DEEP LEARNING FOR BIAS DETECTION ON THE ENGLISH WIKIPEDIA

By

AMANDA LYNN BERTSCH

A Thesis Submitted to The Honors College

In Partial Fulfillment of the Bachelors degree
With Honors in

Computer Science

THE UNIVERSITY OF ARIZONA

M A Y 2 0 2 1

Approved by:

Dr. Steven Bethard
School of Information

Deep Learning for Bias Detection on the English Wikipedia

Amanda Bertsch

University of Arizona

abertsch@email.arizona.edu

Abstract

On Wikipedia, an online crowdsourced encyclopedia, volunteers enforce the encyclopedia's editorial policies. Wikipedia's detailed policy on maintaining a neutral point of view has made the project a popular target for NLP researchers working on bias detection or sentiment analysis. Often, this work focuses on a particular category of bias that Wikipedia identifies; while "weasel words" and "hedges" have both received significant attention, little work has been done on identifying "peacock phrases," phrases that are overly positive without a verifiable source. In this work, we present a model for identifying peacock phrases that achieves a 0.963 f1 score. We also discuss the general issues inherent in building a dataset from Wikipedia, with this project as a case study. Finally, we demonstrate a way to use Wikipedia's public infrastructure to host a tool that uses the trained model to give back to the Wikipedia editor community.

1 Introduction

As one of the world's largest crowdsourced knowledge bases, Wikipedia has strict community guidelines to maintain content quality. One guideline, called the Neutral Point of View policy (Wikipedia, 2021g), outlines best practices for maintaining the encyclopedia impartiality that Wikipedia strives toward. However, with over six million articles on the English Wikipedia alone (Wikipedia, 2021i), no paid or volunteer moderation team can review each edit to each article before it is made.

When Wikipedia started in 2001 (Wikipedia, 2021a), the project had far fewer edits per day. As Wikipedia has grown beyond the moderation capabilities of its volunteer administrators, technology has supplemented some moderation actions, including detecting vandalism and correcting spelling. However, more subjective issues such as maintaining neutral point of view have been left entirely to human editors. While this approach has its advantages, it also means that pages with less views may have lower quality and more issues than pages that many editors frequent. As natural language processing improves and the size of Wikipedia continues to grow, machine learning models have an increasingly large role in maintaining editorial standards.

2 Background and Motivation

Wikipedia is a crowdsourced digital encyclopedia, with different versions available in 321 languages (Wikipedia, 2021b). The English Wikipedia is the largest, with 6.2 million articles and over 3.88 billion words as of April 2021 (Wikipedia, 2021i). Anyone can edit Wikipedia by clicking the "Edit" tab at the top of an article; some protections exist on the most viewed pages, but according to Wikipedia the "vast majority" (Wikipedia, 2021h) have no such requirement.

This freedom to edit inspires vandalism, and Wikipedia has gone to great lengths to build tools to protect against vandalistic edits. The ORES project is a suite of natural language processing tools for detecting vandalism, developed by the Wikimedia Foundation, which maintains Wikipedia. Tools based on ORES can, when the model is sufficiently confident that an edit is vandalism, automatically revert that edit without human intervention; in many other cases, tools acts as report systems to flag edits that may be vandalising for human editors to review (Wikimedia, 2021).

While the Wikipedia community has invested significant effort in anti-vandalism tools to target the most disruptive edits, the vast majority of non-standard edits on Wikipedia are not malicious. These edits, generally by new editors or editors without accounts, violate Wikipedia's policies unintentionally or unknowingly; it is not a requirement to read the editorial policies before editing. Senior editors and volunteer administrators spend significant amounts of time welcoming new editors and reverting these bad edits, but despite this many bad edits on low-traffic pages are left up for months, years, or, in the case of some pages, over a decade.

One common issue with edits is editorial bias. Prior research on identifying biased edits has focused on two categories of bias that Wikipedia identifies: weasel words, which Wikipedia defines as "words or phrases aimed at creating an impression that something specific and meaningful has been said, when in fact only a vague or ambiguous claim has been communicated" (Wikipedia, 2021e), and hedges, which Farkas et. al. define as phrases "indicating that authors do not or cannot back up their opinions/statements with facts" (2010). This work includes the CoNLL-2010 ACL shared task on weasel words, where Georgescul notably achieved the highest f1 score (0.602) using a bag-of-words classification approach (Farkas et al., 2010).

Submissions to the yearly Wiki Workshop at The Web Conference have also considered identifying all types of bias with a single model, using approaches including combining user metadata and machine learning models. Most recently, Hube and Fetahu achieved an f1 score of 0.69 using a generated bias word list and other hand-picked features, including part of speech tags and a context window around each

bias word (2018). However, to the best of our knowledge, no paper before now has focused specifically on the identification of peacock phrases, which Wikipedia defines as words "used without attribution to promote the subject of an article, while neither imparting nor plainly summarizing verifiable information" (Wikipedia, 2021e). Peacock phrases, also called "Puffery," are commonly introduced by editors who have inherent bias, such as the administrators of a community college editing that institution's Wikipedia page, or the owner of a business editing the business's page. Such edits with inherent bias are heavily discouraged (Wikipedia, 2021d), but are hard to detect; Wikipedia does not require any form of identity verification to make an account. While Wikipedia encourages users to disclose conflicts of interest, many casual editors do not read the editorial standards.

