Moving Towards Automatic Classification:
Experiments Using Latent Semantic Indexing and Fiction in a Public Library Context

Nora MacLaren

DOK-3951

Master's Thesis in Documentation Science
Faculty of Humanities, Social Sciences, and Education
University of Tromsø
Spring 2013

Abstract

As we transition from physical to digital library collections, our classification systems need to change as well. But how is this to be done? Focusing on public libraries, this thesis examines how Latent Semantic Indexing could serve as the basis of an automatic classification of fiction using full text and the vector space model. Library patrons are the ultimate judges of any new system of shelf classification or search engine and their opinions are central to this thesis. To begin approaching the issue of an automatic, digitally born classification system, a survey was implemented to find out how patrons want to access the fiction collection at their local public library. Afterwards Latent Semantic Indexing was used in a set of experiments on a fiction corpus. Finally, readers were asked to judge the results of the experiments and their evaluation served as the basis of a discussion about the success and potential improvement of the experiments.

Key findings are 1) genre is an important access point to a public library's fiction collection, and 2) Latent Semantic Indexing has the potential to serve as an automatic fiction classification algorithm.

It is recommended that further testing be done on the connection between word use, fiction, and the vector space model.

Preface

My advisor, Tore Brattli, suggested that I test Latent Semantic Indexing on fiction literature. I found this idea inspiring. I had often thought during the first two semesters of my Master's degree program that I wanted to design a new classification system. Tore's idea combined that wish with my interest in computers and the digital world, the result of which is this thesis. The process has been an enjoyable, valuable learning experience and one that would not have been possible without Tore as my advisor. To him go my most sincere and heartfelt thanks.

Of course, a thesis that takes over a year to complete is hardly a one-woman project. This thesis also depends heavily on friends, family, coworkers, fellow students, and library patrons. In particular (but no particular order), I would like to thank Stine Fjeldsøe and Sigrid Fossland for advice and guidance during my during my internship at the Tromsø public library; Lee and Darcy MacLaren for their everlasting support and encouragement; Tony Hanssen for always being there for me, especially when I was frustrated with certain stages of my thesis; the members of my focus groups for setting aside the time to read and discuss the short stories; and the Gensim community for developing a software package that saved me countless hours of laboring in Excel.

My friends tease me for wanting to be a library revolutionary. My desire was to write a practical thesis with real world implications. I want it to inspire future research and not simply gather dust on a shelf. So in that sense, yes, go forth and revolutionize, dear thesis!

Tromsø, May 2013

Table of Contents

List of Tables, Equations, and Figures

1.0 Introduction

Johannes Gutenberg is credited with creating the printing press that would revolutionize the world's access to literature. Some alternatives used before were to copy by hand or use woodblock printing, which were time-consuming processes. With Gutenberg's printing press and its moveable type, production sped up and works became more readily available and diverse.

Now the world's access to information is changing again on a major scale as computers become ubiquitous. The printing press resulted in more materials in more languages. A computer that is linked to the Internet can send files around the world quickly and distribute those files to a wide audience. Computers have incredible computational power and are changing how search and distribution are done and opening up new avenues of research that would not have been possible before.

As the printing press and the computer have changed the ways that information is distributed so too have the institutions designated to information storage, retrieval, and distribution. Public libraries and their classification systems are a prime example of this continuing development.

Libraries, at least in their form as a repository, have existed for thousands of years. Staikos (2004) began his history of libraries in the Western world with the storing of writing tablets in the Bronze Age by the Minoan and Mycenaean civilizations. He wrote that the first public libraries were founded during the 6th century B.C.E. in Samos and Athens and that the Athenian philosophy schools also had libraries. The Romans had imperial libraries and, by the early part of the fourth century, thirty public libraries. In the Middle Ages books were preserved in the monastic libraries associated with Christianity. Modern libraries range from public, to private, to those associated with a particular institution like a university and they can be comprised of physical or digital collections.

Classification in a general sense is determining degrees of likeness between various entities like objects, emotions, and ideas. For libraries "classification is a process

whose purpose is to (1) specify the location of every bibliographic item on the library's shelves, and (2) display the subject relationships among various bibliographic items in a library's collection" (Chowdhury & Chowdhury, 2007, p. xviii).

Subject categorization has been used for nearly two millennia or more. Hegna (2003, p. 7) found that subject categorization has been in use since at least the third century since writings exist from that time documenting library collections by something other than alphabetization. Another example of subject categorization is much more recent. Thomas Jefferson arranged his personal library by subject in a scheme inspired by Baconian principles. Despite opposition by the Federalists, who thought buying Jefferson's collection was expensive and that it included too many "objectionable" works, Jefferson's classification scheme followed his collection to the United States Library of Congress. Later, head librarians would change the categories but retain the concept of arrangement by subject (Conaway, 2000). In the late nineteenth century, libraries made the transition to using enumerative classification schemes.

Enumerative classification differs from other forms of classification in that it "attempts to assign a designation for every subject concept required in the system" (Taylor & Miller, 2006, p. 533). The advantage to enumerative classification is that the schemes are fairly stable and subject arrangement makes it easier for library patrons to find works on related subjects. On the other hand, a disadvantage to enumerative schemes is that they "have a built-in obsolescence: they enumerate the state of knowledge at the time they were published and have to be frequently revised to accommodate new subjects" (Batley, 2005, p. 113).

An alternative to enumerative classification is a faceted scheme like Colon Classification (Ranganathan, 1960). The goal of such a scheme is to provide more flexibility and make it easier to accommodate new topics. Unfortunately, Colon Classification has not been updated in several decades and was never widely used.

Each classification scheme has advantages and disadvantages and these change depending on its design, as discussed above, and on the medium for which the classification is intended.

1.1 Limitations of Classification in a Physical Medium

Late in the 1800s, Melvil Dewey started work on what has become perhaps the most popular enumerative classification system in the world and has been adapted for many languages and cultures, the Dewey Decimal Classification (DDC). His work inspired Paul Otlet to create the Universal Decimal Classification (UDC) and the world's first fully faceted classification[1] system, Colon Classification (CC), which was created by S.R. Ranganathan. Around the same time the American Library of Congress developed its own classification system. Many other library classification schemes exist, but these four are considered to be the major systems in modern library classification history.

As mentioned earlier, enumerative schemes especially have trouble staying updated. Faceted systems have to be updated as well but their flexibility in describing new subjects makes the process easier.

Perhaps the larger issue at hand however is how these four classification schemes are products of their time. They were designed for a paper-based, physical world rather than a digital one. The most significant limitation is that these systems arrange documents in a one-dimensional order designed for library shelves.[2] The world's information is not one dimensional, a subject can be equally related to several other topics, but in a one-dimensional space each subject can be equidistant from only two other subjects. Forcing such a reduction of dimensional space changes the way entities can be arranged and resulted in some creative attempts to accommodate additional information, such as a secondary subject description.

---

[1] Faceted classification is "a subject concept arrangement that has small notations standing for subparts of the whole topic, which, when strung together, usually in prescribed sequence, create a complete classification notation for a multipart concept" Manning, Raghavan, and Schütze (2008, p. 7).

[2] Depending on one's perspective, a book can have several classification codes assigned to it and copies, dummy or real, could be placed throughout the library for each assigned code. From a practical perspective this can be difficult for a user since she may have to search several places in the library before she finds a copy. Therefore, common practice is to have all copies collected in one location and one main classification code assigned.

The card catalogue was one way to overcome the limitations of a one-dimensional organization. Here, one created as many sets of cards as needed to describe the different kinds of relationships between documents. Their physical form could also be exploited for allowing different kinds of access to the collection. Otlet, for example, theorized about polygonal and index card holders that would allow the cards to be put in at different rotations, giving better access to the multidimensional (faceted) aspects of UDC (van den Heuvel & Rayward, 2011). The card catalogue for a CC-based library could have cards with various holes representing the different facets of the system. One could then stick a thin rod through these holes and pull out the cards with the desired facet. These are solutions that can be implemented outside of the organization of the collection itself.

As regards physical organization of the collection, stickers could be used to enhance the one-dimensional shelf arrangement. Stickers placed on the spines of books could identify them as belonging to a certain category. This could be used for example in fiction collections, which are normally arranged by author's last name, in order to give an overview of each book's genre. By visually scanning the shelves patrons could thus quickly identify books that might be of interest. The sticker method gives patrons better access to the collection but at the same time shows the limitations of a traditional classification system that is designed for a single arrangement of the collection.

For the purposes of this thesis, traditional classification is here defined as classification designed for physical mediums like printed books and maps. This differs from classification-by-algorithm, which is primarily designed for digital collections and could be applied to physical collections provided that digital representations are available.

Traditional classification relies on human involvement to classify a collection's documents. In the 1800s when schemes like DDC, UDC, and CC were in their infancy there was no other choice. Human involvement in the classification process has many consequences, among which are 1) it is resource-intensive, and 2) it is difficult to classify consistently.

It is resource-intensive because of the staff required to inspect and document all items in a library's collection. This job takes time and is never ending as long as the collection continues to grow and develop. Classifiers and catalogers also have to be trained, a process which can take many years. A classification-by-algorithm system would be less resource-intensive as most of the work would be done automatically, thus saving many hours of human labor.

The second issue regarding traditional classification is quite difficult to resolve in a traditional system but could be improved in an algorithm-based one. Depending on a person's background and purpose a document could be classified many different ways. Even throughout a single cataloguer's career or day the same document could be classified differently. In summary it is hard to be consistent and expect that everyone else will follow the same guidelines in the same way. An algorithm on the other hand, which follows the same steps each time, would be able to classify consistently.

The issues discussed above are intended to illustrate the challenges surrounding traditional classification. As we transition to increased use of digital collections, new opportunities are found to meet and perhaps solve some of these challenges.

1.2 Transitioning from Physical to Digital Collections

Digital collections have the advantage of being able to be freed even more from the constraints of the physical medium. Solutions like card catalogues and stickers on book spines have been in use for years to give physical collections more flexibility, but are still fairly limited solutions. It is important that as we transition from physical to digital collections that methods are updated to reflect the possibilities of the new medium.

As an example of how methods need to change to reflect digital collections, take the classification codes that are normally printed on the spines of library books. Some would say that these obscure numbers and letters should be replaced by the subject names, for example optometry and medicine. However, then the logical arrangement would be alphabetical, resulting in music being placed between medicine and

optometry, which are two closely related subjects (Batley, 2005, p. 4). In a digital medium, we could use classification codes for organizing the documents but represent them differently to the user. As Taylor (2004, p. 317) wrote, "A hierarchical or faceted arrangement can be exploited without the user seeing any classification notations." Using a traditional classification system in a digital collection while hiding the codes would be a combination of traditional and newer, digital methods. Another option would be to use classification-by-algorithm to take full advantage of the digital medium. This will be returned to shortly.

WebDewey (http://dewey.org/webdewey/) is also an example of the transition from physical to digital. WebDewey is the digital representation of the printed DDC schedules and has several advantages. In WebDewey one can search for terms or numbers, use Boolean operators and truncation, browse up and down the hierarchy, follow links to related entries, access many more terms than are found in the printed index, and more (Bowman, 2005, p. 123). Another advantage to WebDewey is the turnover rate for updates. While the printed version of the DDC schedules is published annually, its digital counterpart can be updated as often as necessary and users immediately informed of the changes. At the same time, WebDewey could do more to free itself from its roots in the printed schedules. For example, WebDewey only includes about 36,000 of one billion possible classes built and available for search, which is why Brattli (2012) argues for a complete expansion of the DDC codes in WebDewey. Obviously it is not be reasonable to print one billion DDC classes in a reference work but since memory is cheap in the digital world, it is possible to include them all in WebDewey. WebDewey is discussed here as an example of how the digital medium could be exploited with updated methods.

Another example of the need to update methods as we transition to digital collections is the Internet. When the Internet first started to gain popularity, librarians wanted to classify it using the same, or very similar, techniques with which they had been treating their library's collections. For example, there was an attempt to index webpages using DDC (Jenkins, Jackson, Burden, & Wallis, 1998). Yahoo began as an attempt to apply semi-traditional classification methods to the growing Internet. Yahoo created their own classification system that allowed users to browse through categories of webpages (http://dir.yahoo.com/). The Internet quickly grew too large

for a task force of librarians to keep ahead of the massive indexing and classifying project. Techniques for managing the large amounts of information found on the Internet are constantly refined and new methods explored; yet library classification has remained much the same and has not taken advantage of these new techniques.

How then might the transition from physical to digital collections be better represented in the public library? Traditional classification does not exploit all the advantages of a digital collection so a better option should be found.

It was mentioned earlier that a disadvantage to traditional classification is that it was designed for a one-dimensional representation. This is especially a problem for enumerative classification systems like DDC that need to leave room for new subjects that do not yet exist. Also, in a two-dimensional classification scheme even if three or more subjects have equal relevance to another subject, two will have to be prioritized over the others to accommodate the linear arrangement. As we transition to digital libraries we can take advantage of multi-dimensional space to represent the collection.

At the same time we are transitioning from physical to digital libraries, we are also moving from browsing to searching. Therefore any new classification system will need to be able to support both browsing and searching in physical and digital collections. Lesk (2005) wrote that text searching may be the best option for digital libraries but also noted that it worked best for precise searches for somewhat unusual terms. In addition, he noted that vector models were widely used for digital search engines but need to be able to represent popular opinion. Perhaps then a vector and text-based classification would be a good place to start for creating a new library classification system that will aid in the transition from traditional classification to classification-by-algorithm.

1.3 A Brief History of the Classification of Fiction

Until now we have discussed the transition from traditional classification to classification-by-algorithm and some of the possibilities that exist in the latter. An area that lacks a proper, widespread classification system is fiction. The possibilities

presented by classification-by-algorithm invite the opportunity to make a new attempt at classifying fiction, this time using classification-by-algorithm.

First, let us define what is meant by fiction. Non-fiction attempts to treat with facts, whether historical, commentary, or otherwise. The Routledge Dictionary of Literary Terms described fiction as both a process and a result (Childs & Fowler, 2006). The process part referred to fiction arising from an author's imagination. The final result of this process is a work of fiction. It should also be the author's intention that the work be considered fiction, since in genres like historical fiction the border between fiction and description of historical events can often be fuzzy. Fiction can be used to various purposes, ranging from pure entertainment to social commentary or allegory. Therefore, fiction can be defined as a story told for entertainment or purpose based on the imaginations of an author.

In order for library users to find works of fiction, the collection has to be organized in some way. This is especially important as the collection grows in size (Baker, 1988). Fiction classification has generally been based on "classification-by-creator" rather than the "classification-by-subject" approach that is used for non-fiction (Beghtol, 1989). While "classification-by-creator" serves requests for specific titles or authors very well, it does a poor job of addressing other types of fiction requests, like those by theme, main character, setting, or genre.

The point of fiction classification is to "make it easier for library users to find the types of fictional work they want…" (Baker & Shepherd, 1987, p. 246). This is a fairly open requirement and has resulted in a number of different approaches to fiction classification.

Several attempts have been made to create a fiction classification scheme that would aid to guide patrons to desired works, some of which will be mentioned here. One of the most common methods in use in public libraries today is to divide the fiction collection into broad genres and then shelve the books alphabetically by author as a secondary classification system. Davis (1976) suggested that one use Dewey Decimal Classification numbers to classify fiction, using negative numbers so as to avoid confusion. This would give fiction all the same subject divisions as currently used in

non-fiction. Pejtersen (1977) created a faceted system, called Analysis Mediation of Publications, that was based on user needs. The overarching facets in the scheme are subject matter, frame, author's intention, and accessibility, since these are qualities that users often refer to when seeking out new literature to read or expressing their satisfaction with a book already read. While Pejtersen designed her system from the user's perspective, Beghtol (1994) based her system on literary aspects. Another important attempt at fiction classification came from the American Library Association, which recommended subject access along four dimensions; form/genre, characters, settings, and topics (O'Brien & Yu, 1996).

Baker and Shepherd (1987) also referred to two fiction classification systems. One, created by Briggs, divided the fiction collection into eight categories: story collections, fantasy, sports, mystery-suspense, girls' stories, science fiction, historical fiction, and general fiction. The other, created by Borden, kept part of the collection sorted according to author but made smaller collections for lesser known authors. The point of this scheme was to try to introduce readers to new authors. One can assume that if a reader is looking for a popular author, she already knows exactly what she wants, including author and/or title, which means that an alphabetical arrangement is more effective for her purpose. On the other hand, if a reader is open to browsing, having a system that reduces information overload through a smaller collection size is a good incentive for her to browse new authors.

A much more recent attempt to create a classification system came from Vernitski (2007). She designed a system that could be used by humanities scholars studying fiction. Because of the academic focus on intertextuality, the main classes of this system were Quotation, Allusion, Variation, Sequel, and Prequel. Public library users often ask for sequels and can also be interested in books that refer to one another. However, this classification lacks one of the main types of questions that librarians receive from users, which is genre.

Readers' advisory tools often focus on genre and one such tool is shelf classification. Shearer (1996) published a collection of articles dealing with reader's advisory called *Guiding the Reader to the Next Book*, which included a chapter by Harrell arguing for

shelf classification as a passive tool for readers and another by Cannell and McCluskey calling for increased genrefication.[3] In addition, two master's degree studies were completed at the University of North Carolina that focused on user satisfaction post genrefication of the collection (Huff, 2006; Richard, 1999). In Finland, Saarti (1997) did a similar study that also found that user satisfaction increased after genrefication.

Fiction classification does not necessarily have to be a shelf classification. Other tools can serve a similar purpose. For example, Saarti (1999) worked on the Finnish fiction index, Kaunokki, which could serve as the basis of a fiction search engine. In Sweden, EDVIN was developed as a fiction database of subject and genre terms based on user needs while Hidderly advocated democratic indexing like that used for images (Moyer, 2005, p. 222). At the same time Sear and Jennings (1991) noted that most books tend to be chosen directly from the shelves, indicating that shelf classification is quite important.

Another option is user-created classification. Amazon (http://www.amazon.com/) has made use of this by providing links for "Frequently Bought Together," "Customers Who Bought This Item Also Bought," and "Customers Who Viewed This Item Also Viewed" on their site. Since people who liked one item will generally like similar items, having recommendations based on purchase and viewing habits acts as a kind of built in classification by subject and taste. LibraryThing (http://www.librarything.com/) has taken a similar approach by allowing its users define which books are similar. Users do this by applying tags to books they have read. These tags describe ownership, genre, author, main character, type of literature, publication, reading public, and more. These solutions are well adapted to an active digital community but would not work well for designing a library's shelf classification. Having users decide how the books should be arranged would be subject to flux and might demand much in the way of library resources. The goal of a classification-by-algorithm system is both to take advantage of the digital medium and to spare librarians extra work.

[3] Genrefication is defined as the arranging of fiction titles by subject or genre (Taylor & Miller, 2006, p. 533) and is a type of fiction classification.

Kazantseva (2006) did interesting work on automatic identification of sentences that could be used to summarize short stories. The method is heavily based on grammar. These summaries could be used fiction classification since the most important aspects would ideally be extracted for the summary but this is not addressed in Kazantseva's work.

None of these systems have come into widespread use. Instead public libraries have used broad genre categories to organize their fiction collections rather than a full-blown classification system like those proposed by Beghtol and Pejtersen. Perhaps part of the reason is that these schemes were designed to be implemented by human classifiers and, especially in the case of Vernitski, this required incredibly in-depth knowledge of fictional works and multiple higher education degrees in order to classify properly. Baker and Shepherd (1987) argued that Pejtersen's system was simply too difficult to use.

As none of the systems introduced above seem to be acceptable on a wide scale and there has been little research done on fiction classification since 1995 (Moyer, 2005), it is worth examining whether a classification-by-algorithm approach could be a more effective and reasonable way to develop a new fiction classification scheme.

1.4 Research Questions

In the preceding sections the lack of good, useable, fiction classification schemes has been discussed. It has also been established that classification-by-algorithm presents different opportunities for digital collections. Seeking to combine the opportunity for fiction classification with classification-by algorithm, the following research question will be put forward:

> *How well can an algorithm classify fiction literature in a way that is relevant for public library users?*

Focus is placed on public libraries since their users access fiction from a different standpoint than what is common in research libraries. Focus is also placed on what is relevant for users since they will be the ultimate judges of any new classification system.

In order to answer the above question, the following questions will be explored:

1. *How do patrons choose fiction at the public library and what implications does this have for how the literature should be organized?*
2. *How could Latent Semantic Indexing (LSI) be used to make a fiction classification system that self-organizes according to how public library users search for fiction?*
3. *How well does the LSI algorithm, which is based on the vector space model and word use, perform on a corpus of works of fiction?*

1.5 Why Latent Semantic Indexing?

Latent Semantic Indexing, alternately called Latent Semantic Analysis (LSA), has been chosen for this test of automatic classification of fiction for several reasons. First and foremost it has been proven effective as regards synonymy. Another important reason is that it is based on the full text of the document. LSI allows us to arrange documents by relevance and is freely available. These reasons will be discussed in more detail below.

Perhaps one of the most important differences between fiction and non-fiction is that fiction can be ambiguous in its word choice. This is by no means a criticism, merely a statement that non-fiction tends to use exactly the words needed to describe a concept and that these words are fairly consistent for everyone with knowledge of that subject. Fiction, on the other hand, makes use of metaphors, analogies, and symbolism to approach a given subject matter in a variety of ways using many different terms. Therefore, if a classification-by-algorithm system is to be attempted, it is important to choose an algorithm with a good performance history with synonymy. Manning et al. (2008, p. 162) defined synonymy as "the same concept may be referred to using different words." LSI has a good performance record in this area because it analyzes texts on a concept-by-concept basis.

Taylor (2004, p. 241) argued, "Among the reasons for the failure of automated determination of aboutness is that a computer can determine what words are used in a document but cannot determine meaning." However, many other researchers have found algorithms that do a fine job of comparing documents to each other in terms of

aboutness. Rishel, Perkins, Yenduri, and Zand (2007), Gordon and Dumais (1998), and Lochbaum and Streeter (1989), for example, have had successful tests using LSI on non-fiction corpuses.

Another reason for choosing LSI is that it indexes the full text of the documents in the corpus. The two advantages to indexing full text are 1) search, and 2) having a better chance of understanding what the story is about.

In digital libraries, we need to be able to search in the classification system, either by classification number, metadata, or full text. In addition to having good metadata like subject headings, which Taylor and Miller (2006, p. 303) said accounts for an average 35.9% of keyword searches, there is an advantage to being able to search the full text. The full text does not have to be directly available for the user in order for her to search. An alternative would be to search in an index that uses every word from the full text. Since a human classifier or cataloguer does not have time to read every single book in their collection to choose words from the text that may one day be used in a search, a full text index is quick and effective alternative. By indexing the full text from the start, people can search for any term and probably get superior results.

As regards the second advantage of indexing the full text, authors like Bell (1991) have argued that one needs to read a work of fiction in order to properly understand what it is about. In essence Bell believed that understanding comes from having the full text of the document. Since LSI is based on term frequencies and generally uses full text it has a better chance of succeeding. While the relationship between word use and genre will be discussed later in this paper, the classification algorithm already has a better chance of succeeding if it accesses the full text of all the documents in the corpus. Were it to base itself on metadata, like title, it is highly unlikely that it would succeed without large amounts of human intervention, which is exactly what classification-by-algorithm is trying to avoid.

LSI sorts documents by relevance, though not popularity. Relevance is very important in a public library's fiction collection, as people like to read books that are at least somewhat similar to other books that they have read and enjoyed. The advantage to

arranging books by relevance is that it increases the chance that lesser known authors will be read, which is something that Baker and Shepherd (1987) have advocated.

