informatics

MDPI

Article

Domain-Specific Aspect-Sentiment Pair Extraction Using Rules and Compound Noun Lexicon for Customer Reviews

Noor Rizvana Ahamed Kabeer, Keng Hoon Gan * and Erum Haris

School of Computer Sciences, Universiti Sains Malaysia, Penang 11800, Malaysia;
ana76_rez@yahoo.com (N.R.A.K.); he15_com037@student.usm.my (E.H.)
* Correspondence: khgan@usm.my; Tel.: +604-658-4634

Received: 27 August 2018; Accepted: 22 November 2018; Published: 29 November 2018

Abstract: Online reviews are an important source of opinion to measure products' quality. Hence, automated opinion mining is used to extract important features (aspect) and related comments (sentiment). Extraction of correct aspect-sentiment pairs is critical for overall outcome of opinion mining; however, current works still have limitations in terms of identifying special compound noun and parent-child relationship aspects in the extraction process. To address these problems, an aspect-sentiment pair extraction using the rules and compound noun lexicon (ASPERC) model is proposed. The model consists of three main phases, such as compound noun lexicon generation, aspect-sentiment pair rule generation, and aspect-sentiment pair extraction. The combined approach of rules generated from training sentences and domain specific compound noun lexicon enable extraction of more aspects by firstly identifying special compound noun and parent-child aspects, which eventually contribute to more aspect-sentiment pair extraction. The experiment is conducted with the SemEval 2014 dataset to compare proposed and baseline models. Both ASPERC and its variant, ASPER, result higher in recall (28.58% and 22.55% each) compared to baseline and satisfactorily extract more aspect sentiment pairs. Lastly, the reasonable outcome of ASPER indicates applicability of rules to various domains.

Keywords: aspect extraction; sentiment extraction; lexicon; opinion mining; rule-based; sentiment analysis

1. Introduction

With the proliferation of the World Wide Web, the way people share and seek opinions has changed. Both customers and organizations can get freely available opinions expressed by people all around the world. Sites, such as blogs, social networks, review sites, forums, and e-commerce websites, provide a better understanding of the customer's perceptions towards products and services.

Unfortunately, the opinionated texts, which are meant to be helpful, become overwhelming with the passage of time due to the frequency of posts by users. Hence, an automatic way to mine the unstructured contents in text is needed to get a relevant piece of information for better decision making. As a solution to this problem, opinion mining is introduced by researchers.

Opinion mining or sentiment analysis is a field of study that analyzes people's opinions towards an item (e.g., product) or issue (e.g., service) [1,2]. Aspect based opinion mining relatively gives more detailed insights towards a product or service. This is because a typical customer review will have the description about an aspect of the product or service as well as general feedback about a product or service. The emotion or view expressed in the reviews towards the aspect is known as a sentiment.

Even though the reviews are comprised of vast volumes of sentences, not all the sentences express opinions towards products or services. Therefore, extracting terms that represent opinions in sentences would be useful. The important terms that are present in opinionated sentences are aspect and its corresponding sentiment.

Hence, to represent the opinion of the product or service from a review sentence, aspect-sentiment pair extraction is crucial. The aspect-sentiment pairs extracted from opinion review sentences in Table 1 are represented as the decomposed view as in Table 2. By referring to the decomposed view of opinion review sentences, customers or manufacturers can get a glimpse of the pros and cons of a product or service in a short time.

Table 1. Sample opinion review sentences (Input).

No	Laptop Review Sentences
1	Keyboard is great
2	Mac OS is so simple and easy
3	Nice appearance
4	Simple yet stylish design
5	Design of this laptop is fantastic
6	The Photo Booth is a great program

Table 2. Decomposed view of opinions in review sentences (Output).

No	Aspect	Sentiment
1	Keyboard	Great
2	Mac OS	Simple
	Mac OS	Easy
3	Appearance	Nice
4	Design	Simple
	Design	Stylish
5	Design of laptop	Fantastic
6	Photo Booth program	Great

Aspect-sentiment pair extraction is quite a hard task as it involves processing the sentences written by reviewers in different ways. Hence, there is no predefined solution for this problem. Most of the previous works have extracted aspect-sentiment pairs using syntactic based approaches [3,4]. Syntactic based aspect extraction approaches are usually dependency based [5–8] and rules based [9–11]. Some advanced syntactic based works include [12–14]. However, those works still have limitations in identifying multi-word aspects that consist of a parent-child relationship and special compound noun.

Most of the important aspects are missed in the existing works, due to parsing error in the identification of special compound noun aspects. The parsing error causes misidentification of the aspects as sentiments. For example, in the compound noun, “hard disk”, “hard” is identified as a sentiment and “disk” is identified as an aspect. The parent-child aspects are also missed because of inadequate dependency relations and generalized rules are unable to relate the parent-child aspect to their corresponding sentiment. For example, in the sentence, “The screen of laptop is big”, the aspect, “Screen of laptop”, is not extracted as whole, only screen (child) or laptop (parent) is extracted.

To overcome these drawbacks, in this paper, the aspect-sentiment pair extraction model with rules and lexicon is proposed to improve aspect-sentiment pair extraction. Different types of aspects, such as compound nouns, special compound nouns, and parent-child aspects, are identified using rules and compound noun lexicon. With respect to this, the objectives are:

1. To determine the optimized number of review sentences to train the aspect-sentiment pair extraction model; and
2. To improve the aspect-sentiment pairs extraction via enhancement of rules and compound noun lexicon.

In general, the decomposed view (i.e., the output) that consists of aspect-sentiment pairs of customers' review sentences can be a beneficial utilization for consumers, organizations, and for opinion mining systems. Consumers are no longer bound to read a lot of reviews and can save time when making decisions before proceeding to purchasing products or services. Similarly, product manufacturers can avoid the trivial job of processing and analyzing review sentences to obtain customers' feedback. In addition, this decomposed view can also be applied as an input to opinion mining systems, such as the decision making and recommendation system. It could also function as one of the important steps in a sentiment analysis pipeline, e.g., as input for polarity analysis.

In the remainder of this paper, Section 2 gives the definition of aspect-sentiment opinion mining, and a detailed discussion of existing works on aspect-sentiment pair extraction is given in Section 3. Section 4 introduces the proposed model, including all the processes in each phase of the model. Subsequently, Section 5 describes the conducted experiment and discusses the results. Finally, Section 6 concludes the paper.

2. Aspect-Sentiment Opinion Mining

The key tasks in aspect-sentiment opinion mining are aspect and sentiment extraction given an input text. For example, consider a hotel review, "The room was clean and spacious". The first stage is to detect the aspect which is "room", followed by identifying aspect related sentiment words, such as "clean" and "spacious".

In the literature, an aspect is also addressed as a feature, topic, attribute, and target [15]. An aspect can be grouped into two types, which are an explicit aspect and implicit aspect. Explicit aspects are aspects that are directly denoted in sentences and typically represented by noun and noun phrases [1]. For instance, in the sentence, "The screen of iPhone 6 is very big", the aspect, "screen", is explicitly indicated. Implicit aspects are the opposite of explicit aspects because they are often expressed indirectly in sentences [1]. For example, in the sentence, "The phone is very expensive", the aspect is not directly mentioned, but an implicit aspect of "price" can be inferred.

Aspect-based opinion mining yields very fine-grained information, which is useful for applications in various domains [16]. It gives detailed insights towards products or services. This is because a typical customer review will contain a description about the aspect of the product or service and a general view about a product or service.

Alternatively, a sentiment is interchangeably used as an opinion, attitude, appraisal, emotion, and evaluation [1]. It is generally conveyed through the usage of adjectives. Both implicit and explicit aspects as well as sentiment can be analyzed at either the syntactic (or content) or semantic (or concept) level.

The aspect can constitute a single-word or multi-word terms. Single-word aspects are generally represented by a noun, such as "battery". Whereas, a multi-word aspect is represented by noun phrases, such as "battery life". Multi-word aspects are usually in the form of compound nouns or parent-child aspects. A compound noun is a fixed expression, which is made up of more than one word [17]. Such expressions are frequently combinations of two nouns or a noun and other part of speech (POS), such as an adjective, verb, particle, cardinal, etc. [17]. For example, "battery life" is a noun-noun compound noun and "hard disk" is a special compound noun.

Parent-child aspects consist of an aspect that is general (parent) and an aspect that is a specific (child) part of the general aspect [18]. For example, in the sentence, "The screen of the laptop is wide", "laptop" is the parent and "screen" is the child. Besides that, another example of a sentence with a parent-child relationship is "Durability of battery is awesome", where the parent is "battery" and "durability" is the child.

Although most of the aspect-sentiment opinion mining is performed at the sentence level. It could also be extended to the paragraph or document level when more complex analysis is required. For example, for a sentence, like "It is very cheap.", the extraction of a meaningful aspect (rather than pronoun) will involve neighboring sentences. Hence, the opinion mining is deemed to be carried out at the paragraph level.

3. Related Work

Aspect-sentiment pair extraction in opinion mining has been around for more than a decade. Over the years, many researchers have come up with various solutions to in the extraction of aspect-sentiment pairs. In this work, a syntactic based aspect extraction approach is utilized. Syntax of a language is its grammar. They are the rules or principles to decide the order of word's type or vocabulary in a sentence related to a specific language. For example, in the English language, five nouns cannot be simply put together to be assumed as a correct sentence. Appropriate word order should be applied in a sentence for proper syntax. An example of the correct sequence of a sentence in English is "Ali likes durian", which consists of a subject, verb, and object [19]. However, sequences vary between different languages [19].

Similarly, a syntactic based aspect extraction approach makes use of the sentence structure or term position in a sentence to detect the appearance of an aspect and opinion. There are two ways of using syntax as a measure to extract features, such as dependency parsing and rule based approaches. Some of the related works in dependency parsing are [5–8]. All these works used dependency relations, such as a mod, rcmod, and advmod, and combine them to produce aspect-sentiment pairs. In the relations, the noun or noun phrase is identified as an aspect whilst the adjective is identified as a sentiment. The noun phrase that has been found is a noun-noun compound noun. Due to usage of a parser, some error results in the extraction of aspect-sentiment pairs. If the sentence consists of terms, like "hard disk", the parser tagged "hard" is an adjective and "disk" is a noun. "Hard disk" should be extracted as a term as it is a type of compound noun. This issue causes the approaches to produce aspect-sentiment pairs, where "disk" is the aspect and "hard" is a sentiment.

Furthermore, the parent-child aspects' identification is also less considered in these works. If the parent-child aspect is identified, only either one is assigned to a sentiment. For example, in the sentence, "Customization on Mac is impossible", the parent is "Mac" and the child is "Customization". However, only "Mac" is mapped to "impossible", which is an incomplete mapping of the aspect to its sentiment.

The rule-based method that is most similar to our approach is by [9], this work identifies aspect and sentiment by defining nine rules. The rules contain aspect (noun or noun phrases) and sentiment (adjective) POS tags with their mapping. The rules are the patterns that are learned from sentences of training data. For example, in the ("[DT] NN_1 VB [RB] $JJ_{1.1}$ [CC [RB] $JJ_{1.2}$]"), aspect "NN_1" is mapped to sentiments, "$JJ_{1.1}$" and "$JJ_{1.2}$", resulting in aspect-sentiment pairs of ("NN_1, $JJ_{1.1}$") and ("NN_1, $JJ_{1.2}$"). The general type of rules is unable to identify aspect-sentiment pairs if the sentences contain a POS tag sequence of ("NN RB RB VB JJ"), since the sentence must follow the exact sequence of POS tags in rules.

Moreover, due to POS tagging error, the misidentification of the aspect-sentiment pair also occurs, such as for the case of "hard disk" that also happens in the dependency parsing based method. Hence, a special compound noun is unable to be identified by using this approach. The general rules also do not address parent-child aspects and their mapping.

The incomplete aspect identification in aspect-sentiment pairs means some of the aspect-sentiment pairs are not extracted. Due to a lack of different types of aspects, such as a special compound noun and parent-child aspects, the sentiments associated with these aspects are also not discovered. Although these aspects are addressed by many researchers, none of the them completely resolve both types of aspect identification in aspect-sentiment pair extraction. A close related work by [20] proposed a multilayer attention network, where each layer consists of a couple of attentions, like attention for extracting aspect terms and attention for extracting opinion terms. The pro of this approach is that it avoids the limitation of parsing, i.e., avoiding the compound noun issue, however, a relatively large number of training sentences are required to train the deep learning model.

Until now, none of the approaches have addressed improving aspect identification in aspect-sentiment pairs in both types of multi-word aspects, such as a special compound noun aspect and parent-child aspects with the consideration of little training sentences; this work will resolve this problem.

4. Aspect Sentiment Pair Extraction using Rules and the Compound Noun Lexicon Model

In this paper, a model known as the aspect-sentiment pair extraction model using rules and compound noun lexicon (ASPERC) is proposed. As shown in Figure 1, the model consists of three main phases, such as compound noun lexicon generation, aspect-sentiment pair rule generation, and aspect-sentiment pair extraction.

