

Suggestion Miner at SemEval-2019 Task 9: Suggestion Detection in Online Forum using Word Graph

Usman Ahmed¹, Humera Liaquat², Luqman Ahmed³ and Syed Jawad Hussain¹

¹Department of Computer Science and IT, The University of Lahore, Islamabad, Campus

²Department of Computer Science, COMSAT, Islamabad, Pakistan

³Department of Computer Science, UET, Taxila, Pakistan

usman.ahmed@cs.uol.edu.pk

humaira.699@gmail.com

aluqman734@gmail.com

jawad.hussain@cs.uol.edu.pk

Abstract

This paper describes the suggestion miner system that participates in SemEval 2019 Task 9 - SubTask A - Suggestion Mining from Online Reviews and Forums. The system participated in the subtasks A. This paper discusses the results of our system in the development, evaluation and post evaluation. Each class in the dataset is represented as directed unweighted graphs. Then, the comparison is carried out with each class graph which results in a vector. This vector is used as features by a machine learning algorithm. The model is evaluated on, hold on strategy. The organizers randomly split (8500 instances) training set (provided to the participant in training their system) and testing set (833 instances). The test set is reserved to evaluate the performance of participants systems. During the evaluation, our system ranked 31 in the Coda Lab result of the subtask A (binary class problem). The binary class system achieves evaluation value 0.35.

1 Introduction

Suggestion mining or identification of suggestions within the text is a relatively new area which is gaining popularity among many private and public sector organizations, service providers and consumers/customers at large due to its number of uses. Suggestion mining, in general, refers to the extraction of tips, advise or recommendations from an unstructured text, which can lead to a number of use cases (Negi et al., 2018). Currently, suggestion mining is considered to be an intrinsic part of any decision-making process, used by different entities to get an insight into people's perspectives and improve their products or services (Negi et al., 2018), (Jijkoun et al., 2010). To get reviews and suggestions, organizations either ask them from users explicitly or extract them from online reviews, blogs, discussion forums, social

media etc. This is because, these platforms are progressively gaining popularity (due to their expeditious advancements and ease of use) for obtaining public opinions towards events, brands, products, services, entities etc. Consider some examples which have been seen among many online reviews, giving useful suggestions to whom it may concern: "I would recommend doing the upgrade to be sure you have the best chance at trouble-free operation.", "Be sure to specify a room at the back of the hotel" (Negi et al., 2018), "Make sure you bring plenty of sun tan lotion-very expensive in the gift shop" (Negi and Buitelaar, 2015). We can clearly see that these reviews contain suggestions and recommendations for others to make use of a service or a product if they want to avail it at its best. Before the advent and realization of the importance and use of suggestion mining, opinion mining has been used by stakeholders to mine text with a perspective of summarizing opinions on the basis of sentiment polarities (Liu, 2012). Though identifying negative and positive sentiment distribution within a text is important from a lot of perspectives, however, identification of a suggestion oriented text would be more useful for stakeholders looking for improvements in their services or products, and also, for the services seeker and potential buyers of a product (Negi and Buitelaar, 2015). Thus, automatic identification and classification of suggestion oriented text from a large corpus of raw text is the need of the hour, as, is not feasible manually.

Some empirical analysis has been done previously for automatic suggestion mining, as, (Negi and Buitelaar, 2015) used Support Vector Machine (SVM) to identify suggestion oriented sentences within customer reviews. In another study ((Negi et al., 2016)) the authors used Neural Network architecture to classify suggestions from raw text. (Negi and Buitelaar, 2017) used Long Short Term

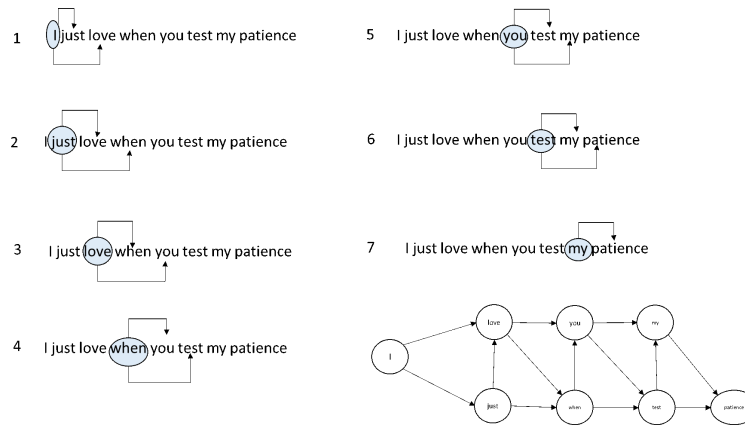


Figure 1: Graph Construction with vicinity size 2 illustrates how the vicinity size move toward the end of the tweet; in this example, the frame is the two following words and for each word some edges and nodes are added to the graph.

