

Development of Domain-Specific Lexicon for Aspect-Based Sentiment Analysis

Prisila Michelle*, Fariska Zakhralativa Ruskanda[†], Ayu Purwarianti[‡]

School of Electrical Engineering and Informatics

[‡]U-CoE AI-VLB

Institut Teknologi Bandung

Bandung, Indonesia

*13516129@std.stei.itb.ac.id, [†]fariska@informatika.org, [‡]ayu@informatika.org

Abstract—Aspect term extraction is one of the main subtasks in aspect-based sentiment analysis. An aspect extraction method based on the Sequential Covering algorithm [2] successfully used a list of aspects and opinion words to improve the performance of aspect extraction. The aspect and opinion list used in the method is crafted manually, costing a significant amount of time and effort. To ease the effort, we proposed a way to automatically build the aspect and opinion list. We also modified the scope of the word list to a larger domain. The resulting word list is therefore called domain-specific lexicon. In this paper, we used word embedding technique to develop domain-specific lexicons of size 1000. The data used in the development of domain-specific lexicon were collected from review websites using a focused web crawler. The resulting domain-specific lexicons were used in the modified Sequential Covering method. We used a total of 3,124 review sentences from the digital camera, handphone, and restaurant domain to evaluate the performance of the modified Sequential Covering method. Experimental results showed that this method managed to yield better F1 scores than the baseline Sequential Covering method. The best F1 scores for each dataset were as follows: 0.645 for Nikon Coolpix 4300, 0.581 for Canon G3, 0.629 for Nokia 6610, and 0.705 for ABSA16_Restaurants_Train_SB1.

Index Terms—domain-specific lexicon, aspect extraction, word embedding, Sequential Covering

I. INTRODUCTION

In recent years, product reviews are everywhere on the internet. For example, one of the many places where you can find product reviews is e-commerce websites. In those kinds of websites, the number of reviews for each product can range from several to hundreds or even thousands of reviews, making it hard for us to review manually. This is where sentiment analysis comes into play.

Sentiment analysis, also called opinion mining, is the field of study that analyzes peoples opinions, sentiments, evaluations, appraisals, attitudes, and emotions towards entities such as products, services, organizations, individuals, issues, events, topics, and their attributes [1]. There are three levels of sentiment analysis: document level, sentence level, and aspect level. Sentiment analysis conducted on the aspect level is called Aspect-Based Sentiment Analysis (ABSA).

The two main subtasks of ABSA are aspect term extraction and polarity identification. In the aspect extraction task, we aim to extract the aspect/opinion target/feature/topic from the review text. For example, in the sentence “The screen is so vivid,” “screen” would be extracted as the aspect.

One of the examples of aspect extraction method is the Sequential Covering algorithm by Ruskanda et al. [2], which is a variation of language rule-based method. In the said method, a list of aspects and opinion words from the annotated review datasets are used to aid the process of extracting aspect and opinion expressions. By calculating the similarities between each extracted word with words in the list and using them to select words that are above a certain bound, the precision of the extraction process was able to be significantly improved.

However, there is a downside to the Sequential Covering method. If the reviews are not yet annotated, we would have to manually annotate the aspect and opinion of each review, costing a significant amount of time and effort. To tackle this downside, we propose a way to automatically build aspect and opinion list by using word embedding. We also expand the scope of the aspect and opinion list to a larger domain.

In our work, we called the aspect and opinion list as domain-specific lexicon because each of the resulting word lists only contains the aspect and opinions that are relevant to a domain. The domain-specific lexicons that we created are limited to restaurant, handphone, and digital camera domains. These domains are chosen based on the review datasets that we used to evaluate the performance of aspect extraction.

The rest of the paper is organized as follows: Section two discusses the works related to our research, sections three and four discuss the detail of the development and usage of domain-specific lexicons, while sections five and six discuss the experiments and results of the experiments. The final section discusses the conclusion.

II. RELATED WORK

The term “domain-specific lexicon” has been used in natural language processing and information access research to describe a lexicon consisting of terms pertaining to a given domain or discipline [3]. Because the manual generation of domain-specific lexicon is expensive, researchers are developing ways to build domain-specific lexicons automatically.