Because it is unfeasible for human editors to monitor all Wikipedia pages for well-intentioned but badly-formed edits, there is a need for bots which can pre-filter pages. Such bots, including bots based on ORES, highlight pages to their users that may require further attention. Prior to this project, there was no such bot for detecting peacock phrases.

3 Objectives and Research Questions

This project aims to build a machine learning model to classify whether sentences contain peacock phrases. The benefits of such a model for Wikipedia are clear: such a model can scan Wikipedia for peacock phrases, allowing editors to quickly find and correct biased articles.

This problem is also interesting from a research perspective, however. At first it may appear quite simple: peacock phrases are a case of positive sentiment, so shouldn't sentiment analysis be able to easily distinguish them? However, this turns out to be incorrect (see section 6.1), leading to further questions. What kinds of context are important for distinguishing peacock phrases? How does a model determine what level of praise is warranted? And, more generally, what are the benefits and pitfalls of treating Wikipedia as a semi-labelled data source for supervised learning?

4 Types of Models

Over the course of this project, we trained several models. They fall into two general types: linear support vector classification and transformers.

Support vector machines are a type of supervised machine learning model. The input is a series of feature vectors– vectors representing each sentence as a series of numbers, encoding information about the words in the sentence– and a series of correct labels. During training, the model uses these labelled examples to learn a series of weights and biases. Then, to predict, the model uses the weight matrix, w,

and the bias vector, b: the SVM calculates $w^T x + b$, predicting the positive class if the resulting value is positive and the negative class otherwise (Goodfellow et al., 2016).

SVM models are powerful, but they have several limitations. The model we trained used the bag of words assumption, which means the model does not consider word order– "dogs chase cats" and "cats chase dogs" have the same representation. In addition, this model is linear; this means the model is relatively simple and may be unable to represent more complex relationships between terms. This type of model is useful because, by examining the weight matrix, we can isolate the terms that are given the highest positive and negative weights; these represent the features with the highest impact on the final prediction. However, for the greatest accuracy, we turn to a transformer model.

A transformer is a specialized kind of neural network. In addition to the nonlinearity that is common in neural nets, transformers have a structure of encoding a sentence and then decoding it, making them powerful for machine translation tasks. Transformers are also traditionally pretrained on a sequence prediction task, using immense datasets; thus, it is believed that they encode some knowledge of language. The particular transformer used for this project, RoBERTa, was pretrained over both a large book corpus and the entirety of the English Wikipedia (Liu et al., 2019). The pretrained model is then trained further with the dataset for this project; this generally leads to higher performance than training a model from random initialization.

5 Data Collection

Wikipedia uses category tags to note issues with an article, article section, or sentence. These tags display a warning at the top of the involved section or a superscript after the sentence; examples of both types of warnings are in fig. 1. One of the possible tags is for "Puffery" or "peacock", which are terms Wikipedia editors use interchangeably to refer to peacock phrases. Examples of sentences that Wikipedia editors identified as containing peacock phrases are in table 1. Following in the same approach as Hube and Fetahu (2018), we take these tags as a positive example set. As there is no Wikipedia tag to indicate that a sentence is *without* issues, we considered several approaches to source negative examples.

For each data collection method, we aimed for an equal balance between positive and negative examples in the training set. At the time of data collection, there were 3,962 articles with at least one sentence or section tagged "Puffery" or "peacock" on Wikipedia. The vast majority of these are sections tagged "peacock;" after data cleaning, 284 sentences with peacock phrases were extracted. Thus, given these sets, the data collection task becomes identifying the same number of sentences or articles without peacock

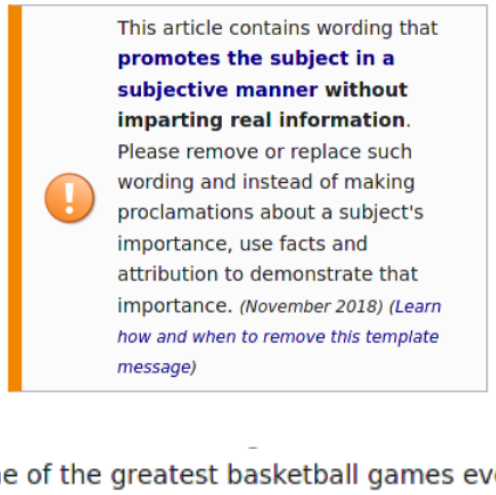

Figure 1: Both types of "peacock phrase" warnings: an article-level warning (top) and a sentence-level warning (bottom).

phrases. In anticipation of using a transformer model with limitations on the maximum sequence length, we focused primarily on sentence-level comparisons, though full-text approaches were also considered.

