# Predicting Author Gender for Yelp Reviews

Keziah Plattner Laura Hunter Lilith Wu

## 1 Introduction

What makes a piece of writing seem "feminine" or "masculine"? Are there concrete features that human readers pick up on–and can they be used to predict author gender computationally?

Previous literature suggests that there are. Many studies have sought to predict author gender for social media. Twitter's company blog noted the usefulness of allowing marketers to promote products by gender, and claimed 90% accuracy with their algorithm [1]. A study using sociolinguistic and lexical features with a stacked SVM model achieved 72% accuracy in classifying Twitter user gender [9] and another study achieved a remarkable 88% accuracy on blogger gender classification with SVM using stylistic and word features, POS sequence patterns, and ensemble feature selection. Despite these successes, researchers are sometimes unable to replicate other studies' findings, and the differences between formal and informal language, American vs. European populations hint at external factors that add complexity [1, 2].

Our study differs from previous work in the nature of the text we are examining– reviews from Yelp. Reviews are of particular interest because being able to identify author characteristics can help reveal hidden customer demographics and demonstrate product success and failure within these groups. While Yelp users are identified by name, users on other review platforms may be less likely to use real names or be completely anonymous. Being able to confidently guess gender from text would provide valuable information for marketing and evaluation.

It is important to note that many studies, including ours, over-simplify gender into binary categories, and make an assumption that gender identity and gender expression match. We recognize that this is not completely accurate.

## 2 Methods and Data

### 2.1 Data and Database Setup

We use the Yelp Academic Database, which consists of 330,071 reviews of 250 businesses located closest to 30 American universities. For each review, we have access to information like location, business, rating, review text, and reviewer first name. This last piece is crucial, since we do not have direct access to the reviewer genders. Instead, we use the Genderize.io API [4] which converts a given first name to guess of gender, a probability score, and number of data points. We use only non-empty reviews for which the gender probability score is above 90% and based on at least 20 examples, leaving us 186,824 remaining reviews.

The JSON dataset was parsed and stored in a Sqlite database. Name-gender associations and extracted features were also stored in this dataset so we could easily query and inspect the 'shape' of the data.

### 2.2 Baseline and Oracle

We established a classification baseline by implementing a very simple Naive Bayes model run on unigram bag-of-words, obtained by splitting each review on whitespace with no additional processing. We ran this baseline classifier on a very small set (n = 50)[2]. This baseline achieved 86% training accuracy and 50% dev accuracy. Our small sample size and untuned model grossly overfit the training data and did not generalize well.

Our desired goal was, of course, to implement a classifier which classifies all reviews perfectly. However, to give ourselves a more reasonable goal, we set our oracle model as that of a human reading the free text of each review and assigning author gender based on that text.

### 2.3 Classifier selection

To select the strongest classifier type for our data we ran a variety of different classifiers available on the scikit-learn Python library [12] on the basic

[1] They only assign gender when their algorithm is confident; 90% accuracy is for these assignments only. [13]

[2] The small sample size for this classifier was due in part to us manually selecting reviews and inputting the review information, since our database was not yet set up.

| Classifier Type | Input/parameters | Training accuracy | Dev accuracy |
|---|---|---|---|
| K-means | Unigram bag of words model, stop words removed | 68% | 61% |
| Bernoulli Naive Bayes | Unigram bag of words model, stop words removed | 75% | 65% |
| Linear SVM | Unigram bag of words model, stop words removed | 79% | 68% |
| Kernel-based SVM (Fig. 2) | Extracted word features only | 56-95% | 56-60% |
| Linear SVM | Extracted word features only | 66% | 64% |
| Linear SVM | Unigram Bag of Words, stop words removed, plus extracted word features | 75% | 71% |

**Table 1:** *Summary of results from trying various classifier models, run on bag of words and extracted features*

bag of words feature of the data. Rather than simply splitting on spaces, we used methods from the scikit-learn feature_extraction library to improve bag of words construction. The main classifier types we ran on the bag of words data were K-means, Bernoulli Naive Bayes, and Linear SVMs. Additionally, we experimented with a variety of SVMs with non-linear kernels. These were run on a set of extracted features (discussed in section 3). We summarize the best results obtained with each classifier type in Table 1, and a breakdown of various kernel performance in Figure 1.

