Genre Identification on Gutenberg Corpus

Tarun Gupta
Digital Engineering
Otto von Guericke University
Magdeburg, Germany
tarun.gupta@ovgu.de

Ritu Gahir
Data and Knowledge Engineering
Otto von Guericke University
Magdeburg, Germany
ritu.gahir@st.ovgu.de

Libin Kutty
Data and Knowledge Engineering
Otto von Guericke University
Magdeburg, Germany
libin.kutty@st.ovgu.de

Mohammed Aftab
Digital Engineering
Otto von Guericke University
Magdeburg, Germany
mohammed.altaf@ovgu.de

Shipra Dureja
Data and Knowledge Engineering
Otto von Guericke University
Magdeburg, Germany
shipra.dureja@st.ovgu.de

I. MOTIVATION AND PROBLEM STATEMENT

Project Gutenberg [2] is a library of over 60,000 free eBooks but for the purpose of this project we have used 996 books in HTML format with 9 labels (genres). The problem statement is to build a genre classification/detection model of a book. Each data-point in this classification task is a fiction book with a label (genre).

The motivation for this task is to determine and extract semantic features from the books and build a genre classification/detection model of a book based on the semantic features and evaluate the performance with the rudimentary Bag-of-Words(BoW) approach.

The references for semantic feature to be extracted were motivated by SIMFIC implementation [5].

II. DATA SET

The data set consists of HTML documents that need to be parsed and made HTML tags free in order to make it usable. For the book name, author and genre of a book, we have a master **csv** file with the mapping of book name, author, HTML file name and the genre. The data set brings out the problem of class imbalance as there are genres [Literary, Detective and Mystery] which have more than 90% [79%, 11%] of the data set respectively.

A sample data set HTML file looks like the following:

<p>"You'll have to get off, sir."

<p>"That's right - at Tucson."

The Gutenberg Dataset is divided into training and test split with 70% data (books) in training set and 30% in test set. The classifiers are then trained on training set and evaluated on test set.

Since we have the problem of class imbalance, we have decided to augment our data which is being used for training the classifier. With the help of 'WordNet', we are replacing word from the data set which has synonym in the WordNet and creating new data. WordNet is a lexical database of English words which resembles Thesaurus [3].

For the semantic feature approach, we have augmented data of the 4 least represented genres ['Allegories', 'Christmas Stories', 'Ghost and Horror', 'Humorous and Wit and Satire'] so that they are represented 10 times each in the training set to manage the problem of class imbalance.

III. CONCEPT

For a supervised machine learning problem, we try different algorithms and techniques to form a general hypotheses. The steps required for pre-processing are applied on the data in order to input it to the classification model. The 2 approaches used are BoW approach, which does not take any semantic relations into consideration in the corpus, and semantic feature extraction which provide us information about the book such as Parts of Speech, Rural or Urban Setting, Sentiments, Ease of Readability of the book and Lexical Richness.

The rudimentary approach of BoW give us a sparse matrix which is lengthy and hence can have higher computational requirements. The more advanced approach of semantic feature extraction gives us a dense matrix and has lower computational requirements. This report in section V discusses the comparison of both models along with reduced feature space on the latter approach described in section IV-C.

For the scope of this classification/genre detection task, we have used 3 classifier models in all the 3 approaches (Bag-of Words, All Features and Reduced Space)

- Random Forest Classifier with 100 *estimators/trees*
- Support Vector Machine (SVM) with *rbf kernel*
- Multi Layer Perceptron (MLP) with 100 *hidden layers*

A. Bag-of-Words

For the machine learning algorithms to understand the text present in provided books, we need to represent the text in a particular format so that it becomes readable by the classification algorithms. The simplest representation of any given text document is BoW, wherein, the text is represented as a count of the number of times the words appeared in the given document. Extracting features from text and representing

in BoW format requires few pre-processing steps which deal with dividing the text into tokens, removal of stopwords, and also representing the words in their stemmed form. The BoW concept produces a large and sparse matrix on which the learning is done by the classification algorithms.

In our project, we have used BoW as ground truth, and have compared the classification models' performance on BoW representation to the performance achieved after extracting semantic features for the given text books.

B. Semantic Features

Semantic features represent basic meaning of any lexical item. It provides a relation between words which is ideally lost in BoW model. The semantic feature can be extracted on word level as well as on the whole text. Few semantic features extracted in this project and the underlying importance of each feature is described in the following subsections.

