CS230

DeepNews: Detecting Quality in News

Eun Seo Jo, Aashiq Muhamed, Shashank Nuthakki, Ayush Singhania
{eunseo, aashiq, nuthakki, ayushs}@stanford.edu

Abstract

News in the internet-era has become more abundant but also increasingly unreliable. The motivation of this project is to automatically detect the quality of news that is produced in the millions every day. We apply several deep learning methods for binary classification - quality or commodity - of a given news article. We compare performance of feedforward CBOW and LSTM-RNN-feedforward mixture models and introduce a new LM classifier set-up. By using POS tags to mask out non-stop words, our approach focuses on writing quality and stylistics as to not bias our classification by topics and themes of providers. We find that deep learning models perform well for this task e.g. we get 94% prediction accuracy by using a simple CBOW feedforward neural network.

1 Introduction

While the abundance of news on the internet has increased the accessibility of news content, the ease of online media has also opened the floodgates for low quality sources of information including, what we have infamously coined 'fake news.' The direct business implication of this phenomenon is the correct matching of advertising. The current news ecosystem does a poor job of matching quality advertising with quality news articles, resulting in featuring outrageous advertisements on highly reputed sources such as the NYT. Our project aims to automatically detect the quality of news articles purely based on the quality of writing uninfluenced by topics. In other words, it is akin to an authorship attribution task where we define the author to be not just one person but entities of "quality" or "commodity" writers.

The input for our model is the processed text from news and from a wide distribution of news providers. Instead of training and testing on the raw text, we masked out topical words with their corresponding POS-tags so as to control for the topical bias (Section 3). On this input, we predict a simple binary output, $y \in \{1 : high, 0 : low\}$ of quality of the given article. Where high indicates value-added news (high-quality) and low indicates commodity news (low-quality). We apply and compare the performance of several methods – a centroid classification baseline, a feedfoward CBOW model, a LSTM-feedforward mixed model, and an experimental LM classification and scoring model. We find that deep learning models all perform with near 90% even with few epochs of training. This suggests also that the difference in style of writing between high and low quality sources of news is significant enough for machines to detect.

2 Related work

Our work builds on literature on the latest improvements in NN-based Sequence Models [5, 6, 7, 8, 13]. Automated essay scoring is a very well researched task which is related to the problem at hand. Traditionally, this task is solved by using many engineered features along with feature based classifiers [2, 3, 10]. There are recent works which use NN-based sequence models for essay scoring [1, 12].

CS230: Deep Learning, Winter 2018, Stanford University, CA. (LateX template borrowed from NIPS 2017.)

Though these works are related, we found they cannot be readily extended to our task, e.g., essay scoring models are designed to give high weights to spelling, correct grammar, sentence coherence etc. which are not very useful for detecting quality across value-added and commodity news sources as even commodity news sources seldom have incorrect language. In this task we are trying to capture subtle stylistic differences between value-added and commodity news articles.

3 Dataset and Features

Table 4 (in appendix) shows our distribution of data across different sources and counts. From this data we extracted two buckets - one of high quality news and another of low quality news. Each bucket was from a uniform distribution of sources which ensures low bias towards a particular source. As human labeling is expensive, we used the reputation of the news provider as the indicator of each article's quality. A news provider's reputation is subjective but we relied on the insight of an experienced journalist, our project advisor Frederic Filloux. While this closely adhered to news quality labelled by journalists on Mechanical Turk, the ground truth can be further refined with more human labelled data.

We processed about 3 million articles that came directly from publisher sources. The sources came in various formats that required processing.[1] For transfer learning, we used 100 dimensional GloVe embeddings [9] as initial word representations. These embeddings were learned further in our training process.

We ran preliminary experiments using both raw text and Part-Of-Speech tagged text [11] (see Figure 6). We mask out non-function words and punctuation with POS tags to remove the topical words which can bias an article to a specific source that may have a focus theme ie. The Economist. It also eliminates the bias of certain topics being correlated with quality such as sports with lower quality sources. The comparison between plain and POS tagged text was performed by training a feedforward CBOW model (Section 4) using a subset of sources then testing the trained model with a set of unseen articles. This gives a good estimate of the generalizing ability of the trained models. We found that the model, trained with POS tagged text showed significant reduction in bias (Figure 7), and performed better on the dev-test set. This observation motivated us to use POS tagged text as our input data to all our deep learning models. An additional advantage of using POS masked text is that it is faster to train and requires smaller memory (especially the Language Model in Section 4) because we can reduce the size of the model $\sim$ 600-fold.

