AUTOMATED GENRE CLASSIFICATION IN LITERATURE

by

EMILY JORDAN

B.S., Kansas State University, 2012

A THESIS

submitted in partial fulfillment of the
requirements for the degree

MASTER OF SCIENCE

Department of Computing and Information Sciences
College of Engineering

KANSAS STATE UNIVERSITY
Manhattan, Kansas

2014

Approved by:

Major Professor
William Hsu

Copyright

Abstract

This thesis examines automated genre classification in literature. The approach described uses text based comparison of book summaries to examine if word similarity is a feasible method for identifying genre types. Genres help users form impressions of what form a text will take. Knowing the genre of a literary work provides librarians, information scientists, and other users of a text collection with a summative guide to its form, its possible content, and what its members are about without having to peruse individual topic titles. This makes automatically generating genre labels a potentially useful tool in sorting unmarked text collections or searching the web.

This thesis provides a brief overview of the problems faced by researchers wishing to automate genre classification as well as the current work in the field. My own methodology will also be discussed. I implemented two basic methods for labeling genre. The results collected using them will be covered, as well as future work and improvements to the project that I wish to implement.

Table of Contents

List of Figures

List of Tables

Acknowledgments

I would like to thank my advisor, Professor Hsu. You advice during the creation of this thesis has been amazing. I would like to thank you for encouraging my research, and for all the interesting avenues of research I was introduced to via the Knowledge Discovery in Databases research group. I would also like to thank my committee members, Professor Andresen and Professor Neilsen, for serving as my committee members. I also want to thank you both for your brilliant comments and suggestions during the defense.

A special thanks to my family and friends. Your prayer for me was what sustained me thus far. Thank you for serving as my sanity as I put in the long hours necessary to finish all this work. I would not have made it without you guys.

Chapter 1

Introduction

This thesis presents my study of the classification of books based upon their summaries. My hypothesis is that it is possible to classify books based on the word content of their written summaries. My technical objective is to write a program that will identify common words that belong to common book genres. Once the program has a list of words in relation to their given genres, it will attempt to classify new books into the predefined genres. The program will return scores for the new summaries relations to the predefined genres. One end goal is to enable easier classification of books, and let people know about possible niche genres, or the overlap of genres between books. This might allow books to be easily identified as more than one genre type.

1.1 Problem Definition

A problem that faces libraries today is the diversification of types of literature. Authors enjoy exploring new ideas and concepts in the fictional works that they write. Because of this, there exist a wide variety of subgenres or mixed genres. When an author presents a new book, editors, publishers, and librarians read it and decide upon what labels the book requires for its records. Often times, books fall under more than one heading. For instance,

a book could be a romantic mystery in which a male detective gets together with the woman who hired all the while trying to find her dead husbands killer. Should this book be marked under the primary genre of mystery, or romance? What qualifies the book to belong more to one genre than another? Without a clear metric to decide how much a book belongs to a specific genre, many books end up poorly classified or shoved under the super-genre heading of fiction. This is why it is important to find a way to classify books and their degree of relativity to a given genre.

Another problem to consider is the question of how best to generate labels for texts that have none. As collections of literary works are migrated into digital format, there exists a need for automatic genre label creation. Currently such labels are made mainly using subjective criteria [1]. For this reason, a better process for identifying genres and classifying them needs to be identified. It is for these reasons that I am looking into the field of genre classification in literature.

1.2 What is Genre?

The Merriam-Webster dictionary defines genre as "a category of artistic, musical, or literary composition characterized by a particular style, form, or context [2]." Genres are what is currently used to classify books by type. Genres are the labels people use when describing a book in its basic form. Genre labels give the person reading them an impression of what the object should be, without giving away specifics of the object. However, the definition of these labels are not static. As Chandler notes in his *Introduction to Genre Theory*, "There are no rigid rules of inclusion or exclusion. Genres are not discrete systems consisting of a fixed number of listable items [3]."

For example. one could take a look at Bram Stoker's *Dracula*. Originally published in May of 1897 [4], this iconic book defined what it is to be a vampire. The popular classification for Dracula related stories and movies is horror, due to the themes and bloodthirsty creatures contained within. Because of this, Vampires came to be considered a typical monster of the horror genre. This raises a question. If vampires are traditionally used in horror books, does this mean that all books with vampires should be classified as horror? *Twilight* is another book containing vampires. Does this automatically make *Twilight* a horror book as well? The answer to that question is no. *Twilight* is actually considered a romance book. Vampires might be a topic within the horror genre, but they do not define the genre itself. Genre provides a way to classify types of data into groupings, rather than topics of data [5].

This is why making an all-inclusive list of rules to define genres can become a near impossible task. While generalizations can be made about individual genres, coming up with specifics for classification can be much more difficult. One author might write vampires as terrifying menaces willing to kill all who stand before them, while another author might write vampires as misunderstood beings who just want to live and find love like the rest of humanity.

While genres can be viewed as sets of rules, ultimately they tend to be more sets of opinions. Consider the following sentence, "Their eyes met across a crowded room, and time stood still." What genre might this sentence have come from? Why would this sentence belong to that genre? Intuitively, a reader could assume that the sentence came from a romance book, as it refers to two people seeing each other and time stood still. We know from pop culture that time standing still is often used to refer to romance. Madonna even wrote a romantic song titled, "Time Stood Still" for a romance movie [6].

However, that is just one interpretation of the sentence. An avid science fiction, or sci-fi

for short, fan might read the sentence, and decide that time stood still because the person's time-stopping freeze ray misfired, or because the hero used his superpowers to stop time. A fantasy fan might think that magick was involved. Possibly one of them was a mage, or maybe time stopped because they are about to communicate telepathically. It's all about perspective. As Chandler says, "Particular features which are characteristic of a genre are not unique to it [3]." Just because a feature indicates a genre does not mean that the book is a part of it.

This is what makes genre classification so complicated. Genre definitions can differ based on society, country, and person to person [3]. They also cannot be defined by a single book or sample from their genre. For instance, J.R.R. Tolkien's *Lord of the Rings* is considered one of the best fantasy books ever written. J.K. Rowling's *Harry Potter* series is also considered to be a great fantasy series. One is based in a land of elves, dwarves, men, and hobbits fighting to do what is right. The other is based in a magical school with children casting spells and fighting evil. If you were to build a definition of fantasy using just one of the above, you would miss so much of what the genre could be. Even with a wide selection of examples from a given genre, there is no guarantee that the sample will include all possible definitions.

What might seem to be an intuitive classification for one person, might be considered wrong by another. While most can agree on the basics of a genre definition, such as romance being about love in some form, or mystery working to answer a question, specifics tend to elude people. One person might classify Max Brooks's *Zombie Survival Guide* as a non-fiction training manual for the upcoming zombie apocalypse. Another would classify it as satirical fiction, playing off the common fear of the mythical upcoming zombie apocalypse.

In summary, genres are currently suffering from the problem of not having a clearly defined genre taxonomy. There is no one correct set of rules of identifying a book or text

document as belonging to a specific genre, as the rules for identification change and flow from work to work. This leads to the problem I am attempting to address, that is, the automation of genre detection. If there are no overarching rules for genre identification, how can a computer be taught to identify genre?

1.3 Unclear Genre Illustration

Now that we know about why it is so hard to define a single genre, we shall examine how this can affect the labeling of books. Below are several books from Jim Butcher's *Dresden Files* series. I've listed out three titles by this author, as well as genres and topic tags with which they have been tagged. The tags are ones that have been added to the books **Ma**chine-**R**eadable **C**ataloging records according to WorldCat by various libraries. They contain bibliographic information about a book that can be interpreted by a computer for electronic storage. WorldCat is an online database of MARC records that has MARC records available for librarians and researches to view and download [7].

The books in Table 1.1 all feature the same main character. The main character is a wizard detective, solving supernatural crime. Each of the books have been given multiple genre and topic related tags in order to help classify them. If we examine the above

Storm Front	**Ghost Story**	**Turn Coat**
Wizards – Fiction	Wizards – Fiction	Mystery Fiction
Murder – Fiction	Fiction	Treason – Fiction
Magic – Fiction	Fantasy	Wizards – Fiction
Science Fiction	Ghost Stories	Detective & Mystery Stories
Paranormal Fiction		Fantasy – Fiction
Private Investigators		
Detective & Mystery Stories		

Table 1.1: *Table of genre classifications & labels given to works by Jim Butcher [7].*

information, we can gather several facts from it. The first is that the books have a common theme of being fictional, and featuring wizards. Several of the tags mention crime, detectives, or investigation. Any of those would indicate mystery as a possible theme. So, from just the given tags we know the books could be classified as fantasy, mystery, or fiction.

This raises the question of which genre should be the primary one. What genre do the books most heavily identify with? Currently, each book has been given tags that could fall into multiple genres. For mystery, we have private investigators, detective and mystery stories, and murder. For fantasy we have wizards, magic, and paranormal. Then we have a single science fiction tag. As many of the subheadings are fiction, the book could be considered general fiction as well. From these labels alone, one could surmise that the book should be placed in either fantasy, mystery, or science fiction. If one were to consider the general fiction labels, then there are quite a few votes to just classify the book as general fiction and forget about it. Classifying as general fiction is the band aid solution to this problem. Oftentimes when books have unclear classification, they are pushed under the penumbra heading of just fiction. While this is a true heading as the books are in fact fiction, it doesn't help readers who are looking for specific types of books.

