Detection of Puffery on the English Wikipedia

Amanda Bertsch
Carnegie Mellon University
abertsch@andrew.cmu.edu

Steven Bethard
University of Arizona
bethard@arizona.edu

Abstract

On Wikipedia, an online crowdsourced encyclopedia, volunteers enforce the encyclopedia's editorial policies. Wikipedia's policy on maintaining a neutral point of view has inspired recent research on bias detection, including "weasel words" and "hedges". Yet to date, little work has been done on identifying "puffery," phrases that are overly positive without a verifiable source. We demonstrate that collecting training data for this task requires some care, and construct a dataset by combining Wikipedia editorial annotations and information retrieval techniques. We compare several approaches to predicting puffery, and achieve 0.963 f1 score by incorporating citation features into a RoBERTa model. Finally, we demonstrate how to integrate our model with Wikipedia's public infrastructure 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, 2021d), outlines best practices for maintaining the encyclopedia impartiality that Wikipedia strives toward. But with few restrictions on who may edit and with 1.9 edits every second (Wikipedia, 2021g), it is impossible to manually review each edit for compliance to such policies.

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 are still left entirely to human editors. Without some assistance to direct humans to potentially problematic articles, the long tail of pages with fewer views will likely have many more issues than the small number of pages that many editors frequently visit.

We apply natural language processing methods to automatically detect sentences in Wikipedia that violate Wikipedia's Neutral Point of View policy by using "puffery" or "peacock phrases", i.e., by promoting a subject rather than simply imparting information. The contributions of this paper are:

- An exploration of methods for extracting training examples of puffery, the most successful of which couples Wikipedia editorial annotations with information retrieval techniques.
- Introduction of several machine-learning approaches to predicting puffery, the best of which achieves 0.963 f1 score by incorporating citation features into a RoBERTa model.
- Integration of the puffery detection model into Wikipedia's public infrastructure.

The code and data for this paper are available at https://github.com/abertsch72/wikipedia-puffery-detection.

2 Background and Motivation

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

This freedom to edit inspires vandalism, so the Wikimedia Foundation, which maintains Wikipedia, has developed the ORES project, a suite of natural language processing tools for detecting vandalism. 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).

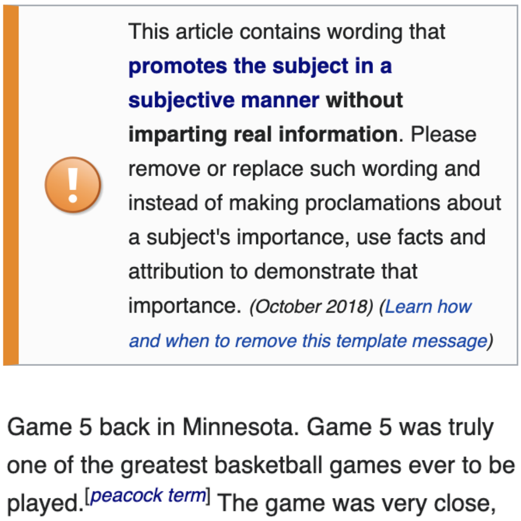
Yet the vast majority of non-standard edits on

Wikipedia are not vandalism or malicious. These edits, generally by new editors or editors without accounts, violate Wikipedia's policies unintentionally or unknowingly. Senior editors and volunteer administrators spend significant amounts of time welcoming new editors and reverting these bad edits, but many bad edits on low-traffic pages are still left up for months, years, or, in the case of some pages, over a decade.

Prior research on identifying problematic edits has considered 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, 2021c), and hedges, which Farkas et al. (2010) define as phrases "indicating that authors do not or cannot back up their opinions/statements with facts. The best models for detecting weasel words in both the CoNLL-2010 ACL shared task on weasel words (Farkas et al., 2010), and a multilingual weasel word corpus (Aleksandrova et al., 2019) use bag-of-words classification approaches (Georgescul, 2010; Aleksandrova et al., 2019).

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. Recently, Hube and Fetahu (2018) 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.

