# Indiana University, Bloomington

FALL 2017

CSCI B645 : Advanced Natural Language Processing

**Final Report**

## Lexical and Statistical Approaches for Genre Detection

Code Reference : https://bitbucket.org/pratsriv/genredetection

Authors:
Inas Nassar
Jeshuran Thangaraj
Prateek Srivastava
Vaishnavi Mukundhan

Date: 12/11/2017

Advisor:
Dr. Damir Cavar

# Abstract

This paper talks about the different approaches for genre detection and their credibility. We examine the roles of different linguistic and statistical features that distinguish the genre classes of text documents automatically. To do this we have taken into account the two main characteristics of text documents viz. its style and content. The current state-of-art systems classifies texts into genres using statistical properties, metadata information[1]. Using Project Gutenberg which provides over 54000 free books, we try to experiment with different ways of classifying texts and finding accuracies of different classifiers.

[1] "recognizing text genres with simple metrics using discriminant analysis." http://www.aclweb.org/anthology/C94-2174. Accessed 13 Dec. 2017.

# 1. Introduction

With the advancements in the field of Natural Language Processing and Data Mining, comes the need of classifying unclassified text. Every research in these fields related to text processing raises the presence of corpus. New corpora that come up from ever increasing number of heterogeneous sources needed for aforementioned researches are needed to be classified with approaches such as some suggested in this paper.
The classification is mainly done by the information present in the corpus. The tangible information that can be extracted from a corpus is not always direct. It can be in the form of implicatures, syntactical information, semantic dissonance and so on.

## Project Goals

Our short term goals focus on researches pertaining on efficient feature extractions for the desired application of documents classification. We will achieve this in 2 ways. We will create naive classifiers with few or more techniques of feature extraction of text to create base classification. This can then be used for classification as explained in the "Methods" section. Second, this will give us some intuition into how much each feature technique contributes to the document similarities or dissimilarities.

**Short Term Goals**

**Feature Extraction**

Our short term goals concentrate on research on classifying genres by different classifiers and measure the accuracy so that we can eventually extract features for the desired application of documents classification.

We will achieve this in 2 ways. We will create naive classifiers with few or more techniques of feature extraction of text to create base classification. This can then be used for classification as explained in the "Method" section.

Second, this will give us some intuition into how much each feature technique contributes to document dissimilarities.

Feature selection is discussed in the "Experiments" section below.

**Proposed Algorithms in Similar Research**

The algorithms proposed are as follows:

- KNN **(used in this paper)**
- Support Vector Machines
- Weak Voting & Boosting
- Neural Networks (CNN/RNN)

**Long Term Goals**

**Domain Independence**

We noticed that most document classification implementations adhere to a single domain. With that observation, we would like to make our algorithms robust to domain change in future prospects.

**Analyze and fine tune sub-classifiers**

Since we are using multiple small classifiers, we would like to do some extensive research to find their importance and fine tune their usage. This will of course change as we use multiple domains.

**Automated Summary**

Our final goal would be to use our genre classifier to extract better summaries from given documents. Most summary generation methods use naive methods which we can try to enhance by using genre knowledge obtained from our classifier.

## Problem Definition

Document classification based on different genres depend on how dissimilar the documents are from one another. There are mainly two types of features that could be used to solve this kind of a problem - Statistical features and Lexical features. Lexical features include features like morphology, word dependencies in a sentence, dependency trees, constituency trees etc. Statistical features include the conversion of sentence to embeddings like word to vectors, frequency distribution of tokens or TF-IDF.

## What is Genre?

Genre of a textual document could be defined or identified in many ways. For instance, if a textual document is classified as a Fiction, it could also be classified under Children's Fiction. This ambiguity in the essence of definition makes this problem harder than it seems.

Different sources where the content can be found might have different list of genre types. However, they all seem to have an overlap. The detection of the strict boundary condition is what makes the problem hard. Developing a standard genre types is something which can't be achieved or we could say hasn't yet been achieved due to the very incompleteness of the definition.

### Why is genre detection important?[2]

Genre Detection holds a lot of value as it is one of the most common methods to organize textual data. It provides a definite hierarchy for the data not just based on its relevance but also its sole objective of creation. Automated genre detection saves the manual work of classifying the different textual data which saves a lot of time and money. This classification also contributes to the metadata information which is used by plethora of other applications which require textual corpora.