Based on the above counts, where should the library potentially shelve the book? What if a librarian doesn't realize the books are a series and so splits them into multiple sections of the library based upon the MARC records? How will patrons of the library find the books they want if the librarians themselves can't figure out where the books should go? It can be difficult to push books into a single genre niche. The existence of multiple topic and genre headings on books MARC records supports this assertion. This fact also implies that books often exist across genre lines. Genres are becoming increasingly crossed as authors try new things and challenge conventional storytelling. Because of this, and because of the inherent difficulty of classifying genres to begin with, it can be difficult to make decisions on exactly

what the super-genre of a given book should be.

This is the primary motivation for the work in this thesis. I want to make an automated program that can help identify the super-genres of books and the similarity they might have to existing genres. I am hoping that my program will help identify genre related tags that should be given to books so that patrons who are looking for cross genre books such as "romantic mysteries" or "fantasy westerns" will be able to find what they are looking for when they do a search. My system will also help identity to what degree the book identifies with a given genre. As each of the five possible genres I have defined will be given a score, it will be possible to see how strongly a given book identifies with each of the genre headings.

1.4 How this relates to libraries

As I mentioned above, most libraries try to split up their fictional books into subtypes for easier patron browsing. Patrons have preferences for what kind of books they enjoy reading. To cater to these preferences, libraries try to arrange their shelving by genre, so that similar books are grouped together. For example, they might shelve all the biography and autobiography books together. Or they might have a children's section where books for younger readers are put into colorful displays.

Librarians also try to split up the fiction section so that readers can find what they are looking for. Readers often don't know what it is they are looking for when they come into the library. A study done in 2013 over a period of 12 months surveyed patrons to ask their reasons for visiting the library. Patrons were asked to fill out a brief survey about the reasons for their visit and the activities they engaged in during their visit. It was found that 73% came in to just browse the collection to look for something to read [8]. The write-up on the

study mentioned the following in regards to browsing, "Many of our focus group members mentioned how they enjoyed browsing the shelves at their local public library. One liked the process of discovery: "The cover can draw you in."" [8] Patrons often don't know what book they wish to check out when they enter the library. They just know that they want to find something good to read. Patrons often have an idea of the types of genre they like, so they start their search near the familiar. If a patron enjoys mystery books, then they will look for other mystery books. If a patron enjoys factual books on the history of the civil war, they will probably look to see if they can find more books of the same genre on that topic.

These browsing practices illustrate the importance of selecting an appropriate tag for the books. If a patron is looking for a mystery book, they will start browsing in the mystery section. They might never find an excellent supernatural mystery book shelved over in the fantasy section simply because it is labeled 'fantasy'.

If my tool can help identify how strongly a book relates to a given genre given the summary from the back, it could help librarian's make decisions on where the book should be shelved. It could also help them build displays or special exhibits. For instance, they library could decide they want to do a display of "romantic westerns" to encourage some of their romance readers to look into the western genre. They could use my tool to identify books with strong romance and western themes in their summary, and gather those books to put in their display. If nothing else, it might give librarians a starting point with which to aim patrons who have, "read everything in their favorite genre" and want to find something new.

Chapter 2

Related Work

2.1 Santini's Approach to Genre Classification

The work of Santini [9] has centered around automatic genre identification [9]. Santini is interested in this field because of its applications in grouping unknown web pages together. Santini points out that it is quite easy to acquire large collections of text data from the web that is essentially unlabeled. Tools need to be created that can take this unlabeled data and split it into collections. One such was of arranging the data that could be of use is to sort by genre. Santini's current research goal is to, create evaluation resources for genre and other non-topical descriptors [9]. This way, when a person acquires a large volume of text such as the kind you find on the internet, it can be grouped according to genre as opposed to being just by topic.

One of Santini's earlier papers concerns what kind of system needs to be developed in order to classify genre on the web. In it, she points out the two questions that are hardest to answer when it comes to building a genre classification tool. They are, in summary, "What is genre?" and "What genre classes are there?" [1]. As discussed earlier, genre is a rather hard subject to actually pin down. In her paper *Automatic Genre Identification:*

Towards a Flexible Classification Scheme she discusses exactly how the definition of these terms can impact the data collected. As she points out, currently most of the collections used for genre experimentation are, "small and mostly built with subjective criteria [1]." This is a problem, as the experiments are getting locked in to what a single person thinks genres should be defined as. As I mentioned in the earlier section on genre theory, most work in that field is still concentrating on deciding how genres should be defined [3]. For this reason, there isn't much that can be done about this problem yet, aside from using the best common sense approaches available and the most current classifications available. Santini herself admits that she doesn't have a solution for this problem yet [1], but wants the reader to be aware that the problem exists.

Santini has done work on establishing a usable genre taxonomy [10], as well as work on figuring how to evaluate the results of the data collected. Santini considers this an important task as, "automation of metadata extraction is crucial to digital curation activities. [10]" As our world becomes increasingly saturated in textual data, the need for automation on classifying things such as genre and topics becomes a more pressing need. Her paper on flexible classification schemes focuses on the classification of web genres, but it points out several interesting ideas.

The first idea from her cross classification paper that I find noteworthy is her idea of needing a flexible genre classification scheme [1]. She believes that if a program is to be written to classify genres on the web, then it needs to be able to handle multiple genres occurring in a single document. The web is not an organized place, so the text populating it is often of mixed types. Santini suggests that pages should be assigned zero, one, or multi genres [1]. This way, the page will be correctly identified, no matter how many elements it pulls upon. It avoids the problems of pigeon-holing a page into a single genre can cause, especially in the case that the deciding factors were a close tie. As Chandler mentioned

in his works, it can be difficult to find the edges of a given genre, as they tend to blur and intertwine together [3]. Close scores could indicate that the page is blurring the lines between genres, and so might need multiple classifiers to be truly defined.

Being able to classify text into their basic genres would allow for better collection profiling [10]. Collection profiling is where known information about a collection is used to infer other possible characteristics or tendencies of the collection. For instance, if we knew a collection was made of 'mystery' books, we could assume that all books will be concerned with answering a question of some kind. In the book *Genres on the Web*, Santini *et al.* say the following, "...knowing the genre to which a text belongs leads to predictions concerning form, function and context of communication [11]." This means that a person forms ideas of what might be inside of a text just by knowing the genre. A researcher could form ideas on the content of a large text collection, and the format it would possibly be presented in, without having to actually peruse individual texts. The more information that can be extracted and labeled automatically from text collections, the easier it is for humans to label the rest.

Santini is approaching the problem using natural language processing and machine learning. She uses a variety of "facets" to identify possible genres [1]. For instance, if a text contained many first person pronouns, her program could make assumptions that the writing might be comments or opinions as those both talk from the first person view [1]. Santini came up with a hundred facets for her flexible classification scheme [1]. Many of them are natural language based. 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.

2.2 Text Classification

Text classification is a topic in information retrieval in which while searching for information the documents or text are placed into classes [12]. 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 [13]. Text classification is used for a wide variety of tasks. It can be used for sentiment analysis, email sorting, topic classification, spam pages, web filtering, and a variety of other jobs [14]. Due to its wide variety of usage, this field is a rather popular on in information retrieval. Text classification includes techniques such as Naive Bayes, Tf-idf, Latent semantic indexing, support vector machines, decision trees, and natural language processing [12]. While this list is by no means exhaustive, it does give an idea of how large of an area of study text classification covers.

All of the above methods can be used for classifying text into groupings using comparisons of some form or another. As one can surmise, these methods could be of use for the classification of text into genre. The problem is figuring out which ones would perform the best for the given task at hand.

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 [15]. Petrenz examined how well the four algorithms performed when tested on newspapers that were not the one they were trained with. His goal was to evaluate if style of the newspaper plays a part in how well the algorithms could classify the articles contained therein [15].

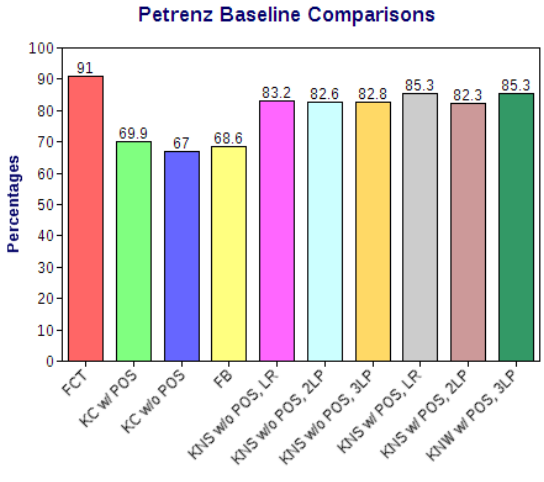

Figure 2.1: *Baseline comparison of different genre classification results as reported by Petrenz[15].*

Figure 2.1 shows the baseline comparison between the different methods. POS stands for parts of speech elements. He ran each test method with several different elements enabled so as to show how the different classifiers can improve or decrease the amount of correct labels generated. Petrenz found that the bag-of-words based support vector machine approach, labeled FCT on the graph, seemed to work the best when compared to the other methods examined for identifying genres when faced with new styles [15]. This success on classifying web genres using bag-of-words gives me hope that my own classification experiments on book summaries might yield good results.

The other methods handled comparatively, with some doing considerably worse than others. Petrenz speculates whether or not structural cues may have played a part in the outcome of the experiment [15]. His paper illustrated how different text classification methods can all yield results when put to the task of identifying genre, some better than others. He suggests that additional comparisons should be made though, as he only managed to test four methods in a total of ten variations out of the multitude currently available. 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. If progress is to be made, then baselines need to be established for testing in order to compare different work in a more efficient manner.

