Utilizing Sentence Level Embeddings for Improved Transfer Learning

Lance Miles
UC Berkeley,
Berkeley, CA

Matthew Vay
UC Berkeley,
Berkeley, CA

Abstract

Google recently released the Universal Sentence Encoder to embed entire sentences into 512 dimensional vectors to assist with transfer learning tasks. This offers a prime opportunity to evaluate whether sentence level embeddings can assist in zero shot learning. In this paper, we investigate the use of the state-of-the-art Universal Sentence Encoder (USE) alongside individual word embeddings to determine if sentence embeddings improved accuracy rates in our transfer learning tasks. The data used to train our models is an Amazon product review dataset consisting of over one million product reviews across 12 different product domains, which includes video games, books, clothing and more. Our models were trained using a convolutional neural network (CNN) architecture similar to that of Kim, 2014. After training our models on the Amazon product review data, we tested our models on IMDB, Yelp, and Twitter datasets and found that the use of sentence level embeddings improved the accuracy of our models in all zero-shot learning tasks.

1 Introduction

Obtaining labeled data for a domain specific task can be labor intensive and expensive to pursue. For this reason, transfer learning (sometimes referred to as domain adaptation) is an important field of research in natural language processing where domain specific data may be limited. Transfer learning allows a researcher to train machine learning algorithms on available data that may be similar to the data or task they want to apply the model to. This pretrained model is then used to accurately predict on a new domain or task. In this way, prior knowledge from one domain is leveraged to predict on a different domain or task.

An important part of information-gathering behavior has been to find out what other people think (Pang and Lee, 2008). Understanding user sentiment towards a domain or product is of great value for many industries. Although there is a wealth of data available, much is left unlabeled and in the form of unstructured reviews in blogs, comments, and internet articles. This makes organizing data time consuming with a lot of manual work done by humans to classify a particular text. However, product review data lends itself to building strong sentiment classifiers because humans write out a review and then assign how they feel about it, generally on a scale from 1 to 5. This an opportunity to take full advantage of these large prelabeled datasets for domain adaptation tasks. Our goal is to discover ways that we can take models trained on these large labeled datasets and transfer the knowledge to other datasets that may not have a rating associated with the text.

2 Background

While much of the initial focus for CNN models were related to image tasks, in recent years, CNNs have proved to be adept at classification tasks dealing with text data (Kim, 2014; Zhang, 2015). In Natural Language Processing, the use of trained or pretrained word embeddings such as Word2Vec (Mikolov et al., 2013) and GloVe (Pennington et al., 2014) has become common practice. These embeddings focus on how frequently words appear together in texts, but this alone does not provide the full picture around the context in which a word is used. Although CNNs allow us to derive context within a fixed window, complex sentences that require longer ranging windows can be harder to classify properly (Song et al. 2018).

While the combination of trained or pre-trained word embeddings with CNNs are strong baselines for sentiment classification, there are ways to improve on them when attempting to solve problems related to transfer learning. In 2018, Cer et al. at Google released the Universal Sentence Encoder (USE), which takes in a sentence or paragraph and encodes it into a 512-dimensional vector. These sentence level embeddings may lend themselves to deriving greater context from available data. We hypothesize that using sentence/paragraph embeddings in conjunction

with word embeddings will improve accuracy of our model, especially in transfer learning tasks.

3 Methodology

Several CNNs were trained using either pre-trained or trained word embeddings to classify sentiment on Amazon product reviews. These models were then used to evaluate sentiment on IMDB movie reviews, Yelp restaurant reviews, and Tweets in our transfer learning tasks. Next, the baseline CNN models were used alongside the embeddings generated from Google's Universal Sentence Encoder to see if the addition of the Universal Sentence Encoder's embeddings improved accuracy on each of these classification tasks. The details for each of the datasets and models used are summarized in this section.

3.1 Datasets

This section provides an overview of the datasets utilized in training and evaluating our models. All datasets used in this experiment were from academic resources or are publicly available.

