A Similarity Score Model for Aspect Category Detection

Zohreh Madhoushi[1], Abdul Razak Hamdan[2], Suhaila Zainudin[3]
Faculty of Information Science and Technology
Universiti Kebangsaan Malaysia 43600 Bangi, Selangor, Malaysia

***Abstract*—Aspect-based Sentiment Analysis (ABSA) aims to extract significant aspects of an item or product from reviews and predict the sentiment of each aspect. Previous similarity methods tend to extract aspect categories at the word level by combining Language Models (LM) in their models. A drawback for the LM model is its dependence on a large amount of labelled data for a specific domain to function well. This work proposes a mechanism to address labelled data dependency by a one-step approach experimenting to decide the best combinatory architectures of recurrent-based LM and the best semantic similarity measures for fostering a new aspect category detection model. The proposed model addresses drawbacks of previous aspect category detection models in an implicit manner. The datasets of this study, S1 and S2, are from standard SemEval online competition. The proposed model outperforms the previous baseline models in terms of the F1-score of aspect category detection. This study finds more relevant aspect categories by creating a more stable and robust model. The F1 score of our best model for aspect category detection is 79.03% in the restaurant domain for the S1 dataset. In dataset S2, the F1-score is 72.65% in the laptop domain and 75.11% in the restaurant domain.**

Keywords*—Aspect category detection; language model; semantic similarity

I. Introduction

The first task in Aspect-based Sentiment Analysis (ABSA) is to detect aspects of an item or product from reviews and categories each aspect into a specific group. There are different methods to detect aspects; namely, language rule methods or LM, sequential methods, topic model methods, deep learning methods, and hybrid methods, which are the combination of the above methods [1]. We have seen exciting outcomes in various NLP tasks in recent years using these emerging models [2]–[5]. Regardless of the existing techniques, the most crucial point in any Natural Language Processing (NLP) task is to find a way to make machines understand language or text.

Deep learning techniques automate the process of representation learning in multi computational layers. These techniques have enabled researchers to improve state of the art for many NLP tasks such as Sentiment Analysis (SA) [6], [7] significantly and in other domains (image, speech, etc.). A drastic advancement happened in the text representation, and many LMs were developed, such as Word2Vec [8], deep LM [9], [10]. However, these emerging LMs have not yet fully addressed aspect category detection, mainly because there is no study to design experiments assessing the effect of different recent advanced LMs on the specific task of aspect category detection. This work hypothesis is that a general deep LM can be used for any specific task in NLP. As [11] states, using a pre-trained LM is better than starting a model with random weights therefore Deep LM is used as a form of transfer learning in this work. The task of a simple LSTM-based LM is to predict the next word in a large corpus. Instead of learning from scratch using random weights, the representation created with the proposed LM can be a better starting point for another LM. Then, it can be used for a specific task in a more related domain.

On the other hand, previous ABSA methods tend to extract aspect terms and categories them in two separate steps. Also, they set model threshold values and seed words for aspects categories manually. Domain-specific models are often not practical for this task. Various semantic similarity measures were proposed in the literature such as WordNet, NGD, Cosine and Soft Cosine. A few works utilize these semantic similarity measures to extract aspect categories in one phase; however, all works performed word-level similarity measurement. Recent works find the semantic similarity of a pair of text [12], [13] at the sentence level. But there is a lack of investigation addressing the effect of semantic similarity measures for the aspect category detection task at sentence level in the literature.

The main contributions of this paper are summarized as follows. First, this study presented a mechanism based on two semantic similarity measures and two standard LMs. The proposed mechanism is based on sentence-level similarity measurement, which extracts aspect categories in one step. The best combination of recurrent language and semantic similarity measures is investigated. This helps the researchers to find the best recent options for semantic similarity measure and neural LM for this task. Second contribution is that the model developed based on the above mechanism works in one step instead of two steps by utilizing semantic similarity measures at both word and sentence level and identifies aspect categories related to implicit aspects, mainly because of using the neural LM and still working in two different areas without any labeled data. The model works without setting seed words for each aspect category. The last contribution is solving the problem of setting several thresholds manually. The model proposed by current study set the thresholds automatically without any human intervention. Therefore, the model has reduced the amount of manual human intervention by removing the need for seed words and automatically setting thresholds. The proposed similarity score model is able to

fine-tune to any new domain by only adding pre-known aspect categories for the new domain and more related domain reviews without any labeled data.

This paper is organized as follow. In Section 2, the literature for aspect category detection is reviewed. Section 3 proposed a new model based on existing deep LM and semantic similarity measure. The experiment and result discussion on the dataset of this study is presented in Section 4. Finally, the study is concluded in Section 5.

