SemEval-2019 Task 9 : Suggestion Mining from Online Reviews and Forums

Bharat Gaind, Karun Thankachan, Varsha Kuppur Rajendra

{bgaind, kthankac, vkuppurr}@andrew.cmu.edu

1 Introduction

Online platforms and e-commerce sites have become increasingly popular and contain a huge amount of data, where users share their views and opinions about specific products and services. As such, being able to leverage this data would prove critical in improving products and services offered. While there has been considerable research conducted in the area of Opinion Mining (i.e. extracting sentiment of users in passages) the task of Suggestion Mining (i,e, identifying if a sentence contains a helpful suggestion or not) remains relatively unexplored. A suggestion mining system could automatically extract and collate suggestions from multiple resources, providing invaluable feedback to businesses.

The main challenges in the domain of suggestion mining are (i) class imbalance as suggestions would be sparse among reviews (ii) figurative expressions, like sarcasm (iii) context dependency, where content surrounding would be determine whether a statement is a suggestion or not and (iv) long and complex sentences which makes it hard to track dependencies and detect suggestions.

The first attempts to solve Suggestion Mining had been limited to pattern matching Goldberg et al. [2] or rule based systems Ramanand et al. [7], which did not generalize well. Feature based work by Brun and Hagage et al [1] eventually led to research that indicated deep learning technique such as CNNs and LSTMs could improve prediction performance. However these models were data hungry. SemEval 2019 Task 9 [4] enables training of deeper neural networks by providing a larger training data set.

In Section 2 we briefly detail the dataset and in Section 3 we lay out the specifics for the task. In Section 4, we will briefly detail the solutions from the top 5 teams on the leaderboard for this task. In Section 5, we provide a detailed comparison of their architectures. In Section 6 we provide a few strategies we believe would be able to improve upon SOTA.

2 Dataset

The dataset consists of user reviews from two domains i.e. software developer suggestion forums (software called UserVoice) and hotel reviews (TripAdvisor). The details of the dataset are show in Table 1. The methodology to derive the dataset and a link to its Github repository can be found in [4].

Subtask	Domain	Suggestion/Non-suggestion		
		Training	Trial Test	Test
A	Software Developer suggestion forums (Uservoice)	1428/4296	296/742	87/746
B	Hotel Reviews (TripAdvisor)	448/7086	404/3000	348/476

Table 1: Details of the released datasets

3 Task Definition

A suggestion can be explicit or implicit. For instance “If you do end up here, be sure to specify a room at the back of the hotel” is an explicit suggestion. On the other hand “There is a parking garage on the corner of Forbes, so its pretty convenient” is an implicit one, because it does not directly act as a suggestion. Given this we have two subtasks in the SemEval task:

(i) **Subtask A** : The model will be trained on labeled sentences on suggestions from software developer domain. Then, given an an explicit software developer review, assign the label **Suggestion**, otherwise assign **Non-Suggestion**.

(ii) **Subtask B** : The model will be trained on suggestions from labeled sentences from software developer domain. Then, given an an explicit hotel review, assign the label **Suggestion**, otherwise assign **Non-Suggestion** i.e. cross-domain suggestion mining.

We aim to **tackle only Subtask A** while keeping Subtask B as a strech goal.

4 Related Work

During this competition, a number of interesting architectures were proposed for both the subtasks. We discuss briefly the architectures of the top 5 models and subsequently, draw a comparison across them.

4.1 Approach 1: Ensemble Modeling

First, we discuss the architecture that won the competition and was ranked first for subtask A and second for subtask B [3]. The model proposed is an ensemble of 4 independently trained BERT-Large-based models (using cross-validation) that are merged using simple voting. For each of the models, a special mark (CLS) is added at the beginning of the input sentence before passing it to the BERT, so that its corresponding output yields an embedding for the whole sentence, in addition to individual word embeddings output by BERT-Large.

The first model is a simple logistic model. It uses the first output (corresponding to CLS) of the BERT model, so as to feed only the entire sentence embedding to a logistic model which outputs probabilities of whether the sentence is a suggestion or not. The second model is based on Gated Recurrent Units. The hidden state of the GRU at time t is fed in a classification layer, and the model’s concatenated vector c (See [3] for more details) is used to train a binary classification logistic layer. This layer also uses individual word embeddings outputs of the BERT-large layer, in addition to the entire sentence embedding. The third model uses two CNNs on top of the BERT-large layer with Batch Normalization, RELU activation and max pooling. The fourth model uses Feed Forward Attention, since its easier to train on a small dataset and appropriately weighs the important elements in a sentence.

4.2 Approach 2: Joint Encoders

The proposed architecture [5] uses two encoders fed into 2-layer MLP that classifies whether a given sentence embedding is a suggestion or not. This architecture ranked 2nd in the competition with a F1-Score of 0.77 on SubTask A and 0.64 on SubTask B. The encoders are - (i) BERT-based encoder: The word embedding from BERT is passed to a CNN with max-pooling to produce a sentence level embedding for SubTask A. For SubTask B word embedding from BERT is passed to a bidirectional simple Recurrent Network. (ii) CNN-based encoder: GLoVe and CoVe word embedding are concatenated and passed to CNN which uses an attention mechanism to weigh them and produce sentence embeddings.

