LIN3012 -Data-Driven Natural Language Processing

Final Project

Project 1: Blog gender identification

February 2020

Lara Brockdorff

110899M

BSc Computing Science – Year 3

Contents

Introduction

This is the report to the assignment LIN 3012 – Data-Driven NLP. From the available tasks, Project 1, titled Blog Gender Identification. This report describes the data and methodology used to attempt this task, while also outlining the contributions of previous work from reviewed literature. Data processing and modelling was done using python3. Code can be found in the repo at: https://github.com/LaraBrockdorff/LIN3012_Assignment or in the uploaded zipped folder on VLE.

The environment used consisted of libraries mentioned on the VLE that were required for the practical sessions.

Problem definition and Motivation

Text classification is a subfield of natural language processing that aims to classify text based on features found in the written piece. This task aims specifically identify the gender of the author of a blog. Blog are forms of journals or articles, published online, where often writers share their views on a subject matter. This task aims to create a model based on statistical properties that can identify the gender of the author given an unseen blog.

The area of author identification using statistical models or classifiers is one of open research as it Can be play an important role in the process of identifying a specific author which can be useful in plagiarism detection and finding ghost authors. Can also be used to identify authors that a reader might prefer, in order to provide more targeted content and recommendation for the reader. (Qian, 2017)

This area can also be of interest to those interested in researching differences present in the way authors write, and how the gender of an author can be identified through their writing style.

Past work / Literature review

The task of author identification from text is one that has been approached and investigated through many studies. A common fundamental competent to the task is identifying the relevant features that can be used to best predict the author of a piece of text, or in our case, the gender of the author of the blog.

(Saleh, 2014) discusses the difference between lexical, character based measures, syntactic measures, semantic , content specific and structural features in author identification. Since it is not feasible to attempt to identify all possible features, attention was focused on lexical and character-based measures, that include components like word frequencies and character sequences respectively.

Rather than word or character ngrams, (Peng, 2016) discusses the use of bit-level ngrams, and mentions how when working with informal language and grammar on social media complicates the use of high level language attributes, motivating their work on bit-level ngrams. This approach does not require interpretation of the given text. With this in mind, comparing unigrams to character ngrams would be an interesting feature to look at.

After examining potential features, the next step would be to identify the right classifier for the task. While looking at author gender identification from text of internet application (NaCheng, 2010) discusses different possible classifiers including Bayesian logistic regression. (Saleh, 2014) also mention the use of both multinomial and Bernoulli naïve Bayes classifiers, despite dealing with Arabic text, and not English.

(J. Schler, 2006),also mentioned how the paragraph lengths vary between genders in the Blogs data set. This raises the question of whether the total blog length can help identify the author of the author. Working with the same data, (Aayush Singhal) also mention how they made use of readability metrics and their effects.

With previous work in mind, the aim of the implemented work was to experiment with blog length, readability, word ngrams and character n grams as potential features to be use for a

Bernoulli Naïve Bayes classifier. The choice of Bernoulli, that takes features as binary representation of feature a feature was present was chosen in order to experiment with the idea of occurrence of a feature, rather than the frequencies, as is often done with a multinomial naïve Bayes classifier.

Data and methodology

Dataset

While using a supervised learning approach when approaching this task, a prelabelled dataset was required. This dataset was obtained from (J. Schler, 2006)

The original data was made up of .xml files, each containing the blogs of a certain author, with the author ID, gender, age and zodiac sign in the file name.

In order to compute and manipulate the data, it first needed to be read, parsed and labelled. The gender label was extracted from the file name stored till the blog content was extracted. This was done using the Beautiful soup Parser (BFS4). This allowed for the xml content to be read and saved in a list. The label and list of blogs were stored as tuples in a list, converted to a data frame and then saved as a CSV file. The file containing this data extraction folder can be found in the repo mentioned and zipped folder, named DataReaderScript.ipynb .

The mentioned csv file can be generated from the given python script but can also be found uploaded here.

https://drive.google.com/drive/folders/1NX_iemQoIgixyKncRAqWBQcfqvkupJpw?usp=sharing

This data consists of 19320 entries, equal number of male and female entries.

