LICD: A Language-Independent Approach for Aspect Category Detection

Erfan Ghadery[1(✉)], Sajad Movahedi[1], Masoud Jalili Sabet[2], Heshaam Faili[1], and Azadeh Shakery[1]

[1] School of ECE, College of Engineering, University of Tehran, Tehran, Iran
{erfan.ghadery,s.movahedi,hfaili,shakery}@ut.ac.ir
[2] Center for Information and Language Processing (CIS) LMU Munich, Munich, Germany
masoud@cis.lmu.de

Abstract. Aspect-based sentiment analysis (ABSA) deals with processing and summarizing customer reviews and has been a topic of interest in recent years. Given a set of predefined categories, Aspect Category Detection (ACD), as a subtask of ABSA, aims to assign a subset of these categories to a given review sentence. Thanks to the existence of websites such as Yelp and TripAdvisor, there exist a huge amount of reviews in several languages, and therefore the need for language-independent methods in this task seems necessary. In this paper, we propose Language-Independent Category Detector (LICD), a supervised method based on text matching without the need for any language-specific tools and handcrafted features for identifying aspect categories. For a given sentence, our proposed method performs ACD based on two hypotheses: First, a category should be assigned to a sentence if there is a high semantic similarity between the sentence and a set of representative words of that category. Second, a category should be assigned to a sentence if sentences with high semantic and structural similarity to that sentence belong to that category. To apply the former hypothesis, we used soft cosine measure, and for the latter, word mover's distance measure is utilized. Using these two measures, for a given sentence we calculate a set of similarity scores as features for a one-vs-all logistic regression classifier per category. Experimental results on the multilingual SemEval-2016 datasets in the restaurant domain demonstrate that our approach outperforms baseline methods in English, Russian, and Dutch languages, and obtains competitive results with the strong deep neural network-based baselines in French, Turkish, and Spanish languages.

Keywords: Aspect-based sentiment analysis · Aspect category detection · Consumer reviews · Soft cosine measure · Word mover's distance

1 Introduction

No one shops alone, even someone who goes shopping alone. People usually care about other people's comments and recommendations. With the advent of web

L. Azzopardi et al. (Eds.): ECIR 2019, LNCS 11437, pp. 575–589, 2019.
https://doi.org/10.1007/978-3-030-15712-8_37

2.0 people tend to express their opinions on the web and share their experiences with each other. Therefore, sentiment analysis and opinion mining for online reviews are attracting a lot of attention [19].

One of the subtasks introduced in SemEval-2016 [21] ABSA is Aspect Category Detection (ACD). Given a review sentence and a set of pre-defined categories, ACD aims to assign a subset of these categories to the review sentence. The pre-defined categories consist of two subcategories: Attribute and Entity in the form of A#E (e.g. 'RESTAURANT#GENERAL', 'FOOD#QUALITY'). For example, the sentence 'The food here is rather good, but only if you like to wait for it.' contains a sentiment about the aspect FOOD#QUALITY while containing another sentiment about the aspect SERVICE#GENERAL. Therefore, given the above sentence, an ACD method should assign these two categories to the review sentence.

Most of the previously developed systems on this task are based on supervised machine learning techniques. Among these supervised methods, some systems use classic classification algorithms such as SVM and Logistic Regression [13,32]. Typically, these methods need a lot of effort to extract hand-crafted features which is time-consuming, and the performance of these methods are dependent on selecting an appropriate set of features. On the other hand, following the recent interest in neural network methods, some authors proposed neural network-based approaches [31,33,34]. Eventually, the performance of these neural-based methods depends on the availability of a sufficient amount of training data. In this paper, we propose a language-dependent method that does not require expensive hand-crafted features or language-dependent external resources[1]. Also, our method can perform ACD well even with a small amount of training data, which can be an advantage, especially for low-resource languages.

