2009-09

Topic detection in online chat

Durham, Jonathan S.

Monterey, California. Naval Postgraduate School

NAVAL POSTGRADUATE SCHOOL

MONTEREY, CALIFORNIA

THESIS

TOPIC DETECTION IN ONLINE CHAT

by

Jonathan S. Durham

September 2009

Thesis Advisor:	Craig H. Martell
Second Reader:	Andrew I. Schein

Approved for public release, distribution is unlimited

THIS PAGE INTENTIONALLY LEFT BLANK

REPORT DOCUMENTATION PAGE *Form Approved OMB No. 0704-0188*

1. AGENCY USE ONLY *(Leave blank)*	**2. REPORT DATE** September 2009	**3. REPORT TYPE AND DATES COVERED** Master's Thesis

4. TITLE AND SUBTITLE Topic Detection in Online Chat

5. FUNDING NUMBERS

6. AUTHOR(S) Jonathan S. Durham

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Naval Postgraduate School
Monterey, CA 93943-5000

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING /MONITORING AGENCY NAME(S) AND ADDRESS(ES)
N/A

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.

12a. DISTRIBUTION / AVAILABILITY STATEMENT
Approved for public release; distribution is unlimited

12b. DISTRIBUTION CODE

13. ABSTRACT (maximum 200 words)

The ubiquity of Internet chat applications has benefited many different segments of society. It also creates opportunities for criminal enterprise, terrorism, and espionage. This thesis proposes statistical Natural Language Processing (NLP) methods for creating systems that would detect the topic of chat in support of larger NLP goals such as information retrieval, text classification and illicit activity detection.

We propose a novel method for determining the topic of chat discourse. We trained Latent Dirichlet Allocation (LDA) models on source documents and then used inferred topic distributions as feature vectors for a Support Vector Machine (SVM) classification system. We constructed LDA models in three ways: We considered the collective posts of authors as documents, hypothesizing that we could detect the topic physics given only one side of the conversation. The resultant classifiers obtained F-scores of 0.906. Next, we considered individual posts as documents, hypothesizing we could detect physics posts. The resultant classifiers obtained F-scores of 0.481. Finally, we considered physics textbook paragraphs as documents, hypothesizing that we could determine the topic of an author or a post based on an LDA model created from a textbook and a sample of noisy chat. The resultant classifiers obtained F-scores of 0.848 and 0.536 respectively.

14. SUBJECT TERMS
Latent Dirichlet Allocation, Support Vector Machine, Natural Language Processing, Chat, Topic Detection

15. NUMBER OF PAGES
103

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	**18. SECURITY CLASSIFICATION OF THIS PAGE**	**19. SECURITY CLASSIFICATION OF ABSTRACT**	**20. LIMITATION OF ABSTRACT**
Unclassified	Unclassified	Unclassified	UU

THIS PAGE INTENTIONALLY LEFT BLANK

TOPIC DETECTION IN ONLINE CHAT

Jonathan S. Durham
Lieutenant, United States Navy
B.S., United States Naval Academy, 2001

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN COMPUTER SCIENCE

from the

**NAVAL POSTGRADUATE SCHOOL
September 2009**

Author: Jonathan S. Durham

Approved by: Craig H. Martell
Thesis Advisor

Andrew I. Schein
Second Reader

Peter J. Denning
Chairman, Department of Computer Science

THIS PAGE INTENTIONALLY LEFT BLANK

ABSTRACT

The ubiquity of Internet chat applications has benefited many different segments of society. It also creates opportunities for criminal enterprise, terrorism, and espionage. This thesis proposes statistical Natural Language Processing (NLP) methods for creating systems that would detect the topic of chat in support of larger NLP goals such as information retrieval, text classification and illicit activity detection.