Methodology

In order to build a classifier, the above-mentioned data was first loaded from the saved CSV file. Initial attempts at solving the task with the whole dataset were unsuccessful due to practical limitations in the hardware (mainly RAM and CPU speed). Due to this limitation, the dataset was reduced to include 3000 samples. These samples were later split into a training(80% of 3000) and test set(20% of 3000). Splitting the data into a training and testing set is common practice within supervised machine learning. This is done in order to be able to test the model on 'unseen' data that was never seen during the training process and give a better indication of the performance of the model.

After the data was loaded and split, the first potential feature that was examined was the blog length. In order to store this a new column was created in the Dataframe which contained the blog length of each corresponding entry. The averages of both male and female authored blogs were computed and compared. It was noted that the percentage difference between the average blog lengths of the 2 genders was insignificant, and so the decision was taken not to build a model based that took the blog length as a feature.

Similarly the readability score was computed using the library found at: https://pypi.org/project/readability/.

The percentage difference of the 2 values was also very low, and so a similar decision was taken not to make use of the readability score as a potential feature.

Following that, a naïve Bayes model using a Bernoulli distribution was built. In order to compute the feature list, the sklearn countvectorizer object was used. 7 Count vectorizers were creating, having the following parameters:

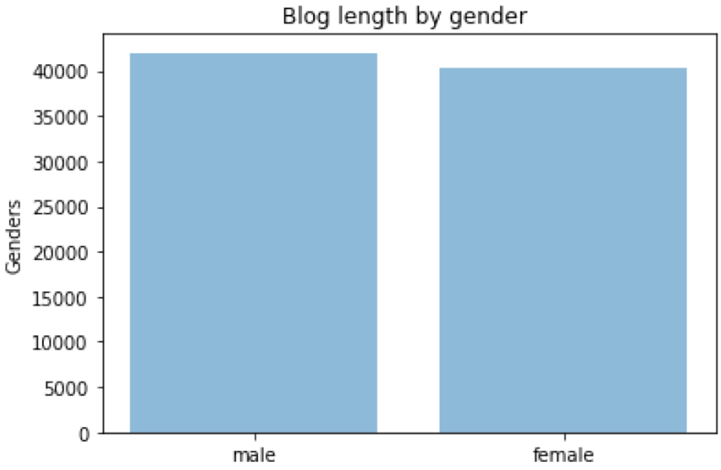
Parameter	**Description**
Ngram_range= (1,3)	Word ngrams containing Trigrams, Bigrams and unigrams
Ngram_range= (3,3)	Word ngrams containing only Trigrams
Ngram_range= (4,4)	Word ngrams containing only ngrams of n=4
Ngram_range= (1,1), analyser = 'char_wb'	Character ngrams of single characters
Ngram_range= (1,2), analyser = 'char_wb'	Character ngrams of single characters and double characters
Stop_words = 'english'	Unigrams, with English stop words removed
NO PARAMETER	Default parameter, without stop words removed

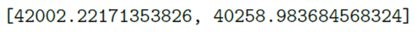
For each counter train features were fitted using the fit_transform method of the countVectorizer. The classifier was then trained to classify the ‘Gender’ label on the given feature list.

Test features were then computed for each countVectorizer respectively and the classifier was run on the test features, and the metrics of the predictions were printed for each classifier.

Results

Results of Blog lengths

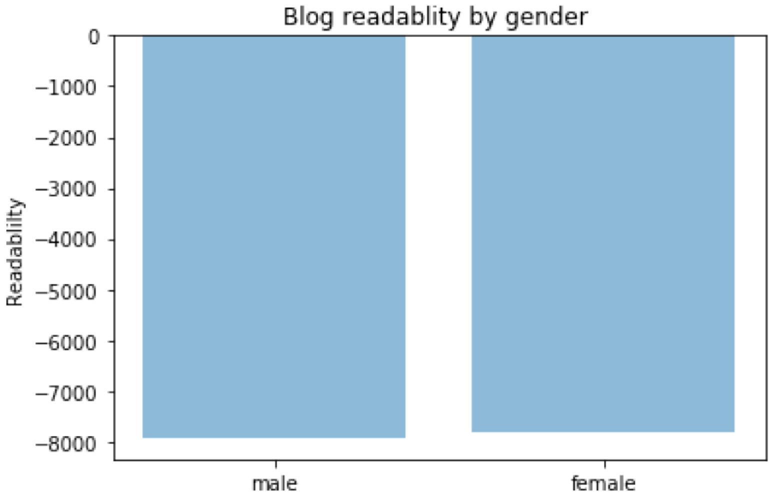




```
[42002.22171353826, 40258.983684568324]
```