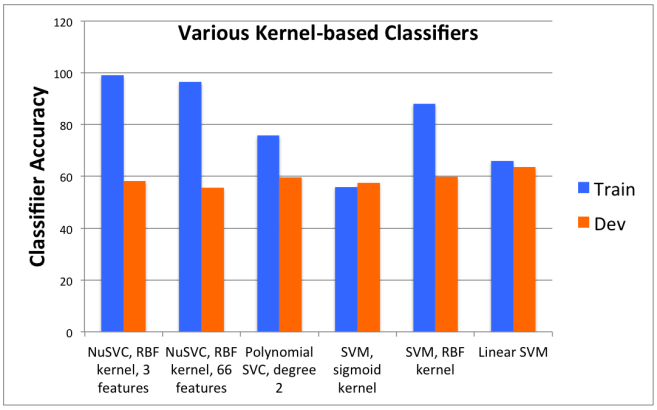



**Figure 1.** *Summary of results from trying various non-linear Kernel-based classification methods*

K-means and Bernoulli Naive Bayes did not classify as accurately as the Linear SVM classifier on the same input. We were excited to try using non-linear kernels after seeing the success with the Linear SVM, however, while training accuracy was very strong with these models, we actually saw the strongest dev classification with the linear SVM we had initially employed. In addition, the linear SVM had very little variation in classification error between train and dev sets, which boded well for the adaptability of the linear SVM to previously-unseen patterns in new data. Consequently, we decided to focus our work on refining this Linear SVM model.

# 3 Feature Extraction

To capture stylistic features, we wrote scripts to extract features from review text and add them to various features tables in our database. It is important to note that throughout this process we use only the review text and no extra information. We embarked on the following iterative feature selection process:

1. Identify new candidate features by looking at reviews and finding features used in other text classification tasks
2. Implement Python scripts, at times drawing from existing libraries, to capture the new features
3. Test success of new features (combined with old features and/or with the bag of words)
4. Use accuracy statistics of the newly-augmented classifier on the train and dev datasets to determine usefulness
5. Look at reviews which were misclassified under the current best classifier to identify new candidate features, or refine existing ones

## 3.1 Bag Of Words

Our initial classifier had utilized a very basic unigram bag of words model, where each review was split on spaces into its component words. To generalize better from individual reviews, we experimented with the application of various scikit vectorizers to clean and tokenize based on words. The vectorizer we ended up using for the majority of our classifications was the sklearn's TfidfVectorizer. We then further experimented to find the most informative bag of words.

**Use of N-grams and skip-grams:** We had to decide how many tokens we wanted to count as each feature. There is a tradeoff between the additional information gained by using a higher order N-gram model, and risking overfitting the training set. Using bigrams resulted in an extreme overfit of the training data, whereas using unigrams as well as bigrams had a high train accuracy but also improved our dev accuracy. Another thing which we wanted to consider[3] was skip-grams, or looking at words which were separated by up to K positions in the sentence. We thought this could lend some of the context found in bigrams, without being quite as rigid. Introduction of these into our classifier resulted in markedly different results on train and dev sets. Specifically, unigram features only had a much lower training accuracy, whereas features encapsulating bigrams had exceptionally high train accuracies due to overfitting. It is worth noting that all of the top 100 highly-weighted features for this classifier were unigram features, not bigram features.

| | Unigram | Bigram | Unigram + bigram | Skipgram |
|---|---|---|---|---|
| Train | 82.51 | 99.71 | 99.01 | 99.96 |
| Dev | 69.91 | 64.76 | 70.79 | 67.86 |

**Handling of stop words:** One known challenge of working with bag of words-style data is the propensity of common "stop words"– such as "and," "the," and "a"– to have very high counts, artificially inflating their importance in the final classification scheme. We used the built-in function in the TfidfVectorizer, which selectively removes English stop words. We also limited words that had a frequency higher than 85%, so we could eliminate other commonly used words that weren't in the English stop words corpus.