Due two computational limitations and the constraint of working with a balanced dataset, we worked with a subset of our entire dataset for this project. We worked with a train-dev-test split of { 600k, 10k, 10k} examples. For future work we will continue to expand this dataset.

4 Methods

We use the nearest centroid classifier as our baseline. Each document is represented as the CBOW of the embeddings that correspond to the tokens in it. We calculate the L^2 distance of each class from the quality and commodity labeled data centroids. The nearest centroid becomes the label for each test article's CBOW representation. The baseline was able to classify with 70.32% accuracy which gives an indication of the tractability of the task at hand.

4.1 Feedforward NN with CBOW Inputs

The feedforward network we use is a 3 hidden layer (1000, 500 and 100 hidden units respectively with ReLU activation) with a sigmoid activation at the output layer. The input layer is 100 dimensional vector representing a single article by CBOW method where we represent each document (d_{cbow}) by the average of its word vectors. $d_{cbow} = \frac{1}{V}\sum_{w \in d} w_{vector}$ where V = total word count and w_{vector} = GloVe vector corresponding to word w. Both w_{vector} and d_{cbow} are of dimension {1, 100}. We used logistic loss as the loss function. The hyperparameters we tuned for this model are #layers, #hidden units, learning rate, different optimizers, dropout rate.

[1] We developed the pipeline to extract data from different format to text files. A small random sample from each source of data was manually studied to find any pattern in format which might bias the model. For instance, all articles from Huffington Post had a sentence repeated in the end of all articles.

4.2 RNN with LSTM units

Our LSTM classifier model is 2 layer LSTM model of $T = 50$ time-steps. Each LSTM cell had a hidden size of 250. The output from the second layer is treated as an encoding of the text and passed through a feedforward with 2 hidden layers of 200 hidden units each, followed by a sigmoid output layer. This RNN is trained to predict the quality of the given sequence of 50 token chunks of articles. The RNN uses dropout for regularization and gradient clipping at 5.0 to prevent exploding gradients. The Hyperparameters we tuned for this model are #layers, LSTM hidden size, learning rate, unroll depth (#time steps), dropout rate. The input to the model is a (batch-size $\times$ time-steps $\times$ embedding-depth) tensor. We also demonstrate that an RNN without a feedforward classifer and one layer performed worse than a plain 2 layer RNN. For training we use Logistic loss with Adam optimizer along with gradient clipping.

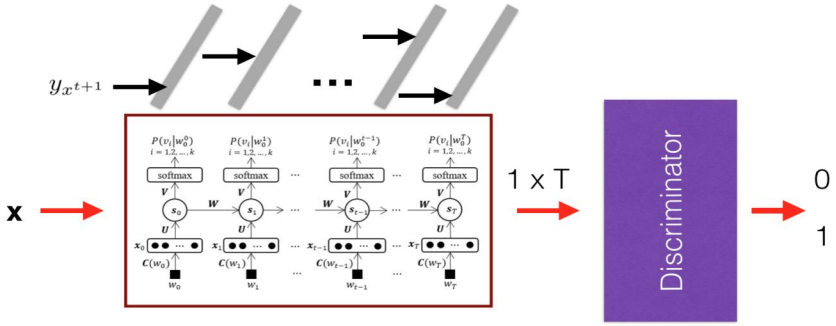

Figure 1: Language Model Classifier

4.3 Language Model

We used a one layer LSTM Language Model (LM) [4, 7] with 1000 hidden units per cell, to learn sequences of length $T = 50$ from high quality sources. We used dropout, Adam optimizer for learning along with gradient clipping. From the output of each time step, we select the softmax probability corresponding to the next word and generate encodings of size 1 x T per example. The trained LM is used to get these encodings which are fed to a feedforward discriminator model during the forward prop. The discriminator is a 2 hidden layer NN with 200, 200 unit hidden sizes, followed by a sigmoid unit. The intuition behind this setup, is that a LM trained on high-quality sources will be good (lower perplexity values) at predicting sequences from high-quality texts, while it will be poor (higher perplexity values) on low-quality sequences. The perplexity scores obtained from the LM is a less data-intensive way to assign a quality score to an article (lower perplexity scores mean higher the quality rating).