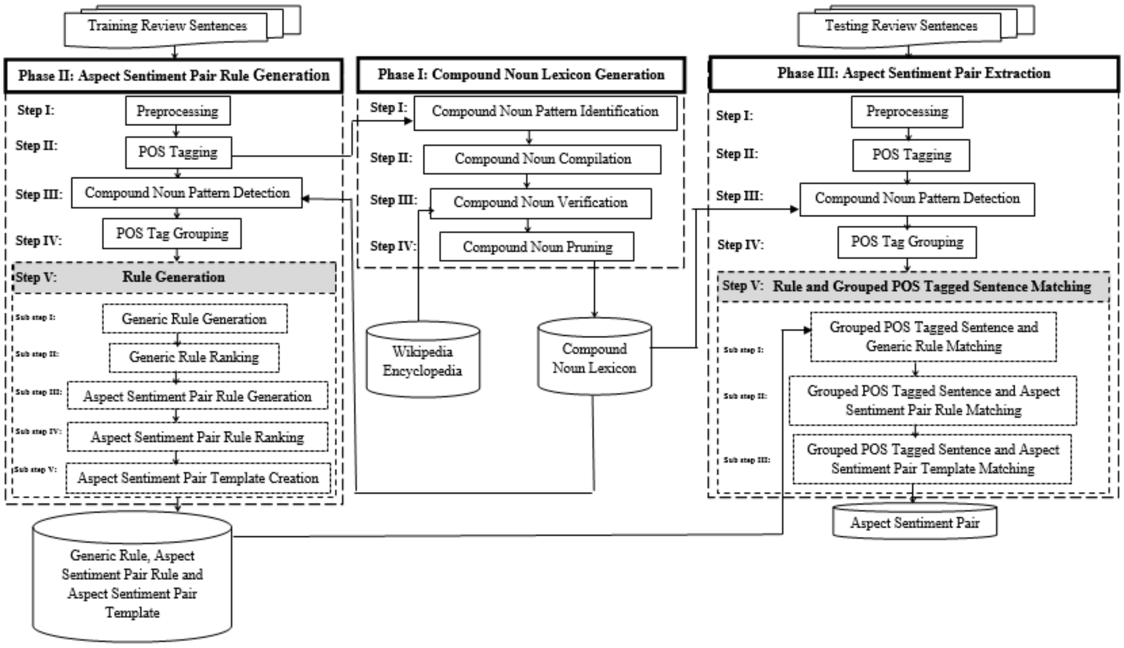

Training Review Sentences
Testing Review Sentences
Phase II: Aspect Sentiment Pair Rule Generation
Step I:
Preprocessing
Step II:
POS Tagging
Step III:
Compound Noun Pattern Detection
Step IV:
POS Tag Grouping
Step V:
Rule Generation
Sub step I:
Generic Rule Generation
Sub step II:
Generic Rule Ranking
Sub step III:
Aspect Sentiment Pair Rule Generation
Sub step IV:
Aspect Sentiment Pair Rule Ranking
Sub step V:
Aspect Sentiment Pair Template Creation
Phase I: Compound Noun Lexicon Generation
Step I:
Compound Noun Pattern Identification
Step II:
Compound Noun Compilation
Step III:
Compound Noun Verification
Step IV:
Compound Noun Pruning
Wikipedia Encyclopedia
Compound Noun Lexicon
Phase III: Aspect Sentiment Pair Extraction
Step I:
Preprocessing
Step II:
POS Tagging
Step III:
Compound Noun Pattern Detection
Step IV:
POS Tag Grouping
Step V: Rule and Grouped POS Tagged Sentence Matching
Sub step I:
Grouped POS Tagged Sentence and Generic Rule Matching
Sub step II:
Grouped POS Tagged Sentence and Aspect Sentiment Pair Rule Matching
Sub step III:
Grouped POS Tagged Sentence and Aspect Sentiment Pair Template Matching
Aspect Sentiment Pair
Generic Rule, Aspect Sentiment Pair Rule and Aspect Sentiment Pair Template

Figure 1. The aspect-sentiment pair extraction using rules and compound noun lexicon (ASPERC) model.

4.1. Phase I: Compound Noun Lexicon Generation

The compound noun lexicon used in the aspect-sentiment extraction model is generated based on the corpus based method. The corpus used in lexicon is the training dataset. A total of 500 training sentences are used to compile laptop and restaurant compound noun lexicon. Two-hundred and fifty laptop training sentences are used to build laptop compound noun lexicon and another 250 are used to build restaurant compound noun lexicon. This is because the compound noun lexicon is domain specific. It requires a domain specific compound noun word from training sentences to be applied into testing sentences.

The amount, 250, follows the number of training sentences that are used to create the rules. Since 250 sentences are utilized to create the rules, the same sentences have also been used to create lexicon.

During the Phase I, Step II (POS tagging), the tags of a compound noun detected by the tagger was observed. It was noted that compound nouns do not always consist of noun ("NN") + noun ("NN") tags. Basically, compound nouns consist of noun phrases. In the noun phrase, one word is a noun and another word can be either a noun or a different POS. The noun in the compound noun ("NN") can be modified by an adjective ("JJ"), noun ("NN"), verb ("VB"), particle ("RP"), cardinal ("CD"), etc. [17].

Hence, the compound noun of different types of POS tag is difficult to be detected by works that regard only a ("NN NN") compound noun to be aspects. However, in this work, besides using

("NN NN") as aspects, the special compound noun is also identified. The identification of other types of compound nouns can simultaneously help in discovering more aspects. As a result, compilation of compound nouns in the lexicon and using it in the model can help to improve aspect identification. In order to generate compound noun lexicon, there are four steps that are carried out. The four steps are: I) Compound nouns pattern identification, II) compound nouns compilation, III) compound nouns verification, and IV) compound nouns pruning. The POS tagged training sentences used are obtained from Phase II. Each of these steps is discussed in detail in this section.

4.1.1. Step I: Compound Nouns' Pattern Identification

In the POS tagged training sentences, the pattern of compound noun lexicon was identified through linguistic inspection. The patterns identified includes ("JJ+ NN+"), ("NN+ CD"), ("VB+ NN+"), and ("VB+ RP"). The plus ("+") sign indicates that the POS tag can have many forms or can have an additional character. For example, the pattern ("NN CD") can be in another form, such as ("NNS CD") and ("NNP CD"). This is because a noun ("NN") can be expressed as singular ("NN"), plural ("NNS"), and a proper noun ("NNP"). Thus, to make a more generic pattern, the common characters are used to represent all the possible forms.

4.1.2. Step II: Compound Nouns Compilation

Using the patterns stated in Table 3, all the occurrences of the words with the POS tags are compiled from the POS tagged training data.

Table 3. Compound noun patterns.

Compound Noun Patterns	Example
JJ+ NN+	Hard/**JJ** disk/**NN**
NN+ CD	Windows/**NN** 7/**CD**
VB+ NN+	Operating/**VBG** system/**NN**
VB+ RP	Boot/**VB** up/**RP**

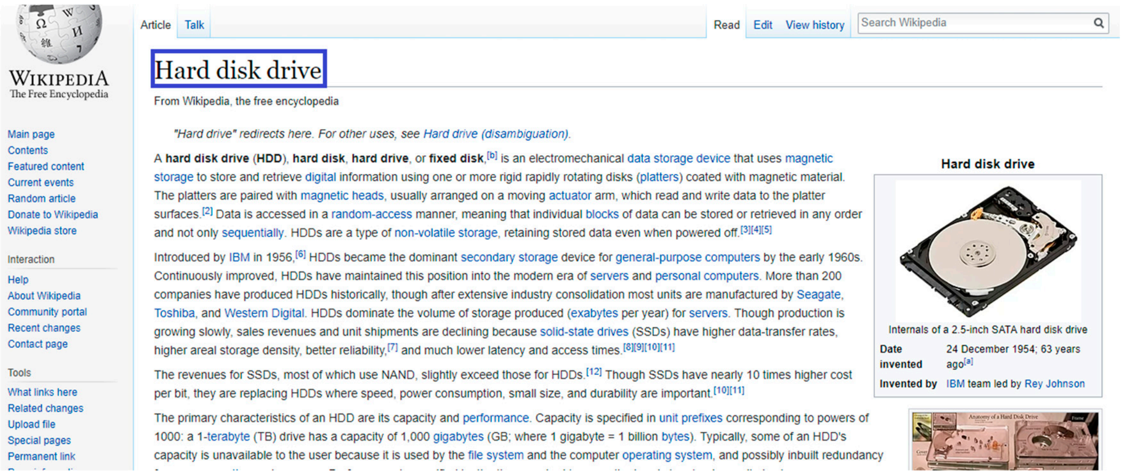
The compound noun patterns are searched using the keyword in context (KWIC) technique [21]. Basically, this technique compiles all the keywords in the dataset and allows the keywords to be searchable with its surrounding words. The AntConc 3.4.4 tool [22] that implements this KWIC technique is used to find compound nouns in the tagged dataset.

Firstly, to search for keywords, the cluster size needs to be defined. Since compound nouns consist of two words, and the tags associated with them are also counted, the cluster size is set to four. For example, "hard_JJ drive_NN" consists of four sequences ("hard", "JJ", "drive", "NN"). Then, the patterns were searched using the search wildcard. For example, to search "hard_JJ drive_NN", wildcards, such as "*_JJ+ *_NN+*", are defined in the search. The output is the list of words with its tags.

As seen in the example, the reason why the cluster size is set to four is the word together with its tag is needed to be displayed. Since a compound noun consists of two words, the corresponding tags of the compound noun are also two. Hence, a total of four clusters are defined to represent the two words and the two tags. Also, compound nouns that are considered consist of a two-word compound noun only; other types of compound noun are not part of this work.

4.1.3. Step III: Compound Nouns Verification

The lists of words with its tags obtained in Step II are verified as in Figures 2 and 3 to determine whether those words are compound nouns. This verification is done by comparing the words in the list with Wikipedia Encyclopedia. Wikipedia is used because of its encyclopedic nature [23]. It contains articles with named entities as well as terms, which is large in size for corpus oriented statistical analysis [23].



Hard disk drive

From Wikipedia, the free encyclopedia

"Hard drive" redirects here. For other uses, see Hard drive (disambiguation).

A **hard disk drive** (**HDD**), **hard disk**, **hard drive**, or **fixed disk**,[b] is an electromechanical data storage device that uses magnetic storage to store and retrieve digital information using one or more rigid rapidly rotating disks (platters) coated with magnetic material. The platters are paired with magnetic heads, usually arranged on a moving actuator arm, which read and write data to the platter surfaces.[2] Data is accessed in a random-access manner, meaning that individual blocks of data can be stored or retrieved in any order and not only sequentially. HDDs are a type of non-volatile storage, retaining stored data even when powered off.[3][4][5]

Introduced by IBM in 1956,[6] HDDs became the dominant secondary storage device for general-purpose computers by the early 1960s. Continuously improved, HDDs have maintained this position into the modern era of servers and personal computers. More than 200 companies have produced HDDs historically, though after extensive industry consolidation most units are manufactured by Seagate, Toshiba, and Western Digital. HDDs dominate the volume of storage produced (exabytes per year) for servers. Though production is growing slowly, sales revenues and unit shipments are declining because solid-state drives (SSDs) have higher data-transfer rates, higher areal storage density, better reliability,[7] and much lower latency and access times.[8][9][10][11]

The revenues for SSDs, most of which use NAND, slightly exceed those for HDDs.[12] Though SSDs have nearly 10 times higher cost per bit, they are replacing HDDs where speed, power consumption, small size, and durability are important.[10][11]

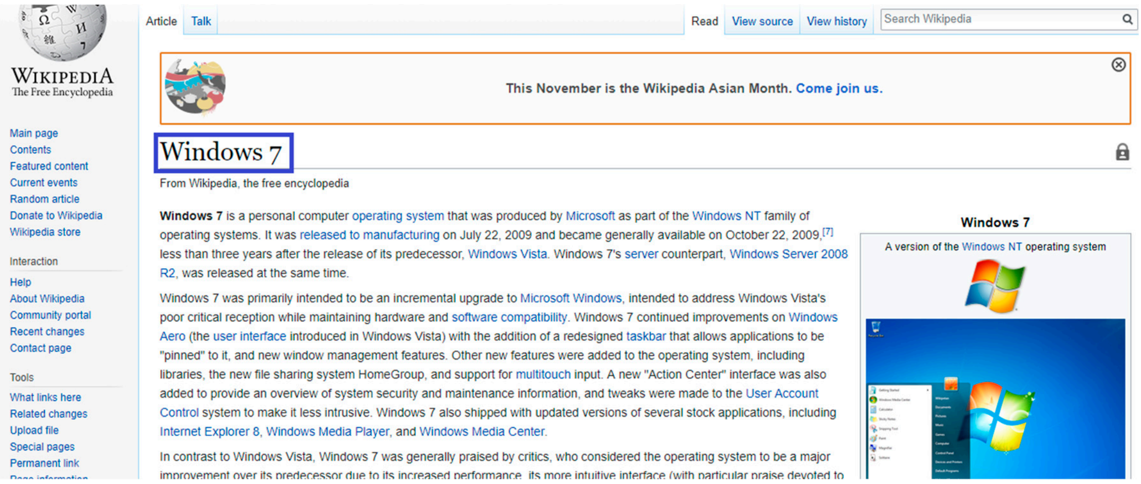
The primary characteristics of an HDD are its capacity and performance. Capacity is specified in unit prefixes corresponding to powers of 1000: a 1-terabyte (TB) drive has a capacity of 1,000 gigabytes (GB; where 1 gigabyte = 1 billion bytes). Typically, some of an HDD's capacity is unavailable to the user because it is used by the file system and the computer operating system, and possibly inbuilt redundancy

Hard disk drive

Internals of a 2.5-inch SATA hard disk drive

Date invented	24 December 1954; 63 years ago[a]
Invented by	IBM team led by Rey Johnson

Figure 2. Verification of compound nouns.



This November is the Wikipedia Asian Month. Come join us.

Windows 7

From Wikipedia, the free encyclopedia

Windows 7 is a personal computer operating system that was produced by Microsoft as part of the Windows NT family of operating systems. It was released to manufacturing on July 22, 2009 and became generally available on October 22, 2009,[7] less than three years after the release of its predecessor, Windows Vista. Windows 7's server counterpart, Windows Server 2008 R2, was released at the same time.

Windows 7 was primarily intended to be an incremental upgrade to Microsoft Windows, intended to address Windows Vista's poor critical reception while maintaining hardware and software compatibility. Windows 7 continued improvements on Windows Aero (the user interface introduced in Windows Vista) with the addition of a redesigned taskbar that allows applications to be "pinned" to it, and new window management features. Other new features were added to the operating system, including libraries, the new file sharing system HomeGroup, and support for multitouch input. A new "Action Center" interface was also added to provide an overview of system security and maintenance information, and tweaks were made to the User Account Control system to make it less intrusive. Windows 7 also shipped with updated versions of several stock applications, including Internet Explorer 8, Windows Media Player, and Windows Media Center.

In contrast to Windows Vista, Windows 7 was generally praised by critics, who considered the operating system to be a major improvement over its predecessor due to its increased performance, its more intuitive interface (with particular praise devoted to

Windows 7

A version of the Windows NT operating system

Figure 3. Verification of compound nouns (Continued).

The compound nouns' words are maintained in the list if the two words in the compound noun exist in a Wikipedia article. The compound noun words can be part of multi-word terms in Wikipedia. This means that a compound noun can occur as a two-word term or as part of more than two-word terms. As long as the compound noun terms occur in either way, they are maintained in the list with their associated tags.

4.1.4. Step 4: Compound Nouns' Pruning

The verified compound nouns in the list are stored, and the compound nouns that do not present in Wikipedia are removed from the list. This final list is stored as compound noun lexicon (Table 4).

4.2. Phase II: Aspect-sentiment Pair Rules Generation

The main idea of this phase is to generate rules based on the training sentences dataset. Several steps are carried out, which finally leads to creation of a knowledge base consisting of aspect-sentiment pair rules and aspect-sentiment pair templates. The steps are described in detail in the following section.

4.2.1. Step I: Preprocessing

Reviews are written by humans. Thus, the sentences can contain many irrelevant non-alphabetic characters, such as ":)", "1)", "1.", "- (only for the beginning of a sentence)". These characters are removed from training sentences so that unnecessary processing during the next phases can be avoided. Besides that, sentence segmentation is also done to split two complete sentences separated by punctuation (e.g., ";", "," and etc.).

