A Style-Based Approach to Gender Identification on Blogs

Daniel Cauchi

Abstract

While having bias is necessary to generalising beyond training data, it may sometimes lead to overfitting. Within the context of gender identification in text, more specifically in the blogging domain, one of these biases is usually the topic of the individual documents. This work attempts to study the problem of gender classification in text by using solely stylistic features.

1 Introduction

In this study, the task of Gender Identification on a Blog Corpus is implemented trough the use of Natural Language Processing techniques. The goal is to build a classifier which is able to recognize the differences in writing style between male and female writers. Usually, in other natural language tasks such as text generation, the issue of gender bias is one which researchers try to avoid, in order to prevent the model from learning stereotypes. In this case, however, the opposite is required, where the features will be chosen to amplify this bias. The task is to find the ideal features which will allow the generated model to discriminate best between male and female writers. For this approach, any feature which is related to topic is avoided, purely focusing on the style of writing. This is particularly difficult because the aim is to try and find similarities across an entire group of authors, because while individuals may have their own style of writing, it is more difficult to generalise the style over a set of people [1].

The solution to the gender identification problem has applications in areas such as forensics [2] and cybercriminal analysis [3], so the identity of a suspect may be narrowed down to one gender. Another application is in marketing, to find the gender demographic of some outlet.

The following chapters will discuss previous work done in this area, a description of the corpus in

question, the approach taken to solve this problem and an evaluation on the model built.

2 Previous Work

The area of text categorization is highly studied and hence many works have been presented, however an optimal feature set has not yet been published [3]. This may be due to different domains having different features or due to the complexity of the problem. [3] uses RNN models for author identification. A particular emphasis is made on GloVe vectors. This method extracts both context, since it uses the actual words, as well as style, due to the RNN storing the sequence and hence the particular order in which the words are used. For this work however, context will not be taken into account, hence there is no use of the raw words themselves.

In [1], the difference between the style of writing of the two different genders in the domain of fiction and non-fiction works has been studied. It was noted that the main distinguishing features are different between the two domains. On the other hand, it was discovered that certain function words as well as parts-of-speech(POS) tags can be used to discriminate between the two genders in both domains. In this work, both of these feature sets are used applied to the blogging domain.

[4] applies the task of gender identification within the e-mail domain. Here, the language background of the author is also considered and classified. It is stated that females tend to use more expressive language. Some of the words used mostly be females include those such as "terribly" and "dreadful". In this work, these are accounted for by taking into account words ending with certain suffixes. A general distinction between the styles of men and females within the e-mail domain is that male conversation tends to resemble 'report-talk' while females prefer 'rapport-talk'.

[2] use the same corpus as this work and define the state of the art of this task. They claim that women write in a style which is more involved, meanwhile, men prefer writing which gives more information, which is similar to the 'report' and 'rapport' talk discussed by [4]. Important features such as the number of URL links and use of pronouns are discussed, and some of these will be used within this work. [2] also uses the topics and individual words of the posts. This is something which this work will avoid, in order to study how well the gender can be predicted using purely style-based features without the bias of topic. The observation that style changes as the age changes is also made. This means that as age differs, the style between the gender also varies and mix into each other, making it more difficult to predict gender based purely off of stylistic features.

3 Data and Methodology

3.1 The Corpus

The corpus used is the Blog Authorship Corpus by [2]. This is a corpus containing posts from 19320 blogs, with 9660 of them being male and another 9660 of them being female. In total there are 344773 male posts and 335010 female posts. Other meta-data includes the age range of the bloggers and their star sign. These will not be considered since the text is the only focus of the task. Other observations on the data include that there are some empty posts, which will be pruned later on. Furthermore, some blogs are spam, with posts containing advertisements or non-sense sentences. These will act as noise in the data. The individual posts were trimmed from trailing spaces and new lines were replaced by full stops prior to any other preprocessing.

