and Karmaker et al.'s [6] datasets.

## 5.2 Explicit feature extraction results

The double embedding mechanism improves the performance and in-domain embeddings are important. We can see that using general embeddings (GloVe-CNN) or domain embeddings (Domain-CNN) alone gives inferior performance. We further notice that the performance on Laptops and Restaurant domains are quite different. Laptops has many domain-specific aspects, such as "adapter". So, the domain embeddings for Laptops are better than the general embeddings. The Restaurant domain has many very general aspects like "staff", "service" that do not deviate much from their general meanings. So general embeddings are not bad. Max pooling is a bad operation as indicated by MaxPool-DE-CNN since the max pooling operation loses word positions.

DE-OOD-CNN's performance is poor, indicating that making the training corpus of domain embeddings to be exactly in-domain is important. DE-Google-CNN uses a much smaller training corpus for general embeddings, leading to poorer performance than that of DE-CNN. Surprisingly, we notice that the CRF layer (DE-CNN-CRF) does not help. In fact, the CRF layer can improve 1-2% when the laptop's performance is about 75%. But it doesn't contribute much when laptop's performance is above 80%. CRF is good at modeling label dependences (e.g., label I must be after B), but many aspects are just single words and the major types of errors (mentioned later) do not fall in what CRF can solve. Note that we did not tune the hyperparameters of DE-CNN-CRF for practical purpose because training the CRF layer is extremely slow.

One important baseline is BiLSTM-CNN-CRF, which is markedly worse than our method. We believe the reason is that this baseline leverages dependency-based embeddings [32], which could be very important for NER. NER models may require further adaptations (e.g., domain embeddings) for opinion texts.

**Table 14:** Explicit features F1 score comparison results

| Model | Laptop | Restaurant |
|---|---|---|
| CRF | 74.01 | 69.56 |
| IHS RD | 74.55 | - |
| NLANGP | - | 72.34 |
| WDEmb | 75.16 | - |
| LSTM | 75.25 | 71.26 |
| BiLSTM-CNN-CRF | 77.8 | 72.5 |
| RNCRF | 78.42 | - |
| CMLA | 77.80 | - |
| MIN | 77.58 | 73.44 |
| GloVe-CNN | 77.67 | 72.08 |
| Domain-CNN | 78.12 | 71.75 |
| MaxPool-DE-CNN | 77.45 | 71.12 |
| DE-LSTM | 78.73 | 72.94 |
| DE-Google-CNN | 78.8 | 72.1 |
| DE-OOD-CNN | 80.21 | 74.2 |
| DE-CNN-CRF | 80.8 | 74.1 |
| DE-CNN | 81.59 | 74.37 |

### 5.3 Implicit Feature extraction results

Here, we denote the Naive Bayes classifier baseline method with the abbreviation "NB" and the Correlation-based baseline method with "CR" and "GFLM" for generative feature language model baseline method. And “DWEC” for our proposed approach Dual Word Embedding Correlation.

*Comparison with baselines*

We have compared our proposed approach DWEC with the baseline algorithms and the summary of the results on different datasets is presented in Table 16. The precision, recall and their corresponding F1 score for each evaluation dataset is shown in the table 16. These results are obtained for a threshold value θ. Our method outperforms the state-of-the-art methods robustly. For example, for Cellular Phone 2 dataset, GFLM achieved maximum F1 score of 0.5125, Precision of 0:6543 and Recall of 0:4212, while DWEC obtained F1 score of 0.8683, Precision of 0:8810 and Recall of 0:8560. This is the best result obtained for DWEC among all datasets. Our proposed method DWEC performs well than the baseline methods in terms of Precision, Recall and F1 score by a large margin. Practically, even though the recall value is higher it has little practical value if the corresponding precision is very low. In DWEC it preserves precision even though the recall is higher for some dataset. But in baseline methods it varies for some datasets.

