CLASSIFICATION SYSTEM USING HYBRID COMPARISON METHODS

Thiri Kyaw[1], Zin Mar Htun[2], Khaing Thanda Swe [3]

Lecturer, MTU, Myanmar

Abstract

When the genre of the book is classified, it is a tedious task to manually read the entire book. Automated book classification uses text based comparison of book summaries to examine whether word similarity is a feasible method to identify the genre of the book. Knowing the genre means, the possible content of the book can be guessed and easier to decide if we like it or not. Automated book classification can automatically generate genre labels for the book. This system focuses on bag of words approach, score comparison method and percentage comparison method to classify genres. Evaluation is done in terms of recall and precision values. Cross-validation and sanity check are tested. The experimental results of the proposed system are over 80% for both precision and recall values. The proposed system is implemented with Python Programming.

Keyword: Cross-validation, Score and Python

1.INTRODUCTION

Automated book classification is generally defined as content-based assignment of one or more predefined categories/genres to books. When the genre of the book is classified, the contents of the book are usually manually analyzed. It is time-consuming. So, automated book classification can be used to get the genres of the books.

The problem about book classification is that there is no one correct set of rules of identifying a book as belonging to a specific genre, as the rules for identification change. Genre definitions can differ based on society, country, and person to person. So it is important to find a way to classify books and their degree of relativity to a given genre.

For automated book classification, there are many algorithms, which are based on different approaches. In this thesis, bag of words approach is used to count the words in the summary. Score comparison method and percentage comparison method are used to identify the genre of the book. Evaluation is done in the terms of precision and recall. Cross-validation and sanity check are tested.

2. LITERATURE SURVEYS

The work of Santini has centered on automatic genre identification [1]. Santini is interested in this field because of its applications in grouping unknown web pages together. Web pages can be grouped according to genre as opposed to being just by topic.

Santini is approaching the problem using natural language processing and machine learning. She uses a variety of 'facets' to identify possible genres [07Mar]. Her goal is to have the machine learn from the data sets how to identify the specific facets and put them together to get better classification genres for the web pages she was looking through [2].

Jordan [2] uses book summary to predict the genre. The author gets word frequency for the predefined genres and gives the scores by comparing their word counts to get the genre. The author's goal is to write a program that can predict the genre of the book based on book summaries to classify books in library.

Text classification is to assign a document to one or more classes or categories. The idea behind text classification is that by splitting pages into groupings, one might be able to gather more information. For instance, when classifying a newspaper a text classifier might split articles by topic. Or it might split email messages into the categories of spam and not spam [3]. Text classification is used for a wide variety of tasks such

as sentiment analysis, topic classification, spam pages, web filtering, and a variety of other jobs [4]. Text classification includes techniques such as Naive Bayes, Tf-idf, latent semantic indexing, support vector machines, decision trees, and natural language processing.

Petrenz took a look at this very problem. In his paper titled Assessing Approaches to Genre Classification, Petrenz examined four methods of genre classification to see how they would perform on a formerly unseen volume of text. He wanted to see if a change in the style of the text they were analyzing would change the results or not. The methods examined included parts of speech tagging, the use of heuristics, and bag of words using support vector machines to predict genre classes [5].

Petrenz's paper illustrated how different text classification methods can all yield results when put to the task of identifying genre, some better than others. Petrenz's work is important because there needs to be methods of baseline comparison established in order to fully assess which methods are actually classifying genres better than others. Currently, each method has its own test data, its own training data, and own methods of determining results [6].

Genre theory is the field that is attempting to define and understand what genre is [7]. Unfortunately, as they are still working on what genre is, there hasn't been as much work as there could be into the how of classifying them automatically [09Phi]. There are not any universally agreed upon algorithms for labeling the genre of a document. The methods that do exist seem to agree that a multi-faceted approach is needed in order to truly identify a genre. As genres encompass all that a text is, it takes more than one classifier to accurately identify the genre.

Much of the current work on genre theory in the classification area has been done in the area of web genre. Santini and Crowston [8] tried different approaches in an attempt to find classifiers that would work for defining genres on the web. Both faced problems mentioned by Chandler in his work. While genre works to create organization, there are still no absolute ways to classify works which is why the field of genre theory is still evolving. This is why the work being done in this field encapsulates such a wide variety of techniques from the field of text classification.

3. METHODOLOGY PHASE

This section introduces some basic principles of comparison methods such as score comparison and percent comparison.

