Suggestion Mining with Transfer Learning via ULMFiT

Humza Haider
Department of Computing Science
University of Alberta
hshaider@ualberta.ca

Abstract

Suggestion mining is a new problem domain focusing on extracting suggestions from unstructured text. The SemEval 2019 Task 9 focuses on identifying suggestions from feedback posts on the Universal Windows Platform. Recent developments in text classification have shown pretrained language models can greatly improve performance across a range of problem areas. By applying the Universal Language Model Fine-tuning for Text Classification (ULMFiT) framework to the SemEval task we show the ULMFiT model is competitive in the domain of suggestion mining. This model outperformed the baseline, an LSTM model, and is currently scored second out of 59 participants in the SemEval Task.

1 Introduction

Suggestion mining is a rising field which aims to detect and tag suggestions from unstructured text (Negi et al., 2018). One use case is the extraction of suggestions from a product feedback website, *e.g.* "Please enhance the user interface". Automating suggestion identification is crucial for applications receiving large amounts of textual data as manual extraction may be infeasible.

Prior works have utilized support vector machines (Negi and Buitelaar, 2015), and different neural network architectures (Negi et al., 2016; Negi and Buitelaar, 2017) to approach suggestion mining. Most effective has been Negi and Buitelaar's use of distant supervision to train word embeddings and utilization of a Long short-term memory (LSTM) model to classify sentences into suggestions and non-suggestions. Specifically, distant supervision was utilized by using text scraped from wikiHow[1] – a website designed to help people perform tasks. The wikiHow website included a "Tips" section which is designed to give suggestions to readers as they complete their task (Negi and Buitelaar, 2017). By training word embeddings on these suggestions, Negi and Buitelaar were able to train a LSTM model which showed greater performance from previous works.

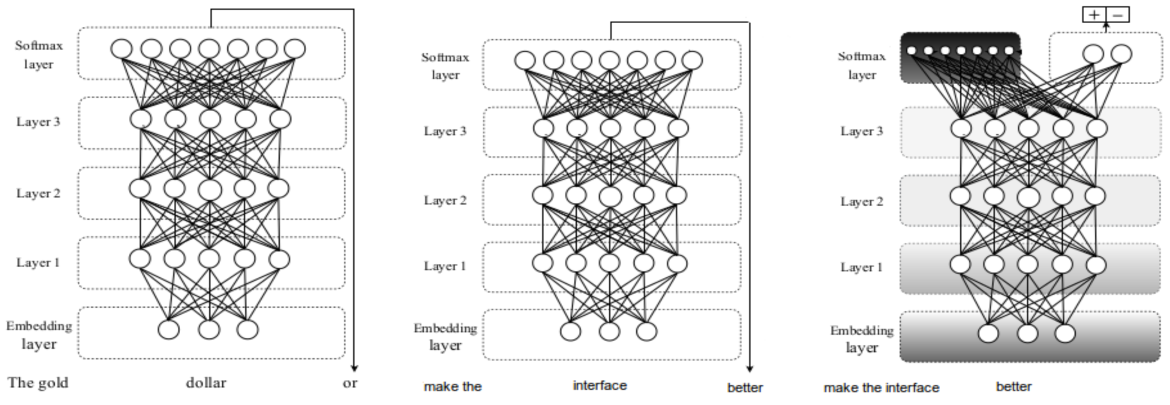
Our work takes the approach of transfer learning – often seen utilized in computer vision (CV) – which trains a neural network on a large corpora of text (images in CV) and alters the last few layers to learn task-specific information. We employ the approach of Howard and Ruder (2018) – the Universal Language Model Fine-tuning for Text Classification (ULMFiT) framework which trains a language model (LM) on a corpus of text and then fine-tunes a classifier on top of this LM. While Howard and Ruder explored this model for a number of text classification domains, suggestion mining is a new field in which transfer learning has not yet been employed. Our paper shows that by utilizing the ULMFiT model in suggestion mining we can achieve very powerful performance, currently ranking second (of 59 participants) on the SemEval 2019 Task 9 competition.

2 Methods

While the work of Howard and Ruder (2018) gives many of the details for the ULMFiT model, we instead give a brief, high level description of the model and encourage interested readers to see the original paper (Howard and Ruder, 2018). There are three stages to the ULMFiT model (see Figure 1); the first stage is to train a LM on a large corpus of text, here a corpus containing 28,595 Wikipedia articles (103 million words) (Merity et al., 2016).

Once this LM has been trained, we fine-tune

[1]https://www.wikihow.com/Main-Page



(a) LM pre-training (b) LM fine-tuning (c) Classifier fine-tuning

Figure 1: Architecture of ULMFiT model. From left to right the 3 stages, (a) the LM being trained on the Wikipedia corpus, (b) the LM being fine-tuned on the suggestion text, and (c) the classification layer being trained to identify suggestions. Figure adapted from (Howard and Ruder, 2018).

the LM with task specific text – the text to be labeled as suggestion and non-suggestion in this work. Doing this lets the LM learn the intricacies of the language that is used within the text classification problem. For example, in suggestion mining, the suggestions often are given by *imperative* sentences, *e.g.* "Please make the interface easier to use", whereas the text given in Wikipedia articles are often *declaritive*, *e.g.* "The University of Alberta is located in Edmonton".

Once the LM has been fine-tuned on the task specific text, a classifier is added to the top of the LM which takes in the pooled last hidden layer states of the text (here the pooled last hidden layer states of the sentence to be labeled as a suggestion or non-suggestion). Additional hyper-parameter information regarding dropout, learning rates, batch size, and gradual unfreezing of layers are given by Howard and Ruder (2018).