**Table 15:** Implicit features F1 score, Precision and Recall comparison results

| Dataset | F1 Score | | | | Precision | | | | Recall | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | NB | CR | GFLM | **DWEC** | NB | CR | GFLM | DWEC | NB | CR | GFLM | **DWEC** |
| Cellular phone[1] | 0.2446 | 0.3092 | 0.4840 | **0.8521** | 0.1818 | 0.2206 | 0.5487 | **0.8601** | 0.3736 | 0.5164 | 0.4329 | **0.8442** |
| Cellular phone[2] | 0.3147 | 0.3420 | 0.5125 | **0.8683** | 0.2157 | 0.2268 | 0.6543 | **0.8810** | 0.581 | 0.6950 | 0.4212 | **0.8560** |
| DVD player | 0.2947 | 0.2671 | 0.5570 | **0.8518** | 0.2406 | 0.2390 | 0.6397 | **0.8293** | 0.3800 | 0.3025 | 0.4933 | **0.8754** |
| Mp3 player[1] | 0.3312 | 0.3380 | 0.3831 | **0.8417** | 0.3253 | 0.3870 | 0.5705 | **0.8515** | 0.3375 | 0.3 | 0.2886 | **0.8321** |
| Mp3 player[2] | 0.2177 | 0.2555 | 0.4439 | **0.8360** | 0.1483 | 0.1619 | 0.5940 | **0.8405** | 0.4090 | 0.6060 | 0.3545 | **0.8315** |
| Digital camera[1] | 0.4051 | 0.4134 | 0.597 | **0.8566** | 0.3775 | 0.4156 | 0.6791 | **0.8522** | 0.4371 | 0.4112 | 0.5325 | **0.8610** |
| Digital camera[2] | 0.4604 | 0.4634 | 0.6666 | **0.8537** | 0.4477 | 0.6495 | 0.6574 | **0.8715** | 0.4739 | 0.3601 | 0.6761 | **0.8367** |
| Router | 0.4805 | 0.5291 | 0.6686 | **0.8132** | 0.4213 | 0.8009 | 0.7487 | **0.8014** | 0.5593 | 0.3951 | 0.6040 | **0.8254** |

### 5.4 Analysis

The reason for our method to outperform all the baseline methods is due to its ability to learn from unlabeled sentences using the domain and general embeddings which are represented in a vector space. Also, it removes the noisy words from the corpus in an efficient way when comparing to the simple heuristic approaches. The detailed analysis of our approach and the behaviors of these algorithms will give a better understanding of our approach.

Our proposed method outperforms the baseline methods in terms of Precision, Recall and F1 score. We could be able to achieve more than 80% of accuracy in extracting the implicit features for all types of dataset which is a major success for our research. Among these dataset “Cellular Phone 2” dataset obtained higher F1 score of 0.8683 with precision of 0.8601 and recall of 0.8442. Since large amount of Cellular Phone specific data was collected, it helped to achieve good results. At the same time “Router” dataset got the lowest F1 score of 0.8132 with precision of 0.8014 and recall of 0.8254. The “Router” dataset’s score has been reduced because of its limitation of domain specific data. However, it outperformed the existing methods. Unlike the baseline methods our method preserves precision and recall while achieving good F1 score. Application of CNN on top of the correlation matrix helped to improve the results. Our research shows that supervised learning approach provides better results compared to unsupervised and semi-supervised learning.

For example, if we take a sentence form the "Cell Phone 2" dataset "At my heaviest usage, I must recharge after 3 days", here the implicit feature mentioned is "battery". Since this review sentence is not directly talking about the feature "battery", so it is nontrivial to predict the feature "battery". While the baseline approaches are not able to identify implicit features accurately, our proposed approach DWEC infers the feature "battery" correctly with a high correlation. If we look deeper

into the explanation why our method is predicting with high accuracy, as the first step the noise words will be filtered out except the words "heaviest", "usage", "recharge", "days". For words "usage" and "recharge", the probabilities of being appear in the cell phone domain is very high. Also, these words "usage" and "recharge" are highly correlated to the feature "battery" than any other features which are available in the feature set. Thus, DWEC inferred the feature "battery" as the implicit feature for the given review sentence. The baseline methods NB, CR and GFLM approaches inferred features "phone", "sound" and "size" respectively. If consider the predictions of the baseline methods, the features "phone", "sound" and "size" are features are related to the overall product at a high level. Here, it is important to note that all the baseline methods fail to predict the actual implicit feature. So, it is important to remove noisy words efficiently and need to consider the domain words more importantly in predicting correctly otherwise it will lead to wrong predictions. If we note that the features "phone" "sound" and "size" are more frequent words compared to the feature "battery". Thus, when the baseline methods are considering the probabilities, the prior probabilities for the features, "phone", "sound" and "size" would have predict these features for the baseline approaches. The baseline methods GFLM, NB and CR are highly depending on the thresholding parameter, a slight change in the parameter results in a big change in the output.