5 Experiments/Results/Discussion

We first tested whether masking out the POS tags would have a positive effect on the performance of our models on unseen data. We found that the model, trained with POS-tagged text showed significant reduction in bias (Figure 7, Appendix), and performed better on the test set containing articles from unseen providers. These results indicate that POS masking was able to overcome topical bias in our dataset that might have hindered the performance of the raw text by overfitting on the topical words. In other words, the POS masking served as a regulator.

The feedforward CBOW was the simplest and fastest deep learning model. After random search over the hyperparameters we found that the combination of the Adam optimizer, learning rate = 0.0001, batch size 256, no dropout, along with the network parameters as discussed in Section 4.1 gave us consistently good results. We find that the feedforward model gives surprisingly high results of 94% accuracy. At this point, our model has not been trained to learn sequential information nor the length of documents. This indicates that there is enough distinction in token frequency distribution

	Truth = 1	Truth = 0
Prediction = 1	4081	326
Prediction = 0	265	5258

Table 1: Confusion Matrix for the Classification Task with ≈10k examples

Output FF layer	#layers	#time-steps	Dev accuracy
Present	2	25	82.13
Present	2	50	89.38
Present	2	75	86.89
Not Present	2	50	79.57
Present	1	50	87.50

Table 2: Performance of various LSTM sequence networks

between news that comes from high and low quality sources (Figure 10 shows a quick sampled relative frequency distribution of token types between the two classes of news). We found that higher quality news tends to use more proper nouns, such as place names or people names, and lower quality news tends to use more adjectives and nouns, perhaps as a way of sensationalizing stories. The confusion matrix for this model is given in Table. 1, we observe that there is no significant bias towards false negatives or positives.

The hyperparameters that worked well for the LSTM classifier (discussed in Section 4.2) are Adam optimizer, learning rate 0.001 (see appendix Figure 9), dropout keep probability as 0.8 (at both inputs and outputs of an LSTM cell). We also experimented with varying network designs as briefly summarized in Table 2. The two main things which significantly increased our model's performance is using 50 time-steps along with a feedforward layer at the output. We observed that having 50-time steps provided the LSTM network with more context to work with, and at the same time was not very hard to train. A prediction on a new article is made by chunking the article into 50 sized sequences and taking the majority vote of the predictions on all sequences. Due to computational and time constraints we were not able to train the sequence model for long times (see epochs columns in Table 3), and were not able to experiment with large hidden sizes, more LSTM layers. Hence our final Test accuracy by using the best sequence model was only 90.19% which is less than what we obtained for the feedforward model.

The hyperparameters we used to train our LM are learning rate 0.001, Adam optimizer and with no dropout. As LM takes a lot of time to train, we didn't a get a chance to work with the entire dataset and search the hyperparameter space to a good extent or use multi-layer LSTM networks. Nevertheless, it still does a decent job in learning the high-quality sentences as we get nearly 30% accuracy (Figure 4,5) i.e. it predicts almost one-third of the 50 POS tagged tokens correctly. Because we trained our LM only on language from news from high quality sources, we expected our LM to give lower perplexity scores for inputs with high quality and higher perplexity for inputs with low quality. Indeed, we observed a stark perplexity difference between high and low quality dev sets as seen in Figure 4. The discriminator feedforward model leveraged this difference for the classification task and attained an accuracy of 81% which is good but not better than the other models. We think given more time to train with bigger models the LM classifier can do better.