Percentage difference : 3 %

Readability Scores



```
-7942.447807842614 -7822.986412130421
```

Naïve Bayes Models

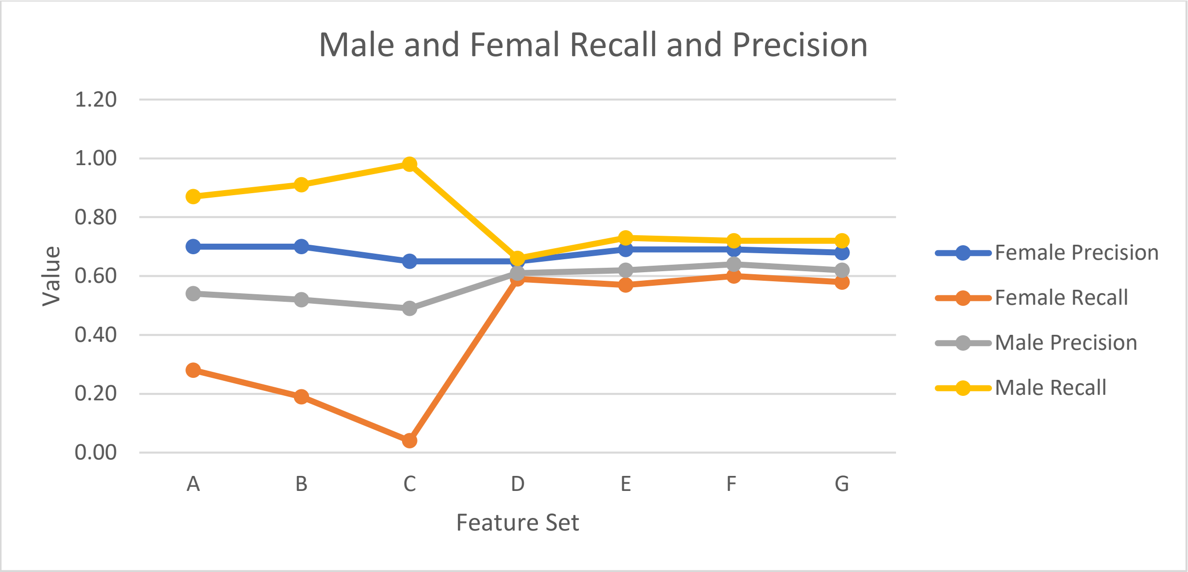
In order to evaluate the naive Bayes models that were built, we must interpret the precision, recall and f1 scores of the model run on the test data.

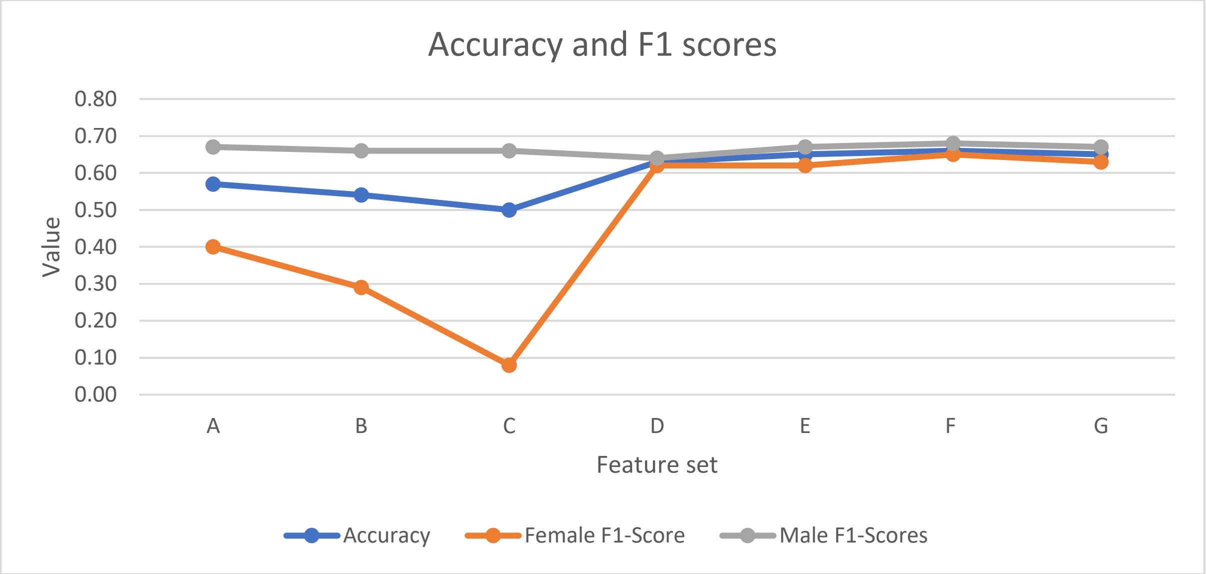
Precision refers to how many of the samples predicted to be positive (in that class), are actually positive. Therefore :True positives / (True positives + false positives).