3.1. Score Comparison Method

The score comparison method system classifies the books by giving them total points score based on which words occur in the books. Once it has completed an execution, each genre type has a final score that can be used to determine which genre had the strongest influence upon the book. Words that occurred more frequently in a given genre should be given more points as they belong more strongly to that genre [8].

Firstly, all the words from book data are counted. The words are stored in the following form. The counts are in order of genre,that is [fantasy, mystery, romance, sci-fi, drama].

word1 [count, count, count, count, count]
word2 [count, count, count, count, count]

The second stage examines the completed list of words and ranks them based on their number of occurrences. It looks at each count and calculates the ordering of the genres based on their counts. It then assigns points to each genre based upon their ranking [9].

cliff [14, 3, 12, 0, 1]
masters [10, 1, 10, 3, 2]

Consider we have the phrase „cliff masters" in the book summary. For the word „cliff", it is 14 word counts in fantasy genre, 3 word counts in mystery genre ,12 word counts in romance genre,0 word count in Sci-fi genre,1 word count in drama genre. The scores are ranked from five to one in order, though if a genre does not have the word at all, it is given zero points.

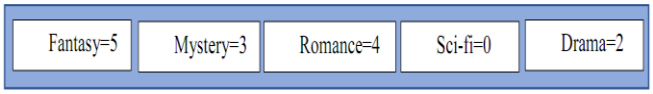

Figure 1. Awarded Points for Words "cliff"

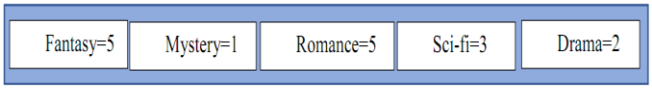

Figure 2. Awarded Points for Words "master"

Figure 1 is genre ranking of the word „cliff". Figure 3.2 is genre ranking of the word „masters". Fantasy and romance had the same starting score. For this reason they are both given five points as they are tied for first place. The other three genres are given the ranks of three, two, and one.

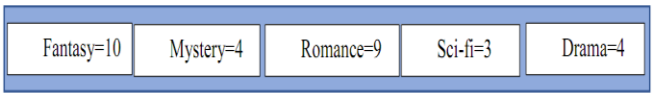

Figure 3. Final Awarded Words for "cliff masters"

When the points are combined to see how the phrase would rank, the phrase is classified most strongly as fantasy, with romance in second place.

3.2. Percent Comparison Method

The percent comparison method takes into account the size of the word collection for each genre. First, it adds up the total word counts for each genre. Then it calculates how many words are stored for that genre. It then divides the word counts by that total to get the percentage that word occurred in the given genre [10].
Once the percentages have been calculated, the program figures out the word counts for the unknown book summary. Once tabulated, the program then calculates the percentage each word occurs in that summary. After that computation has been completed, the program then multiplies the percentage the word occurred in the summary against the percentage the word occurred in each genre to calculate the similarity scores. The percentages of similarity are added up, and the genre with the highest total percentage of similarity is deemed the most similar genre.
Example: genres are listed in order of fantasy, mystery, romance, sci-fi, and drama.
Wordlist
cat [5%, 10%, 25%, 15%, 5%]
dog [10 %, 15%, 20%, 10%, 5%]
horse[10%, 10%, 10%, 5%, 20%]
In the wordlist, one can observe that there are three words in the training data. Listed in the brackets next to each word are the percentage that each of these training data words occurred in the given genre.
Sample Sentence: Cat dog horse horse
cat = 25%, dog = 25%, horse = 50%

In the above sample sentence, the percentage each word is a part of the overall sentence is displayed. So, since "horse" occurred twice in the four word sentence, it is assigned a 50%. "Cat" and "dog" each are given 25% as they occur once in the four word sentence each. Once all entries have been calculated, the percentages of each genre are simply added up. Whichever genre has the highest percentage score is labeled as first, the next as second, etc.

Computation comparison:

cat [.25 * .05, .25 * .1, .25 * .25, .25 * .15 .25 * .05]
dog [.25 * .1, .25 * .15, .25 * .2, .25 * .1, .25 * .05]
horse [.5 * .1, .5 * .1, .5 * .1, .5 * .05, .5 * .2]
Totals: .0875, .1125, .1625, .0875, .125

As can be seen in the example, while romance had the word "cat" and "dog" very often compare to the other genres.

4. DESIGN OF PROPOSED SYSTEM

The implementation steps of Automated Book Classification will be shown in Figure 4. In this system, bag of words model is used to get the word counts for book summaries. Score comparison method and percentage comparison method are used for classification.