From a practical standpoint, another advantage is that the LSI algorithm is available in an open-source programming package. Řehůřek and Sojka (2010) implemented LSI and other relevant tools, using the Python programming language, in a freely available package called Gensim. Gensim (http://pypi.python.org/pypi/gensim) was designed to implement popular semantic structure algorithms like LSI on large corpuses where documents are streamed in as needed, rather than residing in memory. The package is maintained and used by an active community.

This project is essentially a test of the relationship between word use and aboutness. As noted above, there are several reasons that LSI makes a good candidate for a fiction classification algorithm. Whether or not the experiments have successful results, it is still worthwhile to examine this relationship and attempt to apply LSI to fiction.

1.6 Definitions and Thesis Outline

The terms *document*, *short story*, and *genre* will be used often throughout this paper and will now be defined.

Document theory provides a range of definitions for a seemingly simple term, *document*. Traditionally the term was limited a printed or written text like a magazine, book, or handwritten note. The definition has since become broader. In her 1951 treatise *Qu'est-ce que la documentation?* Briet (2006) advocated for calling pictures of stars, rocks in museums, and antelopes in the zoo documents. Her opinion was that as soon as humans had processed something, for example taking a picture or placing it in context, it became a document. Another definition comes from Svenonius (2000, p. 8) who said that a document is "an information-bearing message in recorded form" and further noted "potentially any medium can serve as a carrier of information." These two definitions reflect the modern world in that we have many kinds of information carriers and neither would exclude the traditional definition. Without

meaning to exclude these modern definitions, for the purposes of this paper *document* will be used interchangeably with *book, novel,* and *short story*.

Short story should also be defined since the classification algorithm will be tested on a corpus of short stories. For context, Pasco (2010, p. ix) defined the novel as "a long prose fiction, that is unified, coherent, and literary." A short story is the same but shorter, as the name implies. The Routledge Dictionary of Literary Terms does not have a definition for short story but does discuss short fiction, which came into its own at the beginning of the nineteenth century for publication in magazines (Childs & Fowler, 2006). The dictionary entry cited several prominent short fiction authors who defined the literary form based on its length and focus. Short fiction does not tolerate digressions or moral commentary but can be done in a range of styles from Chekhov's "whimper" to Maupassant's "whip-crack". The stories collected for this project come from anthologies, have a definite focus and are approximately 20 pages or less, and will thus be called short stories throughout this paper.

Pasco called the novel a genre but that is not how the word will be used here. In order to differentiate, *literary form* will be used to distinguish between novels, short stories, plays, and poems. *Genre* will be reserved for describing the themes and styles of collections of stories, or as a way of describing what makes stories similar. Genre is a more general term than theme but will be used interchangeably as that is where the focus of the definition is being placed. Genres are helpful for describing works of fiction as it "provides more than conventions or a writer; it also gives a framework for a reader's expectations" (Mullan, 2006, p. 107). Examples of genres are romance, crime, historical fiction, fantasy, and science fiction. Such categories are already used by a number of public libraries to divide up their fiction collections. Knowing the genre lets readers know what to expect with regard to both style and plot.

These terms will be used frequently throughout the paper as answers to the research questions are sought. In order to answer the research questions presented earlier, the paper will be divided into three major sections.

The first section discusses a user survey that was implemented at Tromsø Bibliotek og byarkiv[4] in November 2012. The goal of the survey was to discover how library users searched for fiction and what implications this has for a good shelf organization.

Next, LSI and theory surrounding the model will be examined in more detail. Tests were run on a corpus of 44 short stories.

In the third section the results from the LSI tests will be presented, evaluated, and discussed. The evaluation is based on criteria created by two focus groups comprised of fiction readers.

In the conclusion of the paper, recommendations have been made for further research in algorithm-based fiction classification.

[4] Tromsø Bibliotek og byarkiv will henceforth be referred to as either Tromsø library or Tromsø public library.

2.0 User Survey

In the introduction to this paper it was noted that library collections are undergoing a transition from the physical to the digital, that methods need to be updated to reflect this change, and how classification-by-algorithm could be used to improve fiction classification. Before the LSI algorithm was tested, we needed to determine the desired result. What sort of connections should we be looking for? It is also important to have an idea of which communities of library users would benefit depending on how LSI would be implemented in a library system. Since the goal is to create an algorithm-based classification system to be used in public libraries, it is important to know how these users search for new fiction to read so that a new system can be tailored to their needs. Other surveys and researchers have stressed the importance of genre as an access point for fiction readers. A survey of users at the Tromsø public library was used to confirm these findings and the results provided guidelines for the LSI tests and evaluation.

2.1 Survey Research Questions and Hypotheses

The LSI algorithm has been tested on a corpus of fiction literature to identify similarities between the documents. Since the results of the algorithm need to be relevant for library users, it is important to establish which kinds of similarities will be most useful and how the results could be implemented. In order to do this, a survey was conducted using the following research questions as a starting point:

1. *How do public library users search for fiction to read and what are they looking for?*
2. *Which kind of organization would best suit each type of user (as defined in the next section)?*

The hypotheses to these research questions are based on the work of Willard and Teece (1983), Richard (1999), Baker (1996), and Goodall (1992). In one study 48.2% of those interviewed responded that they had come to the library to browse[5] while only 18.1% wanted a specific item (Willard & Teece, 1983, p. 59). Richard cited a

[5] Willard and Teece divide browsing into three categories: general browsing, general purposive browsing, and specific browsing. In this paper, browsing will be used to cover all of these types.

number of studies and among them Spiller, who found that 69% of users wanted specific books, and Jennings and Sear, who came to the conclusion that "browsing is the most popular method for finding books" (1999, pp. 9-10). Specifically Jennings and Sear noted that "86 percent of those who borrowed fiction had *not* used the catalog for finding books" (as cited in Baker, 1996). Goodall (1992) and Baker (1996) both found browsers to be a large but neglected population of library users and thus recommended how to improve services in order to meet browsers' needs. The results of these studies have presented a range of percentages related to how many people are browsers but seemed to indicate that there are approximately equal numbers of browsers and non-browsers, those who are searching for something specific. Of course, this does not imply that a user is always a browser or always looking for a specific work. A user's needs vary from visit to visit; she may want a good book without specific criteria one day but desire The Hobbit the next time she visits the library since she just saw the film. Therefore the first part of *Hypothesis #1* states that for any given visit, users can be divided into approximately equal groups of browsers and non-browsers.

The search technique used reflects the patron's desired outcome. Non-browsers normally attempt to find a particular author or title, whereas browsers peruse the collection looking for books that match other criteria. What then are they looking for? Goodall (1992) focused more on browsing in general but her recommendations include that the fiction collection be organized with clearly marked categories and that the catalogue help patrons find certain kinds of books. Both of these suggestions indicate an interest in access via genre. Sear and Jennings (1991) found that patrons' top three methods for choosing fiction were "browsed/looked interesting," "browsed/recognized author's name," and "looked for genre." In addition they found that readers enjoyed 52.3% of books chosen by looking for genre. Readers enjoyed 80.7% of books that were chosen because of the author. Thus while it is a safer bet to choose a book because one is already a fan of that author, it is also generally successful to choose a book based on its genre. The studies discussed below assume that genre is the most important criteria for readers deciding whether they might read a particular work of fiction. Based on this assumption, Goodall's recommendations, and from the author's personal experience working at public libraries, the second part

of *Hypothesis #1* is that patrons are most often looking for a specific work/author or genre.

The study by Richard (1999) and its follow-up by Huff (2006) sought to determine the effect of genrefication on users' borrowing habits. Richard found that though circulation did not increase, library users were more satisfied with the genrefication system than with the old system where fiction had not been classified according to genre. Huff (2006) also received positive feedback for the system. The results from Richard and Huff are supported by Baker (1996), who recommended genrefication as a way to cut down on information overload. When a large collection is clearly divided into smaller, relevant sections, the chance of finding something interesting increases. When the collection is large a browser has to sift through many more books in the hopes of finding something she wants to read and this can result in a feeling of information overload. Shoham (2000, p. 54) also supported genrefication and other forms of categorization because the categories can stimulate the interests of those readers with no fixed need in mind. Pejtersen and Austin (1983), on the other hand, noted that early fiction schemes were difficult to implement in practice, in part because designing the system for shelf classification meant giving a one-dimensional view of the collection. This occurs because each book could only be sorted into one genre and an author's works were often clumped together even though the collection spanned a variety of genres. However, most support is given to genrefication as it seems to help browsers succeed at the library. Therefore the first part of *Hypothesis #2* is that browsers prefer fiction to be organized into genre categories.

If *Hypothesis #1* is correct that library users are roughly divided into browsers and non-browsers, then *Hypothesis #2* also needs to provide an idea of how users searching for specific works would prefer the fiction collection to be organized. The disadvantage with genrefication for a non-browser is that she has to know in which genre the book might be found. If she is not well acquainted with the fiction collection, a non-browser runs the risk of having to search several sections of the library and before eventually finding the desired work. Based on this, it can be assumed that users who are generally non-browsers would prefer all the works of fiction to be gathered together into one, large, alphabetically organized collection.

Hypothesis #2 therefore states that browsers prefer genrefication and non-browsers prefer alphabetization.

2.2 Survey Theory

The goal of the Tromsø library survey was to confirm results about percentage of browsers and search methods from other surveys. Because of this, the simplest method would be to use the same questions as used in these other surveys. However, search methods will be dependent on how the library is organized. Therefore, in order for the survey to be relevant for Tromsø library users, it was decided to write a survey from scratch, using other surveys as guidelines. In designing a good survey, many aspects needed to be taken into consideration and two of the most important areas will be discussed below.

2.2.1 Sample Bias

When determining whose opinion should be consulted for the survey, one option would be to force absolutely everyone in the target population to answer the survey questions. Accomplishing this in a large population, however, requires much in the way of resources and personnel. Therefore it is more common to select a sample from the target population and assume their opinions reflect those of the population as a whole. Ideally the sample would be chosen completely at random so as to avoid bias but this can be difficult to implement in practice. Knowing that bias can be introduced to the survey sample, let us examine how to reduce it.

According to Fowler (2009), three types of sample bias exist that need to be taken into account when selecting participants.

The first type of sample bias deals with the sample frame; where the sample frame is defined as the people who have a chance of being included in the survey. The sample frame is different from the target population in that the target population could be everyone who lives in the USA but the sample frame is limited to people who live at their registered home addresses, which would exclude the homeless and prisoners among others (Fowler, 2009).

Since the goal of this classification-by-algorithm project was to experiment with creating a new fiction classification scheme for public libraries, the target population for the survey was library users. The population was further reduced to adult library users since children's literature was not part of the LSI corpus. The survey was only be given out at Tromsø library so the sample frame was thus further restricted to adult patrons of the Tromsø public library.

Assuming that other kinds of sample bias are reduced, the sample frame will be a representation of itself but perhaps on a larger scale could give indications about Norwegian library users. The surveys referred to above were from the USA and England, so if the results of this survey confirm what has been found before, perhaps the sample frame can also give an indication of what users in these countries might also prefer. However, extrapolation is difficult since cultural differences can have a significant effect.

Another kind of sample bias comes from the participant selection process. The selection process can be a random sample, as in drawing names from a hat, or a nonrandom sample, where people nearby are asked if they are willing to participate (Fink, 2009). In this case a nonrandom sample approach was taken due to the legal issues concerning contacting library users from their personal information in the library database. The participant selection process included asking people at the library to participate as well as advertising a link in the library blog. Both of these processes were dependent on people's goodwill and interest in responding. As the sample was nonrandom, it needed to be taken into consideration that people who are willing to respond to surveys are likely to have slightly different opinions than those who avoid surveys (Fowler, 2009). It is unclear exactly how their opinions might differ but the potential bias effect of a nonrandom sample is worth noting.

The third type of sample bias is "…failure to collect answers from everyone selected to be in the sample" (Fowler, 2009, p. 14). Ideally everyone above the age of 18 who used Tromsø public library or read its blog would have been participated in the survey. However, since the surveyor had limited time and resources, the sample was limited to a total of 58 respondents. Had more resources been available, higher

comprehensiveness could have been achieved. At the same time, Fowler (2009) noted that small changes in the percentage of the surveyed population do not tend to have a significant effect on the results. Having fifty-eight respondents allows us to draw some conclusions, which is enough to say whether the results seem to confirm or deny the other surveys' results mentioned earlier.

Having explored how the sample could be biased it is also important to discuss how the survey responses could vary in accuracy.

2.2.2 Survey Response Accuracy

In addition to determining the sample frame and making conscious decisions to avoid bias in the sample, the survey itself had to be well designed and clearly written so as to collect more reliable data. A well-designed survey is aware of which kinds of information each type of question will collect and what certain kinds of responses can be used for. In addition it must have clearly written questions and account for "dishonest" responses.

Survey responses can provide either objective facts, like height, or subjective states, such as how tired someone feels (Fowler, 2009). This survey included questions seeking both kinds of information. Examples of questions seeking objective facts include those dealing with age, gender, and how often the participant visits the library. Subjective state questions included asking how well acquainted the participant was with the library. Knowing the difference between these two types of information allowed us to draw different kinds of conclusions.

Next it is important to consider how the questions and responses (for when participants were asked to choose from a list) were written and presented. Three main issues needed to be considered: 1) misunderstanding a question, 2) difficulties with open-ended questions, and 3) order of suggested responses.

A possibility always exists that someone will misunderstand a question. If this happens she will give a different response than what she might have otherwise

answered. Thus it was important to write questions and suggested responses as clearly as possible to reduce this risk.

In terms of collecting people's opinions, it was also worth considering how the responses are presented. Open-ended questions have a number of drawbacks including being demanding for participants, forcing participants to recall issues they may not have thought of recently, being difficult to quickly and easily analyze data, introducing many varied responses, and rarely being accurate or being a consistent measure the population (Salant & Dillman, 1994).

When participants are asked to choose from a list of responses, these can be ordered or unordered. Ordered responses normally present a range of opinion from strongly agrees to strongly disagrees. Answering this sort of question is easy for participants but their responses can be highly subjective. In order to write unordered responses, the survey designer needs to be well acquainted with the subject area so as to cover the most likely possible responses. Many of the questions used in this survey had unordered responses. With unordered responses it is important to note that in mail-based surveys, people are more likely to answer from among the first choices and in telephone and interview surveys to choose from among the last choices (Salant & Dillman, 1994). Since people have a tendency to choose from the beginning or end of the list depending on the circumstances, a possible solution to avoiding unintentional bias would be to randomize the order of responses for each participant.

A different kind of inaccuracy in surveys arises when participants want to impress the surveyors or feel better about their own responses. For example, people tend to overstate how often they vote and understate how much they smoke (Fowler, 2009, p. 16). In this survey, questions that were particularly vulnerable to over-estimation were 1) how often participants visit the library, or 2) how much they read. Not only is it difficult to guess an accurate response but the participant may also have wanted the library to receive more funding or to feel smarter based on how much she has read, which can lead to unintentional bias in the survey results. Though response accuracy may be unintentional or well intentioned, the chance of "dishonest" answers still needed to be taken into account during the analysis process.

Knowing that the sample and survey responses may be inclined to bias or inaccuracy, the Tromsø library survey process and results will now be presented.

2.3 Survey Execution and Participant Demographics

A total of 58 responses to the survey were collected, both by asking people in person at Tromsø library and online through the library blog (http://biblifokus.wordpress.com/2012/11/15/brukerundersokelse/). The majority of responses (~45) were from people visiting the library. Target areas in the library for finding potential participants were the main entrance and the floor dedicated to fiction literature for adults. Each participant was presented with an information sheet about their anonymity in the survey, details about what the study entailed and what it would be used for, and contact information if there were any questions. This sheet has been reproduced in Appendix A. While it is likely that all survey respondents were patrons of the Tromsø public library, there is also a possibility that people unaffiliated with the institution responded to the survey out of curiosity or general goodwill towards library surveys.

All responses to the survey were collected digitally through Formsite (http://www.formsite.com/). A link was given out in the library blog and participants at the library used an iPad to access the survey form. Formsite was chosen over similar services due to its ability to have conditional branching (multiple lines of questioning), as this made asking follow-up questions easier.

The survey consisted of ten questions, with room for additional comments, and could be taken in Norwegian or English. The survey and responses gathered can be found in Appendix B. The goal was to create a short survey, which could be completed within five minutes, and could be answered by almost any patron of the Tromsø library. Only one patron turned down the request to participate due to language barriers.

Of the survey participants who responded to the demographic questions, 78% were between the ages of 21 and 65, and 75% were female. Thirty-eight percent came to the library once or twice a month and 40% answered that they read mostly fiction. Thirty-one percent of the participants considered themselves to be well acquainted with the library and another 31% admitted that they were not well acquainted. This is

a subjective question and rather than being taken as a concrete skill level, it should be seen as indicative of how comfortable patrons feel finding materials at the library.

Statistics from Tromsø library from February 2012 show that, of library users above the age of 18, 56.6% are between 19 and 40 and 43.3% are 41 years or older. In the survey, 48% fall into the first category and 46.2% into the latter. Therefore the survey sample was fairly representative of Tromsø library users in terms of age.

The statistics gathered in February 2012 did not cover gender so for comparison purposes a different study, *Brukerundersøkelse*, was referred to, which was also undertaken in November 2012. In the *Brukerundersøkelse* a total of 157 women and 79 men responded, which is 66.5% and 33.5% respectively. As noted above, in the survey done for this paper women comprised 75% of the survey population. A certain amount of statistical variance is allowed and since both surveys had similar percentages of women and men we can assume that these reflected the total population of Tromsø library users.

For the entirety of 2012 at Tromsø library, almost equal amounts of adult fiction and non-fiction were loaned out, which includes renewals and inter-library loans. Specifically this was 51.5% for non-fiction and 48.4% for fiction literature. Interestingly, more people who responded to the survey said that they read fiction, just over 60% in fact. This means that there was a slight discrepancy between the target population and the survey sample. The discrepancy can most likely be explained by the fact that several people turned down the chance to participate because they felt they did not read enough fiction to have an opinion on its organization at the library, which gave the results a bias towards fiction readers.

To summarize, survey respondents included men and women in a range of ages, who spoke Norwegian or English. Almost half visited the library once or twice a month and forty percent read mostly fiction literature. Responses were gathered digitally because it was easier to have conditional branching.

2.4 Survey Responses Regarding Search Techniques and Collection Organization

After collecting demographic data about each participant, the survey presented questions that would collect data about popular search techniques and preferred organization of the fiction collection.

Questions 6-8 were designed to determine whether the survey participant tended towards being a browser or a non-browser when visiting the library and what he or she was looking for, specifically or generally. Each question accepted up to three answers and on average participants gave 2.2 responses.

When the question asked what the participant looked for at the library, responses showed an interest in specific authors (24.2%), genre (17.4%), titles (15.9%), and literary form (13.6%).

The next question was formulated slightly differently and asked *how* the participant searched for books at the library. Here again the majority response was to search for specific authors and titles (26.4%) but a total of 62% of the responses showed interest in a variety of browsing techniques. Participants said that they browsed in a general way (13.9%), looked at books that were recommended by librarians (12.5%), and books that were published that same calendar year (11%). The remaining choices were asking a librarian (10.2%) and other (1.4%). These last two responses are difficult to categorize as browsing or non-browsing because of the wide range of inquiries they represent.

The third question in this section dealt with how much a reader wanted to know about a book before he or she started reading it. Here the majority responses were genre (29%), information contained on the book jacket like author, summary, and reviews (25.8%), and that an acquaintance had read the book (17.7%).

The last section of the survey (apart from the opportunity to give additional comments) asked how library users would prefer the books to be organized. The inspiration for this question comes from Harell and Corns' categorization techniques,

which are spine labeling, separation, and a combination of the two (as cited in Yu & O'Brien, 1999). In question #9 respondents were given three options and each response had two follow-up questions. Participants were asked whether they would prefer 1) all works of fiction to be organized in one section according to author's last name, 2) certain kinds of books to be separated from the main collection, or 3) that the entire collection be divided into smaller categories. Half of the responses to this question indicated that users prefer the entire collection be divided into smaller categories of some kind. These participants were then asked for clarification about which categories they would prefer and 57.1% answered they wanted the collection divided by genre. Over thirty percent of survey respondents indicated that they wanted all the fiction books organized together alphabetically by author's last name. Of this thirty percent, 42.8% said they would mark the books according to genre (e.g. stickers on each book's spine). Genre was also a significant response to the follow-up question from response #2; literary form was the most popular response but here the sample is so small that it is difficult to say whether this is truly representative of the target population.

The last question of the survey gathered opinions about special book displays that could be permanent or temporary. Survey respondents gave an average of 1.6 responses to this question. Here over 60% of responses indicated library users liked the displays of new and recommended literature. Other popular responses included desire for and/or use of displays related to current events (15.6%) and holidays/seasons (13.5%).

At the very end of the survey, participants were given the opportunity to provide extra comments related to the survey or give general feedback to the library.

2.5 Discussion

Based on the summary of the results found in the user survey at Tromsø library, the results will now be discussed in light of the research questions and hypotheses that formed the foundation of the survey.

Hypothesis #1 posited that library users could be divided into approximately equal numbers of browsers and non-browsers. In addition, it posited that users look most likely for a specific work/author or genre. In the survey given at Tromsø library participants were given the opportunity to describe their search habits at the library and results indicated that they divide themselves into more or less equal numbers of browsers and non-browsers. After author, most responses indicated that genre was an important criterion for reading and organizing the collection. This supports *Hypothesis #1*.

Hypothesis #2 proposed that non-browsers would prefer alphabetization and browsers would prefer genrefication. There was a preference among participants who searched for specific authors that everything be organized together alphabetically. The results also indicated that there was a general preference for fiction to be organized according to or marked with each work's respective genre in some way. This seems to confirm *Hypothesis #2*.

An unexpected result from the data gathered was that users who read more non-fiction had a preference for genrefication. It is possible that this can be explained due to the fact that a library's non-fiction collection is typically organized using the Dewey Decimal Classification System or another subject-based scheme. Being used to this system and aware of the effect it has on finding literature, users would likely recommend it for fiction organization as well.

The major indication gathered from the responses to this survey is that readers want to be able to access fiction at the library by genre, in addition to alphabetization by author's last name, and that the sample was roughly equally divided into browsers and non-browsers.

As noted earlier, Baker would most likely support these recommendations while Pejtersen's historical account might oppose them. Both of these arguments are valid. For example, based on the author's experience, separating the crime/detective novels from the rest of the fiction collection at Tromsø library seems to have increased circulation in this category. There were often patrons browsing the shelves and the stock kept in the back room was frequently used to replenish the shelves available to

the public. This supports Baker's view but Pejtersen's point is also valid. After all, some of the books that have been categorized as crime novels at Tromsø library show aspects of other genres as well. Håkan Nesser's *Himmel over London* is described as a new type of book that combines a crime novel's action with fiction's depth (Gyldendal). Should this sort of book be forced to show only one aspect of itself through the classification system? As both Baker and Pejtersen made reasonable points, it is important to find a balance between the two.

Given that the survey results seem to support the hypotheses and taking into consideration Baker and Pejtersen's arguments, the following recommendation can be made: create an algorithm that classifies fiction into genre-like categories that are more specific than the broad categories normally in use at public libraries. Such an algorithm could be helpful in finding the balance between genrefication and alphabetization of fiction depending on how it is implemented. It would also have less influence from human error/inconsistency and save time. To this purpose, the Latent Semantic Indexing algorithm will be tested on a fiction corpus in the next section of this paper.