Model	Test Accuracy	Epochs	Hours Trained
Classification by Centroid	70.32	NA	<1
Feedfoward CBOW Model	94.05	100	<1
LSTM Model	90.17	15	8-9
Language Model Classifier	81.85	8	9-10

Table 3: Performance Comparison across Models

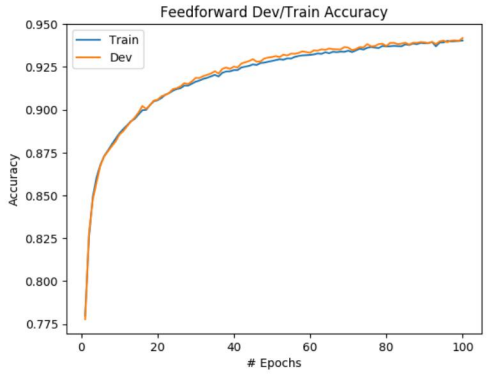

Figure 2: Feedforward Accuracy

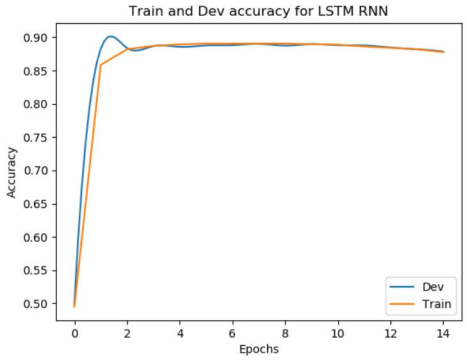

Figure 3: LSTM Accuracy

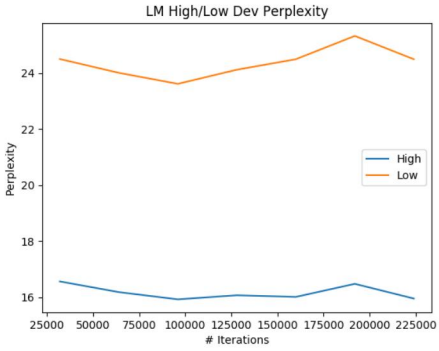

Figure 4: LM Dev Perplexity of High/Low Quality News

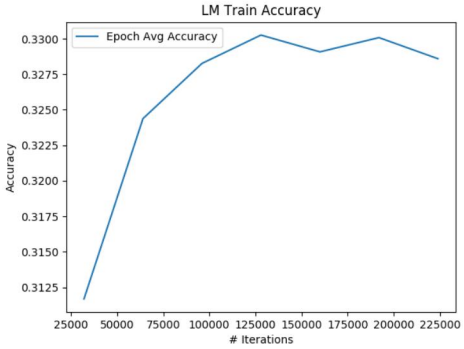

Figure 5: LM Train Accuracy

6 Conclusion/Future Work

- We observed that when trained over many epochs, the simple CBOW feedforward network representing global information performed better than sequence-based models, showing how this encoding is a strong indicator for this classification task. Computational and time constraints prevented us from training the sequential RNNs for as many epochs as the feedforward model (100 epochs). The high accuracy achieved by the feedforward CBOW model seems to suggest that a model that uses an input state vector incorporating both local sequential information (encoding from RNN) and a CBOW encoding should perform with higher accuracy than our current models.
- The relatively poor accuracy of the LM can be attributed to training only on a subset of the data and for a few epochs. The stark difference in perplexity seen between the two articles is a sign that this encodes information about style and can in future be used to score articles (1-5 scale) for quality, based on the perplexity scores. This is a promising aspect of using a LM for this task.
- That even our simplest deep learning model can correctly distinguish the quality of news from just the sequence of function words and POS tags with high accuracy suggests that the difference in writing quality among news sources is stark. This gives us hope that there is much fruitful work to be done in news and media stylometry.
- We also wish to explore how our method compares with other deep learning based models such as as an SVM with bigram inputs

7 Contributions

All members contributed equally to this project.

References

[1] D. Alikaniotis, H. Yannakoudakis, and M. Rei. Automatic text scoring using neural networks. *CoRR*, 2016.

[2] Y. Attali and J. Burstein. Automated essay scoring with e-rater v.2.0. *Journal of Technology, Learning, and Assessment*, 2006.

[3] B. Beigman Klebanov and M. Flor. Word association profiles and their use for automated scoring of essays. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1148–1158. Association for Computational Linguistics, 2013.

[4] C. Chelba. Language modeling in the era of abundant data, 2017.