We are aware of no prior work that has focused on the identification of puffery (also known as 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, 2021c). Examples of puffery include "excellent infrastructure," "renowned journalist," or "defining figure." Note that puffery is distinct from positive sentiment, as Wikipedia allows praise when it has proper attribution and reflects a consensus opinion of reputable sources, for example, "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]"



This article contains wording that **promotes the subject in a subjective manner without imparting real information**. Please remove or replace such wording and instead of making proclamations about a subject's importance, use facts and attribution to demonstrate that importance. *(October 2018) (Learn how and when to remove this template message)*

Game 5 back in Minnesota. Game 5 was truly one of the greatest basketball games ever to be played.[*peacock term*] The game was very close,

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

3 Data Collection

Puffery in Wikipedia is noted by editors using the "puffery" or "peacock" tags, which can be applied to an article, section, or sentence, and display a warning at the top of an article or section, or a superscript after a sentence. Examples are shown in fig. 1. Similar to Hube and Fetahu (2018), we take the sentence tags as positive examples, extracting 284 sentences[1] annotated with tags containing the words "puffery" or "peacock" from Wikipedia articles. However, as there is no Wikipedia tag to indicate that a sentence is *without* issues, we considered several approaches to extract negative examples: two types of random selection, and when we discovered that random selection yielded problematic datasets, an information retrieval-based approach.[2]

Same-article random sampling For each puffery sentence, we sampled one random sentence from the same article as a non-puffery sentence with the same topic. In an attempt to avoid selecting negative sentences that contain puffery but are not tagged as such, we excluded any article where there were any template warnings displayed at the top of the page, and any sentence tagged with any issue.

[1] In all experiments, sentences were extracted with a combination of BeautifulSoup4 and NLTK (Bird et al., 2009).

[2] We also investigated using revision history to look for puffery that had been edited out, but because of the low prevalence of tagged puffery, this approach was overly time-intensive and produced too few results.

Same-article random	Random-article random	Information retrieval
most	school	great
dr	cs1	successful
very	training	many
community	diving	game
career	regiment	robin
famous	film	community
of	burton	impressive

Table 1: The top positive features (indicating a sentence may contain puffery) in linear SVM BoW models for different data selection strategies. Features are listed from most influential at the top to least influential at the bottom.

Random-article random sampling For each puffery sentence, we sampled a random other article (using Wikipedia's "random article" functionality) and a random sentence from that article. We excluded articles with any template warnings or sentences tagged with issues (thus also excluding the original article with the puffery). Because the supermajority of articles do not contain puffery, we expect that these sentences are highly likely to not contain puffery.

Information-retrieval sampling For each puffery sentence, we take a Lucene index of Wikipedia documents, use the puffery sentence as a query, take the top-retrieved article, and select a random sentence from that article. As before, we excluded articles with any template warnings or sentences tagged with issues. Because typically only a small number of words in each sentence are puffery, we expect the articles to be more closely related to the topic of the puffery sentence than articles chosen at random.

Analysis For all data selection techniques, we trained a linear support vector machine (SVM) classifier with bag-of-words (BoW) features and inspected both the model's performance (to see if the task was learnable), and the most important features of the model (to see if sensible things were being learned). Table 1 shows the most important features of each model. Same-article random sampling resulted in an f1-score at 0.542, and some features related to puffery (e.g., “famous”) but others not (e.g., “dr”). A manual inspection of the examples selected by this strategy revealed that many of our “negative” examples were actually positive ones (containing phrases like “strong corporate culture,” “widely acclaimed,” and “renowed journalist”), because non-tagged puffery is common in articles containing a puffery-tagged sentence. Random-article random sampling resulted in an F1-score of 0.780, and a set of features that suggested that the model was simply learning a topic bias: articles about education, film, and business were more likely to be tagged for puffery. Information-retrieval sampling resulted in an F1 score of 0.658, with top features reflecting peacock words, such as “great”, “successful”, and “impressive”.

We thus selected information-retrieval sampling to construct the official evaluation dataset. The final dataset contains 284 puffery sentences and 284 topic-matched non-puffery sentences. These 568 sentences were shuffled together before randomly selecting 10% (57 sentences) as a test set.