Memory (LSTM) model to classify sentences as suggestions or non-suggestions. The authors, in their study trained word embeddings on suggestions (taken from WikiHow "Tips" section) and the resulting LSTM model, showed higher performance from the previous ones. The graph-based centrality measure is also used to classify short text analysis (Ishtiaq et al., 2019). Lubna et al. proposed word sense disambiguation technique to evaluate the adverb, adjectives, and verb combination (Zafar et al., 2017).

By analysing the current suggestion mining techniques and studies ((Negi et al., 2018), (Negi and Buitelaar, 2017)), it is realized that suggestion classification task face many overlapping challenges as with other sentence classification problems. They include: annotations of data, comprehending sentence-level semantics, making sense of figurative and sarcastic expressions, long and complex sentences (covering multiple aspects and diverse domains), catering diverse domain sentences (rather than classifying domain specific ones), imbalanced class distributions (due to the imbalanced availability of suggestion oriented text within the raw text in certain domains) etc (Negi et al., 2018).

This paper describes our proposed model for the SemEval-2019 pilot challenge for suggestion mining's Sub-Task A, i.e. classify a given text as a suggestion or non-suggestion text, using same domain training and testing data. As described in the challenge highlights, the data set will belong to a particular domain i.e. suggestion forum for windows platform developers, which needs to be classified as a suggestion or non-suggestion class.

For this challenge, our proposed model is a hybrid one, inspired by two previous studies ((Gianakopoulos et al., 2008) and (Maas et al., 2011)), in combination with some additional features and word graph similarity score as used by Usman et al (Ahmed et al., 2018). The word graph model used in this research is adopted from Usman et al technique of Iron detection (Ahmed et al., 2018). The detailed description of our model is given under the proposed model section. Rest of the paper includes task overview, data set description, results and evaluation, followed by discussion and conclusion.

2 Task Overview

In SemEval-2019, task 9 contains two sub-tasks A and B, for suggestion mining from a given text. The text (dataset) for this challenge which needs to be classified against each subtask is taken from two domains, i.e. suggestion forums and hotel reviews (Negi et al., 2019).

In this paper, we are focusing on sub-task A, which is, detection of a suggestion or non-suggestion text, from, the text of suggestion forums (dedicated resources used to provide suggestions for improvements in any under-discussed entity). For this task, the training and validation data sets will belong to one domain, and the details of it are covered in the dataset overview section.

3 Proposed Model

Our proposed model is a hybrid approach, in which the given text is represented as a directed unweighted word graph at first (for both the

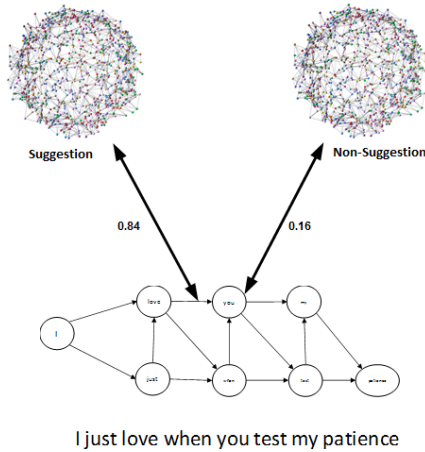


Figure 2: Graph Similarity Feature Extraction for one measure. The graph of a forum review used to compare with training data class graphs, in order to produce two numbers (depending upon the numbers of classes). These numbers will be used as a feature vector. The feature vector is provided to the trained model to predict the class of the new forum review.

classes). The edge between each word is created based on the vicinity window size, as explained in the subsection Graph construction. After graph construction, a comparison is carried out between each graph for construction of a feature vector, which is later used as an input for our machine learning algorithm. The graph is constructed based on a class assignment, which is later used to measure the similarity of a text with each class graph (suggestion or non-suggestion). For similarity measurement between the class graph and text graph, we used containment similarity, maximum common subgraph similarity and its variant compare graph in terms of similarity.

3.1 Dataset Overview

For the training, trial and evaluation dataset are provided by the organizer via Github: <https://github.com/Semeval2019Task9/Subtask-A>. The dataset for this task is annotated in two phases (Negi et al., 2019). In the first phase, crowd-sourced annotators are involved for performing the annotations, whereas in the second phase in-house expert annotators are used (Negi et al., 2019). The finalized datasets include only those sentences which explicitly express suggestions rather than those that only provide information which can be used to infer suggestions. The dataset is collected from a particular suggestion forum’s reviews (uservoice.com) on universal

No.	Features
1	Containment Similarity UniGram
2	Containment Similarity BiGram
3	Maximum Common Subgraph Node Similarity (MCSNS)
4	Maximum Common Subgraph Edge Similarity (MCSES)
5	Maximum Common Subgraph Directed Edge Similarity (MCSDS)
6	Number of Characters
7	Number of Words

Class	Train set	Trial Test set	Test set
Suggestions	2085	296	87
Non-suggestions	6415	296	746

windows platform (Negi et al., 2019). The number of sentences in the dataset is shown in Table 2.