1) POS tagging : Authors have unique style of writing and most of the authors scribe around certain subjects which has a close relation to the genre [7]. Parts of speech are important in order to understand the underlying meaning of the sentence. Same sentence can imply different meanings without the proper use of parts of speech. In general, there are 8 parts of speech(namely: Noun, verb, pronoun, adverb, adjective, preposition, conjunction and interjection) but each can have many types/variants in themselves which gives more specific or generic details of the usage of the word [8]. For Example: A Noun can be common/ proper/ abstract/ collective/ compound/ possessive/ material/countable noun. Each giving relative meaning. How can these be used in genre classification? Taking an example from our data set, one of the genre is Christmas stories. The books pertaining to this genre mostly consists of Nouns like Jesus, Marry, God, Gospel, Church and more. To be specific these are proper nouns. Statistically from Table I, it can be seen that its nouns constitute 25% which becomes a distinguishing feature for this genre.

Thus taking these intricacies in consideration we have chosen 35 categories of 8 standard parts of speech.

2) Rural or Urban Setting: This feature helps us in determining if the book is more of storytelling or first-person narration. Quotes and number of characters help to identify the type of storytelling. Fewer quotes and the number of characters describe more about first-person narration while the higher usage of these illustrate the storytelling books.

- Quotes - Quotes mostly determines dialogue spoken by characters in the story or represents phrases or important text present in the book. Quotes gives an idea about amount of conversation done in a book. It can be determined with the help of pattern matching.
- Number of Characters - Characters are known as any person, animal, or figure represented in the literature. Many types of characters exist in the literature, each with their characteristics. We are extracting character information with the help of Named Entity Recognition. For Example:

  {'PhilipVantine': 'PERSON', 'Parks': 'PERSON', 'Lester': 'PERSON', 'Louis': 'PERSON', 'MichaelAngelo': 'PERSON'}

  The drawback of this method is that there could be a possibility that a name may be written in a short form or in another form; for example, Mark Johnson can be written as Mr. Mark or Mr. Johnson which can be the name of a different person or same person and our method considers this as a different person which can increase the count which can lead to miscalculation of the number of characters.

3) Sentiment: Sentiment plays an important role in identifying the genre of the book. It interprets and identifies the overall emotion of a book, which can be used to categorise the book with same sentiments together under similar genres. Sentiment is broadly divided into positive, negative and neutral sentiments, which depends on the nature of the words used in the book. If positive words like happiness, accomplishment, fine, popular etc are used in a book, the sentiment of a book is defined as positive. Similarly, if negative words like hate, anger, failure, etc. are used in a book, the sentiment of the book is defined as negative. On the other hand, if the number of positive words and negative words are balanced throughout the book, the sentiment is defined as neutral.

Generally, it is observed that in books the sentiment is not usually uniform throughout the book. It might happen that the sentiment appears to be slightly positive in the first half of the book, but changes drastically to be negative in the second half of the book. In such scenarios, identifying the sentiment on the whole book at a time doesn't make sense and may direct the classification process into wrong direction. In order to cope up with the problem, we have divided the whole text of a book into chunks of 10000 words in a circular fashion and identified the sentiment on each chunk. Once done, we have averaged the sentiment identified on each chunk in order to get the overall sentiment of the book.

4) Ease of Readability: The easiness to read a book is an important factor in a book. This feature provides the idea on how easy or difficult a book is to read. There are multiple measures available that provide a quantifiable value of ease of readability of a book. We have used one such measure called Flesch reading score [4]. It uses word length and sentence length to compute the score. The use of this feature is widely present in the tools for text processing such as *Microsoft Office Word*, *Grammarly*.

5) Lexical Richness: Lexical richness can be defined as the measure of the quality of the vocabulary. It quantifies the degree to which a unique and varied vocabulary is used in the text. To look objectively, if we take the importance of genre then there will be a close relationship between vocabulary and the authors writing that content [1]. It helps classify the authors based on their proficiency and vocabulary growth. Learners might pick the book of a particular genre for his/her vocabulary growth. A writer/author can have a large vocabulary base. However, the lexical richness can not be defined only based on vocabulary size. It can vary due

TABLE I
STATISTICAL COMPARISON OF NOUNS WITH OTHER PARTS OF SPEECH

Book Name	Author	Singluar Nouns (%)	Plural Nouns (%)	Proper Noun (Singlar %)	Proper Noun (Plural %)	Weightage
The Seven Poor Travellers	Dickens| Charles	14.21	3.98	5.11	0.15	23.45
Blade-O'-Grass.	Farjeon| B. L.	13.49	3.74	4.18	0.01	21.42
The Prodigal Village	Bachelle| Irving	15.09	3.79	6.55	0.04	25.47
A Christmas Carol	Dickens| Charles	14.22	3.53	4.49	0.01	22.25