2.3 Genre Theory

Genre theory is the field that is attempting to define and understand what genre is [3]. Unfortunately, as they are still working on what genre is, there hasn't been as much work as there could be into the hows of classifying them automatically [15]. 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 [15, 9, 1, 3]. This is because that genre is a multifaceted idea to begin with, which makes a singular identifier insufficient for assessing the whole of the concept. As genres encompass all that a text is, it takes more than one classifier to accurately identify the genre.

In his paper on genre theory, Chandler suggests a list of approximately fifty questions for his students to answer when attempting to analyze a text in relation to genre [3]. They include questions on the feel of the document, how it is organized, the conventions used, how it could be interpreted, and what realities it reflects amongst many other things [3]. His list gives a sampling of what facets of the text have to be examined in order to truly classify the genre. The question then arises of how one could teach a computer to understand all of these concepts. One of the questions involve what feelings the text generates, another asks what knowledge the text takes for granted [3]. While the field of sentiment analysis has come quite a ways, computers still have problems identifying feelings in text [16]. What is the difference between sarcasm and literal speaking to a computer? Also, how can a computer

identify what knowledge isn't mentioned in the paper due to assumptions that the reader already knows the information? The need to answer many of these high-end questions in order to perform genre classification causes problems for actually doing so in a programming setting. Many of them are fields of ongoing research in information retrieval, and others are problems being considered in text classification. This means that genre theory will potentially require a multi-faceted cross-discipline approach if all aspects of its classification are to be considered.

Genre theory is currently mostly concerned with coming up with an exact definition for what each genre is. The researchers in this field are working to find methods that can be applied to all genres for the creation of their definitions, as well as trying to create universal definitions for all genres already in existence. These definitions are important for work in the field of genre classification because without a standard definition, how can one hope to teach a computer what genre is? Computers tend to be rather literal in their rulings. While there are more smart classification systems out there, this lack of formal definition of genre can make using them difficult.

Much of the current work on genre theory in the classification area has been done in the area of web genre. This makes sense, as much of the field of information retrieval and text classification is interested in identifying unknown webpages and other large data collections from the internets. I shall give a brief overview of the methods being used by authors whose work I have referenced in this thesis.

Santini covers using natural language processing for information retrieval in her book titled *Genres on the Web* [11]. Her book provides an overview of common techniques used for identifying genres on the web. She speaks about the relationships between text and language in ways that even beginners to the field can understand. Santini points out that

web genre has crossover with many areas of text related research, such as natural language processing, computational linguistics, web mining, social network analysis, and more [11].

Crowston *et al.* used a mixed heuristic and machine learning approach, as the two methods compliment each other [5]. They plan on running the manual and automatic methods iteratively in order to build on their own results while identifying web genres [5]. Heuristics allow for producing a reasonable result in a relatively short period of time, which is good when working with large corpus of text such as the web [17]. When coupled with a secondary technique for optimization, such as the machine learning of Crowston, their efficiency improves [17].

The above are just two examples of researchers looking into web genre classification. Santini [9] and Crowston [5] 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. I will remind the reader that Chandler points out that due to how entwined genres become in literature, it is hard to find clearly defined edges between them [3]. 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 [18]. This is why the work being done in this field encapsulates such a wide variety of techniques from the field of text classification. Since nobody is certain what will work best, all methods are being tried in the attempt to find the one that works best.

Chapter 3

Methodology

3.1 Motivation and Goals

My motivation for studying this area came in part due to my own experiences of working at the Port Library in Beloit, KS. While working there, I was involved with the digitalization of the library's collection of books. Each book had to be scanned into the system, and then have MARC records downloaded from *WorldCat* [7], or entered by hand when a record didn't exist for them online. It was interesting to see how our classification of books sometimes differed from that which *WorldCat* recommended. It was also challenging to come up with genre and topic labels for books that had no MARC records. How should one go about classifying the book? When working with several hundred unclassified books, it would be impractical for each unknown book to be read in full in order to gain a comprehensive overview of the text contained therein. This led to many books that had no MARC records gaining their classifiers entirely from their summaries. This led me to wonder if a program could be created that would generate such labels for the librarian, in order to speed up the classification process. If a librarian can classify a book into a genre just by reading the summary, could a computer be trained to do so as well? To answer that question, I decided to conduct this research.

My goal with this research is to see if there are identifiable word amount patterns in the summaries of books so as to better tag them for genre classification. My belief is that book genres have common words in them that can be identified, hence a bag-of-words approach might have discernable success in identifying books possibly related to the genre in question. As it is possible for a human to get an idea of the genre of a book just by reading a summary, I want to see if a computer can do the same. My goal is to automate the classification process. The job of a book summary is to describe the contents of the book in short form. As such, authors might use words related to the genre they are writing within so as to lure in potential readers. As I mentioned in 1.4, it has been found that patrons of the library system tend to look for books related to their genres preferences. Authors know of this correlation, so try to make their books sound unique and interesting while still identifiable as a given genre.

Most previous work that I have looked at have viewed classification from the approach of the entire text. Researchers attempt to classify a work as a particular genre by identifying style, topics, or form of the given work [15]. They might examine parts of speech, or run Naive Bayes classifiers upon the entire document system. My research takes a more narrow approach. When classifying books, most can be grouped based on just their summary without needing to read the entire work. As summaries are meant to give an idea of the content of the book it could be interesting to see if genres could be identified or patterns discovered.

The method I will be using is sometimes referred to as bag of words approach. Bag of words takes all the words in a text and stores them in a collection together. It is creating a 'bag' full of words from the text, with their counts with them. This allows for simple keyword comparisons. I decided to keep the analysis simple for the first iteration so as to gather information on if this approach to genre classification has merit. These simple methods will provide me with a baseline program that I can use to establish possible genre, without hav-

ing to delve too deeply into natural language processing, machine learning, or indexing. If this initial method shows results, then I can begin applying more in depth analysis to the topic to see what kind of results might be achieved. All of the papers I looked at emphasized the need for looking at the text using more than one classifier, as genre is a complicated topic. If my method shows results, it could be the first iteration rough estimator from which a more in depth analysis could check itself against for comparison.

A good point of the bag of words approach is that a score is calculated for all genres involved. If I were to expand my work to include more genres, this could be seen as a downside due once again to memory requirements, but for this small dataset it allows for comparison of the degree to which the bag of words identified the genre or misidentified it to be analyzed. This allows for my test data set to consist entirely of cross genre works, that is books with two main classifiers for genre instead of one. It will be interesting to see if my tool manages to correctly identify both genres as being the primary genre tags for the given book. As Santini said in her paper on flexible classification schemes, it is becoming less common for works to belong to just a singular genre [1].

Consider the following summary from J.D. Robb's *Naked in Death.*

> *When a senator's daughter is killed, the secret life of prostitution she'd been leading is revealed. The high-profile case takes Lieutenant Eve Dallas into the rarefied circles of Washington politics and society. Further complicating matters is Eve's growing attraction to Roarke, who is one of the wealthiest and most influential men on the planet, devilishly handsome... and the leading suspect in the investigation. [19]*

The above is the summary from the back of a novel marked mystery according to *World-Cat* [7]. A human reading the summary can intuitively identify that the novel belongs to

both mystery and genre. I'm hoping that my tool will identify words such as "attraction" and "handsome" and tag the book as a possible romance as well as identifying the mystery genre from the words such as "killed", "case", and "suspect".

A weak point of this method is that it is essentially building topic lists, with the relation of the topic determined by the total word counts the topic occurred. Topics are not genres, but they can give a basic idea of where the book might belong to. I spoke on this in Section 1.2. Another weak point is that as the size of the training data set increases, the slower the actual classification program will perform. The training data set could quickly become unwieldy to work with as there are a large number of words in the human language. This could affect speed and performance, as the word table gets bogged down with topic words that may or may not be of use.

To do so, I needed to acquire training data and then pull word count information from the summaries in order to run comparisons between my word dictionary and the summaries on the backs of the books. I decided to use two different kinds of analysis to classify the genres. They are a points-based allocation ("points buy") system and a percentage comparison method.

3.2 Training Data

3.2.1 Acquisition

The problem with any training data, is as was mentioned in Section 2.1 on Santini. My data set that I am using is "small and mostly built with subjective criteria [1]." To attempt to combat the subjectiveness of my own choices, I used book lists that have been approved by multiple people as shining examples of their genre. For instance, I chose the mystery

books training data from NPR's *Audience Picks Top Mysteries, Thrillers, and Crime* [20]. This way I could be certain that the book summaries chosen for my training set are ones that have received thousands of reviews stating that people believe them to be of a specific genre. I wanted to be certain that I would get a wide sampling of books from each genre that critics and readers agree are examples of that genre.

Once I had a list of books, I went to Amazon.com and downloaded summaries into my data sheets. I then repeated this process for each of the five genres I was examining. For fantasy, I pulled books from NPR's list of top fantasy and sci-fi novels [21], as well as a selection from *Fantasy 100*. Fantasy 100 does polls and surveys to identify what are currently considered the top hundred fantasy book series [22]. Another site that was employed was GoodReads. It is a site that allows users to vote on genre tags for books, as well as classify books into lists [23]. I used some of their top results for the romance and western genres to help fill in my data lists where NPR didn't provide quite enough books for a hundred entries per genre. Despite these efforts, my data is still partially subjective in nature. While selecting summaries, I tended to choose ones that I felt were relevant. I discarded more than a few summaries that were either too short, or that I thought weren't descriptive enough for various reasons. Some summaries were basically singing accolades for the books author instead of describing the book itself for instance. Others were only a sentence or so in length. As I was going for what is a typical summary from the back of the book, I tended to discard these in preference of single paragraph summaries of the type usually seen on book backs.