II. Background of Study

There are many current works that focus on Language Rules Models [14]–[17]. These effective syntactic LM still have room for improvement. However, it is challenging to design a set of rules to perform well due to natural languages' flexibility. It also appears from the review that researchers mostly focused on extracting the aspect term instead of aspect category. Aspect category detection is similar to explicit and implicit aspect extraction and then grouping under one category. Explicit aspects are those that one can find the aspects clearly stated as nouns or noun phrases in a review, for example,' picture quality' in "The picture quality of this phone is great." Implicit aspects are not clearly stated in a review but are implied indirectly, for example, 'price' in "This laptop is so expensive". There are language rule models in the literature that extract implicit aspects. These models' problem is that they always need to extract an explicit or implicit aspect term to group them under one category. The above causes several category names creation for one unique category in new datasets. In other words, there are no predefined standard categories that one can assign aspect terms to those categories. They cannot identify aspect categories directly from a review text. However, most of these model does not group the extracted terms into predefined categories in the literature [16], [18]. For the aspect category detection model to be practical, one crucial step is to propose a model that works in fewer steps.

Until recently, a few studies [15], [19] attempted to extract aspect categories directly from review text in a single step using language rule methods. The intermediate task of aspect term extraction is required for most models [16], [20], [21] However, it is hard to use these models on new datasets since manual tuning of various thresholds is required [15], [22]. Another problem of recent models is that they need to find some synonyms for each aspect, in which the result depends on the selection of these synonyms words. They need these lists for every aspect and for every domain, which is a time-consuming activity. Various manual thresholds setting required for new datasets in the models that use similarity to predefined categories [15], [19].

The number of models focused on implicit aspect extraction increased in recent years. While aspect category detection can handle the implicit aspects and explicit aspects, it is much harder to extract implicit aspects with aspect term extraction. Sequential supervised models [23], [24] are better than language rule models to extract implicit aspects. These models' problem is that they need lots of labeled data, which is not easy to get for each area and domain separately. Again, for the aspect category detection models to be practical, the models must work in multiple domains or at least easy to apply or transfer to any new domain. Sequential and Modern deep learning models cannot work in different domain or need lots of training data in each domain [25]–[29]. Topic models on the other hand are too statistical centric which this study can hardly find improvement for it.

This study continues [19] for the model not to use any labeled data. The difference is that, instead of clustering and getting the similarity of a cluster with aspect categories, this study utilizes a representation from sentence-level deep LM. The proposed model does not require any seed words to be set for each aspect category anymore. There is a recent similar work in the literature solving the same issue. [30] rely on the similarity of sentence words and some seed aspects utilizing Word2Vec, Glove and Fastext. They state that their model performs the same as recent neural models for aspect extraction with a less computational cost. Their model is very similar to our similarity score model. However, the approach of finding more than one aspect category for a sentence is not explained clearly.

We have performed a comparison to prove the novelty of the proposed model. The state of the art is summarized via Table I.

The limitations of the previous models are summarized in the last column. The main limitations are first they perform the task of aspect category extraction in two separate phases; second, they need labeled data in every domain; third, only the word level similarity is performed, lastly, they need to set the parameters manually.

TABLE I. ASPECT CATEGORY DETECTION MODELS

Method of aspect category detection	Steps of aspect category detection	Author	Dataset	Domain	Result (average)	Limitation
Language rule (Dependency relation + similarity)	2	Garcia et al. (2014)	SemEval 2015	Laptop Restaurant Hotel	F-score aspects category: Laptop: 24.94 Restaurant: 41.85 Acc. Sentiment: Laptop 68.38 Restaurant 69.46 Hotel 71.09	- Cannot find implicit aspects and sentiment. -It cannot detect context orientation of opinion word.
Language rule (Graph based)	1	Schouten et al. (2017)	SemEval 2014	Restaurant	F1-score:67.0	-Needs to set synonym words for every aspect. -Lots of parameter setting is required.
Language rule (Similarity)	1	Ghadery et al. (2018)	SemEval 2014	Restaurant	F1-score:76.98	-Needs to set synonym words for every aspect. -Word level similarity is performed
Language rule (Similarity)	2	Gaillat et al. (2019)	SE-2015	Financial Microblogs	Accuracy: 42.5	The Word2Vec model is trained on the Google news corpus -Word level similarity is performed
Language rule (Lexicon based)	2	Alqaryouti et al. (2019)	Dataset of (Alqaryouti et al. 2019)	government smart apps	Aspect category detection: Precision Recall F-score 92.63. 84.03 88.12 Sentiment Accuracy: 93.01%	Intermediate task of aspect term extraction is required.
Deep Learning (Auto-encoder with attention)	1	He et al. (2017)*	Citysearch corpus BeerAdvocate	Restaurant review beer review	F1-score: Restaurant: 79.25 Beer: 73.56	-It is domain dependent. Needs large labeled data.
Deep Learning (LSTM)	1	Ma et al. (2018)	SemEval-2015	Restaurant	Aspect extraction: F1-score: 0.75 Sentiment detection Acc. 74.11	It is domain dependent. Needs large, labeled data.

III. PROBLEM STATEMENT

A group of similarity techniques emerged with the advance of text representations in language rule methods to extract aspect categories were summarized in Table I. Nevertheless, only one of these models perform the task in one single step. It is also hard to use these models on new datasets since manual tuning of various thresholds is required. Another problem of these models is that they need to find some synonyms for each aspect which the result depends on the selection of these synonym words. They need these lists for every aspect and for every domain, which is a time-consuming task. All these methods utilize the word level similarity measure.