Recall refers to how many of the samples that where actually labelled positive (in that class) were actually true positive. Therefore :True positives / (True positives + false negatives)

The F1 value is the multiplication of the precision and recall *2, divided by the sum of the precision and recall. This gives an indication of balance between the 2 values and is the harmonised mean.

The result tables of each classifier with the respective features can be found in the Appendix of this report. The following graph represent some of the values obtained





Key	Description
A	Word ngrams containing Trigrams, Bigrams and unigrams
B	Word ngrams containing only Trigrams
C	Word ngrams containing only ngrams of n=4
D	Character ngrams of single characters
E	Character ngrams of single characters and double characters
F	Unigrams, with English stop words removed
G	Default parameter, without stop words removed

Discussion

An important part of creating a model is identifying which features will be used, and how they will be extracted. Following the previously discussed literature, blog size, readability, various word and character ngrams and stop words were considered.

The first potential feature observed was blog size. After computing the average blog lengths and identifying that the differences between the 2 genders were negligible, this potential feature was not processed.

Similarly the readability of each document was also computed. Due to the structures of the parsed blogs, the readability scores were very low (area od -7000). This can be due to the way the blogs were written in an informal range. This unexpected value could also be due to the way the documents were parsed, and since it was not in the expected range, the readability score was discarded as a potential feature. Given more time to work on this task, evaluating why this value was obtained and considering the readability score as a potential feature might add value to the task.

After observing mentioned literature, the Naïve Bayes model with a Bernoulli distribution was chosen as the classifier to be experimented with in this task, with varying feature set, mainly comprised in variations of ngrams. This model assumes independence between features and considered the occurrence of a feature, rather than this frequency. This model was chosen, as the target was to observe whether the presence of certain features (i.e. ngrams) can contribute to identifying the gender of an author.

After observing the result it was noted that the best results were obtained by using the unigrams having stop words removed. This indicated that there were certain words that were made use of by one gender rather than the other. Similar results were obtained using character ngrams, indicating that the presence of certain characters is less common.

It was noted that the overall accuracy of the above-mentioned models was relatively low, with the highest being 0.66. Although this value is better than a randomised guess (i.e. 0.50 accuracy) the error rate is still high. This can be due to the relatively small data sample that was taken, containing only 3000. The reason behind taking only a around 16% of the entire data set was limited computational power on available devices. The task of extracting features of ngrams was time consuming, especially for the case where unigrams, bigram and trigrams were all taken was particularly time and memory consuming.

It was noted that the feature set that gave the worse performance was that Word ngrams containing only ngrams of n=4. This result could be due to the large size of the ngram, which lead to very specific phrases being used as features. Coming across the exact same 4 worked phrase in unseen data must have been uncommon in many cases, leading to the lowest performance. One possible approach to improving this feature set would be to also include all lower ngrams. This was not tested due to the mentioned limitations in computational power.

Possible Improvements and short comings

One potential short coming that might have lead to the limited results, could have been that only basic text pre-processing was done. The initial intention was to leave the text as bear as possible, since there was the possibility of features relating to punctuation and other markers being relevant to the classification. In retrospect, have both the bear text and more pre-processed text and comparing the differences in results would have made this task more complete.

Another potential improvement could be to experiment with and compare different classifiers. (NaCheng, 2010) make use of SVMs and Decision trees in their work on gender identification in email text. This might be work considering on the use of blogs and comparing the results obtained from out classifier.

Another approach that can be considered and compared would be introducing a deep element of neural networks. It would be interesting to compare the results of both a feed forward and recurrent neural network, in order to see whether nay features perform better with the use of history in the work of classification of gender.