There are several fields of research that are related with the development of domain-specific lexicon: research in automatic domain-specific term extraction [3], [4], [5], [6], domain-specific sentiment lexicon induction [7], [8], and specialized dictionary development [9]. The first field of research focuses on the automatic domain-specific term extraction, which is a categorization or classification task in which a term

would be categorized into a previously determined domain [4]. This task could be done in unsupervised [3], [4], [5] or supervised way [6]. The resulting lexicon in this kind of research is usually only consisted of noun terms.

The second field of research related to our work is research about domain-specific sentiment lexicon induction, in which the lexicons are created to be used in ABSA. This is aligned with our intention of use, but there is an important difference: domain-specific sentiment lexicon is used in the polarity identification step while our domain-specific lexicon is used in the aspect extraction step. Because it is a type of sentiment lexicon, the lexicon is usually only consisted of opinions.

The final field of research related to our work is about the development of specialized dictionary. As described in the work of Grefenstette and Muchemi [9], specialized dictionary is a lexicon containing domain-specific words that can be used to understand the concepts of a domain. To build specialized dictionary, Grefenstette and Muchemi used word embedding that is trained on domain-specific corpora. The end results of specialized dictionary development are quite alike with what we have in mind, therefore we used the adaptation of this approach in our implementation.

Domain-specific lexicons are used in ABSA for aspect extraction task [10], [11]. Zhuang et al. [10] used domain-specific lexicons to extract opinion pairs in a sentence, before checking the validity of each opinion pair by matching the dependency path between them with a predefined dependency relation template. Chauhan and Meena [11] used domain-specific lexicons to filter out less prominent aspect terms. The methods of filtering that are used in the study are frequency and similarity-based filtering.

Moreover, domain-specific lexicons are also used for aspect categorization task [12], [13]. An example of this is the work of Anand and Naorem [12], in which domain-specific lexicons are used to map aspects into their corresponding categories. Another example is the work of Mukherjee and Liu [13], in which domain-specific lexicons are used as seeds for a statistical model to aid the process of grouping semantically related terms in the same aspect category.

III. OVERVIEW OF THE DEVELOPMENT AND USAGE OF DOMAIN-SPECIFIC LEXICON

The system that we used to build domain-specific lexicons is adapted from the works of Grefenstette and Muchemi [9]. The general architecture of the adapted system is shown in Fig. 1. At first, we used ACHE web crawler¹ to collect domain-specific corpus. After that, we preprocessed the corpus and used it to build domain-specific word embedding using sense2vec [14]. At the end of the development steps, we used the resulting word embedding to build a classifier for domain-specific words extraction.

Fig. 2 shows the modified aspect - opinion extraction process. The difference from the original process lies in the aspect-opinion expression extraction. In the modified process, we replaced the aspect and opinion list with domain-specific lexicon. We also replaced the word embedding used for similarity checking between lexicon and the candidate word with our pretrained sense2vec.

¹ ACHE: <https://github.com/VIDA-NYU/ache>

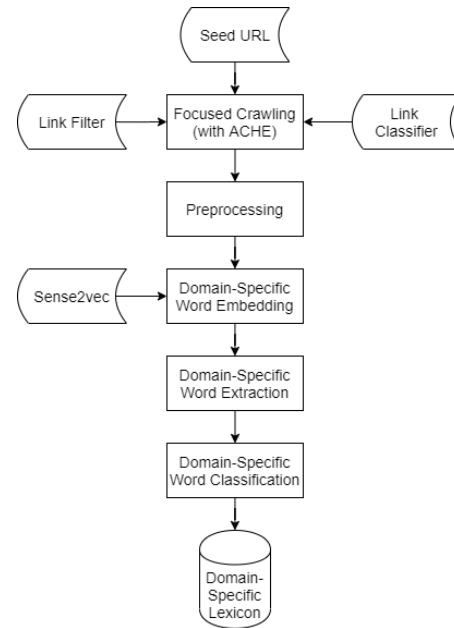


Fig. 1. General architecture of the adapted system.

