Investigating Gender With NLP Techniques

Jake Chanenson
RWRIPSHRG
gchanen1

Adriana Knight
RWRIPSHRG
aknight1

Naomi Park
RWRIPSHRG
npark1

Bayliss Wagner
RWRIPSHRG
jwagner1

Abstract

We explored how two different NLP techniques — dense vector representations of texts and gender classifiers for texts — detect and represent perceived markers of author gender, and in the latter technique, classify a document as either authored by a man or a woman. To accomplish this task we utilized two distinct pre-existing data sets, one consisting of blog entries (Mukherjee and Liu, 2010) and one composed of New York Times (*NYT*) opinion articles (Soler-Company and Wanner, 2014). Our doc2vec had an accuracy of 38.89% for texts identified as most indicative of specific gender tag on the blog data set and an accuracy of 94.44% on the *NYT* data set. Meanwhile our custom classifier, engineered with seven features, had an accuracy of 47% on the blog data set and an accuracy of 64% on the *NYT* data set. Given the fact that the *NYT* opinion data set has a standardized style and the blog data set did not, this suggests that the features we chose for our custom classifier picked up more on style than content.

1 Introduction

For this project, we aimed to investigate both informal and formal written language to see not only how, but also how *well* Natural Language Processing (NLP) tools could identify author gender. To achieve this, we built a feature-based Bayesian classification system as well as dense vector document embeddings. We were interested in building a gender classification system that leveraged a small number of highly predictive features to exceed performance of a naive Bayes baseline, focusing on sentiment analysis and other measures of expressiveness. Our dense vector embeddings, on the other hand, examine the quantitative relationships between texts that are tagged as female and male, to see if that is a reasonable avenue of investigation to use in measuring differences between texts written by men versus texts written by women. This opens up interesting questions about the efficacy of vector representations in determining the relationships between texts in a corpus in addition to observing the most indicative male and female tagged documents. Within our classification task, our baselines include a "stupid" baseline that assumes all documents are written by women to achieve an accuracy of 50% on our balanced corpora and a naive Bayes classifier, which classified texts with 75% accuracy on our blog post data set and 76% accuracy for our *NYT* opinions article data set.[1] Our implementations of feature-based Bayesian classification in addition to dense vector embeddings allowed us to investigate markers of author gender in texts and make progress towards observing performance of gender in written works.

2 Related Work

Gender is a construct of identity grounded in repeated performative acts, including how we (re)present ourselves in day to day speech and writing. In recent years, systems that detect textual predictors of an author's gender have been increasingly studied and improved by NLP researchers, with some reaching accuracies upwards of 95% when trained on the same genre of data that is tested on (Soler-Company and Wanner, 2014). Almost all of these systems use feature extraction to highlight various aspects of the text.

One feature often used for the task of author-gender classification is sentiment analysis, which has been proven as strongly predictive in studies that use a range of classifiers such as neural

[1] One piece by Brian Larson (Larson, 2017) talked about the ethical considerations that come with making gender a central question. So, we were careful to explicitly define what framework of gender we are using and because of the data sets we are working with, we are using a binary notion of gender that is Men and Women while tagging and classifying documents as Male or Female.

networks (Sboev et al., 2016) and support vector machines (Soler-Company and Wanner, 2014). Montero et. al found that adding a small set of emotion-based features to bag of words positively influences performance of gender based classifiers (Suero Montero et al., 2014). As detailed in the 'methods' section, we chose to implement this feature in our Bayesian classifier as well.

Sobev et al., in their study on the efficacy of deep learning networks to classify author gender, note that previous research has employed parameters such as frequency of words to help determine gender but thinks that this approach may skew results when models are trained on data of mixed genre. Indeed, Sobev et. al is wary of previous results being overoptimistic because of "non-stylistic factors such as topic bias in gender that can make the gender detection task easier" (Sboev et al., 2016). As one of our data sets, which draws from *New York Times* opinions articles, includes texts covering a wide array of content areas, we opted to exclude term-frequency-based features from our modified Bayesian classifier.