Distributed representation of sentences and neural LMs are a good source of semantic similarity measurement in the literature. To the best of our knowledge, combining the sentence similarity measurement techniques and deep LMs have not been addressed for the aspect category detection task. Considering that deep learning model that used in the literature for this task are supervised and domain-dependent, this study aims to propose a mechanism for aspect category detection using sentence similarity measurement and recurrent-based LM without using labeled data.

The author in [31] defines recurrent language model where an input vector sequence x = (x1, . . . , xT) is passed through weighted connections to a stack of N recurrently connected hidden layers to compute first the hidden vector sequences hn = (hn1 , . . . , hn T) and then the output vector sequence y = (y1, . . . , yT). Each output vector yt is used to parameterize a predictive distribution Pr(xt+1|yt) over the possible next inputs xt+1. The first element x1 of every input sequence is always a null vector whose entries are all zero; the network, therefore, emits a prediction for x2, the first real input, with no prior information.

The hidden layer activations are computed by iterating the following equations from t = 1 to T and from n = 2 to N. W terms denote weight matrices in equation below. Wihn is the weight matrix connecting the inputs to the nth hidden layer, Wh1h1 is the recurrent connection at the first hidden layer, and so on. The b terms denote bias vectors, and H is the hidden layer function. Given the hidden sequences, the output sequence is computed as follows:

$$h_t^1 = H(W_{ih^1}x_t + W_{h^1h^1}h_{t-1}^1 + b_h^1)$$

$$h_t^n = H(W_{ih^n}x_t + W_{h^{n-1}h^n}h_t^{n-1} + W_{h^nh^n}h_{t-1}^n + b_{h\_}^n)$$

$$\hat{y_t} = b_y + \sum_{n=1}^{N} W_{h^ny}h_t^n$$

$$y_t = y(\hat{y_t})$$

where $\hat{y}$ is the output layer function. The objective function is cross entropy error as the sum over the entire vocabulary at time-step t. |V| is the vocabulary size.

$$J^{(t)}(\theta) = -\sum_{j=1}^{|V|} y_{t,j} \times log(\widehat{y_{t,j}})$$

The entire network therefore defines a function, which is parameterized by the weight matrices, from input histories x1:t to output vectors yt. The output vectors yt are used to parameterize the predictive distribution Pr(xt+1|yt) for the next input. Network direction, layers and variation can be experimented to find the best combination for the LM.

Based on the above discussion and limitations discussed in previous section, there are two general problems in this study.

1) Lack of utilizing existing recurrent-based LMs and experiment on finding the best combinatory architectures of LMs and the best semantic similarity measures for aspect category detection task.

2) Lack of aspect category detection model for fetching aspect categories without the intermediate task of aspect (explicit and implicit) term extraction using no labeled data.

IV. Materials and Methods

The idea to extract the most related aspect category is that with a suitable vector representation of a sentence and pre-known aspects both, a comparison can be performed to find the similarity. The data sets of this study include a list of aspect categories for each area. Even if the list is not available, people usually talk about these aspects found in online review websites. These pre-known aspects are a good source of detecting other aspects with similar meanings. Therefore, a pre-trained Word2Vec LM is used for training. Then, another LM is trained on top of the initial LM with Amazon product review dataset in fourteen areas at the sentence level, and then the model is fine-tuned for the in-domain dataset which is on laptop, restaurant and hotel. Fine-tuning means training an existing model with a new dataset or continuing the current LM training with more data. Fine-tuning in deep learning model means only the model architecture or weights of the model is used in a new model. In this study the initial layers are frozen, not to change the weights of the pretrained LM and top layer are trained. This approach has been used in recent LMs [10], [32]. In this study, the existing Skip-gram LM is fine-tuned with a new dataset and LSTM-based LM is fine-tuned by freezing the initial layer and continuing the training with the more related dataset.

Once the specific LM is trained, the sentence representation that is created can be used for any NLP task. Therefore, for any piece of text a meaning that this LM creates can be represented. This information is a good source for comparing texts. Texts or sentences with similar meaning are close in space. The sentence representation of the final LM is used to extract aspect category detection by comparing the representation of a list of aspects with each review in our in-domain dataset. Therefore, for this LM, the unlabeled S1 and S2 datasets, the list of our aspect categories, a piece of text with meaning and dataset M, is appended. The second and final LM representation will be used to get the semantic similarity of review text and aspect categories. Fig. 1 shows the proposed idea in action. The first LM from left is famous Skip-gram LM which has been pre-trained on Google News. The second LM is the same Skip-gram LM that was fine-tuned on Amazon review dataset (A) with 14 areas and S1 and S2, which are laptop, restaurant, and hotel area. The third and fourth LMs are LSTM-based LMs which simply predicts the next word. The unlabeled datasets S1 and S2 and dataset A are appended.

