Sani Abdullahi, 2025, 13:2
ISSN (Online): 2348-4098
ISSN (Print): 2395-4752

International Journal of Science, Engineering and Technology

An Open Access Journal

Hybrid Approach for Multi Domain Aspect Extraction Using Specific Closed Sequential Pattern and Extended Random Set Technique

Sani Abdullahi
Department of Computer Studies
Hassan Usman Katsina Polytechnic, Katsina State, Nigeria

Abstract Our studies have been focusing on extracting better aspects that carry more weight from a textual dataset. One solution is to use data mining techniques, such as low minimum support for extracting sequential, relevant, specific and informative features .Such methods adopted the concept of extracting sequential patterns and assign high weight to top-k features based on the Extended Random Set (ERS) probability weight by pruning non-closed pattern from the representation by removing noisy features.

Keywords- Multi Domain, Low minimum support, Closed Sequential pattern, Extended Random Set, Aspect extraction

I. INTRODUCTION

In the past, companies and service providers receive the opinion of customers, through survey, word of mouth, observation and questionnaire. With the advent of internet information are available on the web, it is clear that search engines return numbers of documents; however, documents are not necessarily all relevant and beneficial to what the users need (Yan li et al 2002). Data mining is essential step in the process of the knowledge discovery to extract vital information that satisfy the users' needs. Aspect extraction can be reviewed as the process of extracting vital and essential information from the review of opinionated text from large datasets, information that is explicitly or implicitly presented in the data.

The proposed framework was designed to extract aspect from product reviews and produce aspect-based extraction in multi domains. To produce a representative aspect, some relevant, informative, specific and essential information must be extracted. The framework is divided into four major

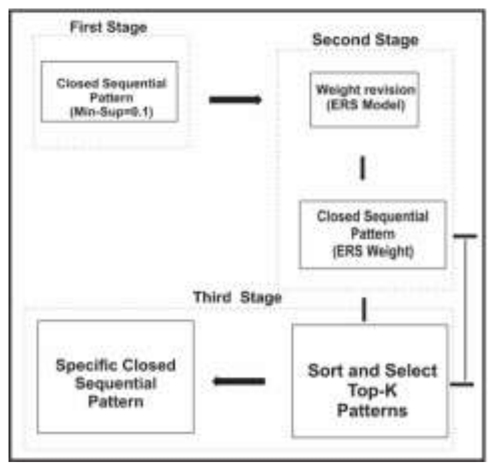

Figure 1: Multi domain knowledge approach for mining opinionated aspects using

tasks to use text files containing product reviews as input and then perform the four tasks to produce the final output aspect extraction. The first task is to mine closed sequential pattern or aspect using low minimum support to extraction specific, relevant and closed aspect and identify the associated opinion orientation of each aspect. The second task is mostly used to get those random set of aspects that are unique by extension.it is commonly used when there is need to be specific on some

© 2025 Sani Abdullahi. This is an Open Access article distributed under the terms of the Creative Commons Attribution License (http://creativecommons.org/licenses/by/4.0), which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly credited.

particular kind of data that are meant to get those random extended set which are unique, relevant and specific set and assign weight to them based on their significance. The third task is to apply ERS to calculate the weight for each aspect based on extended random set. The fourth task is selecting top-k feature aiming at giving important and relevant aspects

Specific closed sequential pattern and Extended Random Set for feature weighting

The proposed framework was designed to extract aspect from product reviews and produce aspect-based extraction in multi domains. To produce a representative aspect, some relevant, informative, specific and essential information must be extracted. The framework is divided into four major tasks to use text files containing product reviews as input and then perform the four tasks to produce the final output aspect extraction. The first task is to mine closed sequential pattern or aspect using low minimum support to extract specific, relevant and closed aspect and identify the associated opinion orientation of each aspect. The second task is mostly used to get those random set of aspects that are unique by extension.it is commonly used when there is need to be specific on some particular kind of data that are meant to get those random extended set which are unique, relevant and specific set and assign weight to them based on their significance. The third task is to apply ERS to calculate the weight for each aspect based on extended random set. The fourth task is selecting top-k feature aiming at giving important and relevant aspects.

Filter data frame to include specific aspects from low minimum support

The aspect we got is not the same as the aspects in the dataset for instance our dataset have 30,000 aspects when you apply low-minimum support based on the threshold you pass to it apriori and association rule mining. You might get your specific minimum support, let’s say 6000 from those 30000 aspect because we don’t want our model to learn noise only those specific aspect we filter in our data frame that has complete row only those 6000 specific unique aspects only rows that are selected by low minimum support not rows that has other aspect that are not selected by low-minimum support.