Our proposed method, LICD, detects categories belonging to a given sentence based on two hypotheses as follows. First, the given sentence belongs to a specific category if it has high semantic similarity to a set of key-words that represent that category. Second, the semantic and structural similarity between sentences is used. If a given sentence is close to another sentence of a specific category with regard to the aforementioned features, then, the given sentence also should belong to that category. To explore the first hypothesis, for each category, we choose a set of words using a feature selection method. These words constitute the representative set of that category. Then, we calculate the semantic similarity between the given sentence and each representative set to obtain a similarity score for each sentence category pair. In order to assess the mentioned semantic similarity, we utilize soft cosine similarity measure [27]. To inspect the second hypothesis, given a sentence, we retrieve the k-most semantically and structurally similar sentences to the sentence. Therefore, we need a similarity metric that measures both structural and semantic similarity at the same time. In this paper, we used the word mover's distance [14] to serve as the similarity metric for this purpose. The similarity score between a given sentence and a category will

[1] Note that, if we want to remove stopwords in the preprocessing step a list of stopwords is required.

be the sum of the inverse of word mover's distance values over the neighbor sentences that contain that category. Based on the two hypotheses mentioned above and using soft cosine measure and word mover's distance, we calculate a set of similarity scores for a given sentence and provide these scores as features to train a set of one-vs-all logistic regression classifier (one classifier for each category), where the output of classifiers is a probability distribution over the predefined categories. Categories that exceed a threshold are assigned to the given sentence. It is worth noting that instead of soft cosine and word mover's distance, any measure that has similar characteristics can be used.

We evaluate our language-independent approach on the multilingual SemEval-2016 datasets [21] in the restaurant domain with data available in 6 languages. Experimental results show that our method outperforms baseline methods in English, Russian, and Dutch languages while obtaining competitive results compared to baselines in French, Turkish, and Spanish.

2 Related Works

ABSA has been well studied in recent years. Schouten and Frasincar's work [26] provides a comprehensive survey on aspect level sentiment analysis specifically. The pioneering work of Hu and Liu [10] started this field. They used Association rule mining to extract frequent nouns and noun phrases assuming aspect terms to be noun or noun phrases, followed by applying a set of rules to prune redundant aspect terms. Another early work in this field is [28]. In this paper, authors detect implicit aspects using point-wise mutual information (PMI) to discriminate aspects from notional words.

In [13], authors tackle ACD using a set of one-vs-all SVM classifiers, one for each category, trained with features extracted from lexicon resources. They used n-grams, word clusters, etc. learned from the Yelp dataset as features. This system was ranked 1st in SemEval-2014 ACD sub-task. [32] has done similar work, but trained an SVM classifier for each predefined Entity and Attribute. They created a set of lexicons using train data, where lexicons are stemmed and un-stemmed n-grams with their precision, recall, and F1-scores calculated from the train data. Ganu et al. in [9] also proposed to use one-vs-all SVM classifiers for aspect category detection. They just used stemmed words as features. Unlike these methods, our method does not need expensive feature engineering and relies only on similarity measures.

In recent years, neural network- based methods, especially deep learning methods, have been proposed to address the ACD task. Khalil et al. [11] used a combination of a Convolutional Neural Network (CNN) and an SVM classifier for predicting aspect categories of a given sentence in the restaurant domain. They used term frequency (TF) weighted by inverse document frequencies (IDF) as features to train an SVM classifier. Also, for the laptop domain, they used one CNN classifier. Zhou et al. [34] proposed a method to learn the representation of words on a large set of reviews with noisy labels. After obtaining word vectors, through a neural network stacked on the word vectors, they generate deeper

and hybrid features for providing to a logistic regression classifier. Deep neural-based methods rely on sufficient enough of training data in order to perform well. However, our proposed method has acceptable performance even with a small amount of training data. Therefore, in low-resource languages, our method can be an alternative to deep learning-based methods.

3 Technical Ingredients

In this section, we will describe Word Mover's Distance and Soft Cosine Similarity measures in more details.

3.1 Word Mover's Distance

Word mover's distance [14] is a text distance measure, inspired from the Earth Mover's Distance [23]. This method interprets the distance between two documents as a transformation problem. Therefore, the distance between two documents is the distance that the embedded words of the first document need to travel to become the embedded words of the second document in the embedding space.

Let D and D' be two documents and let $X \in R^{d \times n}$ be the embedded word vectors, where d and n are embedding dimension and number of words, respectively. x_i is embedding vector of word i in R^d. Let $T \in R^{n \times n}$ be a flow matrix, where $T_{ij} \geq 0$ denotes how much of word i in D travels to word j in D'. The word mover's distance between two documents is given by the following equation,

$$WMD(D_i, D_j) = min_{T \geq 0} \sum_{i,j \in [1..n]} T_{ij}.c(i,j) \tag{1}$$

subject to

$$\sum_{i,j \in [1..n]} T_{ij} = d_i, \quad \forall i \in [1..n] \quad and \quad \sum_{i,j \in [1..n]} T_{ij} = d_j, \quad \forall j \in [1..n] \tag{2}$$

where $c(i, j)$ is defined as the Euclidean distance between x_i and x_j in embedding space.