TABLE II
FLESCH READING SCORE AND INTERPRETATIONS

Score	School level	Notes
100.00–90.00	5th grade	Very easy to read. Easily understood by an average 11-year-old student.
90.0–80.0	6th grade	Easy to read. Conversational English for consumers.
80.0–70.0	7th grade	Fairly easy to read.
70.0–60.0	8th & 9th grade	Plain English. Easily understood by 13- to 15-year-old students.
60.0–50.0	10th to 12th grade	Fairly difficult to read.
50.0–30.0	College	Difficult to read.
30.0–10.0	College graduate	Very difficult to read. Best understood by university graduates.
10.0–0.0	Professional	Extremely difficult to read. Best understood by university graduates.

to language proficiency, the familiarity of the field, writing skills, and much more. Lexical richness can be described by several measures like originality, density, sophistication, and variation. We used Lexical Variation(LV), that can be defined with type/token ratio. The ratio of type(unique words) in the content to the total number of tokens(words).

$$LV/TTR = \frac{Numberoftypes}{Numberoftokens}$$

Type/token ratio(TTR) is highly affected by the length of the text and can be a bit unstable for shorter texts. This makes TTR unsuitable for short texts like essays [1]. On the other hand, if it is calculated over the entire book, then lengthier books will have less TTR in comparison to shorter books/novels. Hence, we have calculated the lexical richness at the chunk level which is at-last averaged to get the overall TTR of the book. TTR is dependent on the definition of the words. It consider words and its derivative to be different. A higher value of TTR will be seen if the word family is considered to be one single word. Though it does not indicate rich vocabulary in terms of word families used. It can not differentiate between a learner who has used derived forms of words from a few families from the learner who has used different word families, as it can not distinguish between what different kind of words are used [1]

IV. IMPLEMENTATION

A. Bag-of-Words

For the BoW implementation, we have used **nltk** python package and performed pre-processing on text of each book, which includes splitting text into tokens, doing stopword elimination and performing stemming. These words are considered as features and the weights are assigned to each word through **TfidfVectorizer** from python's **scikit-learn** package. Features are extracted on the data set and are represented using the above mentioned method. This feature representation is used as input for machine learning algorithm to learn and the performance of each algorithm for the genre classification task is evaluated.

B. Semantic Features

1) POS tags: Tags were extracted using **textBlob** package which internally uses **nltk** library(Penn Tree bank free version). The methods in **textBlob** performs parts of speech tagging on each word in a sentence. We summed up individual tags per text book and normalized it with respect to sum of all the extracted tags. **textBlob** supports the extraction of most used POS tags. Supported POS tag Table III

$$TAG = \frac{Sumofeachtagperbook}{sumofallthetagsperbook}$$

2) Rural or Urban Setting: This feature is extracted with the help of **nltk** package. The feature vector will be of only one dimension as only count is stored.

- Quotes - We are extracting quotes based on pattern matching. We are finding all the quotes present throughout the book and storing the number of quotations present in the whole book.
- Number of characters - We are extracting character detail with the help of Named Entity Recognition(NER). NER helps in identifying key points of the text like person, organization, location and more. With NER, we are finding all the unique people present in the whole book and storing the number of unique characters present in the book. With the assumption that there won't be many changes in the name of the person.

3) Sentiment : As mentioned in section [III-B3], instead of determining sentiment on whole book, we have calculated the sentiments at chunk level which is at-last averaged to get the overall sentiment of the book. From the implementation perspective, we have used **textblob** python package to extract