[5] C. Chelba, M. Norouzi, and S. Bengio. N-gram language modeling using recurrent neural network estimation. Technical report, Google, 2017.

[6] Y. Goldberg. A primer on neural network models for natural language processing. *CoRR*, 2015.

[7] R. Jozefowicz, O. Vinyals, M. Schuster, N. Shazeer, and Y. Wu. Exploring the limits of language modeling, 2016.

[8] V. Kuznetsov, H. Liao, M. Mohri, M. Riley, and B. Roark. Learning n-gram language models from uncertain data. In *Interspeech*, 2016.

[9] J. Pennington, R. Socher, and C. D. Manning. Glove: Global vectors for word representation. In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, 2014.

[10] M. Russell, L. Rudner, L. M. Rudner, L. M. Rudner, T. Liang, T. Liang, and T. Liang. Automated essay scoring using bayes' theorum. *The Journal of Technology, Learning and Assessment*, (1):3–21, 2002.

[11] M. P. Shane Bergsma and D. Yarowsky. Stylometric analysis of scientific articles. In *In Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. Association for Computational Linguistics.*, pages 327–337, June 2012.

[12] K. Taghipour and H. T. Ng. A neural approach to automated essay scoring. In *EMNLP*, 2016.

[13] T. Young, D. Hazarika, S. Poria, and E. Cambria. Recent trends in deep learning based natural language processing. *CoRR*, 2017.

A Appendix

News Provider	#Articles (1000s)	Label
The Guardian	112	High
The Economist	91	High
Reuters	800	High
Financial Times	1000	High
Quartz	800	High
Huffington Post	116	Low
Business Insider	129	Low
CRN	122	Low

Table 4: News Data Collection

THE way that black holes bend light's path through space cannot be smoothed out by human ingenuity. By contrast, a vast distortion in the world economy is wholly man-made. It is the subsidy that governments give to debt. Half the rich world's governments allow their citizens to deduct the interest payments on mortgages from their taxable income; almost all countries allow firms to write off payments on their borrowing against taxable earnings. It sounds prosaic, but the cost and the harm is immense. In 2007, before the financial crisis led to the slashing of interest rates, the annual value of the forgone tax revenues in Europe was around 3% of GDP or $510 billion and in America almost 5% of GDP or $725 billion (see Briefing). That means governments on both sides of the Atlantic were spending more on cheapening the cost of debt than on defence. Even today, with interest rates close to zero, America's debt subsidies cost the federal government over 2% of GDP as much as it spends on all its policies to help the poor. This hardly begins to capture the full damage, which is aggravated by the behaviour the tax breaks encourage….

DT way that JJ NNS VBP NN POS NN through NN can not be VBN out by JJ NN . IN NN , a JJ NN in the world NN is RB JJ . PRP is the NN that NNS give to NN . PDT the JJ world POS NNS VBP their NNS to VB the NN NNS on NNS from their JJ NN : almost all NNS VBP NNS to VB off NNS on their NN against JJ NNS . PRP VBZ JJ , but the NN and the NN is JJ . IN CD , before the JJ NN VBD to the NN of NN NNS , the JJ value of the NN NN NNS in NNP was around CD NN of NNP or $ CD CD and in NNP almost CD NN of NNP or $ CD CD (see NNP) . DT means NNS on both NNS of the NNP were VBG more on VBG the NN of NN than on NN . RB NN , with NN NNS RB to zero , NNP POS NN NNS VBP the JJ NN over CD NN of NNP as much as it VBZ on all its NNS to VB the JJ . DT hardly begins to VB the JJ NN , which is VBN by the NN the NN NN NN . NNS VBP more to VB NN than they otherwise would , VBG NN NNS and VBG NN in JJ NN instead of in NNS that VBP NN . DT NN NNS are largely VBN by the JJ , VBG NN . JJ JJ NNS are VBN by VBG the NN NN on NN instead of the needs of the JJ NN . NNP has many JJ NNS VBG NNS to VB and NNS to VB NN from NN POS NN . CC the NN NNS have VBN the NN in a JJ NN . PRP have VBN a JJ NN that is NN to NNS and VBN against JJ NN : they have VBN JJ NN and VBN NN . PRP are a JJ NN and they need to be …