Sentences with peacock phrases
Game 5 was truly *one of the greatest* basketball games *ever to be played.*
The school has *excellent* infrastructure in terms of classrooms, laboratories, library, playgrounds, sports and recreation.
Holguín later became a *great* orator, debater, and writer.

Table 1: These sentences were tagged as containing peacock phrases by Wikipedia editors. The italicized sections are peacock phrases in each sentence; italics were added by the authors.

5.1 Initial Approach

To preserve topic information, we initially sampled negative examples from the same articles as the positive examples. For this sampling, we excluded articles with top-level issue templates, which we define as articles where there are any template warnings displayed at the top of the page. We then used the Python libraries BeautifulSoup4 and NLTK to process the html webpage into sentences, then excluded any sentences that were tagged with any issue, producing a list of sentences that were not tagged with any issues. For each sentence with a peacock phrase from that article, we randomly sampled a corresponding sentence from the "no-issue" list. This resulted in an f1-score at 0.5416; since the classes are balanced, this is little better than random guessing.

While this approach has the advantage of minimizing topic bias, it fails because of the composition of

these articles. Because one person often writes much of the content for smaller Wikipedia pages, articles that have one or more peacock phrases tagged are very likely to have other, untagged issues as well. A manual inspection of the negative examples reveals this issue; table 2 lists several sentences that were considered negative examples during this data collection that display peacock phrases, such as “strong corporate culture,” “widely acclaimed,” and “renowed journalist.” As bias can be difficult to determine by manual inspection, and the authors do not have prior experience identifying bias by Wikipedia’s editorial guidelines, we instead sought a cleaner dataset.

Sentences with peacock phrases that were sampled as negative examples in the initial approach
The brand has a *strong corporate culture* of listening and receiving feedback, *especially from the consumer’s perspective.*
When launched, the Studio XPS 13 was *widely acclaimed* as the *ultimate notebook*, offering an *irresistible combination* of technology, *cutting-edge design*, performance and *luxury style* in the *increasingly popular market* for lighter notebooks.
He was also a *renowned journalist* and professor of literature and history.

Table 2: Though the negative examples did not have a sentence- or article-level tag for bias, peacock phrases were common. The italicized phrases here are peacock phrases; emphasis is the authors’.

5.2 Full-text approaches

If other sentences from the positive example article are also peacock phrases, then why not use the entire article as a training example? This full-text approach has the advantage of including other, non-tagged peacock sentences. The only difficulty is finding negative examples.

Not every peacock phrase is tagged, but even so it is clear that most articles on Wikipedia do not have peacock phrases. Thus, a reasonable next approach was to choose articles at random to sample negative examples from. Wikipedia has a "random article" functionality that returns an article from the English Wikipedia, and this was used to generate a sequence of random articles to sample from. Articles were excluded from this sequence at sample time if they were found to have any top-level issue templates, consistent with the reasoning from the initial data collection.

Unfortunately, this approach suffered from issues with learning topics as features instead of identifying peacock words. Because the peacock phrases were a relatively small part of the articles, the model focused on higher-level features such as the topic of the article. This is evident in fig. 2, which shows the top ten positively-weighted words (features) in the bag-of-words linear SVM model. The bars represent how much the feature weights the final result toward a classification as peacock. Clearly, it’s not true that words such as “school,”, “training,” and “film” are indicators of bias. Instead, the model has correctly (but

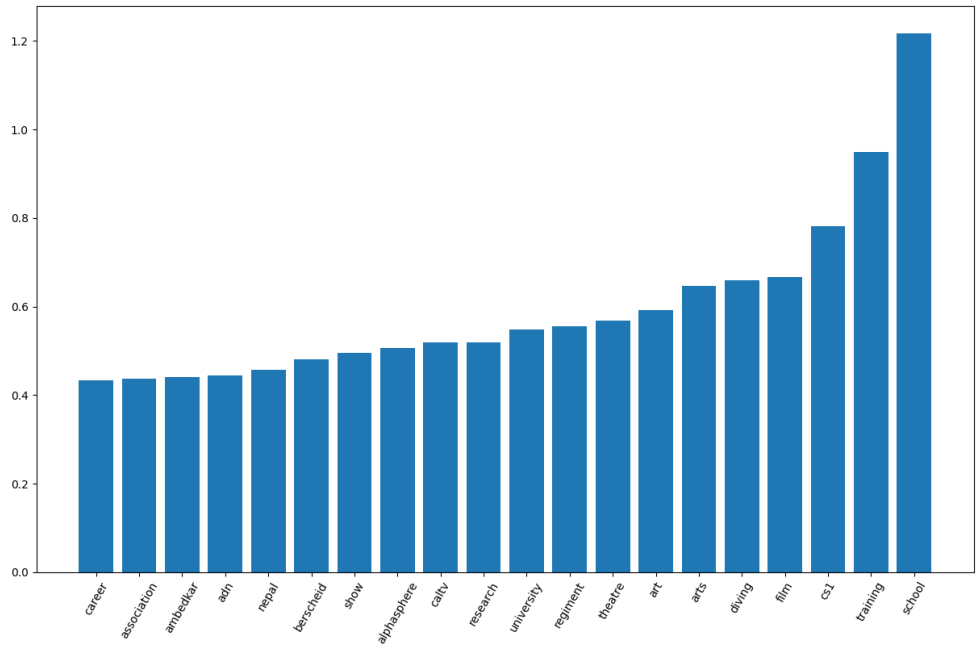

Figure 2: The words that the linear SVM model with bag of words features considered strong indicators of bias included neutral words like "school" and "film." This model was trained for full-document classification.

uselessly) noticed that articles about schools, businesses, and works of art tend to have higher levels of bias than articles about technology or locations.