3.2 Feature Extraction and Filtering

The following 'general' features were extracted from the corpus for each post: Word Count, Sentence Count, Average Word Length, Average Sentence Length, Number of Unique Words/Word Count, Number of URLs/Word Count. The number of words ending with the following suffixes were also counted and divided by the Word Count: able, al, ful, ible, ic, ive, less, ly, ous. This list was obtained from [4] The reason they were divided by the Word Count is because the length of a post may affect the frequency of words in general, hence the features are normalized by this value. It is important to note that the words considered in this step do not include any punctuation.

A few vocabularies were created to extract some important words and bi-grams with regards to the posts. Prior to this however, the posts were split into a train and test set. This was done by first sorting the dataset by file name in ascending order, and getting the number of posts by each author. Then the train set was made to contain around 80% of each gender. This was made so that an author who is in the train set cannot also be in the test set. Precisely, the training set contains the first 7728 male blogs (270106 posts) and the first 7148 female blogs (269911 posts), so that the number of posts in the train set to balance out. The other 20% of posts were left for testing. Validation sets will be extracted from the test set later on during model building.

The posts were then converted twice, once as POS tags using the NLTK averaged perceptron tagger [5] and once as bleached text [6]. In this work's case, the bleaching technique used was to replace consonants with the letter C and vowels with the letter V. Bleaching allows for abstracting

the word itself to avoid content (and language) bias, but rather focusing on the form of the word. The hypothesis here is that men and women use different word forms when writing.

After the conversion process, a vocabulary of bi-grams and their sum of normalized counts is generated on the train set. Bi-grams were chosen over uni-grams to account for ordering and they were chosen over tri-grams because posts tend to be short, hence these might have ended up being too sparse. The sum of normalized counts, which will be abbreviated to SNC from now on, is the sum of the counts of an item within a document, divided by the total items of that type in that document, summed over all the documents. This is used so that the length of the posts does not skew the counts, otherwise lengthier posts would have more weight. So for an item x, the SNC of x (The bi-grams and later on function words, in the case of this study) over all the documents D (the posts, in the case of this study) is:

$$SNC(x) = \sum_{d \in D} \frac{number\ of\ times\ x\ occurs\ in\ d}{number\ of\ items\ with\ the\ same\ type\ of\ x\ in\ d}$$

2 values for SNC were generated on the train set for each POS bi-gram and bleached bi-gram. The first value is the SNC over the male written documents and the second on the documents written by females. This resulted in 1,692 POS bi-grams and 2,010,167 bleached bi-grams. These were then filtered by removing those with a total SNC (SNC of a bi-gram over the male written documents + SNC of a bi-gram over the female written documents) of 100 so that the probability of the same bi-grams appearing in the test set as well is high, therefore avoiding sparseness in the data. These were then sorted by their entropy and the top 50 with the least entropy (highest information gain) from each group was taken, so that in the end 50 POS bi-grams and 50 bleached bi-grams were obtained as features. This was done to not have too many unnecessary features. The idea of using entropy is similar to that of using TF-IDF in order to analyse how well a particular item distinguishes between two documents, or sets of documents in this case. However, due to the use of SNC, this method also accounts for post length.

The final set of features were function words. The entire set consisted of 277 function words which can be found in the Appendix. These were pruned much the same way as the bi-grams. Those with a total SNC of 50 or more were taken, then the top 50 of those with least entropy were chosen. It is important to note that when using SNCs, the words used were generated by 'word_tokenize' from nltk, so punctuation was considered, unlike in the case of the 'general' features.

Actual	**Male**	36731 (57%)	28067 (43%)
	Female	27108 (36%)	47192 (64%)
		Male	**Female**
	Predicted		

Table 1: Logistic Regression Model Confusion Matrix

The final set of selected function words may be found in the Appendix. After the features were decided, feature extraction was done for each document. When it came to the bi-grams and function word counts, these were divided by the total number of bi-grams or words in that document as well, meaning that the SNC was taken, but D was of size 1.