### 5.4.1 Feature Level Analysis

If we look at the feature level analysis, the signal word played important role in predicting implicit features from the review sentences. For example, the words "loud", "earpiece" and "quiet" are the top signal words in predicting the implicit feature sound from the reviews in the cell phone domain, which are closely correlated to the feature sound in the vector space. When human is predicting the feature, they will also predict the implicit feature based on the similar context words. If we take another example, "HD", "video" and "Netflix" are the mostly correlated words in the vector space for predicting the feature "streaming" in the router domain. Table 15 shows some samples of mostly correlated word with the feature words.

This shows that the domain embedding is more important than general embeddings in predicting implicit features because same words can have different meanings in different domains. This analysis shows that our approach is predicting features close to human prediction. Our approach does not require any heuristic measures or any manual parameter tuning to extract the implicit features.

**Table 16:** Feature words and their top correlated words

| Dataset | Feature | word 1 | word 2 | word 3 |
|---|---|---|---|---|
| Cellular phone (1 & 2) | sound | quiet | loud | earpiece |
| | size | small | pocket | fit |
| | battery | charge | life | long |
| | internet | wap | hotspot | end |
| | button | back | press | push |
| | design | robust | sleek | weight |
| | camera | image | flash | picture |
| Mp3 player (1 & 2) | software | install | update | xp |
| | button | cracked | press | pause |
| | storage | disk | huge | space |
| | warranty | date | replacement | repair |
| | transfer | kbps | usb | load |
| | battery | recharge | charge | plug |
| Digital camera (1 & 2) | use | access | highly | fun |
| | price | worth | cheaply | point |
| | design | flaw | plastic | superb |
| | battery | continue | solid | nice |
| | software | raw | os | consistently |
| | memory | fit | reader | large |
| Router | software | XP | proxy | program |
| | streaming | Netflix | video | HD |
| | installation | physically | error | manual |
| | price | worth | refund | Amazon |
| | port | usb | SharePort | Compatible |
| | speed | high | HD | mbps |
| DVD player | format | avi | file | able |
| | sound | optical | cd | vcd |
| | service | answer | busy | response |
| | price | worth | dollar | guess |
| | screen | light | silver | load |
| | quality | build | pretty | begin |

### 5.5 Summary

Explicit feature extraction and implicit feature extraction results are discussed separately. The results show that our proposed method outperformed the baseline methods. The results are increased as our proposed approach is able to learn from unlabeled sentences using the domain and general embeddings which are represented in a vector space. Detailed analysis is done to show how our approach improved the quality of implicit feature extraction.

# CONCLUSION AND FUTURE WORK

## 6.1 Conclusion

Implicit features extraction from a review sentence is a key task in analyzing and summarizing customer reviews. Most of the researches focused on extracting explicit features from the online customer reviews, while only few researches have focused on extracting implicit features. So this research will help to make the customer review analyzing easy. In this research developed a supervised aspect extraction method using deep learning. A novel and yet simple CNN model employed two types of pre-trained embeddings for aspect extraction: general-purpose embeddings and domain-specific embeddings (DE-CNN) associated with a Word Embedding based Correlation (WEC) model. Even though, large scale of general-purpose embeddings are available I have mainly used domain specific embeddings combined with general-purpose embeddings to get the fine grained meaning of each words. Meaning for a word will change in different domains so when extracting the implicit features domain specific knowledge is required. That is the reason behind using domain specific embeddings which gave better results. Word Embedding based Correlation (WEC) model helped to calculate the co-occurrence probability for a given random pair of words in review sentence and feature set pairs, while it takes advantage of the smoothness and continuity of continuous space word representation to deal with new pairs of words that are rare in the training parallel text. Convolution neural network is used to perform sequence labeling and estimate the matching probability to identify the implicit feature(s) of the particular review sentence. Experiment on different datasets and the comparison with the baseline methods shows that the proposed approach has outperformed the baseline methods.

## 6.2 Future Work

The implicit feature identification is heavily depending on the in-domain data, where only a limited amount of in-domain data available. If we gather more in-domain data in the training process, then we can increase the precision and accuracy of the method. Also, a predefined feature set should be provided when predicting the implicit features from the review sentence. Because of this if the review is mentioning a different feature which is not included in the feature set then that feature will not be identified. So, need to find a way to predict the implicit feature with the help of in-domain data without providing a feature set.

# REFERENCES


[1] Quan, C. and Ren, F., 2014. Unsupervised product feature extraction for feature-oriented opinion determination. Information Sciences, 272, pp.16-28.

[2] Su, Q., Xiang, K., Wang, H., Sun, B. and Yu, S., 2006, November. Using point wise mutual information to identify implicit features in customer reviews. In ICCPOL (Vol. 4285, pp. 22-30).