3.1.1 Amazon Reviews: Baseline Model

We obtained over 82 million product reviews from McAuley et al. to build our training dataset and to train our models on. The goal of our experiment was to train a CNN on multiple product domains in hopes of creating a "generalizable" sentiment classifier for our transfer learning tasks. To ensure that we had balanced data across different product domains, we selected the top 12 domains that had the most reviews. These data were randomly sampled and partitioned so that there were equal numbers of positive and negative reviews within each product domain. The script we built gave preference to 1- and 5-star reviews, but also included reviews with 2 and 4 stars. These data were changed into a binary classification task where a 1- or 2-star ratings resulted in a sentiment label of 0 (negative), and a 4- or 5-star review resulted in a sentiment of 1 (positive). This amounted to a training dataset with 1.2 million reviews across 12 product domains, and a 240k review test set across the same 12 product domains.

3.1.2 Yelp Reviews: Transfer Learning

Prior to obtaining literature data, we tested our model on a Yelp dataset that we created. The structure for yelp reviews is very similar to amazon, but cover a different domain that is not captured in the Amazon dataset (restaurant reviews). For this reason, we chose to create a Yelp dataset that mirrored the test dataset we created with Amazon data and test it in our zero-shot learning task. These data were downloaded from Kaggle.

3.1.3 Literature Datasets: Transfer Learning

The following datasets have available literature benchmark results (Table 1) as a comparator for our models trained on Amazon data. It is important to note that in each of these cases the author has trained their models on the training dataset available for each task. We do not expect to achieve these accuracies on our models because we have trained using strictly Amazon review data.

Dataset	Model	Accuracy
IMDB	[JZ14]: CNN	91.34%
Yelp Polarity	[JZ16]: CNN	95.40%
Amazon Polarity	[JZ16]: CNN	94.49%
SST-2	[CM17]: MVCNN	91.20%
Sentiment140	[YH16]: MVCNN	88.20%

Table 1: Benchmark accuracies for datasets used in zero-shot learning tasks in our experiments.
[YS16]: Yin and Hinrich (2016); [CM17]: Camacho-Collados and Mohammad (2017); [JZ14]: Johnson and Zhang (2014) ; [JZ16]: Johnson and Zhang (2016)

3.1.4 IMDB (Zero-Shot Learning):

For our primary zero shot learning task, we used the IMDB movie reviews dataset provided by Maas et al., which consists of 25k positive and 25k negative movie reviews for binary sentiment classification.

3.1.5 Yelp Polarity (Zero-Shot Learning):

Zhang et al (2015) released a series of polarity datasets that reference state of the art results for sentiment classification. For this reason, we felt it would be important to test these datasets on our models. Although we do not expect to achieve state-of-the-art results in our zero-shot learning task, our hope is that our accuracies will be competitive, and that the accuracies of our models will improve with the inclusion of the universal sentence embeddings

3.1.6 Amazon Polarity (Zero-Shot Learning):

As mentioned previously, our training and test set consisted primarily of 1- and 5-star reviews. Zhang et al polarity benchmark datasets include an amazon polarity dataset that includes has much more reviews that were 2 and 4 stars. For this reason, we felt it

would be important to compare our trained model against this benchmark test set.

3.1.7 Sentiment140 (Zero-Shot Learning):

Another interesting dataset that we sought to evaluate was the Sentiment140 dataset released by Go et al. This is a binary sentiment dataset that consists of anonymized tweets collected from Twitter. Based on the methods used to classify tweets as positive and negative, we did not expect these results to translate well in our zero-shot learning task, but we pursued this nonetheless.

3.1.8 SST-2 (Zero-Shot Learning):

Socher et al. released the Stanford Sentiment Treebank for sentiment analysis on movie reviews. For our purposes, we are interested in utilizing the Stanford Sentiment Treebank binary classification test dataset (SST-2) to evaluate our model's accuracies on this transfer learning task.

3.2 Text Preprocessing:

For our baseline CNN models, we standardized the review text by removing contractions, lowercasing words, and standardizing punctuation. We also investigated the removal of stop words, and saw minimal improvements in our accuracy. It is also important to note that reviews were typically around 40 words long so we decided to move forward with keeping stop words for our models. When feeding reviews into the Universal Sentence Encoder, we left the text as close to the original form as possible, except we standardized punctuation and lowercase words. This is because we did not want to make assumptions about how the universal sentence encoder handles text features when it translates the text into a 512-dimensional vector.

3.3 Universal Sentence Encoder:

The Universal Sentence Encoder released by Cer et al. comes in two different formats. For the purpose of our experiments and for testing our models we used the DAN architecture over the Transformer architecture. We tested the outputs of the Universal Sentence Encoder in two key facets. First, we fed in entire reviews (stemmed to 200 words) to generate review level embeddings. Next, we fed in individual sentences (up to three) to generate sentence level embeddings. Another interesting experiment that we pursued was to take the average of the sentence level embeddings for each case and test the accuracy.