Table 4. Compound noun lexicon (only displaying 10 examples each).

Laptop	Restaurant
black_jj model_nn	corned_jj beef_nn
blue_jj screen_nn	fried_jj rice_nn
casual_jj games_nns	black_jj vinegar_nn
chat_jj services_nns	chinese_jj desserts_nns
cooling_jj system_nn	creme_jj brulee_nn
windows_nnp 7_cd	sweet_jj lassi_nns
boot_vb up_rp	smoked_jj salmon_nn
shut_vbd down_rp	green_jj curry_nn
sound_jj card_nn	condensed_jj milk_nn
operating_vbg system_nn	mashed_jj potatoes_nns

4.2.2. Step II: POS Tagging

POS tagging is performed by using the Stanford POS Tagger tool to tag all the training sentences, in which, the part of speech, such as a noun ("NN"), verb ("VB"), adjective ("JJ"), adverb ("RB"), etc., are assigned to each of the words in training sentences. Each of the POS tagged sentences has its original words plus its corresponding tag. The POS tagging is illustrated in Example 4.1.

Example 4.1:
Review sentence: The graphics are stunning.
POS tagged sentence: The/**DT** graphics/**NNS** are/**VBP** stunning/**JJ**./.

4.2.3. Step III: Special Compound Noun Pattern Detection

In this step, the special compound noun patterns, such as ("VB+ NN+"), ("JJ+ NN+"), etc., are searched in the POS tagged sentence. The plus ("+") sign in the pattern indicates that any one character can exist after the tag. For example, in the operating system POS tag ("VB**G** NN"), the additional character is "**G**".

If the pattern is found, then the tags inside the pattern of the POS tagged sentence will be checked. These tags are checked for its validity through compound noun lexicon. For this purpose, only non-noun (not "NN+") tags in the pattern will be compared against the tags in compound noun lexicon. Here, both the tag and the associated word in a POS tagged sentence must be the same as in lexicon. If they are the same, then the matching non-noun tag will be changed to a noun ("NN") tag. Note that only non-noun tags would be changed to nouns if there is a noun tag it remains as it is. This is to enable the application of rules in the sentence. Without changing the non-noun tags to noun tags, the rules cannot be applied to the sentence. If there is no compound noun pattern detected, then, the sentence POS tagged sentence has no changes.

After this process, only the sentences with at least one aspect ("NN") and one sentiment ("JJ") are maintained. This step is to avoid processing sentences that do not have an aspect and sentiment mentioned. An example of how compound noun pattern detection is performed is shown in Example 4.2.

Example 4.2:
Review sentence: *The operating system is solid and friendly to a novice.*
POS tagged sentence: *The/DT operating/**VBG** system/**NN** is/VBZ solid/JJ and/CC friendly/JJ to/TO a/DT novice/**NN**. /.*
Pattern detected: *operating/**VBG** system/**NN***
Matching with lexicon: *operating/**VBG** system/**NN***
Changing POS tag: *operating/**NN** system/**NN***
POS tagged sentence with changed tag: *The/DT operating/**NN** system/**NN** is/VBZ solid/JJ and/CC friendly/JJ to/TO a/DT novice/NN. /.*

Compound Noun Lexicon
operating/VBG
system/NN

4.2.4. Step IV: POS Tag Grouping

It is observed in the training data that an aspect can be a noun or noun phrases. This means that an aspect can be a single-word or multi-word. Similarly, an intensity word, such as an adverb, also occurs more than one time continuously. Both tags are grouped per number of consecutive occurrences.

After grouping, a general tag (nn$_x$ or rb$_x$) is used to represent the grouped noun or adverb tag. Subscript "x" denotes the total number of consecutive occurrences of the noun or adverbs. For example, "also/RB relatively/RB" is grouped "also relatively/*rb*$_2$".

This process is conducted to make the rule generated later to be more generic. The example of grouped POS tagged representation for a single occurrence of a noun and adverb is as in Example 4.3. The example of grouped POS tagged representation for multiple occurrences of nouns and adverbs is as in Example 4.4.

Example 4.3 (Single occurrence of noun and adverb):
Review sentence: *The battery is really long.*
POS tagged sentence: *The/DT battery/**NN** is/VBZ very/**RB** long/JJ. /.*
Grouped POS tagged sentence: *The/DT battery/**nn**$_1$ is/VBZ very/**rb**$_1$ long/JJ./.*

Example 4.4 (Multiple occurrences of noun and adverb):
Review sentence: *The battery life is also relatively good.*
POS tagged sentence: *The/DT battery/**NN** life/**NN** is/VBZ also/**RB** relatively/**RB** good/JJ. /.*
Grouped POS tagged sentence: *The/DT battery life/**nn**$_2$ is/VBZ also relatively/**rb**$_2$ good/JJ. /.*

4.2.5. Step V: Rules' Generation

In this step, the rules, rules ranking as well as the template to extract the aspect-sentiment pairs are explained. There are five sub steps in this step, such as I) generic rule generation, II) generic rule ranking, III) aspect-sentiment pair rule generation, IV) aspect-sentiment pair rule ranking, and V) aspect-sentiment pair template creation. These steps will be discussed in the following sections.

a) Sub Step I: Generic Rule Generation

Using the grouped POS tagged sentences; the tags that appear in every type of sentence are analyzed using linguistic inspection. It is observed that every type of sentence shares some common POS tags among them. For example, common tag sequences that are observed in the sentences includes ("nn VB JJ"), ("JJ nn"), ("JJ IN nn"), ("nn WD VB JJ"), ("JJ JJ nn"), ("nn IN nn JJ"), ("nn JJ"), ("JJ CC JJ nn"), and ("JJ TO nn"). These common tags are assigned as generic rules.

Next, the position of the tags in generic rules is determined in which the aspect ("nn"), sentiment ("JJ"), and also intensity ("rb") are denoted by subscript "a" "b", "c", and etc. The subscript is used to show the position of the aspect, sentiment, or intensity that occurs at the first position ("a"), second position ("b"), third position ("c"), and etc.

The position shows the arrangement of the aspect (“nn”), sentiment (“JJ”), or intensity (“rb”) POS tag that appears in the generic rules. This is important because if the POS tag, such as (“nn”), (“JJ”), and (“rb”), appear more than one time in the generic rule, then the position is important as it will indicate that there are few occurrences of the (“nn”), (“JJ”), and (“rb”).

For example, in the generic rule, (“$\mathrm{JJ_a}$ $\mathrm{JJ_b}$ $\mathrm{nn_{xa}}$”), the subscript, (“a”), indicates the first occurrence of (“JJ”) and subscript, (“b”), indicates the second occurrence of (“JJ”). This position is important for template creation, where the aspect and sentiment will be matched using the template. This is to determine which aspect matches to which sentiment. As in the example, the aspect, (“$\mathrm{nn_{xa}}$”), will be matched with (“$\mathrm{JJ_a}$”) as the first aspect-sentiment pair. Next, aspect (“$\mathrm{nn_{xa}}$”) will be matched with (“$\mathrm{JJ_b}$”) as the second aspect-sentiment pair.

As some generic rules are more important than others, ranking is performed to give weightage to important rules. This is conducted by observing the frequency of appearance of generic rules in the training dataset sentences.

b) Sub Step II: Generic Rule Ranking

Based on the generic rules, all the grouped POS tagged sentences are assigned. The POS tags in generic rules must appear sequentially in the grouped POS tagged sentences and the order must be followed. Table 5 shows the assignment of some sample grouped POS tagged sentences to their generic rules type (Examples 4.5–4.7).

Table 5. Assignment of grouped POS tagged sentences to generic rules.

Example No.	Grouped POS Tagged Sentence	Generic Rules
4.5	*This/DT speed/nn$_1$ is/VBZ fine/JJ. /.*	$\mathrm{nn_{xa}}$ VB $\mathrm{JJ_a}$
4.6	*Good/JJ price/nn$_1$. /.*	$\mathrm{JJ_a}$ $\mathrm{nn_{xa}}$
4.7	*It/PRP’s/VBZ wonderful/JJ for/IN computer gaming/nn$_2$ ./.*	$\mathrm{JJ_a}$ IN $\mathrm{nn_{xa}}$

For the assignment by generic rules, only the first two characters of the POS tags are considered. For example, if a grouped POS tagged sentence has POS tag (“DT nn1 VBZ JJ”) as in Example 4.5, only the first two characters, such as “DT nn1 VB JJ”, are used for categorization. This means similar POS tags (e.g., “VBD”, “VBZ”, “VBP”, and etc.) in grouped POS tag sentences can be generalized to be assigned to generic rules.

After all the grouped POS tagged sentences were assigned by generic rules, the frequencies of assigned grouped POS tagged sentences per generic rules are counted. The generic rules are then ranked by their respective assigned grouped POS tagged sentence frequencies. The rank is ordered in ascending order from highest to lowest frequency. Table 6 shows the generic rules with their corresponding frequency.

Table 6. Generic rules with their corresponding frequency.

Rank	Generic Rules	Frequency of Occurrence
1	$\mathrm{nn_{xa}}$ VB $\mathrm{JJ_a}$	107
2	$\mathrm{JJ_a}$ $\mathrm{nn_{xa}}$	103
3	$\mathrm{JJ_a}$ IN $\mathrm{nn_{xa}}$	22
4	$\mathrm{nn_{xa}}$ IN $\mathrm{JJ_a}$ $\mathrm{nn_{xb}}$	14
5	$\mathrm{nn_{xa}}$ WD VB $\mathrm{JJ_a}$	11
6	$\mathrm{JJ_a}$ $\mathrm{JJ_b}$ $\mathrm{nn_{xa}}$	11
7	$\mathrm{nn_{xa}}$ IN $\mathrm{nn_{xb}}$ $\mathrm{JJ_a}$	9
8	$\mathrm{nn_{xa}}$ $\mathrm{JJ_a}$	6
9	$\mathrm{JJ_a}$ CC $\mathrm{JJ_a}$ $\mathrm{nn_{xa}}$	3
10	$\mathrm{JJ_a}$ TO $\mathrm{nn_{xa}}$	2

c) Sub Step III: Aspect-sentiment Pair Rule Generation

After generic rules are obtained, we can proceed to identify unique rules that are associated to each generic rule. For example, the generic rule, (“$\mathrm{nn}_{\mathrm{xa}}$ VB JJ_{a}”), can have associated unique rules, like (“$\mathrm{nn}_{\mathrm{xa}}$ VB JJ_{a}”), (“$\mathrm{nn}_{\mathrm{xa}}$ VB $\mathrm{rb}_{\mathrm{xa}}$ JJ_{a}”), and (“$\mathrm{nn}_{\mathrm{xa}}$ VB JJ_{a} CC JJ_{b}”).

Besides that, the unique rule generation process is also needed to find the stopping criterion for the number of training sentences required. Hence, the iteration method is used during the unique rule generation process. Firstly, the grouped POS tagged sentences are divided into 50 sentences per iteration. The iteration was repeated six times with different grouped POS tagged sentences.

To identify unique rules, duplicate grouped POS tagged sentences are removed in every iteration. For example, the grouped POS tagged sentences associated with generic rule (“$\mathrm{nn}_{\mathrm{xa}}$ VB JJ_{a}”) are (“$\mathrm{nn}_{\mathrm{xa}}$ VB JJ_{a}”), (“$\mathrm{nn}_{\mathrm{xa}}$ VB $\mathrm{rb}_{\mathrm{xa}}$ JJ_{a}”), and (“$\mathrm{nn}_{\mathrm{xa}}$ VB JJ_{a} CC JJ_{b}”). These sentences can occur many times during the iteration. For instance, (“$\mathrm{nn}_{\mathrm{xa}}$ VB $\mathrm{rb}_{\mathrm{xa}}$ JJ_{a}”) can occur five times during the first iteration. To make it unique, only one occurrence is maintained, and the others are discarded.

The iteration is continued until there is no significant increase in the number of unique grouped POS tagged sentences. Hence, the stopping criterion is at the fifth iteration. The number of training sentences are 250 sentences.

All the unique POS tagged sentences identified until the fifth iteration are assigned as aspect-sentiment pair rules. The number of aspect-sentiment rules in every iteration are shown in Table 7.

Table 7. Iteration and the number of aspect-sentiment rules.

Iteration	Number of Aspect-sentiment Rules
1	21
2	14
3	19
4	5
5	4
6	1

For each aspect-sentiment pair generated, the position of the aspect (“nn”), sentiment (“JJ”), and intensity (“rb”) are denoted by subscript (“a”), (“b”), (“c”), and etc. The position is in accordance with the first appearance in the tag (e.g., “a”) until the subsequent appearances (e.g., “b”, “c”, and etc.). Examples of aspect-sentiment pair rules are shown in Table 8.

Table 8. Generic rules with its corresponding aspect-sentiment pair rule.

Rule Id	Generic Rule Rank	Generic Rule	Aspect-sentiment Pair Rule (R)
R1	1	$\mathrm{nn}_{\mathrm{xa}}$ VB JJ_{a}	$\mathrm{nn}_{\mathrm{xa}}$ VB JJ_{a}
R2	1	$\mathrm{nn}_{\mathrm{xa}}$ VB JJ_{a}	$\mathrm{nn}_{\mathrm{xa}}$ VB $\mathrm{rb}_{\mathrm{xa}}$ JJ_{a}
R3	1	$\mathrm{nn}_{\mathrm{xa}}$ VB JJ_{a}	$\mathrm{nn}_{\mathrm{xa}}$ VB JJ_{a} CC JJ_{b}

d) Sub Step IV: Aspect-sentiment Pair Rule Ranking

The aspect-sentiment pair rules are also ranked with the assigned grouped POS tag sentence. However, in this sub step, all the unique assigned grouped POS tag sentence frequencies are counted. These frequencies are used to rank the aspect-sentiment pair rules. The rank is ordered in ascending order from highest to lowest frequency. Table 9 shows some aspect-sentiment pair rules with their corresponding frequency.

Table 9. Aspect-sentiment pair rule ranking.