3.3 Model Creation

2 models were created using scikit-learn [7]. The first is a Logistic Regression model and was used as a baseline. The second is a multi-layer-perceptron with 1 hidden layer of 265 nodes (the total number of features + 100). Prior to training, the hyperparameters of the models were tuned using scikit-learn's random search, which is similar to grid search but does not search trough all combinations. This evaluates the models created using a cross validation technique. Note: this search was made on 10,000 shuffled examples from the train set. After the best hyperparameters are found, the classifier is fit on the entire train set and tested on the test set. The parameters of the final model may be found in the Appendix.

4 Results and Evaluation

The confusion matrix of the baseline model can be seen in table 4 while that of the main model can be seen in table 4. The accuracy of the baseline ended up being 60% while that of the main model ended up being the same at 60%. These results are very similar and based on the fact that the overall accuracy is above 50% on a significant amount of data shows that some form of pattern was found. However, this score is still very inconclusive, especially when compared to the state-of-the-art presented in [2] which achieved an overall accuracy of 80.1%. Scaling was also later done on the data using the Standard Scaler provided by scikit-learn using the settings 'with mean' and 'with std', but the results were very similar. Next, the reasons as to why these results may have occurred are discussed.

Actual	**Male**	37067 (57%)	27731 (43%)
	Female	28352 (38%)	45948 (62%)
		Male	**Female**
	Predicted		

Table 2: Multi Layer Perceptron Model Confusion Matrix

Firstly, the reason why the state-of-the-art is so low is considered. This may be due to the fact that style varies by age, as pointed out by [2]. It is stated that as people grow older, the style starts resembling that of the male writing more and more. Therefore, as the data contains writers of mixed ages, it becomes increasingly difficult to distinguish between male and female writers. Another obstacle is the fact that each age group does not have the same amount of writers from both genders. In fact, there are more female teenage bloggers, while the older age range consists of mostly men. The final hurdle is the noise in the data, that is, the blogs with spam for content, as well as those bloggers whose authors have provided false information with regards to their identification [2].

Next, the reason why this implementation yielded worse results is discussed. As pointed out by [2], using both style and content-based features yielded the best results. Content-based features however were deliberately avoided, to account for men and women writing about the same things. Furthermore, the number of bi-gram and function word features was trimmed down, so using more of these might increase accuracy. This, however, is likely to provide nothing more than a slight improvement, due to the fact that both frequency and information gain were taken into account, so the remaining bi-grams and function words should have little effect and also lead to more sparseness in the data.

5 Future Work

While this study has revealed some potentially useful features, more work may be done to build upon it. One way is to repeat the process but prune less features. Another is to approach the feature selection for the POS and bleached features differently, such as using more than one n-gram type and uncovering distinguishing features in other n-gram types.

Other potentially useful features are the use of misspelt words, the use of excessive punctuation or the use of emojis. Other forms of bleaching presented in [6] may also be tested, with different n-gram types. Other studies may also opt to perform the work using only a subset of the data, such as considering only one age range.

6 Conclusion

This study has presented a way of distinguishing between gender writing styles in the blogging domain. The method presented does not consider topic, but rather only makes use of style-based features. As such, it does not improve upon the state-of-the-art, but presents a new way of trying to tackle this problem. Almost all the features used, except for a few such as 'Number of URL links' could have been used in other text domains, besides blogging. Although the decisions made were done deliberately to try and reduce the amount of bias of the domain as much as possible, the classification score suffers as a result.

While much has been studied already, it is clear that more work needs to be done in this area to determine which features may distinguish well between the style of writing between the genders. This is especially the case for the less biased approach of classification, where style is the only focus, without the effect of topic.

References

[1] M. Koppel, S. Argamon, and A. R. Shimoni, "Automatically categorizing written texts by author gender," *LLC*, vol. 17, pp. 401–412, 2002.

[2] J. Schler, M. Koppel, S. Argamon, and J. Pennebaker, "Effects of age and gender on blogging.," pp. 199–205, 01 2006.