TABLE III
PARTS OF SPEECH TAG IN PENN TREE BANK

Number	Tag	Description
1	CC	Coordinating conjunction
2	CD	Cardinal number
3	DT	Determiner
4	EX	Existential there
5	FW	Foreign word
6	IN	Preposition or subordinating conjunction
7	JJ	Adjective
8	JJR	Adjective, comparative
9	JJS	Adjective, superlative
10	LS	List item marker
11	MD	Modal
12	NN	Noun, singular or mass
13	NNS	Noun, plural
14	NNP	Proper noun, singular
15	NNPS	Proper noun, plural
16	PDT	Predeterminer
17	POS	Possessive ending
18	PRP	Personal pronoun
19	PRP$	Possessive pronoun
20	RB	Adverb
21	RBR	Adverb, comparative
22	RBS	Adverb, superlative
23	RP	Particle
24	SYM	Symbol
25	TO	*to*
26	UH	Interjection
27	VB	Verb, base form
28	VBD	Verb, past tense
29	VBG	Verb, gerund or present participle
30	VBN	Verb, past participle
31	VBP	Verb, non-3rd person singular present
32	VBZ	Verb, 3rd person singular present
33	WDT	Wh-determiner
34	WP	Wh-pronoun
35	WP$	Possessive wh-pronoun
36	WRB	Wh-adverb

the sentiment of each chunk of the book comprising of 10000 words. Sentiment of the chunk is defined by the polarity score returned by **textblob.sentiment**. Polarity score is a float value that ranges between -1 and 1. The polarity score is interpreted as how positive or negative the sentiment of the sentence is, where a score of 1 gives highly positive sentiment, score of -1 gives highly negative sentiment and score of 0 gives neutral sentiment of the sentence. The value of the polarity score is used to determine how similar the sentiments of two books are and the books with similar sentiment score are classified under same genre by the classifier after getting trained on the training data.

4) Ease of Readability: This feature comprises of calculation of one score - Flesch reading-ease Score(FRES). This score is computed by the following formula [4]:

$$FRES = 206.835 - 1.015(\frac{totalwords}{totalsentences}) - 84.6(\frac{totalsyllables}{totalwords})$$

This feature provides us a numeric value on how a text is easy to read. The Table II from [4] provides information on how to interpret the score values given by this metric. In our implementation, we have used circular chunking with a chunk size of 10,000 words. For the last chunk where it can be smaller than 10,000 words, we have appended the text from beginning of the book to make it a 10,000 words chunk. For programmatic implementation, we have used a python package **textstat** which uses the equation above IV-B4. ***FRES*** is calculated on each chunk of a book and all the ***FRES*** scores of the chunks are averaged over a book to get the average ***FRES*** score for the book.

5) Lexical Richness: TTR is calculated over the entire book. Length of the book or novel affects the TTR value. Longer the book, smaller is the value of TTR. Taking in account this problem, we have used circular chunking similar to the other features over the book. TTR considers unique words, so the chunked text needs to be free of stopwords and punctuations. We pre-processed the data using python's **nltk** package by tokenizing the text and removing the stopwords and punctuations. For instance, in a composition of 500 tokens, say, 200-word types are to be used by the learner who has information about 10,000 words, or another learner who knows 15,000 words. The 500 types of words in the two contents may be from different proficiency interests, even though the number of word types is identical. TTR stresses on how good a learner can express herself/himself with the vocabulary she/he knows and not what types of words she/he knows. TTR distinguishes only between the different words used in the content, but not between the quality of the different words as defined by their uniqueness because similar TTR value can indicate different vocabulary sizes in terms of lexical richness.

C. Reduced Feature Space

We reduced the POS tags from 36 to the standard tags of 8(parts of speech). However, a case study shows that fine grained POS features produces much more better results when compared to Impoverished POS features [6]. Complete Penn tree bank POS tags are not open sourced thus we could not extract all the features.

For the approaches of Semantic Features and its reduced feature space, we have used SMOTE to oversample and then balance the number of training instances on which the classifier is learned.

V. EVALUATION

We have selected 3 classification models for the genre classification task. We are evaluating the performance of our selected models on the BoW feature space, whole semantic feature space and reduced semantic feature space. The evaluation measures used here for model comparison are precision, recall and f1-measure, however, we did not include accuracy as the data set at hand has class imbalance problem. This problem might lead to high accuracy score since the prediction might always be the majority class, which might not always be the case.