In one of the most informative studies to our experiment, Juan Soler-Company and Leo Wanner found that for *NYT* opinions articles, the most predictive features identified text as positive or negative (sentiment analysis) and identified patriotic language, as well as syntactic analysis and dictionary features. Compared to their baseline bag-of-words, content-based approach's accuracy of 67%, the accuracy of their many-feature system fared 20% better, which suggests that content-based approaches cannot suffice for such an analysis. Surprisingly, they achieved even better accuracy, at 96.7%, when they cut out many of these features. Their experiments on different combinations of feature categories showed that the best-performing combinations did not include syntactic dependencies nor sentence-based features (which include number of sentences per text, number of words per sentence, etc.). This suggested that our approach could be optimized by using dictionary-based, word-based and character-based features as well (Soler-Company and Wanner, 2014). This study in particular also aimed to develop a classifier generalizable enough to use on various types and genres of text, citing as a major issue that many similar systems over fit to the style of the training set and as a result cannot retain accuracy on texts of different style. This point informed our decision to draw from multiple corpora in our study. Though we did keep data sets separate in our training and testing, examining the results of our different experiments and how they differ across corpora provides some basic insight into how factors like style or genre cause our models to diverge.

3 Methods

3.1 Training Data

In sourcing data for use in this project, we elected to use corpora used by some of the papers we read in preparation for this project that investigate the applications and implementations of gender classification in NLP. Our hope is that by using the same corpora as prior work, our project may produce results that may exist in conversation with those papers. Moreover, given the nature, scope, and time frame of our project we felt that it would be a poor use of resources to either consolidate and tag pre-existing data or to generate our own data.

From the papers in the related work section we were able to locate and download two corpora: (1) Mukherjee and Liu's (Mukherjee and Liu, 2010) Blog Data set and (2) Soler-Company and Wanner's (Soler-Company and Wanner, 2014) New York Times (*NYT*) Opinion Column Data set.

Mukherjee and Liu's Blog data set has 3227 blog entries contained in an excel file. These entries, each stored as a row with a cell for running text and a cell for gender, are labeled with the gender of the author — either "M" or "F" — and were sourced from blogging sites around the internet such as "blogger.com [or] technorati.com" (Mukherjee and Liu, 2010). Given that this paper was published in 2010, we know that no blog entry is more recent than 2010. This is something that we kept in mind throughout the project because the ways in which people use language to communicate changes over time — especially on the internet. The blog data set is still interesting to us, however, given that blog posts are a much more casual form of speech than formal publications, but are much more robust than other forms of social media. Since these blog posts we collected broadly from all over the internet, the content of the blog posts vary dramatically. However, Mukherjee and Liu report that the blog lengths are fairly consistent, as "the average post length is 250 words for men and 330 words for women" (Mukherjee and Liu, 2010).

Soler-Company and Wanner's *NYT* data set has 5230 entries, consisting of running text from a

NYT opinion column and is stored in its own extensionless file. The gender of each entry can be obtained from the name of each file which follows the following convention generalized via regex `\d\d+_male` or `\d\d+_female` or `\d\d+_unknown`. While Soler-Company and Wanner report that "The corpus is balanced; it contains 836 texts written by more than 100 male and 836 texts written by more than 100 female authors" (Soler-Company and Wanner, 2014), the *NYT* opinion data that Dr. Soler-Company sent us a link to had 5230 entries; 3854 tagged male, 836 tagged female, and 540 tagged unknown. Ultimately, we did import 836 entries authored by men and women respectively — see table 1. Soler-Company and Wanner also write in their paper that the content of these opinion columns are "extremely multithematic, with the authors commenting on science, philosophy, ongoing political and economic affairs in the US and worldwide, etc" (Soler-Company and Wanner, 2014). We cannot guarantee the veracity of that claim since the data set we have seems to not quite match the one described in their paper.

With these two data sets successfully downloaded, we wrote functions that would import the data and store it in a homogeneous format regardless of its source.We elected to do minimal cleaning of the data because we know that non-word data helps a naive Bayes classifier perform better. We did, however, want a balanced corporea composed of 50% men and 50% women authors. The number of entries authored by women authors was the limiting factor in both data sets so we chose that number as our maximum number of entries in each respective data set for both genders. This yielded the largest gender balanced data set as possible; see the following table (table 1) for a breakdown of what we imported.

