Inferring Gender from Colloquial English Text: A Machine Learning Approach Using Facebook Posts

Anonymized

Abstract

Inferring author attributes from text (formal and informal) has been attempted numerous times in the literature. In this project we apply machine learning techniques to gender-inference using Facebook post data. In tackling this project we recognize that none of the numerous online social networking sites provide mechanisms to participants for detecting gender identity fraud. We believe that a robust gender-inference model could be incorporated into a general online author-attribute-inference service for providing protections to online community participants. In this report we discuss our current progress in predicting author gender from Facebook posts—including an overview of our data source and data extraction process, as well as data sets and best models we have developed thus far.

Of all strategies examined, backpropagation produces models with the highest average generalization accuracy, achieving 83% accuracy (based on a model with thirty six features). We calculate features for each of the 10,000 individuals in our data sets based on writing samples collected by aggregating Facebook posts.

1 Introduction

In the literature, linguists present numerous models for predicting author attributes based on text samples, including gender [Moshe *et al.*, 2003]—some more successful than others. In work by Argamon, Koppel, Fine, and Shimoni [Moshe *et al.*, 2003], the authors focus on formal written texts, noting that formal texts can be more difficult to evaluate than spoken language—for instance, novels are often edited for style and grammar by publishers. Similarly, and in contrast to formal writing, we believe that colloquial, spoken English includes numerous gender-specific properties from which a prediction model could infer gender with high accuracy.

A working gender-inference model would be particularly useful within online social networking communities. Consider for example the number of people currently communicating via Internet chat services (both synchronous and asynchronous—including Facebook) with persons that they have never met face-to-face. The ability to infer author traits such as gender (or age) from conversational text would be useful to the average participant to detect fraud. Particularly for adolescent users and their parents, services that detect an individual posing falsely as male or female would provide additional safeguards against online predation.

More generally, a successful gender-inference model could be incorporated into an online author-inference service. Internet communication applications could submit text to the service via an Application Programming Interface (API). In response, the service would evaluate the text sample against a learned model (or models) and return a gender prediction. Ideally the service would also provide some form of confidence assessment in addition to the prediction. The confidence assessment could take into account both model accuracy and the size of the text sample.

In this project we tackle one of the first steps necessary to provide an author-inference service to online communities—developing a reliable gender-inference model. We develop our model using machine learning techniques. In doing so, we acknowledge that other researchers have made attempts to infer gender from text. Although we will not conduct an extensive literature survey as part of this preliminary project, we note that future work on this topic should thoroughly explore existing solutions.

In Section 2 we discuss our data source, data extraction process, data sets (including example data instances) and selected machine learning techniques. In Section 3 we present results obtained from training and optimizing perceptron, backpropagation, KNN, and clustering models on the initial data set. We then discuss improvements, in Section 4, to the data and features, after which we discuss the final (best) results in Section 5. In Section 6 we present our conclusions, followed by a brief discussion of future work (Section 7).

2 Methods

In this section we discuss our methods for predicting author gender from colloquial English texts.

2.1 Data Source

We select Facebook as the data source for this project. Facebook affords several benefits: 1) posts and gender data are public by default (i.e., data availability); 2) Facebook is used

worldwide by millions of people (i.e., sample size and diversity); 3) Facebook provides an API for downloading posts (i.e., data accessibility); and 4) most users declare their gender, which is necessary for supervised learning. In addition to these benefits, we also note that the Facebook posting mechanism is used as an asynchronous conversation tool by many users, and we believe that most users do not heavily edit their postings. Thus Facebook posts represent colloquial language use and (arguably) approximate spoken language. Facebook is also a popular online social networking site and a key potential consumer of any author-attribute-inference service.

In regards to using Facebook data for supervised learning, we recognize the potential for Facebook users to falsify their gender declarations—a potential limitation for this project. However, we believe that the vast majority of users report their gender accurately, and therefore, we expect the machine learning techniques to overcome noise introduced by inaccurate reporting. In general, machine learning algorithms are designed to overcome noise by focusing on consistent relationships in data, and as long as sufficient data are provided, noise can be effectively filtered out. In this regard, Facebook provides millions of usable instances.

We locate individuals within Facebook by querying the Facebook API for non-user-specific posts, which produces a set of posts from seemingly "random" users. We then cache the user names/IDs associated with each post, pulling users in this way as necessary to obtain the desired number of unique users. Drawing names from the person cache, we finally query for individual posts and gender to obtain a raw data set. We use this particular person-acquisition process because 1) it is relatively easy to implement, 2) it provides only users with public posts (requiring less queries), and 3) Facebook does not provide a way to obtain a truly random sample of individuals short of downloading all individuals (which is infeasible considering API throttling constraints). We recognize the limitation of the sample not being truly random.