In total, I collected 500 records for the training data set. I am attempting to identify five different genres, those genres being *fantasy, mystery, romance, sci-fi,* and *western.* Each genre had a hundred records collected for training upon.

3.2.2 Data Cleaning

Before the data can be sorted, it first needs to be cleaned. There were several stages to this process. The first was removing all stop words from the data. Next, a method for removing common names was employed. After that, low occurrence words were removed to help improve run time and trim the training data.

Stoplists were probably the simplest of the data cleaning concerns to be addressed.I used several well-defined stoplists that could be employed to clean the data [24]. Stopwords or stoplists are lists of common words in the English language. As common words such as "was", "am", or "the" tend to be rather neutral in meaning, I wished to remove them from the data count. I examined the stoplists the *A Norm AI* website had to offer, and chose one that I believed to be most inclusive for the terms I was interested in eliminating. In my program, the stop words are eliminated before the word counts are run. When a file is first read in, the lines are split up and processed into Genre data objects. The object contains the uncleaned abstract and the known genre label. Once it has this data, the genre object then processes the abstract it was given and removes all words from it that exist in the stopwords list.

Next, I needed to come up with a way to handle words that had a low occurrence in the text. For instance, a summary might mention the city the book supposedly takes place in. This is the only mention of the word anywhere. The word isn't relevant to defining the genre as it only occurred once. The city in my example could even have been entirely made up, which makes it even less relevant for possibly defining a genre. Another example of this low occurrence problem often occurs in fantasy novels. Fantasy novels often make up enemies such as "stormwings" [25] for their heroes to fight. The creature only occurs in books written by that author, so does not help in building an overall definition for the genre. For this reason, my program needs a way to remove these edge case words that

do not occur often enough to actually have anything to do with defining the given genre. I set the threshold of what was to be kept to k = 3. If the word occurs more than k times across all genres, the word will be kept. Otherwise, the word will be dropped from the database. This will hopefully make the data more compact with less bloating from edge case words. As the database gets larger, having a k value set higher might help with the speed problems that will inevitably occur from using a topical word list for identification.

Named Entity Recognition (NER) is a natural language processing problem that I also needed to find a way to address. NER is a process that identifies named entities such as person names, city names, and other proper nouns. Like the low-occurrence word problem mentioned in the previous paragraph, names also do not add to a definition of a genre. Summaries often have the names of characters from the novel in them. Obviously I don't want named entities to count towards word count totals. For instance, one of my training data summaries is from the *Harry Potter* series. The summary of the book mentions young Harry several times. Without a way to identify named entities, Harry will be counted and listed as a word for the fantasy genre. Harry is not however a defining word of fantasy. It is a name. I need to consider how to handle these names as they appear in text. There are several tools that could be used for this. However, that would require a much higher level of processing on my text than I think is completely necessary for this initial exploration. The focus of my research is not in writing named entity recognition software but in text based comparison on summaries.

To combat the fact that names appear in text, I decided to perhaps deal with names at least partially like I deal with stopwords. To this effect, I began looking into lists of popular names. It quickly became apparent that most of these lists of popular baby names were based off data from the social security website [26]. This makes sense, as social security receives information on all births and deaths in the United States. Some browsing got me

to a page where I could look at the top ten names by year in the United States. Since I wanted a slightly larger list, I expanded my search and managed to find a list of the top 100 names for male and female children from the last century. That is a list of the top 100 male names and, and a second list of the top 100 female names which gives meo 200 names I could eliminate from my data while cleaning. My program removes these names at the same time it removes the stop words. This way both common names and stopwords will get removed from the word count so as to not skew results with extra words that have no real meaning to my survey of data.

3.3 Software and Tools

The programming for this project was done in Java within the *Eclipse* integrated development environment (IDE). For calculating word counts in the training data across the files used, Apache Hadoop MapReduce was employed. The MapReduce jobs were run within the Eclipse environment, using the Eclipse Map/Reduce debugger module. The idea of employing MapReduce was so that if I were to acquire larger test sets for future work, the processing power of Hadoop could be employed to run through the larger dataset in an efficient manner.

Chapter 4

Score Comparison Method

4.1 Method Overview

The score comparison method system classifies the book summaries by giving them a total points score based on which words occur in the books summary. 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. The theory behind this classification is that words that occurred more frequently in a given genre should be given more points as they belong more strongly to that genre.

The method is as follows. The initial stage counts all the words in the summaries. The words are stored in the following form. The counts are in order of genre, that is [fantasy, mystery, romance, sci-fi, western].

```
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. Let's consider the following example.