It may be possible to control for this by sampling articles in similar topics to the articles with peacock words. However, the full-text approach is less useful in general than the sentence-level approach– it is far more relevant to identify a particular sentence that needs revision than to identify that there is at least one issue in a 10,000 word article. As this approach did not demonstrate significant gains in accuracy, we discard the full-text approach in favor of exploring sentence-level approaches in more depth.

5.3 Sampling at random

At this point, it was unclear whether the failure of the full-text approach was solely from the choice to use full articles or impacted by the random sampling. To determine this, we sampled random sentences from random articles and trained a sentence-level model. As before, articles with any top-level warnings were excluded. In addition, to avoid a bias toward a single topic, only a single sentence was sampled from each article chosen randomly.

While there was a clear improvement in performance with this method (f1 score increased to 0.7795), this alone is not enough to declare success. Models can achieve better-than-random performance on a task without identifying appropriate features. An examination of the features weighted positively and

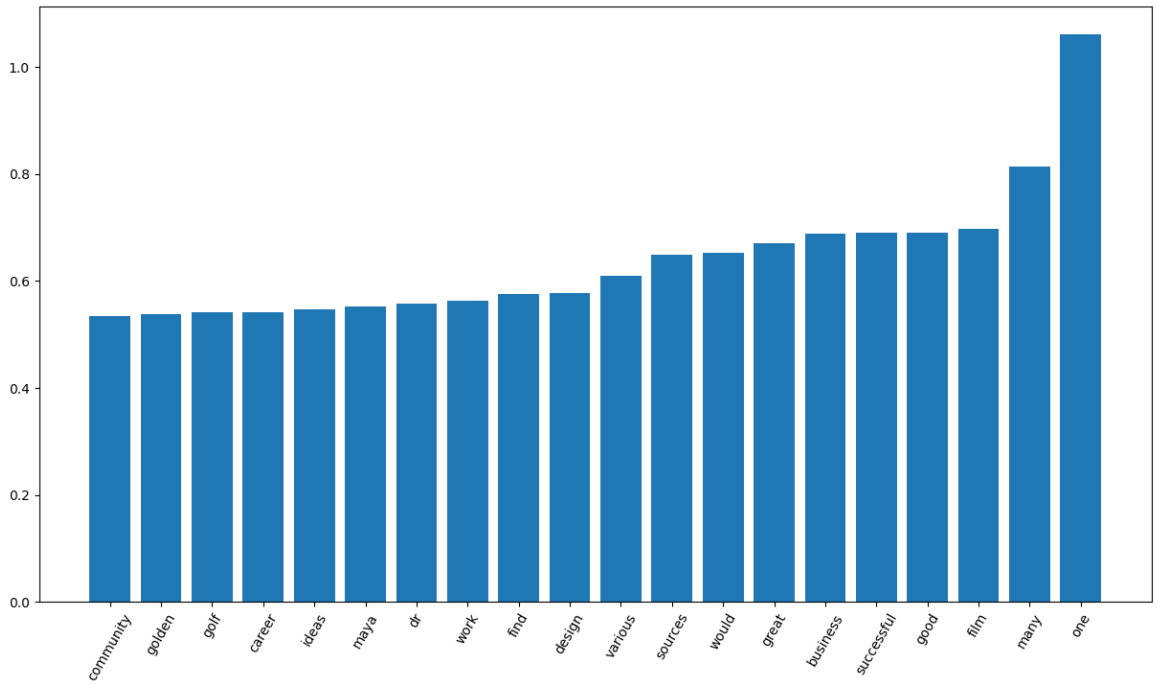

Figure 3: This diagram captures feature information about the trained bag of words linear SVM model. This model was trained on a dataset of peacock sentences and sentences sampled from random articles. The x axis indicates features that the model considers indicators that the sentence should be classified as peacock; the y axis is a measure of the relative importance of these features. Some of these features seem reasonable, but others are clearly topic-based.

negatively by the model, in fig. 3, demonstrates this.

As with the full-text model, many of the features are topic-based, including words such as "film" and "business." While this model finds better features than the full-text model, it still displays topic sensitivity. This topic sensitivity alone is not a granular enough approach to bias detection, and so we continue to search for an appropriate dataset.