3.4 Modeling:

The review text data were used in three ways when developing our sentiment classifiers. First, we used individual word embeddings to pass through a series of convolutional filters in our CNNs. Next, we fed entire reviews that were stemmed to a length of 200 words through the Universal Sentence Encoder to use in our CNNs. Lastly, we fed individual sentence embeddings or averaged sentence embeddings for each review in a series of convolutional filters to see if this would increase our model accuracies. Detailed descriptions for each model can be reviewed, below.

3.4.1 Baseline Models:

Prior to experimenting with the Universal Sentence Encoder, we wanted to investigate the accuracies of our models using several different baselines. These models are described in this section.

Dense Neural Network:

Reviews stemmed to 200 words were fed into the universal sentence encoder to obtain 512-dimensional vector representations of the review. These inputs were fed into a dense layer and then fed into a softmax layer for prediction.

CNN - Individual Words:

The architecture for the three baseline CNNs follow the architecture first proposed by Kim, 2014. We tokenize the inputs (individual words) and feed these into the CNN. Depending on the model, we either train our own embeddings from the amazon data, or we pull embeddings from the pretrained GloVe or Word2Vec embeddings. These embeddings were fed through a series of convolutions and then the outputs were funneled through a max pooling layer. The outputs of the max pooling layer were concatenated and fed into a dense layer and then into a softmax layer to get our prediction.

CNN - Sentence Embeddings:

We also developed CNN models to take in the individual and average sentence level embeddings from the Universal Sentence Encoder. These embeddings were fed into a convolutional layer, and then underwent max pooling. The outputs of the max pooling layer were concatenated and fed into a dense layer and then into a softmax layer for prediction.

3.5 Experimental Models:

The goal of our experiment is to establish whether the inclusion of the Universal Sentence Encoder

Model Accuracies

By Dataset

Dataset	Model Type: Average Sent	CNN Glove	CNN Glove USE	CNN TE	CNN TE USE	CNN w2V	CNN w2V USE	Indiv. Sent	USE Softmax
IMDB Reviews	73.9%	82.9%	86.1%	82.2%	85.5%	84.5%	85.2%	76.0%	79.7%
SST	49.3%	80.3%	82.9%	80.7%	81.2%	81.8%	82.0%	50.2%	75.2%
Twitter	67.4%	65.8%	67.7%	63.9%	64.8%	65.6%	67.9%	65.4%	66.9%
Yelp Reviews	85.1%	95.3%	95.9%	95.3%	95.4%	95.5%	95.9%	85.4%	91.8%
Zhang Amazon	81.2%	88.8%	90.5%	90.3%	90.6%	89.1%	89.7%	83.6%	84.2%
Zhang Yelp Reviews	77.6%	90.3%	90.7%	90.2%	90.5%	90.0%	90.8%	78.5%	84.0%

Figure 1: Model accuracies by dataset.

embeddings alongside word embeddings improved accuracy for our transfer learning tasks when compared to our baseline models.

CNN:

Much like our baseline CNN models, individual word embeddings (trained, GloVe, or Word2Vec) were fed through a convolutional layer and then into a max pooling layer. Next, similar to Bengio et al., we concatenated the outputs of the max pooling layer with the outputs of the review level embeddings from the Universal Sentence Encoder. This was fed into a dense layer and then fed into a softmax layer for prediction.

4 Results

The performance for each of our models on each task is summarized in Figure 1, which shows the accuracies achieved when evaluating on IMDB, SST-2, Twitter (Sentiment140), Yelp, Zhang Amazon Polarity and Zhang Yelp Polarity data. As seen in figure 1, the concatenation of the USE resulted in anywhere between a 0.3 percentage point to as much as 3.2 percentage point increase in accuracy. Figure 2 breaks down the accuracies by sentiment for each dataset and model. We will discuss the results for each of these results, below.