Those aspect that are selected by low minimum support is stored in a variable called frequent-itemsets. Lambda function was applied to convert frequent-itemset into a list; for every itemsets there is corresponding value which is their support. It will gives us every aspect as a unique value, now have aspect uniquely. We convert list to set for faster look-up it is easy for us to look at our aspect. Set give us a better and faster look-up and unique aspect.

Next, we go through our dataset, in this case if there is any aspect in the aspect column of our data that is the same with what we have in our low-minimum support variable frequent-item set keep it, if it is not the same eliminate it we only need those aspect that succeeded the low minimum support. Now, we loop through a dataset and also check through their aspects. Check if the aspect that is coming in if it is string have to convert it to list by splitting and pass it to variable called aspects. Finally, a comparison between aspects and all the aspect in your dataframe column in a list format. To ensure that all the aspect that are in aspects are the same with the important aspects any one that doesn’t with the important aspect will be filtered out, Finally, we only have the low minimum support aspects that made it with their sentence so that we can train our model with those important aspect based on low minimum support threshold and top-k features.

Extended random set application to quantify aspect significance

This is to properly select important aspect that model can learn and be able to identify when giving a review or sentiment. All these processing techniques from low-minimum support, extended random set and selecting top-k features is aim at one thing specific , important and relevant aspect.

Firstly, we have to split these aspects into individual row or make it separately, we have done low-minimum support on filtered-dataframe which is a pre-processed data that has come through low-minimum support. Now, we have to apply extended random set on the cleaned aspects to explode the dataframe to be able to split out other columns and get our clean aspect. That means to separate our columns and split it and store.

Next, calculate the frequency of each aspect when dealing with extended random set we need to know the frequency of each aspect. We need to have the total number of occurrence for normalization. Sum of all the aspect will be stored with their occurrences. A technique we use to normalize or standardize our data so that we are been having any deficit in value, we want aspect frequency to have a balance within a particular range.

Now, to calculate the weight for each aspect based on extended random set, the result of our low-minimum support we apply ERS on it because all these techniques we are doing form low-minimum support to ERS so that we can get our important, specific and relevant aspect. We can train the model with this important aspects and when it is trained and deploy between an applications models can be able to detect this is the most frequent and most important statement that has been mentioned in all of the aspect sentiment analysis so far in this social media.

Extended random set is a technique that mostly used to get those random set of aspects that are unique by extension mostly used when to be specific on some particular kind of data which is meant to get those random extended set which is unique, relevant and specific set.

We are using frequency as a proxy ERS technique have a proxy for which it will work that means a threshold or a factor for which we are calculating ERS own its own. We are calculating ERS on factor based on any of these factors calculating based on rare occurrence, based on frequency, based on accuracy measure etc.

In our case we are calculating ERS based on a proxy called frequency. We have to get the total frequency, we used value count to give us frequency, and to get the total sum of our aspect.
To calculate the extended random set is to use frequency as proxy for importance and normalize it.To get those random set of aspects further unique by extension to be specific to a particular kind of data you need. You can't go through a regular data extended random set will help you like any other pre-processing technique but it does its own just the way it was built in python in different manner unique from TF-IDF, Low minimum support. It meant to get those random unique set.

Extended random set must have a proxy for which it will work that means a threshold or a factor for which we are calculating ERS.

Now, we calculate the ERS and saved it in aspect-weights. To calculate the ERS based on proxy and the proxy is based on frequency. After that we have to convert our result of ERS to dataframe because our aspect is not in structured format we have to unboxed it.Next, apply ERS we have to unboxed our aspect from the structured format which is in a dataframe explode it into individual aspect split by comma because if you apply ERS it will give us an error, it will not work after you have done that apply ERS pre-processing technique on the data by building up from aspect frequency which we calculated for value count to calculate the frequency and total occurrence of the aspects, the result of these two techniques is what we use to calculate ERS.

Next, we have to sort out our values by weight in descending order that is sort the aspect and their ERS by weight in descending order so that the aspect that has higher ERS weight will be at the top follow by the ones with lowest weight.
Lastly, we assign these weights back to the original dataframe. When we create a dictionary for aspect

and weights that is the keys to values. For instance "great food", "awesome service" "the screen", "this player"

"the screen"	0.008241
"the waiter"	0.009332
"this player"	0.007878
"awesome service"	0.001333

Top-k features

We decided to select k=200 in order to have enough aspect for training the model because anything less than 200 we have few amount of dataset to work with. Anything less this you have less amount of aspect or noise ones that you don't want. Set our parameter k to 200 and Firstly, we sort our aspect in descending order based on their weight this top-k features is going to select randomly 200 top-k features.

