**Project Report**

**Gender Classification (based on any English text content)**

**Introduction**

Our goal is trying to form a machine learning based gender classifier which makes decisions by analyzing input text’s language features, such as words contained, emoticons, suffixes etc. We all know that man and woman talks in a different way, if you can identify the user’s gender who behind the text they post, it will be helpful at situations like anti-terrorist, adjusting the marketing strategy for certain product based on reviews, customer locate or identify a homosexual, pretend to be a girl in online chatting (it can tell you that if you are doing a good job)… well, it should be useful.

**Data**

We download a dataset which they are used to test another classifier which makes decision by analyzing semantic feature of an article (blogs), this data contain 3226 valid blogs, which have total 15 million words, for male, it got 1678 instances, 9 million words, for female, it got 1548 instances and 6million words, it is a relatively big data pool, we think it will show the habit of man and woman talking. We also grab some text from internet to do the random test; they are from users whose gender can be find from profile or their names.

**Method**

We assume that all the text content got three feature score. We will use three features (scores) to judge an instance.

*Word Score:*

We will build a word bag have format as below:



To build a word bag like this, we need to first adjust the labeled dataset we got, it need to be uniform which means similar instances and word counts, at the end, we used all the female data, it is 1548 instances have 639229 words in it, and 1484 out of 1687 instances for male which got 646142 words in it. That makes a relatively uniform dataset.

Then we build our pre-process interface to generate a file like above, we process the all the male instances first, got word frequency, then process all female instances, compare two word frequency files to calculate proportion that man and women use the same word. For example, “football” got a 90% and 10%; it means if “football” appears 100 times in the whole uniform dataset (all instances of women and man), 90 times it appears in a man’s instance, only ten times appear in a women’s instance.

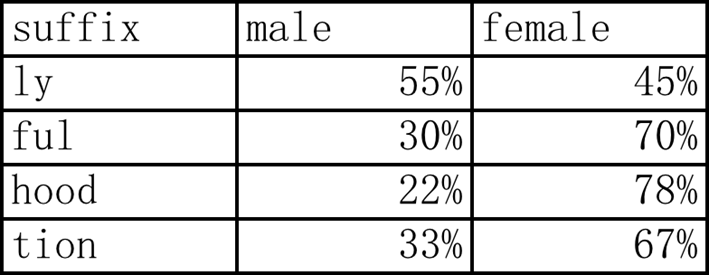
We don’t set any stop word list, because we believe that every word is related to the truth, even word like “the”, ”he”, ”what” etc. will show the special pattern of different gender’s way to speak.

And we made it case-sensitive, we treat “what” and “WHAT” as totally different words, since we believe (actually there is a research says that man like capital letter more) it is part of the habit people typing. Same thing happens on the on purpose typos like “happyyyyyy” and “CCCCCrasy”.

We will assign a word score to any text content processed, if “football” appear in the content, since it got a proportion like “90%,10%” , our score program will add a positive 0.8 (90%-10%) to the content, positive score means biased toward man, negative means biased toward women. All the other scores share this feature too.

All the word that do not exist in our word bag got a (50%, 50%) score.

*Suffix Score*

**

Build suffix list like above, use the same uniform dataset with word score.

*Miscellaneous Score*

For miscellaneous score, this time only special punctuations and emoticons are implemented; link check and number check have not been implemented.

After pre-process, all the instances will turn to be a number set that contain three scores like (1.22, 2.45, -1.11), send this data to our trained classifier, it will show the result.

**Experiments**

At first, we train the classifier by using half of the dataset; it is about 1500 instances and 7 million words. Then we test with the other half of the dataset, it turn out to be about 52% accuracy, which is just like throw a coin.

Then, we try to train the classifier with all the labeled dataset, then process random data which select from the labeled dataset, the accuracy is about 90%.

Then we got about 100 random data from twitter, we got them from users whose gender can be confirmed. The accuracy is still about 50%.

**Related work**

So many other people have done gender classifier before, but the best performance is just about 62%, a little better than throw a coin, so this is really a hard work.

**Conclusion and Future Work**

Our algorithm seems not work very well, future work is about finding out the reason why the accuracy is not good, and try the other methods to enhance the accuracy, if we got more labeled data, the accuracy will increase, maybe building a dynamic word proportion list (self-updated) will help a lot, also the length of the text is very important, if it is short, it will not be sufficient to identify gender.