[2] "Genre Classification | Digital Curation Centre." 6 Jul. 2008, http://www.dcc.ac.uk/resources/briefing-papers/introduction-curation/genre-classification. Accessed 13 Dec. 2017.

# 2. Method

In this section we will be discussing the multiple approaches which are current genre classifiers and will lead to a single feature vector or a weak voting ensemble in future work depending on the accuracy. We plan to have various classifiers based on different NLP and Machine Learning paradigms to create a single feature which will then be our final feature set for genre classification.

The following techniques are being used for our genre classifiers:

## NLP techniques - Semantic Analysis

**Lemmatization**

In natural language, words appear in different forms for grammatical reasons. The two main forms are inflections, where we just add a suffix to the word without changing its grammatical feature, such as plural forms[‘s] of a word or tenses in verbs [-ing], and derivations, which changes the grammatical form of the word like nation [noun] -> national [verb]. Stemming and Lemmatization normalizes words to their root or common form. We have chosen lemmatization over stemming as it does morphological analysis of words and returns the root form or the dictionary form of the word. For example,

- operational -> research
- operating -> system
- operative -> dentistry[3]

**POS Tagging**

Part-of-Speech tagging will help eliminate the words that don't matter like articles, determiners, modal verbs etc. It would also help in extraction of the content words which carry the important information from a sentence. It can also be used for disambiguation, for instance, in the sentence, "They refuse to permit us to obtain the refuse permit", the first "refuse" is a VBZ and the second is a NN according to Brown Corpus where VBZ is a verb and NN is a noun.

**Anaphora Resolution***

This is a process of of resolving pronouns, verb phrases or whole sentences to items seen earlier or later in the text. For example, in the below sentence, "John loves Mary. She is kind", here "she" refers to "Mary". This would help us make our data set less sparse.

---

[3] "Stemming and lemmatization - Stanford NLP Group." https://nlp.stanford.edu/IR-book/html/htmledition/stemming-and-lemmatization-1.html. Accessed 13 Dec. 2017.

* Not used in the current scope because of lack of computation power but is part of future work

**Dependency Parsing & Constituency Parsing**
These methods will again lead to finding disimilarities between the documents in terms of sentence structures. For example a children's story book will have simple sentences while a research paper will have very complex sentence structure.[4]

**Punctuation Analysis**
This method is similar to voice detection wherein it a children's book or a mystery novel would be more expressive in terms of using punctuation marks while a historic not be as much expressive and lenient in using punctuations.

**Metadata Insertion***
We propose to add metadata information such as Author name, date etc. which will lead to better classification. For example, pre-victorian style of writing is different from contemporary style of writing. This also shows in the genres available in the same durations i.e. different eras have different predominant genres.

**Machine Learning - Syntactic Analysis**
There are multiple techniques used in the NLP industry to analyze word based dissimilarities. There are three methods to analyze documents based on the words.

**Bag of words**
This method maintains a vector of words which will consist of all the words present in all the documents. Each document will have that vector with the respective counts of the words occurring in the corresponding document. For instance, say suppose there are 100 documents with 500 unique words used in them. Each document will have a vector of size 500. Suppose the vector was modeled as [John, Mary, Love, Tim, travel ... ]. Now the corresponding vector for documents will look as follows:

```
v1 = [10,0,1, ...]
v2 = [0,32,5, ...]
        .
        .
        .
```

**TF-IDF**
Term Frequency - Inverse Document Frequency also known as TF-IDF is one of the most common technique used to find words of importance from a document. Unlike bag of words technique, it does not rely on just the frequency distribution of words to rate the importance,

[4] "Using Dependency Relations for Text Classification - Semantic Scholar." https://pdfs.semanticscholar.org/d5b1/820fb3902cbc897bd8e2a63b871a422559f7.pdf. Accessed 13 Dec. 2017.

instead it rates the importance based on occurrence of the term taking every other document in consideration.
There are multiple ways to achieve TF-IDF. We will be using the logarithmic inverse document frequency.

## Classifiers - KNN(K-Nearest Neighbors)

After the creation of feature vectors we classify the given text using K-Nearest Neighbors using the Scikit Learn Library. We experiment with results by tweaking the value of K to get the maximum accuracy provided in the "Experiments and Results" section.