3.0 Latent Semantic Indexing and Fiction

Above it was established that library users used browsing techniques about 50% of the time, that they liked accessing fiction via genre, that genrefication aided browsers in finding literature, and that we are in a transition period from physical to digital library collections. Let us now turn to how Latent Semantic Indexing (LSI), alternately known as Latent Semantic Analysis (LSA), might be used to create a digital, automatic, genre classification system for fiction.

To review, the goals of this project include: 1) creating a system that aids library users in finding new works of fiction literature based on genre preferences, 2) a system that could be implemented either digitally or physically (through shelf classification), and 3) saving time for library cataloguers and being less prone to human fallibility in assigning a classification code.

As far as the author is aware, LSI has only been tested on non-fiction corpuses. This project is therefore a first attempt at using LSI on a fiction corpus. The process of preparing a corpus and applying LSI will be examined in detail. Testing will compare various corpus preparation options so that a better understanding will be gained of how these techniques function on a fiction corpus.

3.1 Latent Semantic Indexing's Origins and Functionality

Deerwester introduced LSI in 1988 at the first annual meeting of the American Society for Information Science. He developed LSI in an attempt to improve the ability of search engines' functionality to address two main problems: synonymy and polysemy.

English is an example of a language with a very rich vocabulary. It is therefore common that people do not always, or even consistently, use the same words to describe a concept. At the same time some words may have more than one meaning. Synonymy is the term for when several words can be used for the same concept. Take for example an article that uses the word *buy* and another that uses *purchase*. *Buy* and *purchase* have very similar meanings and ideally if a user queries for *purchase*, articles using the word *buy* should also be returned as relevant results. Polysemy

describes how the same word may have several different meanings. For example *elephant ear*, which can refer to a part of an elephant's body or a type of fried dough covered in cinnamon and sugar. Given *elephant ear* as a query, documents about the animal or the food item could be considered equally relevant. In order for a search engine to function at a high level, it needs to be able to return relevant documents and overcome the difficulties presented by synonymy and polysemy.

LSI addressed synonymy and polysemy "by treating the unreliability of observed term-document association data as a statistical problem" (Deerwester, 1990, p. 391). Deerwester used a mathematical technique called Singular Value Decomposition (SVD) to reduce unnecessary noise in the statistical model of the documents. By doing this LSI understood the higher-order semantic structure of the documents, rather than analyzing simple word use.

LSI makes use of the vector space model for representing the document corpus. This is "an algebraic model that maps the terms in a document into an *n*-dimensional linear space" (Ingersoll, Morton, & Farris, 2013, p. 46). In other words, each document is represented by a vector whose components are based on the terms used in that document. As the characteristics of each document's vector are dependent on the terms of which it is comprised, the representation in the vector space model is influenced by word co-occurrences. Each vector (so long as each terms list is unique) will point in a slightly different direction than all the others. Documents with many words in common will point in similar directions. One way to measure similarity between these vectors, and the one that will be used throughout this paper, is to calculate the cosine of the angle between them. A high cosine similarity, closer to 1.0, indicates similar vectors and a low cosine similarity, closer to -1.0, indicates that the vectors have little in common.

In the original vector space model, where each vector represents a document,[6] the number of dimensions is equal to the number of terms, or attributes of each vector. In

[6] The vector space model can either be used to represent the documents (vector is based on term frequency for that document) or the terms (vector attributes come from which documents the term is used in). Throughout this project the former model will be used.

this model, a considerable amount of variability and statistical noise exists. SVD reduces the number of dimensions so as to eliminate noise and this results in a higher-order representation of the corpus, which reveals its latent semantic structure.

The SVD process begins by factoring the original term-document matrix, A, into three matrixes (U, S, and V^T), which represent term-by-concept, concept-by-concept, and concept-by-document. In order to reduce noise in the matrix and examine its higher-order semantic meaning, each of these three component matrixes are reduced and then recombined to give an approximation of the original term-document matrix.

As discussed above, LSI was designed to solve the problems of synonymy and polysemy in search engines. At the same time, it does have some drawbacks.

A potential disadvantage to LSI is its scalability. Berry (1992) was concerned about reducing computation times and recommended several methods that might aid LSI in terms of scalability. The problem he faced was that large corpuses result in very sparse matrices, since each document uses only a fraction of the words in its language, as long as it is not a nonsense document or dictionary. However it seems as though this problem has been solved since Berry wrote his article. According to experiments done using Gensim on the English Wikipedia, it took 4 hours and 9 minutes to create an LSI model of 3.9 million documents on a MacBook Pro with 16GB of RAM, which equates to approximately 16,000 documents per minute (Řehůřek, 2013). Using this as a reference we can infer that processing 100,000 documents, which is about the size of Tromsø library's collection, would take 6.25 minutes. Perhaps scalability would still be an issue when creating a LSI model of the Internet, which contains between and exabyte and yottabyte of data (Answers Corporation, 2013), but based on the Wikipedia experiments it seems as though processing times have overcome the scalability issue Berry was concerned about.

From a theoretical standpoint it could be disadvantageous that LSI, where each term consists of a single word, does not account for word order. As Landauer, Foltz, and Laham (1998, p. 5) wrote, "It makes no use of word order, thus of syntactic relations or logic, or of morphology. Remarkably, it manages to extract correct reflections of

passage and word meanings quite well without these aids, but it must still be suspected of resulting incompleteness or likely error on some occasions." Word order certainly contributes to meaning and is something that could be considered in future research of the applicability of LSI to fiction. However, since evidence supports that LSI achieves sufficiently accurate results despite its inattention to word order, word order will not be considered further in this project.

Despite these potential drawbacks, LSI presents many positive qualities that make it a worthy candidate for fiction classification/search and retrieval. Though it is doubtful that "the human brain uses the same mathematical algorithms as LSA/SVD, it seems almost certain that the brain uses as much analytic power as LSA to transform its temporally local experiences into global knowledge" (Landauer et al., 1998, p. 34). LSI is thus considered a powerful computational tool and in addition it "can return legitimate answers when there is a terminological mismatch between a query and a document" (Lochbaum & Streeter, 1989, p. 674). The process requires no human involvement and makes no use of human-made dictionaries, grammars, etc. In a test based on a synonym and antonym dictionary, the synonym-antonym pairs showed over 12 times as much similarity in the LSI analysis as words completely unrelated to each other (Landauer et al., 1998). Its computational power, ability to deal with synonymy and polysemy, and minimization of human interference are strong points in LSI's favor.

This project is a test of whether LSI can be applied to fiction corpuses with the same degree of success that it handles non-fiction. The results of previous experiments with LSI "demonstrate close resemblance between what LSA extracts and the way peoples' representations of meaning reflect what they have read and heard, as well as the way human representation of meaning is reflected in the word choice of writers" (Landauer et al., 1998, p. 4). Perhaps the most important difference between fiction and non-fiction in this case is that non-fiction attempts to be very clear and consistent in its word choice, especially for technical subjects. Fiction, on the other hand, makes use of metaphor, irony, allegory, and other techniques to discuss themes without necessarily naming them explicitly. This will be an extra challenge in fiction corpuses

but it is hoped that LSI will handle fiction as well as it does non-fiction due to its skill regarding synonymy and polysemy.

3.2 A Corpus of Short Stories

In this project, LSI was tested on a corpus of 44 short stories, a full list of which can be found in Appendix C.

The short stories in the corpus covered a range of time periods and countries of origin. The stories were chosen because they were available in both English and Norwegian. The LSI computations were run on the English versions because it is a more standardized language and many freely available language analysis tools exist. Norwegian editions of the stories were required because the focus group members, who were from Norway and contributed to the evaluation process, also needed to read the texts and preferred to do so in their native language.

It is of course debatable whether having the focus groups read in one language and running LSA on the same texts in a different language had an effect on the results. A poor translation can change the meaning of the original text to something other than what the author intended. This could potentially have an impact on which themes one discovers in a story, though it is unlikely to change the genre significantly. However, assuming the translations are good enough, it should not make a difference in which language the text was read or analyzed.

Short stories were chosen for this project because a wider range of genres could be represented without the matrix becoming unmanageably large. In order to maintain the same size and variation in a corpus comprising of novels, each document would have to be restricted to the length of a single chapter.

Another advantage to using short stories in this case was that their plot was contained within the length of the text. A chapter from a novel is placed within the context of the rest of the story and thus has plot, themes, and character development that carry on after the chapter's end. For example, consider the opening chapter of *Treasure Island* and that of *The Count of Monte Cristo*. If one were to compare these chapters,

the books might seem quite similar since both refer to adventures at sea. However, the rest of *The Count of Monte Cristo* focuses on Edmond Dantès' quest for revenge while *Treasure Island* is about pirates and lost treasure. Since short stories are contained, coherent entities we can draw conclusions about the entire story in a way that would not be possible with isolated chapters from novels. In summary, short stories were chosen for the corpus because their conciseness created a manageably sized matrix with a range of genres and themes while still being coherent entities.

It is noted that Saarti (1997) considered short stories to be serious fiction and that serious fiction was often difficult to classify. However, short stories still have to be about something and genre is based on what the story is about, in addition to how it is written. Therefore, though it may be more difficult to determine the genre of a short story than a work of genre fiction, where plot and style can seem standardized, it should still be possible to identify each short story's genre and themes. The genres chosen to describe the short stories may not be typical genre fiction categories, but identifying typical genre categories is not the goal of this project. The goal is to determine how similar the documents are according to what they are about and short stories should still be suited to this purpose.

Having established why short stories were used it should also be discussed how they were represented in the corpus and consequently the term-document matrix. Harman (1994) presented a discussion about what should constitute a document or term. A document can range in length from a paragraph to an entire book. Harman, who would prefer to divide a book up into chapters, does not recommend the latter. Choosing terms is not without its complications either. Terms can be single words, phrases, or sentences depending on the goal. In this corpus, it was decided that each short story, being relatively short in length, should be a document and that each word a term.

In summary, the fiction corpus that were used for these initial tests of LSI was comprised of 44 short stories that were chosen because they were available in Norwegian and English and should be able to give some insights into genre. Each short story was considered an entire document and each word a term in the term-

document matrix. Throughout the next sections, the process of preparing the corpus for the SVD transformation will be presented and the surrounding theory discussed.

3.3 Corpus Preparation

The first step in preparing the corpus was to acquire a digital version of each of the 44 short stories. Some of the short stories were available online through sources like Project Gutenberg (http://www.gutenberg.org/). Others were available in printed form in the Tromsø public library collection. These printed texts were scanned page by page and the full text extracted using Adobe Acrobat Pro's Optical Character Recognition (OCR) function. All of the digital texts were proofread for text recognition errors.

With all of the digital texts in place, metadata was removed from the digital texts. This meant removing title, author, chapter numbers, etc., which was a simple matter of manual deletion. Metadata removal was done so that the focus would be on the texts themselves, rather than author, title, etc.

The next step in the corpus preparation process was to reduce the texts to a list of words. To do this, a macro was created in Microsoft Word to accomplish the following:

1. Make all letters lowercase
2. Remove paragraph marks
3. Remove all punctuation[7]
4. Remove numerals[8]
5. Remove extra spaces

The Visual Basic code for this macro can be found in Appendix D.

[7] Even when one wants each word to be a term, dividing up the text based on punctuation and spaces is not entirely simple. Some words require punctuation for meaning, e.g. F-15, which is the name of an air superiority fighter. The process implemented here was not perfect but served for these initial experiments into a new field of research.

[8] Harman (1994, p. 250) noted that keeping numerals can nearly double index size but in some cases are necessary for meaning, like "what were the major breakthroughs in computer speed in 1986". This is probably more important for non-fiction than fiction.

With each text consisting of a simply formatted list of words, the corpus was collected into a single text file with a paragraph mark separating each document. It was now ready for LSI processing and testing.

3.4 The LSI Process

Perhaps the most important part of LSI was the singular value decomposition of the matrix. Before that step was taken however, the corpus went through a number of stages that aimed to give the terms their proper weight. First, the term-document matrix was created based on word frequencies using the bag of words model. Stemming or lemmatization could then be applied. Stopwords and singletons were removed if so desired. Next, term weights were adjusted for factors like normalization and term frequency – inverse document frequency. After these steps, the matrix was factored using singular value decomposition and the resulting matrices reduced to a certain number of dimensions. The final stage was to compare the documents using cosine similarity.

In the tests run on the short story corpus, the term weighting stages were applied in various combinations. All tests used normalization. Since this is, to the author's knowledge, the first time that LSI has been tested on a fiction corpus it was important to discover which term weighting techniques were most applicable. It was not a given that fiction would behave the same as non-fiction when analyzed using LSI. Also, since finding the correct number of dimensions to reduce the matrices to is a matter of experimentation, each test was run for the following values of k: 2, 5, 10, 15, and 18.

Test #	Stemming	Stopwords	Singletons	TF-IDF
1	Porter	List A	Does not remove	Yes
2	Lemmatize	List A	Does not remove	Yes
3	None	List A	Does not remove	Yes
4	Porter	List A	Removes singletons	Yes
5	Porter	List A	Removes singletons and doubletons	Yes
6	Porter	List C	Removes singletons	No
7	Porter	List B	Does not remove	Yes
8	Porter	List C	Does not remove	No
9	Porter	None	Does not remove	Yes
10	Porter	List C	Does not remove	Yes

Table 1: *Overview of LSI Tests*

As noted earlier, Gensim was used for the LSI testing phase because it included many useful tools related to this process like term weighting, word removal, SVD, and cosine similarity calculation. Gensim is implemented in Python (version 2.7, http://www.python.org/) and is dependent on Numpy (http://sourceforge.net/projects/numpy/) and Scipy (http://sourceforge.net/projects/scipy/).

An example of coding process in Gensim/Python is found in Appendix E.

Each step of the process and relevant theory will now be discussed.

3.4.1 Bag of Words

LSI uses the bag of words model to create the term-document matrix. A real world representation of the model would be a bag, containing each word as many times as it appears in the text, that is shaken up. The model ignores word order but accounts for term frequency and is used because "it seems intuitive that two documents with similar bag of words representations are similar in content" (Manning et al., 2008, p. 107). Two similar bag of words representations are considered to represent similar documents because it is assumed that words repeated often throughout a text give a good indication of what the document is about.

At the same time, a remote chance exists that two documents with different meanings, but the same bag of words representation, could be found in the same corpus. Take for example the following sentences:

1. Our garden is in front of our home but not on top of our home.
2. Our garden is on top of our home but not in front of our home.

Semantically these sentences have two different meanings, based on where the garden in is relation to the home. In the first sentence we can imagine a suburban house with plenty of garden space surrounding it and in the second perhaps a rooftop garden like those found in New York City and other densely populated metropolitan areas. Here word order has an effect on meaning and the two sentences should be considered to be different. At the same time, both are referring to gardens and homes and therefore have quite similar subjects. This example shows how the bag of words model can lead to misinterpretation of exact meanings but can still connect documents by subject based on term frequency. As it is unlikely that two documents with the same bag of words representation would occur together, especially as documents get longer, the strengths of the bag of words model overcome its potential drawbacks.

After using the bag of words model to create the term-document matrix for the fiction corpus the LSI process continued with stemming or lemmatizing the terms list.

3.4.2 Stemming and Lemmatization

Words take on many forms that have quite similar semantic meanings. For example, *lock*, *locks*, and *lock's* can all be combined under the root term or stem, *lock*. By using a set of rules to combine these semantically similar forms, the terms list becomes shorter and more relevant connections are made between the documents, which can lead to improved query results. These rules, or algorithms, can take the form of stemming or lemmatization.

According to Manning et al. (2008, pp. 30-31), "stemming usually refers to a crude heuristic process that chops off the ends of words … and often includes the removal of derivational affixes." The first stemmer was made by Lovins in 1968 and used a dictionary of 294 suffixes and complex context rules to determine when suffixes could be removed. Porter's stemming algorithm, introduced in 1980, is much simpler than Lovins' and is one of the most popular English language stemmers. The algorithm followed five steps, which handle plurals and past participles, derivational suffixes, and a final recoding, based on a the idea that each word's stem has to consist of a certain number of vowels and consonants (Willett, 2006). As seen in the implementation in Gensim, the result of the Porter stemmer would sometimes represent a different concept or not be an English word, e.g. *pony* becomes *poni*. Representing a different concept can be problematic but changing *pony* to *poni* does not have a significant impact since humans will not be reading the stemmed term list in order for the process to proceed. The number of times that the stemmer combined words with different meanings should be small and not have a significant effect on performance. In most of the tests to be done here, the Porter stemming algorithm will be used.

In those tests where Porter stemming is not used, lemmatization will be implemented. Lemmatization is the other option for combining different word forms according to the root term. It is a more complex form of stemming that is based on a full morphological analysis, which includes analyzing grammar and context. Warner (1994) would probably have supported lemmatization over stemming because of the need she saw for full grammatical understanding of full text documents in an information retrieval context. The goal of the morphological analysis is to find the root word, or lemma, that one would look up in the dictionary. For example, the

lemma of *better* is *good.* A stemming algorithm would not make the connection between *better* and *good*, especially not if it is as heuristic as Manning generalizes. Though more difficult to design and implement, lemmatization has the potential to find and combine more variations of root terms, which will result in a shorter terms list and most likely improved performance.

Both the Porter stemming algorithm and a lemmatization algorithm, as they are implemented in Gensim, were used when testing LSI on the fiction corpus. The next step in the LSI process was to remove stopwords from the terms list.

3.4.3 Stopwords

Documents use words that do not have a significant, direct impact on the document's content or message. These terms, *stopwords*, are very useful for sentence structure, grammar, and more but taken out of context do not give an indication of what the document is about. Stopwords occur with a high frequency in most documents, are thus of minimal use in comparing document subjects/topics/genres, and can be removed from a terms list without a major impact on performance. Determining which terms are stopwords, however, is a slightly more difficult task.

Dolamic and Savoy (2010) tested the effectiveness of removing stopwords in English, French, Persian, and Hindi corpuses. According to the results, removing stopwords generally improved query performance levels. At the same time, a short stop list of nine words was seen to produce similar results to a longer one of 571 words. This hinted that removing stopwords does have an effect but that it may not be as significant as expected.

It is noted that stopwords could be removed with little to no impact on keyword searches but their removal could have an effect on phrase searches. A classic example is Shakespeare's "To be or not to be" from Hamlet. A keyword search would probably query for *Shakespeare* and *Hamlet*; terms that are very unlikely to be found on a stop list. However, in order to search for the phrase "To be or not to be" one has to use four words that could be considered stopwords. If they were removed from the corpus vocabulary, such a phrase search would be impossible. Therefore a careful

balance has to be negotiated between removing words that seldom give the text meaning and removing words that could be used in a query.

In some of the tests of the LSI process on the fiction corpus, stopwords were removed. The tests made use of three different stop lists; List A, List B, and List C. List A was a combination of common stopwords (The Information Retrieval Group) and English contractions (Nordquist, 2013). Lists B and C were based on word frequency lists for modern fiction (Wiktionary, 2008). List B took the top 300 words from the list and List C used the selection of these words that appeared in 39 or more of the 44 corpus documents (≥88.6% document frequency). The lists can be found in Appendix F.

Stopwords were not removed in every test since it was also desirable to determine whether term weighting could replace stopword and singleton removal. Singletons will be discussed in the following section and will be followed by a presentation of term weighting.

3.4.4 Singletons

In addition to stopwords, singletons can also be eliminated from the terms list. The goal is to remove unnecessary terms and thereby reduce the size of the matrix. A singleton is a term that is used one time in one document.

Like removing stopwords, removing singletons has advantages and disadvantages. One advantage to removing singletons is that the size of the corpus is reduced, which can have a significant impact on computation time and required resources. In this particular fiction corpus, removing singletons reduced the terms list by 48.5% (4,687 terms). Removing doubletons, words that appear twice in the corpus, reduced the terms list by an additional 14.9% (1,437 terms). By removing these infrequently used terms, documents are no longer as different, which can have a positive impact on determining which documents are similar. On the other hand, Lochbaum and Streeter (1989, p. 670) noted that singletons could be used in queries and that their removal

decreases the chances of such a query's success.[9] These points should be kept in mind when evaluating whether to remove singletons from a term-document matrix.

After singletons were removed, term weighting could be applied to the matrix.

3.4.5 Term Weighting

In a term frequency count where weight is directly correlated to frequency, a word that is used eight times in a document would have a weight of eight. This does not necessarily imply that this word is four times as important as a word used only twice, though the term used eight times is bound to be more important to a certain degree. Term weighting is a process that attempts to balance the weights of all terms in a document in relation to each other and their overall importance for that document.

Many different kinds of term weighting exist. Salton and Buckley (1988) discussed how early attempts at term weighting involved complex processes using statistically related terms, term phrases, thesauruses, and knowledge bases to automatically determine term dependencies. Erenel and Altincay (2012) explored non-linear term weighting and come to the conclusion that logarithmic weighting is most effective when term frequencies are high.

Another option for approaching genre-based, thematic term weighting in a fiction corpus would be to make a dictionary of words commonly associated with each target genre and give these words more weight in the term-document matrix. Making such a dictionary would probably require a lot of human involvement, whereas the goal of this project is to find an automatic algorithm that necessitates little human involvement. Another disadvantage to thematic term weighting is that the genres would have to be identified beforehand and this would probably lead to static, broad genres rather than the flexible, precise genres that LSI should be able to compute. None of the above techniques were used in this project and will not be examined further.

[9] One option to overcoming the difficulty that Lochbaum and Streeter point out would be to remove the singletons but have them available in an index that the search engine could access. Creating a functional search engine is outside the scope of this project but the idea could be used in future research.

Today one of the most popular term weighting techniques, and the one used throughout these LSI experiments, is Term Frequency – Inverse Document Frequency (TF-IDF), which was introduced by Spärck Jones in 1972. The TF-IDF algorithm compares how often a term is used in a document to how many documents in the corpus use that term.

$$w_{t,d} = TF_{t,d} \cdot IDF_t$$

Equation 1: *Term Frequency – Inverse Document Frequency*

The above equation states that the global weight is equal to term frequency per term in a document multiplied by the inverse document frequency for a term, thus combining precision and recall. The recall component is a simple count of term frequency. The inverse document frequency component is what gives TF-IDF its precision, as shown in the equation below.

$$\log\left(\frac{\#\ documents\ in\ collection}{\#\ documents\ using\ term}\right)$$

Equation 2: *Inverse Document Frequency Component*

The result of TF-IDF is that more weight is given to terms used often in few documents, less weight to rare terms in a document or terms used in many documents, and the least weight to high frequency terms found in many documents (Manning et al., 2008, p. 109).

TF-IDF lowers the weight of terms that occur frequently and those that seldom appear. As discussed earlier, stopwords are terms used in most documents and singletons used once in a single document in the corpus and part of the experimentation in this project tested whether TF-IDF could replace stopword and singleton removal.

A different form of term weighting is normalization.

3.4.6 Normalization

Normalization can mean one of two things in search and retrieval; 1) term normalization, which is the removal of superficial differences between equal terms (for example *USA* should be equivalent to *U.S.A.*) or 2) term frequency normalization, which makes the each term's weight proportional to the length of the document (Manning et al., 2008). In this project term normalization was treated as part of the stemming and lemmatization processes. Term frequency normalization, hereafter referred to simply as normalization, was used throughout all experiments.[10]

Normalization is the adjusting of vector lengths to account for how often a term is used proportionally to its document's length. For example, a journal article about dogs may use the term *dog* the same number of times as a chapter in a book about mammals in general. Though a simple word count would imply that *dog* is equally relevant to both the article and the book, it is proportionally more significant in the article. In order to adjust for the varying length of documents, normalization is applied.