The bidirectional form of LSTM is used for the LM. This study does not use the LM for text generation; therefore, predicting the next word in both directions is useful for getting better representation of words and sentences. The network is fed with representation of each word in the string that is gained from the training. The model is used for aspect string set (A) and sentence string set (S). Aspect string and sentence string is defined as follow:

$$a = x_{a_1}, x_{a_2}, \dots, x_{a_j}$$

$$s = x_1, x_2, \dots, x_j$$

Where x_j, x_{aj} are vector representation of each word in sentence and aspect string respectively which is gained from the LSTM LM. After freezing the model, maximum, minimum, average and last cell representation are concatenated to define each sentence vector (h_j) and each aspect vector (h_{aj}). If the Soft Cosine similarity of h_j with any of aspects vector (h_{aj}) is more than a specific threshold then the candidate aspect is remained, otherwise it will be ignored. Cosine similarity is a measure of similarity between two vectors of an inner product space that measures the cosine of the angle between them. The Soft Cosine of two vectors is introduced by Sidorov et al. (2014). They propose to modify the calculation of cosine similarity taking into account similarity of features. They named the traditional cosine as "hard cosine", which ignores similarity of features. Given two vectors of attributes, a and b, the Soft Cosine can be derived by using the following formula:

$$soft - cosine(a, b) = \frac{\sum\sum_{i,j=1}^{N} a_{ij} b_{ij}}{\sqrt{\sum\sum_{i,j=1}^{N} a_{ij}^2} \sqrt{\sum\sum_{i,j=1}^{N} b_{ij}^2}}$$

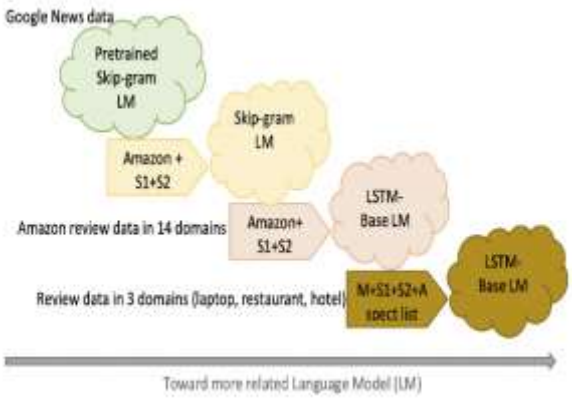

Fig. 1. Fine-tuning LM for the Specific Domain.

To speed up, get rid of for loops and reduce the complexity, vectorization is used for the implementation of similarity measurement. Instead of using each sentence and aspect vector at a time to measure the Soft Cosine similarity, vectorization gets a whole set of sentence and aspect vectors as matrix and compute the similarity of two matrices. Sentence matrix H and aspect matrix H_a is defined as follow, where *n* and *m* are the number of samples in sentence set (S) and aspect set (A) respectively:

$$H = [h_{j1}, h_{j2}, \ldots\ldots, h_{jn}]$$

$$H_a = [h_{aj1}, h_{aj2}, \ldots\ldots, h_{ajm}]$$

If the similarity surpasses a specific threshold, the model adds it to aspect set one. The similarity score model has two modules of deep and non-deep learning. The deep learning module is explained above. Fig. 2 shows the process of this model. This model finds the best result with a simple linear search. A score function is used to calculate a similarity score for each category in our dataset.

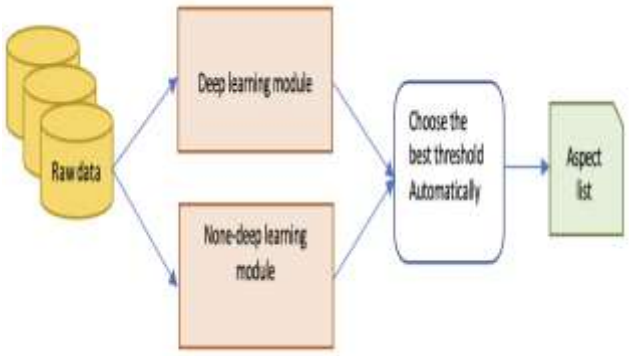

Fig. 2. Similarity Score Model Process.

The similarity score is a combination of the deep and non-deep learning score. Let |pos| be a set of all nouns, verbs, adjectives, and adverbs of a given sentence x and |aspect| be a set of our pre-known aspects. The similarity of the non-deep learning part for one aspect category of the given sentence is calculated from the maximum Soft Cosine similarity of that category to all words of that review after pos-tagging, which this work named it |pos|, is shown in Equation 1. Let h and h_a be the sentence and aspect representation that is taken from our LSTM based LM for the same given sentence and aspect respectively. The similarity of the deep learning part for this aspect category is calculated from the Soft Cosine similarity of h and h_a. Equation 3 shows the calculation of this score. Also, both similarity scores are calculated using sigmoid function on deep and non-deep learning similarity result same as our first model as in equation 2 and 4. Our final score is calculated with the interpolation of second score within the first score as shown in equation 5.