Data set	Blog Data set	NYT Data set
Total	3096	1672
Men	1548	836
Women	1548	836

Table 1: Summary of Data Import

3.2 Doc2Vec

On the other side of our approach, we explored the potential for dense vector representations in measuring the relationship between the texts in our two corpora. Using doc2vec we trained models over varying numbers of epochs (capping at 50 in the interest of runtime) to see how our model measured the relationship between vectors. On each run, we had a total of $n + 2$ vectors, where n corresponds to each text in a given corpus. The additional two vectors were representations of the "male" and "female" tags, aggregated over the course of the training cycle. With these vectors made we then gathered the top ten most similar vectors to the "male" and "female" vectors. With these machine-identified most similar vectors we were then able to check the cosine similarity between the gender vector and the most indicative texts, between the texts identified by the model, and between the gender vectors themselves. With the indices of the top ten most similar texts identified we were also then able to manually examine and annotate the documents to see if four humans found any immediate similarities. We also performed a 50 epoch trial of randomly scrambling the "male" and "female" tags to see how our model represented each corpus when the gender was less obvious, and compared results with the 50 epoch trials of the true tagging for each corpus.

3.3 Baseline Classifiers

We implemented two baseline classifiers against which to compare our custom classifier. They are the "stupid" classifier and a naive Bayes classifier.

"Stupid" Classifier: Our "Stupid" classifier is a replication of work done by Burger et al. (Burger et al., 2011). This classification scheme assumes that all documents are written by women. The accuracy is then proportional to the number of women in the given corpus. For both of our corpora in this project, we had equal entries tagged "M" and "F" which resulted in an accuracy of 50%. As it assigns a single value to any document passed to it, it is our quickest classifier with a runtime of $O(1)$.

Naive Bayes Classifier: We implemented a naive Bayes classifier which used probabilities generated from the frequency of the counts of tokens to determine whether a text is written by a man or a woman. Equation (1) calculated the Laplace-smoothed log-probabilities of a given data set; C denotes a class, V denotes the vocabulary of words in the data set, w is a word in the vocabulary, and α is the smoothing constant. Equation (2) computes the probability that a given document is authored by a man or a woman; the variables are the same from equation (1).

$$P(w) = \frac{count(w) + \alpha}{\sum_{w' \in V} + \alpha |V|} \quad (1)$$

$$P(c|document)\ P(c) \prod_{w' \in V} P(W|c)^{count(w)} \quad (2)$$

3.4 Custom Classifier

In order to create a classifier that could be fitted more specifically to author gender classification, we stayed with a roughly Bayesian approach but added additional feature engineering. We were interested in seeing which of these comprehensive features would be more or less predictive, which ruled out more opaque systems like neural nets. We also decided not to account for every token as naive Bayes does, because that number of features would be impossible to optimize without a logistic regression scheme. To predict the author's gender for a text, we multiplied the values of each feature by their probabilities, which in this case were weights — see Equation (3) and Equation (4). For Equation (3) x_n is the associated weight for a given feature, f_n.

$$P(w) = \prod_{n} x_n f_n \quad (3)$$

$$P(c|document)\ P(c) \prod_{w' \in V} P(W|c) \quad (4)$$

To find and store which combination of weights optimized the classification accuracy for each data set, we ran a bash script that conducted a randomized search for the best combination of weights for each data set at 1, 10, 100, 1000, and $10,000$ epochs. For every epoch, each weight was randomly chosen on the domain of $[1e-5, 3]$. The results of this weight tuning can be seen in figures 1 and 2.

3.4.1 Features

We determined our minimized set of features based on previous studies on features most powerfully predictive of author gender.The features, which we detail further below, included word-, dictionary-, and character-based measures including exclamation point frequency, sentiment analysis, and a lexical diversity index. The two prior studies on gender classification by Montero et al (Suero Montero et al., 2014) and Soler-Company and Wanner (Soler-Company and Wanner, 2014) guided our choice of which features to extract and include in our standard feature vector. The study by Soler-Company and Wanner in particular emphasized that using a particular combination of fewer features not only simplified a classifier, but also improved its accuracy. We restricted our features to the combination of word-based, dictionary-based and character-based features and excluded commonly-used syntactic features and sentence-based features for these reasons.