We concatenate enough posts for each individual to obtain a minimum number of "words" (substrings separated by white space) for each person, but without arbitrarily trimming the last post collected; thus text sample lengths vary slightly. Initially we queried for 1,000 words per person, from which we report initial results in Section 3. However, we found that boosting the sample size (see Section 4) significantly increases accuracy on all models; accordingly we report results in Section 5 based on 5,000-word samples. Also note that if gender or sufficient posts are not available for a given individual, the individual is dropped from the data set.

2.2 Data Sets

The data sets used in this study consist of 10,000 instances, 5,000 representing each gender. Generalization assessments during the model-building process were conducted using independently drawn random hold-out sets. The output variable in each data set is a single class variable representing gender ('M' or 'F'), which we map to a set of 27 continuous (i.e., real-valued) attributes in the initial data set and 35 continuous attributes in the final data set. The initial 27 attributes are listed and defined in Table 1; additional attributes are discussed in Section 4.

To calculate attributes, we perform lexical analysis on text samples to produce a list of tokens. A token represents either a word, a number, an emoticon, a punctuation mark, a symbol, a block of white space, or an "other" character. We then calculate the frequency of language features by dividing each feature's occurrence count by the total number of sampled words, tokens, or non-white space characters, respectively. We use frequencies because text samples are variable in length and frequency metrics are normalized. Some of the attributes in Table 1 measure the frequency of a specific token type. However, several attributes are designed to measure subfrequencies within a particular attribute type. For example, "word" tokens are subdivided into profanity, non-dictionary words, British National Corpus words, etc.

For each frequency attribute, we create an associated dictionary of strings—representing punctuation marks, standard emoticons, profane words, etc.—from which we count feature occurrences. In the case of the misspelling frequency, we use a standard English dictionary. The profanity dictionary was created from publicly available content filter dictionaries. Capitalization is ignored when appropriate (e.g., when comparing words for proper spelling). Also note in Table 1 that some attributes—designated by a superscript asterisk (*)—were derived in part from work by Rayson, Leech, and Hodges on the conversational component of the British National Corpus [Rayson *et al.*, 1997]. Similarly, attributes designated by a superscript plus (+) were derived from the Gender Genie application [BookBlog, 2007]. The Gender Genie is an online tool based on work by Argamon, Koppel, Fine, and Shimoni [Moshe *et al.*, 2003], in which the authors statistically model writing style characteristics in formal writing in order to differentiate between genders.

2.3 Selected Models

Consistent with user expectations of Internet applications, as well as the needs of potential client applications (e.g., Facebook), a gender-inference model should evaluate instances relatively quickly. A slow response time, for instance, would be constraining for chat applications, particularly for synchronous applications. Thus we are most interested in eager learning techniques. Further, continuous numbers are well-suited to measuring frequency-based attributes of text data, which is the predominant feature type in our attribute sets (see Section 2.2); thus we focus on models that naturally handle real-valued attributes.

In light of these two conditions we select the perceptron, backpropagation, KNN, and clustering techniques to model author gender in Facebook posts.[1] In applying these models to the problem, we tune algorithm attributes (where possible) to obtain the most generalizable models. For example, in the case of backpropagation, we optimize the algorithm's learning rate, graph size (number of hidden nodes and layers), and momentum rate to achieve the highest possible generalization accuracy.

[1]Note that perceptron, backpropagation, and clustering are all eager learning strategies, whereas KNN is a lazy learning approach. Perceptron, backpropagation, and KNN are also supervised learning algorithms; clustering is unsupervised.