```
case [10, 30, 5, 15, 0]
closed [15, 5, 20, 25, 25]
```

We have the phrase "case closed" that the program needs to rank. Looking at the information for the above example we can deduce several pieces of information. First, we can figure out the genre ranking for the word "case". That order is mystery, sci-fi, fantasy, romance, and finally western. The genres are ranked from five to one in order, though if a genre does not have the word at all, it is given zero points. In Figure 4.1 one can observe the points that are available for the word "case".

Figure 4.1: *Initial awarded points for word "case".*

Next we can examine the word 'closed'. The order ranking for closed can be seen in Figure 4.2. From looking at the word counts that the word 'closed' had initially, one could observe that sci-fi and western 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 4.2: *Second stage points awarded for word "case".*

Now that the initial points have been assigned to each word, we can then see how they would add up in the phrase 'case closed'. We can see the results of this combination in figure 4.3. When the points are combined to see how the phrase would rank, the phrase

is classified most strongly as sci-fi, with mystery in second place. If the method were to examine a summary that said, 'case case closed', the end point scoring would return the same for this method. The score comparison method doesn't award bonus points for the word occurring more than once, it only awards points for the word having occurred.

Figure 4.3: *Final points awarded for word "case".*

4.2 Problems Encountered

Originally, the genre set sizes in my training data were not identical. When I copied the summaries from Amazon, I wasn't concerned with how long they were. Some summaries were short, while others were several paragraphs. While I had noticed that the summaries for western tended to be rather succinct, I had not realize this would become a problem for the actual testing. When I ran my initial sanity check, I noticed that the score comparison system was rather biased in its scoring towards certain genres. I decided to look into this problem before continuing with my work. I ran a word counter on each genre to get an idea of how much testing data I really had for each genre. The results as seen in Figure 4.4 made me grimace.

Fantasy and sci-fi had over 5000 more words in their training data than the western genre. This means that summaries could be classified as those genres by virtue of those genres having a larger dictionary of words rather than any sort of relevance on their parts. For instance, fantasy might manage to be ranked first for usage of the word Texas just because it had more chances to use it as opposed to the western genre. I accumulated the results from my sanity test into Figure 4.5 in order to be certain of this bias.

Total Words by Genre	
Fantasy	18858
Mystery	17330
Romance	16241
SciFi	18666
Western	13467

Figure 4.4: *Original word counts across all genres.*

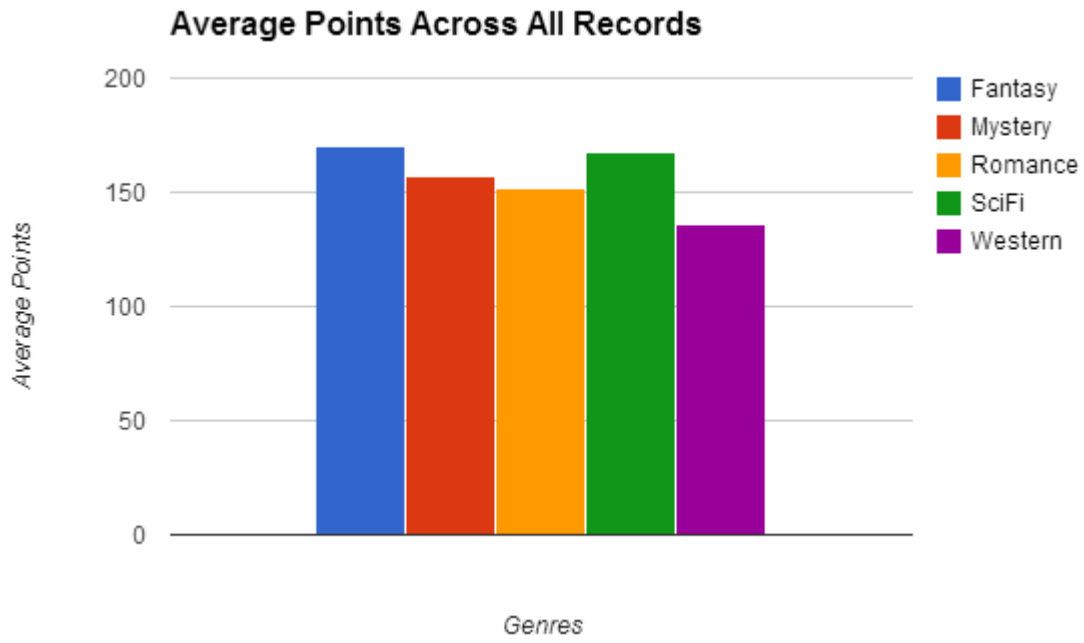

Figure 4.5: *Sanity check with biased points distribution.*

As can be seen the initial results graph, the average points were distributed based entirely on which genre had the most words. The bars are arranged in alphabetical order by genre. The order goes: *fantasy, sci-fi, mystery, romance,* and *western.* The points shown are the average points awarded to each genre while classifying it with the score method. This mirrors the word counts in Figure 4.4. This meant that a word might gain a higher ranking in a given genre not because of relevance, but rather because it had more chances to use the word.

I took two steps to combat this problem. First, I found the shortest of my western summaries and replaced them with longer ones. I chose books written by the same authors, just ones which had wordier summaries available. Second, I took the summaries I had collected for the genres that were not western and shortened their longer entries. Usually this meant deleting the last few sentences from several summaries in each genre, until the word counts were within 1000 of each other. As many words are removed during the stopwords cleaning stage, I felt it wasn't necessary to get closer than that, as many words are lost or removed during the stopword removal stage so the initial word counts aren't exact estimates so much as very good rough ones.

Once this had been done, I had much more promising looking results from the average scoring of the sanity check data. The average scores of each genre across all entries managed to fall within 4 points of each other, indicating that no particular genre had managed to get ahead by virtue of overall word counts.

Chapter 5

Percent Comparison Method

5.1 Method Overview

The percentage comparison method takes into account the size of the word collection for each genre. This method was implemented to hopefully account for the bias different quantities of training data for individual genres might incur. 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.

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.

An example execution can be viewed below.

Example execution:

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. Again, genres are listed in order of fantasy, mystery, romance, sci-fi, and western.

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 these are calculated, the match comparison is run. Each occurrence percentage is multiplied against the occurrence of that word in the given genre. Let's examine the entry for "cat". In fantasy, "cat" occurred 5% of the time. In our sample sentence, "cat" occurred 25% of the time. So, we take 25% * 5% to get the final score for the word "cat" for the fantasy genre. 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, western still managed to become the leader as 'horse' was more relevant to both the western genre and to the sample sentence. The percentage comparison method gives weight to the words occurrence in the summary and in the genre definition, hopefully improving the overall precision and recall.

Chapter 6

Experimental Evaluation and Results

6.1 Testing Hypothesis

My hypothesis was that as the amount of data used for training increases, the amount of correctly identified genres will increase as well. I further predicted that the percentage comparison method will outperform the score comparison method as it accounts for the overall amount of words in each genre.

6.2 Experiment Design

The tests were organized as follows. The first was a sanity check. I then ran a series of comparisons where I used increasing amounts of training data in order to discover if the amount of training data used would make an impact on the correctly identified results. Below I shall outline in more detail what each test entailed.

First was the sanity check comparison run on the training data run by both the score comparison method and percent comparison method. The sanity check was done in order to ascertain if the method could correctly classify its own training data. That is, if I were

to have the method use all of the training data, would it correctly classify the training data set? For this exam, all training data was used to create the word database. The score comparison and percent comparison methods were then run on the training data set. If it couldn't correctly classify the data that it had trained upon, then I would know that my methods were flawed and needed improvement.

Next, comparisons were run using differing amounts of training data for training. Tests were run using 25%, 50%, and 75% of the training data to build the word lists the comparison methods were going to use. The training data not used was then stored to be tested upon to see how well each comparison system would identify unknown single genre books. The word list generated was also applied to the dual genre test data set to see how well it could identify cross-genre books with varying amounts of training data.

The training data was made of 500 records, 100 of each genre as defined in Section 3.2.1. The splits of the training data were done by genre. So, in the case of the 25/75 split, 25 records of each genre type were used for training, and the remaining 75 records were used for doing the single genre tests upon. The same system was used for each of the following splits.

For each of the data splits, one hundred iterations were run. Each run randomly selected training data records to use for training. This way it theoretically tested on a hundred different selections of 25% of the training data, 50% of the training data, and finally 75% of the training data. There was no reason to run multiple iterations for selection on the 100% grouping, as it would have randomly selected every record for testing every time. That is also why there will be no standard deviation calculations on any of the 100% data tests, as there was only a single entry for each.

As for how the methods validate the correctness of the classification outputs, the fol-

lowing methods are used. At the end of any test run, each summary examined has been given five scores. The single genre identification examines the top rated score, and if it was not the one listed as being correct for that book it is marked as incorrect. The dual genre method examines the top two results. Each of the two possible genres were compared against the listed answers. If the calculated answer was 'mystery romance' and the listed result 'mystery sci-fi', then it is marked as '1 0' for correct on mystery, incorrect on sci-fi..

6.3 Baseline Comparison

The baseline comparison evaluates how well each data split performed overall for both the percent and score comparison methods. It is comparing how many genres were correctly identified overall using each method. The results are divided by which method was used, as well as which test data set they were working upon. Thus, the single genre identification results are separate from the dual genre identification results.

6.3.1 Score Baseline

The baseline results of the score comparison method can be seen in Figure 6.1 and Figure 6.2. Figure 6.1 displays how well the score comparison method identified single genre books from their summaries, and Figure 6.2 shows how well it did on the dual genre identification.

In Figure 6.1 one can see the results of the single genre identification for percentage correct. The far right column of the figure represents the sanity test results for the score method. The sanity check for the score comparison method correctly identified 97% of its own training data. I consider this to be acceptable performance for self identification. The training data almost labeled the entirety of itself the way it should be, with only a few cases of misidentification.

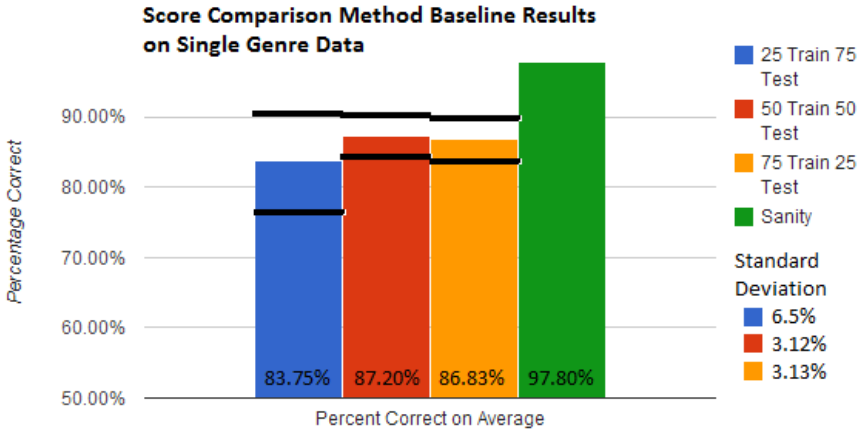

Figure 6.1: *The baseline comparisons of what percent of genres were correctly identified while testing on the single genre data with the score comparison method.*

As for the data splits on the single genre identification, an increasing trend can be observed as the amount of training data used for training increased. There is a noticeable increasing trend as the number of records used for training increased. The use of 50% or the records of each genre and 75% of each genre comparisons performed within a percentage point of each other, with the 75% use of training data performing slightly less well than the 50%. Standard deviation confirms that in this random run of 100 splits of the data into groupings that the 50% train outperformed the 75% split.

