Improving Sentiment Analysis with Multi-Task Learning of Negation

Project Report

By

Team Members

Akshaya Nagarajan

Pooja Patil

Sivaranjani Kumar

Vigneshkumar Thangarajan

December 7th, 2020

Table of contents

Abstract:

Sentiment Analysis is a process where one can mine people's opinions from a piece of text. At first glance, this task may look like a straightforward text classification problem, but once deep dived, one can find out challenges which can affect sentiment analysis accuracy. For instance, predicting sentiment just using the words in a sentence can lead to major pitfalls such as detecting sarcasm, irony, word ambiguity, use of negating words, also multipolarity. Out of all these phenomena, negation is the most prevalent. Any sentiment analysis model must be able to identify negation and try to remove the effect that its scope has on the final sentiment of a text. In this project, we propose to use Multi-task learning approach, using a cascading and hierarchical neural architecture of LSTM layers. This architecture will explicitly train the model with negation as an auxiliary task and eventually help in improving the main task of sentiment analysis.

Introduction:

In this project, we propose to model sentiment prediction along with negation detection in a multi-task learning set-up. [1], annotates prediction tasks on same dataset with same output units, we propose to take auxiliary data from different dataset and domains with different units of classification across tasks. Also, as an auxiliary task, we will experiment learning sequence-labeling of negation cues and scopes based on two negation datasets, whereas our main task will be sentence and tweet level classification of sentiments.

Multi-task learning (MTL), a subfield of machine learning, solves related tasks simultaneously allowing a ML algorithm to include a useful inductive bias by restricting the search space of representations to those that are predictive for both tasks. We plan to incorporate negation information as a cascading architecture for sentiment prediction, where the intermediate representations learned for predicting negation is fed into subsequent layers along with skip-connections. The lower layers of the network are shared as well as supervised by both auxiliary and main tasks. The final layer is committed for the sentiment prediction task.

Related Work:

Negation is the most prevalent phenomenon and has a direct effect on sentiment analysis. Early research proposed by [5] attempted to model negation by reversing polarity of a negated word, whereas [6] attempted to model negation by modifying the polarity of a negated word. An assumption that final polarity/sentiment of a text is a function of prior probabilities of verbs, nous, adjectives present in the text is made. As per [5], a heuristic approach is utilized to determine the scope of negation by finding common negation cues, whereas [6] determines the scope of negation based on the distance from the cue.

In [7] an approach of attaching a negation tag ("neg") to words which were assumed to be scope was followed. However, such an approach leads to issues like sparsity and varying results. Such an approach is not feasible, as the sentiment analysis model could not explicitly consider both original and negated features when it comes to connecting these features.

Most of the sentiment analysis work proposed by [8], [9], [10] used negation detection systems, to improve the feature space of sentiment analysis model, which eventually led to improved results in sentiment classification.

Most of the existing traditional approaches employed for the task of Negation Detection consider both lexical and semantic properties of input text. Additionally, to analyze the input prior to negation detection, syntactic parsing is used bases on constituency based as proposed by [12] and on dependency-based as proposed by [10].

Commented [GU1]: Follow Up: [12] and

Approaches like neural modeling are employed for the task of negation detection. As described by [13] a CNN model is used for detecting negation on the abstract section of Bioscope data corpus. This model is applied to the syntactic parts of cue and candidate tokens. A novel approach presented by [14] compare two neural architectures on Conan Doyle-neg corpus which is a simple feed-forward network and a bi-directional LSTM model.

Since Syntactic information is useful for scope resolution in a text input, cue detection tasks require only simpler surface information. [15] detects cue in text by treating the set of cue words as a closed task and applies a disambiguation-based approach to the problem of cue detection.

Datasets:

Stanford Sentiment Treebank (SST): This dataset contains 11,855 sentences which are taken from the movie reviews specific to English-language. There are two different versions to this dataset. They are SST-fine and SST-binary setting. SST was annotated for fine-grained sentiment with five different class labels. They are strong negative, negative, neutral, positive and strong positive which are specific to the SST-fine. Whereas in the SST-binary setting has 9,613 sentences where the neutral label is removed, instead strong and the normal label in SST-fine are merged to form only positive and negative labels.

Conan Doyle Neg (*Sem 2012): The negation detection model with annotation will work on the sentences to extract the cues (words that change the polarity if the sentences) and scopes (words that gets affected by the cues). This Conan Doyle dataset is used for training the negation model which contains the stories of Conan Doyle and are manually annotated for the negation cues, negation scopes and events. The Bioscope corpus has annotation schemes which was also employed in Conan Doyle but with some major changes. This dataset contains the shared task version from 2012 SEM negation detection. This version consists of 848 negation sentences, out of 787 development sentences, 144 are negated and comprising total 3,640 sentences in training set. The test set consist of total 1089 sentences and out of these 235 sentences are negated. Conan Doyle annotates the different cue types they are sub-token, word-based and multi-word negation scope.