4.1 IMDB and SST-2 Movie Reviews Datasets:

Both IMDB and SST-2 movie review datasets achieved similar accuracies, with Glove + USE achieving the highest accuracy in each task (86.1% and 82.9%, respectively). Inclusion of the USE embeddings improved accuracy by as much as 3.86% (IMDB dataset) when compared to baseline (82.9% accuracy), which is about an 18% reduction in error rates. Figure 2 displays the accuracies for positive and negative sentiment reviews. For IMDB reviews, our models typically predicted negative sentiment reviews with higher accuracy than the positive sentiment reviews. When comparing CNN Glove to the CNN Glove + USE model, we see that the positive sentiment accuracy rate jumps from roughly 73.1% accuracy to 82.4% accuracy, which is a 34.57% reduction in error rates.

4.2 Twitter (Sentiment140) Dataset:

The dataset with the lowest accuracy was Twitter (Sentiment140) which achieved 66% accuracy rate on average. As mentioned previously, Yin and Hinrich (2016) achieved over 88.20% accuracy on the Sentiment140 test dataset. It is important to reiterate that the authors trained their models on twitter data, using twitter specific embeddings and lexicons. Our

Dataset	Sentiment	Model Type: Average Sent	CNN Glove	CNN Glove USE	CNN TE	CNN TE USE	CNN w2V	CNN w2V USE	Indiv. Sent	USE Softmax
IMDB Reviews	0	76.0%	92.7%	89.7%	95.1%	91.9%	89.0%	89.6%	77.6%	86.0%
	1	71.8%	73.1%	82.4%	69.2%	79.1%	80.1%	80.8%	74.4%	73.4%
SST	0	42.2%	80.3%	79.3%	78.2%	82.3%	82.1%	79.8%	42.5%	62.0%
	1	56.3%	80.3%	86.6%	83.2%	80.1%	81.5%	84.2%	58.0%	88.4%
Twitter	0	58.5%	67.8%	59.8%	60.0%	63.5%	68.1%	63.3%	56.5%	54.0%
	1	76.3%	63.9%	75.6%	67.8%	66.1%	63.0%	72.4%	74.2%	79.9%
Yelp Reviews	0	77.5%	96.8%	94.5%	95.7%	95.0%	94.8%	95.0%	76.7%	87.2%
	1	92.7%	93.9%	97.2%	95.0%	95.7%	96.2%	96.8%	94.0%	96.4%
Zhang Amazon	0	80.4%	93.9%	91.8%	94.7%	94.5%	91.6%	90.6%	82.8%	83.4%
	1	82.0%	83.7%	89.2%	85.9%	86.8%	86.5%	88.9%	84.4%	84.9%
Zhang Yelp Reviews	0	64.3%	92.7%	86.9%	92.6%	90.9%	87.1%	88.3%	65.9%	73.5%
	1	90.9%	88.0%	94.4%	87.7%	90.1%	92.9%	93.4%	91.1%	94.4%

Figure 2: Model accuracies by dataset and sentiment

results highlight the importance of evaluating model appropriateness for various transfer learning tasks.

4.3 Yelp Review Dataset:

Our models achieved the highest accuracies on the Yelp reviews dataset, which averaged around 95% with a high of 95.9% from the CNN Word2Vec with USE and CNN GloVe with USE models. Once again, the CNN Glove model saw the most improvement when concatenating the USE outputs. Of note, concatenating the USE outputs to the CNN Glove model improved accuracy for positive sentiment by 3.51%, which translates to a 54.10% decrease in error rates. This comes at the cost of accurately predicting negative sentiment, which reduced by 2.38% or 71.88% increase in error rates.

4.4 Zhang Amazon Polarity and Yelp Polarity Datasets:

When evaluating the Amazon Polarity and Yelp Polarity datasets (Zhang et al, 2015), our best model (Glove + USE) achieved over 90.5% accuracy. In both datasets, the greatest improvement was seen in the model's ability to predict positive sentiment when concatenating the USE embeddings to our CNN models. In the Amazon Polarity dataset, positive sentiment accuracy went from 83.7% to 89.2%, which is a 6.57% increase in accuracy or 32.93% reduction in error rate. For the Yelp polarity dataset, we observed that concatenating USE to the CNN Glove model improved accuracy by 7.27% or a 53% reduction in error rate. Unfortunately, this improvement in predicting positive sentiment for Yelp reviews came at the cost of predicting negative sentiment. We observed that our model's ability to predict negative sentiment reduced by 6.26%, resulting in a 79.45% increase in error rates.