The split that used only 25% of the data for training performed the worst out of the three, but not by as much as I had thought it would. I had believed that the 25% test group would get maybe half the number of correct results when compared to the other groupings. Instead, it performed comparably to the others. Its standard deviation puts it within bounds of having performed as well the tests that used more data for training on occasion.

In Figure 6.2 we can view how well the score comparison method fared for identifying the dual genre test data set. When identifying two genres for each summary, the score

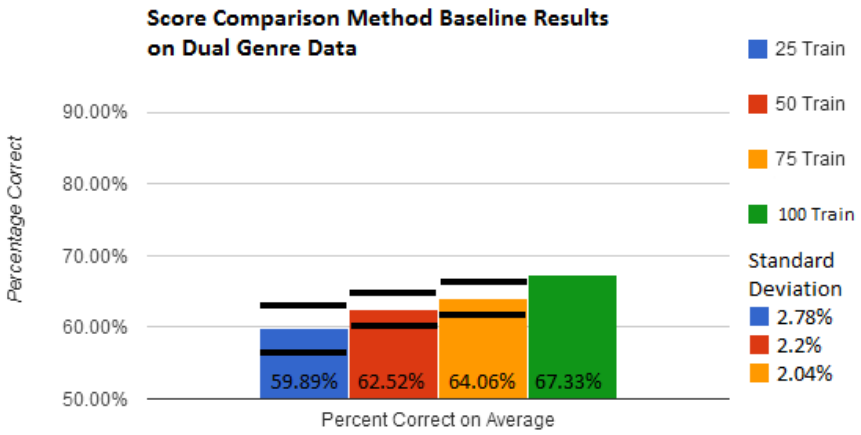

Figure 6.2: *The baseline comparisons of what percent of genres were correctly identified while testing on the dual genre data with the score comparison method.*

comparison method did not perform as well as when it had only a single genre to identify. I wanted to examine how often it found one, both, or none of the dual genres so tabulated those results in Table 6.1. I found it interesting that while only 67% of the total entries for the test data were correctly identified, 95% of the data had at least one of its two genres correctly found. 95% of the results having at least one correct entry in them is pretty good for an initial dual genre identification test. It shows that this method is at least plausible for identifying genre from summaries using topics lists.

% Both Incorrect	% One Correct	% Both Correct	% At least one correct
4.95%	55.45%	39.60%	95.05%

Table 6.1: *Comparison of how many incorrect, partially correct, and completely correct results were found in dual genre identification using the score comparison method.*

6.3.2 Percent Baseline

In Figure 6.3 we can examine the baseline comparison for how well the percent comparison method identified single genres, and Figure 6.3 illustrates how well this method handled

dual genre identification.

By examining the far right column in Figure 6.3 one can observe the results of the sanity test. When trying to label the training data set after having trained using the entirety of said set, the percent comparison method missed almost 8% of the data that it should have identified. This was a worrying initial result, as if the training set cannot identify itself then what hope does it have for the rest of the tests?

The rest of the results for single genre identification were closely bunched. The amount of training data that performed the best was the 50% training data usage one. It beat out the 75% training data method by almost 1%. As with the score comparison method, the standard deviation of using 50% and 75% training data put them within comparable bounds of each other. Using only 25% of the data for training performed the worst, and it also had the largest standard deviation of the training data splits.

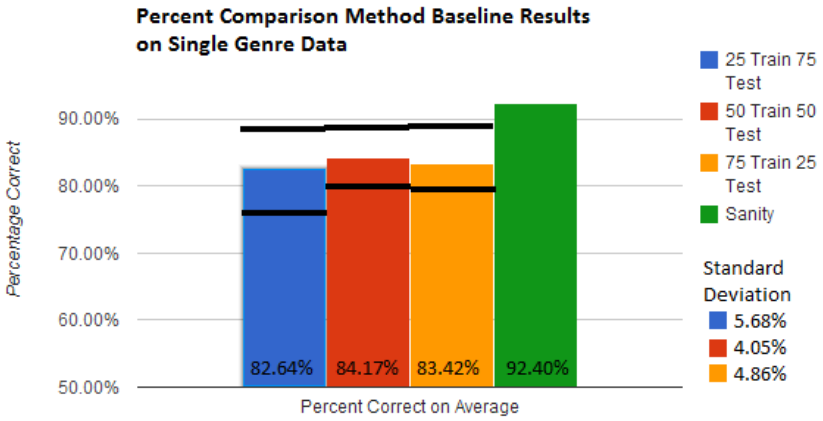

Figure 6.3: *The baseline comparisons of what percent of genres were correctly identified while testing on the single genre test data with the percent comparison method.*

In Figure 6.4 we can examine how well the percent comparison method did at identifying dual genres. As with my hypothesis, the amount it correctly identified rose as the amount

of training data it was given increased. Overall, it managed to identify less than 70% of all the genres listed. However, I once again examined the split of how often it identified both results as incorrect, both as correct, one as correct, and at least one correct. Those results are tabulated in Table 6.2. They showed that 97% of the time, the percent comparison method found one of the two genres the book should have been tagged with. The amount that the percent comparison managed to find both was, however, under 50%. I had hoped that it would be higher, but am still overall pleased at its ability to find at least one of the two genres.

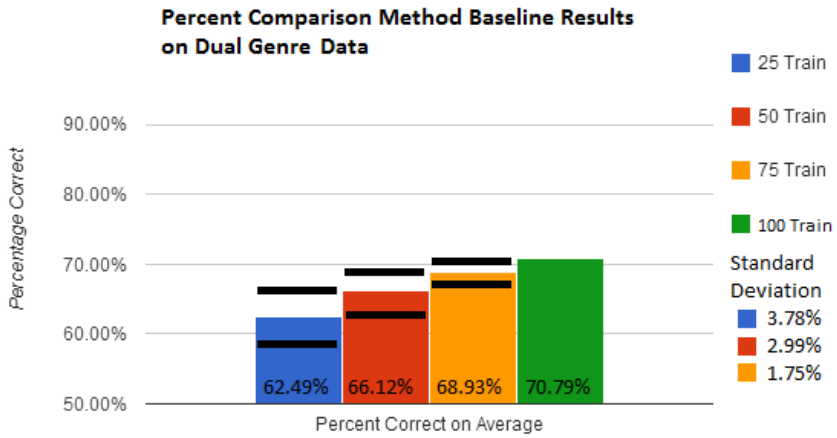

Figure 6.4: *The baseline comparisons of what percent of genres were correctly identified while testing on the dual genre test data with the percent comparison method.*

% Both Incorrect	% One Correct	% Both Correct	% At least one correct
2.97%	52.48%	44.55%	97.03%

Table 6.2: *Comparison of how many partial and complete correct genres identified in test data using 100% train data and score method*

6.4 Precision, Recall, and F-Measure by Genre

In order to gain a better understanding of the performance of individual genre identifications, I calculated the precision, recall, and F-Measure for each of my five genres. Precision examines how many of the results retrieved were actually relevant. Recall tells us how many of the results marked as a particular genre were actually relevant. F-Measure is the weighted average of the precision and recall. It calculates how accurate the genre was. By knowing these values, it is possible to learn which genres are performing the best in terms of identification. This data will give me an idea of which training data sets were functioning the best for identifying genres, as well as which genres were misidentified often. By knowing which genres are performing poorly, it will be possible to ascertain which genres need the most work for future identification efforts.

The formulas for precision, recall, and F-Measure are as follows.

$$P = \frac{TP}{TP + FP}$$

$$R = \frac{TP}{TP + FN}$$

$$F = \frac{2 * P * R}{P + R}$$

P stands for precision, R for recall, F for F-Measure. TP is the number of True Positives, or the number of times that a genre was predicted and correct. FP is False Positives, or the number of times a genre was predicted but was not correct. FN is False Negatives, or the number of times the genre should have happened but didn't.

6.4.1 Single Genre Precision, Recall and F-Measure

I calculated the Precision, Recall, and F-Measure for single genre identification set that had the best overall percentage correct for both the score and percent comparison methods. For both the single genre and dual genre identification this data set was the 50% training data usage set of results.

In Table 6.3 we can see the results of the different genres when compared using the score comparison method. For precision, western topped the table with an almost 100% precision. This means that it had very few false positives overall. Sci-Fi had the highest recall by about 2%, and Romance pulled the best F-Measure. The F-Measure results were very close for the most part, with all occurring in the 80% range and all but one managing over 85%. The genre that performed the worst on the F-Measure was fantasy. Fantasy had the lowest precision, which probably means that multiple items were misclassified as fantasy.

In summary, fantasy performed worst for F-Measure and precision when using the score comparison method on single genre identification. Romance performed best overall in F-Measure, though sci-fi and mystery were very close behind. Western had a very high precision, but a lower recall, indicating that western books were often misclassified as other genres.

	Precision	Recall	F-Measure
Fantasy	77.84%	90.62%	83.74%
Mystery	88.60%	88.74%	88.67%
Romance	89.02%	88.38%	**88.70%**
sci-fi	86.19%	**91.14%**	88.60%
Western	**98.34%**	77.10%	86.43%

Table 6.3: *Precision, recall, & F-Measure by genre for score comparison method on the single genre test data set.*

In Table 6.4 we can see the results of the genres when compared using the percent comparison method on the single genre data. For precision, western performed best. Romance had the highest recall by approximately 2%. F-Measure top result went to sci-fi. The genre which performed least well was fantasy, with the lowest F-Measure and recall scores for the percent comparison method. Mystery had the worst precision, indicating that this method might have a bias towards classifying things it shouldn't as mystery.

	Precision	Recall	F-Measure
Fantasy	83.77%	75.78%	79.58%
Mystery	79.22%	87.52%	83.16%
Romance	80.77%	**90.04%**	85.15%
sci-fi	86.86%	88.94%	**87.89%**
Western	**92.25%**	78.58%	84.87%

Table 6.4: *Precision, Recall, & F-Measure by genre for the percent comparison method on the single genre test data.*

6.4.2 Dual Genre Precision, Recall and F-Measure

Table 6.5 shows the precision, recall, and F-Measure for the score comparison method when used on the dual genre data set. The best values on the table are highlighted in bold. Sci-fi managed the best F-measure, and reasonable precision and recall. It appears that when a book was mislabeled, it was more often than not labeled fantasy. This can be told from the low precision score the fantasy genre had. Precision was highest in the western genre. If a book was identified as western, it usually was correct. However, western had the lowest recall of all genres which indicates that it was the genre with the most misidentified members.

Table 6.6 shows the precision, recall, and F-Measure for the percent comparison method when used on the test data set. The best values on the table are highlighted in bold. Sci-fi once again managed the highest F-Measure. This time around, sci-fi was also the most precise, though mystery had the highest recall by 2% over sci-fi. Fantasy was the least precise,

	Precision	Recall	F-Measure
Fantasy	52.25%	**87.50%**	65.42%
Mystery	73.17%	75%	74.07%
Romance	77.42%	60%	67.61%
sci-fi	72.34%	82.93%	**77.27%**
Western	**81.25%**	31.71%	45.61%

Table 6.5: *Precision, Recall, & F-Measure by genre for the score comparison method on the dual genre test data.*

though mystery was a close second. Western again had the lowest recall indicating that members of this genre were often mislabeled.

	Precision	Recall	F-Measure
Fantasy	62.22%	70%	65.88%
Mystery	63.46%	**82.5%**	71.74%
Romance	78.05%	80%	79.01%
sci-fi	**80.49%**	80.49%	**80.49%**
Western	73.91%	41.46%	53.13%

Table 6.6: *Precision, Recall, & F-Measure by genre for the percent comparison method on the dual genre test data.*

6.5 Conclusions

The hypothesis I stated at the beginning of this chapter was that I believed that as the amount of data used for training increases, the amount of correctly identified genres will increase as well. I further predicted that the percentage comparison method will outperform the score comparison method as it accounts for the overall amount of words in each genre.

6.5.1 Score and Percent Method Comparisons on Single Genre Identification

My hypothesis was supported by the result data in that the more data given, the more accuracy my program managed in identification in genre. Unlike my prediction, in the area of single genre identification the score comparison method outperformed the percent comparison method. I had theorized that the percentage would allow for better weighting of the values, and thus better accuracy. This did not happen while testing on the single genre data, as the percentage comparison method performed 1-5% worse than the score comparison method. Table 6.7 shows in boldface which method scored the best for identification across all genres while testing on the single genre test data. When a single genre needed to be identified, the score comparison method worked best for finding it.

6.5.2 Score & Percent Method Comparisons on Dual Genre Identification