**Tokenization:** Another component of creating our bag of words for classification was how we tokenized the review text. Our first pass, splitting on spaces, had left punctuation present as artifacts in the isolated words. This led to incorrectly creating several features corresponding to a single word depending on what punctuation surrounded it. We found an NLP parser created for use on Twitter datasets [8]. Cutely named "Twokenize," this tokenizer component increases correct classification of tweets by handling cases of excessive punctuation, emoticons, abbreviations, and joined words that traditional tokenizers do not correctly segregate. Because we had identified the presence of netspeak and emoticons in our dataset, we hypothesized that using Twokenize could result in a more representative bag of words model. However, classifying using a bag of word model and Twokenize, versus the identical model with the TGDIF vectorizer, resulted in almost identical performance on the dev set.

**Clustering:** Another thing we wanted to do was gain knowledge about writing style, sentence structure, and relative proportionality of different POS components abstracted away from the context of the individual review. Because the business and restaurant reviews we were sampling from were so diverse, generalizing some of these aspects based off of specific words had limited applicability. While we experimented with tagging various parts of speech using Python NLP libraries, the high levels of typos, slang, and netspeak meant that a sizable chunk of the words seen in reviews remained untagged, and didn't really result in any increase in classifier accuracy.

We then looked at alternate tagging approaches. Using a C++ implementation of Brown clustering [6] we first tokenized our training reviews using the Twokenizer mentioned above, getting us a clean segregation of individual words and punctuation features. We then ran the Brown clustering algorithm, thresholding the number of occurrences a word must have in the corpus in order to be able to placed in a cluster and outputting a file containing (ClusterID, Word) objects. Because each cluster was composed of words of similar meaning or function, it was now possible to use them as proxies for more formally-tagged POS objects. Running a linear SVC on the cluster tags rather than the words themselves, however, resulted in a low classification accuracy, ranging from 64-67% (depending on whether bigrams or unigram features were used). Running a linear SVC on clusters as well as a unigram bag of words model resulted in elevated accuracy, of 69% dev. Running on 400 word clusters, our final list of extracted features, and a unigram bag of words model resulted in 76.34% training and 71.71% dev accuracy. Thus we see that adding word clusters, at least to this model, does not result in an appreciable performance gain.

#### 3.1.1 Stylistic Features

**Initial Feature Selection:**

In addition to tuning the bag of words model to our specific classification problem, we extracted a number of text-based stylistic features. For our initial feature set, we extracted a set of general baseline features. These included number of characters and words, average word and sentence length, number of

[3] Thanks to Ice for this feedback.

unique words, percent of capital letters, number of non-word-initial capitals, percent of all-capital words, number of "you" words, number of "I" words[4], number of new lines, and number of punctuation characters.

Previous studies have indicated that men and women use parts of speech, certain vocabularies, expressive spellings, and internet lingo differently. Men tend to use numbers and statistics more; we extracted the count of numbers in a review[5] [2]. Women tend to use pronouns more while men use noun specifiers (particularly "my" in reference to girlfriends and wives) [1, 11]; we added separate counts for pronouns and noun specifiers. Other studies had also observed that women tend to use more emotional words [2], so we added counts for positive and negative emotional terms. Text emoticons also tend to be used more heavily by women (though in at least one study this has not been the case)[6] [2, 11, 16], so we used those as features as well. Rao et al. identified features of excitement such as repeated exclamation marks and "puzzled punctuation" (a mix of ! and ?) as occurring nearly twice as often in women's tweets [9]. We included features for different lengths of these, bucketing by single, two, three to six, and more than six exclamation marks, for instance, to try to capture degrees of expressed excitement. Backchannel sounds or disfluencies have also been associated with female writers. These include "words" like "ah", "oh", "uh", "ugh", and "hmm." We included a count for these types of words as a group.

We hypothesized that referencing a gendered significant other (SO) could serve as a useful predictive feature. We came up with a lexicon of "female's SO" and "male's SO" words, such as "husband" and "boyfriend" to designate a female's partner, and "wife" and "girlfriend" to represent a male's partner. After running classifiers which used the bag of words model, we identified other heavily-weighted words which were denoted relationship status, such as "bf," "gf," "hubby," and "wifey." Adding these to our SO categories made the features reflective of the language we were seeing in the reviews. The SO feature served as a strong predictive feature for classifiers which did not use the bag of words model.