Rule Id	Generic Rule Rank	Generic Rule	Aspect-sentiment Pair Rule Rank	Aspect-sentiment Pair Rule (R)
R1	1	nn$_{xa}$ VB JJ$_{a}$	1	nn$_{xa}$ VB JJ$_{a}$
R2	1	nn$_{xa}$ VB JJ$_{a}$	2	nn$_{xa}$ VB rb$_{xa}$ JJ$_{a}$
R3	1	nn$_{xa}$ VB JJ$_{a}$	3	nn$_{xa}$ VB JJ$_{a}$ CC JJ$_{b}$

e) Sub Step V: Aspect-sentiment Pair Template Creation

In the aspect-sentiment pair rules, the main components are the aspect and its corresponding sentiment. An aspect is a noun or noun phrases ("nn$_{x}$"), meanwhile, a sentiment is represented by an adjective ("JJ"). Hence, in the rules, there are many tags that are irrelevant, such as ("VB"), ("WD"), etc. In this step, the aspect and its corresponding sentiment will be linked to produce aspect and sentiment pairs. Furthermore, the aspect-sentiment pair rules also have an additional advantage of identifying the intensity ("rb$_{x}$"), which is usually represented by an adverb.

Basically, an aspect can have one sentiment or more. Similarly, a sentiment can have one aspect or more. Sentiments can also have one or more intensities. Thus, the positions of arrangements of ("a", "b", "c", and etc.) of the aspect, sentiment, and intensity defined in the rules are used in this step to associate the aspect to its corresponding sentiment/s. If an intensity exists before the sentiment, it is also included as additional information. This linking between the aspect and its respective sentiment is obtained through linguistic inspection to produce templates. Examples 4.8–4.10 show the aspect-sentiment pair template created based on the rules.

Example 4.8 (One aspect and one sentiment):
Review sentence: *The touchpad is very intuitive*
POS tagged sentence: *The/DT touchpad/NN is/VBZ very/RB intuitive/JJ*
Grouped POS tagged sentence: *The/DT touchpad/nn$_{1}$ is/VBZ very/rb$_{1}$ intuitive/JJ*
Generic rule: *nn$_{xa}$ VB JJ$_{a}$*
Aspect-sentiment pair rule (R2): *nn$_{xa}$ VB rb$_{xa}$ JJ$_{a}$*
Aspect-sentiment pair template creation: *(**nn**$_{xa}$; rb$_{xa}$; **JJ**$_{a}$)*

Example 4.9 (One sentiment and two aspects):
Review sentence: *Amazing colors and resolution.*
POS tagged sentence: *Amazing/JJ colors/NNS and/CC resolution/NN. /.*
Grouped POS tagged sentence: *Amazing/JJ (colors/NNS)/nn$_{1}$ and/CC (resolution/NN)/nn$_{1}$. /.*
Generic rule: *JJ$_{a}$ nn$_{xa}$*
Aspect-sentiment pair rule (R23): *JJ$_{a}$ nn$_{xa}$ CC nn$_{xb}$*
Aspect-sentiment pair template creation: *(nn$_{xa}$; ; JJ$_{a}$), (**nn**$_{xb}$; ; **JJ**$_{a}$)*

Example 4.10 (Two sentiments and one aspect):
Review sentence: *Super fast processor*
POS tagged sentence: *Super/JJ fast/JJ processor/NN*
Grouped POS tagged sentence: *Super/JJ fast /JJ (processor/NN)/nn$_{1}$. /.*
Generic rule: *JJ$_{a}$ JJ$_{b}$ nn$_{xa}$*
Aspect-sentiment pair rule (R52): *JJ$_{a}$ JJ$_{b}$ nn$_{xa}$*
Aspect-sentiment pair template creation: *(**nn**$_{xa}$; ; **JJ**$_{a}$), (**nn**$_{xa}$; ; **JJ**$_{b}$)*

Finally, all the generic rules, aspect-sentiment pair rules, and aspect-sentiment pair template are stored inside a knowledge base as shown in Table 10.

Table 10. Generic rules, aspect-sentiment pair rules, and aspect-sentiment pair templates.

Rule Id	Generic Rule Rank	Generic Rule	Aspect Sentiment Pair Rule Rank	Aspect Sentiment Pair Rule	Aspect Sentiment Pair Template (Aspect; Intensity; Sentiment)
R1	1	nn$_{xa}$ VB JJ$_a$	1	nn$_{xa}$ VB JJ$_a$	(nn$_{xa}$; ; JJ$_a$)
R2	1	nn$_{xa}$ VB JJ$_a$	2	nn$_{xa}$ VB rb$_{xa}$ JJ$_a$	(nn$_{xa}$; rb$_{xa}$; JJ$_a$)
R3	1	nn$_{xa}$ VB JJ$_a$	3	nn$_{xa}$ VB JJ$_a$ CC JJ$_b$	(nn$_{xa}$; ; JJ$_a$), (nn$_{xa}$; ; JJ$_b$),
R4	1	nn$_{xa}$ VB JJ$_a$	4	nn$_{xa}$ VB DT JJ$_a$ nn$_{xb}$	(nn$_{xa}$ nn$_{xb}$; ; JJ$_a$)
R5	1	nn$_{xa}$ VB JJ$_a$	5	nn$_{xa}$ VB rb$_{xa}$ JJ$_a$ CC JJ$_b$	(nn$_{xa}$; rb$_{xa}$; JJ$_a$), (nn$_{xa}$; ; JJ$_b$)
R6	1	nn$_{xa}$ VB JJ$_a$	6	nn$_{xa}$ VB JJ$_a$ nn$_{xb}$	(nn$_{xa}$ nn$_{xb}$; ; JJ$_a$)
R7	1	nn$_{xa}$ VB JJ$_a$	7	nn$_{xa}$ VB DT rb$_{xa}$ JJ$_a$	(nn$_{xa}$; rb$_{xa}$; JJ$_a$)
R8	1	nn$_{xa}$ VB JJ$_a$	8	nn$_{xa}$ VB JJ$_a$ TO nn$_{xb}$	(nn$_{xa}$ nn$_{xb}$; ; JJ$_a$)
R9	1	nn$_{xa}$ VB JJ$_a$	9	nn$_{xa}$ VB DT JJ$_a$	(nn$_{xa}$; ; JJ$_a$)
R10	1	nn$_{xa}$ VB JJ$_a$	10	nn$_{xa}$ VB rb$_{xa}$ JJ$_a$ CC rb JJ$_b$	(nn$_{xa}$; rb$_{xa}$; JJ$_a$), (nn$_{xa}$; rb$_{xb}$; JJ$_b$)
R11	1	nn$_{xa}$ VB JJ$_a$	11	nn$_{xa}$ CC DT nn$_{xb}$ VB JJ$_a$	(nn$_{xa}$; ; JJ$_a$) , (nn$_{xb}$; ; JJ$_a$)
R12	1	nn$_{xa}$ VB JJ$_a$	12	nn$_{xa}$ CC nn$_{xb}$ VB DT JJ$_a$	(nn$_{xa}$; ; JJ$_a$) , (nn$_{xb}$; ; JJ$_a$)
R13	1	nn$_{xa}$ VB JJ$_a$	13	nn$_{xa}$ CC nn$_{xb}$ VB rb$_{xa}$ JJ$_a$	(nn$_{xa}$; rb$_{xa}$; JJ$_a$) , (nn$_{xb}$; rb$_{xa}$; JJ$_a$)
R14	1	nn$_{xa}$ VB JJ$_a$	14	nn$_{xa}$ VB DT JJ$_a$ CC JJ$_b$	(nn$_{xa}$; ; JJ$_a$), (nn$_{xa}$; ; JJ$_b$)
R15	1	nn$_{xa}$ VB JJ$_a$	15	nn$_{xa}$ VB rb$_{xa}$ JJ$_a$ TO nn$_{xb}$	(nn$_{xa}$ nn$_{xb}$; rb$_{xa}$; JJ$_a$)
R16	1	nn$_{xa}$ VB JJ$_a$	16	nn$_{xa}$ VB rb$_{xa}$ DT JJ$_a$	(nn$_{xa}$; rb$_{xa}$; JJ$_a$)
R17	1	nn$_{xa}$ VB JJ$_a$	17	nn$_{xa}$ VB rb$_{xa}$ JJ$_a$ nn$_{xb}$	(nn$_{xa}$ nn$_{xb}$; rb$_{xa}$; JJ$_a$)
R18	2	JJ$_a$ nn$_{xa}$	1	JJ$_a$ nn$_{xa}$	(nn$_{xa}$; ; JJ$_a$)
R19	2	JJ$_a$ nn$_{xa}$	2	rb$_{xa}$ JJ$_a$ nn$_{xa}$	(nn$_{xa}$; rb$_{xa}$; JJ$_a$)
R20	2	JJ$_a$ nn$_{xa}$	3	JJ$_a$ nn$_{xa}$ IN nn$_{xb}$	(nn$_{xa}$ IN nn$_{xb}$; ; JJ$_a$)
R21	2	JJ$_a$ nn$_{xa}$	4	JJ$_a$ nn$_{xa}$ IN DT nn$_{xb}$	(nn$_{xa}$ IN nn$_{xb}$; ; JJ$_a$)
R22	2	JJ$_a$ nn$_{xa}$	5	JJ$_a$ nn$_{xa}$ IN PR nn$_{xb}$	(nn$_{xa}$ IN nn$_{xb}$; ; JJ$_a$)
R23	2	JJ$_a$ nn$_{xa}$	6	JJ$_a$ nn$_{xa}$ CC nn$_{xb}$	(nn$_{xa}$; ; JJ$_a$), (nn$_{xb}$; ; JJ$_a$)
R24	2	JJ$_a$ nn$_{xa}$	7	JJ$_a$ nn$_{xa}$ CC nn$_{xb}$ CC nn$_{xc}$	(nn$_{xa}$; ; JJ$_a$), (nn$_{xb}$; ; JJ$_a$) ,(nn$_{xc}$; ; JJ$_a$)
R25	2	JJ$_a$ nn$_{xa}$	8	JJ$_a$ nn$_{xa}$ IN nn$_{xb}$ CC nn$_{xc}$ rb$_{xa}$ DT JJ$_a$ nn$_{xa}$	(nn$_{xa}$ IN nn$_{xb}$; ; JJ$_a$), (nn$_{xa}$ IN nn$_{xc}$; ; JJ$_a$)
R26	2	JJ$_a$ nn$_{xa}$	9	nn$_{xa}$ VB JJ$_a$ IN DT nn$_{xb}$	(nn$_{xa}$; rb$_{xa}$; JJ$_a$)
R27	3	JJ$_a$ IN nn$_{xa}$	1	JJ$_a$ IN nn$_{xa}$	(nn$_{xa}$ IN nn$_{xb}$; ; JJ$_a$)
R28	3	JJ$_a$ IN nn$_{xa}$	2	nn$_{xa}$ JJ$_a$ IN nn$_{xb}$	(nn$_{xa}$; ; JJ$_a$)
R29	3	JJ$_a$ IN nn$_{xb}$	3	rb$_{xa}$ JJ$_a$ IN nn$_{xa}$	(nn$_{xa}$ IN nn$_{xb}$; ; JJ$_a$)
R30	3	JJ$_a$ IN nn$_{xb}$	4	nn$_{xa}$ VB rb$_{xa}$ JJ$_a$ IN nn$_{xb}$ CC nn$_{xc}$	(nn$_{xa}$; rb$_{xa}$; JJ$_a$)
R31	3	JJ$_a$ IN nn$_{xb}$	5	nn$_{xa}$ VB rb$_{xa}$ JJ$_a$ IN DT nn$_{xb}$	(nn$_{xa}$ IN nn$_{xb}$; rb$_{xa}$; JJ$_a$), (nn$_{xa}$ IN nn$_{xc}$; rb$_{xa}$; JJ$_a$)
R32	3	JJ$_a$ IN nn$_{xb}$	6	nn$_{xa}$ VB rb$_{xa}$ JJ$_a$ IN DT nn$_{xb}$	(nn$_{xa}$ IN nn$_{xb}$; rb$_{xa}$; JJ$_a$)
R33	3	JJ$_a$ IN nn$_{xb}$	7	nn$_{xa}$ rb$_{xa}$ JJ$_a$ IN DT nn$_{xb}$	(nn$_{xa}$ IN nn$_{xb}$; rb$_{xa}$; JJ$_a$)
R34	3	JJ$_a$ IN nn$_{xb}$	8	nn$_{xa}$ VB rb$_{xa}$ JJ$_a$ IN nn$_{xb}$	(nn$_{xa}$ IN nn$_{xb}$; rb$_{xa}$; JJ$_a$)
R35	3	JJ$_a$ IN nn$_{xb}$	9	nn$_{xa}$ rb$_{xa}$ JJ$_a$ IN nn$_{xb}$	(nn$_{xa}$ IN nn$_{xb}$; rb$_{xa}$; JJ$_a$)
R36	3	JJ$_a$ IN nn$_{xb}$	10	nn$_{xa}$ VB JJ$_a$ IN nn$_{xb}$	(nn$_{xa}$ IN nn$_{xb}$; ; JJ$_a$)
R37	3	JJ$_a$ IN nn$_{xa}$	11	rb$_{xa}$ JJ$_a$ IN DT nn$_{xa}$	(nn$_{xa}$; rb$_{xa}$; JJ$_a$)
R38	3	JJ$_a$ IN nn$_{xa}$	12	rb$_{xa}$ JJ$_a$ IN PR nn$_{xa}$	(nn$_{xa}$; rb$_{xa}$; JJ$_a$)
R39	3	JJ$_a$ IN nn$_{xa}$	13	rb$_{xa}$ JJ$_a$ nn$_{xa}$ IN nn$_{xb}$	(nn$_{xa}$ IN nn$_{xb}$; rb$_{xa}$; JJ$_a$)
R40	3	JJ$_a$ IN nn$_{xb}$	14	JJ$_a$ IN DT nn$_{xa}$	(nn$_{xa}$; ; JJ$_a$)
R41	4	nn$_{xa}$ IN JJ$_a$ nn$_{xb}$	1	nn$_{xa}$ IN JJ$_a$ nn$_{xb}$	(nn$_{xa}$ IN nn$_{xb}$; ; JJ$_a$)
R42	4	nn$_{xa}$ IN JJ$_a$ nn$_{xb}$	2	nn$_{xa}$ IN DT JJ$_a$ nn$_{xb}$	(nn$_{xa}$ IN nn$_{xb}$; ; JJ$_a$)
R43	4	nn$_{xa}$ IN JJ$_a$ nn$_{xb}$	3	nn$_{xa}$ IN rb$_{xa}$ JJ$_a$ nn$_{xb}$	(nn$_{xa}$ IN nn$_{xb}$; rb$_{xa}$; JJ$_a$)
R44	5	nn$_{xa}$ WD VB JJ$_a$	1	nn$_{xa}$ WD VB DT JJ$_a$ nn$_{xb}$	(nn$_{xa}$ nn$_{xb}$; ; JJ$_a$)
R45	5	nn$_{xa}$ WD VB JJ$_a$	2	nn$_{xa}$ WD VB rb$_{xa}$ JJ$_a$	(nn$_{xa}$; rb$_{xa}$; JJ$_a$)
R46	5	nn$_{xa}$ WD VB JJ$_a$	3	nn$_{xa}$ WD VB JJ$_a$	(nn$_{xa}$; ; JJ$_a$)
R47	5	nn$_{xa}$ WD VB JJ$_a$	4	nn$_{xa}$ WD VB IN DT nn$_{xb}$ VB rb$_{xa}$ JJ$_a$	(nn$_{xa}$ IN nn$_{xb}$; rb$_{xa}$; JJ$_a$)
R48	5	nn$_{xa}$ WD VB JJ$_a$	5	nn$_{xa}$ WD VB IN DT JJ$_a$	(nn$_{xa}$; ; JJ$_a$)
R49	5	nn$_{xa}$ WD VB JJ$_a$	6	nn$_{xa}$ WD VB IN JJ$_a$ nn$_{xb}$	(nn$_{xa}$ IN nn$_{xb}$; ; JJ$_a$)
R50	5	nn$_{xa}$ WD VB JJ$_a$	7	nn$_{xa}$ WDT VB JJ$_a$ JJ$_b$ nn$_{xb}$	(nn$_{xa}$ nn$_{xb}$; ; JJ$_a$), (nn$_{xa}$ nn$_{xb}$; ; JJ$_b$)
R51	5	nn$_{xa}$ WD VB JJ$_a$	8	nn$_{xa}$ WD VB IN JJ$_a$	(nn$_{xa}$; ; JJ$_a$)