When used to identify dual genres in books, the percent comparison method did better than the score comparison method. The amount of correct genre identifications also increased as the amount of training data increased. These results match my initial hypothesis. I was also pleased to note that both methods correctly identified at least one of the two genres on almost all books. The percent comparison method outperformed the score comparison

Test Method	25% Training Data	50% Training Data	75% Training Data	100% Training Data
Score	**83.75%**	**87.20%**	**86.83%**	**97.80%**
Percent	82.64%	84.17%	83.42%	92.40%

Table 6.7: *Score vs Percent comparison methods across all single genre data training amounts. Bold indicates best performance for that amount of training data for overall identification.*

method by finding a higher percentage of dual genre tags, and had fewer completely wrong entries than the scoring method. The percent method also managed to give 2% more of the books at least one correct tag than the score comparison method. The best performance numbers can be seen in bold in Table 6.9.

6.5.3 F-Measure, Precision, and Recall by Genre Conclusions

For the genre precision, recall, and F-Measure, the following trends were noted.

The single genre identification managed higher numbers across the board when compared to the dual genre identification for best and worst scores. Single genre identification results were between 10-30% better on precision, recall, and F-Measure. This matches the baseline results from Section 6.3.1 and Section 6.3.2. While the dual genre tests usually found at least one genre 95% of the time, it found both only around 40% of the time. This would drop precision, recall, and F-Measure values as going in it would be working with approximately one fourth of the data wrong going in.

For the dual genre comparison, the western genre did the worst in terms of recall and precision. It came in last in both the score and percent comparison methods. Western did however usually manage a fairly high precision probably due to the fact that the few westerns that were found were usually correct. Sci-fi performed the best genre wise on the dual genre classification.

Test Method	Both Incorrect	One Correct	Both Correct	At least One Correct
Score	4.95%	**55.45%**	39.60%	95.05%
Percent	**2.97%**	52.48%	**44.55%**	**97.03%**

Table 6.8: *Score vs Percent comparison methods for how many of the dual genres were identified using 100% of the training data for training. Columns compare how many of the dual genres were correctly identified or not. Bold indicates best performance.*

	Score Single	Percent Single	Score Dual	Percent Dual
Best Precision	98.34%	92.25%	81.25%	80.49%
Best Recall	91.14%	90.04%	87.50%	82.50%
Best F-Measure	88.70%	87.89%	77.27%	80.49%
Worst Precision	77.84%	79.22%	52.25%	62.22%
Worst Recall	77.10%	75.78%	31.71%	41.46%
Worst F-Measure	83.74%	79.58%	45.61%	53.13%

Table 6.9: *Score vs Percent comparison methods on how many partial and complete correct genres identified in dual genre test data using 100% of the training data for training.*

Single genre classification was a much closer contest. Often numbers were within a percent of each other for how well they performed. One pattern that both the score and percent methods did show though, was that fantasy was at the lower end for F-Measure. It was still in the 80% range, but it was lower compared to the other genres. As for best performance, western genre had the highest precision once more.

6.5.4 Comparison to Petrenz Research Baseline Results

In order to establish a measure of reference for my results, I looked to the 2009 thesis of Petrenz[15], wherein he compared various genre identification methods and how the style of the literature they are examining can affect their outcome. Figure 6.5 shows his results. In his work, he compared a number of methods. As can be seen in the charts, he ran the methods with and without parts of speech based features.

In comparison to my own work, my results held up. As can be seen in Figure 6.5, the results Petrenz calculated fell between 67% and 91%. The worst baseline I had was a 62% on the test data set while using only 25% of the training data for training purposes. When identifying a single genre, my accuracy fell within the 80-85% range that many of his meth-

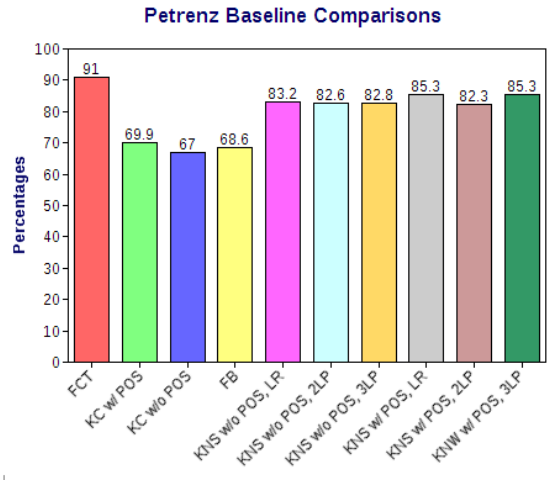

Figure 6.5: *Baseline comparison of 10 methods by Petrenz[15]*

ods did. The dual genre identification fell lower compared to the general state of the field, but my methods 90-95% ability to correctly identify at least one of the two genres given on a book makes me wonder if I should have found a better way to establish baseline on the dual genre classification. Additional work and ideas for better classification shall be discussed in Chapter 7.

In conclusion, my basic methods performed similarly to field methods in use by other genre classifiers today. My results fell within bounds of other genre identification methods in existence, and outperformed the worst results from the Petrenz comparisons[15]. It would be interesting to see if my classifier retains this level of accuracy as the test data pool increases, or if it would begin to fall on a larger data set. For the set I had, however, it performed comparably and in some cases better than the current methods out there.

Chapter 7

Future Research

7.1 Web Genre Classification

As mentioned in Section 2, web genre classification is a field of genre classification that is getting much attention. When polling the web for resources, researchers want their results classified into groupings. Currently, many of those groupings are topic based. Genre based groupings would provide additional information to researchers. In her paper on flexible classification schemes Santini said that people have preconceived notions of what a genre is. When a person is informed of the genre of a page, they can then make assumptions of the content and organization that the page might follow [1]. Genre labels allows for users to talk about documents that aren't necessarily similar while still using a common label [5]. For instance, a book on preparing Sushi and a second book on bread making could both be discussed under the same genre heading of cook books. They are not the same thing, yet they share a common theme.

Currently there are dozens of engines that will search a website or identify a sites topics [27], but very few that attempt to identify the individual web pages genres. As more and more websites filled with information begin to populate the web, there grows a need

for better tools to classify the information that can be found. Genre provides a way to classify types of data into groupings, rather than just sorting by topic [5]. Researchers such as Crowston and Santini believe that being able to classify web documents into their genres would help users to find what they are looking for in a more intuitive manner. Genres are defined by what people think they are, so tend to reflect the view of the people who create them [3]. This can help people understand what a document will be, which is useful when it comes to the sheer amount of data available on the web.

7.2 Optical Character Recognition

The field of optical character recognition, or OCR, might benefit from research into automatic genre classification. When scanning in books, it would be useful to have a program that could identify the genre or theme of the text for automatic labeling. This would be especially useful for unlabeled texts. According to Wikipedia, OCR is the mechanical or electronic conversion of scanned or photographed images of typewritten or printed text into machine-encoded/computer-readable text [28].

There has been a large push in recent years for libraries to get their microfilm collections into a stored digital format. Programs such as reCaptcha exist to help with the scanning methods [29]. The Library of Congress has been acquiring microfilm since the 1940s [30]. They have some six million records, all of which are being slowly converted over into digital format. If a librarian wishes to label what is in a text being read in, they have to either do so manually, or rely on topic catching programs. More tools to help automatically generate labels for recently scanned data could be of great use.

Wouldn't it be useful if genre identification could be run on the text once it had been

brought over to identify themes in the different documents and papers? As mentioned in the section on Santini, there has been work done for identifying web genres and multiple genre identification [1]. Much of the work I examined on web genre theory was working to identify different types of articles within newspaper collections. As much of what is being scanned in is old newspaper archives, these web genre programs could be used to take stock of the older newspapers to get an idea of what each contains. They could also be magazines, or other articles that may have been preserved to consider. Genre doesn't just cover fictional headings. Genres can be autobiography, opinion, magazine, biography, news, travel, poetry, and more.

7.3 Sentiment Analysis

I think the field of sentiment analysis might have some tools in it that could be useful for the task of identifying genres. Sentiment Analysis uses natural language processing to identify and extract important information from text data. In the paper "Sentiment Analysis of Twitter Data" the authors talk about using hand annotated training data to teach their program how to identify positive, negative, or neutral comments from Twitter feeds. They did so by creating lists of common positive and negative emoticons, as well as a table of positive, negative, and neutral words [16]. They then set up a polarity system to rank a comment as either positive or negative. If there were many good words in the review, the polarity moved towards positive. Words with negative connotations moved the review towards the negative polarity [16].

As mentioned in 2.3, some of the questions related to classifying genre have to do with the feelings the text invoke, or are conveying. For this reason, being able to run a sentiment analysis to pick up what emotion the text is generating could be an interesting task to run. For instance, if the analysis could be taught to identify sad emotions, it could be potentially

used to identify stories belonging to the genre of tragedy. Sentiment analysis could be an interesting facet to employ into a genre identifier for this reason.

Polarity could be an interesting system to set up, where there is a slider that moves towards a certain genre based on what words are being used in it. One problem I could see with using polarity, is the researcher would need to identify genres that are "across" from each other. This could potentially be modified to five data points, and just see how far they manage to slide into the territory of each specific genre to form a web graph of interest. Another problem be that this method is identifying the genres entirely by topic, as each topic would have to be classified in an area in order to know which way to move the slider.

7.4 Text Classification Methods: Naive Bayes & LSI