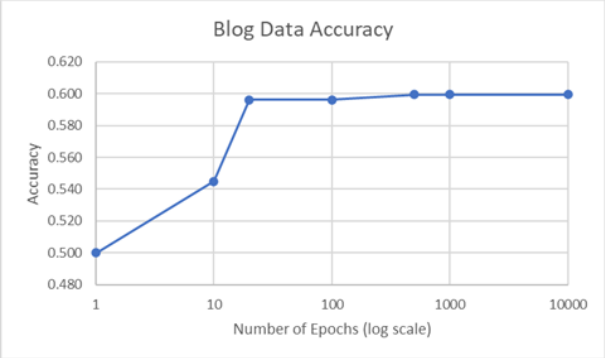

Figure 1: Weight tuning of Blog dataset

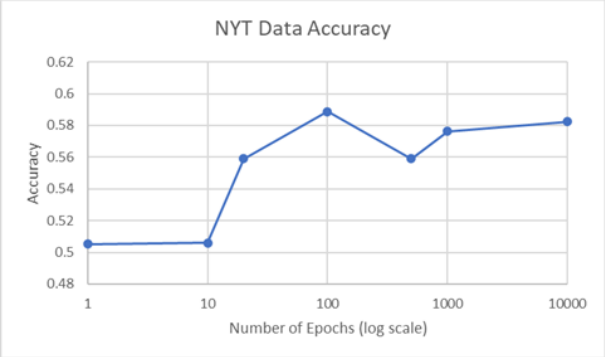

Figure 2: Weight tuning of *NYT* opinion columns dataset

3.4.1.1 Dictionary-based

Sentiment Analysis

It is a stereotype in American society that women are more expressive of their emotions than are men. We thought that this fact would be the basis of a potentially useful feature and our choice was guided by studies that found sentiment analysis, which measures the usage of emotionally charged language in a text, to be an extremely powerful predictor of author gender (Suero Montero et al., 2014), (Soler-Company and Wanner, 2014).

Scouring pre-existing work, we obtained the dictionary produced by Hu and Liu (Hu and Liu, 2004)

and Liu et al. (Liu et al., 2005), which maps 6787 tokens to 'positive' or 'negative' tags. From this data set we created a feature that measures how "emotional" a given entry by the following formula:

```
emotional = # of pos words +
        # of neg word
```

We also separated positive and negative word scores.

3.4.2 Word-based

Lexical Diversity

We included lexical diversity as a feature because we hypothesized that men and women could differ in the breadth of vocabulary used in each text. This feature produced a value for lexical diversity from the following ratio: $\frac{types}{tokens}$.

Character-based

Studies such as that conducted by Waseleski (Waseleski, 2006) have found that in computer-mediated communication like email, women use more than double the amount of exclamation points as men. Given this large difference, we hypothesized that the frequency of exclamation points in a text would be predictive of author gender.

4 Results

4.1 doc2vec

After a 50 epoch run, the most similar vector to each gender tag vector was the other gender tag vector (i,e. the vector with the highest cosine similarity to the "male" vector is the "female" vector and vice versa). For our corpora of blog data, the similarity between "male" and "female" vectors was 0.894282; this is considerably higher than the next most similar "male" text vector (0.531229) or "female" text vector (0.571662) [Table 2]. The *NYT* data set performed similarly, though slightly lower, with a similarity between gender tag vectors of 0.724581 and the most male- and female-tagged text vectors performing at 0.527802 and 0.695751 respectively [Table 2].

On our scrambled trials, in which we randomly assigned texts "M" or "F" tags prior to training, the similarity between gender tag vectors jumps up to 0.993119 for the blog data set and 0.983651 for the New York Times data set [See tables 2 and 3]. The similarity between gender tag vectors and their highest performing texts are within a similar range as those in the original set, with the highest performing vector being a cosine similarity of 0.595489 with the blog data male tag and the lowest being 0.440411 for the New York Times data female tag.

"M"	"F"	"M" Rand.	"F" Rand.
0.894282	0.894282	0.993119	0.993119
0.531229	0.571662	0.595489	0.579998
0.514797	0.545808	0.566980	0.557549
0.450171	0.500117	0.552587	0.543721
0.445003	0.489740	0.532593	0.516254
0.441501	0.489014	0.527061	0.515622
0.420636	0.486815	0.492442	0.458140
0.418215	0.480961	0.468443	0.458031
0.406718	0.474418	0.468434	0.451270
0.399370	0.466961	0.463361	0.448405

Table 2: Top 10 Similarity Rankings for "M" and "F" tags on 50 Epoch runs of our model on our blog corpus.