5.4 Sampling using Information Retrieval

It is clear that a good set of negative examples will feature similar topics to the positive examples, but that sampling from the same article results in too high of a false negative rate to be meaningful. We explored using Wikipedia's article categories to sample articles in the same grouping, but the category system obscures the nesting relationships between categories (i.e. it is not possible to view all supercategories of a category), and the lowest-level categories often have too few articles in them to be useful. Instead, we turn to a different approach: using information retrieval (IR) to "search" over Wikipedia for similar articles.

Using IR on Wikipedia is not a new concept; most notably, an information retrieval system over

Wikipedia was used as part of IBM Watson's system for guessing Jeopardy answers. Using a Wikipedia dump and the Lucene search engine library, we built a simple custom IR system. The system indexes article titles and article content; the content is stemmed, a form of text normalization that uses heuristics to cut suffixes off words. To find a negative example for each positive example, we used the positive example text as a query, searching against the article text; this returns the top 10 articles that the system deems most similar. We excluded all articles that had top-level warnings or tagged peacock phrases, which also has the effect of excluding the article that the positive example is drawn from; then, we sampled a sentence at random from the highest-ranked article left.

At first, this may seem counterintuitive– wouldn't the system find other articles with peacock phrases? However, because peacock phrases are rare relative to topic words, the effect of the topic words is more pronounced; i.e., a sentence such as "She is an excellent doctor" will return articles about doctors instead of articles that use the word "excellent" because an article about doctors will use the word "doctor" many times, while an article is unlikely to use the word "excellent" more than once or twice. To confirm this intuition, we chose ten examples at random and tried querying using the peacock sentence, the title of the article, and the first line of the article (which is generally a summarizing line about the topic). The performance of a linear SVM model trained on bag-of-words was similar for using the peacock phrase and the first line. For ease of programming, we thus used the peacock phrase as the query term.

While the linear models trained on this dataset did not perform much better, the features chosen by the models were greatly improved– see fig. 4. It is clear that the model was learning words with high valence (high positivity) as indicators that the sentence may contain a peacock phrase. This is an indication that the dataset is clean enough to apply a more sophisticated model; the relatively poor performance is not an issue with the unclean data, but with the simplicity of the model.

5.5 Sampling using revision history

While using information retrieval produced a good dataset, the number of positive examples was still quite small (<300). To expand the dataset, we investigated using revision history. Every page on Wikipedia also stores the history of its revisions, and this is accessible through the Wikipedia API. We wrote a script to look through each page in namespace 0 (i.e. each page that is an article instead of a user bio or documentation page), scanning the last 500 revisions for anything tagged with "peacock phrase" or "Puffery." We ran this approach for the first 250,000 articles and found less than 5 examples. Because this approach was time-intensive– taking approximately 20 minutes per 10,000 articles, depending on article

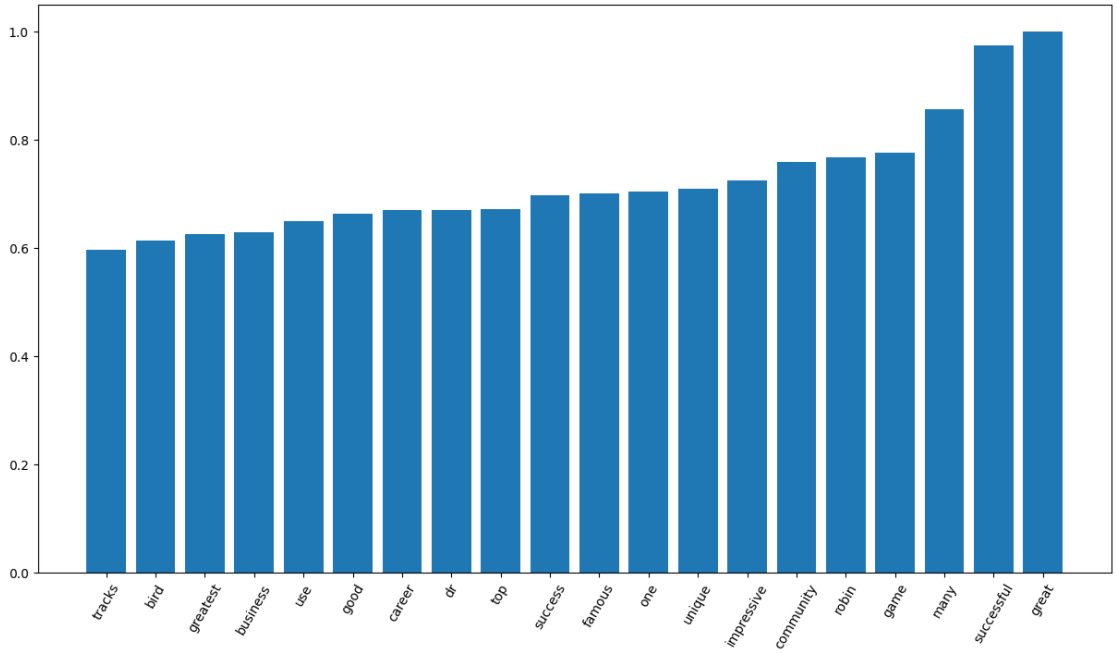

Figure 4: The features indicate that the model is learning to categorize based on words one would expect in a peacock phrase.

length– we discontinued the approach.