3.2 Graph Construction

For the graph construction, we consider the given text (which needs to be classified as either suggestion or non-suggestions) as a set of words, based on their vicinity. Each word is considered as a labelled node in a graph, which is joined with a directed edge, depending on the window size. For our analysis, we used a vicinity size of 2, adopted by analysing our results during the tuning phase of our model. Figure 1 illustrates the complete graph construction process of a sentence, with vicinity size 2. We can clearly see how nodes and edges are added in the graph against each word. Further, in order to check text graph similarity with the class graph, our model makes use of the containment similarity (non-normalized value), maximum common subgraph similarity and its variant compare graph.

4 Feature Engineering

4.1 Containment Similarity

Containment similarity measure is used to measure two graphs similarity by calculating the common edges between them by the number of edges of the smaller graph (Aisopos et al., 2012). Equation 1 illustrates the mathematical expression of containment similarity measure, where GT (target graph) is the text word graph, G_s (source graph) is the word graph of suggestion class. e is the edge of a word graph and the size of the graph is the

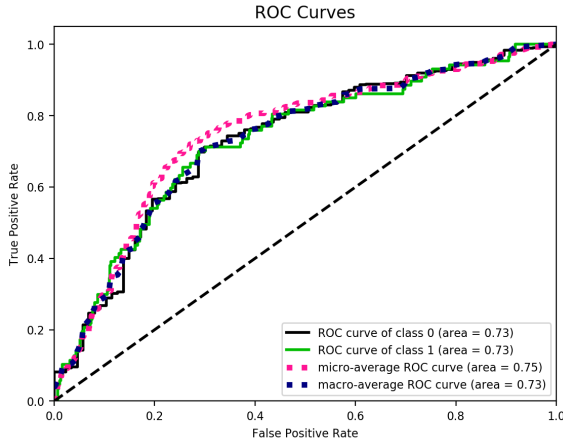


Figure 3: Precision Recall Curve of Binary Class problem

number of nodes or edges.

$$CS(G_T, G_S) = \frac{\sum_{e \in G_T} \mu(e, G_S)}{\min(|G_T|, |G_S|)} \quad (1)$$

4.2 Maximum Common Sub Graph

We used three variations of maximum common sub graph metric to find similarity between text graph and class graph. Equation 2 illustrates the node similarity calculation method, equation 3 illustrates the edge similarity of the two graphs where as equation 4 illustrates directed edge similarity.

$$MCSNS = \frac{MCSN(|G_T|, |G_S|)}{\min(|G_T|, |G_S|)} \quad (2)$$

Maximum Common Sub graph Node Similarity (MCSNS) is the difference of target (GT) and source graph (GS).

$$MCSUES = \frac{MCSUE(|G_T|, |G_S|)}{\min(|G_T|, |G_S|)} \quad (3)$$

Maximum Common Sub graph Edge Similarity (MCSUE) is the total number of edges contained in the MCS after taking the difference of target graph (GT) and source graph (GS), without considering the direction.

$$MCSDES = \frac{MCSDE(|G_T|, |G_S|)}{\min(|G_T|, |G_S|)} \quad (4)$$

Maximum Common Sub graph Directed Edge Similarity (MCSDES) is the number of the edges contained in the MCS that have the same direction in the graphs. A full feature set is mentioned in Table 1.

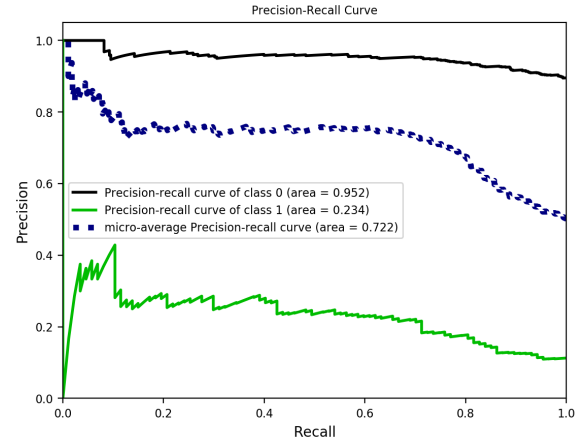


Figure 4: Precision Recall Curve of Multi Class problem

4.3 Model Selection

To solve binary class suggestive review classification, we used Tree-based Pipeline Optimization Tool (TPOT), (Olson et al., 2016). The labelled data is given as an input to TPOT, which returns the hyper tuned model for the binary class classification problem. After close analysis it is observed that the data suffers from class imbalance problem. To handle this problem, we used SMOTE ((Cummins et al., 2017)) a Python toolbox. TPOT gives extreme gradient boosting classifier tune parameters i.e. *GradientBoostingClassifier(learning_rate=1.0, max_depth=7, max_features=0.35, min_samples_leaf=19, min_samples_split=10, n_estimators=100, subsample=1.0)* for binary classification.