Top-aspects is the variable that holds well sorted aspect based on descending order in form of list. There are some that are not really aspect like which, that, they, then, when, all, yourself, times, these, others, anything we have to get rid of them because they are not really aspect.

After removing unwanted aspects we have to check if there is any aspect in the list of top-k aspect in the row. We apply top-k aspects to our dataframe because we are not only training our model with aspects and sentence with their domains arrangement. Any time we have done pre-processing technique on aspect column we have to go back and apply filtering to the dataframe so that only rows that has the result of the pre-processing technique that will be maintained. We are going to pick the complete rows from where those aspects that are very important and made it through ERS, Top-k features those aspects with their rows we are going to pick and filter them and the ones that has aspects that didn't make it will be filtered out.

Smote

Now, we structured our data in such away we can be able to use SMOTE for balancing. We have a nearest k-neighbour to have a frequency of at least 2 of that aspect frequency combination. For aspect to be saved must occur at least two times if it doesn't occur it will not be saved. Next, we have to filter the rows to keep only aspect with frequent aspect combination. The frequency must be greater than or equal to the threshold that is meant to filter our dataframe it only include aspect combinations that occur 2 times or more than that.

Model training

In training our model we used Bi-LSTM which means bidirectional long short term memory. The idea behind recurrent neural network is the situation whereby model can be able to learn a context of a word by going back and forth to retrain the model. In bi-directional LSTM is following that process. The reason why we use Bi-LSTM is that we are dealing with sequential data that is sequence of sentences. We used Random search CV to be able to get the best parameters that will give us best output for different kind of parameter.

The hyper parameters used in training our model are LSTM_Units=64, dropout rate=0.3, batch size=32, number of epochs=25, and learning rate= 0.001. Dropout rate we stay in between 0.3 because we tried other rate and noticed that the model is underperforming our accuracy is 0.4 is overdroping it, batch size = 32 because of memory space anything more than that you will consume your memory space you can train your sequential data in 32 sizes. We used different epoch and only one we used in our model will be able to learn properly. Learning rate is the heart of our model performance, grad search CV will tell the exact learning that is suitable for your model that will allow your model to be able to learn properly.

Firstly, we try to reshape our training data to have three vector space or 3-dimension that is 0-dimension,1-dimension and inside the training data there is another 1-dimension if the length of this shape is not equal to three make it 3. The length of x-train has to be equal to three because we have done vector space embedding,sentence embedding BERT embedding automatically gives you 776 vector space which is one-dimension of your data due to embedding ,second dimension of your data is total amount of your data of the rows and columns.Now, we build the LSTM model to make an instance of that class. We start using model to create bi-directional LSTM our LSTM_Unit was set to 64,our drop_out rate is set to 0.3 which will help to reduce overfitting or model memorizing or overlearning.We started adding layers inside the model,the first one is bidirectional layer followed by normal LSTM layer.Now, we add more deeper layers we pass 128 unit or neurone dense layer,the cativation function is relu.Next, is the output layer we create layers inside the model to get the unique values from training data and activation function is softmax because we are dealing with multi class classification.Our optimizer is adam which will allows us to pass our learning rate(The rate at which our model learn either faster,slowly or steadily) we use loss function sparse categorical-crossentropy to compile our model which is meant to reduce the rate at which the model prediction is not been accurate.We use earlystopping in our model to measure the validation loss, that means the model started to underlearn, if the loss function start to increase instead of decrease you have to track it down or dropped and stop training and give the history of the training. Now, we have to check for validation loss and validation accuracy and prediction on validation data.

Experimental Datasets

In this section, we verify the effectiveness of our proposed methods on real world review data sets.

In this experiment, we used the popular datasets of the product reviews of six electronics products that were introduced by Bing Liu (Rana & cheah 2020). The products involved in this study are Nokia, Nikon, Apex, Creative, Canon, Laptop and Restaurant.

The evaluation process of this study only consider those review sentences that includes opinions about the product features across different domain. The details of the entire product datasets used in our experiment are shown in table 1 and 2.

Although our approach uses supervised approach this experiment is conducted to compare our approach, this experiment is conducted to compare our proposed work of aspect extraction supervised baseline system used in the six datasets to measure the performance of the system.

Table 1: presents six different domains showing the number of sentences and their respective aspects.

Dataset	#Review	#Sentences	#Aspects
Cell phone: Nokia	99	740	99
Nikon	33	346	96
Mp3: Apex	95	1716	57
DVD: Creative	41	546	67
Canon	45	597	79

Table 2: SemEval challenge datasets showing the number of sentences and their respective aspects.