Methods:

According to [4], sarcasm in the sentence is detected as negative sentiment. We leverage this by improving by using two models, sentiment classification and negation detection. The two models are implemented using multitask learning fashion, where a single neural network is used to perform more than one classification task.

Task Definition:

We solve two tasks with a single network. Given a sentence, we split the sentence and assign it both a sentiment tag (strong positive, negative, positive, neutral, and strong negative) and a negation tag.

Input Representation:

The inputs sentences are passed through the embedding layer where we employed the google.txt file to create the input representations. We use bidirectional Long Short-term memory (BiLSTM) to extract the features from the embedding layer. The CRF with Viterbi decoding is used to find the assignment of labels and train the model to minimize the negative log likelihood. Given a sequence of tokens, negation model first embeds, these in an embedding layer and then uses the BiLSTM to create a contextualized representation of these tokens.

Classification:

Multi-task learning aims to learn multiple different tasks simultaneously while maximizing performance on one or all of the tasks. A cascading architecture is adopted to perform the classification task. At the lower layer, we perform the negation cue and negation scope prediction which is defined as auxiliary task. The higher layer, which is defined as main task does the polarity prediction. This cascading architecture helps to pass the intermediate representation rather than passing the negation predictions directly.

Training:

The model is implemented using PyTorch libraries. The learning rate for sentiment analysis task is 1 X 10^-3 and learning rate for auxiliary task is 1 X 10^-4. The model is trained in a hierarchical fashion for 5 iterations. In each iteration, we train the model for 10 epochs. We also defined random seeds in order to get reproducible results in training. Pretrained embedding model from Google is used as initial embedding layer on text input.

Architecture:

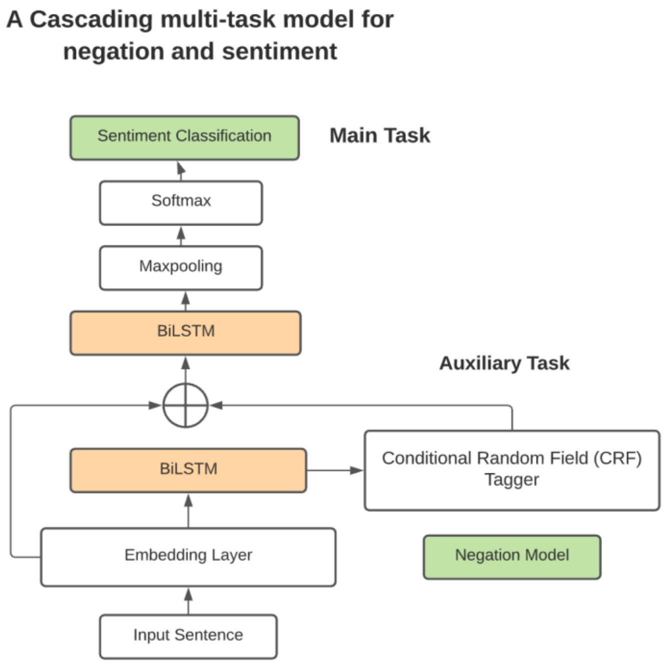

Fig.1 Architecture Diagram

Our implementation involves a cascading architecture where two BiLSTM models are used. The lower level of the model is for the auxiliary task, such as negation cue and scope prediction. The higher-level of the model is for the main task, i.e., sentiment classification. The first BiLSTM in the lower level is shared between the sentiment and negation tasks and feeds the output into a higher-level BiLSTM. The BiLSTM in higher-level is dedicated to sentiment prediction. The architecture involves two models – negation and sentiment models.

Negation Model:

The BiLSTM layer extracts the features from the embedding layer. The input sentences are provided with negation cues and scopes by BiLSTM along with the CRF tagger. The conditional Random Field Tagger is used along with the Viterbi decoding to find the most probable assignment of labels. This BiLSTM model predicts the cues and scopes in one pass and works towards minimizing the negative log-likelihood.

Sentiment Model:

The embedding layer used earlier and the low-level BiLSTM layer are used to create a contextualized representation of the input tokens. The skip connections are utilized to concatenate each of the original embeddings to the contextualized representation. This sequence is the input for the BiLSTM at a high-level. The next layer involves performing max pooling on the output of BiLSTM and passing it to the SoftMax layer to calculate the sentiment class probabilities as final steps.

Experiments:

For model training, we imported all the necessary libraries like torch, Autograd, RNN's packed sequence. We also import tensor board for visualization. The model as we saw in architecture section, consists of two layers one that caters to the negation cues and scope and the other that caters to the sentiment classification. The output of negation model is used to feed as input to sentiment classification model. We define two different variables of learning rate for both the auxiliary task and the main task. Also, two different Adam optimizer variables are defined to expedite the learning task for both the negation task and the sentiment task.