[3] L. Zhou and H. Wang, "News authorship identification with deep learning," 2016.

[4] O. Vel, M. Corney, A. Anderson, and G. Mohay, "Language and gender author cohort analysis of e-mail for computer forensics," *In Proceedings of the Digital Forensic Research Workshop. Syracuse, NY*, 01 2002.

[5] E. Loper and S. Bird, "Nltk: The natural language toolkit," in *In Proceedings of the ACL Workshop on Effective Tools and Methodologies for Teaching Natural Language Processing and Computational Linguistics. Philadelphia: Association for Computational Linguistics*, 2002.

[6] R. van der Goot, N. Ljubešić, I. Matroos, M. Nissim, and B. Plank, "Bleaching text: Abstract features for cross-lingual gender prediction," in *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, (Melbourne, Australia), pp. 383–389, Association for Computational Linguistics, July 2018.

[7] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

Appendix

Full list of considered Function Words

i, we, you, he, she, it, they, me, us, him, her, them, myself, ourselves, yourself, yourselves, herself, himself, itself, themselves, someone, anyone, noone, everyone, nobody, something, anything, nothing, everything, whoever, whatever, others, mine, ours, yours, hers, theirs, my, our, your, his, its, their, one, first, second, third, once, this, these, that, those, a, an, the, all, alone, another, any, both, each, either, enough, every, few, former, latter, last, least, less, lot, lots, many, more, most, much, neither, next, none, only, other, several, same, some, such, top, whole, and, but, or, nor, although, as, because, if, while, however, whenever, wherever, whether, whyever, thereby, therein, thereupon, thereafter, whereafter, whereas, whereby, wherein, whereupon, again, also, besides, moreover, namely, furthermore, hence, so, therefore, thus, else, instead, otherwise, after, afterwards, before, meanwhile, now, then, until, anyhow, anyway, despite, even, nevertheless, though, yet, eg, ie, per, re, etc, about, above, across, against, along, among, amongst, amoungst, around, at, behind, below, beside, between, beyond, by, down, during, except, for, from, in, inside, into, near, of, off, on, onto, outside, over, since, than, thence, to, toward, towards, under, up, upon, through, thru, throughout, via, with, within, without, am, are, is, was, were, be, been, being, became, have, has, had, do, does, did, done, will, shall, may, can, cannot, would, could, should, might, ought, need, must, used, dare, yes, no, not, already, always, anywhere, beforehand, elsewhere, ever, everywhere, formerly, further, here, hereafter, hereabouts, hereinafter, heretofore, herewith, hereunder, hereby, herein, hereupon, indeed, latterly, mostly, never, nowhere, often, oftentimes, out, perhaps, somehow, sometime, sometimes, somewhat, somewhere, still, there, thereabouts, thereof, thereon, together, well, almost, rather, too, very, who, whom, whose, what, which, when, where, why, how, whither, whence