Datasets	Domain	Train		Test	
		Sentence	Aspect	Sentence	Aspect
SemEval 2014-L	Laptop	3041	2358	800	654
SemEval 2014-R	Restaurant	3045	3693	800	1134

Evaluation methods

To evaluate the effectiveness of this study, several means will be used, specifically precision & recall measures. Based on precision & recall, different means will be used, specifically: mean average precision (MAP), F-Scores, the f1-score measure, the break-even point (b/p), and interpolated precision. This evaluation metrics are widely used in information retrieval research. In our measurement, we have a collection of reviews and every aspect is known to be either relevant or irrelevant to the review.

The precision p is the fraction of the retrieved documents that is relevant to the topic, where recall r is the fraction of relevant documents that have been retrieved.

Table 3: Contingency table

Human judgement		
	Yes	No
System judgement	Yes	TP
	FP	
	No	FN
	TN	

In the table above, precision and recall measures are calculated as follows (Yuefeng Li and Ning Zhong, 2006)

Experimental Setting

All the experiments reported in this research have been carried out on a PC with an Intel(R), Core (TM) i5, 8265U, CPU@1.60GHz and 8GB memory running a windows 11 operating system. The application of the proposed model was programmed using python programming language version as the development environment. The data collected from two different sources.

The (Sem Eval 2014-L and SemEval 2014-R) which comprises of two different domains and used in our experiments without any modifications. At the same time another dataset comprises of five domains would be used which consists of Apex,Canon, Laptop, Nokia, Creative, Nikon, restaurant was collected and processed. The information concerning relevance judgements for each topic in both training and test datasets was also derived from our pre-processed data.

The value of minimum support used for the experiments in different; for more consistency we test all the models using the same min_sup=0.1. such as in the PCM model.However, in the SCSP model we try to reduce the minimum support to extract more long patterns.For this reason we used min_sup= 0.1.Moreover, the loop to extract the features in the proposed algorithms should stop and exit when no more features are found except in some cases when the loop did not seem to stop, as some reviews in these sentences contain a large number of long features.

Evaluation procedures

In order to evaluate the proposed models, we applied them in aspect extraction task. As mentioned in previous section, the aspect extraction system aims to filter the incoming aspects and remove irrelevant aspects based on user needs. The aspect extraction system can be classified into three different types: adaptive filtering, batch filtering and routing system as described in (Y. Singer. B 2000).

This research uses adaptive filtering to avoid the use of thresholds and the system's performance is measured based on the ranked documents.

To evaluate the proposed models, the system used two datasets: SemEval 2014 and aspect extraction datasets. The general evaluation procedure starts by using each review in the datasets as feedback collection given by the user. Table 1 & 2 and show that both of the reviews dataset consist of two sets of datasets in the training and testing stages. All the reviews in these two datasets are used in the stage of aspect extraction and review evaluation. The following section will describe each stage of the general procedure for evaluation of our methods.

Testing and Evaluation

To evaluate the effectiveness of this study, several means will be used, specifically precision & recall measures. Based on precision & recall, different means will be used, specifically: mean average precision (MAP), the f1-score measure, the break-even point (b/p), and interpolated precision. This evaluation metrics are widely used in aspect extraction research. In our measurement, we have a collection of datasets and every aspect is known to be either relevant or irrelevant to the extraction.

The precision p is the fraction of the retrieved aspects that is relevant to the topic, where recall r is the fraction of relevant aspects that have been retrieved.

Extended Random Set (ERS) Model Evaluation

The aim of the section is to calculate the probability (weight) of the features in documents. The ERS model weighting is based on giving a review or sentiment is of high significance. This model has been applied and tested using low minimum support (Apriori) and Association rule mining to find the relationship between features for importance and sort & select top-k features and assign high weight to them. This section presents the results and discussion of the specific aspects from the top-k features with the use of the ERS model to calculate the aspect weights.

ERS Evaluation procedure

Proper selection of important aspects which the model can learn with the aim of giving a review or sentiment is of high significance. Extended random set is a technique that is mostly used to get those random set of aspects that are unique by extension. It is commonly used when there is need to be specific on some particular kind of data that are meant to get those random extended set which are unique, relevant and specific set.

In this research, the ERS is calculated based on a proxy called frequency. It is required to get the total frequency which value count was used to get the frequency and to get the total sum of the aspect.

In this experiment, we focused on closed sequential patterns only and implemented this method on product review dataset for aspect extraction in the training dataset. We compare our results using the ERS model with four different type of feature selection methods: RFD, TF-IDF, ROCCHIO and ERS using (min-sup=0.01). The results show that using the ERS model to weight the features based on its importance and improved the performance results for the extracted patterns significantly.