3.2 Soft Cosine Similarity

Soft cosine [27] is a method for calculating the similarity between two feature vectors, in our case two documents, even when they have no features in common. For measuring the similarity between words, soft cosine can leverage Levenshtein distance, WordNet similarity, or cosine similarity in word embedding space. As described in [6], given two documents X and Y, soft cosine similarity can be defined as Eq. 3,

$$cos_M(X, Y) = \frac{X^t.M.Y}{\sqrt{X^t.M.X}.\sqrt{Y^t.M.Y}} \tag{3}$$

$$X^t.M.Y = \sum_{i=1}^{n}\sum_{j=1}^{n} x_i m_{i,j} y_j \tag{4}$$

where M is a similarity matrix between words. The similarity between words is defined as the cosine similarity between word embedding vectors of words.

4 LICD: Language-Independent Category Detector

We describe LICD in this section. The architecture of the LICD is depicted in Fig. 1. It contains the following three main components: SCM similarity calculation, WMD similarity calculation, and classification. In order to identify aspect categories belonging to a given sentence, first of all, we calculate two similarity scores per category: SCM similarity, which is the score between the given sentence and set of representative words, and WMD similarity, which is the score between the sentence and the category calculated using word mover's distance. This gives us $2 \times c$ similarity scores for each sentence, where c is the number of categories. These scores are provided to a one-vs-all logistic regression classifier as features. In the following subsections, we will discuss each of these components in detail.

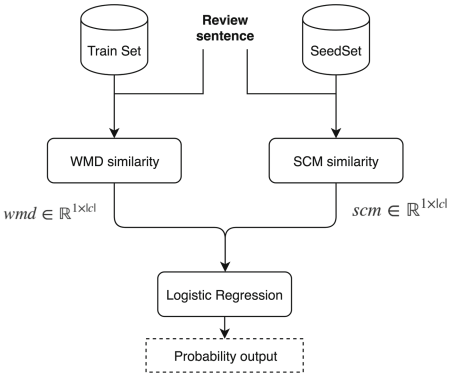


Fig. 1. The architecture of LICD.

4.1 SCM Similarity Calculation

Seed Selection. Every category has some specific words that can represent the category well. For example, the words 'food' and 'delicious' are good representative words for the 'FOOD#QUALITY' category. Each of these words has several semantically similar words (e.g 'cuisine' is a synonym for 'food'). A list of representative words plus their semantically similar words can represent a category well. To extract the set of representative words for different categories, we investigated different feature selection methods [8,30].

The best performance was achieved by the chi-square (χ^2) method. The selected words for each category constitute its seed words. After obtaining seed words, a list of top-k most similar words to each seed word is extracted from a word embedding model as semantically similar words. Table 1 illustrates an example of two extracted seed words and top-3 similar words for 'FOOD#QUALITY' and 'AMBIENCE#GENERAL' categories.

Table 1. A list of top 3 most similar words for 2 seed words of the categories 'FOOD#QUALITY' and 'AMBIENCE#GENERAL'.

Category	Seed word	Top 3 similar words
FOOD#QUALITY	Food	Cuisine - foods - grub
	Delicious	Delish - yummy - tasty
AMBIENCE#GENERAL	Atmosphere	Ambiance - ambience - environment
	Decor	Ambiance - decoration - decore

We compose a sentence using seed word followed by its semantically similar words. We can see this composed similar words as a sentence that describes a specific aspect of a category. The set of these sentences for each category constitute its SeedSet. For example, according to Table 1, the SeedSet of 'FOOD#QUALITY' category will contain the sentences 'food cuisine foods grub' and 'delicious delish yummy tasty'.

SCM Similarity. A sentence is likely to belong to a specific category if there is a high semantic similarity between sentence and the SeedSet of that category. Soft cosine similarity has an advantage compared to the more traditional method of cosine similarity on bag-of-words for computing the similarity between two sentences in that it also takes into account the semantic similarity between words. This advantage motivates us to utilize soft cosine similarity for obtaining similarity values between sentences and SeedSets of "the SeedSets" of each category. If a review contains words that are semantically related to one of the sentences in $SeedSet_i$, we expect the given review and $SeedSet_i$ to gain a high soft cosine similarity value.