TABLE IV
EVALUATION

Model	Classifier	Precision	Recall	F1-measure
BoW	Random Forest	0.785	0.721	0.803
	SVM	0.785	0.721	0.803
	MLP	0.811	0.799	0.843
All Features	Random Forest	0.815	0.828	0.846
	SVM	0.794	0.803	0.816
	MLP	0.788	0.803	0.819
Reduced Space	Random Forest	0.773	0.786	0.806
	SVM	0.782	0.721	0.686
	MLP	0.771	0.745	0.732

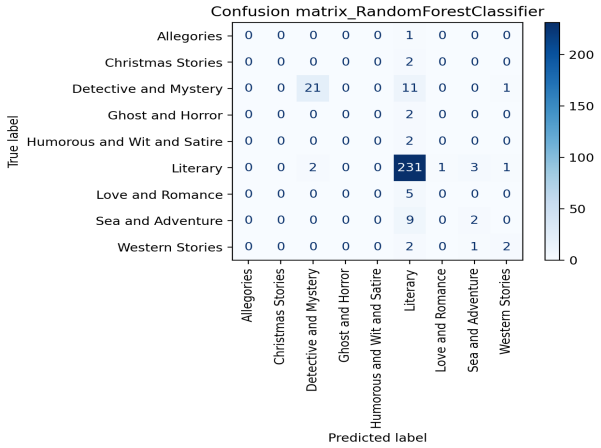

Fig. 1. Confusion Matrix-Random Forest for all Features

A. Results and Interpretation

As shown in fig.[4] and [5], it can be interpreted that the performance of classifiers on both BoW and Semantic features is almost same. The only difference observed is that Multi Layer Perceptron classifier performed better on BoW compared to other classifiers. On the other hand, Random Forest works better on semantic features.

Moreover, on further reducing the semantic feature space, our performance was reduced a bit (by 4%) because less feature possess less information. But, also since the number of features are less the time taken to train the classifier is reduced. Hence, we are not reducing the semantic features and keeping a total of 41 features.

VI. CONCLUSION

In this project, we aimed at performing genre classification for the query book preferring the approach based on semantic features instead of BoW features. We evaluated the performance of three classifiers namely Multi Layer Perceptron, Support Vector Classifier and Random Forest Classifier based on precision, recall and f1-measure on BoW feature space, semantic feature space, and reduced feature space. On comparing the results, it is observed that our semantic feature space with only 41 features gives better result than the BoW feature space and also reduces the feature vector size. The results shown in Table IV show that the Random Forest Classifier on semantic features perform the best out of the 3 approaches used. The