4 Models

We considered two machine-learning models for puffery prediction.

BoW-SVM Similar to weasel word detection approaches (Georgescul, 2010; Aleksandrova et al., 2019), we train a linear support vector machine on bag-of-words features.

RoBERTa We also consider a transformer-based neural network, RoBERTa, which has shown excellent performance on a variety of natural language processing tasks (Liu et al., 2019). RoBERTa, unlike the bag-of-words model, can consider the context in which potential puffery words occur. We fine-tuned RoBERTa with the puffery data, using a learning rate of 1×10^{-5} and a batch size of 5.

+CITE Preliminary experiments with the above two models revealed that they still struggled with sentences that were mildly positive, but factual and accompanied by citations. We thus augment the model inputs by inserting a pseudo-token “CITE” at each position in the sentence where a citation occurred. In the BoW-SVM model, this means the bag-of-words CITE feature will give the count of citations in the sentence. In the RoBERTa model, this means the transformer's self-attention can decide how much to attend to each CITE (if present) when evaluating a potential puffery word.

5 Results

Table 2 shows the performance of all models in terms of precision, recall, and F1 of detecting

Model	Input	P	R	F1
BoW-SVM	words only	0.600	0.729	0.658
BoW-SVM	words + CITE	0.627	0.743	0.680
RoBERTa	words only	0.630	0.853	0.726
RoBERTa	words + CITE	**0.937**	**0.989**	**0.963**

Table 2: Performance of models with different input types on the puffery prediction test set.

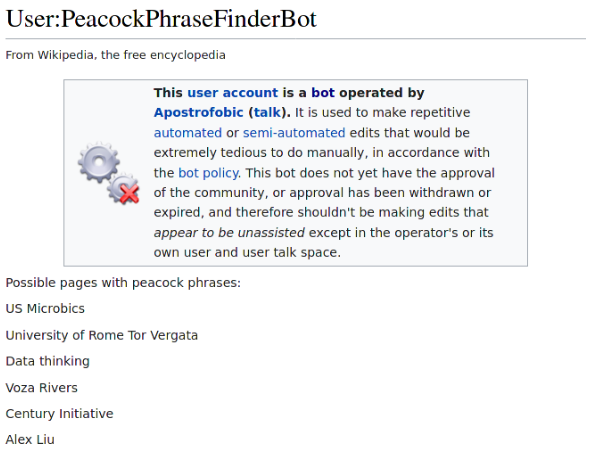
puffery. The BoW-SVM model, even with citation information, achieves only 0.680 F1. RoBERTa without citation information has a much higher recall than the BoW-SVM models (+0.110 or more higher). Representing citations in the input leads to huge gains for RoBERTa (+0.307 precision, +0.136 recall), with this best model achieving 0.963 F1.

Qualitative analysis Adding citation information to the BoW-SVM model allowed it to correctly classify "She was inducted into the Alabama Women's Hall of Fame in 1992 [citation]" as non-puffery, while the model without citations incorrectly labelled this as puffery. But only RoBERTa with citation information was able to correctly classify "Their colour names, such as Elephant's Breath, have becoming talking points [citation] in themselves." as non-puffery. (This sentence cites an article where the colors are a major talking point, so there is no puffery here.) This demonstrates that is it important not only to know when citations are present (what the BoW-SVM sees), but where exactly they occur (what RoBERTa sees).

6 Integration

Having developed a model that can accurately detect puffery, 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" hosted on Toolforge, a hosting environment with access to replicas of Wikipedia's production databases (Wikitech, 2021a). Bots may read data from Wikipedia by querying these MariaDB databases, using schema documentation available on MediaWiki and a Python library, `toolforge` (Wikitech, 2021b).

Wikipedia has a detailed policy on bots, which forbids bots from making changes to address subjective issues (Wikipedia, 2021b). This includes bias detection, as editors may have differing opinions on whether a particular sentence is biased or not. This policy means that a bot cannot add the puffery tag to sentences.