For each sentence, we calculate a SCM similarity vector $\mathrm{scm} \in \mathbb{R}^{\mathrm{c}}$, where c is the number of categories. We define $\mathrm{scm_i(x)}$ as the soft cosine similarity score between sentence x and the i^{th} category, which is calculated by averaging soft cosine similarity values between x and each sentence in the i^{th} category's SeedSet. Since each sentence in a SeedSet covers one of the aspects of the category. So for review sentences that belong to a category, the average of the soft cosine similarity between these review sentences and the sentences belonging to the SeedSet of that category will yield larger values compared to sentences not belonging to that category. Furthermore, it gives relatively higher value to a review sentence that covers at least one of the category's aspects compared to

the sentences that do not cover any aspects of a category. $scm_i(x)$ is calculated using Eq. 5,

$$scm_i(x) = \frac{\sum_{s \in SeedSet_i} softcossim(x,\ s)}{|SeedSet_i|} \tag{5}$$

where SeedSet_i is the SeedSet of the ith category, $|\text{SeedSet}_\text{i}|$ is the number of sentences in SeedSet_i, and softcossim is the soft cosine similarity measure.

4.2 WMD Similarity Calculation

People tend to use similar sentence structures to express similar emotions [16]. For example, two sentences 'The food was good' and 'The cuisine was perfect' are two structurally similar sentences that express a similar opinion about the same category. Because part of the word mover's distance algorithm involves matching words, this measure can be quite efficient in measuring the similarity between a set of both structurally and semantically similar documents. Our motivation for utilizing this measure is based on the hypothesis that categories of a review sentence can be captured from it's structurally similar and semantically relevant sentences. Table 2 shows an example of the closest sentences to a given sentence retrieved by word mover's distance measure. As we see in this example, the word mover's distance measure tends to provide a minimum distance for the sentences that share a similar structure and semantically similar words.

Table 2. Example of the closest sentences to a given sentence retrieved by word mover's distance.

Given sentence: Ambiance is relaxed and stylish	
Closest sentences	Distance
The atmosphere is relaxed and casual	10.02
Atmosphere is nice and relaxed too	10.78
Zero ambiance to boot	16.38
Decor is charming	17.91
The decor is very simple but comfortable	18.00

In this part, analogous to SCM similarity, for a given sentence we calculate WMD similarity vector $\text{wmd} \in \mathbb{R}^\text{c}$, where c is the number of categories. We define $\text{wmd}_\text{i}(\text{x})$ as the word mover's distance similarity score between sentence x and the i^{th} category. For calculating WMD similarity vector wmd, first, we retrieve the top-k closest sentences to the given sentence, where the distance measure is the word mover's distance. Then, the similarity score between the given sentence and the i^{th} category will be the sum of similarity scores between the given sentence and all top-k closest sentences that have category i as their label, where similarity score is defined as the inverse of word mover's distance value.

We choose to sum the similarity scores to emphasize the majority categories between the top-k closest sentences and hash out the non-common categories.

Let $X = \{x_\ell, y_\ell\}_{\ell=1}^{n}$ be the train set, where x is a train sentence and y is the set of labels corresponding to x, and n is the number of train sentences. So $\mathrm{wmd_i(x)}$ can be calculated using Eq. 6,

$$wmd_i(x) = \sum_{x_k \in neighbor(x)} sim_i(x,\, x_k) \tag{6}$$

where $neighbor(x)$ is the set top-k closest neighbors to the sentence x and $\mathrm{sim_i(x,\, x_k)}$ is the similarity value between sentences x and $\mathrm{x_k}$ which is defined as follows,

$$sim_i(x,\, x_k) = \begin{cases} \frac{1}{1+wmdistance(x,\, x_k)} & \text{if } i \in y_k \\ 0 & \text{otherwise} \end{cases} \tag{7}$$

where $wmdistance(x,\, x_k)$ is word mover's distance between x and x_k sentences and $\mathrm{y_k}$ is the set of labels corresponding to sentence x_k.

One of the shortcomings of word mover's distance measure is its high time complexity. To speed up finding the closest neighbors using word mover's distance, we first find a large number of closest neighbors using cosine similarity between the average of the embedding vectors of words in the sentences to obtain a smaller and approximately relevant set of sentences. Then we find closest neighbors based on word mover's distance measure on the smaller set of obtained sentences.

4.3 Classification

For each category, a logistic regression classifier is trained using *scm* and *wmd* vectors obtained from the previous sections. The output of the classifiers is a probability distribution over the predefined categories. A category is assigned to a sentence if it's probability exceeds a threshold. A single optimum threshold for all the categories is found using a simple linear search.