Conclusion

The work described in this report reviews the effect of considering the occurrence of different ngrams, classified using a Naive Bayes classifier. It was noted that the best set of features observed were unigrams. The low accuracy rates indicate that the presence ngrams alone might note be sufficient to be used as features to identify the gender of a blog author.

Bibliography

Aayush Singhal, A. P. (n.d.). *Author Profiling from Personal Content Blog.*

BFS4. (n.d.). *Beautiful Soup 4 Parser.* https://beautiful-soup-4.readthedocs.io/en/latest/.

J. Schler, M. K. (2006). Effects of Age and Gender on Blogging. *AAAI Spring Symposium on Computational Approaches for Analyzing Weblogs.*

NaCheng, R. (2010). *Author gender identification from text.* Department of Electrical and Computer Engineering, Stevens Institute of Technology, Hoboken, NJ 07030, USA.

Peng, J. &.-K. (2016). Bit-level N-gram based forensic authorship analysis on social media: Identifying individuals from linguistic profiles. *Journal of Network and Computer Applications*.

Qian, C. H. (2017). Deep Learning based Authorship Identification.

Saleh, A. &. (2014). Naïve Bayes classifiers for authorship attribution of Arabic texts. *Journal of King Saud University - Computer and Information Sciences*.

Aayush Singhal, A. P. (n.d.). *Author Profiling from Personal Content Blog.*

BFS4. (n.d.). *Beautiful Soup 4 Parser.* https://beautiful-soup-4.readthedocs.io/en/latest/.

J. Schler, M. K. (2006). Effects of Age and Gender on Blogging. *AAAI Spring Symposium on Computational Approaches for Analyzing Weblogs.*

NaCheng, R. (2010). *Author gender identification from text.* Department of Electrical and Computer Engineering, Stevens Institute of Technology, Hoboken, NJ 07030, USA.

Peng, J. &.-K. (2016). Bit-level N-gram based forensic authorship analysis on social media: Identifying individuals from linguistic profiles. *Journal of Network and Computer Applications*.

Qian, C. H. (2017). Deep Learning based Authorship Identification.

Saleh, A. &. (2014). Naïve Bayes classifiers for authorship attribution of Arabic texts. *Journal of King Saud University - Computer and Information Sciences*.

Appendix – Full result Values

Word ngrams containing Trigrams, Bigrams and unigrams
Parameter : Ngram_range= (1,3)

	precision	recall	f1-score	support
female	0.70	0.28	0.40	306
male	0.54	0.87	0.67	294
accuracy			0.57	600
macro avg	0.62	0.58	0.53	600
weighted avg	0.62	0.57	0.53	600

Word ngrams containing only Trigrams
Parameter : Ngram_range= (3,3)

	precision	recall	f1-score	support
female	0.70	0.19	0.29	306
male	0.52	0.91	0.66	294
accuracy			0.54	600
macro avg	0.61	0.55	0.48	600
weighted avg	0.61	0.54	0.47	600

Word ngrams containing only ngrams of n=4
Parameter : Ngram_range= (4,4)

95	precision	recall	f1-score	support
female	0.65	0.04	0.08	306
male	0.49	0.98	0.66	294
accuracy			0.50	600
macro avg	0.57	0.51	0.37	600
weighted avg	0.57	0.50	0.36	600

Character ngrams of a single character
Ngram_range= (1,1), analyser = ‘char_wb’

96	precision	recall	f1-score	support
female	0.65	0.59	0.62	306
male	0.61	0.66	0.64	294
accuracy			0.63	600
macro avg	0.63	0.63	0.63	600
weighted avg	0.63	0.63	0.63	600

Character ngrams of a single and double character
Ngram_range= (1,2), analyser = ‘char_wb’

97	precision	recall	f1-score	support
female	0.69	0.57	0.62	306
male	0.62	0.73	0.67	294
accuracy			0.65	600
macro avg	0.65	0.65	0.65	600
weighted avg	0.65	0.65	0.65	600

Unigrams, with English stop words removed

98	precision	recall	f1-score	support
female	0.69	0.60	0.65	306
male	0.64	0.72	0.68	294
accuracy			0.66	600
macro avg	0.66	0.66	0.66	600
weighted avg	0.67	0.66	0.66	600

Unigrams, with default parameter

99	precision	recall	f1-score	support
female	0.68	0.58	0.63	306
male	0.62	0.72	0.67	294
accuracy			0.65	600
macro avg	0.65	0.65	0.65	600