Attribute	Description	Gender=F	Gender=M
Average word length	Average number of characters per word.	3.78169	4.20816
Longest word length	Maximum number of characters in any standard English word.	18.00000	18.00000
Misspelling frequency	Number of words per word token not appearing in a standard English dictionary (excluding contractions).	0.24547	0.16531
Profanity frequency*	Number of profane words per word token.	0.00704	0.00612
Capital word frequency	Number of all-capitalized alphabetic words per word token.	0.05433	0.03878
Gender Genie male-typical word frequency+	Number of male-typical words per word token (based on a dictionary of Gender Genie male-typical words [BookBlog, 2007]).	0.02314	0.02449
Gender Genie female-typical word frequency+	Number of female-typical words per word token (based on a dictionary of Gender Genie female-typical words [BookBlog, 2007]).	0.00503	0.00510
British corpus male-typical word frequency*	Number of male-typical words per word token (based on a dictionary of British male-typical words [Rayson *et al.*, 1997]).	0.00000	0.00000
British corpus female-typical word frequency*	Number of female-typical words per word token (based on a dictionary of British female-typical words [Rayson *et al.*, 1997]).	0.00805	0.00612
Self-reference frequency*	Number of personal pronouns per word token.	0.11670	0.05918
Third-person male reference frequency	Number of third-person male pronouns per word token.	0.00704	0.00306
Third-person female reference frequency*	Number of third-person female pronouns per word token.	0.00302	0.00204
Third-person gender-neutral reference frequency	Number of third-person neuter pronouns per word token.	0.01811	0.01224
Internet acronym frequency	Number of Internet acronyms per word token (e.g., "lol").	0.01408	0.00000
Line break frequency	Number of line breaks per character.	0.00184	0.00055
Exclamation point frequency	Number of exclamation points per character.	0.00285	0.00658
Question mark frequency	Number of question marks per character.	0.00041	0.00146
Quotation mark frequency	Number of quotation marks (single & double) per character.	0.00041	0.00421
Em-dash frequency	Number of em-dashes per character.	0.00020	0.00146
Digit frequency	Number of digits per character ("1,250" = 4 digits).	0.22961	0.23811
Capital letter frequency	Number of capital alphabetic letters per character.	0.03140	0.02725
Word token frequency	Number of word tokens per token.	0.47221	0.44708
Number token frequency	Number of number tokens per token.	0.00570	0.00639
Punctuation token frequency	Number of punctuation tokens per token.	0.03420	0.08531
Symbol token frequency	Number of symbol (non-alphanumeric) tokens per token.	0.00095	0.00137
White space token frequency	Number of white space tokens per token.	0.48171	0.45849
Emoticon token frequency	Number of emoticon tokens per token.	0.00523	0.00137

Table 1: Initial attributes (with descriptions and example instances) for training models to predict author gender from Facebook posts. Example instances are taken from the actual data set. One example is shown for each gender. All attributes are continuous.

*Based on work by Rayson, Leech, and Hodges [Rayson *et al.*, 1997].
+Based on work by Argamon, Koppel, Fine, and Shimoni [Moshe *et al.*, 2003].

3 Initial Results

For the initial round of model building, we tested the learning performance of standard perceptron, backpropagation, KNN, and clustering algorithms on the initial data set with 27 features. All models performed better than random (i.e., better than 0.5 accuracy), with backpropagation outperforming the other three. K-means clustering showed the least promise with an average generalization accuracy (over 3 runs) of 0.62.[2] For the clustering approach, we tested values for K up to 100. Cluster gender was assigned based on the most common gender within each cluster. For testing, new instances were assigned to a cluster (i.e., gender) based on the closest distance to a cluster centroid. The linear perceptron model performed remarkably better than clustering, with an average generalization accuracy (over 3 runs) of 0.69. Distance-weighted (Euclidean) KNN, with $K = 10$, achieved an average generalization accuracy (over 3 runs) of 0.70, slightly better than perceptron, and backpropagation outperformed all three. We discuss the performance of backpropagation in detail.

Initially backpropagation produced an average generalization accuracy (over 3 runs) of 0.71. In this case the multi-layer perceptron included three hidden layers, with 54 nodes (twice the input size) per layer. We also used a learning rate of 0.1 and and a momentum rate of 0.9. With these baseline results, we then optimized the backpropagation algorithm's parameters for best generalization accuracy by independently testing four different, independently developed versions of the algorithm. Each author worked with various settings and stopping conditions to achieve the best possible generalization accuracy.

For the best-performing backpropagation algorithm (of the four implementations), changes in the number of hidden nodes and layers, learning rate, and momentum rate had only small impact on average generalization accuracy. Setting a learning rate extremely high, or the number of hidden nodes extremely low, did significantly (negatively) impact accuracy, but within reasonable limits optimizing these parameters only provided marginal accuracy gains. Overall, these adjustments led to accuracy fluctuations within a range of one to two percent. The other backpropagation algorithms, which did not perform as well as the baseline—peaking between 67% and 69% generalization accuracy—fluctuated more significantly, over a range of about 10% each, in response to parameter changes.