We propose a novel method for determining the topic of chat discourse. We trained Latent Dirichlet Allocation (LDA) models on source documents and then used inferred topic distributions as feature vectors for a Support Vector Machine (SVM) classification system. We constructed LDA models in three ways: We considered the collective posts of authors as documents, hypothesizing that we could detect the topic physics given only one side of the conversation. The resultant classifiers obtained F-scores of 0.906. Next, we considered individual posts as documents, hypothesizing we could detect physics posts. The resultant classifiers obtained F-scores of 0.481. Finally, we considered physics textbook paragraphs as documents, hypothesizing that we could determine the topic of an author or a post based on an LDA model created from a textbook and a sample of noisy chat. The resultant classifiers obtained F-scores of 0.848 and 0.536, respectively.

THIS PAGE INTENTIONALLY LEFT BLANK

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF ACRONYMS AND ABBREVIATIONS

NLP	Natural Language Processing
ML	Machine Learning
LDA	Latent Dirichlet Allocation
SVM	Support Vector Machines
pLSA	Probabilistic Latent Semantic Analysis
pLSI	Probabilistic Latent Semantic Indexing
LSA	Latent Semantic Analysis
LSI	Latent Semantic Indexing
CMC	Computer Mediated Communication
SMS	Short Message Service
SNS	Social Networking Service
ASR	Automatic Speech Recognition
ASU	Automatic Speech Understanding
IR	Information Retrieval
IA	Information Assurance
tf-idf	Term Frequency-Inverse Document Frequency
IS	Information Science
SVD	Singular Value Decomposition
EM	Expectation Maximization
MCMC	Markov chain Monte Carlo
KKT	Karush Kuhn Tucker
KL	Kullback-Leibler
GMM	Gaussian Mixture Model
Mallet	MAchine Learning for LanguagE Toolkit

THIS PAGE INTENTIONALLY LEFT BLANK

ACKNOWLEDGMENTS

There have been many people who have helped me with this thesis. I owe them great thanks.

My thesis advisor, Prof. Craig Martell, who I'm certain still has some experiments he wants me to run.

Prof. Kevin Squire, who is a very good teacher and very helpful.

Prof. Andrew Schein, who helped me complete this work.

David Dreier, who is king nerd.

Jenny Tam, who is a robot or a ninja or a robot ninja.

Johnnie Caver, who is refreshingly honest.

Brian Hawkins, who is a math junky.

Marco Draeger, who talks funny, has Mallet and knows how to use it.

Juice Newton, Constantine Perepelitsa, king of sarcasm, and handy with the linux.

Mrs. Sue Higgins, who knows everyone and takes time to listen and share.

Captain Danelle Barrett, USN, who has been a constant source of encouragement for me throughout my career.

My wife, who sustains me even from afar, an amazing woman. All that I have is found in you.

" … Of making many books there is no end, and much study is a weariness of the flesh."

Ecclesiastes 12:12 , ESV

THIS PAGE INTENTIONALLY LEFT BLANK

I. INTRODUCTION

A. MOTIVATION

One would have to be cut off from society not to have noticed the growing immergence of Computer Mediated Communication (CMC). Forms of CMC such as chat, blogs, email, and Short Message Service (SMS) all permeate the daily lives of people nearly everywhere. Chat in particular is being used as a means of socializing, marketing goods, providing services, and creating value in business. On the other hand, chat has been used to coordinate terrorist activities, recruit new terrorists, conduct criminal activity and victimize children [1]. Systems that are able to detect the topic of a conversation could be very useful for law enforcement, national security agents, corporations, and parents seeking to maintain control of their children's online activities.

Many organizations could benefit from these systems. Military organizations use chat in tactical and non-tactical environments. They could use a topic predicting system to detect information leaks over unclassified networks. Law enforcement would find it useful in sifting through chat during criminal investigations. National security agents could monitor chat for topics of interest such as bomb making, and anti-American sentiment. Parents could monitor the type of conversations their children are having with others. They could also use it to prevent online bullying or sexual predation. Businesses could use such systems for protecting trade secrets, ensuring employees make the best use of their time, or trying to determine what consumers are interested in based on their topic of conversation. Business may also be interested in protecting their public image or at least monitoring it by learning of unfounded rumors spread through chat. Libraries could use the system to determine what areas of study people are researching, and then plan accordingly.