The most common method for applying normalization in the vector space model is cosine normalization. The normalization factor is calculated using the TF-IDF weights for each document (Singhal, Buckley, & Mitra, 1996).

$$\sqrt{w_1^2 + w_2^2 + w_3^2 \cdots + w_t^2}$$

Equation 3: *Cosine Normalization*

It was not expected that normalization would have different effects on fiction and non-fiction corpuses. Therefore, since the corpus documents in this project did vary in length, normalization was applied in all experiments.

[10] It is noted that normalization should in theory not have any effect on cosine similarity. Normalization only affects the length of the vectors and cosine similarity measures the angles between them. Should future experiments compare document vectors using techniques other than cosine similarity, they should also test for the effects of normalization.

The goal of the previously discussed stages was to prepare the term-document matrix and adjust the terms list and corresponding term weights. The next step was to apply singular value decomposition.

3.4.7 Singular Value Decomposition

What makes LSI unique, and therefore perhaps the most important stage of the process, is the singular value decomposition of the normalized, weighted matrix. This stage is the one that does the most to remove "noise" from the matrix and as a result discover the higher-order, semantic relationships between documents.

At the same time that noise is removed by reducing the number of dimensions it is important that information is not lost. To explain SVD, Geiß (2008) gave an example of a taking a picture of fish in an aquarium. The three-dimensional aquarium is reduced to a two-dimensional picture while at the same time a good photograph would still show all fish in the aquarium. Thus dimensional space has been reduced while maintaining a good representation of the information, in Geiß's example the fish.

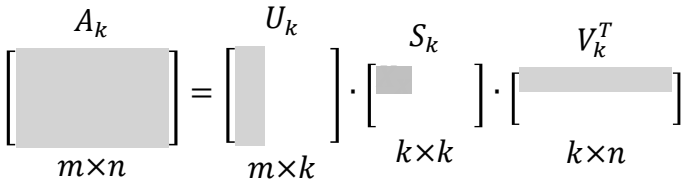
Berry (1992, p. 5) formulated the result of SVD as "terms that do not actually appear in a document may still be used as referents, if that is consistent with the major patterns of association in the data." In other words, SVD can identify documents A and B as similar though their vocabularies are different, so long as enough other documents use words from A and B's vocabularies together. This is especially helpful when analyzing fiction, since a story about love need never use the word explicitly. So long as there is enough shared vocabulary between similar documents, it will not matter than certain documents use a limited portion of that vocabulary.

SVD begins by factoring the term-document matrix, A, into three matrices, which represent term-by-concept, concept-by-concept, and concept-by-document, as shown in the following equation.

$$[A] = [U] \cdot [S] \cdot [V^T]$$

Equation 4: *Factoring the Term-Document Matrix*

The dimensions of the factored matrices are then reduced to a certain amount, k. This results in a U matrix with $m \times k$ dimensions, a S matrix with $k \times k$ dimensions, and a V^T matrix with $k \times n$ dimensions, where the original A matrix was $m \times n$. The reduction of the matrices looks like this, where retained columns and rows are highlighted:



Equation 5: *Reduction of the Term-Document Matrix*

Finding the best value for k is a matter of experimentation (Lochbaum & Streeter, 1989). A well-chosen value, generally between 200 and 500 for a matrix with thousands of documents, will result in terms with similar meanings being moved closer together and those with dissimilar meanings remaining far apart (Ampazis & Perantonis, 2004). This occurs because matrix cells that were originally filled with zeros are now non-zero, with values that draw similar words together in the vector space.

Though Ampazis and Perantonis recommended reducing the term-document matrix to 200-500, the matrix used in this project, which consists of only 44 documents, was simply not large enough for this kind of reduction. Instead testing was done when k was equal to 2, 5, 10, 15, and 18 since these values were proportionally similar to the reductions done by other researchers. Eighteen was chosen as the largest number of dimensions because it already reduces the matrices by more than half. The experiments by Landauer et al. (1998) also reduced U, S, and V^T by significantly more than fifty percent so $k = 18$ seemed reasonable for this corpus. The minimum was set to $k = 2$ because it was hypothesized that it would remove too much noise from the matrix and could thus be used for comparison. It is important to experiment with different values of k since, as Geiß (2008) noted, too little reduction means leaving significant amounts of noise in the matrix but if one reduces too much the latent semantic structure will be destroyed.

The SVD phase concluded the LSI processing of the term-document matrix. In order to test which documents are similar, their cosine similarity was calculated. This process is described in the next section.

3.4.8 Cosine Similarity

In this project cosine similarity, which measures the angle between vectors, was the method chosen to compare documents and terms in the vector space model. Cosine similarity is most commonly used in a search and retrieval context to compare queries with documents and terms from the corpus. In this paper it will be used to compare the documents to each other, as if each document were a query.

The result of the cosine similarity calculation is a value between 1 and -1. Vectors that are similar will point in more or less the same direction (an angle close to 0º), thus the cosine of the angle between them would be closer to 1. The cosine of the angle between vectors that are diametrically opposed is -1.[11] Once the cosine similarity between documents is determined, we can rank them. The actual calculation is done using the inner product, alternately called dot product or scalar product, where vectors a and b are being compared.

$$a \cdot b = \| a \| \| b \| \cos\theta$$

Equation 6: *Cosine Similarity*

The inner product will differ from one pair of documents to the next, which presents an easy way to rank their similarity.

Gensim included a tool for calculating the cosine similarity between all documents and the tables produced by this tool served as the basis for evaluating the effectiveness of the test. An evaluation of the document similarities will be presented in the following sections.

[11] Cosine similarity can be bounded to positive space, meaning that results would range from 0 to 1. Most results found using Gensim ranged from 0 to 1 though on occasion negative similarities also occurred.

4.0 Evaluation

The goal of this project was to test whether LSI could serve as the basis of a new, automatic way of comparing works of fiction. In order for the research to be considered a success, it needed to be both relevant and produce results similar to how the readers themselves would have compared the documents in question. Though "information systems cannot be designed to serve each and all individual user's many different projects and purposes" (Hjørland & Albrechtsen, 1999, p. 134), perhaps LSI can mimic the general consensus of the reading public. To test this idea, two focus groups were asked to read a selection of the short stories in the corpus and sort them into genre-like groups. The data produced during this process will be used to evaluate the relevancy of the LSI algorithm.

After the focus group results have been presented, they will be discussed in light of the methods used in the LSI process and an evaluation given. This evaluation will especially focus on the relationship between word choice and genre.

4.1 Using Focus Groups to Evaluate LSI

As mentioned above, it would be difficult to create an information system that would reflect everyone's needs and opinions. At the same time, the system needs to be grounded in reality and represent the opinions of fiction readers. In order to do this, it was therefore decided to use focus groups comprised of people who like to read fiction.

4.1.1 Background Theory

Kitzinger and Barbour (as cited in Barbour, 2007, p. 2) defined focus group as "any group discussion … as long as the researcher is actively encouraging of, and attentive to, the group interaction." Here the focus group model was implemented as a group discussion. The focus centered on the results rather than the process because, though it is beyond LSI's capabilities to mimic how the group got to their consensus, it would hopefully come to a similar conclusion.

As it is beyond LSI's capabilities to mimic the discussion process, it is also impossible for it to reflect each individual's opinion. Focus groups were chosen as the qualitative research model instead of individual interviews in order to gain an understanding of fiction readers' opinions in general, rather than each individual's interpretation.

An advantage to using focus groups for data collection is that they are helpful in drawing out opinions from people who might find one-on-one interviews intimidating (Barbour, 2007). For this project, it was necessary for focus group members to read the texts before gathering to discuss them. The discussion process could have made participants feel that they were under pressure to do well and get the right answers. In a group, attention is not placed so heavily on each individual, as it would have been in a one-on-one interview. Thus being part of the group can give a better sense of security for suggesting answers and discussing various opinions.

According to the theory discussed by Brinkmann, Tanggaard, and Hansen (2012), there are three main forms of focus group discussions; open, stringent, or a mixture of the two. The goal of the open model is to analyze how the discussion progresses, whereas the stringent model's goal is to find answers to specific questions or solve certain tasks. The focus group discussions in this project perhaps fell more into the stringent category than the open or combination models. The goal was not to analyze how the focus groups discussed their way to the genre categories. Instead, the goal was to come to a consensus that would serve as the basis of evaluation of the LSI algorithm's application to fiction.

An alternative to using focus groups would be to base criteria on popular genre categories for fiction. In that case the research of authors like Saricks (2001) and Harrell (1996) would have been relevant since they have studied which categories libraries use for their fiction. It would take a much larger test than the one done here to determine if broad categories emerged that reflected these popular fiction categories. In order to produce a good experiment, it would require using a corpus of popular novels that traditionally fall into specific categories or attempt to categorize the corpus texts oneself. The former might have positive results but it is doubtful that the latter would.

Another alternative to using focus groups would have been cluster analysis. Cluster analysis would be a fully automated process, which is the goal of classification-by-algorithm. At the same time, criteria for evaluation success should come from user preferences and it was decided that focus groups be used for this purpose. Therefore, in addition to the reasons noted above, it was decided to create a corpus from scratch and rely on focus groups to analyze and categorize the texts.

As will be shown below, the focus groups were small but representative of a large proportion of library users. Since the focus groups were not completely representative of all library users and their total numbers were small, the results of their analysis should not be considered universal. The results are indicative but it would be recommended that future renditions of this research make use of a larger and more diverse population of participants.

4.1.2 Focus Group Demographics

Two reading circles, with a total of 12 members, were found through contacts at Tromsø library that agreed to serve as focus groups for this project. Both focus groups, hereafter referred to as FG-1 and FG-2, consisted of middle-aged, Norwegian women who used the library occasionally and read mostly fiction. To be more specific, FG-1's members were all between the ages of 41 and 65 and FG-2 had a few members under the age of 40 but the rest were 40+. All focus group participants were library users who visited the library at least several times per year, except for one member who worked full-time at the library. Being semi-regular users, most of the participants felt that they were at least somewhat acquainted with the library's collection and how it was organized. Based on this data we can say that the focus group members represent a significant proportion of Tromsø library users, as seen in section 2.3, though not all demographic groups.[12]

[12] Should focus groups be used in future research, an effort should be made to include all demographic groups. It is especially important to include male readers of whom there were none in FG-1 and FG-2.

As in the user survey presented earlier in this paper, focus group members were also asked how they would prefer the library fiction collection to be organized. Over 75% responded that they would like the entire collection to be divided according to certain criteria. Though this was also the most popular answer in the user survey, the percentage is much higher among the focus group members. When asked what the criteria should be for dividing up the fiction collection, members of FG-1 divided their answers equally between literary form, genre, and language. FG-2 showed a slight preference for literary form over genre. In the user survey we saw that the most popular way to divide up the collection was genre, with literary form coming in second. Thus the focus groups show slightly different, yet still similar, opinions to those found among the survey respondents. Assuming that the survey provides a good representation of Tromsø library users in general, we can say that the opinions expressed by the focus group members are also mostly representative of the same population.

Some of the advantages that these focus group members provided were that 1) they represent one of the largest library user groups, 2) the members already knew each other and enjoyed discussing literature together, 3) they met regularly to discuss fiction, which made it easier to arrange times to work on this project, and 4) they were enthusiastic readers already well acquainted with a range of fiction genres.

4.1.3 Reading Assignment

Each focus group member was asked to read approximately 150 pages worth of short stories. All members of FG-1 received the same stories, 18 of the total 44. FG-2 was slightly larger than FG-1 so it was decided that members would receive different collections of stories to read so that between them, all 44 stories would be read.[13] Six of the short stories were common for all members of FG-2 and the rest were divided so that 2-3 participants would read each story.

After an initial meeting in November 2012, the focus groups were given approximately four months to complete their readings. For each short story the

[13] Not all stories ended up being read and discussed due to illness, other engagements, etc.

participants were asked to write down 5-10 keywords that described that story's genre. The goal of this was to help them to think about genre while reading the stories and to serve as a memory aid when it came time to discuss the stories with the other members of the group.

4.1.4 How the Focus Groups Discussed the Readings

In February/March 2013, the focus groups met again to discuss what they had read. In these discussions it was emphasized that there was not a "correct" answer and that all opinions were valid.

To aid in the discussion process, focus group members were introduced to the idea of cluster maps and asked to arrange the short stories in a similar way. It was explained that the goal was to create groups of stories with similar genre traits and to arrange these groups in relation to each other. This process was accomplished through the use of notecards, one per short story, on which was printed the author and title. These notecards also had room for six keywords and one main topic.

The first step in creating the short story cluster map was to discuss the genre keywords participants had written down when reading the stories on their own. Each story was discussed individually. A focus group member was assigned to write down on the corresponding notecard the keywords that the group decided were the most important for each story. Either afterwards or during the course of the discussion, the notecards were arranged on a table in a continually evolving cluster map.

The graphical representations of the focus groups' results shown below will be discussed in the next section.

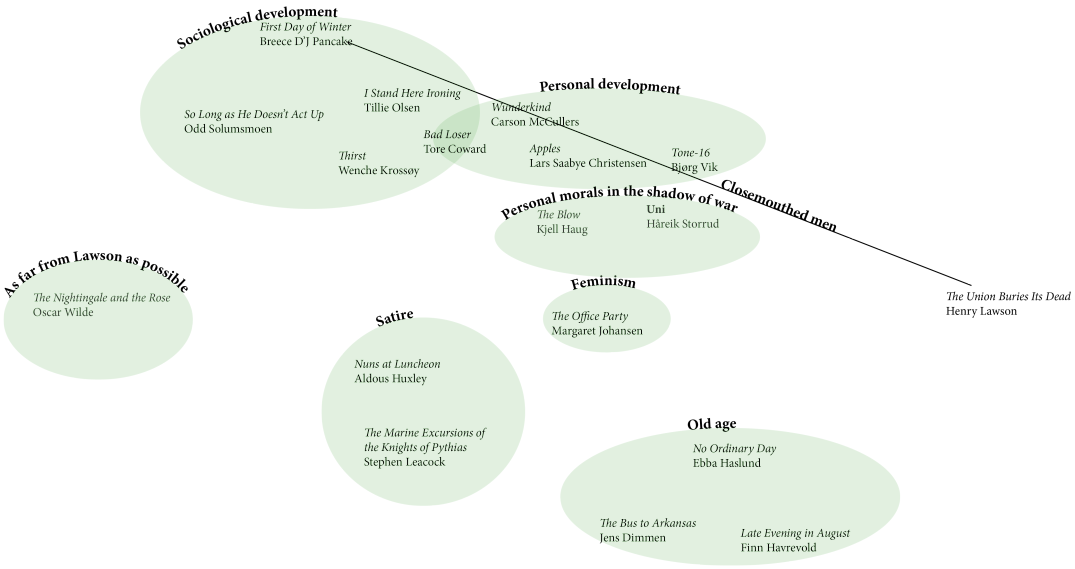


Figure 1: *Cluster Map from Focus Group 1*

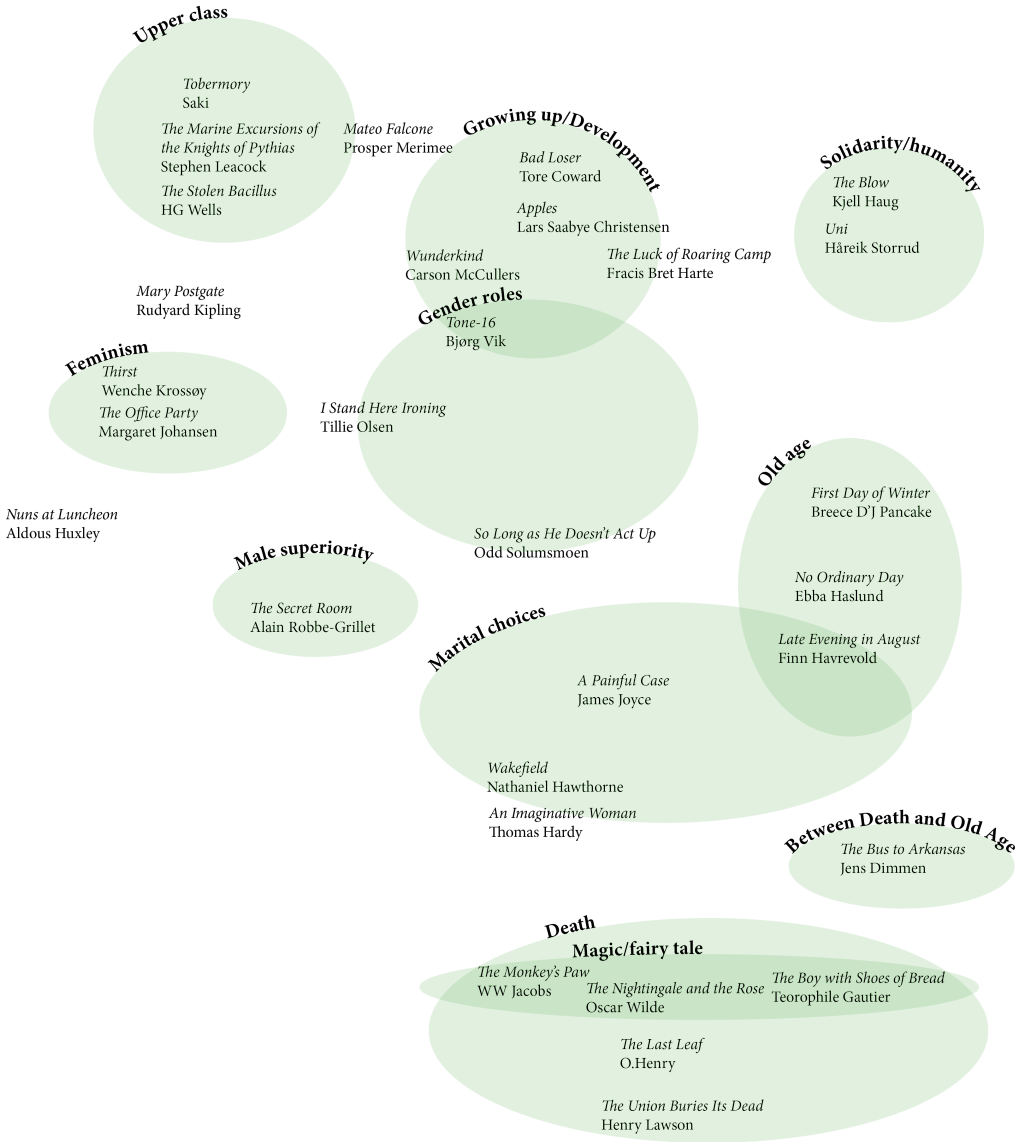


Figure 2: *Cluster Map from Focus Group 2*

4.1.5 Presentation of the Focus Groups' Cluster Maps

The focus groups created cluster maps for slightly different collections of short stories, where FG-2 had a larger collection that also encompassed all the stories read by FG-1. Since FG-2 had more information to analyze, their categories were necessarily different than FG-1's but remarkably still quite similar. Below are the classes into which each focus group sorted their short stories.

Focus Group 1	**Focus Group 2**
Sociological development	Upper class
Personal development	Growing up/development
Closemouthed men	Solidarity/humanity
Personal morals in the shadow of war	Gender roles
Feminism	Feminism
Satire	Male superiority
Old age	Old age
As far from Lawson as possible	Marital choices
	Death
	Magic/fairy tale
	Between death and old age

Table 2: *Short Story Categories*

It is interesting to note that three categories cross over between the two lists. Both groups classified stories under *old age*, *feminism*, and *development*.

Within these categories some stories were common for both groups and others not. For example, both groups categorized Johansen's *The Office Party* as *feminism*. Krossøy's *Thirst* was *feminism* in one group but the other focused more on the sociological development aspects and categorized it accordingly. Those stories that FG-1 placed under *personal development* were also categorized under *development* by FG-2. However, like *Thirst,* those in *sociological development* were removed from FG-2's *development* category and other aspects emphasized. Haslund, Havrevold, and Dimmen's stories all had the category *old age* attached to them by both groups. In the case of Dimmen's *The Bus to Arkansas*, the story was given its own category called

between death and old age since FG-2 thought that both aspects were equally important. Since both groups had similar categories it hints that a wider population might also come to the same conclusions.

Though there was similarity between the two groups, FG-2 had more crossover between categories than FG-1. Kipling, Hardy, Vik, and Dimmen's stories belonged to several groups and thus wholly to none. Kipling's *Mary Postgate*, for example, showed aspects of both *feminism* and *upper class*. Since nether dominated over the other, both genres were considered equal and Kipling was placed between them accordingly. It was not an easy task to determine the proper categories that would avoid crossover without each story being the only entity in its group. The crossover shows how important it is that all aspects of a story be considered when categorizing, rather than forcing it into one group or another.

Another interesting aspect of FG-2's analysis was that the *death* category had a subcategory called *magic/fairy tale*. The three stories placed in this subcategory certainly had elements of death, whether it was the son or the bird that died, but each had magical elements relating to superstition, fairy tales, or religion. Regardless, the group decided that while death was the main aspect that bonded the stories of Jacobs, Wilde, Gautier, Henry, and Lawson together, it was important to note the magic associated with Jacobs, Wilde, and Gautier's stories. Similar to the crossover mentioned above, subclasses show how difficult it is to force stories into broad, predetermined categories.

FG-1's analysis also showed a case of what could be called an anti-category, *as far from Lawson as possible*. In this case the group decided that the determining factor for placement of Wilde's *The Nightingale and the Rose* in the cluster map was its relationship to Lawson's *The Union Buries its Dead*. The relationship was not identified by its similarity but rather its dissimilarity. The group could not decide whether Wilde's story was similar to any others but were convinced that someone who liked it would probably not be interested in Lawson's story and vice versa. Therefore the stories were placed at opposite ends of the cluster map. In terms of cosine similarity, the group would probably say that their relationship was -1, in other words that the vectors were diametrically opposed.

The arrangement of the short stories by the focus groups was used for evaluating the LSI process. This process is detailed in the following section.

4.1.6 Focus Group Data Determines Precision and Recall

The focus group's results were combined using Gephi (http://gephi.org) and are displayed in the figure below.

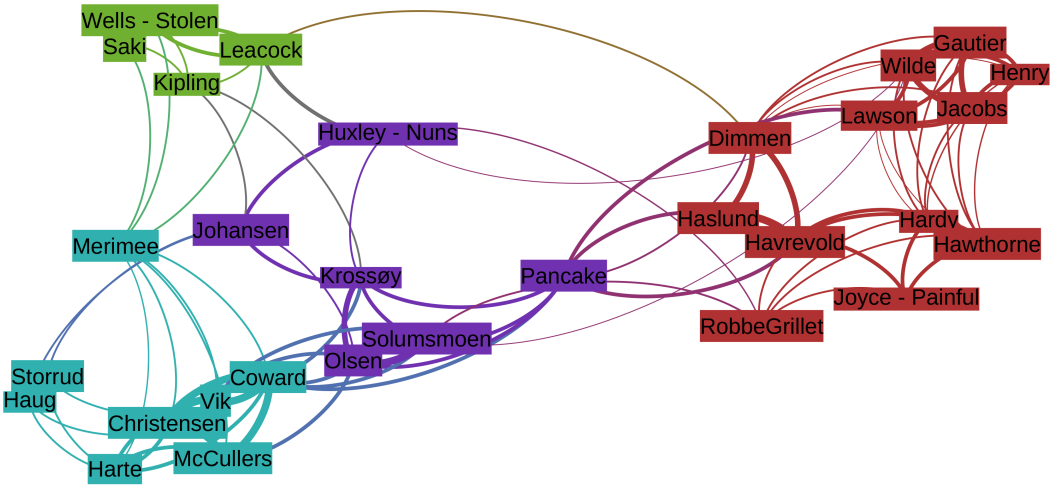


Figure 3: *Combined Results from Focus Groups*

From the combined data four categories of short stories emerged: *upper class*, *(personal) development*, *societal roles (and attempts to change them)*, and *age and death*, as marked with green, blue, purple, and red respectively.