We tested the optimal backpropagation algorithm on a range of hidden nodes from 2 to 64. Increasing the number of hidden nodes improved average generalization accuracy (over 15 runs) from 0.708 to 0.719.[3] We tested learning rates for this algorithm between 0.01 and 10.00, for which accuracy ranged between 0.690 and 0.719. Momentum rates ranging between 0.00 and 0.95 led to accuracies between 0.711 and 0.719. The best generalization accuracy performance of 0.719 resulted from 10 hidden nodes (1 hidden layer), a learning rate of 0.1, and a momentum rate of 0.0. We also tested the algorithm with two hidden layers, but for all settings, two hidden layers performed more poorly than with a single layer.

We further tested backpropagation with varying sizes for the training set—from 5,000 to 9,500 randomly selected instances—using the remaining data set instances for testing. We obtained the accuracy of 0.719 during the optimization process by training on 5,000 instances. As expected, increasing the training set size led to increases in the average generalization accuracy, with some fluctuation. We achieved the highest average generalization accuracy (over 15 runs with optimal parameter settings) of 0.726 when training on 9,500 instances and testing on the remaining 500.

4 Data and Feature Improvements

With an optimal average generalization accuracy of about 72%, we were unsatisfied with our best gender-inference model. In looking to improve performance, we considered ensemble approaches, increasing the training set size, topic analysis, subdividing feature dictionaries, and boosting the sampling size for Facebook posts.

To help determine which of these approaches might be most effective, we configured the backpropagation algorithm to maximally overfit the training data. Interestingly, running backpropagation on a multi-level perceptron graph with 3 hidden layers, 54 nodes per layer, and with all overfit avoidance mechanisms disabled, we could never achieve an accuracy above approximately 73%. With a gain of only approximately 1%, these results indicate that the data is naturally noisy, and thus an ensemble approach may not be the most effective next step. Of course, various models may deal with noisy data differently, and we would accordingly anticipate an accuracy gain from this strategy. However, if noise is a significant inhibitor, then we should focus first on improving the data to reduce noise. Ensembles and other more advanced modelling techniques are best addressed after reducing data noise as much as possible (see Section 7).

To reduce (or control) noise in the data, the most obvious approach is to increase the number of training instances. We tested the potential impact of increasing the training set size by running the best backpropagation model on training sets varying from 5,000 to 9,500 instances. As discussed in Section 3, an increase of 4,500 instances improved average generalization accuracy by less than three quarters of one percent. The initial data set also required nearly ten hours, with appropriate throttling, to download from Facebook. Consequently, we decided instead to try boosting the Facebook post samples from 1,000 to 5,000 words per person.[4] Theoretically, boosting the data in this way should provide more accurate estimates of language usage, which would reduce noise among the feature estimates.

We also decided to make specific feature changes. In this regard, we considered topic analysis as one possible technique for discovering latent features in Facebook posts that correlate highly with gender; we decided against this strategy, however,

[2] All generalization accuracy tests were run on independently drawn random hold-out sets, each representing 10% of the data set.

[3] For these tests only 50% of the data set (5,000 instances) was used for training (i.e., 50% random hold-out sets were used for testing). We reduced the training set size during the parameter optimization process because the process involved running over 300 models.

[4] Downloading 10,000 instances at 5,000 words per person required approximately 30 hours with throttling.

due to time constraints (see Section 7 for a description of this strategy). Instead, we decided to increase the granularity of our attributes by subdividing some of the feature dictionaries. Recognizing that more features generally require more data, we anticipated that 10,000 instances would be sufficient to support at least 35 features. We also hoped that more granular features would allow the backpropagation model to learn which dictionary subsets are most important.

Accordingly, we subdivided four of the initial attributes—Gender Genie male-typical word frequency (originally 17 words), Gender Genie female-typical word frequency (originally 16 words), British corpus male-typical word frequency (originally 26 words), and British corpus female-typical word frequency (originally 25 words). These attributes represent the most typical words of male/female speech from two prior studies ([Moshe *et al.*, 2003] and [Rayson *et al.*, 1997], respectively). We subdivided each of these attributes (roughly equally) into three sub-dictionaries, representing the highest, lowest and middle-frequency[5] words for each gender—thus increasing the total number of attributes in our data set from 27 to 35.

5 Final Results

We tested our models from the initial round of model building on the boosted data with both the original 27 attributes and with the modified 35 attributes. Boosting the data (increasing the sample size from 1,000 to 5,000 words per person), significantly increased generalization accuracy on all models. For the original 27 attributes, K-means clustering model accuracy increased on average by 5% from 0.62 to 0.67; perceptron model accuracy increased on average by 8% from 0.69 to 0.77; KNN model accuracy increased by 8% from 0.70 to 0.78; and the baseline backpropagation model increased on average by 10% from 0.71 to 0.81.[6] With optimized parameters, backpropagation achieved an average generalization accuracy (over 15 runs) of 0.823 on the boosted data.