5 Error Analysis

Although all datasets did see an increase in overall accuracy when using the USE outputs, the magnitude of the improvement varied based on the model and dataset. For example, on IMDB review data the CNN GloVe saw an accuracy improvement of 3.2 percentage points when adding the USE, while for the same model our Yelp reviews only saw an uptick of 0.6 percentage points. When determining where the errors were coming from, we took the approach of identifying where the models differed the most in the sentiment scoring between using the USE and not using it. When focusing on the IMDB transfer learning task with our Word2Vec model, if the review had a positive sentiment for the movie it was reviewing, but contrasted that with a negative movie review, the USE tended to interpret the review as negative. For example, one review states the following, "why is it that 'B' movies like 'American Wedding' and 'Eurotrip' get widescreen and fullscreen releases, and often a special edition with multiple commentaries and extras, while great art pieces like 'The Dead' are all but forgotten?". The USE model has a hard time understanding this is actually a positive sentence in terms of the movie being reviewed. The rest of the review was fairly positive so the baseline Word2Vec model attributed a positive sentiment to this review while the Word2Vec + USE model attributed a negative sentiment. However, the Word2Vec + USE model proved valuable in handling other complex reviews where the baseline models failed. For example, a positive review for a movie with a depressing or tragic plot was generally predicted incorrectly by the baseline models. The models that included the USE however, were able to correctly interpret reviews stating "disaster has destroyed virtually all the population" and "repressive state-run reformatory school, where brutalization and humiliation…." as having positive sentiment. These correct predictions happened when USE was applied to both Word2Vec and GloVe models.

Figure A.1 and A.2 (Appendix) show the spread of our predictions for each sentiment type and dataset. For example, the negative boxplots for Yelp are all condensed in the 0 - 0.5 range showing that we have a low variability for our predictions of negative sentiment reviews. The twitter data has a large spread since our accuracy was lower and many of our predictions for negative labeled tweets were actually positive. When evaluating the actual reviews from the Sentiment140 dataset, it was difficult to tease out why some of the tweets were classified positive or negative. Another insight from this chart was IMDB reviews had fairly high variability when we were predicting on positive reviews but low variability on negative reviews. This was also confirmed in figure 2 and discussed in the results section. After exploring this issue, many of the positive IMDB reviews seem to be written by reviewers struggling to determine whether they enjoyed a movie or not enough to rate it positively. For example, reviews where the model struggled on predicting correct sentiment contained the following somewhat vague statements "one of those movies that you sit back and enjoy it for what it

is and what it is not" and "this is not a perfect movie by any means" and "too dull to be completely enjoyable". These vague statements cause the model to sometimes struggle with the true feelings of a reviewer. Further discussion on error analysis may be found in the Appendix.

6 Conclusion

Although our models trained strictly on Amazon product review data did not achieve state of the art accuracies cited in literature, we observed that concatenating review level USE embeddings alongside individual word embeddings improved the accuracies of our models relative to baseline in all transfer learning tasks. Based on our preliminary findings, it appears that the majority of the benefit in using the Universal Sentence Encoder comes from predicting positive sentiment. The greatest gains in all experimental model types (baseline + USE) were seen in our recall scores at the cost of precision. In the case of the IMDB transfer learning task, we found that the combination of the USE embeddings with the GloVe embeddings resulted in a 34% improvement in error rate for positive sentiment classification. These results may suggest that the Universal Sentence Encoder is good at capturing positive sentiment, but it may also suggest that the Universal Sentence Encoder helps the CNNs generalize their predictions better. Of note, we found that simply feeding review level embeddings from the USE into a dense neural network provided surprisingly accurate results. Although the USE improves accuracy of our models, it is apparent from literature results that we are far from achieving the best possible models in these zero-shot learning tasks. Discussions on future work may be found in the Appendix section.

Acknowledgments

We thank the W266 teaching staff for all of their guidance on this final project.

References

Bengio, Yoshua, et al. "A neural probabilistic language model." *Journal of machine learning research* 3.Feb (2003): 1137-1155.

Camacho-Collados, Jose, and Mohammad Taher Pilehvar. "On the role of text preprocessing in neural network architectures: An evaluation study on text categorization and sentiment analysis." arXiv preprint arXiv:1707.01780 (2017).

Cer, Daniel, et al. "Universal sentence encoder." arXiv preprint arXiv:1803.11175 (2018).

Go, Alec, Richa Bhayani, and Lei Huang. "Twitter sentiment classification using distant supervision." CS224N Project Report, Stanford 1.12 (2009): 2009.