## Future Classifiers

In the above section, we explained the different Linguistic and Machine Learning techniques to classify the documents. However, this section will be giving the above section some perspective of how the different ensembles of classifiers will contribute to the classification of the genres given the text.
We will create multiple weak classifiers and use ensemble methods like boosting to achieve our final classification.

**SVM Approach**

To explain the whole architecture we will have each classifier contributing to a bit or group of bits in the feature vector. Each document will have this final feature vector which will finally be classified using SVM.

**Weak Voting Approach**

This is the ensemble approach to use the multiple classifiers. We will use boosting techniques like AdaBoost to weight the importance of each weak classifier and obtain the classification over multiple iterations.

> For Instance, we could have a weak classifier which is trained to classify genre based on date of publication. This classifier will just give the era of the text eg. Elizabethan era, Victorian era, Gothic era etc.
> This classifier will have a certain weight in the final classification.
>
> Similar such classifiers will classify the same text based on other simple and complicated criterion thereby giving us the final classification.

**Neural Network approach**

We propose to either use a Convolution Neural Network or a Recurrent Neural Network to classify the documents. Unlike the SVM and weak voting approach(above approaches), we will train these neural-nets with the complete document text.

**Note: Our final goal is to compare these methods with each other and against popularly available NLP tools.**

# 3. Experiments and results

## Bigrams of POS tags

For the first experiments, POS tags were used to achieve genre classification. Single word POS tags are pretty uninformative. Therefore, bigrams of POS tags were used.

**Procedure:**

- A single vector of all possible pairs of POS tags was taken as the features.
- The feature vectors were the frequency distribution of each feature assigned to each document in the training set.
- The same pre-processing was done for the test set.
- Classification was done using a K-Nearest Neighbour classifier over the train set.

The intuition behind using POS tags was that the bigrams will implicitly encode the style of writing of the document.
The classifier achieved an average accuracy of 84% over 100 randomized runs.

## Topic modeling

For the second classifier, we tested the topic modelling technique for classification. The intuition behind this classifier was that documents of same genre will at some level talk about the same topics.

**Procedure:**

- For the topic modelling, instead of using LDA, we decided to use basic TF-IDF to gain some insight into the topics and save precious computation time.
- Each feature was a word from the topics of all documents in the train set.
- The feature vector for each document was then a boolean bag of words of each word.
- The test set was pre-processed with the bag of words for each of the words obtained from the train dataset. New topics from the test set were ignored.
- This was then passed into a K-NN classifier using the boolean ‘AND’ operator as a similarity measure.

Extensive testing of the classifier revealed interesting results. (All accuracies are average over 100 runs).

**Results:**

- With lemmatization of topics : 87%
- Without lemmatization of topics: 91%

- Varying number of topics per document in train set:
    - Top 100 topics: 87%
    - Top 50 topics: 89%
    - Top 5 topics: 84%

As you can see, the fact that lemmatization reduces the accuracy is very counterintuitive. We believe this occurs as the richness of the topics is lost during lemmatization. Further research is required to confirm this as it could also be over-fitting.

Varying the number of topics per document gave lots of insight. As you can see the best accuracy from our experimentations was achieved with 50 topics per document. This is most probably due to the fact that the features become very sparse as the number of topics increases.

## Sentence length analysis

The intuition behind the sentence length classifier was that in general different genres have different styles of writing which affect the length of sentences written.
For example, children's books have very long sentences due to excessive use of adjectives and lack of pronouns.

**Procedure:**

- The average sentence length of each document in the training set was obtained from the train and test sets.
- The test set was classified using a K-NN classifier over the train set.

This classifier gave us a bad accuracy of just 32%

On further investigation we found that this was happening due to overlapping data of multiple genres. The mean sentence length of each genre is given below:

```
Mean sentence length for different genres
==================================
Religion  :  28.9276182665
Animals  :  24.4105895584
Medicine  :  27.850378578
Fiction  :  26.4388886411
Children  :  23.0763251987
```

Even though the accuracy is low. Over 5 genres, this is still more than random labeling and hence could give us some intuition in our weak voter or boosting techniques.

## Word length analysis:

Similar to sentence length, the intuition behind the word length classifier was that certain genres will have less or more complicated words.