5 Result Analysis and Conclusion

The model achieved rank in the Coda Lab challenge is 31, with an evaluation value of 0.34. After the release of Gold set, the model is tuned again using the same TPOT library, which is then trained and evaluated. The results are shown in the figure and .

This work describes our suggestion mining technique which is a hybrid of graph structuring and classification algorithm. Our technique uses graph similarity metrics to find similar graphs from the dataset, which later serves as an input (feature vector) to the classification algorithm. The technique generates word graphs against given reviews which are replicated throughout the dataset using graph similarity techniques. Though the results need improvement

however they are convincing enough to show that use of word graphs with different vicinity window can be helpful in classifying suggestions related reviews within a domain. For further model improvements, use of different similarity metrics can be adopted as well as graph constructions using different vicinity window size can be tested.

References

- Usman Ahmed, Lubna Zafar, Faiza Qayyum, and Muhammad Arshad Islam. 2018. Irony detector at semeval-2018 task 3: Irony detection in english tweets using word graph. In *Proceedings of The 12th International Workshop on Semantic Evaluation*, pages 581–586.
- Fotis Aisopos, George Papadakis, Konstantinos Tserpes, and Theodora Varvarigou. 2012. Content vs. context for sentiment analysis: a comparative analysis over microblogs. In *Proceedings of the 23rd ACM conference on Hypertext and social media*, pages 187–196. ACM.
- Chris Cummins, Pavlos Petoumenos, Zheng Wang, and Hugh Leather. 2017. Synthesizing benchmarks for predictive modeling. In *2017 IEEE/ACM International Symposium on Code Generation and Optimization (CGO)*, pages 86–99. IEEE.
- George Giannakopoulos, Vangelis Karkaletsis, George Vouros, and Panagiotis Stamatopoulos. 2008. Summarization system evaluation revisited: N-gram graphs. *ACM Transactions on Speech and Language Processing (TSLP)*, 5(3):5.
- Asra Ishtiaq, Muhammad Arshad Islam, Muhammad Azhar Iqbal, Muhammad Aleem, and Usman Ahmed. 2019. Graph centrality based spam sms detection. In *2019 16th International Bhurban Conference on Applied Sciences and Technology (IBCAST)*, pages 629–633. IEEE.
- Valentin Jijkoun, Wouter Weerkamp, Maarten de Rijke, Paul Ackermans, and Gijs Geleijnse. 2010. Mining user experiences from online forums: an exploration. In *Proceedings of the NAACL HLT 2010 Workshop on Computational Linguistics in a World of Social Media*, pages 17–18.
- Bing Liu. 2012. Sentiment analysis and opinion mining. *Synthesis lectures on human language technologies*, 5(1):1–167.
- Andrew L Maas, Raymond E Daly, Peter T Pham, Dan Huang, Andrew Y Ng, and Christopher Potts. 2011. Learning word vectors for sentiment analysis. In *Proceedings of the 49th annual meeting of the association for computational linguistics: Human language technologies-volume 1*, pages 142–150. Association for Computational Linguistics.
- Sapna Negi, Kartik Asooja, Shubham Mehrotra, and Paul Buitelaar. 2016. A study of suggestions in opinionated texts and their automatic detection. In *Proceedings of the Fifth Joint Conference on Lexical and Computational Semantics*, pages 170–178.
- Sapna Negi and Paul Buitelaar. 2015. Towards the extraction of customer-to-customer suggestions from reviews. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 2159–2167.
- Sapna Negi and Paul Buitelaar. 2017. Inducing distant supervision in suggestion mining through part-of-speech embeddings. *arXiv preprint arXiv:1709.07403*.
- Sapna Negi, Tobias Daudert, and Paul Buitelaar. 2019. Semeval-2019 task 9: Suggestion mining from online reviews and forums. In *Proceedings of the 13th International Workshop on Semantic Evaluation (SemEval-2019)*.
- Sapna Negi, Maarten de Rijke, and Paul Buitelaar. 2018. Open domain suggestion mining: Problem definition and datasets. *arXiv preprint arXiv:1806.02179*.
- Randal S Olson, Nathan Bartley, Ryan J Urbanowicz, and Jason H Moore. 2016. Evaluation of a tree-based pipeline optimization tool for automating data science. In *Proceedings of the Genetic and Evolutionary Computation Conference 2016*, pages 485–492. ACM.
- Lubna Zafar, Muhammad Tanvir Afzal, and Usman Ahmed. 2017. Exploiting polarity features for developing sentiment analysis tool. In *EMASW@ESWC*.