Beyond these practical motivations, there is a deeper interest involved, embodied in the question: How is it that humans are able to use symbols to convey meaning? Furthermore, how is it that different types of symbols are able to perform the same conveyance regardless of the mode—be it sound, little bumps on paper, ink on a piece of paper or differences in light intensity on a computer monitor? There exists a relationship between a book, speech, a few lines of chat and sign language. What is the nature of that relationship? Particularly, can abstractions such as topic be found beyond the borders of the mode, or must they necessarily be expressed and extracted?

What is a topic? Ask several friends what a particular song is *about*. You will likely get several different answers. Ask the same people on a different day, you will probably get a different set of answers. You have probably been in a disagreement with friends about the topic of such a song. This thesis does not settle that argument or resolve the matter of what a topic is. Instead, it explores a small segment of what it means to find a topic within chat. In our experiments, we seek first to create statistical models that naively agree with human beings about the topic of physics using only past examples of things people have called physics, using statistical Natural Language Processing (NPL) techniques. Then, we explore how to leverage another mode of human communication to determine the topic in chat. We use a physics text to try to determine whether people in a chat room are discussing physics. Hopefully, this research will help us understand human language more fully, and lead to systems that are able to automatically detect topics in chat.

B. STRUCTURE OF THIS THESIS

This thesis is organized in the following manner. In Chapter I, we discuss the various motivations for determining the topic of chat. In Chapter II, we provide a summary of previous research in this field. We include: 1) an analysis of chat as it relates to other modes of human communication, most prominently, traditional written text, and speech; 2) a brief survey of NLP and its relation to Chat; 3) an example of the types of debates regarding the uncertain definition of Topicality and its relation to our research; and 4) a discussion of the machine learning techniques we employ, and measures of evaluating their results. In Chapter III, we discuss the experimental methodology employed. We include: 1) the sources of data used in our experiments; 2) the way in which we intend to use the data to perform four classes of experiments, which are classification tasks; and 3) how we determine settings for the NLP models that we have chosen. In Chapter IV, we provide the results of all four types of experiments. We include a detailed discussion of the primary performance parameters observed as a result of the experiments and what their relationships to the other experiments entail. In Chapter V, we summarize the results of the experiments and propose future work.

THIS PAGE INTENTIONALLY LEFT BLANK

II. BACKGROUND

Chat is an increasingly important form of CMC. It is employed by many sectors of society to improve communication, create value, and commit crimes. In this chapter, we explore chat first as it relates to other human language modalities. Then, we explore Natural Language Processing (NLP) and its goals, followed by its applicability to chat. Finally, we discuss the idea of topicality and previous Machine Learning (ML) techniques used to detect topics in chat.

The objective of automatically revealing the topic of any form of communication is twofold. The first motive is to increase the knowledge of how humans communicate, to unravel the mystery of information conveyance. The second is to build useful systems. Topic detection in this context is a step toward automating tasks that would otherwise be untenable, because the sheer volume of data makes it impractical. Section A provides working definitions of CMC, chat, and natural human languages.

A. CHAT'S RELATIONSHIP TO OTHER NATURAL LANGUAGE CONSTRUCTS

This work considers chat to be a mode of natural language. Natural language is simply a language that humans speak [2]. This definition is a bit restrictive because it does not account for written text, sign language, Braille and manual languages developed by deaf-blind people. However, it is helpful in the sense that it conveys the origin of language—people. Speaking, writing, Braille, signing, and CMC are all methods that represent the innovations and phenomena of natural language. We briefly compare and contrast chat to two of these modes—speaking and traditional writing.

According to Herring, CMC is “communication that takes place between human beings via the instrumentality of computers” [3]. CMC itself takes many forms to include e-mail, Weblogs (blogs), micro-blogs, video, audio, text chat, text messaging, instant messaging, bulletin boards, and list-servs. Social networking services (SNS) and online games also use computers and computer networks to mediate communication by combining many CMC technologies. Many online gaming venues feature some chat functionality or audio service. SNSs combine nearly all of these CMC forms.