Adding the feature modifications (i.e., subdividing attribute dictionaries) also increased generalization accuracy for three of the four modelling strategies. K-means clustering did not produce significantly different generalization accuracy with the modified features, but perceptron, backpropagation, and KNN all achieved an accuracy gain of approximately 2%.

The best model overall was produced by the optimized backpropagation algorithm run on the boosted data with the modified features; this model achieved an average generalization accuracy (over 15 runs) of almost 83% (0.829). The optimized algorithm included 1 hidden layer, 10 hidden nodes, a learning rate of 0.1, and a momentum rate of 0.0, as mentioned previously. The stopping condition also reserved 25% (2,250 instances) of the training data for a validation set, and the algorithm stopped after 300 epochs without accuracy improvement on the validation set. We tested the algorithm at higher thresholds for the required number of epochs to search without validation set accuracy improvement, but tests up to 3,000 epochs did not find further accuracy improvement. Throughout this process, the model weights which performed best on the validation set were retained for final testing to assess generalization accuracy. Generalization accuracy is reported as the average over 15 runs, tested on independently drawn random hold-out sets. Test sets represented 10% of the data set (i.e., 1,000 instances), randomly drawn from the data set prior to partitioning the random validation sets.

[5] Frequencies in this case were based on the word-frequencies observed in the original research studies, from which these dictionaries were taken.

[6] Based on averages over 3 runs; accuracy tested against independently drawn random hold-out sets, each representing 10% of the data set.

6 Conclusions

Of all the improvements tested—including algorithm optimization and feature changes—data boosting (increasing the number of words sampled per person) provided the most significant accuracy gains. For backpropagation, which was the most effective learning strategy tested, data boosting improved model accuracy by more than 10%. Optimizing algorithm parameters, increasing the training set size, and subdividing feature dictionaries each impacted accuracy by less than 2%. Collectively, we were able to improve accuracy on the backpropagation model by roughly 12% (from 0.71 to 0.83)—a significant gain—and relative to the worst performing model, we improved accuracy by roughly 21%.

7 Future Work

Despite significant accuracy improvement over niave models, we believe that our gender-inference techniques can be further improved to well over 90% accuracy. Although further boosting of the data would likely provide additional gains, 5,000 words is already a rather large sample size for the Facebook context. In order for a gender-inference tool to be practical, the required sample size must be somewhat limited. Of course, we have not yet experimented with training models on boosted data and then testing them on data with smaller sample sizes, which may prove reasonably accurate. Nevertheless, we believe that the most promising next step in developing gender-inference models would be to further refine the data features.

7.1 Feature Refinements

Feature exploration was limited in this preliminary study. For example, we only had time to test 35 features—most of which we selected based on theoretical arguments—and although we derived some of the 35 features from two prior studies on gender-based language use, the literature presents many other options for reasonable features that could be tested. Methods in natural language processing (NLP) also provide several unsupervised techniques for identifying latent features in text. In particular, applying latent Dirichlet allocation (LDA) to topic analysis seems promising. NLP tools already exist to perform LDA topic analysis, and this type of analysis, if correlated with gender, may reveal new, more effective feature dictionaries (based on words that fall within prominent topics).

7.2 Other Models

Beyond topic analysis, NLP research is a rich source for machine learning models, and we believe that NLP techniques

could produce better, more specialized models than backpropagation because of their focused emphasis on the analysis of text data. Bayesian techniques, which are commonly used in NLP, are also particularly suited to frequency-based feature data.

References

[BookBlog, 2007] BookBlog. The gender genie. `http://bookblog.net/gender/genie.php`, 2007. Last accessed, March 2011.

[Moshe *et al.*, 2003] Shlomo Argamon Moshe, Jonathan Fine, and Anat Rachel Shimoni. Gender, genre, and writing style in formal written texts. *Text-Interdisciplinary Journal for the Study of Discourse*, 23(3):321–346, 2003.

[Rayson *et al.*, 1997] Paul Rayson, Geoffrey Leech, and Mary Hodges. Social differentiation in the use of English vocabulary: Some analyses of the conversational component of the British National Corpus. *International Journal of Corpus Linguistics*, 2(1):133–152, 1997.