As mentioned earlier, low min-sup would generate a large number of aspects and most of them would be noise aspects. Thus, in this experiment we tried to extract the specific extracted pattern by sorting and selecting top-k aspects. The performance of the extracted aspects' changes is based on the proportion of the selected top-k aspects. From our observations, we found that on product review dataset, using the top 200 of the extracted aspects improved performance of our results.

Evaluation of finding specific patterns (Sort and select top-k features)

To find the relevant specific top-k features with the use of low minimum support to weight the aspects is an issue in text mining. As mentioned earlier using low minimum support to extract aspects is an issue because of the aspects frequency and noise, especially when we have a long document. This section illustrates the results and discussion the method of finding top-k features with the use of an ERS model to weight the extracted aspects.

Procedure for finding specific aspects (top-k features)

The steps required for the whole evaluation procedure are as follows:

A selection of k=200 was made in order to have enough aspect for the training of the model. The reason of the selection to this amount (k=200) is because anything less than 200 there is tendency to achieve less amount of aspect, or ones with noise. The parameter is set to k=200 and the aspect is sorted in descending order based on their weight and 200 top-k features are selected randomly.

Top-aspects is the variable that holds well-sorted aspect based on descending order in form of list. Some are not really aspect like; which, that, they, then, when, all, yourself, times, these, etc. As such they need to be rid of since they are not really aspect.

After removing unwanted aspects, a check is performed to find if there is any aspect in the list of top-k aspect looking at the row. The top-k aspects are applied to the dataframe since the training the model comprises of both aspects and sentence with their domains arrangement. Any time pre-processing technique is performed on aspect column, filtering to the dataframe is performed so that only rows that has the result of the pre-processing technique will be maintained. Complete rows are picked form those aspects that are very important and made it through ERS. Top-k features aspects with their rows are also picked and filtered, thereby removing those aspects that didn’t make it.

Specific patterns

Relevance feature discovery (RFD) model (Yuefeng Li et al..,2010) has been that long patterns are more important and specific than short patterns we used low minimum support to extract long patterns from text reviews. To provide good performance using min-support=0.1 extracts the best closed sequential patterns. In our study we tried to find aspects that are more specific in the extracted closed sequential pattern.

Roc Curve

In deep learning and machine learning, AUC or ‘Area under the ROC curve’ is the widely used measurement metric for prediction of the best classes in the model. ROC curve (Receiver operating characteristic curve) is a graphical plot showing the performance of a classification model at all thresholds by considering the parameters True positives (TF) on x-axis and True negative on y-axis.

Figure 8: Multi class ROC-AUC curve

The AUC values are lies in between 0.5 and 1.0 for binary classification, the higher accuracy indicates the model performs best whereas the classifier with AUC value 0.5 is consider as a bad performer. Figure 8 represents multi class ROC-AUC curve.

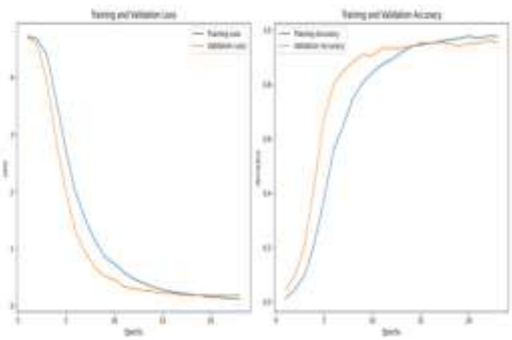

Figure 9: predicted training and Validation loss vs Training and validation accuracy

II. RESULTS

The top-k features results can be shown in two stages. In the first stage the high level closed sequential patterns are extracted and the ERS model applied to calculate the probability of the extracted aspects. In the second stage the closed sequential pattern were extracted with low minimum support to gather specific aspects from the top-k aspects. The effectiveness of our model is measured by the four different means listed in the previous section. The larger a measure score the better the system performs.

Table 1. presents a comparison of the results of extracting all closed sequential patterns and applying the ERS model in the first stage. Here, % change means percentage from using the ERS model (LMS+ERS+TOP-K) compared with the best result in the presented feature selection models in aspect extraction. The most important findings revealed in table 1. are that low minimum support with ERS and top-k features perform better than RFD with ERS and other feature selection methods for the important measures (MAP and F1) and that using the ERS model on the top-k features largely outperforms the RFD with ERS pattern as shown in table 1. below, using the ERS model with low minimum support and top-k features achieves

excellent performance with (107% maximum) in F1 and 80% minimum in precision (IAP) for all the six aspects extraction datasets we have used.