These four categories were used for evaluating the cosine similarities between the short stories. Deerwester (1990) also evaluated his tests using relationships that users would have considered relevant. In this case, stories were considered a match if they fulfilled two criteria: 1) both belong to the same category (*upper class*, *development*, *societal rules*, or *age and death*), and 2) they have a cosine similarity that is greater than or equal to 0.8.

Neither of these criteria are perfect but instead should be taken as indicative of performance. The first criteria for example, does not allow for relationships that are

strong yet crossover between groups. For example, Coward had a close relationship to Olsen, Solumsmoen and Pancake, as seen by the thickness of the lines connecting them in Figure 3. At the same time the color coding, as determined by Gephi's modularity ranking (see http://wiki.gephi.org/index.php/Modularity), separated Coward into the blue group while the other stories had the highest statistical similarity to the purple group. The second criterion places high demands on the performance of the algorithm, perhaps too high for these initial tests of LSI on fiction. Also, it is noted that for lower values of k the vectors would be forced closer together simply because of lack of dimensional space for them to spread out in.[14] This resulted in artificially high cosine similarities. However, a cosine similarity of 0.8 can be seen as a goal since a fiction search engine will want to return top results first. Though these two criteria are perhaps too stringent, they do make it easy to evaluate the algorithm's performance and will therefore be implemented.

The tests run using LSI were evaluated according to their precision and recall. Both precision and recall were dependent on a large number of matches, as determined by the above criteria. The most successful tests had both high precision and high recall. Based on Manning et al. (2008, p. 5) the following equations for precision and recall were written. Precision examines the relationship between the number of matches and all documents retrieved by the system. Retrieval in this case was determined by having a cosine similarity greater than 0.8.

$$Precision = \frac{\#\ of\ matches}{all\ retrieved}$$

Equation 7: *Precision*

[14] For example, say that we have three documents that are completely dissimilar. Spread evenly throughout three dimensions the document vectors would be orthogonal to each other (one points along x, the next along y, and the third along the z-axis). In two dimensions, on the other hand, there is not as much room. In order to space the documents as far from each other as possible, one goes along the x-axis, one along the y-axis, and the third at 45º. This increases the average cosine similarity from 0 to 0.5. Though the original documents are just as dissimilar the reduction of dimensional space makes them seem more similar.

Recall examines the relationship between the number of matches and all the documents that are considered relevant. Relevance in this case was determined by both documents being in the same category in the combined focus group data.

$$Recall = \frac{\#\ of\ matches}{all\ relevant}$$

Equation 8: *Recall*

In Appendix G, precision and recall values are graphed for each test of the LSI algorithm. These values will serve as the basis of the discussion and evaluation that follows.

4.2 Discussion

In the following sections, the LSI process will be reviewed in light of the data collected from the focus groups. First the LSI experiment results will be presented and comments made about the applicability of the various stages. Afterwards the relationship between LSI and fiction will be discussed on a broader scale. Discussion points include how precision and recall could be improved, the relationship between word use and genre, and differences between approaching fiction and non-fiction.

4.2.1 Test Observations and Comments on LSI Process

To begin this discussion of the LSI tests some observations about best and worst cases will be presented. The test numbers referred to are found in Table 1, which is on page 38. Test #3 had the highest recall percentage: 78.33% in two dimensions. Test #8 had the highest precision percentage: 24.49% in eighteen dimensions. There is variation between all the tests as regards precision and recall performance but values were generally highest in two dimensions and lowest in fifteen dimensions. Compared to the average, Tests #6 and 8 performed the best in both precision and recall, except in two dimensions. With these cases in mind, let us discuss what each test was trying to accomplish and the implications the results might have for the application of LSI to fiction.

The discussion will follow the steps of the LSI process as detailed earlier in this paper. Some steps were the same for all tests and will not be discussed further here. Below is a tabular summary of what will be presented.

	Stemming & Lemmatization	**Stopwords**	**Removal of Singletons & Doubletons**	**TF-IDF** *(when # of dimensions is high)*
Best results	Lemmatization	List C	No removal	TF-IDF
	Stemming	List A	Singletons & Doubletons	
Worst Results	Neither	List B	Singletons	Singletons & Stopwords

Table 3: *Performance Summary*

The first step of the LSI process that was experimented with in these tests was stemming and lemmatization. Test #1 used stemming, Test #2 used lemmatization, and Test #3 used neither. Surprisingly there were only small differences between the results of these tests. The most significant difference was for recall in five dimensions where values ranged from 32.5% to 39.44%. Most tests had corresponding precision and recall values that were within 1-5% of each other. It was expected that Test #3 would have much lower precision and recall values than the other two since there would be more forms of each word.

Since there were no significant differences in the effect of stemming and lemmatization on the results, it is possible that they have less effect on fiction than non-fiction. However, many articles have defended the effectiveness of Porter stemming and lemmatization (see for example Conrado, Gutierrez, & Rezende, 2012; Porter, 1980; Willett, 2006) so it is more likely that their apparent failure in these tests has more to do with the size of the corpus than the techniques themselves. It is possible that longer documents would have resulted in more significant performance differences between these three tests. Each short story is 10,000 words or less, with one as short as 553 words. A corpus with longer and a larger number of documents may therefore have had the expected impact.

Another set of experiments was designed to determine the effect of stopwords on the results. Tests #2, 7, and 10 followed the same steps except that they used different stop lists. Test #2 had better precision in all dimensions tested and better recall in all dimensions except for fifteen. This is unexpected because the stop list used for Test #2 (List A) was based on non-fiction frequency lists and contractions whereas the stop list used in Test #7 (List B) was designed for fiction literature. List C, as used in Test #10, was a refined and shorter version of List B. Since List C was adapted for this corpus it was expected that it would have the best results overall. Test #10 had better precision than Test #2 in four cases and better recall three times. Test #10 also performed better than Test #7 regarding precision in eighteen and fifteen dimensions. Test #10 had better results in four of five cases with regard to recall.

Though it is difficult to draw any broad conclusions from so few tests, it does seem that List C showed the most promise. This hints that the most successful stop lists will be shorter and designed for a specific corpus. The latter conclusion seems fairly obvious; a system designed for something else should not perform as well. The former conclusion is supported by other stop lists, which can be less than 10 words long (Dolamic & Savoy, 2010).

The next step was to test the effect of singletons and doubletons on the corpus. Tests #1 and 4 were the same except that Test #4 removed singletons. Test #4 performed better, as regards both precision and recall, in five and ten dimensions. Test #5 also followed the same steps but removed both singletons and doubletons. The added effect of removing doubletons resulted in Test #5 performing better than Test #4 three times out of five for both precision and recall. Taken together, Test #1 performed the best, followed by Test #5, and Test #4 performed the worst overall. In other words, leaving singletons in the corpus had the best result but removing doubletons and singletons had better results than just removing singletons. It was expected that removing singletons would give better results than leaving them in but that removing doubletons would worsen results.

One of the reasons for removing singletons from a corpus is to have fewer terms that mark the documents as being different. At the same time singletons do not make connections between documents since they appear only once in the entire corpus.

Perhaps their role as filler has more of an impact than their role in differentiating the documents. If this is the case, it seems that they should be allowed to stay in the corpus.

The effects of TF-IDF were also tested. Tests #6 and 8 do not use TF-IDF and performed better than all other tests in ten, fifteen, and eighteen dimensions for both precision and recall. Test #9 had better recall and precision in five dimensions, though Tests #6 and 8 still had better results than all other tests. In two dimensions, Tests #6 and 8 had the worst results of all the experiments. Tests #6 and 9 were designed to test whether TF-IDF might be a replacement for stemming and singleton removal or vice versa.

From these results, it seems as though TF-IDF had a better, more pronounced effect when more dimensions were retained. When the matrix had more information removed, it seems that deleting singletons and stopwords produced better results. It is not the goal to have all documents be considered similar, which is the end result of removing too much information from the matrix. Therefore, it is not recommended that the removal of stopwords and singletons be used as a replacement for TF-IDF, as TF-IDF produces better results for more desirable values of k.

Through the discussion presented in this section the following conclusions were made:

1. Stemming and lemmatization had similar effects on the corpus.
2. Shorter stop lists that are designed for that corpus produce better results.
3. Leaving singletons in the corpus had better results than removing them.
4. Removing singletons and stopwords is not a replacement for TF-IDF.

Some of these results were expected and others were not. The variations in precision and recall provide the basis for these conclusions. At the same time, precision and recall performed poorly and the next section will discuss why this might be and how it could be fixed in future projects.

4.2.2 Improving Precision and Recall

The overall performance of precision and recall in these experiments will now be discussed on three counts: values, expected results for a full text database, and relationship to each other.

Lesk (2005, p. 210) wrote, "Many systems operate at around 50% recall and 50% precision; half of what is found is relevant, and half of what is relevant is found." These values are much higher than the test results presented above, except for certain tests where recall went above 60% in two dimensions. While it is encouraging that precision and recall are not expected to perform at 100%, values of less than 1% cannot be considered a success. Precision especially had low values compared to Lesk's expectations but according to Rasmussen, this might be an acceptable start.

Rasmussen (1994, p. 242) wrote that recall is higher for full text databases and lower in indexed file searches but the opposite is true for precision. Assuming that fiction behaves similarly to non-fiction, a corpus like the one used here, which is based on full text, should have high recall and low precision. On this count, therefore, precision and recall behaved somewhat as expected. It would have been interesting to use the same corpus to create an indexed file for comparison but that was outside the scope of this project.

The third count on which precision and recall should be evaluated is their direct relationship to each other. As noted above, recall should be proportionally higher than precision in a full text database. At the same time, the two should have an inverse relationship, meaning that as one increases the other decreases (Rasmussen, 1994, p. 241). In the tests presented here, recall increased as the number of dimensions decreased but precision stayed more or less constant throughout. This is not as Rasmussen predicted and it is difficult to say whether the direct cause is that the corpus was comprised of fiction or something else.

Recall and precision had both expected and a non-expected outcomes in these tests. First, recall was normally much higher than precision, which implies that Rasmussun's statement about full text databases will also apply to fiction databases. On the other hand, recall and precision did not display the inverse relationship that

Rasmussen would have expected. Also, precision and recall values did not often reach the 50% mark at which Lesk says search engines tend to operate.

Several possible explanations exist for why precision and recall did not meet the standards set by Lesk. First of all, it may be due to the corpus comprising of short stories, which would have ramifications for LSI's potential to analyze fiction. However, this is difficult to determine without more testing. Two other possible explanations should be examined before deciding that low precision and recall are directly determined by a fiction corpus: these are corpus size and evaluation criteria. It is possible that the corpus simply was not large enough to provide statistical diversity. The criteria by which precision and recall were determined can also be flawed, as discussed in section 4.1.6. Corpus size and evaluation criteria should be excluded as possible explanations for overall low precision and recall before it is decided that a fiction corpus is the direct cause.

Landauer also noted the importance of having a large corpus when using LSI. He writes, "In general, [LSI] performs best when used to simulate average results over many cases" (1998, p. 33). Since it is doubtful that 44 documents would be considered *many*, it would be recommended to try the tests again on a much larger corpus.

It is also possible that the shape of the precision and recall curves would have been different if a full range of values of k had been tested.

4.2.3 Word Use and Genre

Since precision and recall have performed as well as could have been hoped, it is important to evaluate LSI's applicability to fiction in a more general sense. This project has been based on the hypothesis that fiction would behave similarly to non-fiction and that we would be able to arrange short stories according to similarity based on their full texts. As far as the author knows, no work has been done in this area before and it is therefore especially important to present some comments about the relationship between word use and aboutness/theme/genre.

In order to test this relationship, an experiment was done to determine whether readers might be able to recreate the story and themes from a word frequency list and vice versa. Three volunteers were found to this purpose that had not participated in the focus group work discussed above. This experiment should not be taken as universally representative due to so few participants and it would be recommended to test again on a larger scale in the future.

In the first part of the experiment three volunteers were given word frequency lists for either *Uni* or *The Office Party* and asked to write some sentences about what they thought the story was about based on that list.

Two volunteers received the word frequency list for *The Office Party*. Both volunteers understood, though they did not receive the title of the story, that the story described an office party, relationships between the colleagues, and what they talk about. Both mentioned that there was an after party though the word only occurs three times in the text. One volunteer marked *equality* as being an important word for the text but admitted that he forgot to write it into his summary. That he thought it was important is good because both focus groups had placed the story in the *feminism* category of their cluster maps.

The third volunteer received the word frequency list for *Uni*. He correctly identified that the story took place in Norway during World War II. He also noticed that the characters were involved in a resistance movement and there was a romantic theme as well. At the end of his summary, he posited that the story was about how war changes people and relations. This is a fairly appropriate description of *Uni*.

The second part of the experiment took the opposite approach: these same three volunteers read a short story each and marked which words they thought described the genre. Their understanding of the genre progressed as they read the story; they were not told what the genre was ahead of time. If a volunteer received the word frequency list for *Uni* they were given the text of *The Office Party* so as to avoid prior knowledge and preconceptions about the text.

Two volunteers read *Uni* and marked which words they thought described the genre and plot of the story. Two themes that one volunteer focused on were *romance* and *war*. This volunteer marked words like *kiss*, *naked*, *Germans*, and *pistol*. The other volunteer marked many more terms and these seem to focus on the good times (*food*, *clothing*, *bodies*, *kisses*) and the bad times (*jealousy*, *Germans*, *drinking*).

Interestingly, the focus groups seemed to focus mostly on the fact that *Uni* was about World War II since they were comparing it to Haug's *The Blow*, which is set during the same time period. Yet the volunteers focus on different themes in addition. Since the volunteers had a different task, they were allowed to focus on as many themes as they liked, rather than try to fit the story into a category that other stories could be placed in or near.

The third volunteer read and marked terms in *The Office Party*. Both focus groups had identified this story as being mainly about feminism. Of the words that the volunteer marked, few are directly related to feminism (*equality* being a major exception). His words focused on the party (*drinking*, *dancing*) and describing people at the party and their reactions (*man*, *age*, *warning*, *laughable*).

Of the words marked, some had a very high frequency and others were used no more than once. In *The Office Party* some words that were considered important by the volunteer were *Miss*, *office*, *party*, *man*, and *director*. These words are used in the text 18, 9, 11, 11, and 13 times respectively. Some of the more popular words in *Uni* were *factory* (30), *weapon* (8), *kiss* (9), *person* (5), *each other* (8), and *German* (21). Some words that were used only twice each, but were still important for describing the theme in each story, were *jealousy*, *passion*, *cocktail*, and *education*. In fiction, a theme can be described by the plot and not identified specifically by name.

It was expected that it would be easier to make connections between words and aboutness/themes/genre in *Uni* than *The Office Party*. From the observations given above, it seems that this was correct. Words chosen from *Uni* and the recreated summary reflect that the story is about romance and war. Themes in *The Office Party* are more difficult to identify because on the surface it is merely about people at a party. The words have to be taken in the context of the story in order to understand

that the main character is making a stand for feminism by acting the way she does. When the words are out of order, it is no longer clear who is acting how and how that might determine the story's themes and genre.

From this small test of word use and genre, some observations can be made. First of all, it may be more difficult to identify genre/subject from word use in fiction than it is in non-fiction. Success may be highly dependent on the text being analyzed, as some texts are more direct in describing their themes. Since it was possible, to a certain degree, to recreate the story and themes from a word frequency list there is still hope for the vector space model and LSI's application to fiction. At the same time it is noted that this process will be more challenging than for fiction and that results may vary.

4.2.4 What Makes Fiction Difficult to Categorize

So why is it then that results may vary when analyzing fiction based on word use? Three suggestions will be put forward here: 1) fiction draws readers with many different intentions, 2) there is a clearer relationship between word use and subject in non-fiction than in fiction, and 3) fiction changes with context.

The first suggestion reflects the idea that at some level we approach a particular work of fiction for a variety of reasons, whereas non-fiction is more likely to be read specifically for its main subject. As an example, let us compare a book about biochemistry and *Anne Karenina* by Leo Tolstoy. One is most likely to read about biochemistry because they want to learn more about biochemistry. *Anna Karenina*, on the other hand, can be approached for many reasons. Perhaps one wants to compare it to the film or because one is interested in classic literature, realism, Russia, romance, adultery, or fashionable society in the late 1800s, or because it was recommended by an acquaintance. The point is that a work of fiction is more likely to be read for different reasons, meaning that people will place emphasis on different aspects. This makes classification of fiction more difficult as it will be impossible for everyone to agree what each story is really about and how it compares to other stories. It also makes it difficult to say which criteria make two stories similar or dissimilar.

As to the second suggestion, we can look at the corpus used by Deerwester (1990) in his first tests of the LSI algorithm. He used nine documents that he divided into two classes, *human-computer interaction* and *graphs*. Though one has to understand the subject area to know that *trees* are a kind of graph, the terms are used consistently and correctly for their subject area. There is also a very clear relationship between the subject matter and the terms used. This is not necessarily the case in fiction where one can evoke a subject without ever naming it directly by using various literary techniques.

Another difficulty that arises when reading or classifying fiction, and one that may be particularly relevant for this corpus, is the context in which it is read. Many of the stories in this corpus were written long ago; some are over 100 years old and thus written in a different society than modern Norway. Our understanding of the story may differ quite dramatically from the original intention. This is a discussion that one should have with schoolchildren reading *Huckleberry Finn* for example. The children need to understand that Mark Twain was reflecting opinions common at the time in his writing but that does not mean that modern society thinks the same way. Since the stories were read out of the context in which they were written, this may have had an effect on both the focus group analysis and LSA, though it is difficult to say for certain.

The context can also be quite personal and change as one gets older or is in a different mood. As we age and change, so does our perspective and what we will emphasize in a text. It is doubtful that a child would categorize a collection in the same way that an adult or elderly person would, simply because they have different amounts of experience and perspectives on life.

The points brought up in the discussions above have shown that the LSI test results were not as successful as desired but that this may be the consequence of a small corpus with short documents or simply a general difficulty with classifying fiction. Knowing that fiction can be a challenge, let us conclude by discussing where the research could go from here.

5.0 Conclusion

This paper has taken the first steps towards creating an automatic genre classification system for fiction. Its goals have been to reconfirm that genre is the most reasonable access point to fiction from a reader's point of view; to apply Latent Semantic Indexing to a fiction corpus; and to evaluate the results of the tests with criteria created by readers.

LSI was chosen for these experiments on a fiction corpus for several reasons. First, it would be able to make as many clusters or categories as seem fitting. Because of this a book that spans mystery and romance, for example, would not have to choose a single genre to represent it best. Instead a more specific cluster could be made to represent the book better. A librarian classifying a fiction collection would have to plan out how many genres and of which type she would want to use from the beginning so as to avoid extra work later. This leads to another reason for choosing LSI as a test algorithm. An automatic method like LSI can change the genres on the fly and reclassify books as more information is added to the system. Making changes in real time is more practical in a digital system than a physical one. With digital libraries on the rise, the need for classification-by-algorithm systems is becoming more prevalent all the time.

As discussed above, results were both positive and negative. Precision and recall did not behave exactly as expected and the number of matches was not as high as could be hoped. At the same time, the experiments were not a complete failure. The LSI tests always showed some matches between documents with which the focus groups would have agreed. In addition, volunteers were able to recreate to a certain degree stories based on word frequency lists. This hints that the vector space model is still a viable candidate for representing fiction, though more research needs to be done to improve the system.

Perhaps the most important change that would be recommended for future implementations of this research would be to utilize larger corpuses. As mentioned above, Landauer et al. (1998) found that LSI worked best on average over many

cases. A larger corpus might also resolve the issues surrounding precision and recall experienced in these tests in terms of low values and unexpected relationships.

A number of other options are also available for potential improvement of the research begun here. First, as the corpus becomes larger it would be worth examining some techniques for handling large, sparse matrices as discussed in Berry (1992). Manning et al. (2008, p. 7) recommended an inverted index for the same purpose. Second, a larger corpus will less likely be homogenous so partitioning the documents could be examined (Gao & Zhang, 2005). Third, if stemming and lemmatization do not seem to produce significant improvements, a different option would be Parts-of-Speech taggers (Rishel et al., 2007). Fourth, it would be worth experimenting with weak and strong stemmers as done in the OKAPI experiments (Harman, 1994). Finally, while it still seems possible that the vector space model and LSI will work on fiction, if they are found to be ineffective other options would include random projection and self-organizing maps (Ampazis & Perantonis, 2004). These options and others unknown to the author may be worth testing on fiction corpuses to see if they produce more desirable results when attempting to create a genre classification.

It has not been the goal of this paper to create a fully functional classification system but instead to take a few first steps in evaluating whether fiction could be represented using the vector space model and analyzed using LSI. Should a fiction classification system be created, it will have to "make it easier for library users to find the types of fictional work they want" (Baker & Shepherd, 1987, p. 246). It should support browsing since the user survey implemented here reconfirmed research by Baker (1996) and Willard and Teece (1983) that browsing is a technique used by many library users. The system should also be able to introduce readers to new authors based on what the reader might like. Arranging fiction closely by genre and theme should be help to accomplish this.

Smith (2001) would have placed even more stringent criteria on a fiction classification scheme. He discussed types of appeal, pacing, kind of action represented, characters, the nature of the world represented, emotional impact on the reader, and demands placed on the reader. Smith placed emphasis on how a fiction classification or fiction search engine should use these types of appeal as an

intellectual foundation since the criteria used for non-fiction cannot be directly applied to fiction (p. 94).

Though Smith had a point that fiction must be evaluated using different criteria than non-fiction, the result cannot be a system that is too complex to be used. Yu and O'Brien (1999, pp. 170-171) critiqued Beghtol for creating descriptions so complex as to be unlikely to be used as the basis of a search. Similar critiques could be made of Pejtersen (1977) and Vernitski (2007). Therefore, though the system needs to be adopted for fiction, it cannot be too difficult to use or represent works in a way inaccessible to readers.

If a fiction classification system is eventually made that helps to bridge the gap between traditional classification and classification-by-algorithm by making the process automatic there are many ways that it could be implemented.

First, since the goal is still to be useful for a library's physical collection, a shelf classification could be created based on the algorithm's results. Once a successful analysis has been made of a collection of fictional works, the relationships could be translated into codes that would arrange the works on bookshelves. The shelf classification will have to be updated from time to time as more documents are added to the collection, which could imply moving some books around to make room for new genre divisions. It is hoped that the system would become fairly stable as the corpus size increases, decreasing the chance that librarians would have to constantly move books around on the shelves. This should also be tested in future research.

Second, a classification-by-algorithm scheme could be implemented digitally. Here the most obvious option is a fiction search engine. Such a search engine would be able to return documents based on keywords, genre descriptors, and typical metadata searches including author and title.

Another digital implementation option would be in a digital library. The opportunities here are many. One example would be the possibility for multiple types of cross-classification. The whole point of card catalogues was to be able to organize information according to different needs. In a digital system, memory is cheap and

sorting documents is only limited to one's imagination and resources. Saarti (1997) would argue for a fiction index since it is the best way to help people find specific works. This could be implemented in a digital search engine or library.

Hopefully further research will find that LSI, and consequently the vector space model, is suited to classifying fiction by genre similarities. Hjørland and Albrechtsen (1999, p. 133) wrote, "…established disciplines do not always represent newer territories…" but LSI might be able to adapt regardless of being an established discipline within non-fiction. Being a digital system, LSI would be able to update its organization of the fiction documents as often as required. Thus, the newest genres would always have a place in the system.