5 Experiments

5.1 Datasets

For our experiments, we used the SemEval-2016 Task 5 datasets for English [21], French [2], Russian [15], Spanish, Dutch [7], and Turkish languages in restaurant domain. The number of training sentences is 2000, 1711, 1733, 3490, 2070, and 1104 for English, Dutch, French, Russian, Spanish, and Turkish, respectively, with 676, 575, 696, 1209, 881, and 144 test sentences. These statistics are borrowed from [21].

5.2 Experiment Setup

As the pre-processing step, tokenization and stop word removal is performed using the NLTK tool [3]. The scikit-learn package [20] is used to implement logistic regression classifier and chi-square feature selection method. For word mover's distance and soft cosine similarity and training word embeddings, we used the implementation provided by the gensim package [22]. CBOW method [17] is used for training word embeddings for the English language on the unlabeled Yelp restaurant dataset[2] with the dimension size of 300. For other languages, we used pre-trained word embeddings provided by [24]. All the parameters are optimized on the validation set. The best number of seed words for English, French, Spanish, Dutch, Turkish, and Russian are found as 1, 3, 5, 19, 15, and 25, respectively. The best number of nearest neighbors for WMD similarity for English, French, Spanish, Dutch, Turkish, and Russian languages are found as 9, 9, 15, 15, 10 and 10, respectively. For CONV and LSTM baselines we used multi-label soft margin as loss function. Both models are trained using a minibatch size of 128 and Adam optimizer [12] with a learning rate of 1×10^{-3}. Both models are trained for a maximum of 100 epochs for which early stopping is performed with patience set to 10. For the CONV baseline, the number of kernels, hidden layer size, and drop out rate is set to 300, 150, and 0.5, respectively. For the LSTM baseline, hidden size and drop out rate is set to 150 and 0.5, respectively.

5.3 Baseline Methods

Since one of the contributions of our method is being language-independent, for each language, we listed the current best model for that language. The compared methods are as follows:

- **NLANGP** [31]: This model utilizes the output of a CNN model and a set of features including word clusters, name lists, and head words to train a set of binary classifiers for each category.
- **XRCE** [4]: In this method the classification task is done in two steps. First, aspects with explicitly mentioned targets are classified using an ensemble method containing a Singular Value Decomposition and a One-vs-all Elastic Net Regression using a set of features. Then, aspects with implicit targets are classified using the same set of features but at the sentence level.
- **UFAL**: [29] This method uses a LSTM based Deep Neural Network. As features, this model only utilizes word embeddings. This is the only team that participated in all the languages in SemEval2016.
- **TGB**: [5] In this model, the classification task is done in two stages: At first, a set of one-vs-all logistic regression classifiers are trained for each attribute and entity. In the second stage, the output of the aforementioned classifiers along with other textual features (such as n-grams) are used as features for another set of one-vs-all logistic regression models for each category.

[2] https://www.yelp.com/dataset/challenge.

- **GTI**: [1] This work utilizes a set of one-vs-all SVM classifiers trained on textual features such as bi-grams, POS tags, lemmas, etc.
- **INSIGHT**: [25] This method utilizes a CNN with word embeddings as features. The word embeddings for languages other than English were initialized randomly.
- **UWB**: This method trains a Maximum Entropy Classifier using a large number of features including word embeddings, several kinds of bag-of-words features, POS tags, etc.
- **SemEval 2016 Baseline**: [21] For evaluation, SemEval 2016 provided a baseline. The baseline used an SVM with a linear kernel. For features, a set of 1000 most common unigrams was used.

The results of the baselines mentioned above are borrowed from [21]. Among several competitors in the SemEval-2016 workshop, the best results in English, French, Spanish, Russian, Dutch, and Turkish languages are achieved by NLANGP, XRCE, GTI, UFAL, TGB, and UFAL, respectively. Furthermore, we implement two methods as deep neural network-based baselines as follows:

- **CONV**: A CNN with one convolutional layer followed by a max-pooling layer and two fully-connected layers. We used tanh activation function for the convolutional layer and ReLU for the hidden layer. We used the sigmoid activation function on the output.
- **LSTM**: A bidirectional Long Short-Term Memory network with one layer followed by two fully-connected layers. We used ReLU after recurrent layer and sigmoid at the output layer.

