Multi-Class Text Sentiment Analysis Using Amazon Review Data

Motivation and Objective

Predicting a review's numeric rating based on the textual review is a quintessential multiclass text classification problem and an interesting research topic in natural language processing. New advances in NLP including the development of Glove, and Word2Vec, have increased the range of approaches available to address this question, and insights gained on this problem can generalize across sentiment analysis and NLP multiclass classification problems in general. We leveraged the extensive corpus of multi-class labeled Amazon data to apply sentiment analysis.

Objective: Given a text book review, predict one of the three (positive, neutral, negative) sentiment classes.

Data Description

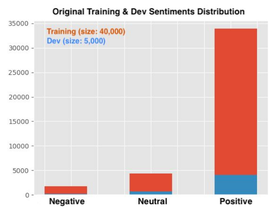
Dataset: 50,000 labeled Amazon Book reviews from 2018
(8:1:1 train, dev, test split)

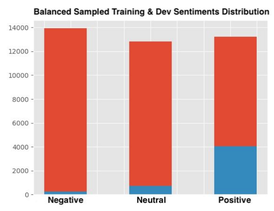
Example Text:
(max length per review= 50 words)
'Spiritually and mentally inspiring! A book that allows you to question your morals and will help you discover who you really are!'

Example Label:
5 (Rating between 1 … 5)

Original Label Modified to Fit Our Prediction Model:
{Negative = 1, Neutral = 3, Positive = 5}

Reviews rated 1, on a scale of 1 to 5, are relabeled as "negative," reviews scored 3 as "neutral," and reviews rated 5 as "positive."





In order to overcome skewness in the original data distribution, we applied undersampling on positive class and oversampling on negative and neutral classes.

Baseline Methods

Method	Test Accuracy
Textblob	0.646
Vader	0.741
Naive Bayes v.1	**0.815**
Naive Bayes v.2	0.764
Naive Bayes v.3	0.793

Confusion Matrix

true \ pred			
	121	66	54
	73	425	211
	147	374	3529

*Confusion matrix from Naive Bayes v.1

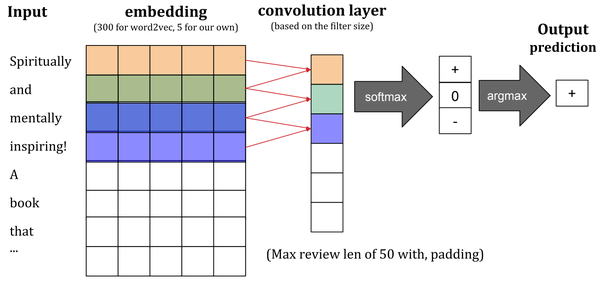
Neg	Neu	Pos
0.42	0.54	0.90

*v1: only with frequent words, filtered stop words
2: same as v1 on balanced sampled training set
3: balanced sampled training set without filter

Best indicator words per class (+, o, -)

awesome, beautifully, wonderful, loves, wait	*slower, fairly, wordy, somewhat, decent*	*waste, pointless, wasted, redeeming, puerile*

Text CNN Model



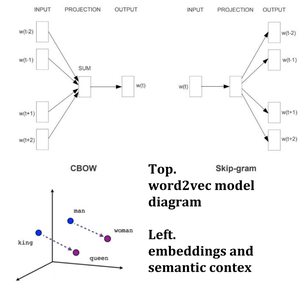
CNNs have been shown as effective text classifiers by Yoon Kim. Input features are a n-by-d matrix, where n is the max number of words and d is the dimensionality of a word embedding. Kernels are of size k-by-d and convolve input features within their range to form "n-gram" representations. The softmax function regularizes this layer, and we take the argmax of the three classes to obtain our predicted label.

Learned Embeddings

Used the pytorch.nn Embeddings module to map words to n-dimension. Unlike pretrained word2vec, these embeddings are trained concurrently with the CNN model.

Pretrained Embeddings

We used word2vec as a pretrained mapping of words to 300-dimension embeddings. The mappings are trained on Google's news corpus and utilizes both CBOW and Skip-gram methods to obtain vector representations of words.