This paper has shown how genre is the most reasonable access point for fiction and that there is a connection between word use and genre in fiction literature. Though, as noted above, the relationship between LSI, the vector space model, and word use as regards fiction requires more testing. Recommendations include other techniques discussed above and, above all, to run tests on a larger corpus. The research presented here is promising for future endeavors and it is highly recommended that the research be continued.

Works Cited

Ampazis, N., & Perantonis, S. J. (2004). LSISOM – A Latent Semantic Indexing Approach to Self-Organizing Maps of Document Collections. *Neural Processing Letters, 19*, 157-173.

Answers Corporation. (2013). How Large is the Internet? Retrieved February 18, 2013, from http://wiki.answers.com/Q/How_large_is_the_Internet

Baker, S. L. (1988). Will Fiction Classification Schemes Increase Use? *RQ, 27*(3), 366-376.

Baker, S. L. (1996). A Decade's Worth of Research on Browsing Fiction Collections. In K. D. Shearer (Ed.), *Guiding the Reader to the Next Book* (pp. 127-147). New York, NY: Neal-Schuman Publishers, Inc.

Baker, S. L., & Shepherd, G. W. (1987). Fiction Classification Schemes: The Principles behind Them and Their Success. *RQ, 27*(2), 245-251.

Barbour, R. (2007). *Doing Focus Groups* (Vol. 4). London: SAGE.

Batley, S. (2005). *Classification in Theory and Practice*. Oxford: Chandos Publ.

Beghtol, C. (1989). Access to Fiction: A Problem in Classification Theory and Practice. Part I. *International Classification, 16*(3), 134-140.

Beghtol, C. (1994). *The Classification of Fiction: the Development of a System Based on Theoretical Principles*. Metuchen: Scarecrow Press.

Bell, H. K. (1991). Indexing Fiction: A Story of Complexity. *The Indexer, 17*(4), 251-256.

Berry, M. W. (1992). Large Scale Singular Value Computations. *International Journal of Supercomputer Applications, 6*(1).

Bowman, J. H. (2005). *Essential Dewey*. London: Facet.

Brattli, T. (2012). Why Build Dewey numbers? The Remediation of the Dewey Decimal Classificiation system. *Nordlit*(30), 189-206.

Briet, S. (2006). *What is Documentation?* Lanham, Md.: Scarecrow Press.

Brinkmann, S., Tanggaard, L., & Hansen, W. (2012). *Kvalitative metoder*. Oslo: Gyldendal akademisk.

Cannell, J., & McCluskey, E. (1996). Genrefication: Fiction Classification and Increased Circulation. In K. D. Shearer (Ed.), *Guiding the Reader to the Next Book*. New York, NY: Neal-Schuman Publishers, Inc.

Childs, P., & Fowler, R. (2006). *The Routledge Dictionary of Literary Terms*. London: Routledge.

Chowdhury, G. G., & Chowdhury, S. (2007). *Organizing Information: From the Shelf to the Web*. London: Facet Publ.

Conaway, J. (2000). *America's Library: The Story of the Library of Congress 1800-2000*. New Haven, Conn.: Yale University Press.

Conrado, M. D., Gutierrez, V. A. L., & Rezende, S. O. (2012). Evaluation of Normalization Techniques in Text Classification for Portuguese. In B. Murgante, O. Gervasi, S. Misra, N. Nedjah, A. Rocha, D. Taniar & B. O. Apduhan (Eds.), *Computational Science and Its Applications - Iccsa 2012, Pt Iii* (Vol. 7335, pp. 618-630). Berlin: Springer-Verlag Berlin.

Davis, C. H. (1976). Pragmatic Expansion of an Enumerative Classification Scheme. *Journal of the American Society for Information Science, 27*(3), 174-176.

Deerwester, S. (1990). Indexing by Latent Semantic Analysis. *Journal of American Society for Information Science, 41*(6), 391-407.

Dolamic, L., & Savoy, J. (2010). When Stopword Lists Make the Difference. *Journal of the American Society for Information Science and Technology, 61*(1), 200-203. doi: 10.1002/asi.21186

Erenel, Z., & Altincay, H. (2012). Nonlinear Transformation of Term Frequencies for Term Weighting in Text Categorization. *Engineering Applications of Artificial Intelligence, 25*(7), 1505-1514. doi: 10.1016/j.engappai.2012.06.013

Fink, A. (2009). *How to Conduct Surveys: A Step-by-Step Guide*. Los Angeles: Sage.

Fowler, F. J. (2009). *Survey Research Methods*. Los Angeles: Sage.

Gao, J., & Zhang, J. (2005). Clustered SVD Strategies in Latent Semantic Indexing. *Information Processing and Management, 41*, 1051-1063.

Geiß, J. (2008). *Latent Semantic Indexing and Information Retrieval: A Quest with Bosse*. Saarbrücken: Verlag Dr. Müller.

Goodall, D. (1992). *Browsing in Public Libraries* (new impression, 1992 ed. Vol. 1). Loughborough: Library and Information Statistics Unit.

Gordon, M. D., & Dumais, S. (1998). Using Latent Semantic Indexing for Literature Based Discovery. *Journal of American Society for Information Science, 49*(8), 674-685.

Gyldendal. Himmel over London. Retrieved April 2, 2013, from http://www.gyldendal.no/Skjoennlitteratur/Romaner-og-noveller/Himmel-over-London

Harman, D. (1994). Automatic Indexing. In R. Fidel, T. B. Hahn, E. M. Rasmussen & P. J. Smith (Eds.), *Challenges in Indexing Electronic Text and Images* (pp. 247-264). Medford, NJ: Learned Information, Inc.

Harrell, G. (1996). Use of Fiction Categories in Major American Public Libraries. In K. D. Shearer (Ed.), *Guiding the Reader to the Next Book* (pp. 149-158). New York, NY: Neal-Schuman Publishers, Inc.

Hegna, K. (2003). Universell bibliografisk kontroll : mål, midler og teknologi. *Norsk tidsskrift for bibliotekforskning, 6*(16), 35-69.

Hjørland, B., & Albrechtsen, H. (1999). An Analysis of Some Trends in Classification Research. *Knowledge Organization, 26*(3), 131-139.

Huff, K. L. (2006). *Genre Fiction Classification: A Continuation Study of its Reception by Patrons in the Durham County (NC) Public Library.* (Master of Science in Library Science), University of North Carolina at Chapel Hill.

Ingersoll, G. S., Morton, T. S., & Farris, A. L. (2013). *Taming Text: How to Find, Organize, and Manipulate it*. Shelter Island, N.Y.: Manning.

Jenkins, C., Jackson, M., Burden, P., & Wallis, J. (1998). Automatic Classification of Web Resources Using Java and Dewey Decimal Classification. *Computer Networks and ISDN Systems, 30*(1–7), 646-648. doi: http://dx.doi.org/10.1016/S0169-7552(98)00035-X

Kazantseva, A. (2006). *Automatic Summarization of Short Fiction.* (Master of Computer Science), Ottawa-Carleton Institute for Computer Science, Ottawa, Ontario, Canada.

Landauer, T. K., Foltz, P. W., & Laham, D. (1998). An Introduction to Latent Semantic Analysis. *Discourse Processes, 25*, 259-284.

Lesk, M. (2005). *Understanding Digital Libraries*. Amsterdam: Elsevier.

Lochbaum, K. E., & Streeter, L. A. (1989). Comparing and Combining the Effectiveness of Latent Semantic Indexing and the Ordinary Vector Space Model for Information Retrieval. *Information Processing and Management, 25*(6), 665-676.

Lovins, J. B. (1968). Development of a Stemming Algorithm. *Mechanical Translation and Computational Linguistics, 11*, 22-31.

Manning, C. D., Raghavan, P., & Schütze, H. (2008). *Introduction to Information Retrieval*. Cambridge: Cambridge University Press.

Moyer, J. E. (2005). Adult Fiction Reading: A Literature Review of Reader's Advisory Services, Adult Fiction Librarianship, and Fiction Readers. *Reference and User Services Quartlery, 44*(3), 220-231.

Mullan, J. (2006). *How Novels Work*. Oxford: Oxford University Press.

Nordquist, R. (2013). A List of Standard Contractions in English. Retrieved April 2, 2013, from http://grammar.about.com/od/words/a/EnglishContractions.htm

O'Brien, A., & Yu, L. (1996). Domain of Adult Fiction Librarianship. *Advances in Librarianship, 20*, 151-189.

Pasco, A. H. (2010). *Inner Workings of the Novel: Studying a Genre*. New York: Palgrave Macmillan.

Pejtersen, A. M. (1977). *Klassifikation af skønlitteratur basert på en analyse af låner-bibliotekar kommunikation.* Paper presented at the Second International Research Forum of Information Science, Copenhagen.

Pejtersen, A. M., & Austin, J. (1983). Fiction Retrieval: Experimental Design and Evaluation of a Search System Based on Users' Value Criteria (Part 1). *Journal of Documentation, 39*(1), 230-246.

Porter, M. F. (1980). An Algorithm for Suffix Stripping. *Program-Automated Library and Information Systems, 14*(3), 130-137. doi: 10.1108/eb046814

Ranganathan, S. R. (1960). *Colon Classification: Basic Classification* (Vol. 26). Madras: The Association.

Rasmussen, E. M. (1994). Introduction. In R. Fidel, T. B. Hahn, E. M. Rasmussen & P. J. Smith (Eds.), *Challenges in Indexing Electronic Text and Images* (pp. 241-245). Medford, NJ: Learned Information, Inc.

Řehůřek, R. (2013). Experiments on the English Wikipedia. Retrieved April 22, 2013, from http://radimrehurek.com/gensim/wiki.html

Řehůřek, R., & Sojka, P. (2010). *Software Framework for Topic Modelling with Large Corpora*. Paper presented at the Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks. http://radimrehurek.com/gensim/lrec2010_final.pdf

Richard, A. J. (1999). *Genre Fiction Classification: A Study of the Durham County Library.* (Master of Science in Library Science), University of North Carolina at Chapel Hill.

Rishel, T., Perkins, L. A., Yenduri, S., & Zand, F. (2007). Determining the Context of Text Using Augmented Latent Semantic Indexing. *Journal of American Society for Information Science and Technology, 58*(14), 2197-2204.

Saarti, J. (1997). Feeding with the Spoon, or the Effects of Shelf Classification of Fiction on the Loaning of Fiction. *Information Services & Use, 17*, 159-169.

Saarti, J. (1999). Fiction Indexing and the Development of Fiction Thesauri. *Journal of Librarianship and Information Science, 31*(2), 85-92.

Salant, P., & Dillman, D. A. (1994). *How to Conduct Your Own Survey*. New York: Wiley.

Salton, G., & Buckley, C. (1988). Term-Weighting Approaches in Automatic Text Retrieval. *Information Processing & Management, 24*(5), 513-523. doi: http://dx.doi.org/10.1016/0306-4573(88)90021-0

Saricks, J. G. (2001). *The Readers' Advisory Guide to Genre Fiction*. Chicago: American Library Association.

Sear, L., & Jennings, B. (1991). Organizing fiction for use. In M. Kinnell (Ed.), *Managing Fiction in Libraries* (pp. 101-119). London: Library Association Publishing.

Shearer, K. D. (Ed.). (1996). *Guiding the Reader to the Next Book*. New York, NY: Neal-Schuman Publishers, Inc.

Shoham, S. (2000). *Library Classification and Browsing : The Conjunction of Readers and Documents*. Brighton: Sussex Academic Press.

Singhal, A., Buckley, C., & Mitra, M. (1996). *Pivoted Document Length Normalization*. Paper presented at the Proceedings of the 19th annual international ACM SIGIR conference on Research and development in information retrieval, Zurich, Switzerland.

Smith, D. (2001). Reinventing Reader's Advisory. In K. D. Shearer (Ed.), *The Readers' Advisor's Companion*. Englewood, CO: Libraries Unlimited.

Spärck Jones, K. (1972). A Statistical Interpretation of Term Specificity and its Application in Retrieval. *Journal of Documentation, 28*(1), 11-21.

Staikos, K. S. (2004). *The History of the Library in Western Civilization*. New Castle, Del.: Oak Knoll Press.

Svenonius, E. (2000). *The Intellectual Foundation of Information Organization.* Cambridge, MA: MIT Press.

Taylor, A. G. (2004). *The Organization of Information*. Westport, Conn.: Libraries Unlimited.

Taylor, A. G., & Miller, D. P. (2006). *Introduction to Cataloging and Classification*. Westport, Conn.: Libraries Unlimited.

The Information Retrieval Group.). Stop words. Retrieved March 6, 2013, from http://ir.dcs.gla.ac.uk/resources/linguistic_utils/stop_words

Tromsø Bibliotek og Byarkiv. (2012a). *Brukerundersøkelse 2012*.

Tromsø Bibliotek og Byarkiv. (2012b). *Statistikk 2012*.

van den Heuvel, C., & Rayward, W. B. (2011). Facing Interfaces: Paul Otlet's Visualizations of Data Integration. *Journal of the American Society for Information Science and Technology, 62*(12), 2313-2326. doi: 10.1002/asi.21607

Vernitski, A. (2007). Developing an Intertextuality-Oriented Fiction Classification. *Journal of Librarianship and Information Science, 39*, 41-52.

Warner, A. J. (1994). The Role of Linguistic Analysis in Full-text Retrieval. In R. Fidel, T. B. Hahn, E. M. Rasmussen & P. J. Smith (Eds.), *Challenges in Indexing Electronic Text and Images* (pp. 265-275). Medford, NJ: Learned Information, Inc.

Wiktionary. (2008, January 30, 2011). Wiktionary:Frequency lists/Contemporary fiction. Retrieved February 25, 2013, from http://en.wiktionary.org/wiki/Wiktionary:Frequency_lists/Contemporary_fiction

Willard, P., & Teece, V. (1983). The Browser and the Library. *Public Library Quarterly, 4*(1), 55-63.

Willett, P. (2006). The Porter Stemming Algorithm: Then and Now. *Program-Electronic Library and Information Systems, 40*(3), 219-223. doi: 10.1108/00330330610681295

Yu, L., & O'Brien, A. (1999). A Practical Typology of Adult Fiction Borrowers Based on their Reading Habits. *Journal of Information Science, 25*(1), 35-49.

Appendix A: User Survey Consent Information

Information about the survey/Consent to participate

You are hereby requested to participate in a study that is part of a Master's thesis in document science at the University of Tromsø. The topic of the thesis is categories/arrangement of fiction from a user-friendly perspective. Therefore the thesis includes a user survey - to find out library patrons' preferences as regards arrangement of fiction.

If you participate, you will be one of about 50 participants and all responses will be anonymous. Your participation is voluntary, you do not need to answer all the questions, and you can stop at any time. Responses to the survey will also be shared with Tromsø Public Library.

It takes about 5 minutes to fill out the survey and you will not receive any compensation.

The results of the survey will give Tromsø Public Library and library researchers insight about how the arrangement of fiction can be improved.

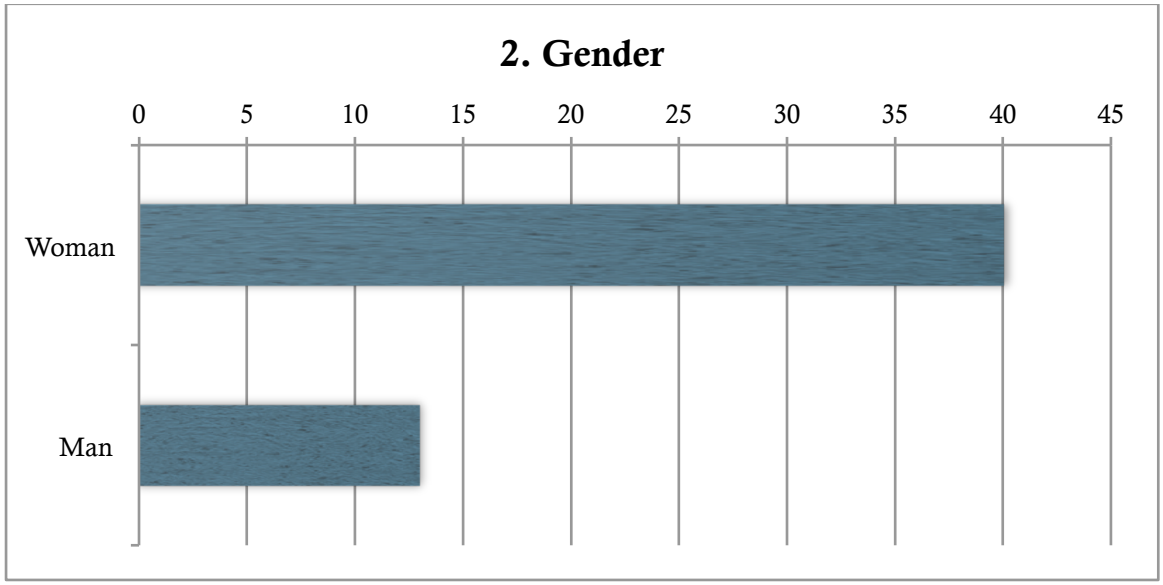
When it is finished in May 2013, the Master's thesis will be made freely available in Munin. Munin is the University of Tromsø's open research archive and can be found at http://munin.uit.no/

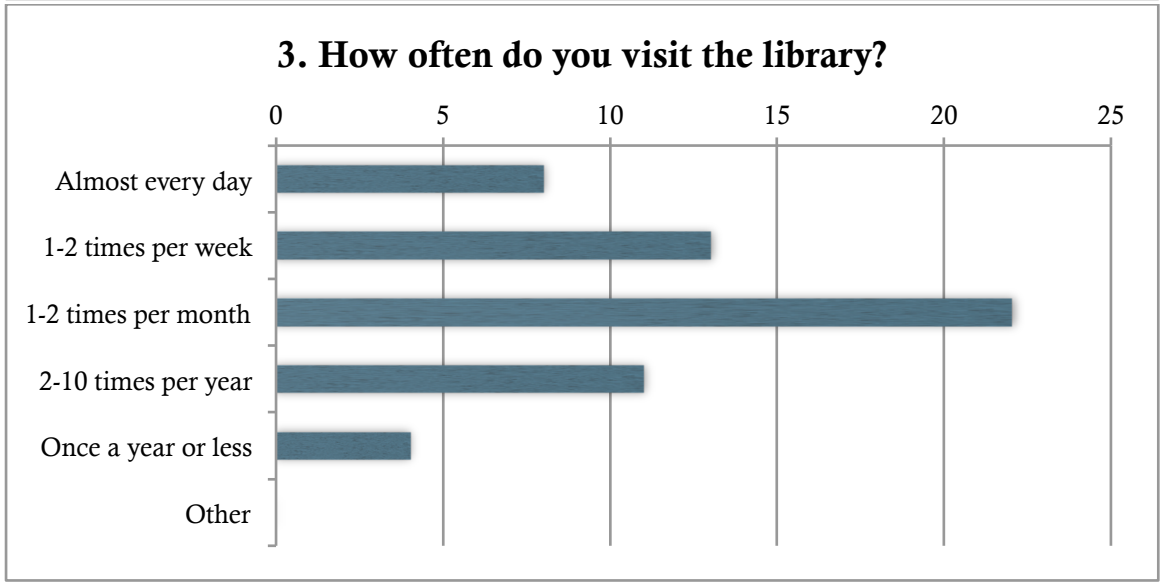
If you have any questions or comments regarding the study you can email Nora MacLaren (Master's student) at nma010@uit.no or Stine Fjeldsøe (administrator at Tromsø library) at stine.fjeldsoe@tromso.kommune.no

By filling out the survey you are consenting to participate in the study.

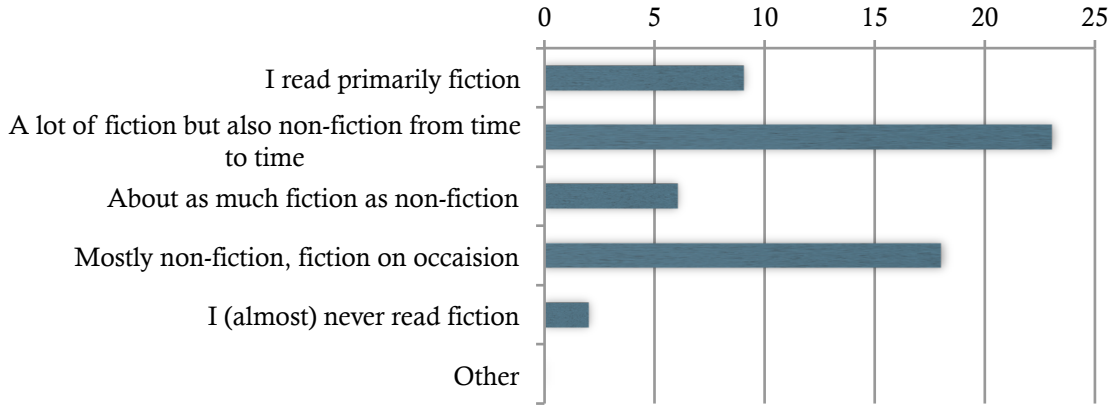
Appendix B: Survey Questions and Responses

Comments have been translated from Norwegian by the author.