Text chat is a subcategory of chat that is a near-synchronous form of CMC. Chat may be categorized as text chat, voice chat and video chat. This research is dedicated solely to text chat. Any further reference to chat will mean text chat. Further, we narrow our meaning of chat to that which is near-synchronous, multi-member conversation contributed and conversation interleaved. *Near-synchronous* means that conversation contributors interact in near real-time. They are temporally proximate to each other. *Multi-member conversation contributed* refers to the fact that there may be more than two people contributing to the chat statements (posts) at the same time. *Conversationally interleaved* indicates that the many conversations (threads) may occurring at the same time.

Chat communication is affected significantly by its technological implementation—its computer mediation [4]. All chat implementations share some common characteristics. First, there is a main dialog. It is a relatively large text field, which displays the *posts* created by all chat room participants. The main dialog is public to all. If a poster wishes to “say” something, the *main dialog* is where it is displayed for all to see [4]. Second, there is a *personal dialog*. This is a text field where each user composes his or her particular posts. It is, for the most part, private to each particular user.[1] After the user has completed composing her message, she posts the message to the main dialog. The message she composed in private is now public to all participants [4].

Implementations vary significantly, but these are chat’s essential properties. Any implementation may seem similar to other forms of CMC and

[1] Features of some chat software implementations allow other users to view whether or not individuals are in the act of typing which reveals something about the user—it is not totally private.

somewhat similar to writing a letter or having a speaking conversation. Varying degrees of these characteristics, along with a fair amount of anonymity, profoundly and uniquely determine the way chat participants interact, and hence, the way chat must be analyzed linguistically.

Those who have participated in chat forums know that people do not use chat the same way they use spoken and traditional written communication. The chatter's lexicon includes emoticons and unusual spellings of common words and phrases [4]. These are uncommon or altogether absent from traditional written language and spoken language. Chatters do not have the same visual or auditory cues that usually accompany speaking. Chatting and speaking are usually synchronous[2] [4], whereas traditional writing usually is not.[3] Speaking is usually a one-to-one or one-to-many act. Writing is usually a one-to-one or one-to-many. Chatting offers participants a method for one-to-one, one-to-many, many-to-one, or many-to-many conversations, whether the participants wish it or not.

Beyond these intuitions, linguists and sociologists have engaged in serious study of language and conversation in the context of these three domains (written, spoken, and chatted). Forsyth uses Nystrand's [4] constructions of *Context of Production* and *Context of Use* to demonstrate that chat technology causes notable distinctions between chat and the other communication domains. Forsyth quotes Zitzen and Stein's [5] characterization of spoken language and written language in the following ways. First, face-to-face spoken language is characterized by *concurrent* contexts of production and use [4]. Thus, participants monitor and interact with each other as the conversation unfolds. Second, traditional writing is characterized by contexts of production and use that

[2] Speeches are forms of spoken communication that is not always synchronous.

[3] Passing notes is a type of written communication that is synchronous—we do not include this in our definition of traditional writing.

are not concurrent [4]. Rather, they are separate spatiotemporally. Thus, participants are unable to monitor and interact with each other as the conversation develops [4].

Forsyth applies this analysis to chat, asserting that chat's private personal dialog serves as the chatter's *Context of Production* and "the public main dialog functions as [the] *Context of Use"* [4]. On one hand, he observes that chat is like written language in that its *Context of Production* is private and therefore the contexts of production and use are not concurrent. On the other hand, chat's *Context of Use*, which is public, is like spoken language because the conversation participants interact proximately in time. Like speakers, chatters are able to use paralinguistic information[4] [5] in their discourse.

The disparity between *Context of Production* and *Context of Use* in the different domains provides the chatter with a situation where she must maintain a conversation or multiple simultaneous conversations. She must post enough per unit time in order to avoid a great deal of "silence" and yet she must type her messages, which costs time [4]. This is a situation where the writer would have no time constraint and a speaker would have no problem "filling the space" between utterances. Chatters overcome the problem by a number of mechanisms.