**Procedure:**

- The feature selected were the average length of the 100 most common words in the document.
- To prevent overfitting, stop words were removed from the documents.
- A K-NN approach was used to classify the test set over the train set.

This classifier suffers the same problems as the sentence length classifier above. Again, the classification is better than random and therefore can be used as a feature in future.

The classifier achieved an average accuracy of 39% over 100 randomized runs.

## Richness of vocabulary

Type token ratio measures the richness of the vocabulary of a textual document. The intuition behind this classifier is that the richer the vocabulary of a text, the advance its reader base would be and thus will have some significant difference in each genres.

**Procedure:**

- Tokenize the whole corpus.
- In our implementation we selected only Nouns and its variation. We could not do anaphora resolution and NER as it took a lot of time with the limited computation power we had for the enormous textual data we were using.
- After filtering the token set to only nouns, we divide the occurrence of the tokens with the total number of words present in that particular document.
- Repeat the process for all documents.
- Each document will have a corresponding type token ratio which will be part of the feature vector.
- Pass this vector to the classifier to get the corresponding prediction.

**Result:**

We could not get results for this classifier due to memory and computing power constraints. An I7 Processor (3.4GHz Clock speed, 16GB RAM) ran out of memory after running for four hours.

## Complexity of sentence

Complexity of a sentence is determined by the presence of sub-clauses. A sentence with just a main or matrix clause is simple, one with embedded clauses is complex. However, clausal detection itself is an elaborate project. We tried using the stanford dependency parser's dependency score because of the enormous dataset. Using the library textstat, we found the complexity classifier to give 32%. The reason for the low accuracy is again due to the ambiguity of the definition of the complexity of a sentence.

# 4. Future Work

In the current work , we develop different classifiers which classify the Textual Data into different Genres. However, the final goal is to join the different classifiers either as an Ensemble or as a Long Feature Vector without losing information. Also, we plan to compare these classifiers with NLP tools available on the market. eg. TiMBL, Mallet.(Future Work)

## Applications

Authorship Attribution:
Build Graph database:

1. use metadata information like author, year publisher house.
2. Scientific/ journals: keywords

## Automatic Summary Generation

This module will generate a summary for different textual data and could be used for Journal Summarization, Thesis Summarization, Movie Script Crux Generation.

## Feature Extraction

**Named Entity Recognizer(NER):** It is an enhancement for the Type Token Ratio Classifier where we just used nouns. We could not used Stanford CoreNLP Parser for NER as we did not have enough computation power, it would have taken months to train with the current computing resources.

**Narrative vs Non-narrative discourse:**
According to one of the papers[5] this idea seems worth exploring. It states as follows.

> "Certain types of narrative discourse (autobiography and narratives of personal experience) naturalize the notion of the original utterance, since the narrative purports to represent or re-present historical or biographical incident. In contrast, the occurrence of direct speech in non-narrative discourse unmasks or foregrounds the constructed, rhetorical nature of direct speech 'reporting' in these discourse contexts"

[5] "Direct speech: What's it doing in non-narrative ... - ScienceDirect.com." https://www.sciencedirect.com/science/article/pii/0378216694000743. Accessed 14 Dec. 2017.

# References

Biber, Douglas. “Variation across Speech and Writing.” 1988, doi:10.1017/cbo9780511621024.

“The Difference Between Narrative & Non-Narrative Writing.” *Synonym*, classroom.synonym.com/difference-between-narrative-nonnarrative-writing-12136843.html.

Karlgren, Jussi, and Douglass Cutting. “Recognizing Text Genres with Simple Metrics Using Discriminant Analysis.” *Proceedings of the 15th Conference on Computational Linguistics -*, 1994, doi:10.3115/991250.991324.

Sheikha, Fadi Abu, and Diana Inkpen. “Automatic Classification of Documents by Formality.” *Proceedings of the 6th International Conference on Natural Language Processing and Knowledge Engineering(NLPKE-2010)*, 2010, doi:10.1109/nlpke.2010.5587767.

Xu, Zhijuan, et al. “Text Genre Classification Research.” *2017 International Conference on Computer, Information and Telecommunication Systems (CITS)*, 2017, doi:10.1109/cits.2017.8035329.