2.1 Baseline and comparison models

As discussed previously, Negi and Buitelaar (2017) used distant supervision to train word embeddings and then used an LSTM to predict suggestions. In this work they found that using the part of speech (POS) tags embeddings produced the model with the highest F1-score and so we use these same embeddings and LSTM model framework as a comparison model. Additionally, we compare against a baseline given by SemEval which identifies suggestions from key words (*e.g.* recommend) and POS tags (a modal verb followed by a base form verb, *e.g.* "should include").

3 Experiments

Data is provided by the SemEval 2019 Task, which is comprised of 8053 training and 592 development sentences from a feedback forum posted on the Universal Windows Platform. Testing data is to be released on January 10, as per the SemEval Task schedule. After removing duplicate sentences we were left with 6847 sentences which were split into 6162 training sentences and 685 development sentences – the 592 sentences originally supplied for development were treated as the testing set. The testing and development sets were balanced (50% each of suggestion/non-suggestion) whereas 23% of the training sentences were suggestions. Prior to model training, suggestions were randomly oversampled to create a balanced training set.

The evaluation metric (as given by the SemEval Task) is the F1-score. For the distant supervision method given by Negi and Buitelaar (2017), we convert words into their respective POS tags using the Natural Language Toolkit (NLTK) and use the same hyper-parameters specified in (Negi and Buitelaar, 2017).

Due to computational overhead, training of the ULMFiT model used default hyper-parameter settings given by Howard and Ruder (2018) except for dropout and number of epochs. A variety of combinations of these parameters were tested – details of which can be found in the attached code. In addition to these models we also examined the impact of the dataset used to pretrain the

Model	F1 - Dev	F1 - Test
Baseline	0.718	0.722
Distant Supervision	0.655	0.653
ULMFiT - Wiki	**0.873**	**0.837**

Table 1: F1-score results on development and test sets. Here the LM for ULMFiT is pretrained using the full Wikipedia corpus.

Model	F1 - Dev	F1 - Test
ULMFiT - Wiki	**0.876**	**0.841**
ULMFiT - wikiHow	0.867	0.826
ULMFiT - None	0.849	0.805

Table 2: The F1-score using the ULMFiT model pretrained on differing datasets.

LM (Stage 1 of the ULMFiT model). Namely we used (1) the Wikipedia corpus originally used by Howard and Ruder, (2) the wikiHow dataset used in Negi and Buitelaar (2017) and (3) we performed no pre-training and instead skipped to Stage 2 of the ULMFiT model.

3.1 Results - Baseline, Distant, and ULMFiT - Wiki Models

Table 1 gives the F1-score for the three models (Baseline, Distant, and ULMFiT - Wiki). Interestingly, the baseline did much better than the LSTM model using word embeddings trained on wikiHow. The ULMFiT model showed a much higher performance than both the LSTM model and the baseline scoring 0.155 points higher than the baseline on the development set and 0.115 points higher on the test set. Additionally, the ULMFiT model scores second overall (out of 59 current participants) for the SemEval 2019 Task.

3.2 Results - Modifying the pretrained LM

Table 2 gives the results when changing the dataset used for pretraining the LM portion of the ULMFiT model. The LM trained on the Wikipedia dataset outperforms the model with no pretraining, indicating the ULMFiT model is indeed learning some language structure by pretraining on the Wikipedia corpus. Pretraining on the wikiHow corpus also showed improvement over no pretraining but did not outperform the model pretrained on the Wikipedia corpus. This is likely due to the Wikipedia corpus being roughly 100 times larger than the wikiHow corpus.

3.3 Error Analysis

The baseline model had a very strong recall (0.918) but very poor precision (0.585) on the development set. This is partially due to the baseline including key words such as "should" which labels some supportive sentences as suggestions, *e.g.* "If a user wants an application to host content that should be their prerogative". Notice this sentence is not a suggestion but rather support for the suggestion they are requesting.

The ULMFiT model saw similar strength in the recall (0.965) and a much higher precision (0.803). Of the suggestions missed, the most common type were phrased as commands, *e.g.* "Add left to right orientation", instead of requests, *e.g.* "Please add right to left orientation". The baseline model had the same challenge but the ULMFiT model was able to still identify many more commands than the baseline.

For false positives, the ULMFiT model struggled the most with requests that were not suggestions, *e.g.* "please tell me when this is fixed" and "please make it happen". While these sentences are requests, they are not requesting a feature and thus are not considered a suggestion. The baseline also struggled with these sentences, labeling them as suggestions.

4 Conclusion

We applied the ULMFiT model to the domain of suggestion mining, leading to very competitive results (placing second overall in the SemEval 2019 Task 9 competition). Additionally, this model outperformed the baseline and the distant supervision LSTM model. We showed that the dataset used for pretraining the LM in the ULMFiT model greatly impacts the F1-score performance. Future work should examine if more variation in hyperparameters of the ULMFiT model can improve performance. Additionally, work which can identify requests that are not suggestions could greatly impact the precision and thus the overall model performance.

Acknowledgments

We thank Greg Kondrak for his insightful comments during the construction of this manuscript. We would also like to express our gratitude to the three anonymous reviewers for their suggestions and recommendations.

References

Jeremy Howard and Sebastian Ruder. 2018. Universal language model fine-tuning for text classification. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. volume 1, pages 328–339.

Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. 2016. Pointer sentinel mixture models. *arXiv preprint arXiv:1609.07843* .

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, Maarten de Rijke, and Paul Buitelaar. 2018. Open domain suggestion mining: Problem definition and datasets. *arXiv preprint arXiv:1806.02179* .