A chatter may read and write, or read and wait at the same time, because other chatters are similarly engaged in typing as well [4]. Such delays allow the chatter to switch between multiple conversations easily because she has time, which would normally be reserved for formulating discourse or responding to paralinguistic communication in a spoken situation. This situation is convenient because it allows the chat participant to engage in multiple conversations without being considered rude [4].

[4] Paralinguistic information—information conveyed by something other than the meaning of words. In spoken language—intonation, body language, etc. In chat—bolding, all capitalization, emoticons, capitalization, font type, font color, etc.

Chatters tend to balance their level of activity in order to maintain the conversation, while not “hogging the conversation” [4]. This favors a shorter more precise post. A short post does not take much time to construct and is more likely [6] to appear in the main dialog next to the post to which it is responding or interacting. Long posts take time to create and fill up the screen, in effect pushing out the other conversation contributors. Such posts are therefore not favored [4]. Other methods include ways of constructing many small posts instead of one long post [4, 5].

Forsyth citing Freiermuth [7] notes additional differences between chat and the other domains caused by the unique time constraints imposed and allowed by chat [4]. Chatters, having more time to interact conversationally than speakers, tend to include more varied and creative words and more word forms. They may also consider their words more wisely and “elevate” their vocabulary more than in a speaking situation. The chatters, however, have less time than a writer, and thus simplify their sentence structures. Sometimes this is also the result of technological restrictions for some applications have a maximum character length per post. The demands of relatively fast conversation and multiple conversation occurring at the same time, place the chatter in a position where speed is usually preferable to precision [4].

The chatter is liberated to use language and behave conversationally in a way that suits her, because there is little social recourse for not adhering to written language norms [4]. They use colloquialisms and contractions. The chatter does not “hedge” her statements: indicate whether or not she is satisfied with her word selection. They may leave the conversation or chat room with very little warning—anathema in spoken conversation. They may engage in long periods of silence or selective silences[5] that would not be tolerated in face-to-face conversations [4].

[5] When one member of a conversation deliberately does not speak to another member of the conversation.

Chat relates to other types of human language use, but its distinctive properties create challenges for NLP practitioners. In the following section we outline NLP, NLP tasks, chat features and previous efforts to use NLP methods to solve chat related problems.

B. NLP AND CHAT

NLP problems are approached in two basic ways—rules-based NLP and statistical NLP. Rules based NLP practitioners assume that humans possess a great deal of underlying knowledge of language that allows humans to learn particular languages. Their objective is to model these mental processes in order to create a system that mimics or duplicates the functioning of the human brain. This process usually starts with creating rules that mimic such functionality [8].

Statistical NLP practitioners agree with the rules-based-inclined in that they proceed from the assumption that humans possess something innate that enable them to recognize patterns, which allows humans to learn how to communicate. They, however, differ on the degree to which humans possess this ability [8]—the former more and the latter less. Statistical NLP's general approach is to build statistical models of language and then use ML techniques to validate those models. This research approaches the problems using statistical NLP methods.

NLP is used in problems, which involve spoken language or speech after automatic speech recognition (ASR) has been applied. ASR seeks to build a mapping between sounds and strings. Automatic speech understanding (ASU) takes this goal one-step further and tries to understand the words in the broader context of a sentence [2]. Like ASU, conversational agents leverage NLP. Conversational agents such as SGT Star[6], a U.S. Army avatar that chats with potential Army enlistees, receives chat input from users and outputs textually, visually and audibly. Just as with telephone menus, SGT Star's capabilities are limited, but indicate an ideal direction for such systems.

[6] Available from U.S. Army Web site at http://www.goarmy.com/ChatWithStar.do.

Some NLP goals are relate to improving the language experience or use. For instance, most word processors include a spell checker and grammar checker, enabling thesis students everywhere the opportunity to reduce time in editing. These two types of systems are emblematic of lexical and syntactic analysis, which capture the way symbology is used in language. Unfortunately, human language does not follow simple and static rules, which makes lexical and syntactic analysis difficult, making these systems unwieldy and unreliable. Language scientists face challenging problems such as reference resolution, pronoun resolution and word sense disambiguation that must be addressed in lexical and syntactic analysis. Often the semantics of language interact with the lexical and syntactic components of the language, which causes even greater difficulty in disambiguation.