Table 1. Comparison of All feature selection-based methods with Extended Random Set

METHODS	B/P	MAP	F_β	IAP
LMS+ERS+TOP-K	0.964	0.982	0.954	0.9116
RFD+ERS	0.467	0.484	0.460	0.506
RFD	0.444	0.462	0.451	0.482
TF-IDF	0.396	0.398	0.412	0.423
ROCCHIO	0.340	0.406	0.420	0.434
% Change	106%	103%	107%	80%

Discussion

Low minimum support using min-support of 0.01 by applying apriori and Association rule mining of minimum threshold of 0.01 to find the relationship between aspects and extract those that are closed sequential pattern. Apriori is used to identify frequent item sets form the datasets. However, they are still limited by noise, irrelevant and inconsistencies and absence of some long and specific patterns due to common data mining process for extracting these patterns.

However, using low minimum support to extract closed sequential patterns are not very efficient in answering users' needs. (Yuefeng Li et al...,2010). Using ERS model gave us a new method for weighting extracted top-k features, such as patterns and helps to find the specific closed sequential patterns (SCSP).

Thus, the process of selecting and sorting top-k features consists of two main steps: Top-k specific aspects extraction with SCSP using min-support=0.1 , and then using the ERS model to calculate the aspect's probability (weight) and select top-k specific features based on the new weight.

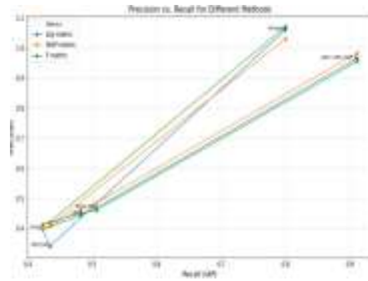

Figure 8: Average of closed sequential pattern (LMS), ERS and Top-k features

The evaluation process of this study only consider those review sentences that includes opinions about the product features across different domains. The details of the entire product datasets used in our experiment are shown in table 1 and 2.

III. CONCLUSION

We cite five reasons for selecting these studies to evaluate the effectiveness of our approach.

Simplicity Vs. Complexity

Their approach is more complex because they use a combination of syntactic, semantic, lexicon, and cluster features, along with a CRF model for sequential labelling. This can be effective in cases where context and relationships between words are essential for identifying aspect terms.

Our approach is more straightforward: by relying on TF-IDF and frequency pruning, we avoid complex feature engineering, which makes our system simpler and faster to implement.

Scalability

Our approach using TF-IDF and frequency pruning is likely more scalable because it's less computationally expensive. We are not performing complex feature extraction or training sophisticated models like CRF.TF-IDF can be calculated efficiently even for large datasets, and frequency pruning helps reduce noise quickly.

Their approach may be slower and more resource-intensive, particularly when dealing with large datasets, since CRF models and feature engineering can be computationally heavy.

Handling implicit aspects

Their approach might be better at handling implicit aspects or more complex sentence structures because CRF models can learn to understand the context in which words appear (e.g., recognizing that "it's slow" refers to "service" in "The service is slow" or "it's slow") Our approach may be more limited when it comes to implicit aspects.TF-IDF is more suited for explicitly mentioned terms(e.g., "camera", "battery life").If an aspect is not directly mentioned, our approach might miss it.

Focus on important terms

Our advantage lies in the frequency pruning: by removing terms that appear less frequently(using the threshold of 2), we ensure that our system focuses on important and frequent aspect terms, reducing noise this can be very helpful in cases where there's a lot of irrelevant or noisy data.

Their approach may retain more of the rare terms or noise since they don't explicitly prune based on frequency. However, their use of various features helps reduce some of this noise as well.

REFERENCE

1. Agrawal, R. and R. Srikant (1994). Fast algorithms for mining association rules. Proc. 20th int. conf. very large data bases, VLDB.
2. Bancken, W., et al. (2014). Automatically detecting and rating product aspects from textual customer reviews. Proceedings of the 1st international workshop on interactions between data mining and natural language processing at ECML/PKDD.
3. Ben-David, S. and R. Urner (2012). On the hardness of domain adaptation and the utility of unlabeled target samples. International Conference on Algorithmic Learning Theory, Springer.
4. Blitzer, J., et al. (2008). Learning bounds for domain adaptation. Advances in neural information processing systems.
5. Blitzer, J., et al. (2007). Biographies, bollywood, boom-boxes and blenders: Domain adaptation for sentiment classification. Proceedings of the 45th annual meeting of the association of computational linguistics.
6. Blitzer, J., et al. (2006). Domain adaptation with structural correspondence learning. Proceedings of the 2006 conference on empirical methods in natural language processing.
7. Bollegala, D., et al. (2015). "Cross-domain sentiment classification using sentiment sensitive embeddings." IEEE Transactions on Knowledge and Data Engineering 28(2): 398-410.
8. Bollegala, D., et al. (2012). "Cross-domain sentiment classification using a sentiment sensitive thesaurus." IEEE Transactions on Knowledge and Data Engineering 25(8): 1719-1731.
9. Bollegala, D., et al. (2014). Learning to predict distributions of words across domains. Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers).
10. Cambria, E., et al. (2012). "Sentic computing for social media marketing." Multimedia tools and applications 59(2): 557-577.
11. Chauhan, G. S., et al. (2020). An Unsupervised Multiple Word-Embedding Method with Attention Model for Cross Domain Aspect Term Extraction. 2020 3rd International Conference on Emerging Technologies in Computer Engineering: Machine Learning and Internet of Things (ICETCE), IEEE.
12. Chen, L.-C., et al. (2019). "Exploration of social media for sentiment analysis using deep learning." Soft Computing: 1-11.
13. Daumé III, H. (2009). "Frustratingly easy domain adaptation." arXiv preprint arXiv:0907.1815.
14. Daumé III, H., et al. (2010). Frustratingly easy semi-supervised domain adaptation.