Table 10. *Cont.*

Rule Id	Generic Rule Rank	Generic Rule	Aspect Sentiment Pair Rule Rank	Aspect Sentiment Pair Rule	Aspect Sentiment Pair Template (Aspect; Intensity; Sentiment)
R52	6	JJ$_a$ JJ$_b$ nn$_{xa}$	1	JJ$_a$ JJ$_b$ nn$_{xa}$	nn$_{xa}$; ; JJ$_a$), (nn$_{xa}$; ; JJ$_b$)
R53	6	JJ$_a$ JJ$_b$ nn$_{xa}$	2	rb$_{xa}$ DT JJ$_a$ JJ$_b$ nn$_{xa}$	(nn$_{xa}$; rb$_{xa}$; JJ$_a$), (nn$_{xa}$; ; JJ$_b$)
R54	7	nn$_{xa}$ IN nn$_{xb}$ JJ$_a$	1	nn$_{xa}$ IN DT nn$_{xb}$ VB rb$_{xa}$ JJ$_a$	(nn$_{xa}$ IN nn$_{xb}$; rb$_{xa}$; JJ$_a$)
R55	7	nn$_{xa}$ IN nn$_{xb}$ JJ$_a$	2	nn$_{xa}$ IN DT nn$_{xb}$ VB JJ$_a$	(nn$_{xa}$ IN nn$_{xb}$; ; JJ$_a$)
R56	7	nn$_{xa}$ IN nn$_{xb}$ JJ$_a$	3	nn$_{xa}$ IN nn$_{xb}$ VB rb$_{xa}$ DT JJ$_a$	(nn$_{xa}$ IN nn$_{xb}$; rb$_{xa}$; JJ$_a$)
R57	7	nn$_{xa}$ IN nn$_{xb}$ JJ$_a$	4	nn$_{xa}$ IN nn$_{xb}$ rb$_{xa}$ VB JJ$_a$	(nn$_{xa}$ IN nn$_{xb}$; rb$_{xa}$; JJ$_a$)
R58	8	nn$_{xa}$ JJ$_a$	1	nn$_{xa}$ rb$_{xa}$ JJ$_a$	(nn$_{xa}$; rb$_{xa}$; JJ$_a$)
R59	8	nn$_{xa}$ JJ$_a$	2	nn$_{xa}$ JJ$_a$	(nn$_{xa}$; ; JJ$_a$)
R60	9	JJ$_a$ CC JJ$_b$ nn$_{xa}$	1	JJ$_a$ CC JJ$_b$ nn$_{xa}$	(nn$_{xa}$; ; JJ$_a$), (nn$_{xa}$; ; JJ$_b$)
R61	9	JJ$_a$ CC JJ$_b$ nn$_{xa}$	2	rb$_{xa}$ JJ $_a$ CC JJ$_b$ nn$_{xa}$	(nn$_{xa}$; rb$_{xa}$; JJ$_a$), (nn$_{xa}$; ; JJ$_b$)
R62	10	JJ $_a$ TO nn$_{xa}$	1	rb$_{xa}$ JJ$_a$ TO nn$_{xa}$	(nn$_{xa}$; rb$_{xa}$; JJ$_a$)
R63	10	JJ $_a$ TO nn$_{xa}$	2	JJ$_a$ TO nn$_{xa}$	(nn$_{xa}$; ; JJ$_a$)

4.3. Phase III: Aspect-sentiment Pair Extraction

The motive of this phase is to extract aspect-sentiment pairs based on testing sentences dataset. Few steps are involved to match the testing sentences to the rules and templates in the knowledge base created in Phase II and produce aspect-sentiment pairs. Step I until Step IV in Phase III is similar with Phase II, except Step V. The difference is, in Phase I, training sentences are used, whereas, in Phase III, testing sentences are used.

4.3.1. Step I: Preprocessing

In this step, a cleaning process is done to the testing sentences in the dataset. Some of the irrelevant tokens, such as ":)", "1)", "1.", "- (only for beginning of sentence)", are removed from the sentences. Besides that, sentence segmentation is also done to split two complete sentences separated by punctuation (e.g., ";", ",", and etc.).

4.3.2. Step II: POS Tagging

The testing sentences are parsed using the Stanford Parser tool to obtain the tagged words for each sentence. For example:

Example 4.11:
Review sentence: *The technical support was not helpful as well.*
POS tagged sentence: *The/**DT** technical/**JJ** support/**NN** was/**VBD** not/**RB** helpful/**JJ** as/**IN** well/**RB**. /.*

4.3.3. Step III: Compound Noun Pattern Detection

The compound noun pattern is detected in the POS tagged sentence. If the pattern exists, then, the non-noun tag (not "NN+") is changed to noun tag ("NN"). Next, only the sentences with at least one aspect ("NN") and one sentiment ("JJ") are maintained.

Example 4.12:
Review sentence: *The technical support was not helpful as well.*
POS tagged sentence: *The/**DT** technical/**JJ** support/**NN** was/**VBD** not/**RB** helpful/**JJ** as/**IN** well/**RB**. /.*
Pattern detected: *technical/**JJ** support/**NN***

Matching with lexicon: *technical/**JJ** service/**NN*** ⇦ Compound Noun Lexicon
Changing pattern tag: *technical/**NN** service/**NN***
POS tagged sentence with changed tag: *The/DT technical/**NN** support/**NN** was/VBD not/RB helpful/JJ as/IN well/RB. /.*

4.3.4. Step IV: POS Tag Grouping

The POS tags that occur consecutively in the POS tagged sentences are grouped together. These grouped tags are denoted by new tags (e.g., nn_x, rb_x, and etc.) indicating there are multiple tags inside it.

Example 4.13:
Review sentence: *The technical support was not helpful as well.*
POS tagged sentence: *The/DT (technical/**NN** support/**NN**) was/VBD (not/**RB**) helpful/JJ as/IN (well/**RB**). /.*
Grouped POS tagged sentence: *The/DT technical support/**nn**$_2$ was/VBD not/**rb**$_1$ helpful/JJ as/IN well/**rb**$_1$. /.*

4.3.5. Step V: Rule and Grouped POS Tagged Sentence Matching

In this step, the grouped POS tagged testing sentences will be used to match the rules in the knowledge base. In the matching process, only the first two characters of the POS tag are used. For example, ("NNS") in grouped POS tagged sentences have three characters inside the tag, but for matching with rules, only the first two characters, such as ("NN"), are used. This is to allow the testing sentences with various types ("NN", "NNS", "NNP", etc.) of the same tag ("NN") to match with the rules.

The sub steps involved are: a) Grouped POS tagged sentences and generic rules matching, b) grouped POS tagged sentences and aspect-sentiment pair rules matching, and c) grouped POS tagged sentences and aspect-sentiment template matching.

a) Sub Step I: Grouped POS Tagged Sentence and Generic Rule Matching

In this sub step, the grouped POS tagged sentences will be matched with all the generic rules produced in Phase (II). The purpose is to shortlist the generic rule that can be applied to grouped POS tagged sentences. Matching is performed from the generic rule rank one until the generic rule rank ten. In the matching process, each of the POS tags in grouped POS tagged sentences is compared with the generic rules POS tags.

There can be optional POS tags to the left and right of the matching tag of the sentence with a generic rule tag. In Example 1, there is an optional tag ("DT") to the left of the matching tag ("nn$_2$"), which matches with the generic rule tag ("nn$_{xa}$").

However, when a sentence is matched, the order of the POS tag inside the generic rule must be followed. As in Example 1, the first tag ("nn$_{xa}$") must match with the tag of the sentence, and then only the next tag is matched ("VB").

In short, the sequence of the generic rule POS tags must be in order. All the tags of the sentence must match with the tags of generic rules. This is to shortlist the generic rules that match with the sentence.

Also, for matching purposes, only the first two characters of the POS tag is taken. For example, only ("NN") is matched in the POS tag ("NNS"). In the POS tag, ("nn$_2$"), only ("nn") is matched. However, there are no changes made to the original tag. It is retained as it is. The example is as shown.

Example 4.14 (Matching with generic rule rank 1):
Grouped POS tagged sentence: *The/DT technical support/nn$_2$ was/VBD not/rb$_1$ helpful/JJ as/IN well/rb$_1$. /.*

Matching of POS tag of grouped POS tagged sentence with tags of generic rule is conducted by looping each POS tag in the sentence by the sequence it appears.

i. "DT"

POS tags of grouped POS tagged sentence: DT nn$_2$ VBD rb$_1$ JJ IN rb$_1$

POS tags of generic rule rank 1: nn$_{xa}$ VB JJ$_a$

Outcome: "DT" is compared with "nn$_{xa}$", "VB" and "JJ$_a$". There is no match found. Next tag is matched.

ii. “nn$_2$”

POS tags of grouped POS tagged sentence: DT nn$_2$ VBD rb$_1$ JJ IN rb$_1$

POS tags of generic rule rank 1: nn$_{xa}$ VB JJ$_a$

Outcome: nn$_2$ is compared with “nn$_{xa}$”. The match is found. Next tag is matched.

iii. “VBD”

POS tags of grouped POS tagged sentence: DT nn$_2$ VBD rb$_1$ JJ IN rb$_1$

POS tags of generic rule rank 1: nn$_{xa}$ VB JJ$_a$

Outcome: “VBD” is compared with “VB” because “nn$_{xa}$” had already matched previously. The match is found. Next tag is matched.

iv. “rb$_1$”

POS tags of grouped POS tagged sentence: DT nn$_2$ VBD rb$_1$ JJ IN rb$_1$

POS tags of generic rule rank 1: nn$_{xa}$ VB JJ$_a$

Outcome: “rb$_1$” is compared with “JJ$_a$” because “VB” had already matched previously. No match. Next tag is matched.

v. “JJ”

POS tags of grouped POS tagged sentence: DT nn$_2$ VBD rb$_1$ JJ IN rb$_1$

POS tags of generic rule rank 1: nn$_{xa}$ VB JJ$_a$

Outcome: “JJ” is compared with “JJ$_a$”. The match is found. All the tags in rule has been matched. The next tag is matched starting from first POS of rule again.

vi. “IN”

POS tags of grouped POS tagged sentence: DT nn$_2$ VBD rb$_1$ JJ IN rb$_1$

POS tags of generic rule rank 1: nn$_{xa}$ VB JJ$_a$

Outcome: “IN” is compared with “nn$_{xa}$”, “VB” and “JJ$_a$”. No match. The next tag is matched.

vii. “rb$_1$”

POS tags of grouped POS tagged sentence: DT nn$_2$ VBD rb$_1$ JJ IN rb$_1$

POS tags of generic rule rank 1: nn$_{xa}$ VB JJ$_a$

Outcome: “rb$_1$” is compared with ”nn$_{xa}$”, “VB”, and “JJ$_a$”. No match. All the tags in the sentence have been matched with the rule tags.

A similar way of matching has been performed with all the generic rules. The sentence matches two generic rules with rank one and rank eight. The matched generic rules are taken as a shortlisted rule. The non-matching rules are discarded. Table 11 shows all the matching outcomes with generic rules.

b) Sub Step II: Grouped POS Tagged Sentence and Aspect-sentiment Pair Rule Matching

The aspect-sentiment pair rules associated with the shortlisted generic rules will be used to perform matching with the grouped POS tagged sentence. For this purpose, similarity scores between the sentence and each aspect-sentiment rule are calculated.

A Levenshtein distance [24] is used as the similarity matching technique. This technique computes differences between two sequences. The two sequences are the source sequence and target sequence. In this case, the grouped POS tagged sentence is the source sequence and the aspect-sentiment pair rule is the target sequence.

The Levenshtein distance between two sequences is the minimum number of editing operations required to transform the source sequence to the target sequence [24]. The editing operations involved are insertions, deletions, and substitutions.

Each of the editing operations had their assigned cost. Insertion and deletion has a cost of one, whereas, substitution costs two [24]. This is because substitution involves both deletion and insertion.

Table 11. Matching of grouped POS tagged sentence with generic rules.

Grouped POS Tagged Sentence	Generic Rule Rank	Generic Rule	Matching
DT nn_2 VBD rb_1 JJ IN rb_1	1	nn_{xa} VB JJ_a	Match
DT nn_2 VBD rb_1 JJ IN rb_1	2	JJ_a nn_{xa}	No match
DT nn_2 VBD rb_1 JJ IN rb_1	3	JJ_a IN nn_{xa}	No match
DT nn_2 VBD rb_1 JJ IN rb_1	4	nn_{xa} IN JJ_a nn_{xb}	No match
DT nn_2 VBD rb_1 JJ IN rb_1	5	nn_{xa} WD VB JJ_a	No match
DT nn_2 VBD rb_1 JJ IN rb_1	6	JJ_a JJ_b nn_{xa}	No match
DT nn_2 VBD rb_1 JJ IN rb_1	7	nn_{xa} IN nn_{xb} JJ_a	No match
DT nn_2 VBD rb_1 JJ IN rb_1	8	nn_{xa} JJ_a	Match
DT nn_2 VBD rb_1 JJ IN rb_1	9	JJ_a CC JJ_b nn_{xa}	No match
DT nn_2 VBD rb_1 JJ IN rb_1	10	JJ_a TO nn_{xa}	No match

The distances between a grouped POS tagged sentence and rules are measured by calculating the total cost involved to transform the sentence into rules. After computing the cost, the similarity score is calculated as follows:

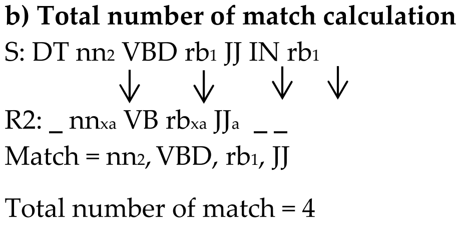
$$Similarity\ score = \frac{1}{total\ cost + 1} \times total\ number\ of\ matches$$

This formula states that, in this work, the preference for a similarity score is low cost and high number of matches. Hence, a higher matching score is desirable when matching the sentence with the rules.