This combined encoding architecture helps to address the instability that the model seemed to face when using BERT encoder alone hon out of domain samples, while at the same time leveraging the contextualized embedding advantages provided by BERT.

4.3 Approach 3: Fine-tuning BERT

The paper [8] proposes different strategies for each subtask. For subtask A, the model uses target domain pre-training and for subtask B iterative distant supervision. The model starts with pre-trained BERT model and runs additional steps of pre-training using unlabelled corpus extracted from windows forum (target domain). The documents are split and the model is trained on two unsupervised tasks - masked language modelling and next sentence prediction. Finally, the model is fine-tuned for the target task. Performance improvement is seen on using BERT LARGE instead of BERT BASE. To further improve performance, an ensemble of three BERT models with different pre-training checkpoints are used and output scores are averaged.

For subtask B, the model does not use the provided training set, instead target domain corpus is extracted and divided into N chunks; starting with the initial rule-based baseline provided, predictions are made for the 1st chunk. The model is iteratively trained on subsequent chunks and predicting the next, till all chunks are trained on. This approach constructs a more suitable training set for the model to be trained on and the iterative training approach prevents over-fitting.

4.4 Approach 4: Hybrid Augmented Approach

The architecture by Zuang et. al. [9] has 3 blocks of layers - input encoding, model encoder and output layer. Pre-trained contextualized embeddings are used for encoding inputs for 2 reasons - disambiguation and sentence modelling, and transfer of external knowledge. Both ElMo and BERT is used and the latter is retained due to better performance. BERT here is used as static feature extractor (i.e. parameters are not updated) due to memory issues. To improve performance, linguistic features such as part-of-speech tagging and NER are concatenated with contextualized vectors. A variant of Self-Attention Network(SAN) is used for modelling long term dependencies. A fully connected block is used for output of encoding layer along with gate

Rank in competition (Subtask A)	Model	Subtask A	Subtask B
1	Ensemble	**0.7812**	0.8579
2	Joint Encoders	0.7778	0.6486
3	Fine-tuning BERT	0.7761	0.793
4	Hybrid-augmented	0.7629	NA
5	Rule-based	0.7488	**0.858**

Table 2: A comparison of the various models for Suggestion Mining

mechanisms to refine tokens by their importance. For predicting final probabilities, the output layer uses soft-max and loss includes an auxiliary component which assigns non-zero weights to only the contributing tokens and thus is helpful in filtering out trivial tokens.

To reduce the class imbalance challenge encountered in the dataset, prior probabilities of training instances is used to compute final probabilities ($Predictedprob/priori$). To deal with smaller training set, data augmentation technique is used to create additional training instances; translating existing records to Chinese and translating back into English. With BERT Large and an ensemble of 8 models like the one described above, the model achieved rank 4 F1 score for subtask A.

4.5 Approach 5: Augmenting R-CNN with a Rule Based Model

This paper proposes a R-CNN model (i.e. biLSTM followed by CNN with max pooling) which uses GloVe word embeddings [6] and is trained on software developer suggestions labeled dataset for prediction. In addition, a rule based model is used to detect word features (whether it contain words like suggest, should, allow etc), lexical patterns (modal words such as would, could and their negations) and semantic patterns (e.g. [do not]/[if only]). Based on the presence of these features that model increases confidence score by certain hard-coded amounts. The R-CNN model and rule/pattern based models are combined together in such a way that if the score of the decision from the R-CNN is above a particular threshold and rule-based model provides a conflicting decision then the decision from the R-CNN model is taken.

This rule/pattern based model proves to be unique among the solutions proposed and give that this solution was SOTA in SubTask B, the rule/pattern based approach shows great promise in cross-domain suggestion mining

5 Comparison of SOTA

Park et al [5] shows detailed ablation studies that indicate using BERT-Large based sentence encoders in a model would cause it to perform poorly on out of domain samples. As such they **utilized GloVe and CoVe embeddings to bolster performance on out of domain samples**. In [8] Yamamoto et al. however shows the pre-tuning BERT on web scrapped windows developer feedback website was able to alleviate this problem to an extent and provde competitive results. In addition Liu et al [3] used BERT-Large and seems to have **overcome the disadvantage by ensembling multiple models all trained on BERT-Large**.

While Park et al. attained high results in SubTask A, they score comparatively low on SubTask B (0.64 F1-Score). Ablation studies further indicated this score came in large parts due to correct identification only on short sentences. Comparing that architecture with Liu et al. we see that it lacks the ability to retain information as GRU would be doing. The same can be said for the biLSTM utilized by Potamias et al [6]. Potamias et al provides the SOTA for SubTask B, largely due to their rule/pattern based model that augments the R-CNN. It leverages the idea that **explicit suggestion has several word and lexical patterns that be detected**.