Previous work [7] had shown that the terminal two or three characters of a word could indicate reactive, descriptive reviews. We implemented this feature by adding columns to our database of features for endings "able", "al", "ful", "ible", "ic", "ive", "less", "ly", and "ous".

After running a linear SVM on this initial feature set, we noticed that certain features bubbled up, or were assigned high weights by the classifier, while others bubbled down. Emotional words were weighted extremely low, which surprised us initially, but made sense in light of the fact that reviews are inherently opinionated, and reviewers are self-selecting in that people might be more likely to write reviews when they have strong emotions about the business.

Note that these features were extracted using scripts we wrote, and although we describe the target features here, and checked small sets of data, areas where the extraction might have failed were impossible to manually verify across such a large dataset.

**Modification of captured features:** We noticed that features related to capitalization were repeatedly assigned high weights so we worked on improving the granularity of these features. From our original capitalization features, we kept a boolean indicating presence of capital letters and percentage of capital letters, and added word "shape" features: counts and percentages of capitalized, lowercased, uppercased (all capital letters), title-cased, and nonsense-cased (non-uppercased words with capital letters in the middle). Another decision we had to make was whether to store word and stylistic features as their number of occurrences, their number of occurrences normalized, or a simple binary marker indicating feature presence or absence. Additionally, some features seemed to make the most sense as interaction terms: for example, we created a feature recording the number of "you" words versus the number of "I" words.

We postulated that, since many words in the lexicon were weighted heavily either male or female, it could perhaps be sufficient to merely record the presence or absence of the word in the review, rather than keeping track of how many times a given word appeared in the review.

Converting all word features to such a binary model, however, resulted in a decrease in accuracy for both train and dev classification. In a similar vein, we tried normalizing the count of any given feature to the review length by dividing its count by the total number of words in the review. Going from counts

[4] "You" words being "you", "you'd", "you're", "you've," and "I" words being the analogous first-person forms.

[5] This was a count of arabic numerals, since the space of forms that written numbers can take in informal writing is quite large and unwieldy to handle.

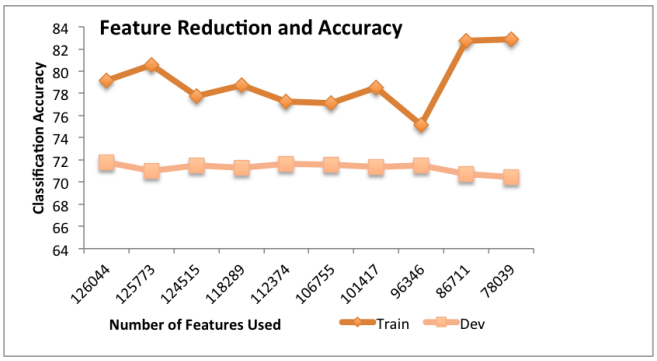
[6] The demographic of this study was specifically teenage bloggers, however, so that or the small sample size (100) may have influenced their results. [5]

| Length requirement | Training Accuracy | Test accuracy |
|---|---|---|
| None | 78.08 | 71.79 |
| > 500 characters | 81.29 | 74.96 |
| > 1000 characters | 81.29 | 74.96 |
| > 250 words | 81.16 | 73.62 |

**Table 2:** *Results of best classifier on different text lengths*

to normalized counts of words and features reduced classification accuracy by almost 10%.

**Removal of artifact features:** We wanted to try to remove features whose predictive power was limited. One way we did this was by capturing the weights assigned to each feature or feature class, and removing those with frequently low weights.
Another method we used for normalizing the number of features used was L1 normalization on the feature vectors. L1 normalization should make vectors sparser and surface features of greater importance, but for our classification task the L1-reduced vectors saw no marked improvement. Identical accuracy on the dev set (73.4%) was seen when comparing a Linear SVM using bigram features and L1 to the same classifier which normalized with L2, and the same lack of impact was seen in the case when unigram features were used, with the two classifiers having the same accuracy within 0.1%.