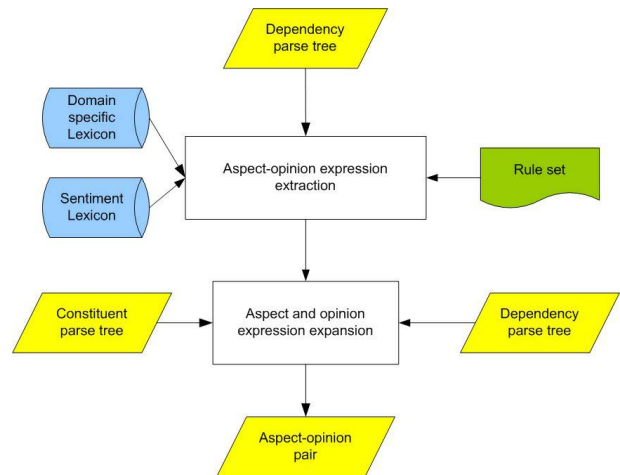


Fig. 2. Modified aspect - opinion extraction process.

To determine the threshold for similarity value used in the extraction process, we experimented with a range of values from 0.1 to 0.9. Then, we conducted the extraction process using these values. Afterwards, we compared the performance of each extraction process and selected the threshold value of 0.3 which gave us the best F1 score.

IV. DEVELOPMENT OF DOMAIN-SPECIFIC LEXICON

The explanations for each component of the system on Fig. 1 are written on the following sections.

A. Focused Crawling

The tool used in this step is ACHE, which is an open source focused web crawler. There are three important elements needed to start the crawler:

- 1) *Seed URL*: List of URLs that will be visited when the crawler starts. We provided about 10 seeds for each domain.
- 2) *Link Classifier*: List of words in regex compatible form, used to determine if a web page is relevant to the domain.

3) *Link Filter*: List of URLs that can be visited by the crawler (whitelist) and cannot be visited by the crawler (blacklist).

The crawling process took about an hour for each domain, resulting in about 3000 crawled web pages per domain. The crawled data is stored in a DEFLATE file consists of JSON objects. Each JSON object contains the details of a web page.

B. Preprocessing

We performed four steps of preprocessing, as follows:

1) *Preprocessing on Crawled Data*: First, we checked the relevance of each JSON object in the crawled data. If the object is relevant to the domain, extract its content.

2) *Cleaning Crawled Data*: To clean crawled data, we deleted every character that is not in ASCII, name of currencies, and whitespace. After that, we split the data to form a file in which each line contains a sentence.

3) *Preprocessing on a Sentence*: We preprocessed each sentence using a pretrained model from spaCy² to create a Doc. Doc is a data structure from spaCy which contains a sequence of tokens. The process of building a Doc is illustrated on Fig. 3. After the Doc was built, we continued with token merging.

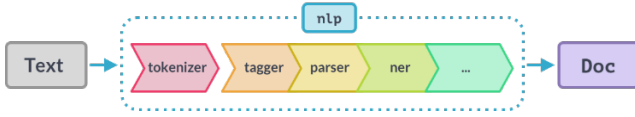


Fig. 3. The process of building a Doc [18].

a) *Merge Noun Phrase*: In this step, we checked every phrase in the generated noun chunks property from Doc. Noun chunks are phrases which have a noun as the head. If the phrase is prefixed with determiner or conjunction, then the prefix would be removed. If the phrase starts with an adjective or adverb, then it would not be merged, unless it is in the list of exceptions described in Table I.

TABLE I
LIST OF EXCEPTIONS WITH EXAMPLES

| Exceptions | Examples |
|------------------------------------|-------------------------------|
| Phrase is a term in WordNet | digital camera, mineral water |
| The adjective is a nationality | French, English, Welsh |
| The adjective is a color | red, blue, yellow |
| The adjective is an ordinal number | first, second, third |

b) *Merge Named Entities*: We merged named entities of type name, location, time, number, and money.

c) *Merge Adjective Phrase*: To extract adjective phrase, we used the textacy library which is an extension of spaCy. The extraction was done using the pattern described in Fig. 4., which means: Find an adjective token, can be followed by an adverb token or not, which is followed by another adjective token and can be followed by another adverb token or not.

4) *Cleaning a Sentence*: After the preprocessing steps were done, we proceeded to clean each sentence by removing stopwords, URLs, and named entities.