User:PeacockPhraseFinderBot

From Wikipedia, the free encyclopedia

This user account is a bot operated by Apostrofobic (talk). It is used to make repetitive automated or semi-automated edits that would be extremely tedious to do manually, in accordance with the bot policy. This bot does not yet have the approval of the community, or approval has been withdrawn or expired, and therefore shouldn't be making edits that *appear to be unassisted* except in the operator's or its own user and user talk space.

Possible pages with peacock phrases:

US Microbics

University of Rome Tor Vergata

Data thinking

Voza Rivers

Century Initiative

Alex Liu

Figure 2: Screenshot of the bot's output after it has seen pages with puffery.

After consultation with a Wikipedia administrator, CaptainEek, we designed a bot to maintain a list of pages that required investigation, similar to the Category:Articles with peacock terms page, but generated from model predictions rather than manual editor tags. The intent of both pages is that Wikipedia editors with spare time will check the page and fix the articles on it. Because of technical limitations on Wikipedia's server, we trained and used the smaller AlBERT model instead of RoBERTa. On the same train and test set as table 2, AlBERT achieves 0.899 F1.

The model runs once per hour, considering only the 1,000 most recently edited pages. To reduce computational cost, the model runs on 15 randomly selected sentences from each of these pages. As the bot is working at a document level instead of a sentence level, the bot labels a page as containing puffery if more than half of these selected sentences are labeled as puffery. This threshold also reduces the false positive rate.

The bot's work is visible on its user page[3]. A sample of such work is shown in fig. 2.

7 Limitations

The largest limitations of this work are the size of the dataset (284 puffery sentences) and the balanced ratio between puffery and non-puffery sentences. The balanced ratio was chosen to better align topics between puffery and non-puffery examples; however, it is not an accurate reflection

[3] `https://en.wikipedia.org/wiki/User:PeacockPhraseFinderBot`

of Wikipedia, where puffery is very rare. Thus, while this dataset is useful for refining bias detection models, real-world performance may suffer from over-prediction of puffery.

8 Conclusions & Future Work

This project demonstrates the challenge of collecting data for puffery detection models, and emphasizes the importance of careful dataset curation to avoid spurious correlations in the data. Our best dataset construction approach avoids these problems by using information retrieval techniques to ensure positive and negative examples are topically matched.

Our work also demonstrates the importance of thinking carefully about input representations, even in the age of transformer models like RoBERTa. With the standard sequence of words input encoding typically used with RoBERTa, performance was only 0.068 F1 above a simple bag-of-words SVM. However, when we augmented the input with the location of citations, RoBERTa's performance jumped 30% to 0.963 F1.

Despite our model's excellent performance, challenges remain. Our model could likely be fooled by attaching spurious or unreliable citations to sentences with puffery, since the model knows nothing of the content of the citations, only their locations in the sentence. This may argue for coupling our puffery detection techniques with techniques for evaluating the quality of citations, such as considering scientific journal impact scores (Nielsen, 2007).

References

Desislava Aleksandrova, François Lareau, and Pierre André Ménard. 2019. Multilingual sentence-level bias detection in Wikipedia. In *Proceedings of the International Conference on Recent Advances in Natural Language Processing (RANLP 2019)*, pages 42–51, Varna, Bulgaria. INCOMA Ltd.

Steven Bird, Ewan Klein, and Edward Loper. 2009. *Natural Language Processing with Python: Analyzing Text with the Natural Language Toolkit*, 1st edition edition. O'Reilly Media, Beijing ; Cambridge Mass.

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.

Maria Georgescul. 2010. A hedgehop over a max-margin framework using hedge cues. In *Proceedings of the Fourteenth Conference on Computational Natural Language Learning – Shared Task*, pages 26–31, Uppsala, Sweden. Association for Computational Linguistics.

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.

Finn Årup Nielsen. 2007. Scientific citations in wikipedia. *CoRR*, abs/0705.2106.

Wikimedia. 2021. ORES - MediaWiki.

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

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

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

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

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

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

Wikipedia. 2021g. Wikipedia:statistics. Page Version ID: 1020753220.

Wikitech. 2021a. Portal:Toolforge - Wikitech.

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