List of chosen Function Words chosen alongside their SNC

Function Word	SNC Male	SNC Female	Entropy
him	214.2922	356.0366	0.955
against	51.0377	30.7305	0.955
i	1567.799	2414.59	0.9671
she	307.7963	449.0412	0.9747
alone	32.5691	46.7369	0.9769
her	356.612	495.0073	0.9809
myself	115.229	159.5427	0.9812
sometimes	37.2033	50.7401	0.9828
me	1215.922	1658.114	0.9829
am	433.7418	585.2614	0.984
etc	37.0564	27.7385	0.985
so	938.9657	1240.295	0.9862
my	1906.424	2444.647	0.9889
has	487.4663	382.151	0.9894
which	273.685	216.637	0.9902
he	535.2909	676.1345	0.9902
yes	56.1893	70.6743	0.9906
always	124.6748	156.6017	0.9907
everything	85.4209	106.8266	0.991
their	262.6877	210.9015	0.9914
may	122.9839	98.8918	0.9915
someone	111.4211	138.4302	0.9916
never	166.3325	206.3223	0.9917
rather	53.5806	43.4593	0.9921
between	55.1349	45.0286	0.9926
together	54.0341	66.1386	0.9927
also	192.9389	158.05	0.9929
too	267.9576	325.9133	0.9931
near	27.9716	23.0536	0.9933
inside	32.3806	39.2204	0.9934
few	149.1153	123.3524	0.9935
as	777.8161	643.778	0.9936
when	404.2057	486.1151	0.9939
during	54.5989	45.517	0.9941
often	36.3503	30.3731	0.9942
an	490.6424	411.0569	0.9944
its	254.2393	213.3131	0.9945
did	354.1284	420.9786	0.9946
because	267.6371	317.5187	0.9948
some	552.0361	466.6829	0.9949
by	590.3954	500.1218	0.9951
these	160.9013	136.5932	0.9952
second	51.3389	43.5993	0.9952
do	956.8416	1125.672	0.9953
shall	31.19	26.5235	0.9953
of	3729.425	3176.029	0.9954
most	158.4158	135.0636	0.9954
mine	39.7702	46.551	0.9955
under	48.3208	41.4277	0.9957
even	213.6388	249.0733	0.9958

List of chosen POS bi-grams chosen alongside their SNC

Word1	Word2	SNC Malr	SNC Female	Entropy
VB	VBP	65.6627	112.052	0.9503
.	.	1394.375	2198.363	0.9636
NNP	NNPS	74.0089	48.0923	0.9673
JJ	VBP	342.4056	514.7306	0.9706
NNP	“	211.6405	142.709	0.9725
NNP	TO	234.835	158.397	0.9726
DT	NNP	1512.683	1030.576	0.9739
NNP	MD	143.5035	97.807	0.974
NNP	)	196.837	135.6201	0.9754
.	)	112.0248	159.5941	0.9778
CD	NNP	250.9739	177.784	0.9789
.	NN	1222.887	1720.136	0.9793
NNP	NNS	316.1016	226.1596	0.9801
)	PRP	54.9069	76.5361	0.9804
NN	VBP	531.0249	739.9092	0.9804
PRP	CC	182.1071	251.6921	0.9814
(	NNP	183.5413	133.8577	0.9823
“	DT	120.2373	87.8999	0.9825
NNP	(	254.8624	187.066	0.983
PRP	WRB	44.8118	60.821	0.9834
NNP	VBZ	786.689	580.3003	0.9835
POS	NNP	167.0235	123.3307	0.9836
(	CD	82.5568	61.0877	0.9838
“	NNP	323.2489	239.3142	0.9839
IN	NNP	2204.289	1647.922	0.9849
NNS	VBZ	64.5387	48.3555	0.9851
NNP	CD	409.7641	307.7386	0.9854
.	VBN	64.2881	85.3735	0.9856
JJ	PRP	240.6108	317.6176	0.9862
VBG	NNP	152.1905	115.2999	0.9862
CD	)	114.2074	86.8631	0.9866
CD	,	182.5331	139.5307	0.9871
VBP	VBP	94.7247	123.5693	0.9874
NNP	DT	223.3337	171.3437	0.9874
VBN	VBN	109.9374	84.5027	0.9876
.	(	108.0842	140.3996	0.9878
NNP	IN	1210.078	932.5676	0.9879
NNP	:	883.097	680.8323	0.9879
CD	:	144.7537	111.6544	0.9879
:	NNP	575.1356	445.26	0.9883
VBP	PRP	972.5374	1246.799	0.989
VBP	WP	63.8502	81.4921	0.9893
PRP	PRP	94.7288	120.6831	0.9895
PRP	NN	100.8661	128.4812	0.9895
VBP	WRB	48.0088	61.1278	0.9896
PRP	.	863.0469	1098.13	0.9896
WP	NN	53.5319	68.0592	0.9897
DT	“	102.6618	80.7661	0.9897
PRP	VB	429.3954	545.7472	0.9897
VBN	DT	270.9949	214.0776	0.99