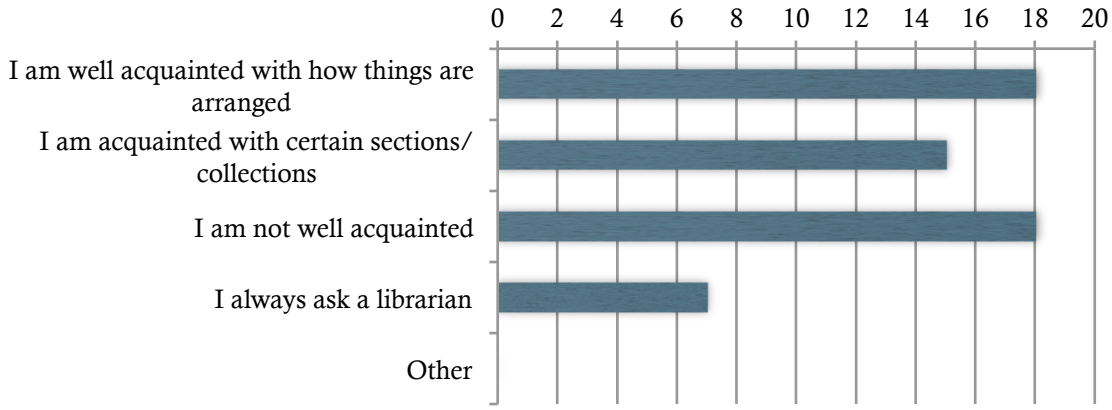


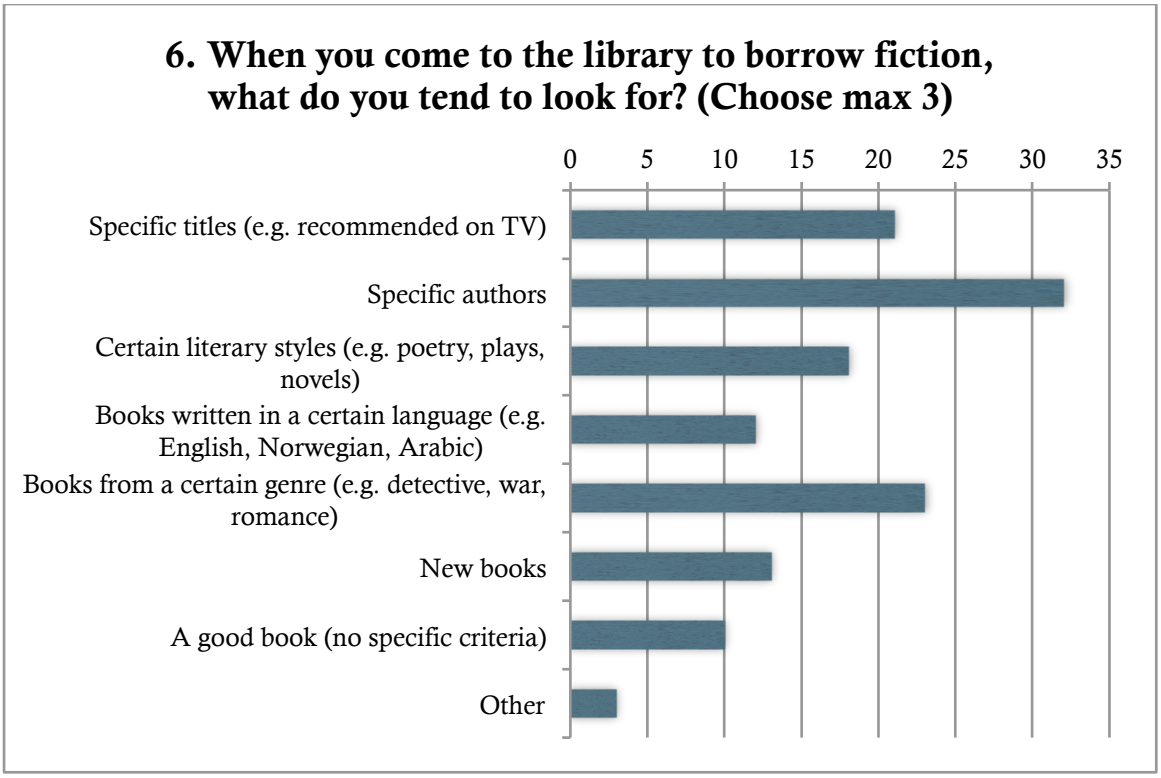


4. How much fiction do you read (including audio books) as opposed to non-fiction?



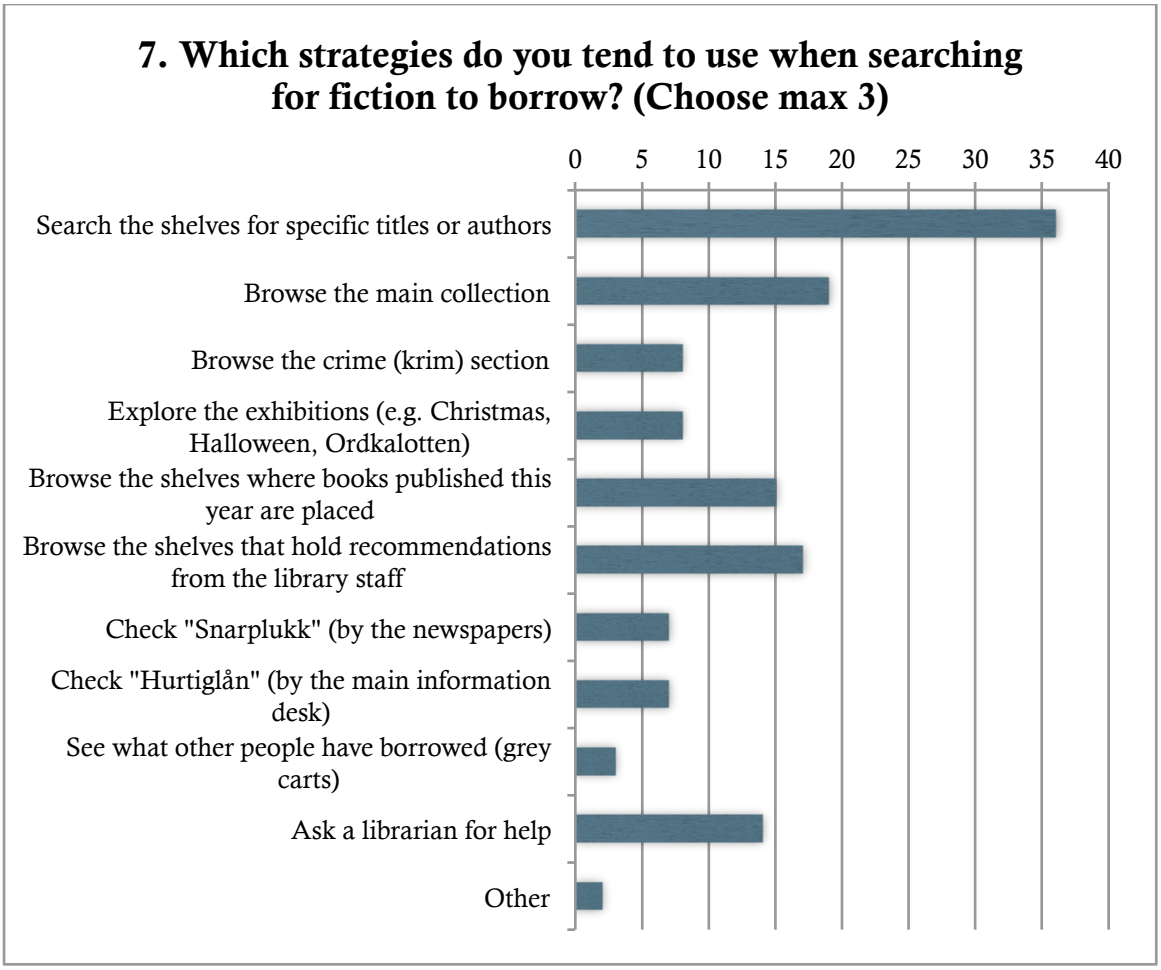
5. How well acquainted are you with the arrangement of fiction at the library you use the most?





Other:

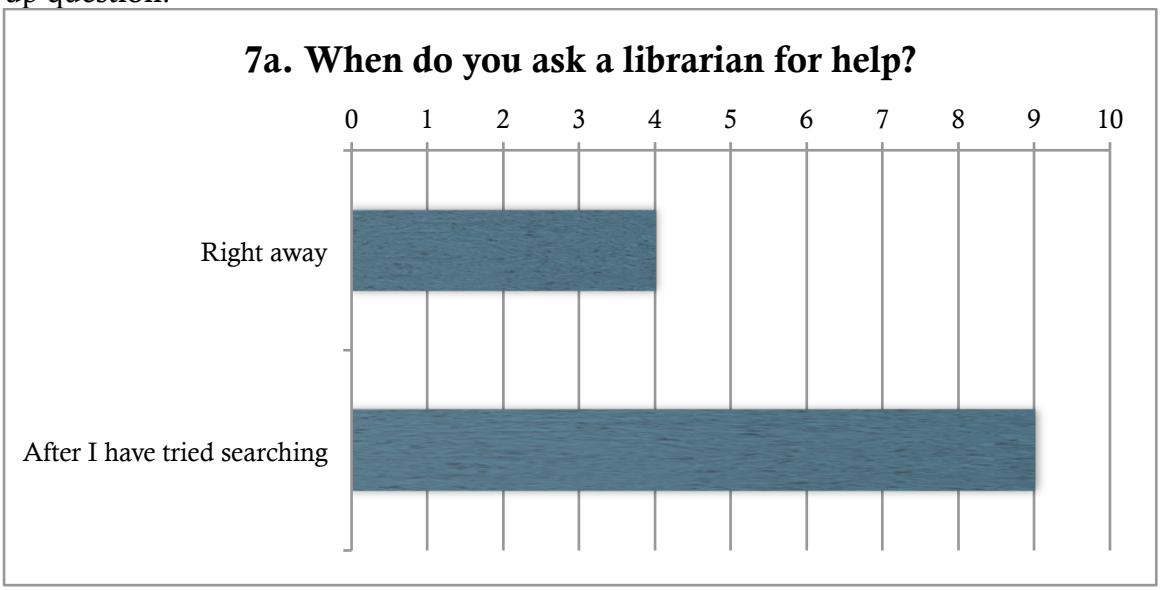
- *Classics one should have read. Hamsun, Ibsen etc.*
- *Nordic literature*
- *Thin books*
- *Audiobooks*
- *Interesting covers*



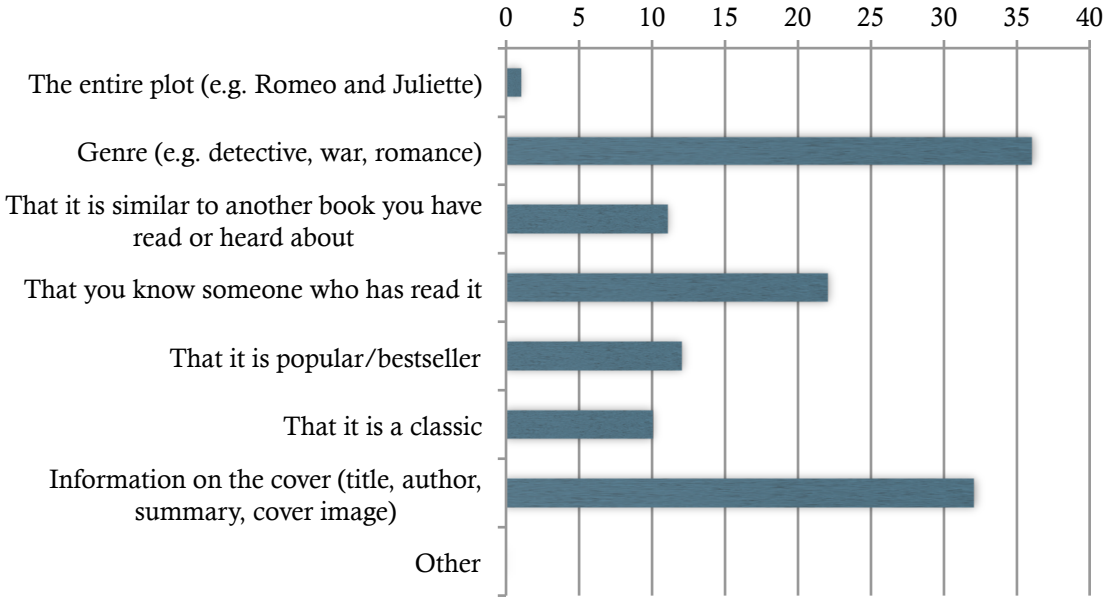
Other:

- Search the shelves
- Look at what might be found in the audiobook shelves
- Use mostly the Internet to find books I want to borrow

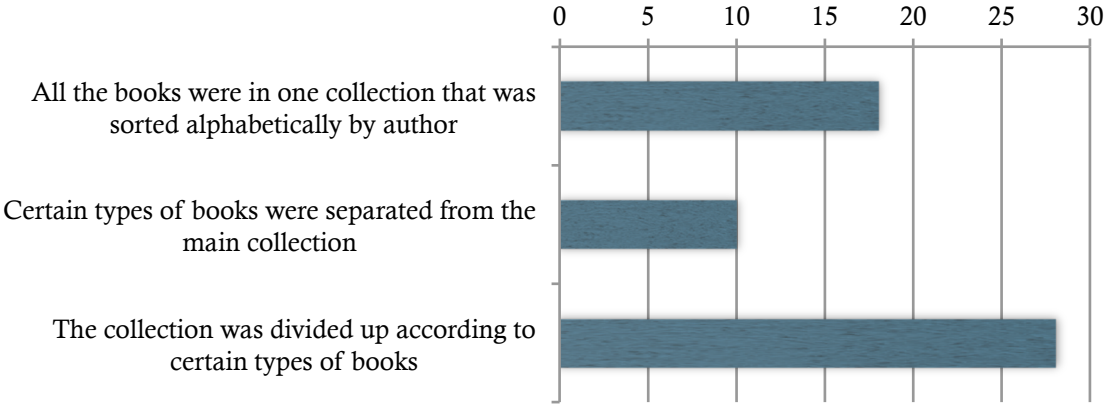
If participant answered "Ask a librarian for help" they were given 7a as a follow up question.



8. How much do you want to know about a book before you start reading it? (Choose max 2)

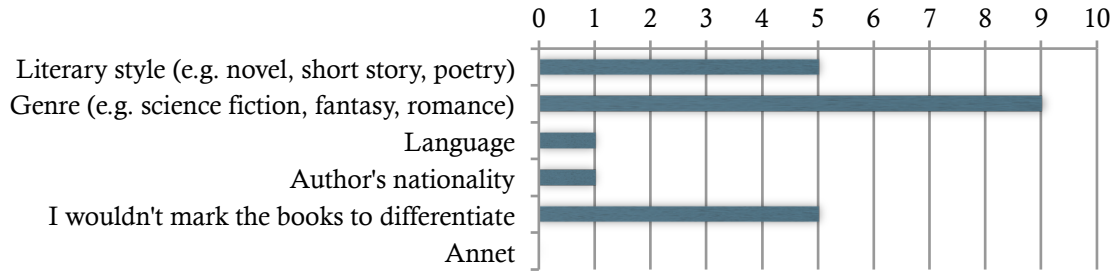


9. If you could decide how fiction was arranged at the library, would you prefer…?



If the participant answers "All books" the follow up questions are 9a and 9b. If "Certain types" then the follow up questions are 9c and 9d. If "Collection" then the follow up questions are 9e and 9f.

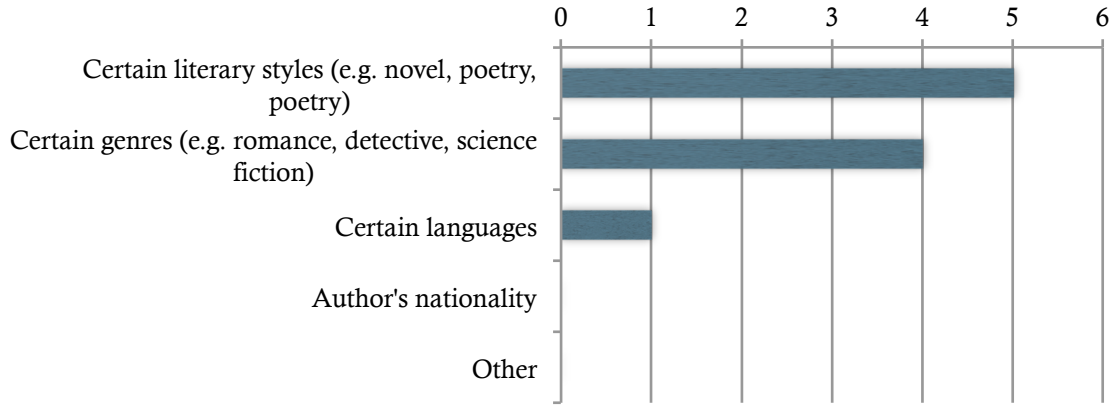
9a. Would you mark the books to let people know which type of book it was? If yes, which types of books do think it is most important to mark? (The marks could be stickers on the book's spine, for example.) (Choose max 2)



9b. Which literary styles/genres/languages/nationalities/other do you think should be marked?

- *Novel, detective/crime*
- *Best sellers! New*
- *Non-fiction vs. fiction vs. children*
- *Norwegian*

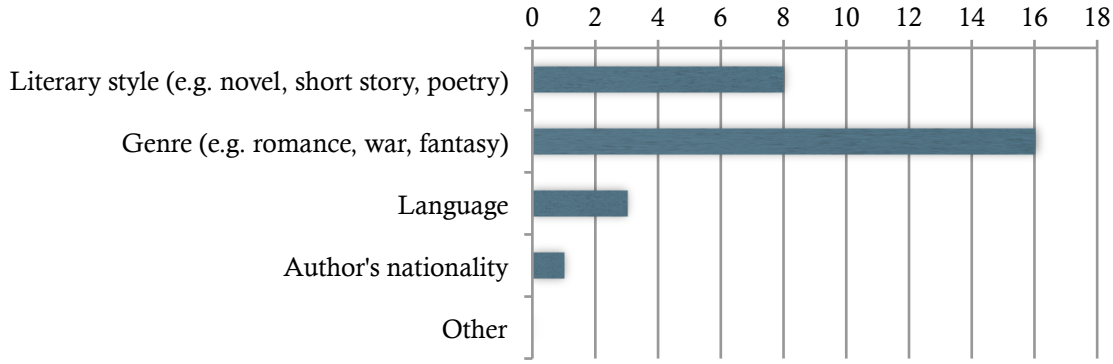
9c. Which types of books would you separate from the main collection? (Choose max 2)



9d. Which literary styles/genres/languages/nationalities do you think should be separated from the main collection?

- *Paranormal*
- *Thin books. Read mostly thin books where the words are worth their weight in gold*
- *English, historical novels*

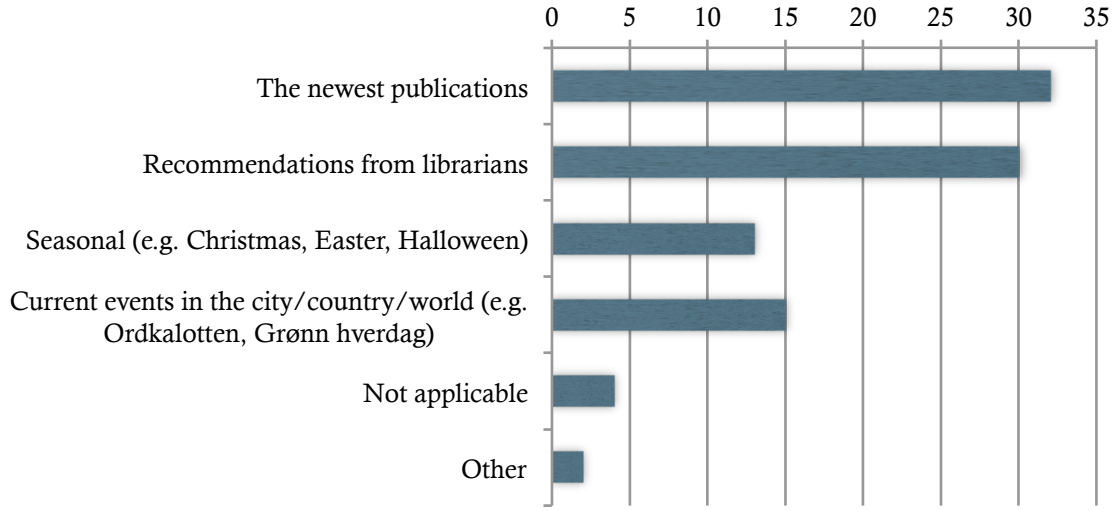
9e. All the books are arranged based on type. Which type do you think is the most important for categorizing the fiction collection at the library?



9f. Which literary styles/genres/languages/nationalities would you use to categorize the books?

- *Prose should be separated from poetry and theater which should be easy to find in their own shelves*
- *Language of course. I don't want to trawl through books in different languages I don't understand to find something readable. But it can certainly be divided into more categories than just language.*
- *Novel, Nordic*
- *War histories*
- *Romance*
- *Detective/crime, Sami, Norwegian classics*
- *Modern literature, classics, crime, fantasy, science fiction, relevant literature, fiction (politics/social studies/literature studies etc.)*
- *Different languages, different literary styles like suggested above. Not by genre – I think it would confuse me. Think also books from certain parts of the world could be features – like Middle East, South America – while all the other books are organized in an alphabetical catalogue, divided as suggested above.*

10. Regardless of how you would arrange the fiction collection, which kinds of special collections would you use? (Choose max 3, collections can be permanent or temporary)



Other:

- *Thin books*
- *New and popular*
- *Sami fiction, classics*

11. Do you have any additional comments related to this study?

- *Everything is solvable with a little help*
- *Recommendations from librarians can in principle be interesting but are quite dependent on the person*
- *Think it is great that one always gets help when asking the personnel. But maybe there should be a tool with a function that makes it easier for people to find books that they will like, for example a form on the computer that gives recommendations*
- *How books are arranged and displayed has a lot to say for what one will borrow and that there is room for the books*
- *Please keep the system as it is, acquainted with it and easily find what I want*
- *Would be best if the stacks were searchable multiple ways, e.g. arranged to be discoverable by genre, but item locations cross-referenced in an alphabetical by author database or guide*

12. Do you have any feedback for the library regarding their fiction collection?

- *Would be nice if all books were available as audiobooks but that is being worked on*
- *Not really. But I love the library, a nice place to be.*
- *Live in Bodø but like Tromsø library quite a lot*
- *Search system on ground floor, better map overview (more understandable)*

- *Librarians are very friendly*
- *That books for the elderly were available on the ground floor. It is very nice to be helped by the librarians. They are very helpful.*
- *I think this library is great*

Appendix C: Corpus Short Stories

1. J. M. Barrie – *The Courting of T'Nowhead's Bell*
2. Blicher – *Rector of Veilbye*
3. Kate Chopin – *The Story of an Hour*
4. Lars Saabye Christensen – *Apples*
5. Tore Coward – *Bad Loser*
6. Stephen Crane – *The Open Boat*
7. Alphonse Daudet – *The Last Lesson*
8. Jens Dimmen – *The Bus to Arkansas*
9. Théophile Gautier – *The Boy with Shoes of Bread*
10. Thomas Hardy – *An Imaginative Woman*
11. Francis Bret Harte – *The Luck of Roaring Camp*
12. Ebba Haslund – *No Ordinary Day*
13. Kjell Haug – *The Blow*
14. Finn Havrevold – *Late Evening in August*
15. Nathaniel Hawthorne - *Wakefield*
16. O. Henry – *The Last Leaf*
17. Aldous Huxley – *Nuns at Luncheon*
18. Aldous Huxley – *The Tillotson Banquet*
19. W. W. Jacobs – *The Monkey's Paw*
20. Margaret Johansen – *The Office Party*
21. James Joyce – *A Painful Case*
22. James Joyce - *Counterparts*
23. Rudyard Kipling – *Mary Postgate*
24. Wenche Krossøy - *Thirst*
25. Auguste Villiers de L'Isle Adam – *The Torture By Hope*
26. D. H. Lawrence – *A Sick Collier*
27. Henry Lawson – *The Union Buries Its Dead*
28. Stephen Leacock – *The Marine Excursions of the Knights of Pythias*
29. Katherine Mansfield – *The Garden Party*
30. Katherine Mansfield – *The Stranger*
31. Carson McCullers – *Wunderkind*
32. Prosper Mérimée – *Mateo Falcone*
33. Tillie Olsen – *I Stand Here Ironing*
34. Breece D'J Pancake – *First Day of Winter*
35. Edgar Allan Poe – *The Fall of the House of Usher*
36. Alain Robbe-Grillet – *The Secret Room*
37. Saki (Hector Hugh Munro) - *Tobermory*
38. Jean-Paul Sartre – *The Wall*
39. Odd Solumsmoen – *So Long As He Doesn't Act Up*
40. Håreik Storrud – *Uni*
41. Bjørg Vik – *Tone - 16*
42. H. G. Wells – *The Stolen Bacillus*
43. H. G. Wells – *The Star*
44. Oscar Wilde – *The Nightingale and the Rose*