V. Experiment

Previous sections presented our models for aspect category detection. This section explains about the dataset that is used in this study and discusses the evaluation method for this task. We are interested in producing a sentence representation that keeps the sentiment of several aspects from different areas. Therefore, this study chooses the Amazon product review dataset (A) introduced in [33] as a training corpus to learn the distributed representation of the sentences. This dataset contains over 82 million product reviews from May 1996 to July 2014, amounting to over 38 billion training bytes. The second dataset is the dataset of SemEval 2014 competition task 4 (S1).

A. Baselines

The model is compared to the state-of-the-art models for aspect category detection. To have a fair comparison, the comparison partners should be aspect category detection models with no labeled data. Also, they should work on a multi-domain dataset. But due to lack of enough similar work for this task, this study shall compare the model with nearest work available. The models are compared with seven baselines. These baselines are for aspect category detection and not for aspect term extraction. Two baselines are on S1, three baselines are on S2 and one baseline is on both datasets. The following subsections describe about these baselines separately. Language rule baselines are presented in subsections a, b and c. Deep learning baselines including the winner of SemEval 2015-task 12 is remined in sub section d. The result of aspect category detection is presented in these works either as a direct task or as a subtask of aspect term extraction.

The first baseline is V3 [20] is a language rule model on both S1 and S2 that does not need labeled data. They use a similar implementation of [34] on SemEval 2014 task 4 dataset to extract aspect terms first and then compare with the category words using the similarity measure. The category with the highest similarity measure is then selected if it surpasses a manually set threshold. And d is SemEval 2014 task 4 baselines.

|pos| = p1, ..., ps
|aspect| = a_1, ..., a_k

$$SentSim1_{ai}(x) = Max_{j=1}^{|p|}\left(Softcosim(p_j, a_i)\right) \quad (1)$$

$$SentScore1_{ai}(x) = \frac{e^{SentSim1\_ai\,(x)}}{1+e^{SentSim1_{ai}(x)}} \quad (2)$$

$$SentSim2_{ai}(x) = Softcosim(h, h_{ai}) \quad (3)$$

$$SentScore2_{ai}(x) = \frac{e^{SentSim2\_ai\,(x)}}{1+e^{SentSim2_{ai}(x)}} \quad (4)$$

$$Aspect\_Score = \alpha * SentScor1 + (1-\alpha) * SentScore2 \quad (5)$$

Spreading activation is the second baseline. [15] developed a model called spreading activation that does not need labeled data. They used some seed words and co-occurrence matrix of words to create a digraph for aspect category detection using association rule mining. The similarity baselines are the closest baselines to our model since they try to find the similarity of a given sentence to some pre-known categories. The baseline is [19] model. They use similarity of average word vectors to pre-known aspect categories. They cluster sentences and use the closest cluster's similarity to a given sentence as different similarity measurement. Our model is very similar to [19]. The difference is instead of averaging the word embedding of all the words in a sentence; the sentence

embedding is gained from the LSTM based LM. Also, instead of clustering sentences to get the second similarity measurement to the pre-known aspect categories, sentences are POS-tagged. The similarity of each selected word to the pre-known aspect categories is calculated.

One of the deep learning baselines is SemEval 2015-task 12, NLANGP [35], on S2. They used feedforward network with sigmoid to train binary classifiers for each category in the training set. Another recent deep learning baseline is [36]. They proposed an LSTM base model which combines implicit and explicit knowledge. The model adopted a sequence-encoder and a self-attention mechanism to calculate and incorporate common-sense knowledge into LSTM-based model to jointly extract aspect categories and predict sentiment for them.

B. Result and Discussion

The developed models can solve the problems that we discussed in the previous models. Both models use no labeled data. They performed the aspect category detection in one single step. The models can identify aspect categories for implicit aspect as well.

Cosine is the most common similarity measurement method, and Soft Cosine is an improvement over it. There are also modern LMs in the literature which Word2Vec and LSTM-based LMs are most common among them. We compare the F1-score of aspect category detection using two similarity measurement method, namely Cosine and Soft Cosine and two LMs, namely Word2Vec and recurrent base LM, with a few architecture differences on S1 and S2 respectively. Soft Cosine with two layers Bi-LSTM initialized with Word2Vec trained on Amazon dataset shows better performance than other combinations by scoring F1 score of 76.25 for Laptop and 75.11 for Restaurant. Therefore, this combination is used for the rest of the comparisons with baselines of this study for aspect category detection.

The comparison is done comparing the baselines F1 score for aspect category detection in two domains. Consider that the similar work in the baselines, [15], [19], need to set a large number of seed words for each aspect category. The result is undoubtedly related to choosing the right list of seed words. The model of this study does not need to choose a set of synonyms for each aspect category. Also, this model sets the similarity threshold automatically which makes it more robust and applicable to different datasets and areas. The F1-score results of dataset S1 and S2 are presented in Tables II and III on S1 and S2, respectively. Since there is no work in the baselines reported on the S1 laptop dataset; therefore, the presented results for S1 are only on the restaurant domain. Also, no work is reported to the author's knowledge on the aspect term or aspect category detection performance on the hotel domain on the S2 dataset. The presented results are from baseline's reported results, which have been explained in this study's scope. The result on S1 shows that the similarity score model performs better than the unsupervised language rule baseline V3 and only outperform by two supervised baselines. V3 is suitable to extract explicit aspects, but it performs so poorly when it comes to implicit aspects. Because the dataset is full of implicit aspects, its performance is lower than the similarity score model. This fact is more precise about the results on S2, and the reason is that most of the reviews contain implicit aspects. The similarity score model performs better than Ghaderi [19]. The model uses the average word vectors from pre-trained Word2Vec on Google news to represent each sentence, while for the similarity score model, the LSTM based LM is trained on a large related review dataset (A) to get the representation for sentences.