List of chosen Bleached bi-grams chosen alongside their SNC

Word1	Word2	SNC Male	SNC Female	Entropy
CVV	?	56.5679	111.4084	0.9217
VCCCVCC	CVCCVC	139.2356	71.8518	0.9252
–	–	213.5822	110.9263	0.9266
VCCCVCC	CVCCVCC	83.4459	43.5447	0.9276
!	V	155.412	294.238	0.9301
VCCCVCC	CCCC	116.2228	62.4204	0.9335
VCV	CVV	79.5646	143.5611	0.9398
CVV	!	52.4206	90.4633	0.9482
CCCC	:	212.5414	124.1742	0.9497
!	CV	64.0405	109.3699	0.9501
!	!	1056.766	1749.012	0.9556
!	CVC	118.8016	195.9028	0.9563
.	:	48.9806	80.0803	0.9577
CVCCC	!	71.0977	115.77	0.9584
!	)	44.8555	72.599	0.9594
!	VCC	53.1816	85.6058	0.9603
?	?	153.8778	246.3557	0.9611
!	VC	85.8364	136.6319	0.9621
CV	!	82.0581	130.2348	0.9625
!	CVCV	47.5408	74.5157	0.9645
CVCV	VCCCVCC	104.4577	66.825	0.9649
CC	VCCCVCC	81.1055	125.1035	0.9669
CVC	!	217.6588	332.8318	0.9682
CCV	VCCCVCC	94.8332	62.226	0.9687
VC	VCCCVCC	150.8388	99.4074	0.9693
?	!	56.4455	84.4103	0.9714
VC	!	79.854	118.7488	0.9722
CVCC	!	161.2059	237.9964	0.9731
CCVC	VCCCVCC	79.9944	54.3574	0.9736
VCCCVCC	CVCVC	103.4261	70.7286	0.9744
!	VCCCVCC	149.2193	215.628	0.976
CC	CVVCC	64.7836	93.5462	0.9761
CCV	“	59.3288	41.4081	0.977
CCV	’	74.5903	105.5288	0.9786
CVCVC	!	63.9389	90.4529	0.9786
!	CCVC	64.6387	91.0119	0.9792
CVCV	!	120.317	167.0237	0.9809
CVCVVC	VC	60.6417	43.7691	0.9811
!	CVCC	69.0059	95.3272	0.9814
VCCCVCC	CVCC	236.5959	171.6028	0.9816
CVCVVCV	V	82.9509	113.8992	0.9821
:	V	66.1958	90.8847	0.9821
:	)	114.3231	156.4926	0.9824
:	“	78.2702	57.8101	0.9836
CVVC	!	97.7399	131.8733	0.984
V	VC	403.2419	543.9764	0.984
CCV	CVCCVCV	66.8201	49.6087	0.9842
VCCCVCC	CVVC	114.0275	84.9357	0.9845
CV	’	98.2984	131.1969	0.9851
CCV	CVCVCVC	62.8008	47.1078	0.9852

Parameters of the baseline logistic regression model

LogisticRegression(C=9.50814306409916, class_weight=None, dual=False, fit_intercept=True, intercept_scaling=1, l1_ratio=None, max_iter=100, multi_class='auto', n_jobs=-1, penalty='l2', random_state=42, solver='newton-cg', tol=0.0001, verbose=1, warm_start=False)

Parameters of the final multi-layer perceptron model

MLPClassifier(activation='relu', alpha=0.018192496720710064, batch_size=100000, beta_1=0.9, beta_2=0.999, early_stopping=True, epsilon=1e-08, hidden_layer_sizes=(265,), learning_rate='constant', learning_rate_init=0.0184404509 max_iter=300, momentum=0.8, n_iter_no_change=30, nesterovs_momentum=True, power_t=0.5, random_state=42, shuffle=True, solver='adam', tol=0.0001, validation_fraction=0.2, verbose=True, warm_start=False)