This strategy of looking through revision history may be useful for other problems where there is a higher occurence rate; the code to perform the revision search is available in this project's Github repository.

6 Model Approaches

6.1 Sentiment Analysis

The problem of peacock phrases appears closely tied to sentiment analysis. Peacock phrases are an issue of overly positive sentiment; it seems logical that sentiment analysis would be able to detect these phrases. We used two out-of-the-box sentiment models, TextBlob and vaderSentiment (Gilbert, 2014), on several randomly-selected sentences from the dataset. Not only did the sentiment models fail to distinguish between the classes, but some non-peacock-phrase sentences had higher positive sentiment than some peacock phrase sentences. We present two possible reasons for this counterintuitive result.

Wikipedia's goal is to be a neutral source; as such, most of Wikipedia has very low levels of sentiment. While peacock phrases have higher positive sentiment than the rest of Wikipedia, that does not necessarily mean they have exceptionally high positive sentiment. In table 3 there are several sentences tagged for peacock phrases that are relatively neutral. A single word may not impact the sentiment classification

much, but may be enough for Wikipedia editors to deem the sentence biased.

Sentences with peacock phrases that have relatively neutral sentiment
Astra (Romania) was the most important aircraft builder in Romania from 1920 to 1927. .
He is the only one of their ex-owners of liquidated banks who proposed to cover the bank's debts to the Deposit Guarantee Fund and the NBU.
Dr. Luis M. Mabilangan - An institution in the field of Philippine Pediatrics

Table 3: The peacock phrases here– "most important," "only one," and "institution" – are relatively subtle.

The second, likely more prominent reason, is that Wikipedia allows some forms of cited praise. For instance, Wikipedia's neutral point of view page gives this example sentence: "Dylan was included in Time's 100: The Most Important People of the Century, in which he was called "master poet, caustic social critic and intrepid, guiding spirit of the counterculture generation".[1] By the mid-1970s, his songs had been covered by hundreds of other artists.[2]." There are several highly positive words in this sentence– e.g "most important people," "master poet"– but it is not considered to contain a peacock phrase because it has proper attribution and, crucially, reflects the opinions of a reputable source rather than the opinions of the editors. This fine-grained distinction is not sentiment-based.

6.2 Preliminary Models

Having discarded the sentiment analysis approach for the reasons above, we treat the problem of whether a sentence contains a peacock phrase as a binary classification problem.

As discussed in section 5, we first trained a simple bag-of-words SVM model to evaluate each dataset. The advantage of this approach was that we were able to examine the most important features to ensure the model was learning something about sentiment instead of topic. Once we settled on a final dataset, we performed one final experiment with the SVM model: testing whether it was beneficial to leave in stopwords.

Running the model on the dataset with stopwords included resulted in a 0.04 drop in f1 score, and several stopwords appeared in the most influential features. The model was likely learning some small difference in the occurrence of stopwords between the two datasets, since both datasets were relatively small; it seems unlikely that prevalence of any stopword is a useful feature. For the rest of the experiments, we continue to omit stopwords.

6.3 RoBERTa

Initially, training RoBERTa resulted in gains of almost 10 percentage points. After tuning a model on the dataset with stopwords removed, the final model had an f1 score of 0.726, compared to 0.658 for the best SVM model. (The model trained on full articles is excluded, as its features were entirely topic-based.)

An error analysis revealed a number of false positives where sentences had mildly positive but completely factual statements. Often, in the original article, these claims were cited. The breakthrough here occurred when we considered the impact of citations.

Previously, all in-line citations were removed in the data cleaning step. However, at this point we revised the data cleaning to instead leave a keyword, CITE, at each point where a citation had been. This resulted in significant gains: performance increased, for a final f1 of 0.963. This model is the final version used in the integration.

We completed a manual inspection of the data to confirm that several positive examples had citations and many negative examples did not. At this point, we also trained another SVM model on the dataset with the citation keyword, achieving an f1 of 0.680. Similar positive features were used in this model as in the previous SVM model; in addition, the CITE keyword was a strong negative feature. However, some sentences with citations were still predicted to contain peacock phrases. This appears consistent with the our observations of the data.

Table 4 shows the relative performance of the models trained. The SVM model with citations showed improved performance over the SVM model without citations; for instance, the model with citations correctly identified the sentence "She was inducted into the Alabama Women's Hall of Fame in 1992 [citation]" as acceptable, while the model without citations incorrectly labelled this sentence as containing a peacock phrase. This is expected, as this sentence is about a verifiable entity with a very positively-loaded name. However, the SVM model with citations struggled with more subtle citations, such as the sentence "Their colour names, such as Elephant's Breath, have becoming talking points [citation] in themselves." This sentence cites an article where the colors are a major talking point, so there is not a peacock phrase here; however, both SVM models identify the sentence as containing one. The RoBERTa model with citations is the only model to correctly identify this sentence as non-peacock.