5.4 Results and Analysis

For evaluation, we used micro F1-score of all the category labels, similar to the evaluation metric for SemEval-2016 ACD subtask. Table 3 shows the best results of the baseline methods and LICD in all languages. The best result for each language is marked in bold. LICD outperforms best systems in English, Russian, and Dutch languages. These results show the effectiveness of our approach in multilingual environments. In French, Spanish, and Turkish languages LICD achieves result competitive to the best-performed baselines. Our method is ranked 2^{nd} among baselines in French and Turkish, and is ranked 3^{rd} among baselines in Spanish. However, we would like to emphasize that the best systems in French and Turkish are deep neural network methods, which means their success depends on the existence of a sufficient volume of train data. However, as we will show in further experiments, LICD does not need a large volume of data, and with just about half of the training samples can achieve reasonable results. On the other hand, the best result for Spanish belongs to GTI method, which used several hand-crafted features such as n-gram, POS-tag, lemmas, and words. LICD is based on text matching and thus does not need such hand-crafted features.

5.5 Analysis of Different Components

Table 4 shows the comparison results between the different components of our method on the English dataset. When we trained the classifier with just *scm* vectors as features, the F1-score is 67.81, while F1-score obtained by *wmd* vectors as features is 70.60. Although *wmd* achieves better result compared to *scm*, the best result is obtained when we combine it with *scm* features. This shows the effectiveness of combining two different kinds of *scm* and *wmd* features. The interesting point is that results obtained by each of *scm* and *wmd* features alone outperforms most of the baseline methods.

Regarding computational complexity of our method, given n as the size of the vocabulary, the complexity of word mover's distance computation [14] is $O(n^3 log(n))$ and the complexity of computing soft-cosine similarity [18] is $O(n^3)$, respectively. In order to be able to utilize this method in the real-world applications, we need to reduce its computational complexity which we plan to address in the future works.

Table 3. The F1-score of LICD compared to baseline methods in each language.

Method	English	French	Spanish	Russian	Dutch	Turkish
CONV	71.78	60.90	67.53	68.95	57.88	61.79
LSTM	73.30	**63.27**	65.78	68.22	60.94	**64.80**
NLANGP	73.03	-	-	-	-	-
XRCE	68.70	61.20	-	-	-	-
UWB	68.20	-	61.96	-	-	-
INSIGHT	68.10	-	61.37	62.80	56	49.12
GTI	67.714	-	**70.58**	-	-	-
TGB	63.91	-	63.55	-	60.15	-
UFAL	59.3	49.92	58.81	64.82	53.87	61.02
SemEval	59.92	52.61	54.69	55.88	42.81	58.90
LICD	**73.54**	61.57	65.99	**69.76**	**61.30**	62.96

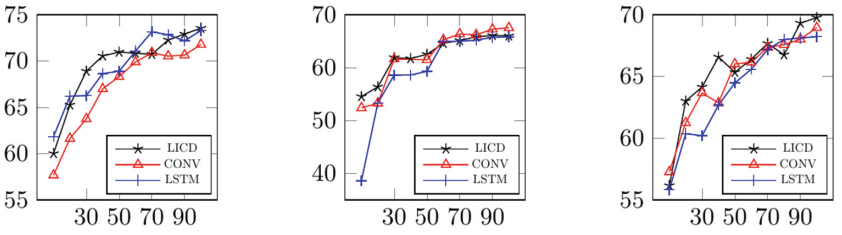
Table 4. Different component analysis results.

Components	F1(%)
scm	67.81
wmd	70.60
scm + wmd	73.54

5.6 Effect of Training Set Size and the Number of Seed Words

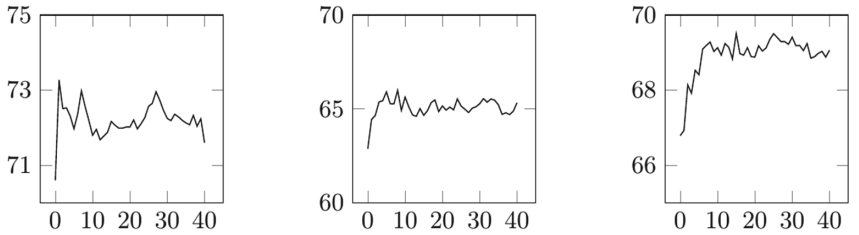
Fig. 2 presents the F1-score of the LICD method compared to CONV and LSTM baselines with different sizes of the training data. To perform this experiment, we split the original training set into ten equally divided portions, using stratified sampling method. Initially, we use only one part of the training set and gradually increase the number of utilized training set portions by one portion at each step. In all the experiments evaluation was performed on the full test set. Due to the limitation of space, we only show the result of the conducted experiment in three languages English, Russian, and Spanish. According to Fig. 2, in all of the three languages, our method achieves better or competitive results compared to the other baselines when there are few numbers of train samples. Figure 2(a) shows that in English, LICD achieves a stable F1-score approximately in the range of 40–70% (800–1300 sentences) of the training data volume, while the performance of other baselines increases steadily up to the end. Figure 2(b) shows the results in Spanish. We observe that deep learning-based baselines perform very poorly when the number of training samples is lower than 400 train sentences, especially the LSTM baseline. Similarly, we can see from Fig. 2(c) that in the Russian, LICD performs better than deep learning-based baselines when the training data volume is less than 50% (1745 sentences) of the original training data volume. These results indicate that LICD is not very sensitive to the volume of training data and can obtain reasonable performance compared to the results corresponding to the full-sized train data.