"M"	"F"	"M" Rand.	"F" Rand.
0.724581	0.724581	0.983651	0.983651
0.527802	0.695751	0.475684	0.440411
0.484346	0.370129	0.401962	0.385861
0.473782	0.360146	0.401515	0.366697
0.472133	0.359593	0.400358	0.361953
0.450180	0.353284	0.390316	0.359466
0.449767	0.350628	0.371446	0.356423
0.445340	0.340671	0.364802	0.349881
0.437192	0.339258	0.361523	0.348585
0.437011	0.335882	0.356209	0.346143

Table 3: Top 10 Similarity Rankings for "M" and "F" tags on 50 Epoch runs of our model on our *New York Times* corpus.

Of our document vectors that ranked as having the highest similarity with a given gender vector, not every document that doc2vec highlighted was actually labelled with the corresponding gender tag. For our blog data set, of the eighteen documents whose vectors were highlighted as having high cosine similarity to one gender vector or another only seven were in the correct category, or in other words our rankings had an accuracy of only 38.89%. The *NYT* corpus fared better with seventeen out of eighteen of the "most X-gendered" documents ranking high in comparison to the vector representing their gender tag, resulting in a much higher accuracy of 94.44%.

4.2 Custom Classifier

We split our data set by 80%, 10%, and 10% to create training, development, and test sets. When

we trained and tested our naive Bayes approach on these sets, it performed significantly better than our 'stupid' baseline did using the same data for both corpora. On the *NYT* data it achieved an accuracy of 75.3%, and on blog data it achieved 76.95% accuracy. These results can be seen in Figure 3 and Table 4. We had hypothesized that using several specific features and excluding the many term features of the naive Bayes that would lead to improved accuracy in less space on our custom classifier; however, that system was not able to exceed the performance of our naive Bayes approach for either the *NYT* or blog data sets, with a worse-than-random accuracy of 46.99% on *NYT* data and 63.34% on blog data.

Our optimized classifier weighed lexical diversity almost exactly the same for male vs. female *NYT* authors, at 1.01 and 1.00, (see Table 5) which suggests that lexical diversity did not differ significantly between the two genders in that data set. As for sentiment analysis, we found that while our optimized program weighed the 'emotion' score of male-authored texts slightly more than for female-authored texts, it gave both positive and negative word usage a greater weight for female-authored texts. It seems our optimized Bayesian classifier picked up on this, as it weighed this feature much more heavily for women (2.92) than for men (0.38) for the blog data set and similarly heavily for the *NYT* data set. Other character-based features such as those that measure frequency of periods, question marks and commas also proved to be predictive.

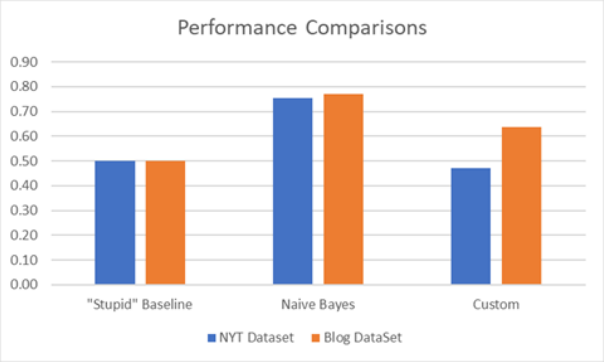

Figure 3: Results of the three different classifiers on the test sets

5 Discussion and Conclusion

Our two distinct approaches to investigating how and how accurately NLP techniques identify author gender provide insight into both the nature of Doc2Vec and Bayesian classifiers and potential differences between the average male-authored and female-authored blog post or *NYT* article.

Classifiers	"Stupid"	Naive Bayes	Custom
NYT Data set	0.5000	0.7530	0.4699
Blog Data set	0.5000	0.7695	0.6364

Table 4: Various classifier performance on the *NYT* and Blog dev sets.