NLP addresses each of these problems individually for improving the use of language and it addresses them collectively in support of larger goals such as document and text classification, author classification, and discourse analysis.[7] The results of these goals may be used alone or in concert to build Information Retrieval (IR) systems, Information Assurance (IA) systems and machine translation systems.[8]

Document classification or categorization is a common goal in NLP. Its goal is to divide documents into different categories based on the characteristics of the document. Research has been put into categorizing news articles into different topical categories [9], such as health, sports, entertainment. One example of such research is Cohen et al.'s attempt to classify email messages into categories based on their motive for being sent. He analyzed emails according to four types of business acts: request, proposal, delivery, and commitment [10]. Document classification is often a precursor to the broader goals of IR, IA and criminal activity detection.

.[7] Classifying statements as parts of a conversation or conversations.

[8] Automatically converting one language to another, e.g., English to Spanish.

Author classification or author profiling addresses questions that relate to the attributes of an author, given some sample of the author's use of language. For instance, given a set of Weblog posts, chat posts or text documents from a single author, is it possible to identify the author, the author's gender or the author's age? Subsets of author classification include stylometry and authorship attribution. Author classification has often been used to study the authenticity of documents such as the works of Shakespeare. In the case of chat, law enforcement officials may be interested in using author classification to determine whether a chat participant who is chatting with a young person is lying about his or her age. Such behavior may be indicative of sexual predation [11], [12] or other malevolence.

Information Retrieval (IR) is a board area of research, which has received a great deal of focus. It considers the problem of storage and retrieval of data. A common use of IR is the Internet search engine. Minimally, such technology takes a user's request in the form of a set of words and returns Internet resources—Web sites, audio, video, image, and document data. For such IR applications, NLP may provide improved results using techniques such as word sense disambiguation, word stemming[9] and document classification.

IA is also a broad area of research, but not often thought of in the context of NLP. NLP may be able to aid in preventing information leaks by detecting topics that an organization considers sensitive [13], [1]. Once the topic is detected, its transmission may be halted or the offender questioned. More sophisticated systems might be able deduce particular pieces of information that are undetectable based on topic such as technical information embedded in a discussion about entertainment. The military may be more interested in controlling information about locations of military assets or future troop movements. Such things can be discussed in ways that are difficult to detect,

[9] Stemming means to determine the root of a word (e.g., run is the root of running, and ran).

given the broad range of possible topical contexts in which they may reside. Such system outputs might also be indicators of subversive behaviors.

CMC has become a hot bed for criminal activity. Terrorists, sexual predators, organized crime, bullies, prostitution rings and others use CMC, chat in particular, to commit and plan criminal activity [1]. A recent case of online bullying resulted in one child taking her life [14]. The SNS, Craigslist, is often used by prostitution rings to solicited sex [15]. Terrorists are known to have used the Internet as a medium for recruitment, and planning and coordination of terror acts [16]. Sexual predators frequently use chat rooms to connect with other predators and arrange meetings with their potential victims [17]. NLP offers law enforcement officials an automated tool for detecting such activities by text classification or by author classification. Parents and concerned citizens may also find such tools useful for preventing victimization were they to be used in monitoring applications [13].

C. NLP DIFFICULTIES

The problem for NLP scientists is that human language is full of ambiguity. Many words may have the same meaning and a single word may have many meanings. Sentence structure, the valid sequence of parts of speech, may have more than one reasonable structure—making it difficult to disambiguate the sentence meaning. Pronouns may have many plausible and equally likely antecedents. People handle these and many other ambiguities very well, but they pose problems for machines. This is the challenge for NLP in general and is particularly difficult in chat, where parts of speech have not yet been well studied and formalized. Statistical NLP practitioners are able to side-step formalization, by using of statistical ML methods. They presume that language phenomena follow statistical laws and therefore many of the hidden properties are accessible via probabilistic means.