As presented in Section 4.3, 77% of the aspect categories are related to implicit aspect in this dataset. This number is 83% in S2. Therefore, the model can find explicit aspects and many of the implicit aspects in two different domains. The result also shows that the similarity score model is more stable than the baselines in various domains. As stated above a drawback of [19] and Spreading activation models is that the models need a set of manually pre-known aspect seeds for each category. They reported the result on S1 only. The number of aspect categories for laptop domain is 70. In this study, the work is replicated on the restaurant area for S2 dataset but not on laptop area because the seed data is not available for this area. To develop their model, one needs to generate an extensive list of seed words for 70 aspect categories which is not available in this study.

To find the best threshold alpha for the similarity score model a linear search is performed. Fig. 3 shows the sensitivity of the similarity score model to different thresholds. It shows that the optimum value for alpha is around 0.75 for laptop domain, about 0.70 for restaurant domain and around 0.65 for hotel domain.

Because of the high number of aspect categories in each domain, the results of classifiers NLANGP Toh & Su (2015) and Ma et al. (2018) are lower than similarity score model. Another drawback of these models is that they need a large number of labeled data to improve their performance on any dataset and domain.

TABLE II. ASPECT CATEGORY DETECTION F1-SCORE RESULTS ON S1.

Model/dataset	**Restaurant**
V3	60.20
Spreading activation	67.0
(Ghadery et al., 2018) [18]	76.98
Similarity score model	79.03

TABLE III. ASPECT CATEGORY DETECTION F1-SCORE RESULTS ON S2.

Model	**Laptop**	**Restaurant**
V3	24.94	41.85
NLANGP	50.86	62.68
(Ma et al. 2018) [34]	69.85	75.00
(Ghadery et al., 2018) [18]	-	73.96
Similarity score model	72.65	75.11

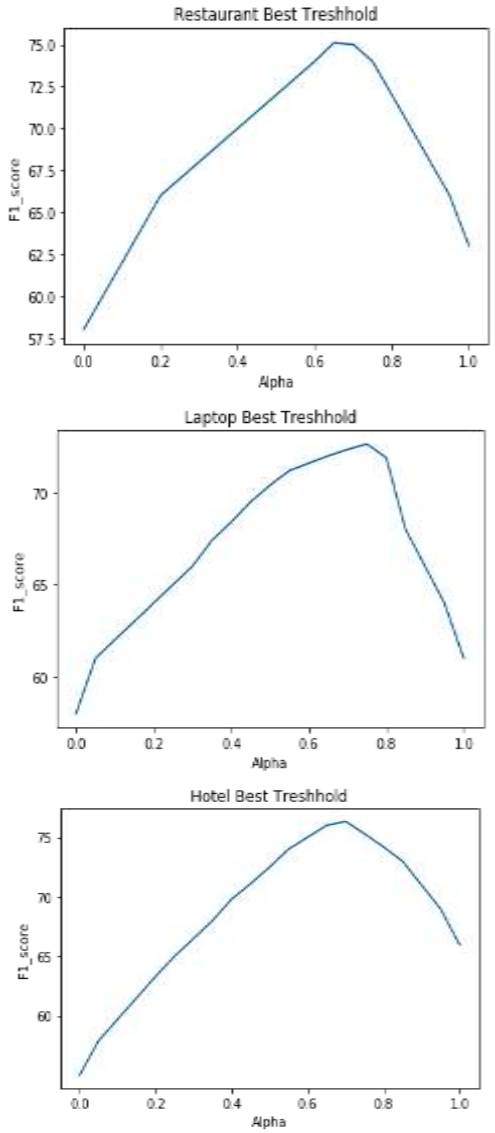

Fig. 3. Best Alpha Thresholds for our three Areas.

VI. Conclusion and Future Work

This study presented a new mechanism based on recurrent-based LM and semantic similarity measure for aspect category detection. A new model for aspect category detection was proposed, combining the above mechanism with existing language rules to extract aspect category in one step. The proposed similarity score model sets the similarity threshold automatically with a linear search. The f1-score results are presented for aspect category detection and compared with the baselines of this study on two datasets of S1 and S2. The work shows the priority of the proposed model compares to baselines on both datasets.

Unsupervised deep LMs may be effective in other NLP tasks since the context of previous and next sentences in a review affects the aspect category detection of the whole review. A direction for future is to work on review level instead of sentence level and extract all aspect categories of a given review. Also, one can investigate the best approach to replace ambiguous words in the review. For example, in a review with two sentences "the sushi is one of the best. You will find it delicious if you try it". One can do dependency parsing to find out that 'it' relates to Sushi in the sentence and replace 'it' with sushi.