In Sub step III), the shortlisted generic rules are rank one and rank eight generic rules. Rank one and rank eight has their own associated aspect-sentiment rules.

In Example 4.15 a), the calculation of the total cost between a grouped POS tagged sentence and rank two aspect-sentiment pair rule is shown. In Example 4.15 b), the calculation of the total number of matches between a grouped POS tagged sentence and rank two aspect-sentiment pair rule is shown. In Example 4.15 c), the similarity score calculation is shown.

Example 4.15 (Calculation of total cost between grouped POS tagged sentence):
Grouped POS tagged sentence: *The/DT technical support/nn_2 was/VBD not/rb_1 helpful/JJ as/IN well/rb_1. /.*
a) Total Cost Calculation
POS tags of grouped POS tagged sentence = S
POS tags of aspect-sentiment rule (R2) rank 2 = R2
S: DT nn_2 VBD rb_1 JJ IN rb_1
R2: _ nn_{xa} VB rb_{xa} JJ_a _ _
Operation: delete ("DT")
delete ("IN")
delete ("rb_1")
Total cost = 1 + 1 + 1
= 3



b) Total number of match calculation
S: DT nn_2 VBD rb_1 JJ IN rb_1
↓ ↓ ↓ ↓
R2: _ nn_{xa} VB rb_{xa} JJ_a _ _
Match = nn_2, VBD, rb_1, JJ

Total number of match = 4

c) Similarity score calculation
$Similarity\ score = \frac{1}{total\ cost+1} \times total\ number\ of\ match$

Similarity score = $\frac{1}{3+1} \times 4$
= 1

All the similarity scores between grouped POS tagged sentences and aspect-sentiment rules are calculated in the same way. After comparing all the similarity scores, rule (R2) provides the highest similarity matching value. Hence, rule (R2) is selected as the final rule for the sentence. If two rules have a similar score, then the ranking of aspect-sentiment pair rules will be used to select the final rule.

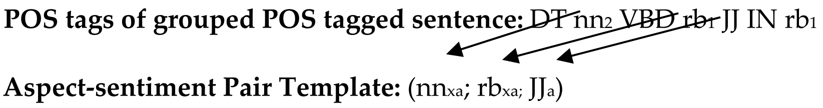
c) Sub Step III: Grouped POS Tagged Sentence and Aspect-sentiment Template Matching

The aspect-sentiment template associated with the aspect-sentiment pair rule selected in Sub step ii) will be used to match with the POS tags of a grouped POS tagged sentence. After that, the POS tags of a grouped POS tagged sentence will be inserted into the template to produce the aspect-sentiment pair.

Then, the aspect-sentiment pair POS tag will be associated with its words. The words in a grouped POS tagged sentence are aligned to the aspect-sentiment pair. The position, ("a"), ("b"), and etc., is used to indicate the position of the POS tag as appearing as first, second, and etc. Example 4.16 shows the matching between a grouped POS tagged sentence and the aspect-sentiment pair template.

Example 4.16:
Grouped POS tagged sentence: *The/DT technical support/nn$_2$ was/VBD not/rb$_1$ helpful/JJ as/IN well/rb$_1$. /.*
POS tags of aspect-sentiment rule (R2) rank 2: nn$_{xa}$ VB JJ$_a$

The POS tags of the grouped POS tagged sentence are matched with the template. The position is indicated by ("a") means, that particular POS tag appearing as the first occurrence. Similar to ("b"), means that the tag appears as second and so on.



POS tags of grouped POS tagged sentence: DT nn$_2$ VBD rb$_1$ JJ IN rb$_1$

Aspect-sentiment Pair Template: (nn$_{xa}$; rb$_{xa;}$ JJ$_a$)

The matched POS tag will be inserted into the aspect-sentiment pair template. The resultant form is the aspect-sentiment pair.

Aspect-sentiment Pair: (nn$_2$; rb$_1$; JJ)

Next, the aspect-sentiment pair is aligned with its associated words as in a grouped POS tagged sentence. Thus, the final product is the aspect-sentiment pair with its associated words.

Aspect-sentiment Pair: (technical support; not; helpful)

5. Results and Discussion

This section elaborates the evaluation of the aspect-sentiment pair extraction with rules and compound noun lexicon (ASPERC) model, the dataset collection, performance measure, results, and result analysis.

5.1. Dataset

The dataset used for the training and testing phase is introduced in the following subsections. In the data collection, the benchmarked data are described in terms of its domain, number of sentences, and proportion of the training and testing dataset. In the benchmarked dataset section, the dataset is described where the reannotation is done by the experts in terms of aspect and sentiment terms.

5.1.1. Data Collection

The dataset used for the experiments are benchmark data from SemEval 2014 Task 4 [25]. The dataset is used for aspect and its polarity identification in SemEval 2014 tasks that involve supervised and unsupervised approaches. We have chosen this dataset because they are the first one tagged at the aspect-level with a multiword aspect and single-word aspect. This dataset consists of two domains, which are laptop and restaurant. Both datasets were divided into a training and testing dataset. The restaurant training data consists of 3041 sentences with annotations of aspect terms for some sentences. The restaurant testing data consists of 800 sentences with annotations of aspect terms for some sentences. The restaurant dataset is a subset of [26] restaurant datasets. Additionally, more sentences are crawled from scratch in the restaurant review. The laptop dataset consists of 3845 training sentences and 800 testing sentences with aspect term annotations for some sentences. This dataset is extracted from the laptop customer review.

In this work, a total of 250 training sentences from 6086 sentences and 126 testing sentences from 1600 sentences are randomly taken from the annotated sentences. A common measure used to split for determining the number of training and testing data is 80: 20 ratios based on the Pareto principle [27]. Hence, for the laptop dataset, the training and testing dataset are 250 and 63. The 250 laptop training dataset is used to train the rules.

To test the rules created using laptop dataset, 63 laptop testing datasets and 63 restaurant testing datasets are used. Table 12 shows the information of the training and testing dataset used.

Table 12. The information of the training and testing datasets.

Domain	Training Dataset	Training Dataset Used	Testing Dataset	Testing Dataset Used
Laptop	3041	250	800	63
Restaurant	3045	-	800	63
Total	6086	250	1600	126

5.1.2. Benchmarked Dataset

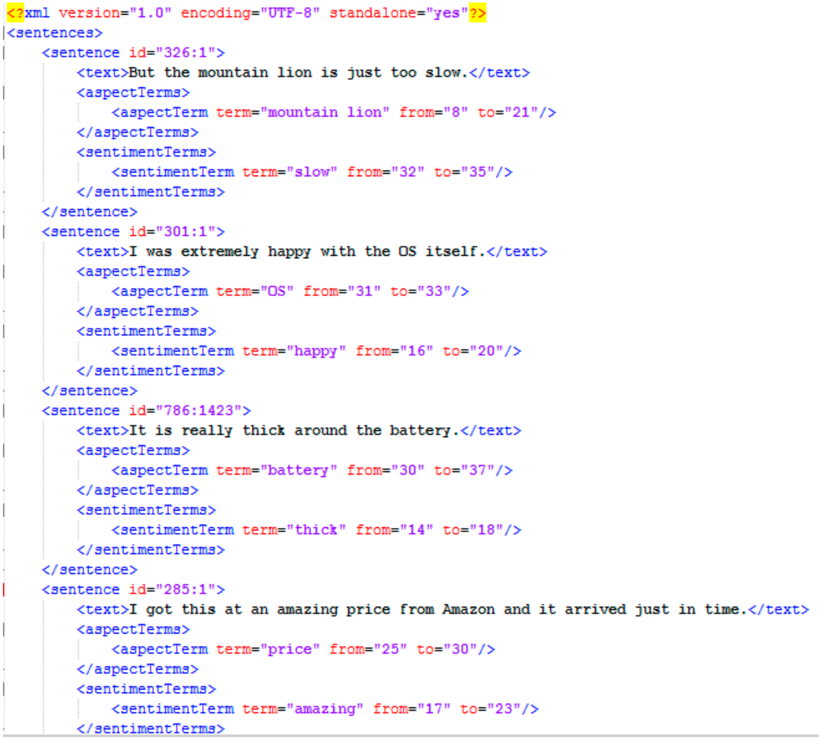
The SemEval 2014 Task 4 dataset consists of datasets that have aspect terms annotated. However, for sentiments, there is no annotation. Thus, two human experts are assigned to annotate the testing dataset to include sentiments and add the missing aspects in annotation. For example, in the sentence, "The specs are pretty good too.", the sentiment, "good", is added as a sentiment term. For example, in the sentence, "There restaurant is very casual", the aspect, "restaurant", is added as an aspect term.

Each human expert annotates 63 sentences for testing datasets. An explanation and examples of aspects and sentiments are given to the human experts to annotate the aspect and sentiment pairs. This dataset is then used as the golden standard for the experiment evaluation. Refer to Figures 4 and 5 for some of the testing review sentences' annotation (golden standard) in the laptop and restaurant dataset. The aspect terms and the sentiment terms are the terms annotated by the user.

5.2. Performance Metric

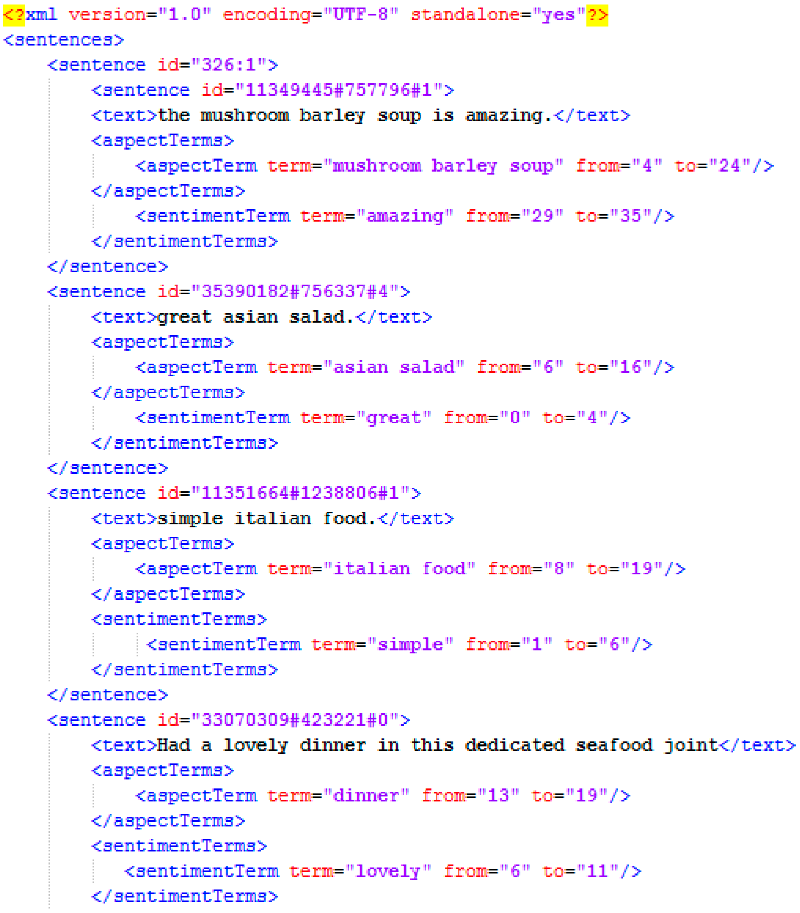
The performance is evaluated by comparing the aspect and sentiment pairs extracted by the approach with the set of gold standard aspect-sentiment pairs. The gold standard is the "true" aspect-sentiment pair that should be extracted by the approach for validation of results. This is to check whether the aspect-sentiment pair extracted by the approach is valid or not valid. The gold standard refers to the annotated testing dataset by the users, where the aspect and sentiment are labelled in the sentences. Hence, to measure the validity of the aspect-sentiment pairs, two measures are used. The two commonly used matrices in opinion mining are utilized, i.e., precision and recall.

Precision is measured to determine the number of correct aspect-sentiment pairs that are extracted by all the approaches. On the other hand, recall is measured to find out the coverage of aspect-sentiment pairs that can be extracted.



```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<sentences>
    <sentence id="326:1">
        <text>But the mountain lion is just too slow.</text>
        <aspectTerms>
            <aspectTerm term="mountain lion" from="8" to="21"/>
        </aspectTerms>
        <sentimentTerms>
            <sentimentTerm term="slow" from="32" to="35"/>
        </sentimentTerms>
    </sentence>
    <sentence id="301:1">
        <text>I was extremely happy with the OS itself.</text>
        <aspectTerms>
            <aspectTerm term="OS" from="31" to="33"/>
        </aspectTerms>
        <sentimentTerms>
            <sentimentTerm term="happy" from="16" to="20"/>
        </sentimentTerms>
    </sentence>
    <sentence id="786:1423">
        <text>It is really thick around the battery.</text>
        <aspectTerms>
            <aspectTerm term="battery" from="30" to="37"/>
        </aspectTerms>
        <sentimentTerms>
            <sentimentTerm term="thick" from="14" to="18"/>
        </sentimentTerms>
    </sentence>
    <sentence id="285:1">
        <text>I got this at an amazing price from Amazon and it arrived just in time.</text>
        <aspectTerms>
            <aspectTerm term="price" from="25" to="30"/>
        </aspectTerms>
        <sentimentTerms>
            <sentimentTerm term="amazing" from="17" to="23"/>
        </sentimentTerms>
```

Figure 4. Annotated testing dataset for the laptop domain.



```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<sentences>
    <sentence id="326:1">
        <sentence id="11349445#757796#1">
        <text>the mushroom barley soup is amazing.</text>
        <aspectTerms>
            <aspectTerm term="mushroom barley soup" from="4" to="24"/>
        </aspectTerms>
            <sentimentTerm term="amazing" from="29" to="35"/>
        </sentimentTerms>
    </sentence>
    <sentence id="35390182#756337#4">
        <text>great asian salad.</text>
        <aspectTerms>
            <aspectTerm term="asian salad" from="6" to="16"/>
        </aspectTerms>
            <sentimentTerm term="great" from="0" to="4"/>
        </sentimentTerms>
    </sentence>
    <sentence id="11351664#1238806#1">
        <text>simple italian food.</text>
        <aspectTerms>
            <aspectTerm term="italian food" from="8" to="19"/>
        </aspectTerms>
        <sentimentTerms>
             <sentimentTerm term="simple" from="1" to="6"/>
        </sentimentTerms>
    </sentence>
    <sentence id="33070309#423221#0">
        <text>Had a lovely dinner in this dedicated seafood joint</text>
        <aspectTerms>
            <aspectTerm term="dinner" from="13" to="19"/>
        </aspectTerms>
        <sentimentTerms>
           <sentimentTerm term="lovely" from="6" to="11"/>
        </sentimentTerms>
```

Figure 5. Annotated testing dataset for the restaurant domain.