Recursive Feature Elimination (RFE) was the third method we tried to reduce the number of features. We had thought that RFE would isolate and reveal the most important features being used by our classifier, obtaining a high level of classification by not being swayed by artifacts present in the data. We used RFE from scikit-learn's recursive feature selection module [12] on our bag of words classifier. Interestingly, we actually noticed increased overfitting as the number of parameters decreased, while classification accuracy on the dev set did not really change.

**Subsetting Reviews:** Looking through misclassified reviews, we noticed that many very short reviews consisted only of placeholders (sequences of periods, an emoticon, etc). We instantiated various cutoff lengths of 250, 500 and 1000 characters. These cutoffs increased our dev accuracy as our word cutoff increased. Training and running our classifier on reviews which contained between 100 and 500 words boosted classification accuracy significantly, getting a dev accuracy of 74.79%.

To check for confounding external factors, we tried training the data on buckets by location, by business, and by category. None of these provided gains, achieving accuracies of 65-70%, and most did worse, probably due to the significant reduction in data for any given "bucket." In addition, since the data was gathered from university areas, we postulate that many of the reviewers in college towns may have come from varying geographic locations anyway, reducing our chances of seeing how language choice differs by location.

# 4 Analysis

Our best classifier uses unigram features and additional stylistic features as previously described.

## 4.1 Comparisons to manual classification

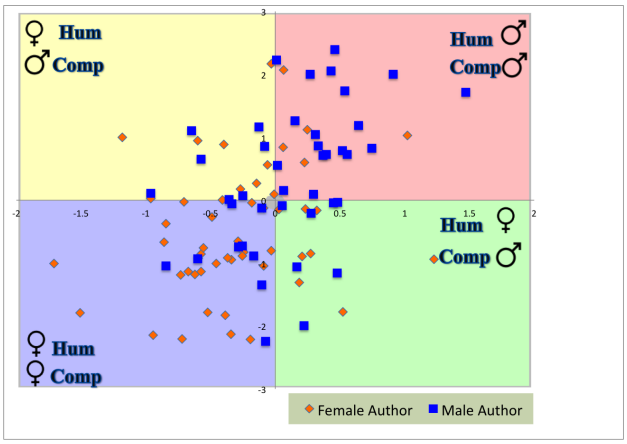



**Figure 2.** *Human versus computer gender predictions*

In addition to looking at accuracy on the test set, we had defined our goal as classification success compared to the "gold standard" of human reviewers. Therefore, we wanted to see how well our final

classifier was able to assign author gender to reviews compared to how well human reviewers did at the same task (see Figure 2). The results were very positive. Out of 100 randomly selected reviews, our classifier misclassified 28, and a group of human classifiers misclassified 29. 16 of these misclassifications were on the same reviews, which shows that there is some overlap in the mistakes that people and our classifier make.

| Female | Weights | Male | Weights |
|---|---|---|---|
| husband | -4.86% | man | 2.91 |
| boyfriend | -4.64% | wife | 2.72 |
| bf | 5.4% | buddy | 1.83 |
| hubby | 7.4% | percentCaps | 1.76 |
| percentAllCaps | -2.76% | barber | 1.58 |
| lovely | -2.59% | value | 1.46 |
| yummy | -2.44% | place | 1.32 |
| adorable | -2.35% | notch | 1.30 |

Top weights incl. extracted features

## 4.2 Misclassification Analysis

Overall, our classifier did worse in classifying men over women, even within the training set [See fig]. We investigated our misclassifications to determine why this occurs, and there seem to be many factors at play in correctly classifying a review as male.

**Word weight:** Overall, female associated words had a much higher weight than male associated words–aside from the top 15 weights, many of the highly weighted male words were between 1 and 0, while the many of the weights of the female associated words were between 2 and 0. As a consequence, there were several reviews that had male associated features, but the addition of a few heavily weighted female features led to misclassification.

**Categories:** We have very high accuracy with predicting on female-associated categories and categories, but surprisingly, a low accuracy with predicting male-associated categories. (See Fig. ) After analyzing the numbers, it starts to make sense–we have several female-associated businesses that are frequented almost exclusively by women, while the male-associated businesses have a much more even gender breakdown. We are more likely to pick female over male (with word weight and within a gendered category), but since the gender breakdown favors women, our accuracy goes up. However, in the male associated categories, we are more likely to pick male over female, yet since the gender balance is less skewed we perform worse.