Acknowledgment

Acknowledgement for grants GUP-2020-089 dan TT-2020-015 from Universiti Kebangsaan Malaysia for the support.

References

[1] Z. Madhoushi, A. R. Hamdan, and S. Zainudin, "Aspect-Based Sentiment Analysis Methods in Recent Years," Asia-Pacific J. Inf. Technol. Multimed., vol. 08, no. 01, pp. 79–96, 2019, doi: 10.17576/apjitm-2019-0801-07.

[2] Z. Madhoushi, A. R. Hamdan, and S. Zainudin, "Sentiment analysis techniques in recent works," 2015, doi: 10.1109/SAI.2015.7237157.

[3] I. S. Ahmad, A. Abu Bakar, M. R. Yaakub, and M. Darwich, "Beyond sentiment classification: A novel approach for utilizing social media data for business intelligence," Int. J. Adv. Comput. Sci. Appl., vol. 11, no. 3, pp. 437–441, 2020, doi: 10.14569/ijacsa.2020.0110355.

[4] S. M. Al-Ghuribi, S. A. Mohd Noah, and S. Tiun, "Unsupervised Semantic Approach of Aspect-Based Sentiment Analysis for Large-Scale User Reviews," IEEE Access, vol. 8, no. December, pp. 218592–218613, 2020, doi: 10.1109/ACCESS.2020.3042312.

[5] J. Awwalu, A. A. Bakar, and M. R. Yaakub, "Hybrid N-gram model using Naïve Bayes for classification of political sentiments on Twitter," Neural Comput. Appl., vol. 31, no. 12, pp. 9207–9220, 2019, doi: 10.1007/s00521-019-04248-z.

[6] D. S. Sachan, M. Zaheer, and R. Salakhutdinov, "Revisiting LSTM networks for semi-supervised text classification via mixed objective function," 33rd AAAI Conf. Artif. Intell. AAAI 2019, 31st Innov. Appl. Artif. Intell. Conf. IAAI 2019 9th AAAI Symp. Educ. Adv. Artif. Intell. EAAI 2019, pp. 6940–6948, 2019, doi: 10.1609/aaai.v33i01.33016940.

[7] C. Raffel et al., "Exploring the limits of transfer learning with a unified text-to-text transformer," arXiv, vol. 21, pp. 1–67, 2019.

[8] Y. Li, P. Xu, and M. Pang, "Adversarial Attacks on Word2vec and Neural Network," in ACAI 2018: Proceedings of the 2018 International Conference on Algorithms, Computing and Artificial Intelligence, 2018, pp. 1–5, doi: 10.1145/3302425.3302472.

[9] Y. Gu et al., "An enhanced short text categorization model with deep abundant representation," World Wide Web, vol. 21, no. 6, pp. 1705–1719, Nov. 2018, doi: 10.1007/s11280-018-0542-9.

[10] A. Radford, R. Jozefowicz, and I. Sutskever, "Learning to Generate Reviews and Discovering Sentiment," arXiv Prepr., 2017, Accessed: Aug. 13, 2017. [Online]. Available: https://arxiv.org/pdf/1704.01444.pdf.

[11] L. Li et al., "On Robustness and Bias Analysis of BERT-based Relation Extraction," pp. 1–17, 2020, [Online]. Available: http://arxiv.org/abs/2009.06206.

[12] H. T. Nguyen, Q. H. Vo, and M. Le Nguyen, "A Deep Learning Study of Aspect Similarity Recognition," Proc. 2018 10th Int. Conf. Knowl. Syst. Eng. KSE 2018, pp. 181–186, 2018, doi: 10.1109/KSE.2018.8573326.

[13] H. Gong, T. Sakakini, S. Bhat, and J. Xiong, "Document similarity for texts of varying lengths via hidden topics," ACL 2018 - 56th Annu. Meet. Assoc. Comput. Linguist. Proc. Conf. (Long Pap., vol. 1, pp. 2341–2351, 2018, doi: 10.18653/v1/p18-1218.

[14] S. Poria, E. Cambria, L.-W. Ku, C. Gui, and A. Gelbukh, "A rule-based approach to aspect extraction from product reviews," Soc. 2014, p. 28, 2014.

[15] K. Schouten, O. Van Der Weijde, F. Frasincar, and R. Dekker, "Supervised and Unsupervised Aspect Category Detection for Sentiment Analysis With," IEEE Trans. Cybern., pp. 1–13, 2017.

[16] T. Gaillat, B. Stearns, G. Sridhar, R. McDermott, M. Zarrouk, and B. Davis, "Implicit and Explicit Aspect Extraction in Financial

Microblogs," pp. 55–61, 2019, doi: 10.18653/v1/w18-3108.

[17] C. Henríquez, F. Briceño, and D. Salcedo, "Unsupervised model for aspect-based sentiment analysis in Spanish," IAENG Int. J. Comput. Sci., vol. 46, no. 3, 2019.