Johnson, Rie, and Tong Zhang. "Convolutional neural networks for text categorization: Shallow word-level vs. deep character-level." *arXiv preprint arXiv:1609.00718* (2016).

Johnson, Rie, and Tong Zhang. "Effective use of word order for text categorization with convolutional neural networks." arXiv preprint arXiv:1412.1058 (2014).

Kim, Yoon. "Convolutional neural networks for sentence classification." arXiv preprint arXiv:1408.5882 (2014).

Lakkaraju, Himabindu, Julian McAuley, and Jure Leskovec. "What's in a name? understanding the interplay between titles, content, and communities in social media." Seventh International AAAI Conference on Weblogs and Social Media. 2013.

Maas, Andrew L., et al. "Learning word vectors for sentiment analysis." Proceedings of the 49th annual meeting of the association for computational linguistics: Human language technologies-volume 1. Association for Computational Linguistics, 2011.

Mikolov, Tomas, et al. "Distributed representations of words and phrases and their compositionality." Advances in neural information processing systems. 2013.

Pang, Bo, and Lillian Lee. "Opinion mining and sentiment analysis." Foundations and Trends® in Information Retrieval 2.1–2 (2008): 1-135.

Pennington, Jeffrey, Richard Socher, and Christopher Manning. "Glove: Global vectors for word representation." Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP). 2014.

Socher, Richard, et al. "Recursive deep models for semantic compositionality over a sentiment treebank." Proceedings of the 2013 conference on empirical methods in natural language processing. 2013.

Song, Xingyi, Johann Petrak, and Angus Roberts. "A Deep Neural Network Sentence Level Classification Method with Context Information." arXiv preprint arXiv:1809.00934 (2018).

Wang, Alex, et al. "Glue: A multi-task benchmark and analysis platform for natural language understanding." arXiv preprint arXiv:1804.07461(2018).

Yin, Wenpeng, and Hinrich Schütze. "Multichannel variable-size convolution for sentence classification." arXiv preprint arXiv:1603.04513(2016).

Zhang, Ye, and Byron Wallace. "A sensitivity analysis of (and practitioners' guide to) convolutional neural

networks for sentence classification." arXiv preprint arXiv:1510.03820(2015).

Zhang, Xiang, Junbo Zhao, and Yann LeCun. "Character-level convolutional networks for text classification." Advances in neural information processing systems. 2015.

7 Appendix

7.1 Error Analysis Continued:

One important piece that was not discussed in the results and error analysis section was the disparity between literature scores on the Amazon polarity benchmark and the scores we obtained from our models trained on Amazon data. This may be due to a number of reasons. For one, our data selection script favored reviews that had 1 and 5 stars over 2 and 4. It is possible that the Amazon Polarity dataset from Zhang et al. captures more subtle differences in sentiment due to the number of 2 and 4 star reviews they may have included in the test dataset. Another reason could be that nature of the reviews they selected. Our selection process was random so we may have included more "misclassified" or ambiguous reviews in our dataset, thereby introducing bias into our models. Unfortunately, we do not have data on the product domains or the true ratings (in the 1-5 scale) of the reviews in the Amazon Polarity dataset, so investigating this further will be challenging. One possible workaround would be to retrain all models on the Amazon Polarity train dataset.

We also focused on investigating cases where the baseline models substantially differed in their predictions from the baseline + USE models. More specifically, we looked into cases where all baseline models predicted a negative or positive sentiment, while the baseline + USE models unanimously predicted the complete opposite class. Although these cases were rare, more often than not, the models with the baseline + USE was correct. In the context of the IMDB dataset, we specifically observed that the baseline + USE models were getting the prediction correct for reviews that were longer in nature. When reviewing the text, much like what was stated in the earlier section, we find that positive sentiment is intertwined with negative sentiment. The reviewer writes "highly questionable recanting of historical texts", and follows that shortly with "He literally oozes charm and sex appeal from every pore that easily melts the heart of his loyal heroine". Although the overall sentiment is clearly positive, the baseline models predicted that it was very likely a negative sentiment. This could be due to a number of reasons. For one, the removal of stop words would like improve the probability of our baseline models, but it may not help entirely. The outputs of the Universal Sentence Encoder may be able to encode context better, and therefore help predict the overall sentiment better. For this reason, it is possible that including an attention aspect to the convolutional neural network would help improve our predicted scores.