The example of input and final output from the preprocessing steps is displayed on Fig. 5. In the output, each token is followed by its part-of-speech (POS) tag obtained from spaCy. While doing the POS tagging, spaCy followed the Universal POS tags format from the Universal Dependencies scheme.

```

patterns = [{"POS": "ADV"}, {"POS": "ADV", "OP": "?"}, {"POS": "ADJ"}, {"POS": "ADV", "OP": "?"}]

```

Fig. 4. Patterns for extracting adjective phrase.

| | |
|---------------|---|
| Input | Roughly half the price of Huawei's flagship P-series phones, the P30 Lite looks just as good but lacks key features. |
| Output | price NOUN huawei's_flagship_p_series_phones NOUN p30_lite PROPN looks VERB just_as_good ADJ lacks VERB key_features NOUN |

Fig. 5. An example of input and final output.

C. Domain-Specific Word Embedding

To build the embedding, we used an implementation of sense2vec provided by Explosion AI³. The sense2vec model was trained using word2vec via FastText. To avoid using n-gram, we set the value of the maximum n-gram size parameter to 0. There are two types of word2vec model that we used to build the embedding: CBOW and Skip-gram. For each model, we created word vectors of size 300 and 400.

D. Domain-Specific Word Extraction

At first, we manually labelled every unique word in the preprocessed text into three classes: aspect, opinion, and other (neither an aspect nor an opinion). Then, we used the following approach to choose the words for test data, while the rest of the words were used as the train data:

1) *Using Term Frequency Only*: In this approach, we used 1000 most frequent words as the test data. Examples of the resulting domain-specific lexicons are shown in Table II.

2) *Using Term Frequency and Cosine Similarity*: First, we chose 50 most frequent words as the seed words. Next, we expand the seed words using 20 most similar words from each seed. The number of most frequent and most similar words were chosen arbitrarily. Table III shows the resulting domain-specific lexicon.

E. Domain-Specific Word Classification

After separating train and test data, we classified the words in the test data into their respective classes using supervised and unsupervised approach. The explanations for each approach are as follows:

1) Supervised Approach:

a) *SVM Classifier with RBF Kernel*: In our research, we opted to use shallow learning over deep learning because there is no sequence in our train and test data. Our data consisted of single words only. SVM was chosen over other shallow learning algorithms because of its proven ability to produce good results on text classification tasks [15]. As for the feature, we used normalized word vectors.

²spaCy: <https://github.com/explosion/spaCy>

³sense2vec: <https://github.com/explosion/sense2vec>

TABLE II
EXAMPLES OF DOMAIN-SPECIFIC LEXICON CREATED USING TERM FREQUENCY

| Restaurant | | Handphone | | Digital Camera | |
|-----------------|---------------|--------------|----------------|----------------|---------------|
| Aspect | Opinion | Aspect | Opinion | Aspect | Opinion |
| dishes NOUN | lovely ADJ | display NOUN | huge ADJ | camera NOUN | large ADJ |
| restaurant NOUN | best ADJ | price NOUN | very_good ADJ | sensor NOUN | fine ADJ |
| menu NOUN | enjoyed VERB | feature NOUN | quick ADJ | quality NOUN | detailed ADJ |
| flavour NOUN | delicious ADJ | phone NOUN | clear ADJ | model NOUN | quick ADJ |
| cooking NOUN | good ADJ | battery NOUN | impressive ADJ | lens NOUN | excellent ADJ |

TABLE III
EXAMPLES OF DOMAIN-SPECIFIC LEXICON CREATED USING TERM FREQUENCY AND COSINE SIMILARITY

| Restaurant | | Handphone | | Digital Camera | |
|-------------------------|---------------|--------------------------|---------------|--------------------|---------------|
| Aspect | Opinion | Aspect | Opinion | Aspect | Opinion |
| restaurants NOUN | little ADJ | phone NOUN | new ADJ | camera NOUN | different ADJ |
| tasting_menu NOUN | very_best ADJ | device NOUN | good ADJ | image_quality NOUN | excellent ADJ |
| prices NOUN | excellent ADJ | touchscreen_display NOUN | massive ADJ | fidelity PROPN | best ADJ |
| dishes NOUN | efficient ADJ | performance NOUN | fantastic ADJ | algorithm NOUN | decent ADJ |
| ingredient_quality NOUN | greatest ADJ | quality NOUN | decent ADJ | aps_c NOUN | big ADJ |

b) *Cosine Similarity*: In this method, we compared the similarity of each word to the averaged word vector of aspect class, averaged word vector of opinion class, and averaged word vector of other class. After that, we chose the nearest averaged word vector as the class.

