Fake News Detection Using Neural Network Language Models

Christopher Farrer
farrerc@wwu.edu

Maxwell Schultz
schultm8@wwu.edu

Alex Sitzman
sitzmaa@wwu.edu

Abstract

This project aims to classify news article titles as sarcastic or non-sarcastic based on their content and context. Initially focused on sarcasm detection, the project pivoted to fake news detection due to dataset limitations. We used datasets from Kaggle and implemented multiple models, including Neural Bag of Words, Recurrent Neural Networks (RNNs), and Transformers. Our findings reveal that while Transformer models exhibit high accuracy, they struggle with nuanced contexts such as satire, and result in longer training times over other models. This report details our methodology, experimental results, and future directions for improving sarcasm detection.

1 Introduction

Neural Networks have become pivotal in various NLP tasks due to their ability to learn complex patterns from large datasets. This project aimed to leverage neural networks and traditional machine learning models to detect sarcasm in news headlines. Sarcasm detection holds significant ethical benefits, as it can improve the understanding and moderation of online communication, reducing misunderstandings and promoting healthier online interactions. However, limitations in the dataset necessitated a pivot to fake news detection. Fake news detection carries ethical benefits as well, as it helps combat misinformation, which can influence public opinion and decision-making processes. By ensuring the authenticity of news articles, such systems contribute to a more informed and rational public discourse. This paper discusses our approach, experimental setup, results, and the implications of our findings.

2 Related Work

Fake News Detection The proliferation of fake news has become a major concern, impacting political, economic, and social landscapes. (Agarwal et al., 2019) conducted a comprehensive study on fake news detection using a combination of NLP techniques and machine learning classifiers. They employed methods such as bag-of-words, n-grams, count vectorizer, and TF-IDF, training their dataset on five different classifiers. Their findings indicated that precision, recall, and F1 scores are critical metrics in determining the most effective model for fake news detection .

Sarcasm Detection Sarcasm detection poses unique challenges due to its inherently ambiguous nature. (Jain et al., 2017) explored sarcasm detection in tweets by utilizing ensemble-based approaches, specifically a voted ensemble classifier and a random forest classifier. Their research highlighted the difficulty in detecting sarcasm due to the evolving nature of language and the use of emoticons, which can alter the polarity of text. Their methodology diverged from traditional sentiment analysis by focusing on the presence of positive sentiment attached to negative situations, thus improving the detection accuracy in social media contexts .

Similarly, (Băroiu and Ștefan Trăușan-Matu, 2022) provided a systematic literature review on automatic sarcasm detection, tracing its evolution from 2010 to the present. They emphasized the growing popularity of multi-modal approaches and transformer-based architectures in recent years. Their work not only critiqued past research but also proposed future directions, underscoring the necessity for advanced models capable of handling the complexities of sarcasm in natural language .

3 Approach

Our approach utilizes the PyTorch library to implement and train various sequence classification models for sentiment analysis, specifically targeting movie reviews. The implementation details and models are based on the tutorials provided by (Trevett, 2023), which offer a comprehensive guide on using PyTorch for sentiment analysis tasks. Below, we outline the main steps and models used in our approach.

3.1 Models and Tutorials

3.1.1 Neural Bag of Words:

This tutorial introduces the basic workflow of a sequence classification project using a neural bag-of-words model. It covers data loading and preprocessing using the datasets and torchtext libraries. The model is simple yet effective for understanding the foundational concepts of sequence classification.

3.1.2 Recurrent Neural Networks (RNN):

Building on the basic workflow, this tutorial focuses on improving the model's performance by switching to a recurrent neural network (RNN) model. Specifically, it implements a long short-term memory (LSTM) RNN, which is one of the most commonly used variants of RNNs due to its ability to handle long-range dependencies in sequential data.

3.1.3 Transformers:

The final tutorial demonstrates how to use the transformers library to load a pre-trained transformer model, specifically BERT (Bidirectional Encoder Representations from Transformers). BERT, introduced in the paper by Devlin et al., provides high performance for various NLP tasks, including sequence classification. The tutorial covers loading the pre-trained BERT model and fine-tuning it for sentiment analysis.

3.2 Datasets

- News articles tagged by sarcasm sentiment (Misra, 2019).
- Fake news classification dataset (Shahane, 2024).
  Note: This dataset required some white space culling and the removal of non-English characters to be used.

4 Experiments

4.1 Preprocessing

The Fake News Classification dataset included many blank space, white space and in some cases Arabic characters which our models were not configured to handle. The set had to be culled of around 500 rows to be usable

4.2 Neural Bag Of Words

4.3 Recurrent Neural Network

4.4 Transformer

4.5 Footnotes

Footnotes are inserted with the `\footnote` command.[1]

4.6 Tables and figures

See Table **??** for an example of a table and its caption. **Do not override the default caption sizes.**

4.7 Hyperlinks

Users of older versions of LaTeX may encounter the following error during compilation:

```
\pdfendlink ended up in different
nesting level than \pdfstartlink.
```

This happens when pdfLaTeX is used and a citation splits across a page boundary. The best way to fix this is to upgrade LaTeX to 2018-12-01 or later.