Unfortunately, all of the zero-shot learning datasets were sparse in that we do not have any data with regards to the topic being discussed other than the review text and the sentiment. For this reason, we were left to investigate the differences in sentiment classification accuracy for different product domains in our test dataset. To our surprise, the accuracies were fairly consistent across all product domains. This may be due to our model being fairly general since we have only trained on 3 epochs.

7.2 Future Work and Other Considerations:

It is important to note that there are a number of improvements that can be made for our future work. For one, the training time of each model was substantial so we reduced the number of epochs to three. Although this provided fairly accurate results, it is likely that more training will improve the accuracies of our models. Also due to time constraints, we were unable to complete the development and testing of our CNN model that combines the max pooling layers for sentence level embeddings alongside individual word embeddings. This may prove to be a fairly accurate and robust model given the accuracy improvements we saw with review level embeddings. Another important observation to note is that we did not remove stopwords for the individual word embeddings in our CNN model. As mentioned previously, this may inherently affect the accuracies for the hard cases where the CNN was close to predicting the opposite sentiment. More will need to be done on the preprocessing of the data to properly test the effect of stopwords on models trained with more epochs. As mentioned in the Error Analysis Continued section, we would like to investigate the use of a Convolutional Neural Network with attention, and compare these results to our baseline models and experimental models.