2) *Unsupervised Approach*: In this approach, we clustered the word vectors in train data using K-Means and K-Medoids method. Then, we used the computed centroids to predict the test data.

V. EXPERIMENTS

A. Experiments on Word Classification

The experiments on word classification were conducted on three domains: digital camera, handphone, and restaurant. In the word classification task, we aimed to classify each word to aspect, opinion, and other class. The statistics of each domains are listed on Table IV.

TABLE IV
WORD CLASSIFICATION DATASET STATISTICS

| Domain | Train Data | | | Test Data | | |
|----------------|------------|------|------|-----------|-----|-----|
| | A | O | X | A | O | X |
| Restaurant | 9335 | 2885 | 2742 | 428 | 252 | 320 |
| Handphone | 8701 | 3009 | 3231 | 431 | 219 | 350 |
| Digital camera | 8057 | 2540 | 3122 | 372 | 203 | 425 |

^aA: Aspect; O: Opinion; X: Other.

To evaluate the experiments, we used multiple metrics such as F1 score, accuracy, and Adjusted Rand Index (ARI). The F1 score and accuracy were used as the metrics for the supervised classification while ARI was used as the metric for the unsupervised classification. F1 score was chosen because of the imbalance in the datasets. By using F1 score, we hoped to see whether each class could be classified correctly or not. The results of the experiments are as follows:

1) *Experiments on Unsupervised Classification with K-Means and K-Medoids*: The ARI score was computed using the class labels and the predicted labels. The maximum ARI score is one if the clusters between class labels and predicted labels are identical. Table V shows that the resulting ARIs that

we obtained are mostly near zero, meaning that our results are almost equal to the results of random clustering. Some of the scores are even lower than zero, less than what is expected from a random result. There are no significant differences between the ARI scores of K-Means and K-Medoids.

TABLE V
EXPERIMENTAL RESULTS OF UNSUPERVISED WORD CLASSIFICATION

| Model | Feature Size | Domain | ARI Score | |
|-----------|--------------|----------------|--------------|--------------|
| | | | K-Means | K-Medoids |
| CBOW | 300 | Restaurant | 0.039 | 0.048 |
| | | Handphone | 0.014 | -0.004 |
| | | Digital camera | 0.019 | 0.010 |
| | 400 | Restaurant | 0.040 | 0.041 |
| | | Handphone | 0.012 | -0.003 |
| | | Digital camera | 0.019 | 0.008 |
| Skip-gram | 300 | Restaurant | 0.031 | 0.018 |
| | | Handphone | 0.024 | 0.025 |
| | | Digital camera | 0.034 | 0.040 |
| | 400 | Restaurant | 0.027 | 0.016 |
| | | Handphone | 0.025 | 0.012 |
| | | Digital camera | 0.033 | 0.023 |

2) *Experiments on Supervised Classification with Cosine Similarity*: Based on the results displayed on Table VI, we could see that other than the classification on the restaurant domain with CBOW model, the accuracies of classification with word vectors of size 300 as the feature are higher than classification with word vectors of size 400. In the restaurant and digital camera domain, the use of Skip-gram model has given us better results than CBOW model, but in the handphone domain the reverse is true.

3) *Experiments on Supervised Classification with SVM Classifier*: In Table VI, we could see that the use of word vector created by CBOW model has given us better results than Skip-gram model. In all experiments other than the experiments in handphone domain by using word vectors created by Skip-gram model, the use of word vectors of size 300 results in better accuracy than using word vectors of size 400. In every domain, the highest accuracies are achieved by using word vectors of size 300 created by CBOW model.