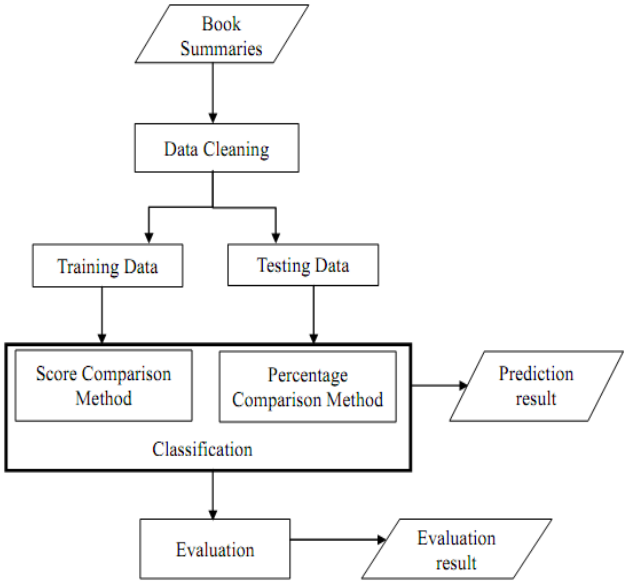

Figure 4. Flow Diagram of Proposed System

The proposed system summarizes the book according to the classification. First the system needs to transform the data for data cleaning into two phases: training data and testing data. Then the system is applied into classification methods: score comparison and percentage comparison. After that the system gives the predicted results to the users.

5. TEST AND IMPLEMENTATION PHASE

5.1. Input Data

The dataset used in this system contains around 16000 book summaries along with their genres, publication dates, authors and other information. It is CMU Book Summary Dataset. Firstly, unwanted information is deleted and only kept title, summary and genre.

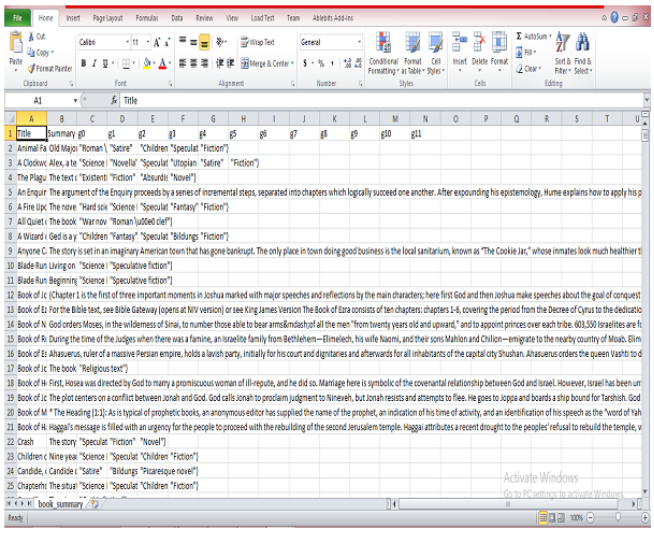

Figure 5. Dataset After Removing Unwanted Information

Figure 5 is dataset after removing unwanted information. Then books, which genres are fantasy, mystery, romance, sci-fi or drama, are only kept. For cross validation test, dataset is divided into training dataset and testing dataset. If only one sample summary is tested to predict unknown genre of that book, sample summary can be put in a given text file. Text can be written whatever in a given text file i.e. it can be classified whatever book summary to predict the genre.

5.2. Data Cleaning

Data cleaning is removing unimportant information from the book summaries for the purpose of easier classification. Data cleaning process involves removing punctuation, removing stop words such as was, am, the and removing low occurrence words.

Figure 6. Book Summaries Dictionary

5.3. Classification

Classification is done by score comparison method and percentage comparison method. Now, training book summaries dictionary and sample book summary dictionary are obtained. Sample book summary is only needed if only one book is tested. If not, dataset can be divided into training summaries and testing summaries. In the following examples, for only one summary will be classified. But the concept is the same for the multiple book summaries.