Top. word2vec model diagram

Left. embeddings and semantic contex

Experiment & Results

Method	Balanced?**	Hyperparameter(s)	Accuracy
SVM	**n**	**RBF kernel**	**0.838**
KNN-5	**n**		**0.816**
QDA	**n**		**0.816**
MLP	**n**		**0.811**
Random Forest	n		0.795
Decision Tree	n		0.720
NN + 100-dim GloVe	**y**		**0.434***
NN + word2vec	y	L2 regularization, 0.01	0.692*
Text CNN + 5-dim	**y**	**learning rate = 0.01**	**0.486***
Text CNN + 55-dim	y	learning rate = 0.005	0.648*
Text CNN + 150-dim	y	learning rate = 0.005	0.675*
Text CNN + 55-dim	**y**	**kernel = 5, lr = 0.005**	**0.809***
Text CNN + word2vec	**y**	**kernel = 5, lr = 0.01**	**0.871***

Fig. t-SNE visualization of word2vec embeddings

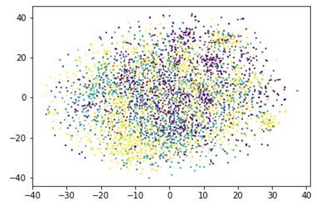


Fig. Confusion matrix, F1 scores for Text CNN

181	14	8
34	**302**	36
121	432	**3872**

Neg	Neu	Pos
0.887	0.831	0.889

*Highest accuracy from 500 epochs
**Best results from balanced vs. standard

ML Classifiers

First, we created vector representations of each token using word2vec and used its mean as the text representation. The right figure is a t-SNE visualization of the word2vec embeddings of texts, with different colors representing different classes. Different classes are mixed in some areas and are distinct in others, which makes sentiment classification based on ML classifiers difficult. The best performing classifier was SVM with RBF kernel, which achieved 83.8% test accuracy. Other methods. such as K-Neighbors, Quadratic Discriminant Analysis and MLP, also achieved comparable results, with all of the three classifiers reaching 81% accuracy.

Neural Network

As the next step, we experimented with a neural network architecture with three densely connected layers. The model was built on top of GloVe and word2vec text embeddings identical to those used in machine learning classifiers. Neural Network with GloVe embeddings achieved 43.4% accuracy and word2vec embeddings achieved 69.2% accuracy.

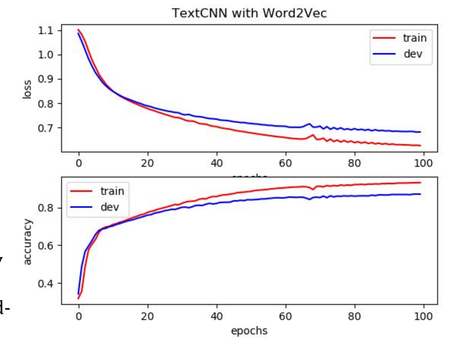


Fig. Training vs. Validation loss and accuracy

Text CNN

Next, we experimented with Text CNN architecture for sentiment classification. Two models based on Text CNN architecture were used with different text representations: learned embeddings and word2vec embeddings. Text CNN with 55-dim learned embeddings performed the best among learned embedding models, achieving 80.9% accuracy. Text CNN with word2vec embeddings performed the best overall, achieving state of the art 87.1% accuracy. The model was trained for 100 epochs and the learning curves in the above reveal no signs of overfitting. Both training and validation accuracy steadily increases; training and validation loss seems to be ideal, as both are decreasing over training. Left is the confusion matrix for test prediction and we can observe that, while the test data is also skewed towards more positive reviews, the model makes balanced predictions over all three classes. The F1 scores support this claim because the scores are over 0.8 across all three classes.

Discussion

This project showed that sentiment analysis based on the Text CNN model can work on the paragraph level, in addition to the sentence level. We addressed the challenge of skewed data by undersampling our training set after observing that baselines run on undersampled data yielded more desirable confusion matrices. We concluded that Text CNN used with word2vec embeddings achieved the best accuracy because the convolutional layers could predict the sentiment based on the relationship between adjacent words. However, the fundamental limitation of word2vec embeddings is its inability to capture the contextual meaning of a word, while words can have different meaning depending on its context. Also, our model was only trained and tested on the Amazon Book Reviews dataset, so it doesn't perform well on text from other domains; a possible extension of the architecture would be to apply the model to text from other domains and to observe its performance.

Future Work

- Develop finer-grained sentiment analysis models to classify text into all five review categories. Extend the model to domains other than Amazon Book Reviews.
- Explore methods of encoding reviews with arbitrary length, for example by aggregating the representations for all words beyond a specified length
- Experiment with Transformer based models such as BERT or XLNET