Appendix D: Corpus Preparation in Microsoft Word

```
Sub V1SuperMacro()
'
' V1RemoveAllStopwords Macro
' Call remove punctuation macro. Make everything lowercase. Remove
paragraph marks.
' Remove numbers 0-9. Remove spaces
'
' Call remove punctuation macro
    Application.Run MacroName:="V1RemovePunctuation"
' Make everything lower case
    Selection.WholeStory
    Selection.Range.Case = wdLowerCase
' Remove paragraph marks
    With Selection.Find
        .Text = "^p"
        .Replacement.Text = " "
        .Forward = True
        .Wrap = wdFindContinue
        .Format = False
        .MatchCase = False
        .MatchWholeWord = False
        .MatchWildcards = False
        .MatchSoundsLike = False
        .MatchAllWordForms = False
    End With
    Selection.Find.Execute Replace:=wdReplaceAll
    With Selection.Find
        .Text = "^l"
        .Replacement.Text = " "
        .Forward = True
        .Wrap = wdFindContinue
        .Format = False
        .MatchCase = False
        .MatchWholeWord = False
        .MatchWildcards = False
        .MatchSoundsLike = False
        .MatchAllWordForms = False
    End With
    Selection.Find.Execute Replace:=wdReplaceAll
' Remove numerals 0-9
    With Selection.Find
        .Text = "[0-9]"
        .Replacement.Text = " "
        .Forward = True
        .Wrap = wdFindContinue
        .Format = False
        .MatchCase = False
        .MatchWholeWord = False
        .MatchWildcards = True
        .MatchSoundsLike = False
        .MatchAllWordForms = False
    End With
    Selection.Find.Execute Replace:=wdReplaceAll
' Remove extra spaces
    With Selection.Find
        .Text = "([ ])[ ]{1,}"
        .Replacement.Text = "\1"
        .Forward = True
        .Wrap = wdFindContinue
        .Format = False
        .MatchCase = False
```

```
        .MatchWholeWord = False
        .MatchWildcards = True
        .MatchSoundsLike = False
        .MatchAllWordForms = False
    End With
    Selection.Find.Execute Replace:=wdReplaceAll
End Sub


Sub V1RemovePunctuation()
'
' RemovePunctuation Macro
' Remove . , ! ? " ' - / () ; : {} [] … _ ~ and replace with space.
'
    Selection.Find.ClearFormatting
    Selection.Find.Replacement.ClearFormatting
    With Selection.Find
        .Text = "~"
        .Replacement.Text = " "
        .Forward = True
        .Wrap = wdFindContinue
        .Format = False
        .MatchCase = False
        .MatchWholeWord = False
        .MatchWildcards = False
        .MatchSoundsLike = False
        .MatchAllWordForms = False
    End With
    Selection.Find.Execute Replace:=wdReplaceAll
    With Selection.Find
        .Text = "_"
        .Replacement.Text = " "
        .Forward = True
        .Wrap = wdFindContinue
        .Format = False
        .MatchCase = False
        .MatchWholeWord = False
        .MatchWildcards = False
        .MatchSoundsLike = False
        .MatchAllWordForms = False
    End With
    Selection.Find.Execute Replace:=wdReplaceAll
    With Selection.Find
        .Text = "…"
        .Replacement.Text = " "
        .Forward = True
        .Wrap = wdFindContinue
        .Format = False
        .MatchCase = False
        .MatchWholeWord = False
        .MatchWildcards = False
        .MatchSoundsLike = False
        .MatchAllWordForms = False
    End With
    Selection.Find.Execute Replace:=wdReplaceAll
    With Selection.Find
        .Text = "—"
        .Replacement.Text = " "
        .Forward = True
        .Wrap = wdFindContinue
        .Format = False
        .MatchCase = False
        .MatchWholeWord = False
```

```
    .MatchWildcards = False
    .MatchSoundsLike = False
    .MatchAllWordForms = False
End With
Selection.Find.Execute Replace:=wdReplaceAll
With Selection.Find
    .Text = "—"
    .Replacement.Text = " "
    .Forward = True
    .Wrap = wdFindContinue
    .Format = False
    .MatchCase = False
    .MatchWholeWord = False
    .MatchWildcards = False
    .MatchSoundsLike = False
    .MatchAllWordForms = False
End With
Selection.Find.Execute Replace:=wdReplaceAll
With Selection.Find
    .Text = "."
    .Replacement.Text = " "
    .Forward = True
    .Wrap = wdFindContinue
    .Format = False
    .MatchCase = False
    .MatchWholeWord = False
    .MatchWildcards = False
    .MatchSoundsLike = False
    .MatchAllWordForms = False
End With
Selection.Find.Execute Replace:=wdReplaceAll
With Selection.Find
    .Text = ","
    .Replacement.Text = " "
    .Forward = True
    .Wrap = wdFindContinue
    .Format = False
    .MatchCase = False
    .MatchWholeWord = False
    .MatchWildcards = False
    .MatchSoundsLike = False
    .MatchAllWordForms = False
End With
Selection.Find.Execute Replace:=wdReplaceAll
With Selection.Find
    .Text = "!"
    .Replacement.Text = " "
    .Forward = True
    .Wrap = wdFindContinue
    .Format = False
    .MatchCase = False
    .MatchWholeWord = False
    .MatchWildcards = False
    .MatchSoundsLike = False
    .MatchAllWordForms = False
End With
Selection.Find.Execute Replace:=wdReplaceAll
With Selection.Find
    .Text = "?"
    .Replacement.Text = " "
    .Forward = True
    .Wrap = wdFindContinue
    .Format = False
```

```
    .MatchCase = False
    .MatchWholeWord = False
    .MatchWildcards = False
    .MatchSoundsLike = False
    .MatchAllWordForms = False
End With
Selection.Find.Execute Replace:=wdReplaceAll
With Selection.Find
    .Text = """"
    .Replacement.Text = " "
    .Forward = True
    .Wrap = wdFindContinue
    .Format = False
    .MatchCase = False
    .MatchWholeWord = False
    .MatchWildcards = False
    .MatchSoundsLike = False
    .MatchAllWordForms = False
End With
Selection.Find.Execute Replace:=wdReplaceAll
With Selection.Find
    .Text = "'"
    .Replacement.Text = " "
    .Forward = True
    .Wrap = wdFindContinue
    .Format = False
    .MatchCase = False
    .MatchWholeWord = False
    .MatchWildcards = False
    .MatchSoundsLike = False
    .MatchAllWordForms = False
End With
Selection.Find.Execute Replace:=wdReplaceAll
With Selection.Find
    .Text = "-"
    .Replacement.Text = " "
    .Forward = True
    .Wrap = wdFindContinue
    .Format = False
    .MatchCase = False
    .MatchWholeWord = False
    .MatchWildcards = False
    .MatchSoundsLike = False
    .MatchAllWordForms = False
End With
Selection.Find.Execute Replace:=wdReplaceAll
With Selection.Find
    .Text = "/"
    .Replacement.Text = " "
    .Forward = True
    .Wrap = wdFindContinue
    .Format = False
    .MatchCase = False
    .MatchWholeWord = False
    .MatchWildcards = False
    .MatchSoundsLike = False
    .MatchAllWordForms = False
End With
Selection.Find.Execute Replace:=wdReplaceAll
With Selection.Find
    .Text = "("
    .Replacement.Text = " "
    .Forward = True
```

```
    .Wrap = wdFindContinue
    .Format = False
    .MatchCase = False
    .MatchWholeWord = False
    .MatchWildcards = False
    .MatchSoundsLike = False
    .MatchAllWordForms = False
End With
Selection.Find.Execute Replace:=wdReplaceAll
With Selection.Find
    .Text = ")"
    .Replacement.Text = " "
    .Forward = True
    .Wrap = wdFindContinue
    .Format = False
    .MatchCase = False
    .MatchWholeWord = False
    .MatchWildcards = False
    .MatchSoundsLike = False
    .MatchAllWordForms = False
End With
Selection.Find.Execute Replace:=wdReplaceAll
With Selection.Find
    .Text = ":"
    .Replacement.Text = " "
    .Forward = True
    .Wrap = wdFindContinue
    .Format = False
    .MatchCase = False
    .MatchWholeWord = False
    .MatchWildcards = False
    .MatchSoundsLike = False
    .MatchAllWordForms = False
End With
Selection.Find.Execute Replace:=wdReplaceAll
With Selection.Find
    .Text = ";"
    .Replacement.Text = " "
    .Forward = True
    .Wrap = wdFindContinue
    .Format = False
    .MatchCase = False
    .MatchWholeWord = False
    .MatchWildcards = False
    .MatchSoundsLike = False
    .MatchAllWordForms = False
End With
Selection.Find.Execute Replace:=wdReplaceAll
With Selection.Find
    .Text = "["
    .Replacement.Text = " "
    .Forward = True
    .Wrap = wdFindContinue
    .Format = False
    .MatchCase = False
    .MatchWholeWord = False
    .MatchWildcards = False
    .MatchSoundsLike = False
    .MatchAllWordForms = False
End With
Selection.Find.Execute Replace:=wdReplaceAll
With Selection.Find
    .Text = "]"
```

```
        .Replacement.Text = " "
        .Forward = True
        .Wrap = wdFindContinue
        .Format = False
        .MatchCase = False
        .MatchWholeWord = False
        .MatchWildcards = False
        .MatchSoundsLike = False
        .MatchAllWordForms = False
    End With
    Selection.Find.Execute Replace:=wdReplaceAll
    With Selection.Find
        .Text = "{"
        .Replacement.Text = " "
        .Forward = True
        .Wrap = wdFindContinue
        .Format = False
        .MatchCase = False
        .MatchWholeWord = False
        .MatchWildcards = False
        .MatchSoundsLike = False
        .MatchAllWordForms = False
    End With
    Selection.Find.Execute Replace:=wdReplaceAll
    With Selection.Find
        .Text = "}"
        .Replacement.Text = " "
        .Forward = True
        .Wrap = wdFindContinue
        .Format = False
        .MatchCase = False
        .MatchWholeWord = False
        .MatchWildcards = False
        .MatchSoundsLike = False
        .MatchAllWordForms = False
    End With
    Selection.Find.Execute Replace:=wdReplaceAll
End Sub
```

Appendix E: Example Python Code

```
import os
from gensim.parsing import stem_text
from gensim import corpora, models, similarities

#create class to read in documents (one per line in txt file), split
them into tokens word for word, and stem using the Porter stemming
algorithm

class StemCorpus(object):
      def __iter__(self):
            for line in open("original corpus.txt"):
                  yield stem_text(line).split()

#create object to read in docs when needed

corpusmf = StemCorpus()

#create dictionary from stemmed, tokenized corpus

dictionary = corpora.Dictionary(corpusmf)

#create stop list from List C, give stopwords ids from dictionary

stop list = stem_text('a about after again all an and are as at away
b back be been before but by c can come could d do down e even eye f
face first for from g go good h had hand have he head her his him how
i if in into is it j just k know l last like little long look m made
make man more much my n no not now o of on only open or other out
over p put q r s say said see so some something still t that the they
their them then there this thing thought through time to too two u up
us v very w was way we went were what when where who with would x y
you z').split()
stop_ids = [dictionary.token2id[stopword] for stopword in stop list
if stopword in dictionary.token2id]

#create list of singletons with ids from dictionary

once_ids = [tokenid for tokenid, docfreq in
dictionary.dfs.iteritems() if docfreq == 1]

#remove stopwords and singletons

dictionary.filter_tokens(stop_ids + once_ids)
dictionary.compactify()

#save dictionary

dictionary.save("dictionary.dict")
dictionary.save_as_text("readable dictionary.txt")

#changing the corpus using the bag of words model

corpus = [dictionary.doc2bow(text) for text in corpusmf]

#save corpus in market matrix format

corpora.MmCorpus.serialize ("corpus.mm", corpus)

#transform corpus using TF-IDF

tfidf = models.TfidfModel(corpus)
```

```
corpus_tfidf = tfidf[corpus]

#wrap TF-IDF transformation in LSI transformation, change num_topics
to desired value of k

lsi = models.LsiModel(corpus_tfidf, id2word=dictionary, num_topics=2)
corpus_lsi=lsi[corpus]

#calculate cosine similarity between document vectors

index = similarities.MatrixSimilarity(corpus_lsi)
index.save("index.index")
for similarity in index:
      print similarity

#exit

os._exit(1)
```

Appendix F: Stopword Lists

List A

a about above across after afterwards again against all almost alone along already also although always am among amongst amoungst amount an and another any anyhow anyone anything anyway anywhere are aren arent around as at
b back be became because become becomes becoming been before beforehand behind being below beside besides between beyond bill both bottom but by
c call can cannot cant co computer con could couldn couldnt cry
d de describe detail did didn didnt do does doesn doesnt done don dont down due during
e each eg eight either eleven else elsewhere empty enough etc even ever every everyone everything everywhere except
f few fifteen fifty fill find fire first five for former formerly forty found four from front full further
g get give go
h had hadn hadnt has hasn hasnt have haven havent he hence her here hereafter hereby herein hereupon hers herself him himself his how however hundred
i ie if in inc indeed interest into is isn isnt it its itself
j
k keep
l last latter latterly least less ll ltd
m made many may me meanwhile might mightn mightnt mill mine more moreover most mostly move much must mustn mustnt my myself
n name namely neither never nevertheless next nine no nobody none noone nor not nothing now nowhere
o of off often on once one only onto or other others otherwise our ours ourselves out over own
p part per perhaps please put
q
r rather re
s said same see seem seemed seeming seems serious several shall shan shant she should shouldn shouldnt show side since sincere six sixty so some somehow someone something sometime sometimes somewhere still such system
t take ten than that the their them themselves then thence there thereafter thereby therefore therein thereupon these they thick thin third this those though three through throughout thru thus to together too top toward towards twelve twenty two
u un under until up upon us
v ve very via
w was we well were weren werent what whatever when whence whenever where whereafter whereas whereby wherein whereupon wherever whether which while whither who whoever whole whom whose why will with within without would wouldn wouldnt
x
y yet you your yours yourself yourselves
z

List B

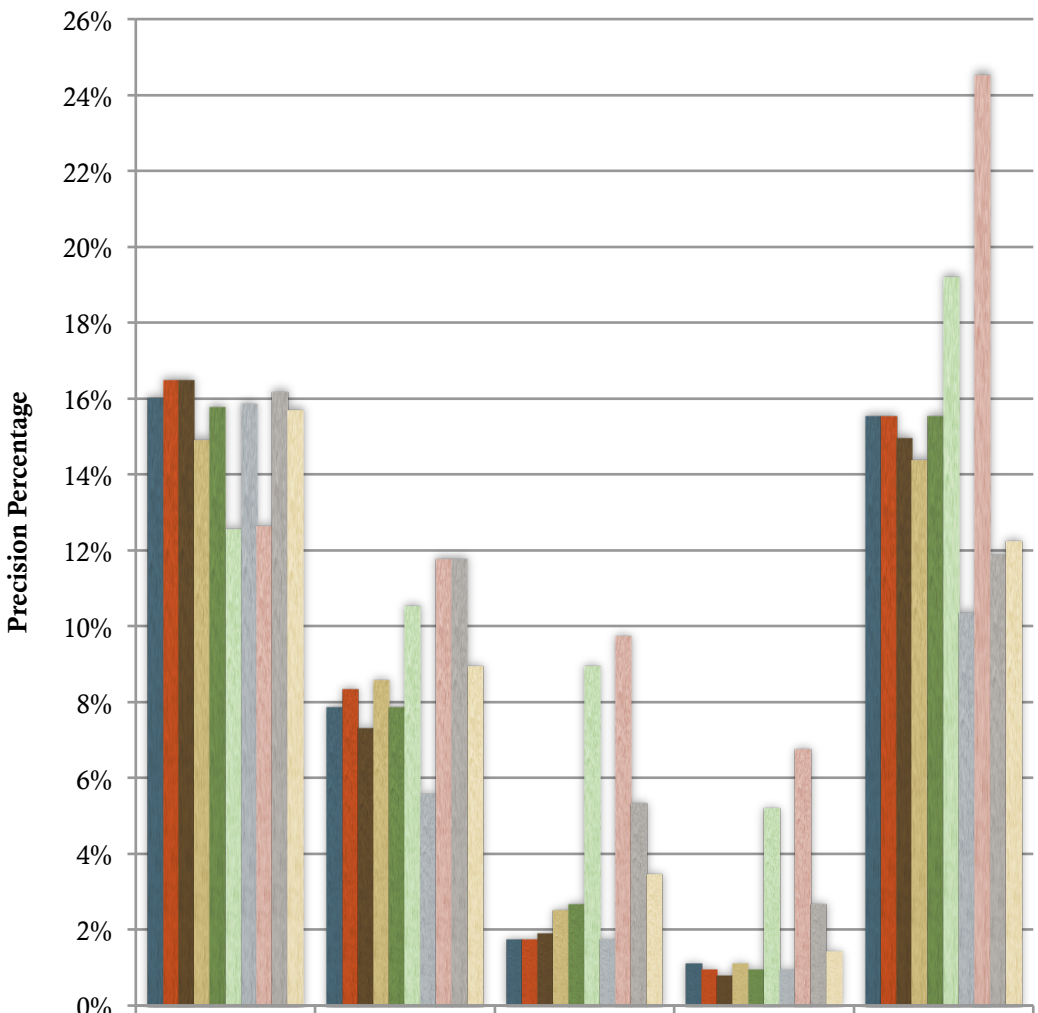
a about actually after again against all almost already always am an and another answer any anything are arm around as ask at away
b back be because bed been before began behind better big bit black both boy but by
c call came can cannot car close come could couldn
d dad day did didn do don door down
e each else end enough even ever every everyone everything eye
f face feel felt few finally find first for found friend from front
g gave get girl give go good got guess guy

h had hair hand happen hard has have he head hear heard help her here hey him himself his home house how
i if in into is it
j just
k keep kiss knew know
l last late laugh leave left let life light like little live ll long look lot love
m made make man maybe me mean mind minute mom moment more most mother move much my myself
n name need never new next night no nod not nothing now
o of off oh okay old on once one only open or other our out over own
p people pick place play point probably pull put
q quickly
r re really remember reply right room run
s said same sat saw say school second see seem she should shoulder side sigh since sit small smile so some someone something soon sorry sound stand stare start step still stood stop sure
t table take talk tell than thank that the their them then there they thing think this those though thought three through time to told too took toward try turn two
u until up us use
v ve very voice
w wait walk want was wash wasn watch way we well went were what when where which while who why will with word work would wouldn
x
y yeah year yes you your
z

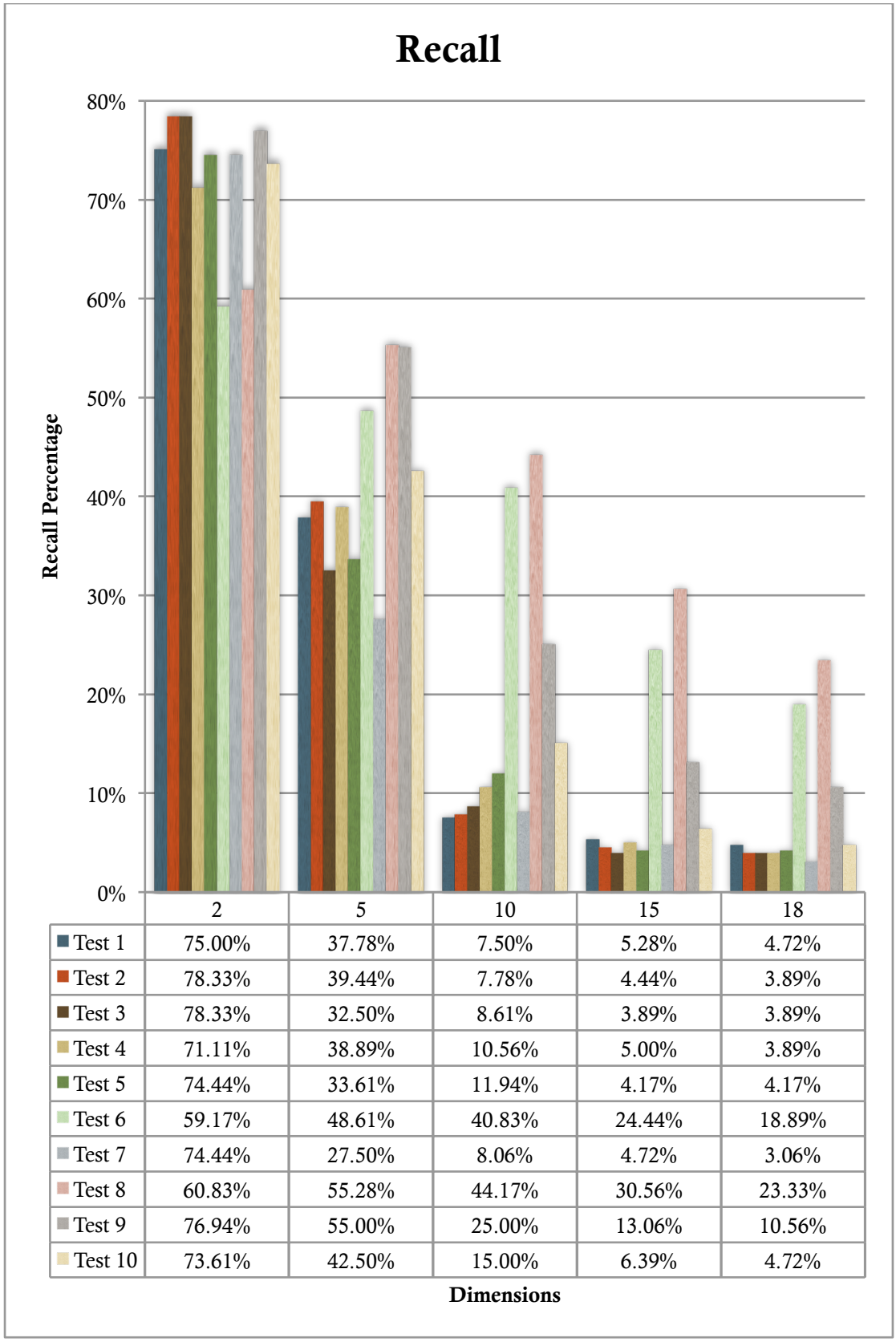
List C

a about after again all an and are as at away
b back be been before but by
c can come could
d do down
e even eye
f face first for from
g go good
h had hand have he head her his him how
i if in into is it
j just
k know
l last like little long look
m made make man more much my
n no not now
o of on only open or other out over
p put
q
r
s say said see so some something still
t that the they their them then there this thing thought through time to too two
u up us
v very
w was way we went were what when where who with would
x
y you
z

Appendix G: LSI Test Results



	2	5	10	15	18
Test 1	15.99%	7.84%	1.72%	1.10%	15.52%
Test 2	16.46%	8.31%	1.72%	0.94%	15.52%
Test 3	16.46%	7.29%	1.88%	0.78%	14.94%
Test 4	14.89%	8.54%	2.51%	1.10%	14.37%
Test 5	15.75%	7.84%	2.66%	0.94%	15.52%
Test 6	12.54%	10.50%	8.93%	5.17%	19.17%
Test 7	15.83%	5.56%	1.72%	0.94%	10.34%
Test 8	12.62%	11.76%	9.72%	6.74%	24.49%
Test 9	16.14%	11.76%	5.33%	2.66%	11.86%
Test 10	15.67%	8.93%	3.45%	1.41%	12.21%



	2	5	10	15	18
Test 1	75.00%	37.78%	7.50%	5.28%	4.72%
Test 2	78.33%	39.44%	7.78%	4.44%	3.89%
Test 3	78.33%	32.50%	8.61%	3.89%	3.89%
Test 4	71.11%	38.89%	10.56%	5.00%	3.89%
Test 5	74.44%	33.61%	11.94%	4.17%	4.17%
Test 6	59.17%	48.61%	40.83%	24.44%	18.89%
Test 7	74.44%	27.50%	8.06%	4.72%	3.06%
Test 8	60.83%	55.28%	44.17%	30.56%	23.33%
Test 9	76.94%	55.00%	25.00%	13.06%	10.56%
Test 10	73.61%	42.50%	15.00%	6.39%	4.72%