[3] Hai, Z., Chang, K. and Kim, J.J., 2011. Implicit feature identification via co-occurrence association rule mining. Computational Linguistics and Intelligent Text Processing, pp.393-404.

[4] Wang, W., Xu, H. and Wan, W., 2013. Implicit feature identification via hybrid association rule mining. Expert Systems with Applications, 40(9), pp.3518-3531.

[5] Xu, H., Zhang, F. and Wang, W., 2015. Implicit feature identification in Chinese reviews using explicit topic mining model. Knowledge-based systems, 76, pp.166-175.

[6] Karmaker Santu, S.K., Sondhi, P. and Zhai, C., 2016, October. Generative Feature Language Models for Mining Implicit Features from Customer Reviews. In Proceedings of the 25th ACM International on Conference on Information and Knowledge Management (pp. 929-938). ACM.

[7] Zhang, Y. and Zhu, W., 2013, May. Extracting implicit features in online customer reviews for opinion mining. In Proceedings of the 22nd International Conference on World Wide Web (pp. 103-104). ACM.

[8] Liu, L., Lv, Z. and Wang, H., 2013. Extract Product Features in Chinese Web for Opinion Mining. JSW, 8(3), pp.627-632.

[9] Poria, S., Cambria, E., Ku, L.W., Gui, C. and Gelbukh, A., 2014, August. A rule-based approach to aspect extraction from product reviews. In Proceedings of the second workshop on natural language processing for social media (SocialNLP)(pp. 28-37).

[10] Zeng, L. and Li, F., 2013. A classification-based approach for implicit feature identification. In Chinese Computational Linguistics and Natural Language Processing Based on Naturally Annotated Big Data (pp. 190-202). Springer, Berlin, Heidelberg.

[11] Bhatnagar, V., Goyal, M. and Hussain, M.A., 2016, August. A Proposed framework for improved identification of implicit aspects in tourism domain using supervised learning technique. In Proceedings of the International Conference on Advances in Information Communication Technology & Computing (p. 56). ACM.

[12] Wagstaff, K., Cardie, C., Rogers, S. and Schrödl, S., 2001, June. Constrained k-means clustering with background knowledge. In ICML (Vol. 1, pp. 577-584).

[13] Bauman, K., Liu, B. and Tuzhilin, A., 2017, August. Aspect Based Recommendations: Recommending Items with the Most Valuable Aspects Based on User Reviews. In Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (pp. 717-725). ACM.

[14] Zhang, L., Liu, B., Lim, S.H. and O'Brien-Strain, E., 2010, August. Extracting and ranking product features in opinion documents. In Proceedings of the 23rd international conference on computational linguistics: Posters (pp. 1462-1470). Association for Computational Linguistics.

[15] Hu, M. and Liu, B., 2004, August. Mining and summarizing customer reviews. In Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining (pp. 168-177). ACM.

[16] Jindal, N. and Liu, B., 2008, February. Opinion spam and analysis. In Proceedings of the 2008 International Conference on Web Search and Data Mining (pp. 219-230). ACM.

Qiu, G., Liu, B., Bu, J. and Chen, C., 2011. Opinion word expansion and target extraction through double propagation. Computational linguistics, 37(1), pp.9-27.

[17] Lalithamani, N., Thati, L.S. and Adhikesavan, R., 2014. Sentence level sentiment polarity calculation for customer reviews by considering complex sentential structures. IJRET: International Journal of Research in Engineering and Technology, 3.

[18] Xu, H., Liu, B., Shu, L. and Yu, P.S., 2018. Double embeddings and cnn-based sequence labeling for aspect extraction. arXiv preprint arXiv:1805.04601.

[19] Shen, Y., Rong, W., Jiang, N., Peng, B., Tang, J. and Xiong, Z., 2017, February. Word embedding based correlation model for question/answer matching. In Thirty-First AAAI Conference on Artificial Intelligence.

[20] Mikolov, T., Le, Q.V. and Sutskever, I., 2013. Exploiting similarities among languages for machine translation. arXiv preprint arXiv:1309.4168.

[21] Mikolov, T., Sutskever, I., Chen, K., Corrado, G.S. and Dean, J., 2013. Distributed representations of words and phrases and their compositionality. In Advances in neural information processing systems (pp. 3111-3119).

[22] Mikolov, T., Chen, K., Corrado, G. and Dean, J., 2013. Efficient estimation of word representations in vector space. arXiv preprint arXiv:1301.3781.