TABLE VI
EXPERIMENTAL RESULTS OF SUPERVISED WORD CLASSIFICATION

| Model | Feature Size | Domain | With Cosine Similarity | | | | With SVM Classifier | | | |
|-----------|--------------|----------------|------------------------|--------------|--------------|--------------|---------------------|--------------|--------------|--------------|
| | | | F1 Score | | | Accuracy | F1 Score | | | Accuracy |
| | | | Aspect | Opinion | Other | | Aspect | Opinion | Other | |
| CBOW | 300 | Restaurant | 0.567 | 0.576 | 0.583 | 0.575 | 0.830 | 0.782 | 0.780 | 0.802 |
| | | Handphone | 0.741 | 0.649 | 0.642 | 0.686 | 0.833 | 0.796 | 0.746 | 0.798 |
| | | Digital camera | 0.544 | 0.707 | 0.461 | 0.528 | 0.750 | 0.737 | 0.700 | 0.730 |
| | 400 | Restaurant | 0.577 | 0.567 | 0.583 | 0.576 | 0.829 | 0.763 | 0.762 | 0.791 |
| | | Handphone | 0.736 | 0.635 | 0.635 | 0.678 | 0.834 | 0.789 | 0.720 | 0.790 |
| | | Digital camera | 0.526 | 0.568 | 0.470 | 0.520 | 0.743 | 0.747 | 0.692 | 0.726 |
| Skip-gram | 300 | Restaurant | 0.575 | 0.591 | 0.606 | 0.590 | 0.798 | 0.709 | 0.755 | 0.760 |
| | | Handphone | 0.700 | 0.616 | 0.617 | 0.652 | 0.811 | 0.740 | 0.706 | 0.762 |
| | | Digital camera | 0.578 | 0.566 | 0.464 | 0.539 | 0.737 | 0.717 | 0.701 | 0.719 |
| | 400 | Restaurant | 0.580 | 0.590 | 0.584 | 0.584 | 0.800 | 0.719 | 0.723 | 0.754 |
| | | Handphone | 0.689 | 0.607 | 0.603 | 0.640 | 0.819 | 0.744 | 0.704 | 0.766 |
| | | Digital camera | 0.573 | 0.562 | 0.447 | 0.531 | 0.722 | 0.696 | 0.695 | 0.706 |

B. Experiments on Aspect Extraction

We evaluated the performance of aspect extraction methods across four datasets, listed on Table VII. Canon G3, Nikon Coolpix 4300, and Nokia 6610 datasets are a part of Customer Review Datasets created by Hu and Liu [16], while ABSA16_Restaurants_Train_SB1 dataset is from SemEval 2016 Task 5: Aspect-Based Sentiment Analysis [17]. We compared the performance of modified Sequential Covering method with the baseline Sequential Covering method.

TABLE VII
ASPECT EXTRACTION DATASET STATISTICS

| Dataset | Count | |
|------------------------------|-----------------|--------|
| | Review Sentence | Aspect |
| Nikon Coolpix 4300 | 329 | 188 |
| Canon G3 | 569 | 262 |
| Nokia 6610 | 518 | 295 |
| ABSA16_Restaurants_Train_SB1 | 1708 | 1880 |

The results of the experiments are as follows:

1) *Baseline Sequential Covering without Lexicon*: On Table VIII, we could see that this method yields very high recall but low precision. The reason is because a lot of irrelevant opinion pairs are also extracted.

2) *Modified Sequential Covering*: We used the best configurations from the previous experiments to build the domain-specific lexicons that we used in this method. In all of the experiments, we used lexicons of size 1000.

a) *With Domain-Specific Lexicon Created Using Term Frequency and Cosine Similarity*: To create the lexicon, we used 50 most frequent words expanded with 20 most similar words for each word. The results on Table VIII shows that this method managed to yield better F1 scores than the baseline method, but the recalls are lower. For Nokia 6610 dataset, this method gives the highest precision and F1 score.

b) *With Domain-Specific Lexicon Created Using Term Frequency Only*: We used 1000 most frequent words to create the lexicon used in this experiment. On Table VIII, we could see that in every dataset other than the Nokia 6610 dataset, this lexicon gives better results than the lexicon created with the previous approach. Even though the recalls are still lower than the baseline Sequential Covering, the F1 scores from this method are higher than the other methods.

VI. THE COST AND BENEFIT OF USING DOMAIN-SPECIFIC LEXICON OVER ASPECT AND OPINION LIST

To build a domain-specific lexicon, we need labelled data. In our experiments, the labelling process is done manually, making it a rather costly process because it takes time and effort. Some knowledge of the domain is also necessary to make sure that the classes are labelled correctly.

Even so, the scope of the domain-specific lexicon is bigger than the original aspect and opinion list, allowing it to be used for multiple datasets in the same domain. We tried to reduce the cost by using POS tags to roughly split the data into their respective classes before going over the data manually.