The RoBERTa model without citations achieves slightly higher performance than the SVM models, but fails on sentences where the peacock phrases are subtle or could conceivably be taken from cited phrase. For instance, the RoBERTa model without citations fails to identify "Under the rule of the country's first King, Croatia became one of the most powerful kingdoms in the Balkans" as a peacock sentence, but

	Model			
	SVM: words only	SVM: words+CITE	RoBERTa: words only	RoBERTa: words+CITE
Precision	0.600	0.627	0.630	0.937
Recall	0.729	0.743	0.853	0.989
F1	0.658	0.680	0.726	0.963

Table 4: The models with citation information generally performed better, but both RoBERTa models outperformed the SVM models.

the RoBERTa model with citations correctly tags this as a peacock sentence. Interestingly, the SVM model with citations also correctly tags this sentence, despite having lower overall performance than either RoBERTa model. Neither using transformers nor marking citations is alone enough to achieve high performance on this task; both combined are necessary to capture the nuances of the classification problem.

7 Integration

After developing a model with good performance, our focus shifted to sharing it with the Wikipedia community. The Wikimedia Foundation has a robust system for technical contributions, in the form of user-maintained "tools." Tools can be hosted on Toolforge, a hosting environment with access to replicas of Wikipedia's production databases (Wikitech, 2021a). This section describes how this tool was deployed on Toolforge and how it can be used.

7.1 Deployment

Deployment on Toolforge requires a free Wikimedia developer account. Account creation requests are approved by community administrators. With an account, users can access Toolforge, put files on the server, and create and maintain tool accounts, which are a rough analogue to Github organizations. Each tool account manages a tool or set of related tools. For this project, the tool account is simply named "peacock-finder."

The primary way that bots interact with data from Wikipedia is by querying Wikipedia's MariaDB databases, using schema documentation available on MediaWiki. A Python library named toolforge simplifies the setup for querying (Wikitech, 2021b). Wikipedia has a detailed policy on what bots may and may not do. For this project, the relevant policy is that bots may not edit pages with changes that are context-sensitive and normally are performed by human editors (Wikipedia, 2021c). This includes bias detection, as editors may have differing opinions on whether a particular sentence is biased or not. This policy means that the bot cannot add the peacock phrase tag to sentences.

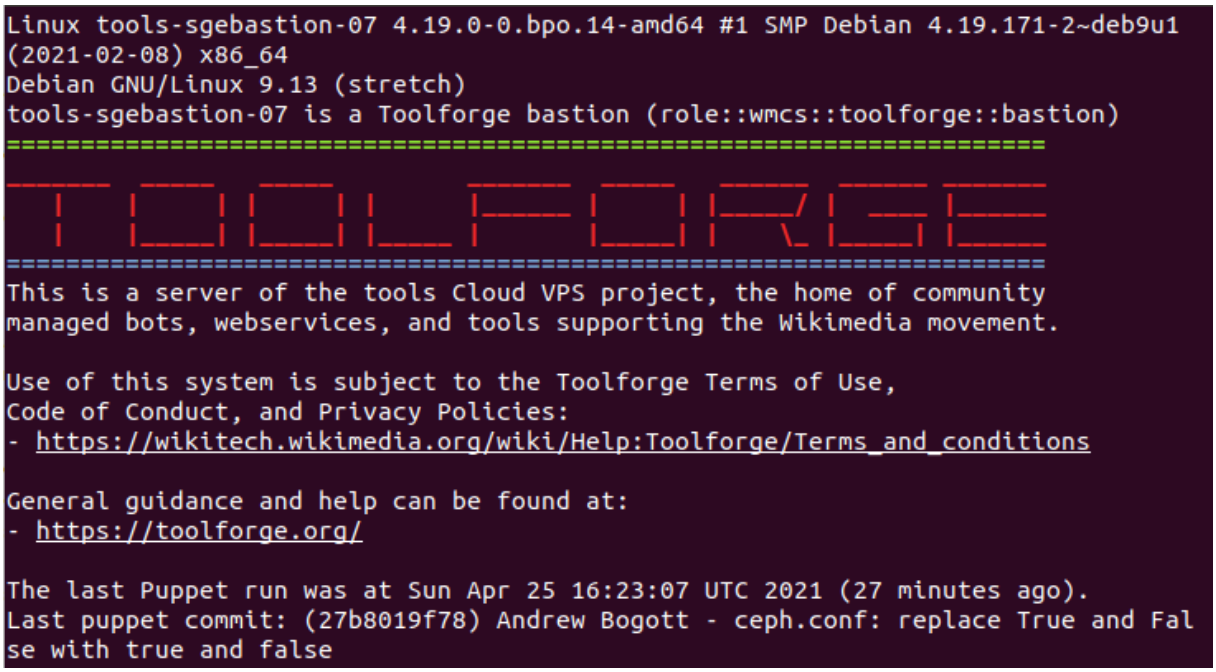

Figure 5: The Toolforge login message

After consultation with Wikipedia administrator CaptainEek, the authors determined that the most useful application of the project was for the bot to maintain a list of pages that required investigation. This is similar to the Category:Articles with peacock terms page, where all articles with at least one tagged peacock phrase are listed; the intent of both pages is that Wikipedia editors with spare time will check the page and fix the articles on it.