5.2.1. Precision

Precision measures the correctness of aspect-sentiment pairs extracted by the approach with the aspect-sentiment pair annotated in the gold standard. The equation of precision is as follows:

$$Precision = \frac{|Relevant\ aspect\ sentiment\ pair \cap Retrieved\ aspect\ sentiment\ pair|}{|Retrieved\ aspect\ sentiment\ pair|}$$

- Relevant aspect sentiment pair indicates the correct aspect-sentiment pair that is extracted by the approach as in the gold standard aspect-sentiment pair; and
- retrieved aspect sentiment pair indicates the aspect-sentiment pair in the gold standard that is supposed to be retrieved by the approach, but it is not retrieved. Also, added with the aspect-sentiment pair that is retrieved by the approach, but not present in the gold standard.

5.2.2. Recall

Recall measures the coverage of the aspect-sentiment pair extracted by the approach with the aspect-sentiment pair in the gold standard. The equation of recall is as follows:

$$Recall = \frac{|Relevant\ aspect\ sentiment\ pair \cap Retrieved\ aspect\ sentiment\ pair|}{|Relevant\ aspect\ sentiment\ pair|}$$

5.3. Experimental Result

In this section, the results of the approach used are discussed based on evaluation on two perspectives, which are the optimized number of review sentences to train the aspect-sentiment pair extraction model as well as the precision and recall of aspect-sentiment pair extraction on testing review sentences.

5.3.1. Evaluation on Optimized Number of Review Sentences to Train the Aspect-sentiment Pair Extraction Model

Here, the number of training review sentences that is optimal to train the model is evaluated. The training sentences are divided into 50 sentences per run. In each run, the rules that can be generated from the sentences are recorded. The rules increase per run. The unique rules are stored in an aggregation manner for each of the runs. This means, after one run, there will be some rules produced, and then in the next run, the new rules are added on to the existing rules.

The number of sentences is taken as reasonable until there is no significant increase in the number of rules. As can be seen in Figure 6, after running 250 sentences, there is only a little increase in the number of rules. Thus, for this work, 250 sentences in datasets are used as training sentences.

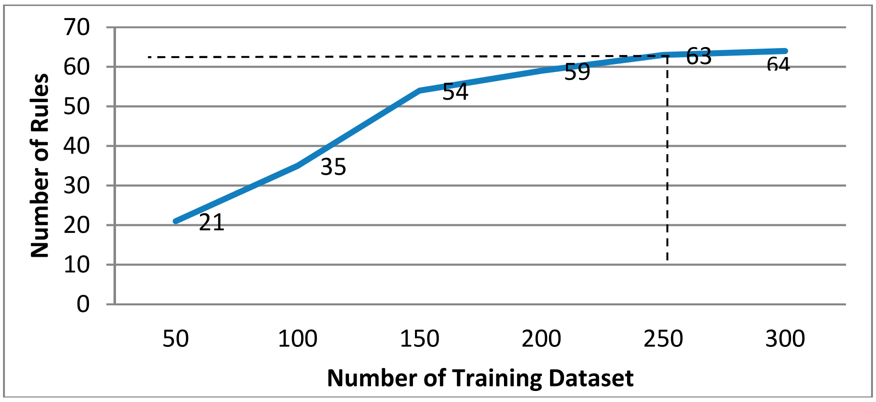


Figure 6. Optimized number of review sentences to train the aspect-sentiment pair extraction model.

5.3.2. Evaluation on Precision and Recall of Aspect-sentiment Pair Extraction on Testing Review Sentences

The motivation of doing this evaluation is to measure the performance of our approach and the baseline approach in aspect-sentiment pair extraction. To test the performance, the annotated testing sentences from SemEval 2014 dataset (golden standard) is used to evaluate the baseline (BE), ASPER and ASPERC to ensure validity. About 126 testing sentences are used, where 63 of sentences are from laptop domain and another 63 sentences are from restaurant domain.

The proposed approach (ASPERC) and its variant (ASPER) are tested. The main concern here is to find out whether ASPER and ASPERC can better extract aspect-sentiment pairs without much aspect-sentiment pair loss. In other words, this is to find out the ability of the approach to extract more aspect-sentiment pairs.

The intention of evaluating ASPER is to determine whether the rules of our work can be used across domains. This is to test whether, even if there is no usage of lexicon with the rules, the rules itself are able to better discover aspect-sentiment pairs. The evaluation is carried out to compare the following approaches in aspect-sentiment pair extraction.

Baseline rule based (BE) approach:

- The baseline [8] has nine rules for extraction of aspect-sentiment pairs. The rules consist of a POS tag sequence to extract aspect-sentiment pairs where a noun is extracted as an aspect and an adjective as a sentiment. The position of the aspect and sentiment is denoted by the arrangement of the noun and adjective in rules;
- this approach considers extraction of aspect-sentiment pairs with occurrence of aspect-sentiment pairs, such as one aspect with one sentiment, one aspect with many sentiments, many aspects with one aspect, one sentiment with one aspect, one sentiment with many aspects, and many sentiments with one aspect; and
- for the multi-word, the aspect is only considered a noun-noun compound noun. Special compound noun aspects and parent-child aspects are not identified.

Aspect-sentiment pair extraction with rules and compound noun lexicon (ASPERC) model:

- This model consists of 63 rules and a template for extraction of aspect-sentiment pairs. The position of aspects and sentiments is denoted with respect to the arrangement of nouns and adjectives in the rules and in the template;
- the model considers extraction of aspect-sentiment pairs with the occurrence of aspect-sentiment pairs, such as one aspect with one sentiment, one aspect with many sentiments, many aspects with one aspect, one sentiment with one aspect, one sentiment with many aspects, and many sentiments with one aspect; and
- it is also able to identify multi-word aspects, such as a noun-noun compound noun, special compound noun aspect, and parent-child aspect.

Aspect-sentiment pair extraction with rules (ASPER) model:

- This model is the variant of ASPERC where the usage of a compound noun is excluded;
- it can identify a multi-word aspect, such as a noun-noun compound noun and parent-child aspect. A special compound noun aspect is not identified; and
- the model is used to test the rules applicability across various domains.

The result on the precision and recall of the aspect-sentiment pair extraction model using rules (ASPER) for the extraction of aspect-sentiment pairs are shown in Figure 7.

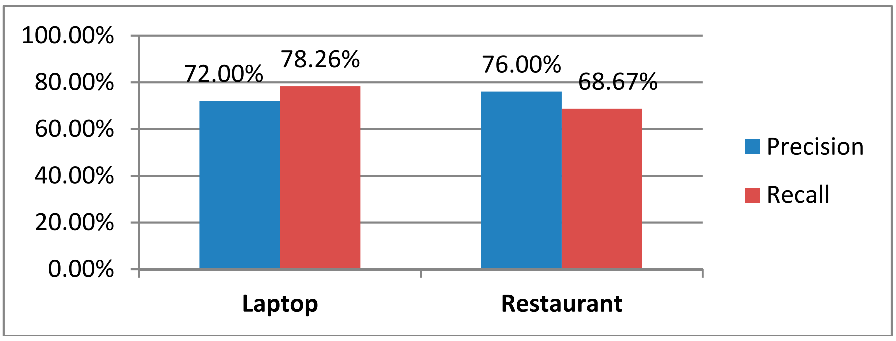


Figure 7. Precision and recall of the ASPER model.

The precision of the ASPER model for aspect-sentiment pair extraction in the laptop domain is 4% lower than in the restaurant domain. However, the recall of the ASPER model for the aspect-sentiment pair extraction model in the laptop domain is 9.59%.

The result on precision and recall of the aspect-sentiment pair extraction model using rules and lexicon (ASPERC) for the extraction of aspect-sentiment pairs is shown in Figure 8.

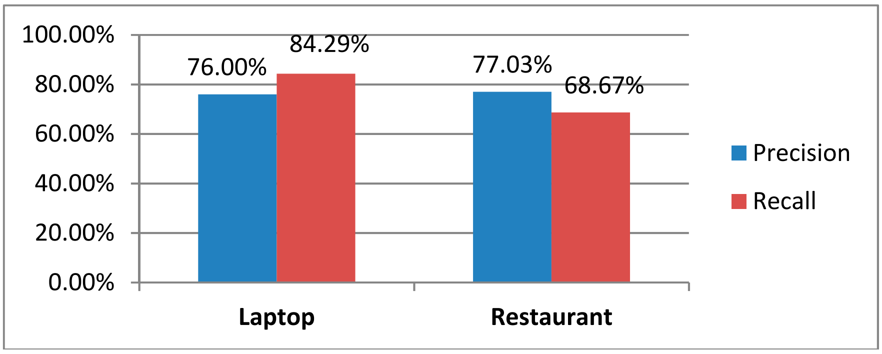


Figure 8. Precision and recall of the ASPERC model.

Similarly, the precision of ASPERC for aspect-sentiment pair extraction in the laptop domain is 1.03% lower than in the restaurant domain. However, the recall of the ASPERC model for the aspect-sentiment pair extraction model in the laptop domain is 15.62% higher.

The results on precision and recall of the ASPER model, ASPERC model, and BE baseline on the laptop dataset are shown in Figure 9.

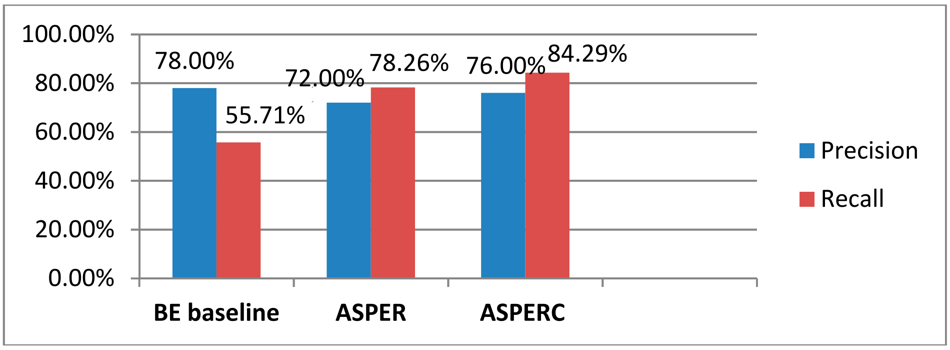


Figure 9. Precision and recall of ASPER model, ASPERC model, and BE baseline in the laptop dataset.

ASPER precision and recall is lower by 4% and 6.03% than ASPERC. The precision of ASPER and ASPERC is slightly lower than the BE baseline. However, both models have higher recall compared to the BE baseline. Even tough precision of ASPER is 6% lower than the BE baseline, yet ASPER still managed to increase by 22.55% higher in terms of recall. In a similar manner, there is a little drop of precision by 2% in ASPERC, but still, it advances the BE baseline in recall by 28.58%.

The results on precision and recall of the ASPER model, ASPERC model, and BE baseline on the restaurant dataset are shown in Figure 10.

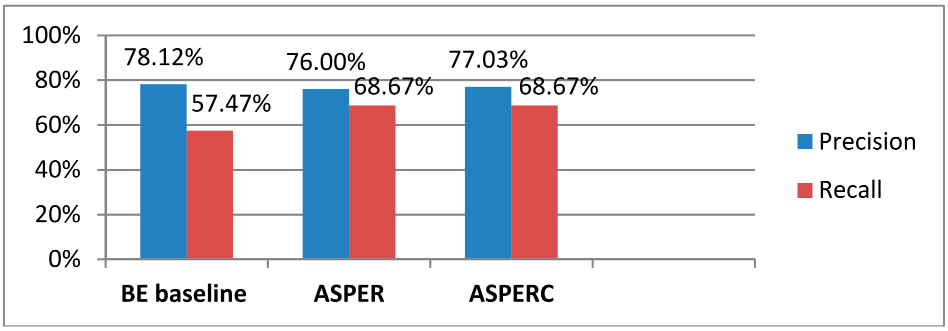


Figure 10. Precision and recall of the ASPER model, ASPERC model, and BE baseline in the restaurant dataset.

The precision of ASPER is less by 1.03%, but the recall is the same with ASPERC. The precision of ASPER and ASPERC is slightly lower than the BE baseline. However, both models have higher recall compared to the BE baseline.

Even tough precision of ASPER is 2.12% lower than the BE baseline, yet ASPER still managed to increase by 11.2% in terms of recall. In a similar manner, there is a little drop of precision by 1.09% in ASPERC, but still, it advances the BE baseline in recall by 11.2%.

5.4. *Result Analysis*

In this section, the results obtained are analyzed. The two main evaluations conducted are discussed in detail.

5.4.1. Optimized Number of Review Sentences to Train the Aspect-sentiment Pair Extraction Model

Determining the number of training sentences helps to set the number of testing sentences. Based on the stopping criterion, the amount of training sentences is set as 250 sentences. Following the proportion of 80:20, the amount of testing sentences is set to 63 corresponding to 250 training sentences. By using 63 testing sentences, the precision and recall of aspect-sentiment pairs extracted in the laptop domain and restaurant domain are obtained.

It is observed that 63 testing sentences can provide a precision of 72.00 (ASPER)% and 76.00 (ASPERC) for the domain. Similarly, a precision of 76.00% (ASPER) and 77.03% (ASPERC) is obtained for the restaurant domain. Meanwhile, a recall of 78.26% (ASPER) and 84.29% (ASPERC) in the laptop dataset and 68.67% (ASPER and ASPERC) is achieved in the restaurant dataset. With the usage of 250 training sentences, a decent performance in terms of precision and recall can be attained in both the laptop and restaurant dataset. Hence, this result obtained from this evaluation can be used as a guideline in terms of the range of the number of training sentences that can be used to train a similar extraction model, especially in the domain of review texts.

5.4.2. Precision of Aspect-sentiment Pair Extraction on Testing Review Sentences

Precision is a measure to find out the correct aspect-sentiment pairs extracted by the approaches. In both the laptop and restaurant domain, the precision of the ASPERC is higher than ASPER because the use of compound noun lexicon has enabled ASPERC to find correct aspect-sentiment pairs.