Figure 6: Raw text vs POS-tagged text

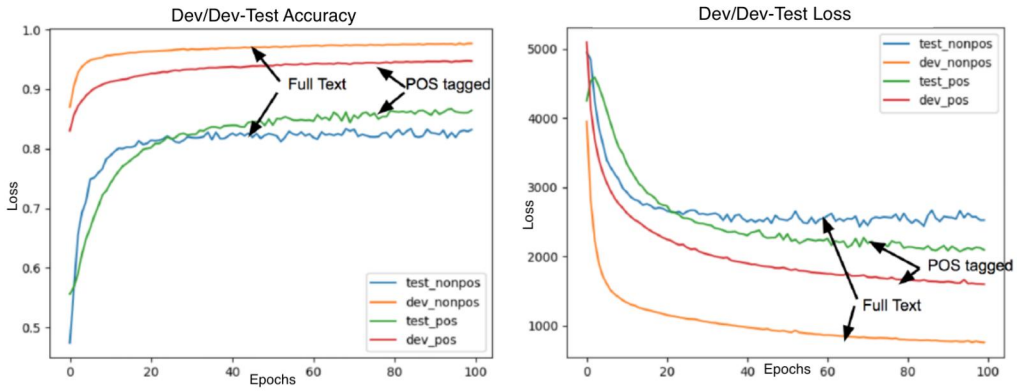

Figure 7: POS-tagged/not tagged difference in performance on feedforward CBOW

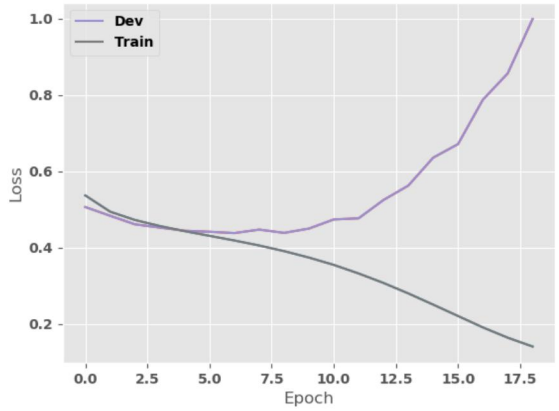

Figure 8: Loss curve for LSTM-RNN

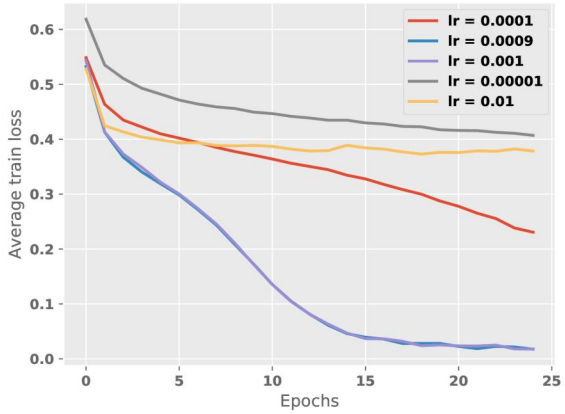

Figure 9: Hyperparameter tuning of learning rate for LSTM sequence model

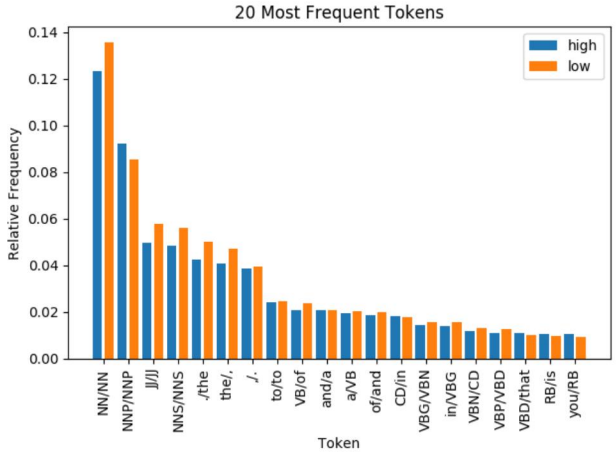

Figure 10: 20 Most Frequent Tokens for High and Low (High/Low) Quality Sources of Writing