**Lack of defining features:** As mentioned above, we have more heavily weighted female features than male features. It appears that what almost defines a male review is the lack of female attributes, while what defines a female review is the presence of one or more female attributes. This results in several reviews with gender-neutral language being classified as male (Lucy L.), or a generally male-associated review with a female characteristic being misclassified (Mike C.). This lack of strong male features was an interesting problem we dealt with throughout the process, and bears further investigating. Perhaps there are more subtle extracted male features that were not captured in our process, and can be focused on in future work.

*Misclassified Lucy L., despite gender-neutral language* i've tried alot of fro-yo, from berkeley to seattle, I think this has the best original fro-yo. It doesn't have other flavors like it's counterparts but the original has the special tangy taste that mimics the yogurt drinks very well.

*Misclassified Mike C. thanks to punctuation and caps.* I actually loved the Birch Aquarium. I could spend hours in this place if it were larger! The place was perfect, but it's like I left and just felt short changed because they were so small. Call me selfish on my shellfish! LOL! The place is well kept, and in a very beautiful and secluded spot in San Diego. I highly recommend it for a visit!

| Actual Gender | Training Error | Dev Error | Testing Error |
|---|---|---|---|
| Men | 28% | 37% | 38% |
| Women | 14% | 19% | 19% |

| | Error | Gender Breakdown | % Misclassified |
|---|---|---|---|
| IT Services & Computer Repair | 35.7% | 53% Men, 47% Women | 30% Men, 42% Women |
| Videos & Video Game Rental | 31.5% | 53% Men, 23% Women | 41% Men, 47% Women |
| Nail Salons | 5.4% | 6% Men, 94% Women | 61% Men, 2% Women |
| Day Spas | 7.4% | 9% Men, 91% Women | 60% Men, 2% Women |
| Gay Bars | 30.2% | 58% Men, 42% Women | 34% Men, 24% Women |

## 4.3 Expected Misclassifications

We have some reviews that are quite reasonably misclassified.

*Misclassified man*: YUMMY. The idea of cookies between ice cream is GENIUS. Perfect for the hot day, but prepared to make a little mess. ;) ... it was OMGSOGOODILOVESTRAWBERRY amazing! ...If you get lucky, you can get the cookies that come fresh out of the oven, which makes it even BETTER!

*Misclassified woman*: Star-worthy: Made-to-order burritos with decent quality chicken and whole black beans. Pretty cheap (about the same as Baja and La Salsa). Not so star-worthy: Burritos are served COLD (yuck!). Rice is bland and often undercooked. No salsa bar.

For the first one, the use of emoticons and all-caps clearly indicate female, but is incorrect. Same with the second one–short sentences tend to be attributed to men. In these cases our classifier is performing as well as a human reader can, and not much can be done to avoid these errors.

While SO was a useful feature, it proved to be a double-edged sword in some cases. (See fig above for error rates.) LGBTQ reviewers might be misclassified if they mention their SO, since the weighting of SO terms is strongly heteronormative. If we were able to also detect features of text that indicate the reviewer's orientation and reverse or otherwise modify the weight of SO indicators, we believe that these terms could still be very useful. However, few reviewers mentioned their sexual orientation in the reviews, so there was little we could do to counter this misclassification.

# 5 Discussion and Next Steps

## 5.1 Limitations

**Data:** It is important to recognize the limitations of our generated "golds." Though many names are distinctly gendered, and we chose to use only those names with very high probabilities of being either male or female. However, anyone who has a predominantly opposite-gendered name will be mislabeled. Consider, for example, "Chris," categorized male with 91% probability and 4511 examples. Most users named Chris probably are male, but any female users who go by or are named Chris will be mislabeled. This means that we likely have some false negatives and false positives.