To address the lack of data for sub-tasks, nearly all models leverage large amounts of available unlabelled data in the form of pre-trained models like BERT. While Liu et. al. [3] and other models use BERT as an initialized embedding layer, [8] Yamamoto et.al. uses an additional BERT like step for pre-training on the corpus related to the task (e.g. windows forum). With this strategy of **target domain pre-training, Yamamoto et.al. achieves competitive F1 scores with fewer and simpler ensemble models** (averaging 3 models differing only in pre-training checkpoints)

Liu et. al. [3] uses CNNs to model the spatial perspectives and mimic n-gram models, GRU to capture the task structure and attention mechanism to model long range dependencies. Finally the results of multiple such models each addressing a specific structure of the inputs is ensembled using a voting strategy. This multiple model strategy is combined in the form of a Self Attention Network structure in Zhuang et. al. [9] which has a combination of convolution, self-attention and feed forward layers followed by blocks of convolution layers and gated mechanism at the end of block.

6 Proposed methods

We are planning to implement the Joint Encoder model (Rank 2 in Table 2) as the State of the art model to solve Subtask A. As mentioned before, Subtask B is a stretch goal and the primary focus will be on Subtask A. The reason is that first, the difference in performance of Ensemble model (Rank 1) and the Joint Encoder model in the competition for Subtask A is negligible. Second, the Joint Encoder paper [5] is much more detailed in terms of hyperparameters used and other model configurations, as compared to the Ensemble model, so it would be easier for us to reimplement in Assignment 3. It should be noted that there is no public codebase available for any of the 5 models discussed in this report. More details of the architecture on the Joint Encoder model are discussedin Section 4.2 and [5].

For Assignment 4, we have several ideas on how to outperform State of the art. First, to introduce a time-based perspective for processing sequential data, we are planning to introduce a GRU model and ensemble it (like [3] did) with the Joint Encoder model (which does not use RNNs for Subtask A in their paper). Next, just like [8] we are planning to fine-tune BERT on a domain-specific unlabeled corpus scraped from the universal windows platform developer feedback website. Intuitively, this will bring the overall system closer to the domain/problem at hand and expected to improve performance. Finally, inspired by [6], we are planning on incorporating rule-based models to the ensembled architecture as well, which should help in resolving standard repeating templates directly and increase performance.

References

[1] Caroline Brun and Caroline Hagege. Suggestion mining: Detecting suggestions for improvement in users' comments. *Research in Computing Science*, 70(79.7179):5379–62, 2013.

[2] Andrew B. Goldberg, Nathanael Fillmore, David Andrzejewski, Zhiting Xu, Bryan Gibson, and Xiaojin Zhu. May all your wishes come true: A study of wishes and how to recognize them. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 263–271, Boulder, Colorado, June 2009. Association for Computational Linguistics.

[3] Jiaxiang Liu, Shuohuan Wang, and Yu Sun. Olenet at semeval-2019 task 9: Bert based multi-perspective models for suggestion mining. In *Proceedings of the 13th International Workshop on Semantic Evaluation*, pages 1231–1236, 2019.

[4] Sapna Negi, Tobias Daudert, and Paul Buitelaar. Semeval-2019 task 9: Suggestion mining from online reviews and forums. In *Proceedings of the 13th International Workshop on Semantic Evaluation (SemEval-2019)*, pages 783–883, 2019.

[5] Cheoneum Park, Juae Kim, Hyeon-gu Lee, Reinald Kim Amplayo, Harksoo Kim, Jungyun Seo, and Changki Lee. Thisiscompetition at semeval-2019 task 9: Bert is unstable for out-of-domain samples. *arXiv preprint arXiv:1904.03339*, 2019.

[6] Rolandos Alexandros Potamias, Alexandros Neofytou, and Georgios Siolas. Ntua-islab at semeval-2019 task 9: Mining suggestions in the wild. In *Proceedings of the 13th International Workshop on Semantic Evaluation*, pages 1224–1230, 2019.

[7] J. Ramanand, Krishna Bhavsar, and Niranjan Pedanekar. Wishful thinking - finding suggestions and 'buy' wishes from product reviews. In *Proceedings of the NAACL HLT 2010 Workshop on Computational Approaches to Analysis and Generation of Emotion in Text*, pages 54–61, Los Angeles, CA, June 2010. Association for Computational Linguistics.

[8] Masahiro Yamamoto and Toshiyuki Sekiya. m_y at semeval-2019 task 9: Exploring bert for suggestion mining. In *Proceedings of the 13th International Workshop on Semantic Evaluation*, pages 888–892, 2019.

[9] Yimeng Zhuang. Yimmon at semeval-2019 task 9: Suggestion mining with hybrid augmented approaches. In *Proceedings of the 13th International Workshop on Semantic Evaluation*, pages 1267–1271, 2019.