VII. ISSUES ON THE DOMAIN-SPECIFIC LEXICON AND PRETRAINED WORD EMBEDDING

There are multiple issues on the domain-specific lexicon and the pretrained word embedding that could hinder the performance of the modified Sequential Covering method, causing it to not work as effectively as intended. We will address these issues in the following sections.

A. The Extracted Aspect/Opinion Does Not Exist in Word Embedding

The filtering process in Sequential Covering method uses cosine similarity. If a word does not exist in word embedding (OOV) then the similarity returned would be 0, causing the word to be marked as not relevant to the domain. While building the word embedding, the value of minimum count parameter is set to 5, meaning that if a word appears less than 5 times in the dataset, then it would be regarded as a noise.

An aspect could have a very low frequency because of the change of time causing a difference in aspects that are usually discussed in a review. This is especially more apparent in the product review. Nokia 6610, Canon G3, and Nikon Coolpix 4300 dataset were created in 2004, while the data used to build domain-specific lexicon were crawled in 2020. Example of aspects in the Nokia 6610 dataset that are not in the embeddings: “CSR”, “PIM”, and “PC Suite”.

If an aspect is misspelled in the review dataset, the aspect could also have a very low frequency. For example, the word “canera” exists as an aspect of the Canon G3 dataset. Because

TABLE VIII
EXPERIMENTAL RESULTS FOR ASPECT EXTRACTION

| Dataset | Sequential Covering without Lexicon | | | Modified Sequential Covering with TF+CS DSL | | | Modified Sequential Covering with TF DSL | | |
|------------------------------|-------------------------------------|--------------|-------|---|-------|--------------|--|-------|--------------|
| | P | R | F | P | R | F | P | R | F |
| Nikon Coolpix 4300 | 0.371 | 0.895 | 0.525 | 0.609 | 0.647 | 0.628 | 0.606 | 0.689 | 0.645 |
| Canon G3 | 0.311 | 0.877 | 0.459 | 0.508 | 0.582 | 0.543 | 0.525 | 0.651 | 0.581 |
| Nokia 6610 | 0.379 | 0.898 | 0.533 | 0.618 | 0.641 | 0.629 | 0.575 | 0.624 | 0.598 |
| ABSA16_Restaurants_Train_SB1 | 0.509 | 0.913 | 0.653 | 0.695 | 0.621 | 0.656 | 0.715 | 0.695 | 0.705 |

^aP: Precision; R: Recall; F: F1 Score; TF: Term Frequency; CS: Cosine Similarity; DSL: Domain-Specific Lexicon.

it is a misspelling from the word “camera”, there is a very low chance that the word would exist in the word embedding.

B. No Similar Aspect/Opinion Found in the Domain-Specific Lexicon

Because of the limitations on the size of domain-specific lexicons that we used in the experiments, some aspect or opinion words may not be included in the domain-specific lexicon. For example, in the Nokia 6610 dataset the aspects: ring, infrared, customer service, voice dialing, and vibration are not similar with any of the aspects in the domain-specific lexicon for handphone domain.

The smaller the frequency of the aspect/opinion words, the lower the probability that it would be included in the lexicon. To tackle this problem, we suggest to increase the size of domain-specific lexicon or try different combinations of most frequent words and most similar words.

C. Error in Labelled Data Used for Building Domain-Specific Lexicon

There is a possibility of human error in the data labelling process because it is conducted manually. This problem could cause a decrease in the performance of the modified Sequential Covering method.

VIII. CONCLUSION

We have adapted the works of Grefenstette and Muchemi [9] to automatically build domain-specific lexicon for the digital camera, handphone, and restaurant domain. In the development of domain-specific lexicon, we obtained the best results for word classification task by using SVM classifier with vectors of size 300 created by CBOW model. The resulting domain-specific lexicons were used in the modified version of Sequential Covering algorithm for aspect extraction [2]. Even though the recalls obtained from the modified method are still lower than the recalls obtained from baseline Sequential Covering, this method managed to yield better F1 scores than the baseline method.

For future works, we suggest increasing the size of the lexicons and using different combinations of most frequent words and most similar words. We also suggest reducing the scope of the domain to create smaller domains such as Nikon digital camera, Canon digital camera, and Nokia phone. By using a smaller domain, we hope that the aspects that are specific to the product would be included in the domain-specific lexicon.