[23] Pennington, J., Socher, R. and Manning, C., 2014. Glove: Global vectors for word representation. In Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP) (pp. 1532-1543).

[24] LeCun, Y. and Bengio, Y., 1995. Convolutional networks for images, speech, and time series. The handbook of brain theory and neural networks, 3361(10), p.1995.

[25] Levy, O., Goldberg, Y. and Dagan, I., 2015. Improving distributional similarity with lessons learned from word embeddings. Transactions of the Association for Computational Linguistics, 3, pp.211-225.

[26] Xu, H., Liu, B., Shu, L. and Yu, P.S., 2018. Lifelong domain word embedding via meta-learning. arXiv preprint arXiv:1805.09991.

[27] Bojanowski, P., Grave, E., Joulin, A. and Mikolov, T., 2017. Enriching word vectors with subword information. Transactions of the Association for Computational Linguistics, 5, pp.135-146.

[28] Sienčnik, S.K., 2015, May. Adapting word2vec to named entity recognition. In Proceedings of the 20th nordic conference of computational linguistics, nodalida 2015, may 11-13, 2015, vilnius, lithuania (No. 109, pp. 239-243). Linköping University Electronic Press.

[29] Liu, P., Joty, S. and Meng, H., 2015. Fine-grained opinion mining with recurrent neural networks and word embeddings. In Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing (pp. 1433-1443).

[30] Wang, S., Mazumder, S., Liu, B., Zhou, M. and Chang, Y., 2018, July. Target-sensitive memory networks for aspect sentiment classification. In Proceedings of the 56th Annual

Meeting of the Association for Computational Linguistics (Volume 1: Long Papers) (pp. 957-967).

[31] Li, X. and Lam, W., 2017, September. Deep multi-task learning for aspect term extraction with memory interaction. In Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing (pp. 2886-2892).

[32] Tsvetkov, Y., Faruqui, M. and Dyer, C., 2016. Correlation-based intrinsic evaluation of word vector representations. arXiv preprint arXiv:1606.06710.

[33] Tsvetkov, Y., Faruqui, M., Ling, W., Lample, G. and Dyer, C., 2015. Evaluation of word vector representations by subspace alignment. In Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing (pp. 2049-2054).

[34] Shen, Y., Rong, W., Sun, Z., Ouyang, Y. and Xiong, Z., 2015, February. Question/answer matching for CQA system via combining lexical and sequential information. In Twenty-Ninth AAAI Conference on Artificial Intelligence.

[35] Faruqui, M., Tsvetkov, Y., Rastogi, P. and Dyer, C., 2016. Problems with evaluation of word embeddings using word similarity tasks. arXiv preprint arXiv:1605.02276.

[36] Pennington, J., Socher, R. and Manning, C.D., 2014, October. Glove: Global vectors for word representation. In Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP) (pp. 1532-1543).

[37] He, R. and McAuley, J., 2016, April. Ups and downs: Modeling the visual evolution of fashion trends with one-class collaborative filtering. In proceedings of the 25th international conference on world wide web (pp. 507-517). International World Wide Web Conferences Steering Committee.

[38] McAuley, J., Targett, C., Shi, Q. and Van Den Hengel, A., 2015, August. Image-based recommendations on styles and substitutes. In Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval (pp. 43-52). ACM.

[39] Panchendrarajan, R., Ahamed, N., Murugaiah, B., Sivakumar, P., Ranathunga, S. and Pemasiri, A., 2016, June. Implicit aspect detection in restaurant reviews using cooccurence of words. In Proceedings of the 7th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis (pp. 128-136).

[40] Feng, J., Cai, S. and Ma, X., 2019. Enhanced sentiment labeling and implicit aspect identification by integration of deep convolution neural network and sequential algorithm. Cluster Computing, 22(3), pp.5839-5857.

# APPENDIX A

**Table 17:** True Positive, False Positive, False Negative and True Negative counts of the evaluation dataset

| Dataset | True Positive | False Positive | False Negative | True Negative |
|---|---|---|---|---|
| Cellular phone[1] | 6253 | 1017 | 1154 | 1432 |
| Cellular phone[2] | 422 | 57 | 71 | 37 |
| DVD player | 520 | 107 | 74 | 138 |
| Mp3 player[1] | 1095 | 191 | 221 | 304 |
| Mp3 player[2] | 6124 | 1162 | 1241 | 1820 |
| Digital camera[1] | 415 | 72 | 67 | 88 |
| Digital camera[2] | 251 | 37 | 49 | 42 |
| Router | 9523 | 2360 | 2014 | 3956 |