Proceedings of the 2010 Workshop on Domain Adaptation for Natural Language Processing.
15. Esuli, A. and F. Sebastiani (2006). Sentiwordnet: A publicly available lexical resource for opinion mining. LREC, Citeseer.
16. Federici, M. and M. Dragoni (2016). Towards Unsupervised Approaches For Aspects Extraction. EMSA-RMed@ ESWC.
17. Franco-Salvador, M., et al. (2015). "Cross-domain polarity classification using a knowledge-enhanced meta-classifier." Knowledge-Based Systems 86: 46-56.
18. Gao, S. and H. Li (2011). A cross-domain adaptation method for sentiment classification using probabilistic latent analysis. Proceedings of the 20th ACM international conference on Information and knowledge management.
19. Glorot, X., et al. (2011). Domain adaptation for large-scale sentiment classification: A deep learning approach. ICML.
20. Gräßer, F., et al. (2018). Aspect-based sentiment analysis of drug reviews applying cross-domain and cross-data learning. Proceedings of the 2018 International Conference on Digital Health.
21. Gutiérrez, L. and B. Keith (2018). A Systematic Literature Review on Word Embeddings. International Conference on Software Process Improvement, Springer.
22. Htay, S. S. and K. T. Lynn (2013). "Extracting product features and opinion words using pattern knowledge in customer reviews." The Scientific World Journal 2013.
23. Hu, M. and B. Liu (2004). Mining and summarizing customer reviews. Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining.
24. Hu, M. and B. Liu (2004). Mining opinion features in customer reviews. AAAI.
25. Ionescu, R. T. and A. M. Butnaru (2018). Transductive learning with string kernels for cross-domain text classification. International Conference on Neural Information Processing, Springer.
26. Jiang, Z., et al. (2019). Cross-domain Aspect Category Transfer and Detection via Traceable Heterogeneous Graph Representation Learning. Proceedings of the 28th ACM International Conference on Information and Knowledge Management.
27. Jihan, N., et al. (2017). Multi-domain aspect extraction using support vector machines. Proceedings of the 29th Conference on Computational Linguistics and Speech Processing (ROCLING 2017).
28. Jing, B., et al. (2018). Cross-domain labeled lda for cross-domain text classification. 2018 IEEE International Conference on Data Mining (ICDM), IEEE.
29. Lakkaraju, H., et al. (2014). Aspect specific sentiment analysis using hierarchical deep learning. NIPS Workshop on deep learning and representation learning.
30. LeCun, Y., et al. (2015). "Deep learning." nature 521(7553): 436-444.
31. Li, F., et al. (2012). Cross-domain co-extraction of sentiment and topic lexicons. Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers).
32. Li, S., et al. (2013). Active learning for cross-domain sentiment classification. Twenty-Third International Joint Conference on Artificial Intelligence.
33. Lin, C., et al. (2011). "Weakly supervised joint sentiment-topic detection from text." IEEE Transactions on Knowledge and Data Engineering 24(6): 1134-1145.
34. Liu, B. (2012). "Sentiment analysis and opinion mining." Synthesis lectures on human language technologies 5(1): 1-167.
35. Liu, B., et al. (1998). Integrating classification and association rule mining. KDD.
36. Liu, B., et al. (2005). Opinion observer: analyzing and comparing opinions on the web. Proceedings of the 14th international conference on World Wide Web.
37. Liu, B. and L. Zhang (2012). A survey of opinion mining and sentiment analysis. Mining text data, Springer: 415-463.