[18] R. Panchendrarajan, N. Ahamed, B. Murugaiah, P. Sivakumar, S. Ranathunga, and A. Pemasiri, "Implicit Aspect Detection in Restaurant Reviews using Cooccurence of Words," pp. 128–136, 2016, doi: 10.18653/v1/w16-0421.

[19] E. Ghadery, S. Movahedi, H. Faili, and A. Shakery, "An Unsupervised Approach for Aspect Category Detection Using Soft Cosine Similarity Measure," 2018, [Online]. Available: http://arxiv.org/abs/1812.03361.

[20] A. Garcıa-Pablos, M. Cuadros, and G. Rigau, "V3: Unsupervised Aspect Based Sentiment Analysis for SemEval-2015 Task 12," SemEval-2015, pp. 714–718, 2015, [Online]. Available: https://www.aclweb.org/anthology/S15-2121.

[21] F. Nurifan, R. Sarno, and K. R. Sungkono, "Aspect based sentiment analysis for restaurant reviews using hybrid ELMo-wikipedia and hybrid expanded opinion lexicon-senticircle," Int. J. Intell. Eng. Syst., vol. 12, no. 6, pp. 47–58, 2019, doi: 10.22266/ijies2019.1231.05.

[22] T. A. Rana and Y. N. Cheah, "A two-fold rule-based model for aspect extraction," Expert Syst. Appl., vol. 89, pp. 273–285, 2017, doi:10.1016/j.eswa.2017.07.047.

[23] S. Chatterji, N. Varshney, and R. K. Rahul, "AspectFrameNet: a frameNet extension for analysis of sentiments around product aspects," J. Supercomput., vol. 73, no. 3, pp. 961–972, 2017, doi: 10.1007/s11227-016-1808-6.

[24] W. Wang, S. J. Pan, D. Dahlmeier, and X. Xiao, "Recursive Neural Conditional Random Fields for Aspect-based Sentiment Analysis," 2016, Accessed: Nov. 11, 2017. [Online]. Available: https://arxiv.org/pdf/1603.06679.pdf.

[25] S. Kiritchenko, X. Zhu, C. Cherry, and S. Mohammad, "NRC-Canada-2014: Detecting aspects and sentiment in customer reviews," in Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014), 2014, pp. 437–442.

[26] D. Marcheggiani, O. Täckström, A. Esuli, and F. Sebastiani, "Hierarchical multi-label conditional random fields for aspect-oriented opinion mining," Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics), vol. 8416 LNCS, pp. 273–285, 2014, doi: 10.1007/978-3-319-06028-6_23.

[27] M. Yang, W. Tu, J. Wang, F. Xu, and X. Chen, "Attention-Based LSTM for Target-Dependent Sentiment Classification," pp. 5013–5014, 2017, doi: 10.1146/annurev.neuro.26.041002.131047.

[28] E. Cambria, B. B. Schuller, Y. Xia, and C. Havasi, "Building a Sentiment Summarizer for Local Service Reviews," IEEE Intell. Syst., vol. 28, no. 2, pp. 15–21, 2013, doi: 10.1109/MIS.2013.30.

[29] Y. Ma, H. Peng, T. Khan, E. Cambria, and A. Hussain, "Sentic LSTM: a Hybrid Network for Targeted Aspect-Based Sentiment Analysis," Cognit. Comput., vol. 10, no. 4, pp. 639–650, 2018, doi: 10.1007/s12559-018-9549-x.

[30] D. S. Vargas, L. R. C. Pessutto, and V. P. Moreira, "Simple Unsupervised Similarity-Based Aspect Extraction," 20th Int. Conf. Comput. Linguist. Intell. Text Process. (CICLing 2019), 2020, [Online]. Available: http://arxiv.org/abs/2008.10820.

[31] A. Graves, "Generating Sequences With Recurrent Neural Networks," Arxiv, pp. 1–43, 2013, [Online]. Available: http://arxiv.org/abs/1308.0850.

[32] R. Kiros et al., "Skip-Thought Vectors," Adv. neural in- Form. Process. Syst., pp. 3294–3302, 2015, doi: 10.1017/CBO9781107415324.004.

[33] J. McAuley, C. Targett, Q. Shi, and A. Van Den Hengel, "Image-based recommendations on styles and substitutes," SIGIR 2015 - Proc. 38th Int. ACM SIGIR Conf. Res. Dev. Inf. Retr., pp. 43–52, 2015, doi: 10.1145/2766462.2767755.

[34] G. Qiu, B. Liu, J. Bu, and C. Chen, "Opinion word expansion and target extraction through double propagation," Comput. Linguist., vol. 37, no. 1, pp. 9–27, 2011.

[35] Z. Toh and J. Su, "NLANGP: Supervised Machine Learning System for Aspect Category Classification and Opinion Target Extraction," Proc. 9th Int. Work. Semant. Eval., vol. 14, no. SemEval, pp. 496–501, 2015.

[36] Y. Ma, H. Peng, and E. Cambria, "Targeted Aspect-Based Sentiment Analysis via Embedding Commonsense Knowledge into an Attentive LSTM," 2018.