Due to the complexity and heterogeneity of review sentences written by customers, some irrelevant aspect-sentiment pairs also got extracted in ASPER and ASPERC approaches, which results in a slight decrement as compared to the BE baseline approach. Besides that, ASPER and

ASPERC approaches are based on the existence of some specific grammatical language rules, which the sentence structure needs to follow the exact structure of the rules to be selected as an aspect-sentiment pair.

The BE baseline can maintain a slight increment in precision, due to the generality of rules in aspect-sentiment pair extraction. The author combined all the specific rules to form a general rule that can be used in different sentence structures.

5.4.3. Recall of Aspect-sentiment Pair Extraction on Testing Review Sentences

Recall is a measure to find out the coverage of aspect-sentiment pairs extracted by the approaches. In the laptop domain, the recall of the ASPERC is higher than ASPER because the use of compound noun lexicon has enabled ASPERC to find more aspect-sentiment pairs. However, in the restaurant domain, the recall of both ASPER and ASPERC are the same, as in this domain, the compound noun that is needed to be resolved involves a food name. Few food names are not able to be found in the lexicon due to the complexity of the name itself.

We have enhanced the rules and have decent coverage. Hence, ASPER and ASPERC can achieve an improvement in recall compared to the BE baseline. Also, the aspect-sentiment pair template of the rules that associate an aspect to its corresponding sentiment contributes to recall. Another reason of the increase in recall is the use of compound noun lexicon. By utilizing compound noun lexicon, more aspects are discovered, thus resulting in relevant aspect-sentiment pair extraction.

Since the baseline rules do not give enough coverage to various types of aspect extraction, such as a parent-child aspect and special compound noun aspect, it can be concluded that the baseline cannot extract enough aspects. When some of the aspects are not identified, eventually, the sentiment is also not identified. This results in extraction of incomplete aspect-sentiment pairs.

5.5. Comparison of Aspect-sentiment Pair Extraction ASPER Model, ASPERC Model, and BE Baseline

Table 13 shows the comparison of aspect-sentiment pair extraction by all the approaches in both laptop and restaurant training sentences. It is notable that the BE baseline is not able to identify the parent-child aspect, e.g., "Customization on Mac", with the rules. In addition to that, the special compound noun aspect, like "Set up", is also not discovered in the baseline. In contrast, ASPER can identify the parent-child aspect with the aspect-sentiment pair rules. With the advantage of compound noun lexicon, ASPERC can identify both the parent-child aspect and special compound noun aspect.

Table 13. Comparison of aspect-sentiment pair extraction ASPER model, ASPERC model, and BE baseline.

No.	Example of Testing Sentence	Approach		
		BE Baseline	**ASPER**	**ASPERC**
1	Set up was easy	No	No	Yes
2	MS Office 2011 for Mac is wonderful	No	Yes	Yes
3	But the Mountain lion is just too slow	No	Yes	Yes
4	I was extremely happy with the OS itself	No	Yes	Yes
5	The specs are pretty good too	Yes	Yes	Yes
6	The technical support was not helpful as well.	No	No	Yes
7	Great product	Yes	Yes	Yes
8	Customization on Mac is impossible	No	Yes	Yes
9	Excellent durability of battery	No	Yes	Yes
10	Great Asian salad	Yes	Yes	Yes
11	Great and attentive staff	Yes	Yes	Yes
12	Their twist on a pizza is healthy	No	Yes	Yes
13	Definitely try the taglierini with truffles it was incredible	No	Yes	Yes

6. Conclusions

In this paper, the motivation was to provide a decomposed view of opinions in review sentences that consist of aspect-sentiment pairs. In the state-of-the-art, there are many researchers who have applied aspect-based opinion mining to extract aspect-sentiment pairs. However, none of the approaches completely resolve the issues of multi-word aspects' identification, such as special compound noun aspects and parent-child aspects.

Hence, to improve aspect-sentiment pair extraction, aspect identification was enhanced by proposing the aspect-sentiment pair extraction using rules and compound noun lexicon (ASPERC) model. Basically, a total of three phases were included in the model, such as compound noun lexicon generation, aspect-sentiment pair rules generation, and aspect-sentiment pair extraction. The outcome of Phase I was compound noun lexicon for the laptop and restaurant domain. In Phase II, the outcome was the knowledge base consisting of generic rules, aspect-sentiment pair rules, and an aspect-sentiment pair template. In Phase III, the outcome of the model was the aspect-sentiment pair extractions.

With the usage of rules, the parent-child aspect was identified. Different types of order of arrangements in the extraction of aspect-sentiment pairs were also tackled with the rules. The rules have the template to match the aspects to sentiments in which the aspects or sentiments may appear first and later in the sentence. Additionally, the compound noun lexicon helps to identify more aspects.

In the evaluation, the ASPERC model and its variant, ASPER, can decently perform aspect-sentiment pair extraction with reasonable precision and recall. Better precision and recall were obtained after using compound noun lexicon. The models achieved precision of 72.00% (ASPER) and 76.00% (ASPERC) in the laptop domain and precision of 76.00% (ASPER) and 77.03% (ASPERC) for the restaurant domain. The recall was 78.26% (ASPER) and 84.29% (ASPERC) in the laptop domain and 68.67% (ASPER and ASPERC) in the restaurant domain.

With respect to the objective and evaluation outcome, we highlight two main contributions from this work. First, the optimal number of sentences to train the ASPERC model were identified. For training the model, usually, a higher number of training sentences would result in higher performance. However, to use fewer sentences, as acquiring sentence datasets can be expensive and time consuming, the rules were used as the stopping criteria. The sentences were categorized according to aspect-sentiment pair rules in every iteration of the experiment. Iteration was repeated until there was only a little increase in the number of rules. The number of sentences that corresponds to the number of rules that were not increasing much were taken as the optimal number of sentences. The number helps to determine a reasonable and minimal size of training sentences that will be giving a reasonable precision and recall.

Second, the ASPERC was proposed to improve the aspect and sentiment extraction through rules and compound noun lexicon. This model utilizes rules for discovering both single-word and multi-word aspects. The multi-word aspect, such as the parent-child aspect, was extracted through the rules. The rules were also able to extract the aspect or sentiment, which appears first or later in sentences, using the template matching between aspect and sentiment. Moreover, compound noun lexicon enabled identification of non-noun-noun compound nouns as the aspect. By enhancing aspect identification in aspect-sentiment pair extraction, more aspect words could be obtained. When the aspect can be obtained, its corresponding sentiment can be discovered. Hence, this results in greater extraction of aspect-sentiment pairs. Both the ASPERC and its variant (ASPER) models are able to extract aspect sentiment pairs in a decent performance in precision and recall. The ASPER model contains rules without compound noun lexicon. Even though these rules were trained in the laptop domain and tested in both the laptop and restaurant domain, we can still achieve a reasonable performance. This indicates that the rules are domain independent and can be applied across different domains.

In conclusion, aspect and sentiment pair extraction is beneficial for both customers and business organization. By extracting aspect-sentiment pairs, the opinion in the sentences can be simplified.

Individuals can save time by not reading an overwhelming amount of reviews when looking for an opinion towards an item or matter.

As future work, the model can be enhanced by integrating ontology to discover the hierarchy of the aspects in the aspect-sentiment model. Since our approach is already able to extract parent-child aspects and special compound noun aspects, representing them in a hierarchy with associated sentiments can provide a structured detail of products or services opinions. In terms of the semantics of the aspect-sentiment model, this research can further integrate polarity scores for the extracted aspect-sentiment pair. On top of that, the pair can also be linked to a finer grain of sentiment analysis, such as emotions, like anger, happy, sad, etc.

Currently, besides the aspect-sentiment pairs extraction, there is no grouping of an aspect to its category. Grouping different mentions of aspects makes the aspects extracted be less redundant in the extraction's output.

Author Contributions: Conceptualization, K.H.G.; Data curation, N.R.A.K.; Formal analysis, K.H.G.; Funding acquisition, K.H.G.; Investigation, N.R.A.K.; Methodology, N.R.A.K.; Supervision, K.H.G.; Validation, N.R.A.K. and E.H.; Writing—review & editing, E.H.

Funding: This research was supported by USM Research University Grant (1001/PKOMP/811335: Mining Unstructured Web Data for Tour Itineraries Construction), Universiti Sains Malaysia.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Liu, B. Sentiment analysis and opinion mining. In *Synthesis Lectures on Human Language Technologies*; Morgan & Claypool Publishers: San Rafael, CA, USA, 2012; Volume 5, pp. 1–167.
2. Perikos, I.; Hatzilygeroudis, I. Aspect based sentiment analysis in social media with classifier ensembles. In Proceedings of the IEEE/ACIS 16th International Conference on Computer and Information Science (ICIS), Wuhan, China, 24–26 May 2017; pp. 273–278.
3. Hu, M.; Liu, B. Mining and summarizing customer reviews. In Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Seattle, WA, USA, 22–24 August 2004; pp. 168–177.
4. Popescu, A.M.; Etzioni, O. Extracting product features and opinions from reviews. In *Natural Language Processing and Text Mining*; Springer: London, UK, 2007; pp. 9–28.
5. Moghaddam, S.; Ester, M. On the Design of LDA Models for Aspect-based Opinion Mining. In Proceedings of the 12th ACM International Conference on Information & Knowledge Management, Maui, HI, USA, 29 October–2 November 2012; pp. 803–812.
6. Ravi, K.; Raghuveer, K. Web User Opinion Analysis for Product Features Extraction and Opinion Summarization. *Int. J. Web Semant. Technol.* **2012**, *3*, 69–82. [CrossRef]
7. Chinsha, T.C.; Shibily, J. A Syntactic Approach for Aspect Based Opinion Mining. In Proceedings of the 9th International Workshop on Semantic Evaluation, Denver, CO, USA, 4–5 June 2015; pp. 730–735.
8. Bancken, W.; Alfarone, D.; Davis, J. Automatically Detecting and Rating Product Aspects from Textual Customer Reviews. In Proceedings of the DMNLP, Workshop at ECML/PKDD, Nancy, France, 15 September 2014.
9. Bross, J.; Ehrig, H. Automatic construction of domain and aspect specific sentiment lexicons for customer review mining. In Proceedings of the 22nd ACM International Conference on Information & Knowledge Management, San Francisco, CA, USA, 27 October–1 November 2013; pp. 1077–1086.
10. Veselovská, K.; Tamchyna, A. UFAL: Using Hand-crafted Rules in Aspect Based Sentiment Analysis on Parsed Data. In Proceedings of the 8th International Workshop on Semantic Evaluation, Dublin, Ireland, 23–24 August 2014; pp. 694–698.
11. Poria, S.; Cambria, E.; Gelbukh, A.; Gui, C. A Rule-Based Approach to Aspect Extraction from Product Reviews. In Proceedings of the Second Workshop on Natural Language Processing for Social Media, Dublin, Ireland, 24 August 2014; pp. 28–37.

12. Zhao, Q.; Wang, H.; Lv, P.; Zhang, C. A Bootstrapping Based Refinement Framework for Mining Opinion Words and Targets. In Proceedings of the 23rd ACM International Conference On Conference On Information and Knowledge Management, Shanghai, China, 3–7 November 2014; pp. 1995–1998.
13. Carpenter, T.; Golab, L.; Syed, S. Is the Grass Greener? Mining Electric Vehicle Opinions. In Proceedings of the 5th International Conference on Future Energy Systems, Cambridge, UK, 11–13 June 2014; pp. 241–252.
14. Li, Y.; Wang, H.; Qin, Z.; Xu, W.; Guo, J. Confidence Estimation and Reputation Analysis in Aspect Extraction. In Proceedings of the 22nd International Conference on Pattern Recognition, Stockholm, Sweden, 24–28 August 2014; pp. 3612–3617.
15. Kim, H.; Ganesan, K.; Parikshit, S.; ChengXiang, Z. Comprehensive review of opinion summarization. 2011; in press.
16. Schouten, K.; Frasincar, F. Survey on Aspect-Level Sentiment Analysis. *IEEE T Knowl. Data En.* **2016**, *28*, 813–830. [CrossRef]
17. Hurford, J.R. *Grammar: A. Student's Guide*; Cambridge University Press: Cambridge, UK, 1994.
18. Kim, S.; Zhang, J.; Chen, Z.; Oh, A.; Liu, S. A Hierarchical Aspect-Sentiment Model for Online Reviews. In Proceedings of the 27th AAAI Conference on Artifical Intelligence, Bellevue, WA, USA, 14–18 July 2013; pp. 526–533.
19. Xu, P.; Kang, J.; Ringgaard, M.; Och, F. Using a Dependency Parser to Improve SMT for Subject-Object-Verb Languages. In Proceedings of the Annual Conference of the North American Chapter of ACL, Boulder, CO, USA, 31 May–5 June 2009; pp. 245–253.
20. Wang, W.; Pan, S.J.; Dahlmeier, D.; Xiao, X. Coupled Multi-Layer Attentions for Co-Extraction of Aspect and Opinion Terms. In Proceedings of the AAAI, San Francisco, CA, USA, 4–9 February 2017; pp. 3316–3322.
21. Luhn, H.P. Key word-in-context index for technical literature (KWIC index). *Am. Doc.* **1960**, *11*, 288–295. [CrossRef]
22. Anthony, L. AntConc: Design and Development of a Freeware Corpus Analysis Toolkit for the Technical Writing Classroom. In Proceedings of the 2005 International Professional Communication Conference, Limerick, Ireland, 10–13 July 2005; pp. 729–737.
23. Barri'ere, C.; Ménard, P. Multiword noun compound bracketing using Wikipedia. In Proceedings of the First Workshop on Computational Approaches to Compound Analysis, Dublin, Ireland, 24 August 2014; pp. 72–80.
24. Jurafsky, D.; Martin, J.H. *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, And Speech Recognition*, 2nd ed.; Prentice Hall PTR: Upper Saddle River, NJ, USA, 2014.
25. Pontiki, M.; Galanis, D.; Pavlopoulos, J.; Papageorgiou, H.; Androutsopoulos, I.; Manandhar, S. Semeval-2014 Task 4: Aspect Based Sentiment Analysis. In Proceedings of the 8th International Workshop on Semantic Evaluation, Dublin, Ireland, 23–24 August 2014; pp. 27–35.
26. Ganu, G.; Elhadad, N.; Marian, A. Beyond the Stars: Improving Rating Predictions using Review Text Content. In Proceedings of the 12th International Workshop on the Web and Databases, Providence, RI, USA, 28 June 2009; pp. 1–6.
27. Kiremire, A.R. *The Application of The Pareto Principle in Software Engineering*; Louisiana Tech University: Ruston, LA, USA, 2011.