**Sample Length:** As we saw, review length has a substantial impact on how well the classifier does. We would suggest that future work on the kind of dataset in which text length varies widely try using separate classifiers for ranges of text length. We noted that there were gains in accuracy when limiting data to be similar length, but we are curious whether this is a) because people express themselves differently in different text lengths, in which case separate classifiers may work better, or b) because shorter reviews simply contain more noise, or are written with less care, are less personal or expressive, etc. (in which case, they simply don?t help). Bucketing by length greatly reduces our dataset for any given bucket, so we do not make any claims on this front.

Length may also be a factor in why observations from Twitter data did not seem as applicable to our data. Due to its micro-blogging format, Twitter's users need to find ways to express themselves powerfully and succinctly. As gender performance plays into personal expression, the gender cues in tweets might be more "concentrated"– that is, every character and every word carries more weight. These same features may become "diluted" in longer forms of writing. Additionally, because we were classifying per review and not per user, we were not able to capitalize on multiple reviews by the same user. Aside from having access to more content lengthwise, multiple reviews could also give a more comprehensive snapshot of a user's writing style and help correct for words and style markers that appear because of the specific context of a given review.

**Location:** Using a dataset with a wider distribution, or converse a more concentrated one, might yield different results. One of the confounding factors we posited was geographic location, which can affect speech and writing patterns. As mentioned above, training classifiers on data from a specific location didn't yield any increase in accuracy, but the businesses from which this particular dataset are drawn are near universities, so there is a chance that these populations simply have a higher variance in where they come from or grew up, so bucketing by business location would not accomplish the kind of geographic filtering we were aiming for.

**Age:** Another potential confounder is age. Many previous studies that have looked at author gender have also looked at age. Since we have no information about reviewer's ages, we weren't able to test whether this was a confounding factor. However, other studies have identified features that we used as also being indicators for age, such as "internet speak" [11] and "alphabetic repetition" or creative spelling [9]. Particularly if the by-gender usage ratio is different among different age groups, it is possible that age might skew our results, depending on its distribution in our dataset.

## 5.2 Challenges

**Feature Granularity:** It isn't always clear how granular to make features - should emoticons as a group be one feature? What about "equivalence"– should "hmm" count the same as "hm" or "HMM",

or ";)" the same as ";-)"? For example, our classifier treats "internet speak" as one feature, but one study observed that women tend to "lol" while men tend to "lmfao" [7]. This happened not to be the case in our data, but nonetheless illustrates the importance of choosing the right granularity for a given feature.

**Reviews:** In the course of building our classifier, we made some observations about the "medium" of reviews which may be useful for future work. Reviews differ from tweets and blogs in that the latter two generally build personas and are centered around the user/author, while reviews have less of an "identity coherence" – that is, even with sites which have user pages aggregating their reviews, the focus is not on those user pages, but on the business or product pages. Twitter accounts and blogs can have fans and followers; reviewer accounts, less so. In light of this, we may reveal less about ourselves in the way we write reviews than the way we tweet or blog.

Additionally, as other authors on the subject of gender identification have noted [11], we must be careful in applying the findings of one study to another context. Reviews are a very particular type of writing. Even reviews of businesses versus reviews of products, or reviews of particular types of businesses and products may have different characteristics.

Lastly, with a unigram bag-of-words approach, certain words may become heavily weighted female or male because of a particular product or business in the dataset. The earlier example of the classifier run on only nail salons, for instance, shows how gender-biased certain businesses can be. "Nail" gets heavily weighted female, though in another context it could refer to the type of nails used in construction.

### 5.3 Improvements and Next Steps

We were particularly interested in the strength of significant-other (SO) indicators in reviews and the implications for classifying LGBT reviewers. This also begs the question if orientation can be detected through features of writing–particularly whether two people with different orientations but the same-gendered performance (e.g. gay male, straight female, performing as feminine) are distinguishable by their writing; this could be an interesting topic for future work.

A second problem with SO terms is that the writer may not be referring to their own SO. A review that mentions another couple, for instance, may use phrases like "his wife" or "the wife." There isn't enough context with these bigrams to tell whether the latter refers to the writer's wife. Being able to identify these other-referring SO terms and decrease their effect on classification could improve accuracy. Some ways to approach this might be to look at bigrams around SO terms, to only count SO terms if preceded by "my", or look at the number of "we" terms in the review as a rough indicator of whether it's likely the writer was there with a SO.