Fig. 3 shows the effect of the number of seed words in 3 languages. For some languages like English, with a few seeds, here with just one seed, we achieve the best result, but in others, we need more seeds to achieve the best performance of our system. This behavior may occur because of the characteristics of languages. In some languages, few words can represent a category, while in other languages we need more words to represent a category. Furthermore, we observe that increasing the number of seed words after a certain point yields no improvement in the performance of the system as the seed words start to contain general terms. These general terms are not discriminative, and therefore, do not influence the results.



(a) English (b) Spanish (c) Russian

Fig. 2. Effect of different sizes of training data. The vertical axis represents F1-Score, and the horizontal axis represents data volume (in % of total).



(a) English (b) Spanish (c) Russian

Fig. 3. Effect of the number of seed words. The vertical axis represents F1-Score, and the horizontal axis represents number of seed words.

6 Conclusions

We proposed LICD, a text matching based method for aspect category detection, which does not require feature engineering or any language-specific tools. We proposed to use soft cosine and word mover's distance to assess the similarity between a given sentence and the set of seed words and find structurally and semantically similar sentences to the given sentence respectively. Experimental results on multilingual datasets demonstrate that our method outperforms baselines in several languages, and achieves competitive results in others. For future works, we plan to lower the computational complexity of our model and investigate the suitability of other document distance measures for this task.

References

1. Alvarez-López, T., Juncal-Martinez, J., Fernández-Gavilanes, M., Costa-Montenegro, E., González-Castano, F.J.: GTI at SemEval-2016 task 5: SVM and CRF for aspect detection and unsupervised aspect-based sentiment analysis. In: Proceedings of the 10th International Workshop On Semantic Evaluation (SemEval-2016), pp. 306–311 (2016)
2. Apidianaki, M., Tannier, X., Richart, C.: Datasets for aspect-based sentiment analysis in French. In: LREC (2016)
3. Bird, S., Klein, E., Loper, E.: Natural Language Processing with Python: Analyzing Text with the Natural Language Toolkit. O'Reilly Media Inc., Sebastopol (2009)
4. Brun, C., Perez, J., Roux, C.: XRCE at SemEval-2016 task 5: feedbacked ensemble modeling on syntactico-semantic knowledge for aspect based sentiment analysis. In: Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016), pp. 277–281 (2016)
5. Çetin, F.S., Yıldırım, E., Özbey, C., Eryiğit, G.: TGB at SemEval-2016 task 5: multi-lingual constraint system for aspect based sentiment analysis. In: Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016), pp. 337–341 (2016)
6. Charlet, D., Damnati, G.: SimBow at SemEval-2017 task 3: soft-cosine semantic similarity between questions for community question answering. In: Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017), pp. 315–319 (2017)