We ran the training loop for 5 iterations and in each iteration, we train the model for ten epochs. Initially, we tried 5 different learning rates for the auxiliary task and main task. Those experiments helped us to finalize on the best learning rate for both the tasks. All the model hyperparameters are finalized after several experiments.

Deployment:

The model is deployed as web application using Render. Render is a cloud provider which provides UI based model deployment services. It comes with a default python, flask-based web application template in GitHub. This template is available for image classification model which is trained using FastAI. To accommodate the text classification model which is trained using PyTorch APIs we need to modify the deployment logic.

The text classification model is a custom hierarchical model as we discussed in the architecture section. This model has a predict function which accepts a specific batch of text input. The output of the model will be any of five classes of SST fine dataset. The web application gets a sentence as text input. This text input is then tokenized based on the availability of the words in Vocabulary. The tokenized words are then passed on to the embedding layer and subsequently the model analyzes it for the negation cues which is then used for final sentiment classification.

Conclusion:

The main objective of this paper is not to achieve new state-of-art results for sentiment analysis but rather to gauge the relative contribution of negation task as auxiliary task in sentiment analysis. However, we still achieved competitive results at the end. The single-task model achieves an average accuracy of 46.49 on SST-fine. These results are better than standard performance for a Bidirectional LSTM model 45.6 and competitive with similar models. The extensive analysis of the results reveals several effects of using negation detection as an auxiliary task. On one hand, we find that even a small amount of annotated negation data allows a multi-task learner to improve its performance, while on the other hand, it is necessary to have enough sentiment data to achieve relatively good performance in order to see improvements in single task learning models.

Reference:

[1] Bingel, J. and Søgaard, A. (2017). Identifying beneficial task relations for multi-task learning in deep neural networks. In Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics, pages 164–169, Valencia, Spain.

[2] Improving Sentiment Analysis with Multi-task Learning of Negation J E R EM Y B A R N E S , E R I K V E L L D A L , and L I L J A Ø V R E L I D Language Technology Group, University of Oslo email: {jeremycb,erikve,liljao}@ifi.uio.no

[3] De Mattei, Lorenzo., Cimino, Andrea., & Dell'Orletta, Felice, "Multi-Task learning in deep neural network for sentiment polarity and irony classification", in ILC-CNR

[4] Majumder, Navonil., Chhaya, Niyati., Poria, Soujanya., Cambria, Erik., Peng, Haiyun., & Gelbukh, Alexander (Mar 2019), "Sentiment and Sarcasm classification with multitask learning"

[5] Hu, M. and Liu, B. (2004). Mining opinion features in customer reviews. In Proceedings of the 10th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pages 168–177, Seattle, USA.

[6] Taboada, M., Brooke, J., Tofiloski, M., Voll, K., and Stede, M. (2011). Lexicon-based methods for sentiment analysis. Computational Linguistics, 37(2):267–307.

[7] Pang, B., Lee, L., and Vaithyanathan, S. (2002). Thumbs up? Sentiment classification using machine learning techniques. In Proceedings of the 2002 Conference on Empirical Methods in Natural Language Processing, pages 79–86, Philadelphia, USA.

[8] Councill, I., McDonald, R., and Velikovich, L. (2010). What's great and what's not: learning to classify the scope of negation for improved sentiment analysis. In Proceedings of the Workshop on Negation and Speculation in Natural Language Processing, pages 51–59, Uppsala, Sweden.

[9] Cruz, N. P., Taboada, M., and Mitkov, R. (2016). A machine-learning approach to negation and speculation detection for sentiment analysis. Journal of the Association for Information Science and Technology, 67(9):2118–2136.

[10] Lapponi, E., Velldal, E., Øvrelid, L., and Read, J. (2012b). UiO2: Sequence-labeling negation using dependency features. In Proceedings of the First Joint Conference on Lexical and Computational Semantics, pages 319–327, Montreal, Canada.

[11] Morante, R. and Blanco, E. (2012). *SEM 2012 shared task: Resolving the scope and focus of negation. In Proceedings of the First Joint Conference on Lexical and Computational Semantics (*SEM), pages 265–274, Montr´eal, Canada.

[12] Read, J., Velldal, E., Øvrelid, L., and Oepen, S. (2012). UiO1: Constituent-based discriminative ranking for negation resolution. In Proceedings of the First Joint Conference on Lexical and Computational Semantics (*SEM), Montreal, Canada.

[13] Qian, Z., Li, P., Zhu, Q., Zhou, G., Luo, Z., and Luo, W. (2016). Speculation and negation scope detection via convolutional neural networks. In The 2016 Conference on Empirical Methods in Natural Language Processing.

[14] Fancellu, F., Lopez, A., Webber, B., and He, H. (2017). Detecting negation scope is easy, except when it isn't. In Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics, pages 58–63, Valencia, Spain.

[15] Velldal, E., Øvrelid, L., Read, J., and Oepen, S. (2012). Speculation and negation: Rules, rankers, and the role of syntax. Computational Linguistics, 38(2):369–410.