Error Analysis

Distribution of Predicted Sentiment vs. Actual Sentiment

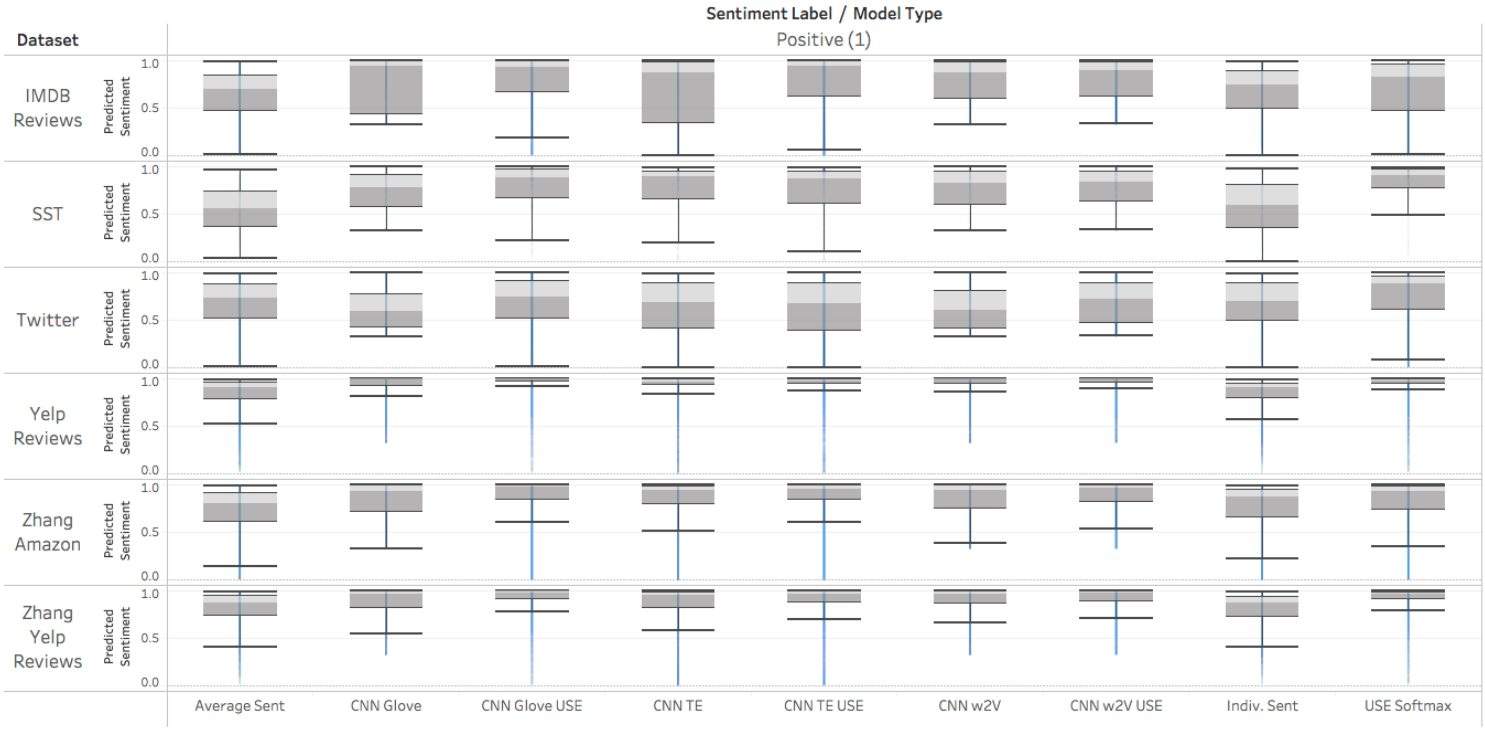

Figure A.1: Box plots of the distribution of predicted sentiment probability versus actual positive sentiment classes.

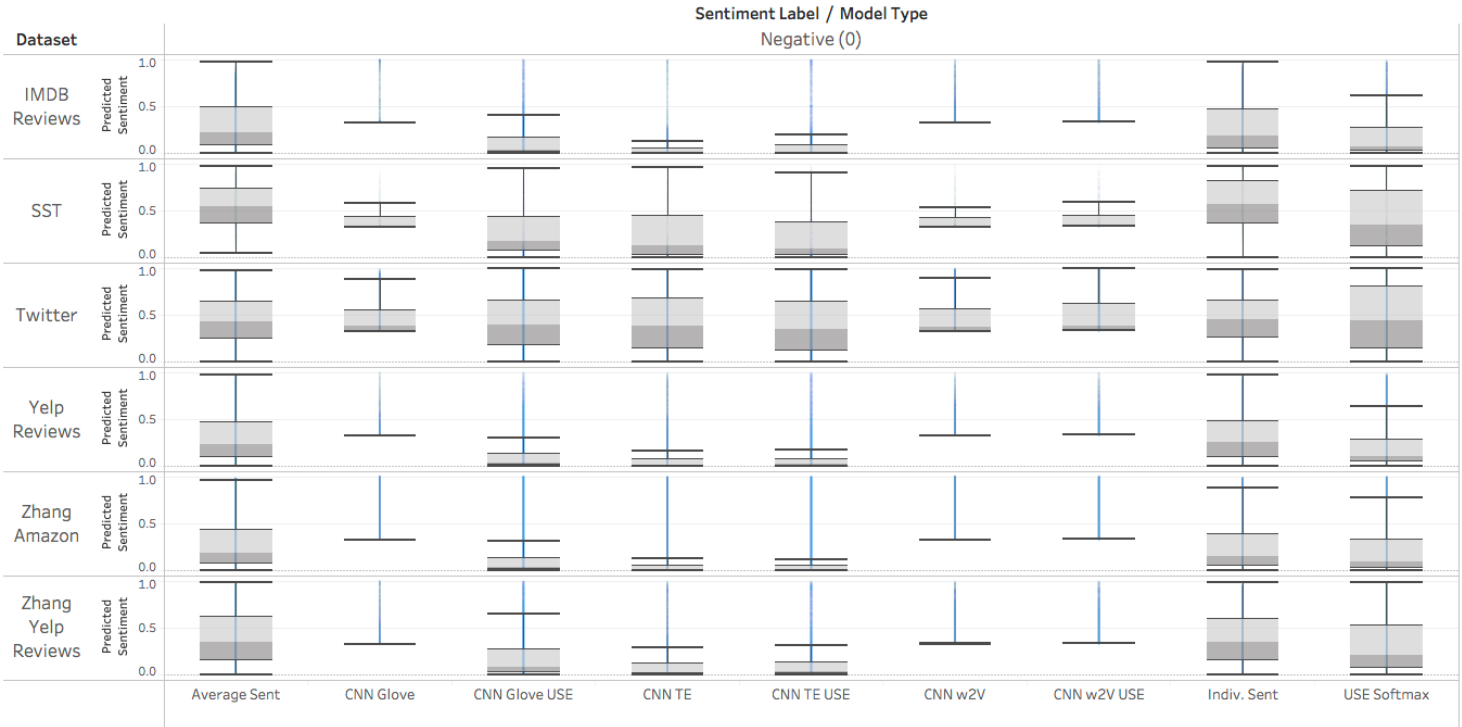

Figure A.2: Box plots of the distribution of predicted sentiment probability versus actual negative sentiment classes.