7. De Clercq, O., Hoste, V.: Rude waiter but mouthwatering pastries! an exploratory study into Dutch aspect-based sentiment analysis. In: Tenth International Conference on Language Resources and Evaluation (LREC2016), pp. 2910–2917. ELRA (2016)
8. Forman, G.: An extensive empirical study of feature selection metrics for text classification. J. Mach. Learn. Res. **3**(Mar), 1289–1305 (2003)
9. Ganu, G., Elhadad, N., Marian, A.: Beyond the stars: improving rating predictions using review text content. In: WebDB, vol. 9, pp. 1–6. Citeseer (2009)
10. Hu, M., Liu, B.: Mining and summarizing customer reviews. In: Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 168–177. ACM (2004)
11. Khalil, T., El-Beltagy, S.R.: NileTMRG at SemEval-2016 task 5: deep convolutional neural networks for aspect category and sentiment extraction. In: Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016), pp. 271–276 (2016)
12. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014)
13. Kiritchenko, S., Zhu, X., Cherry, C., Mohammad, S.: NRC-Canada-2014: detecting aspects and sentiment in customer reviews. In: Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014), pp. 437–442 (2014)
14. Kusner, M., Sun, Y., Kolkin, N., Weinberger, K.: From word embeddings to document distances. In: International Conference on Machine Learning, pp. 957–966 (2015)
15. Loukachevitch, N., Blinov, P., Kotelnikov, E., Rubtsova, Y., Ivanov, V., Tutubalina, E.: SentiRuEval: testing object-oriented sentiment analysis systems in Russian. In: Proceedings of International Conference Dialog, vol. 2, pp. 3–13 (2015)
16. Ma, W., Suel, T.: Structural sentence similarity estimation for short texts. In: FLAIRS Conference, pp. 232–237 (2016)
17. Mikolov, T., Chen, K., Corrado, G., Dean, J.: Efficient estimation of word representations in vector space. arXiv preprint arXiv:1301.3781 (2013)
18. Novotnỳ, V.: Implementation notes for the soft cosine measure. In: Proceedings of the 27th ACM International Conference on Information and Knowledge Management, pp. 1639–1642. ACM (2018)
19. Pang, B., Lee, L., et al.: Opinion mining and sentiment analysis. Found. Trends® Inf. Ret. **2**(1–2), 1–135 (2008)
20. Pedregosa, F., et al.: Scikit-learn: machine learning in Python. J. Mach. Learn. Res. **12**, 2825–2830 (2011)
21. Pontiki, M., et al.: SemEval-2016 task 5: aspect based sentiment analysis. In: Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016), pp. 19–30 (2016)
22. Řehůřek, R., Sojka, P.: Software framework for topic modelling with large corpora. In: Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks, pp. 45–50. ELRA, Valletta, May 2010. http://is.muni.cz/publication/884893/en
23. Rubner, Y., Tomasi, C., Guibas, L.J.: A metric for distributions with applications to image databases. In: Sixth International Conference on Computer Vision 1998, pp. 59–66. IEEE (1998)
24. Ruder, S., Ghaffari, P., Breslin, J.G.: A hierarchical model of reviews for aspect-based sentiment analysis. arXiv preprint arXiv:1609.02745 (2016)

25. Ruder, S., Ghaffari, P., Breslin, J.G.: Insight-1 at SemEval-2016 task 5: deep learning for multilingual aspect-based sentiment analysis. arXiv preprint arXiv:1609.02748 (2016)
26. Schouten, K., Frasincar, F.: Survey on aspect-level sentiment analysis. IEEE Trans. Knowl. Data Eng. **1**, 1 (2016)
27. Sidorov, G., Gelbukh, A., Gómez-Adorno, H., Pinto, D.: Soft similarity and soft cosine measure: similarity of features in vector space model. Computación y Sistemas **18**(3), 491–504 (2014)
28. Su, Q., Xiang, K., Wang, H., Sun, B., Yu, S.: Using pointwise mutual information to identify implicit features in customer reviews. In: Matsumoto, Y., Sproat, R.W., Wong, K.-F., Zhang, M. (eds.) ICCPOL 2006. LNCS (LNAI), vol. 4285, pp. 22–30. Springer, Heidelberg (2006). https://doi.org/10.1007/11940098_3
29. Tamchyna, A., Veselovská, K.: UFAL at SemEval-2016 task 5: recurrent neural networks for sentence classification. In: Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016), pp. 367–371 (2016)
30. Tan, S., Zhang, J.: An empirical study of sentiment analysis for chinese documents. Expert Syst. Appl. **34**(4), 2622–2629 (2008)
31. Toh, Z., Su, J.: NLANGP at SemEval-2016 task 5: improving aspect based sentiment analysis using neural network features. In: Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016), pp. 282–288 (2016)
32. Xenos, D., Theodorakakos, P., Pavlopoulos, J., Malakasiotis, P., Androutsopoulos, I.: AUEB-ABSA at SemEval-2016 task 5: ensembles of classifiers and embeddings for aspect based sentiment analysis. In: Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016), pp. 312–317 (2016)
33. Xue, W., Zhou, W., Li, T., Wang, Q.: MTNA: a neural multi-task model for aspect category classification and aspect term extraction on restaurant reviews. In: Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 2: Short Papers), vol. 2, pp. 151–156 (2017)
34. Zhou, X., Wan, X., Xiao, J.: Representation learning for aspect category detection in online reviews. In: AAAI, pp. 417–424 (2015)