4.8 References

The LaTeX and BibTeX style files provided roughly follow the American Psychological Association format. If your own bib file is named `custom.bib`, then placing the following before any appendices in your LaTeX file will generate the references section for you:

```
\bibliographystyle{acl_natbib}
\bibliography{custom}
```

You can obtain the complete ACL Anthology as a BibTeX file from `https://aclweb.org/anthology/anthology.bib.gz`. To include both the Anthology and your own .bib file, use the following instead of the above.

```
\bibliographystyle{acl_natbib}
\bibliography{anthology,custom}
```

Please see Section 5 for information on preparing BibTeX files.

4.9 Appendices

Use `\appendix` before any appendix section to switch the section numbering over to letters. See Appendix A for an example.

[1] This is a footnote.

5 BibTEX Files

Unicode cannot be used in BibTEX entries, and some ways of typing special characters can disrupt BibTEX's alphabetization. The recommended way of typing special characters is shown in Table **??**.

Please ensure that BibTEX records contain DOIs or URLs when possible, and for all the ACL materials that you reference. Use the `doi` field for DOIs and the `url` field for URLs. If a BibTEX entry has a URL or DOI field, the paper title in the references section will appear as a hyperlink to the paper, using the hyperref LATEX package.

Limitations

ACL 2023 requires all submissions to have a section titled "Limitations", for discussing the limitations of the paper as a complement to the discussion of strengths in the main text. This section should occur after the conclusion, but before the references. It will not count towards the page limit. The discussion of limitations is mandatory. Papers without a limitation section will be desk-rejected without review.

While we are open to different types of limitations, just mentioning that a set of results have been shown for English only probably does not reflect what we expect. Mentioning that the method works mostly for languages with limited morphology, like English, is a much better alternative. In addition, limitations such as low scalability to long text, the requirement of large GPU resources, or other things that inspire crucial further investigation are welcome.

Ethics Statement

Scientific work published at ACL 2023 must comply with the ACL Ethics Policy.[2] We encourage all authors to include an explicit ethics statement on the broader impact of the work, or other ethical considerations after the conclusion but before the references. The ethics statement will not count toward the page limit (8 pages for long, 4 pages for short papers).

Acknowledgements

This document has been adapted by Jordan Boyd-Graber, Naoaki Okazaki, Anna Rogers from the style files used for earlier ACL, EMNLP and NAACL proceedings, including those for EACL 2023 by Isabelle Augenstein and Andreas Vlachos, EMNLP 2022 by Yue Zhang, Ryan Cotterell and Lea Frermann, ACL 2020 by Steven Bethard, Ryan Cotterell and Rui Yan, ACL 2019 by Douwe Kiela and Ivan Vulić, NAACL 2019 by Stephanie Lukin and Alla Roskovskaya, ACL 2018 by Shay Cohen, Kevin Gimpel, and Wei Lu, NAACL 2018 by Margaret Mitchell and Stephanie Lukin, BibTEX suggestions for (NA)ACL 2017/2018 from Jason Eisner, ACL 2017 by Dan Gildea and Min-Yen Kan, NAACL 2017 by Margaret Mitchell, ACL 2012 by Maggie Li and Michael White, ACL 2010 by Jing-Shin Chang and Philipp Koehn, ACL 2008 by Johanna D. Moore, Simone Teufel, James Allan, and Sadaoki Furui, ACL 2005 by Hwee Tou Ng and Kemal Oflazer, ACL 2002 by Eugene Charniak and Dekang Lin, and earlier ACL and EACL formats written by several people, including John Chen, Henry S. Thompson and Donald Walker. Additional elements were taken from the formatting instructions of the *International Joint Conference on Artificial Intelligence* and the *Conference on Computer Vision and Pattern Recognition*.

References

Vasu Agarwal, H. Parveen Sultana, Srijan Malhotra, and Amitrajit Sarkar. 2019. Analysis of classifiers for fake news detection. In *International Conference on Recent Trends in Advanced Computing 2019 (ICRTAC 2019)*. Elsevier.

Alexandru-Costin Băroiu and Ștefan Trăușan-Matu. 2022. Automatic sarcasm detection: Systematic literature review. *Information*, 13(8):399.

Tanya Jain, Nilesh Agrawal, Garima Goyal, and Niyati Aggrawal. 2017. Sarcasm detection of tweets: A comparative study. In *2017 Tenth International Conference on Contemporary Computing (IC3)*, Noida, India. IEEE.

Rishabh Misra. 2019. News headlines dataset for sarcasm detection. Kaggle. Updated 5 years ago.

Saurabh Shahane. 2024. Fake news classification on welfake dataset. Kaggle.

Ben Trevett. 2023. Pytorch sentiment analysis. GitHub repository.

A Example Appendix

This is a section in the appendix.

[2] `https://www.aclweb.org/portal/content/acl-code-ethics`