I think if I had to choose what methods to apply to analyze the text next, I would use either Naive Bayes or Latent Semantic Indexing(LSI). Given my current bag of words approach, and the data that I have collected, I think either of these could provide interesting new results.

Naive Bayes especially might do well, as the data set is small enough to be supervised. All records are labeled, so it could be set up to run, then check itself against the human labeled results. If additional test data could be acquired, it might become even more accurate. This is a type of supervised learning, which could be good or bad due to the subjectivity of genre. It would be a good thing because it would be getting the opinion of a human as to what the genre of the book should be. It could be bad as the data would be biased towards what the human thinks genres are as opposed to their true definitions.

LSI could be interesting to run as well, just to see what kinds of groupings it might come up with. I'd like to run this kind of analysis on the data without telling it what individual genres there are, just to see how it groups them based on their content. I could then check the groupings against actual classifications to see if it naturally placed the summaries from the same genres together or not.

It could be interesting to run multiple analysis on the text and then have them "vote" for where to classify a given summary. Each method could have a vote for where it thinks the new summary should best be placed, and the genre with the most votes is where it is classified. If there was disagreement, that could be noted as well to see how the program is thinking for indexing the summaries.

7.5 Relational Data

It could be of interest to attempt building a relational database. For instance, it is quite possible to acquire information on who the author of a given book is. If the author is known, it might be possible to look up the author and see what are common genres the author is known to work with. This might help for classification on the more prolific writers, or those who are writing series. This would not be a perfect method, as authors are not obligated to work within the confines of a single genre, but it would introduce a preliminary classification that the program could use to know which genres are more likely for a given publication.

Publisher information could also be of interest. Many publishers tend to concentrate on specific kinds of books. *Baen Books* is a publisher that specializes entirely in science fiction and fantasy genre books [31]. If a relational database of publishers could be built and the publisher data acquired for new books then it might be possible to sort books into

preliminary categories based on the genres the publishers are known to actually publish.

7.6 Data Cleaning

Changing the data cleaning methods could provide information of interest as well. I think it could be interesting to re-run the test data without stopword removal, so see if certain common words would cause the classification to shift. For instance, I am wondering if the mystery genre might use common words such as who, what, or where than the other genres do, and if their presence might make a difference on the results of the data.

Another interesting experiment to run would be to see if the use of different stop-word lists would change the results by a large margin. Would a more comprehensive list of words change the outcome? If I were to change the classifier k that is setting low occurrence words to disappear to be higher or lower, would this change the results?

Another idea might be to load in a standard English dictionary. Then if a word doesn't occur in the dictionary, it is removed from the text. This would take care of the problem of made up words and names being used in the text. The downside to this would be a hit on speed and memory usage. As the program would be working with a much larger data set, individual search times for words in the summary would rise.

Stemming the words could be an interesting experiment as well. Stemming refers to the process of using a program of cutting a word down to its base components. For instance, "swimming" and "swimmer" would be stemmed down to their root of "swim". This allows for words to be properly added up instead of all their conjugations being put each in their own sub-heading. It might be interesting to see how stemming words might change the

results of the data and whether it would provide more accuracy.

7.7 Named Entity Recognition

One area that could definitely use improvement is the named entity recognition in my program. At the moment, my program is using a combination of a name list and low occurrence removal to try to remove character names that occur in my training data. The training data does not need to hold the names of specific characters from books. Currently, if a name occurs more than three times across all genres, it doesn't get removed from my word count creator. This means that if a book mentions it's hero 'Zzzaphlx' more than three times in the summary, 'Zzzaphlx' is added as a word defining that particular genre. This bloats the training data word list which can slow computations. It also means that a word that has nothing to do with defining the genre has managed to make its way onto the genre definition list, which for the percentage method is bad as it lowers the overall percentage scoring system for that genre.

Irrelevant proper nouns bloat the training data and slow down look up, which is undesirable. My current method of using a name list from the social security website does its job of removing basic names, and works well for my small training set of 500 entries. However, if my work is to be expanded then a more permanent solution will need to be found.

Implementing a Named Entity Recognition program to identify and remove names from summaries would is the solution that I would like to implement. A named entity recognition tool would allow for cleaner training data, which is important as the test size increases. A properly implemented named entity recognition program could also help identify parts of speech. It could perhaps be used to eliminate prepositions or articles of speech that

shouldn't be in the data. It could also identify the names of companies or countries in the data for possible removal.

7.8 Training and Test Data

Another area for improvement is in the acquisition of training data. My data had 500 training records and 100 cross genre test records total. For training data, one hundred records of each genre type is a rather small sampling for defining everything that a genre is. I am not familiar with using data scrapers, so the majority of the data I used in these experiments was collected and labeled by hand. While a hundred records of each genre plus a one hundred entry set of dual genre books is a good starting point, I feel that having more would make the program more accurate and allow for better testing. It would also be of interest to acquire more records for testing. The one hundred mixed genre entries I had for test data allowed for some interesting results, but I would like additional single genre test files and dual genre test files to run analysis on to see how well the program holds up. It would be interesting to find out if there is a saturation point on the data for performance, where if it goes over that threshold performance begins to drop.

Bibliography

[1] Marina Santini. Automatic genre identification: Towards a flexible classification scheme, 2007. URL http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.101.3273&rep=rep1&type=pdf.

[2] Merriam-Webster.com. Genre, 2014. URL http://www.merriam-webster.com/dictionary/genre.

[3] Daniel Chandler. An introduction to genre theory, 1997. URL http://www.aber.ac.uk/media/Documents/intgenre/intgenre.html.

[4] Wikipedia. Dracula, March 2014. URL http://en.wikipedia.org/wiki/Dracula.

[5] Barbara H. Kwasnik, Kevin Crowston, Michael Nilan, and Dmitri Roussinov. Identifying document genre to improve web search effectiveness, 2001. URL http://surface.syr.edu/cgi/viewcontent.cgi?article=1133&context=istpub&sei-redir=1.

[6] Mad-Eyes. Time stood still, 2014. URL http://www.mad-eyes.net/disco/misc/time-stood-still.htm.

[7] OCLC. Worldcat, 2014. URL https://www.worldcat.org.

[8] Kathryn Zickuhr, Lee Rainie, and Kristen Purcell. Library services in the digital age, January 2013. URL http://libraries.pewinternet.org/2013/01/22/part-2-what-people-do-at-libraries-and-library-websites.

[9] Marina Santini. Marina santini - automatic genre identification, 2010. URL https://sites.google.com/site/marinasantiniacademicsite/.

[10] Microsoft Research. Automated document genre classification workshop: Supporting digital curation, information retrieval, and kownledge extraction, September 2009. URL http://www.dcc.ac.uk/events/workshops/automated-document-genre-classification-workshop-supporting-digital-curation-inform

[11] Marina Santini, Alexander Mehler, and Serge Sharoff. *Genres on the Web: Coputational Models and Empirical Studies*, volume 2011. Springer, 2010.

[12] Wikipedia. Document classification, March 2014. URL http://en.wikipedia.org/wiki/Document_classification.

[13] Yiming Yang and Thorsetn Joachims. Text categorization, October 2011. URL http://www.scholarpedia.org/article/Text_categorization.

[14] Christopher D. Manning, Prabhaka Raghawan, and Hinrich Schutze. *Introduction to Information Retrieval*. Cambridge University Press, 2008.

[15] Philipp Petrenz. Assessing approaches to genre classification, 2009. URL http://www.inf.ed.ac.uk/publications/thesis/online/IM090692.pdf.

[16] Apoorv Agarwal, Boyi Xie, Ilia Vovsha, Owen Rambow, and Rebecca Passonneau. Sentiment analysis of twitter data, 2014. URL http://www.cs.columbia.edu/~julia/papers/Agarwaletal11.pdf.

[17] Wikipedia. Heuristic (computer science), March 2013. URL http://en.wikipedia.org/wiki/Heuristic_(computer_science).

[18] Wikipedia. Genre studies, March 2014. URL http://en.wikipedia.org/wiki/Genre_studies.

[19] J.D. Robb. *Naked in Death*. Berkley, 1995.

[20] National Public Radio. Mysteries, thriller, & crime, 2014. URL http://www.npr.org/books/genres/10114/mystery-thrillers-crime/.

[21] National Public Radio. Your picks: Top 100 science-fiction, fantasy, August 2011. URL http://www.npr.org/2011/08/11/139085843/your-picks-top-100-science-fiction-fantasy-books.

[22] Peter Sykes. Top 100 fantasy series, 2014. URL http://fantasy100.sffjazz.com/lists_books.html.

[23] Inc. Goodreads. Goodreads, 2014. URL http://www.goodreads.com/.

[24] Mandi. List of english stop words, April 2009. URL http://norm.al/2009/04/14/list-of-english-stop-words/.

[25] Tamora Pierce. *The Realms of the Gods*. Simon Pulse, 2006. ISBN 978-1416908173.

[26] Social Security Administration. Top names over the last 100 years, February 2013. URL http://www.socialsecurity.gov/OACT/babynames/decades/century.html.

[27] Learn Stuff. Topics based search engines, November 2012. URL http://www.learnstuff.com/topics-based-search-engines/.

[28] Wikipedia. Optical character recognition, March 2014. URL http://en.wikipedia.org/wiki/Optical_character_recognition.

[29] Google. Recaptcha: Digitalizing books one word at a time, 2014. URL http://www.google.com/recaptcha/learnmore.

[30] Library of Congress. Microform reading room, August 2013. URL http://www.loc.gov/rr/microform/.

[31] Baen Publishing Enterprises. Baen publishing, 2014. URL http://www.baen.com/.