Wikipedia has several different namespaces, which separate articles into categories (e.g., to distinguish articles about bias as a concept from Wikipedia policies on bias) (Wikipedia, 2021f). To edit its peacock phrase page, the bot has its own account. Each account receives its own page in the user namespace; most users use their page to write a brief biography or other information. The bot uses the Wikipedia API to write to its own user page. Periodically, the bot checks for new revisions on the pages it has flagged; if a page has new revisions, it re-runs the model to check if the page should remain tagged. Users can communicate with the maintainers using the bot's talk page, a forum-like space associated with the user page where editors can discuss the page's content.

The bot's work is visible on its user page.

8 Conclusions & Future Work

This project demonstrates the importance of dataset curation, especially when working with semi-labelled data. Transformer models such as RoBERTa are powerful, but can fall prey to the same issues as simpler,

linear models when spurious correlations are frequent in the data. Data collection and cleaning remains an important and often under-considered part of training a neural net.

While this project achieves satisfactory performance on detecting peacock phrases, more work remains. Sentiment analysis alone was not sufficient for this task. However, adding a step that screens sentences for high positive sentiment (or using a transformer model pre-trained on sentiment analysis) may further improve performance, though this project did not explore these possibilities. In addition, this project does not distinguish between citations. One can imagine a citation that is spurious or unreliable being attached to a peacock sentence. While it is not always possible to examine the source text of a cited work (e.g. if the work is a physical book or behind a paywall), examining citations for quality may also improve performance on distinguishing between peacock sentences with citations and cited, appropriate praise. Finally, Adler et al. (2011) had great success with using article and editor metadata to identify likelihoods of vandalism, and a future model for peacock phrases could weight this as a factor.

While the specific category of peacock phrases is unique to Wikipedia, bias detection problems are not. This project demonstrates the value of reaching beyond traditional sentiment analysis approaches. While sentiment analysis is powerful, considering domain-specific features (e.g. citations on Wikipedia) can lead to bigger gains; additionally, neither approach performs well without a quality dataset.

References

B. Thomas Adler, Luca de Alfaro, Santiago M. Mola-Velasco, Paolo Rosso, and Andrew G. West. 2011. Wikipedia Vandalism Detection: Combining Natural Language, Metadata, and Reputation Features. In *Computational Linguistics and Intelligent Text Processing*, Lecture Notes in Computer Science, pages 277–288, Berlin, Heidelberg. Springer.

Richárd Farkas, Veronika Vincze, György Móra, János Csirik, and György Szarvas. 2010. The CoNLL-2010 shared task: Learning to detect hedges and their scope in natural language text. In *Proceedings of the Fourteenth Conference on Computational Natural Language Learning – Shared Task*, pages 1–12, Uppsala, Sweden. Association for Computational Linguistics.

CJ Hutto Eric Gilbert. 2014. Vader: A parsimonious rule-based model for sentiment analysis of social media text. In *Eighth International Conference on Weblogs and Social Media (ICWSM-14). Available at (20/04/16) http://comp. social. gatech. edu/papers/icwsm14. vader. hutto. pdf*.

Ian Goodfellow, Yoshua Bengio, and Aaron Courville. 2016. *Deep Learning*. MIT Press. `http://www.deeplearningbook.org`.

Christoph Hube and Besnik Fetahu. 2018. Detecting Biased Statements in Wikipedia. In *Companion Proceedings of the The Web Conference 2018*, WWW '18, pages 1779–1786, Lyon, France. International World Wide Web Conferences Steering Committee.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized BERT pretraining approach. *CoRR*, abs/1907.11692.

Wikimedia. 2021. ORES - MediaWiki.

Wikipedia. 2021a. History of Wikipedia. Page Version ID: 1014548047.

Wikipedia. 2021b. Wikipedia. Page Version ID: 1019154774.

Wikipedia. 2021c. Wikipedia:Bot policy. Page Version ID: 1015757461.

Wikipedia. 2021d. Wikipedia:Conflict of interest. Page Version ID: 1018992805.

Wikipedia. 2021e. Wikipedia:Manual of Style/Words to watch. Page Version ID: 1019528666.

Wikipedia. 2021f. Wikipedia:Namespace. Page Version ID: 1017853864.

Wikipedia. 2021g. Wikipedia:Neutral point of view. Page Version ID: 1015224310.

Wikipedia. 2021h. Wikipedia:Protection policy. Page Version ID: 1017833570.

Wikipedia. 2021i. Wikipedia:Size of Wikipedia. Page Version ID: 1019560660.

Wikitech. 2021a. Portal:Toolforge - Wikitech.

Wikitech. 2021b. User:Legoktm/toolforge library - Wikitech.