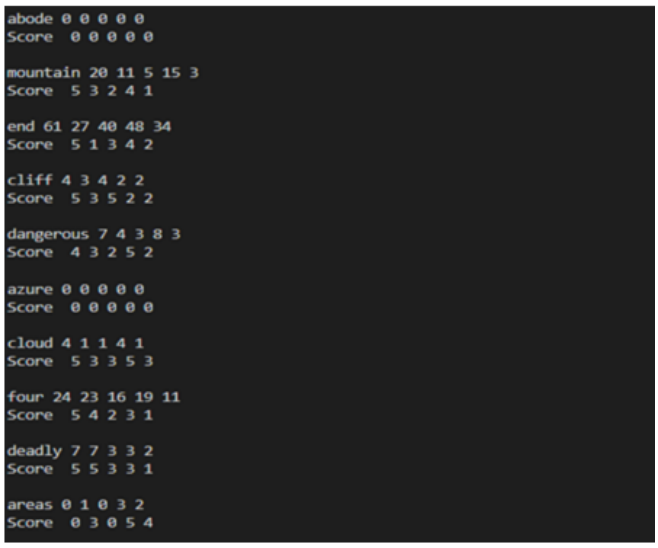

Figure 7. Word Count and Scores for Score Comparison Method

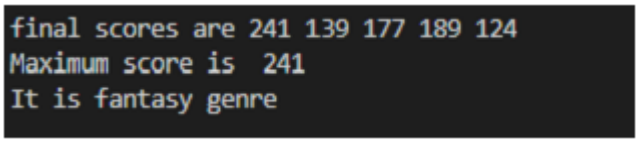

Figure 8. Final Results for Score Comparison Method

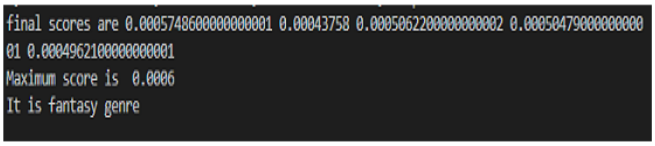

Figure 9. Final Results for Percent Comparison Method

The maximum score 0.0006 is in fantasy genre. So the result of the prediction is fantasy genre.

6. PERFORMANCE ANALYSIS

The overall Automated Book Classification is tested by using Visual Studio Code Software IDE and Python. Python Pandas is used to process the dataset. Visual Studio Code is a proprietary cross-platform source code editor with a Python application programming interface (API). It natively supports many programming languages and markup languages. Python is a great general-purpose programming language on its own, but with the help of a few popular libraries (NumPy, SciPy, matplotlib) it becomes a powerful environment for scientific computing functions can be added by users with plugins.

Genre	Precision	Recall
Fantasy	40.91%	**90%**
Mystery	**100%**	55%
Romance	80%	80%
Sci-fi	84.62%	55%
Drama	91.67%	55%

Figure 10. Precision and Recall Values for 100 Books in Score Comparison Method

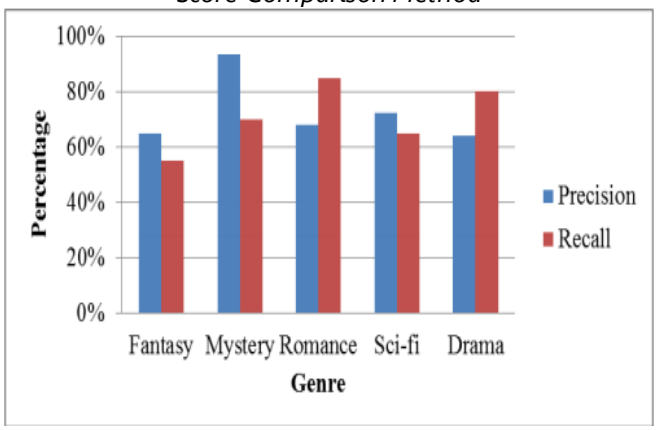

Figure 11. Precision and Recall Values for 100 Books in Percentage Comparison Method

REFERENCES

[1] Lena Hettinger, Martin Becker, Isabella Reger, Fotis Jannidis and Andreas Hotho.: Genre classification on German novels, (2015)

[2] Emily Jordan.: Automated Genre Classification in Literature, (2014).

[3] David Bamman and Noah Smith.: CMU Book Summary Dataset, 2013, URL: http://www.cs.cmu.edu/~dbamman/booksummaries.htl

[4] Yiming Yang and Thorsetn Joachims.: Text categorization, October (2011).

[5] Marina Santini.: automatic genre identification, (2010)

[6] Microsoft Research.: Automated document genre classification workshop: Supporting digital curation, information retrieval, and knowledge extraction, September (2009).

[7] Philipp Petrenz.: Assessing approaches to genre classification, 2009.

[8] Christopher D. Manning, Prabhaka Raghawan, and Hinrich Schutze.: Introduction to Information Retrieval. Cambridge University Press, (2008).
[9] Marina Santini.: Automatic genre identification: Towards a flexible classification scheme, (2007).
[10] Barbara H.: Kwasnik, Kevin Crowston, Michael Nilan, and Dmitri Roussinov.: Identifying document genre to improve web search".