5.1 Doc2Vec

The results of our doc2vec models offer up an intriguing insight into how dense vector representations of documents handle "male" and "female" tagging. In determining the relationship between gender vector tags "M" and "F", by 50 epochs we saw that the most similar vector to "M" was "F" and vice versa; we suspect that the similarity between gender tag vectors has less to do with a dissolution of the concept of gender and more to do with the fact that the gender tags, being the only two vectors aggregated from more than one individual text, are encoded similarly in our doc2vec model. This similar encoding would account for the sudden increase in similarity from text-gender tag measurements to gender tag-gender tag measurements.

Randomization of tags was the most effective tool in proving the efficacy of our doc2vec modeling. It is of note that as the gender tag similarity approaches the ceiling in our scrambled trials, though the models built off of our "true" tagged documents do not reach that height. This suggests that there was in fact some difference between the two tag categories that was lost when texts in our corpora were assigned randomly. This is reinforced by the *NYT* data set; this corpus' model produced a lower cosine similarity between gender vectors but had a considerably higher accuracy in ranking the documents most indicative of a given gender tag. From this we may extrapolate that the stronger sense of distinction between our two gender tags gave us our higher accuracy in our trials. So though imperfect, our model did in fact determine some difference in gender categories in its training. The results of our gender classifier, shown in the next section, complicate our ability to state what exactly it was that differentiated the two categories in training our doc2vec model. Though men and women are equally represented as authors in both of our

Feature	NYT "M"	NYT "F"	Blogs "M"	Blogs "F"
pos	2.114927	2.895645	1.403100	2.997508
neg	0.432695	2.934783	0.720010	2.028113
emotion	1.556579	0.178933	1.064332	1.502774
excl	0.104325	2.457673	0.379823	2.917318
commas	2.107341	0.364539	2.832119	0.634230
periods	2.874545	2.294893	2.431861	0.129944
questions	2.837059	0.975527	2.287983	1.736809
lexical	1.012764	0.996507	1.138903	2.950337

Table 5: Final Feature Weights

corpora, they may be over- or underrepresented in the context of the topics that they are writing about. As dense vector representations such as doc2vec are not human interpretable in what prompts decision making, we cannot state for certain what parameters define the categories that our model made from the "M" and "F" tags. However, though, it is not negligible that our doc2vec model experiments allowed us to quantifiably stratify different tag categories for examination and comparison.

This is not to say that our doc2vec model is at all comparable to a proper classifier. The only document that was incorrectly categorized by our doc2vec model in the *NYT* data set was in fact marked as the "most female" document in the entire corpus with a cosine similarity of 0.695751. Even more intriguing is that this document was *also* identified as the most similar vector to the "M" tag with a cosine similarity of 0.527802. We underwent human annotating of this text in an effort to see what made it stand out as high as it did. Of note is that it covers the historical period after the end of the American Civil War; one of the features explored within our gender classification task identified "patriotic" words as "masculine." This falls in line with the fact that this document was male authored. However, the much higher cosine similarity to the "F" vector raises questions of the effectiveness of our doc2vec model in accurately measuring gendered language and producing meaningful distinctions between our two categories.

5.2 Classifiers

Our modified Bayesian classifier identified author gender with a worse-than-random accuracy of 46.99% on *NYT* data and 63.34% on blog data. It performed poorly in comparison to our naive Bayes classifier, which achieved around 75% accuracy for both texts. The minimal-feature model classified author gender of our *NYT* data roughly 30% less accurately and blog post author gender 12% less accurately than our naive Bayes model. We attribute our custom classifier's weak performance to the fact that we replaced the naive Bayes raw frequency counts with binary occurrence features.

Our feature-based custom classifier discriminated author gender of blog posts significantly better (16%) than it did for *NYT* columns. We attribute this discrepancy to the fact that blog posts have much more range of style and emotional expressiveness than *NYT* columns, which are standardized in style and generally restrained in strong language. Our experimental results suggest that in author-gender classification, especially in cases where stylistic expressiveness is limited (as in the case of newspaper articles) word-based features are a critical measure. As mentioned in our introduction, word-based features often are subject to topic bias when data includes works written about various content areas. As one of our data sets, which draws from *New York Times* opinions articles, includes texts covering a wide array of content areas such as science, the stark drop in accuracy between naive Bayes and modified Bayes suggests that women are over-represented in *NYT* opinions articles about certain topics. It also suggests that topic bias artificially inflated the accuracy of the naive Bayes model.