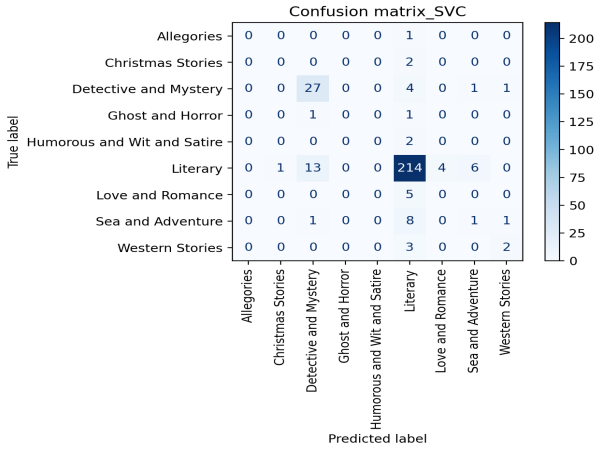

Fig. 2. Confusion Matrix-SVM for all Features

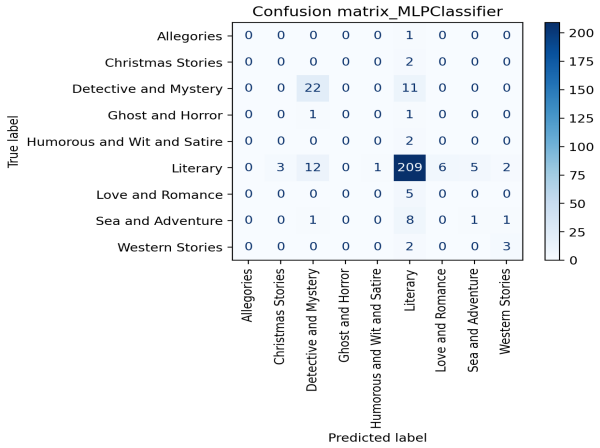

Fig. 3. Confusion Matrix-MLP for all Features

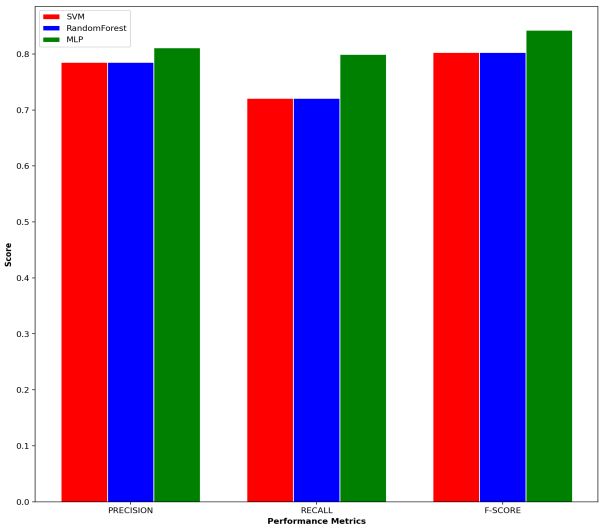

Fig. 4. Model Comparison on Precision, Recall and F1-measure with Bag-of-Words

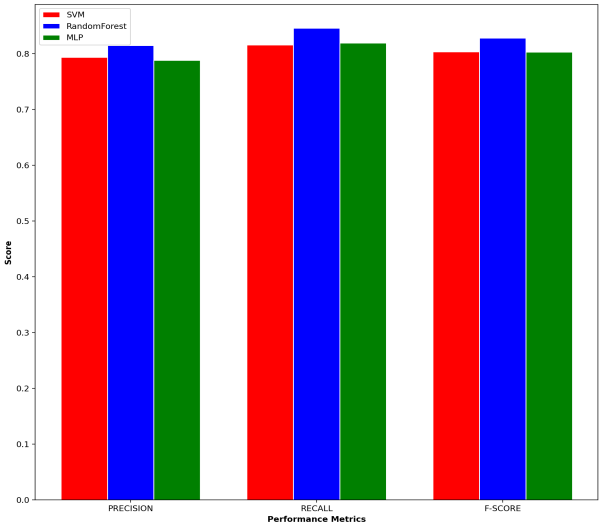

Fig. 5. Model Comparison on Precision, Recall and F1-measure with all semantic features

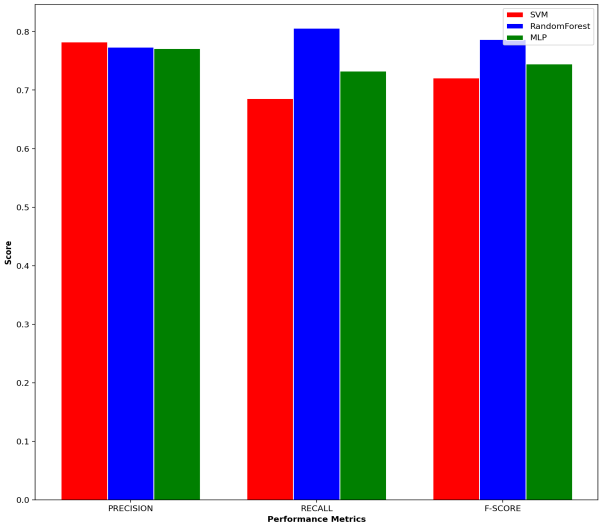

Fig. 6. Model Comparison on Precision, Recall and F1-measure with reduced semantic features

conclusion we reach is that use of semantic features supports our motivation that the computational requirements are lower as compared to BoW approach and we also achieved a slight improvement in performance as well.

VII. FUTURE WORK

As part of future work, we can include more features that have been researched upon for textual classifications of books apart from the ones we have used and integrate them into our classifier training. Currently, to handle the class imbalance problem we have augmented the data of minority genres to 10 instances each based on 'WordNet' Thesaurus, we can try to augment data to more instances or include multiple Thesauruses or more dense Thesaurus to represent the minority classes better.

REFERENCES

[1] Laufer, Batia, and Paul Nation. "Vocabulary size and use: Lexical richness in L2 written production." Applied linguistics 16, no. 3 (1995): 307-322.
[2] https://www.gutenberg.org
[3] https://wordnet.princeton.edu/
[4] Flesch, Rudolf. "How to write plain english: a book for consumers and lawyers." (1979).
[5] Polley, S., Thiel, M., Kotzyba, M., & Andreas, N. (n.d.). SIMFIC: An Explainable Book Search Companion.
[6] A. C. Fang and J. Cao, "Enhanced genre classification through linguistically fine-grained POS tags," PACLIC 24 - Proc. 24th Pacific Asia Conf. Lang. Inf. Comput., no. 2001, pp. 85–94, 2010.
[7] Matthew L. Jockers. Macroanalysis: Digital Methods and Literary History. University of Illinois Press, Champaign, IL, USA, 1st edition, 2013.
[8] https : / / grammar . yourdictionary . com / parts-of-speech / nouns / types-of-nouns.html