ACKNOWLEDGMENT

The authors would like to thank LPPM-ITB for providing us with funding through the P3MI Program.

REFERENCES

- [1] B. Liu, *Sentiment Analysis and Opinion Mining*. San Rafael, CA: Morgan and Claypool Publishers, 2012.
- [2] F. Z. Ruskanda, D. H. Widyantoro, and A. Purwarianti, “Sequential covering rule learning for language rule-based aspect extraction,” 2019 Int. Conf. on Adv. Comput. Sci. and Inf. Syst. (ICACSIS).
- [3] H. Avancini, A. Lavelli, F. Sebastiani, and R. Zanolli, “Automatic expansion of domain-specific lexicons by term categorization,” *ACM Trans. on Speech and Lang. Process. (TSLP)*, 3(1):130, 2006.
- [4] S. N. Kim, T. Baldwin, and M.-Y. Kan, “Extracting domain-specific words - a statistical approach,” in *Proc. of the 2009 Australas. Lang. Technol. Assoc. Workshop*, Sydney, Australia, pp. 9498.
- [5] F. Xu, D. Kurz, J. Piskorski, and S. Schmeier, “A domain adaptive approach to automatic acquisition of domain relevant terms and their relations with bootstrapping,” *Proc. 3rd Int. Conf. Lang. Resour. and Eval. (LREC 2002)*, 2002.
- [6] R. Wang, W. Liu, and C. McDonald, “Featureless domain-specific term extraction with minimal labelled data,” in *Proc. of the Australas. Lang. Technol. Assoc. Workshop 2016*, Melbourne, Australia, December 2016, pp. 103112.
- [7] S. Park, W. Lee, and I.-C. Moon, “Efficient extraction of domain specific sentiment lexicon with active learning,” *Pattern Recognit. Lett.*, vol. 56, pp. 3844, 2015.
- [8] W. Hamilton, K. Clark, J. Leskovec, and D. Jurafsky, “Inducing domain-specific sentiment lexicons from unlabeled corpora,” *Proc. of the 2016 Conf. on Empirical Methods on Natural Lang. Process. (EMNLP)*, 2016.
- [9] G. Grefenstette and L. Muchemi, “Determining the characteristic vocabulary for a specialized dictionary using Word2vec and a directed crawler,” *Int. Conf. on Lang. Resour. and Eval. (LREC 2016)-GLOBALEX 2016*, pp.8185.
- [10] L. Zhuang, F. Jing, X.-Yan Zhu, and L. Zhang, *Movie review mining and summarization*, CIKM-06, 2006.
- [11] G. S. Chauhan and Y. K. Meena, “DomSent: domain-specific aspect term extraction in aspect-based sentiment analysis,” *Smart Systems and IoT: Innovations in Computing*, pp. 103-109, 2020.
- [12] D. Anand and D. Naorem, “Semi-supervised aspect based sentiment analysis for movies using review filtering,” *Procedia Comput. Sci.*, 84, 86-93.
- [13] A. Mukherjee and B. Liu, “Aspect extraction through semi-supervised modeling,” in *Proc. of the 50th Annu. Meeting of the Assoc. for Comput. Linguistics (Volume 1: Long Papers)* (pp. 339-348).
- [14] A. Trask, P. Michalak, and J. Liu, “Sense2vec-a fast and accurate method for word sense disambiguation in neural word embeddings,” *arXiv preprint arXiv:1511.06388*, 2015.
- [15] T. Joachims, “Text categorization with support vector machines: learning with many relevant features,” in *Proc. of ECML-98, 10th Eur. Conf. on Mach. Learn. (Chemnitz, Germany, 1998)*, 137-142.
- [16] M. Hu and B. Liu, “Mining opinion features in customer reviews,” *Am. Assoc. Artif. Intell.*, pp. 755760, 2004.
- [17] M. Pontiki et al., *SemEval-2016 task 5: aspect based sentiment analysis*, pp. 1930, 2016.
- [18] *Explosion AI, spaCy 101: Everything you need to know*. 2017 [Online]. Available: <https://spacy.io/usage/spacy-101>. [Accessed: 12- Jun- 2020]