In regards to our optimized feature weights, without a logistic regression test, we cannot draw strong conclusions about gender performativity in writing. However, despite the mediocre performance of our system, the fact that the weights of individual features improved our system's accuracy by roughly 10% for the blog data set allows us to postulate that they picked up on some predictive features. The results tend to confirm prior studies

finding that identified usage of exclamation marks and emotionally-charged language are predictive of author gender, but find no association between author gender and lexical diversity.

6 Future Work

In future work, we would solely focus on developing a custom classifier. We have explored everything we want to explore with doc2vec, aside from perhaps expanding the number of epochs to see how that improves rank accuracy. We would focus future methods solely on building a better classifier.

We would compare our proposed classifier to a second custom classifier that appends our binary features to the original naive Bayes model of terms as features. With this new classifier, we would then train it over a series of progressively longer epochs to find the most optimal weight values. We would also experiment with additional features; this could include adding part of speech frequency and comparing document terms with dictionaries of patriotic language and curse words, for example. In addition, a logistic regression test would be useful in future studies to verify the associations that our feature weights and doc2vec rankings suggested and explore new ones.

Following the lead of Soler-Company and Wanner, it could also be interesting to mix data sets and see how our proposed new classifiers would perform being trained from one source and tuned on another. NLP tools are still often specifically tailored for one type or source of text, and it could be interesting to see if there exists any middle ground in which we could build a more general classifier (Soler-Company and Wanner, 2014).

While text data sets that include author gender are limited, many social media users now identify themselves by their gender in their bios. These could provide interesting, if not entirely accurate, information when run through a gender classification system. Reddit data, where users identify gender and age in many posts, would provide a fascinating case study. Finally, expanding this study to include multiple genres and time periods, such as Victorian romantic literature vs. modern sci-fi or letters vs. emails, could suggest differences in societal performances of gender within those different contexts.

—

References

John D. Burger, John Henderson, George Kim, and Guido Zarrella. 2011. Discriminating Gender on Twitter. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 1301–1309, Edinburgh, Scotland, UK. Association for Computational Linguistics.

Minqing Hu and Bing Liu. 2004. Mining and summarizing customer reviews. In *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '04, pages 168–177, New York, NY, USA. Association for Computing Machinery.

Brian Larson. 2017. Gender as a Variable in Natural-Language Processing: Ethical Considerations. In *Proceedings of the First ACL Workshop on Ethics in Natural Language Processing*, pages 1–11, Valencia, Spain. Association for Computational Linguistics.

Bing Liu, Minqing Hu, and Junsheng Cheng. 2005. Opinion observer: analyzing and comparing opinions on the Web. In *Proceedings of the 14th international conference on World Wide Web*, WWW '05, pages 342–351, New York, NY, USA. Association for Computing Machinery.

Arjun Mukherjee and Bing Liu. 2010. Improving Gender Classification of Blog Authors. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, pages 207–217, Cambridge, MA. Association for Computational Linguistics.

A. Sboev, T. Litvinova, I. Voronina, D. Gudovskikh, and R. Rybka. 2016. Deep Learning Network Models to Categorize Texts According to Author's Gender and to Identify Text Sentiment. In *2016 International Conference on Computational Science and Computational Intelligence (CSCI)*, pages 1101–1106.

Juan Soler-Company and Leo Wanner. 2014. How to Use Less Features and Reach Better Performance in Author Gender Identification. page 5.

Calkin Suero Montero, Myriam Munezero, and Tuomo Kakkonen. 2014. Investigating the Role of Emotion-Based Features in Author Gender Classification of Text. In *Computational Linguistics and Intelligent Text Processing*, Lecture Notes in Computer Science, pages 98–114, Berlin, Heidelberg. Springer.

Carol Waseleski. 2006. Gender and the Use of Exclamation Points in Computer-Mediated Communication: An Analysis of Exclamations Posted to Two Electronic Discussion Lists. *Journal of Computer-Mediated Communication*, 11(4):1012–1024.

—