38. Liu, K. and J. Zhao (2009). Cross-domain sentiment classification using a two-stage method. Proceedings of the 18th ACM conference on Information and knowledge management.
39. López, D. and L. Arco (2019). Multi-domain Aspect Extraction Based on Deep and Lifelong Learning. Iberoamerican Congress on Pattern Recognition, Springer.
40. Ma, B., et al. (2013). "An LDA and synonym lexicon based approach to product feature extraction from online consumer product reviews." Journal of Electronic Commerce Research 14(4): 304.
41. Marcacini, R. M., et al. (2018). "Cross-domain aspect extraction for sentiment analysis: A transductive learning approach." Decision Support Systems 114: 70-80.
42. Meng, X. and H. Wang (2009). Mining user reviews: from specification to summarization. Proceedings of the ACL-IJCNLP 2009 conference short papers.
43. Mikolov, T., et al. (2013). "Efficient estimation of word representations in vector space." arXiv preprint arXiv:1301.3781.
44. Moghaddam, S. and M. Ester (2010). Opinion digger: an unsupervised opinion miner from unstructured product reviews. Proceedings of the 19th ACM international conference on Information and knowledge management.
45. Nakatsuji, M., et al. (2017). "Semantic social network analysis by cross-domain tensor factorization." IEEE Transactions on Computational Social Systems 4(4): 207-217.
46. Pan, S. J., et al. (2010). Cross-domain sentiment classification via spectral feature alignment. Proceedings of the 19th international conference on World wide web.
47. Peng, M., et al. (2018). Cross-domain sentiment classification with target domain specific information. Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers).
48. Pereira, L. A. and R. da Silva Torres (2018). "Semi-supervised transfer subspace for domain adaptation." Pattern Recognition 75: 235-249.
49. Poria, S., et al. (2016). "Aspect extraction for opinion mining with a deep convolutional neural network." Knowledge-Based Systems 108: 42-49.
50. Rana, T. A. and Y.-N. Cheah (2015). Hybrid rule-based approach for aspect extraction and categorization from customer reviews. 2015 9th International Conference on IT in Asia (CITA), IEEE.
51. Rana, T. A. and Y.-N. Cheah (2017). Improving aspect extraction using aspect frequency and semantic similarity-based approach for aspect-based sentiment analysis. International conference on computing and information technology, Springer.
52. Remus, R. (2012). Domain adaptation using domain similarity-and domain complexity-based instance selection for cross-domain sentiment analysis. 2012 IEEE 12th international conference on data mining workshops, IEEE.
53. Rietzler, A., et al. (2019). "Adapt or get left behind: Domain adaptation through bert language model finetuning for aspect-target sentiment classification." arXiv preprint arXiv:1908.11860.
54. Samha, A. K., et al. (2014). "Aspect-based opinion extraction from customer reviews." arXiv preprint arXiv:1404.1982.
55. Sharma, R., et al. (2018). Identifying transferable information across domains for cross-domain sentiment classification. Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers).
56. Strapparava, C. and A. Valitutti (2004). Wordnet affect: an affective extension of wordnet. Lrec, Citeseer.
57. Tseng, H.-Y., et al. (2020). "Cross-domain few-shot classification via learned feature-wise transformation." arXiv preprint arXiv:2001.08735.
58. Wang, W. and S. J. Pan (2018). Recursive neural structural correspondence network for cross-

domain aspect and opinion co-extraction. Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers).

59. Wei, W. and J. A. Gulla (2010). Sentiment learning on product reviews via sentiment ontology tree. Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics.
60. Xu, H., et al. (2018). "Double embeddings and cnn-based sequence labeling for aspect extraction." arXiv preprint arXiv:1805.04601.
61. Yan, Z., et al. (2015). "EXPRS: An extended pagerank method for product feature extraction from online consumer reviews." Information & Management 52(7): 850-858.
62. Yang, M., et al. (2018). Cross-domain aspect/sentiment-aware abstractive review summarization. Proceedings of the 27th ACM International Conference on Information and Knowledge Management.
63. Yang, W., et al. (2019). "A simple regularization-based algorithm for learning cross-domain word embeddings." arXiv preprint arXiv:1902.00184.
64. Zhang, K., et al. (2019). Interactive attention transfer network for cross-domain sentiment classification. Proceedings of the AAAI Conference on Artificial Intelligence.
65. Zhang, L. and B. Liu (2014). Aspect and entity extraction for opinion mining. Data mining and knowledge discovery for big data, Springer: 1-40.
66. Zhang, Y., et al. (2017). "Aspect-augmented adversarial networks for domain adaptation." Transactions of the Association for Computational Linguistics 5: 515-528.
67. Zhu, P., et al. (2019). "Aspect Aware Learning for Aspect Category Sentiment Analysis." ACM Transactions on Knowledge Discovery from Data (TKDD) 13(6): 1-21.