We also believe that more discriminating feature selection and elimination—particularly those for which male and female usage do not differ significantly—would improve the classifier. Although limiting high frequency words and manual elimination of certain features helped a little, it made a very small difference in the feature space. We regret that recursive feature elimination did not provide improvement, and that we do not have time to try $\chi^2$ or information gain methods, which other studies have found useful [15].

## 6 Conclusion

Ultimately, it's important to remember that in the task of gender classification, we are essentially classifying gender expression rather than gender identity. Despite error rates on our expected misclassifications, the usefulness of classifying gender expression can still have many applications, including the potential impact of businesses trying to understand their demographic.


**Figure 3.** *Words weighted heavily female.*

# References


[1] Aragamon, S., Koppel, M., Fine, J., Shimoni, A.R. (2006) Gender, genre, and writing style in formal written texts. Interdisciplinary Journal for the Study of Discourse. Volume 23, Issue 3, Pages 321?346, ISSN (Online) 1613-4117, ISSN (Print) 0165-4888, DOI: 10.1515/text.2003.014

[2] Bamman, D., Schnoebelen, T. (2014) Gender identity and lexical variation in social media. Journal of Sociolinguistics 18(2). 135-160. doi: 10.1111/josl.12080

[3] Fleishman, Glenn. An algorithm to figure out your gender. BoingBoing. 1 Sep. 2014. Web. (http://boingboing.net/2014/09/01/twitter-uses-an-algorithm-to-f.html)

[4] Genderize.io. (http://genderize.io/)

[5] Huffaker, D.A., Calvert, S.L. (2005) Gender, Identity, and Language Use in Teenage Blogs. Journal of Computer-Mediated Communication 10: 1?10. doi: 10.1111/j.1083-6101.2005.tb00238.x

[6] Liang, P. C++ implementation of the Brown word clustering algorithm (2012), GitHub repository, https://github.com/percyliang/brown-cluster

[7] Mukherjee A. and Liu B. (2010) Improving gender classification of blog authors. In Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing (EMNLP '10). Association for Computational Linguistics, Stroudsburg, PA, USA, 207-217.

[8] Ott, Myle. Python port of the Twokenize class of ark-tweet-nlp. Github repository, https://github.com/myleott/ark-twokenize-py

[9] Rao, D., Yarowsky, D., Shreevats, A., and Gupta, M. (2010) Classifying latent user attributes in twitter. In Proceedings of the 2nd international workshop on Search and mining user-generated contents (SMUC '10). ACM, New York, NY, USA, 37-44. doi=10.1145/1871985.1871993

[10] Santosh, K., Bansal, R., Shekhar, M., Varma, V. (2013) Author Profiling: Predicting Age and Gender from Blogs. Notebook for PAN at CLEF 2013

[11] Schwartz, H.A., Eichstaedt, J.C., Kern, M.L., Dziurzynski, L., Ramones, S.M., et al. (2013) Personality, Gender, and Age in the Language of Social Media: The Open-Vocabulary Approach. PLoS ONE 8(9): e73791. doi:10.1371/journal.pone.0073791

[12] Scikit-learn: Machine Learning in Python, Pedregosa et al., JMLR 12, pp. 2825-2830, 2011.

[13] Underwood, A. Gender targeting for Promoted Products now available. The Twitter Advertising Blog. Twitter. 25 Oct. 2012. Web. Dec. 2014. (https://blog.twitter.com/2012/gender-targeting-for-promoted-products-now-available)

[14] Yelp. Yelp's Academic Dataset (2014) Available from https://www.yelp.com/academic_dataset

[15] Yang Y. and Pedersen J. O. (1997) A comparative study on feature selection in text categorization. In Proceedings of the Fourteenth International Conference on Machine Learning (ICML), 412?420.

[16] Zimmer, B. How Twitter language reveals your gender — or your friends'. The Boston Globe. 4 Nov. 2012. Web. (http://www.bostonglobe.com/ideas/2012/11/03/how-twitter-language-reveals-your-gender-your-friends/e68H6Z0Z2GAfnJ6UjU3IxO/story.html)