D. THE FEATURE VECTOR

The feature vector is the heart of nearly every statistical NLP application. The feature vector allows the scientist the ability to abstract linguistic objects such as words, sentences, phrases, documents, topics, etc. into a mathematical framework. A scientist may then use these models to test various hypotheses using ML techniques. For example, a simple model may consider a document as a probability distribution of words over a common dictionary. A dictionary in this case would be the set of all unique word level linguistic objects—*types*. In this case, the feature vector is composed of real values ranging from zero to one, whose total is one—the probability of each *type*. The dimensionality of such a vector is the size of the dictionary. Just from this simple model, one may now train several different types of classifiers such as Naïve Bayes, Support Vector Machine and k-means clustering.

The feature vector is not limited to words. Anything can be considered a feature vector so long as it can be distinguished from other things, counted or measured. In Lin's master's thesis she used average post length, vocabulary size, and emoticon use to create an author model of a chatter [11]. This model was used to construct a Naïve Bayes classifier. Other such features may include parts of speech, dialog acts and other linguistic or paralinguistic artifacts found in a text.

Feature vectors are not always straightforward. In Forsyth's master's thesis, he created a dialog act tagger for chat. He used what may seem like a very obscure set of features, which included "the number of posts ago the poster last posted," and "the number of posts ago the poster made a spelling error." He then used these features to train a back-propagating neural network and a Naïve Bayes classifier [4].

Adams master's thesis demonstrates two commonly encountered chat NLP problems that he addressed in feature vector selection: inflated values of unimportant words due to their high frequency and discounted value of important words due to their infrequency [13]. His work involved thread disambiguation and dialog act detection as a way of complementing Forsyth's work in dialog act tagging and part of speech tagging.

In order to overcome these problems, he altered the feature vectors, which were composed of word frequencies, by using term frequency-inverse document frequency (tf-idf). tf-idf is a weighting scheme that discounts the frequency of a term according to the logarithm of the inverse of the proportion of documents that contain the word to the number of total documents.

Adams addresses another common chat NLP problem—the lack of sizable discourse for statistical significance. He employed *text augmentation*. Text augmentation strategies add or replace words that have some relation to the data in view, thereby enlarging the amount of data or reducing the variety of *types*. Adams used hypernym augmentation: the addition of or replacement of a particular word for a more semantically generic version [13]. This type of augmentation is a direct addition or replacement of words in the data. Another way to augment a text is to weight the feature vector differently according to certain criteria. He did so with nickname augmentation, which weighted each post more heavily if it came from the same author [13].

Similarly, Wang, attempted to improve upon Elsner et al.'s [18] thread disentanglement results, by using different "contexts" within the chat data as a means by which to augment a particular post's chat data [19]. Recognizing that short chat utterances may not produce statistically significant results, she probabilistically augmented each post based on references to other chat participants, all other posts by the author, and the time between the post and all other posts. She then used single-pass clustering to group each post into a particular thread, based on cosine similarities of the word distributions (which have now been adjusted base on augmentation). She reported better results than Elsner, but only marginally so [19].

E. TOPICALITY

The objective of our experiments is to demonstrate computationally viable ways of determining the topic of chat utterances. Topicality has many theoretical problems. There are many formal definitions for topic and divergent ideas concerning the scope and approach one should take in determining the topic of a document. We present Hjorland's perspectives on the issue of topicality and demonstrate how this idea may affect our results.

Hjorland writes in response to Bruza et al. concerning their approach to *aboutness.* Hjorland equivocates *aboutness* with subject, topicality and a number of other related ideas [20]. He contends that their approach was too narrow to properly describe the full range of ideas expressed by that term. Bruza et al. concluded that there is a "common-sense" approach to determining a documents subject, which is detectable by a rules-based framework [21]. Hjorland flatly denies this contention and deals with the idea in a comprehensive way. He builds his arguments within the context of how Information Science (IS) practitioners ought to construct IR systems, in light of ambiguities and the dynamic nature of topicality in its various social, epistemological, and structural settings. He provides several definitions of what a subject is and how it relates to other near complementary ideas such as field, discipline, theme, topic, domain, content, and relevance [20].

He states that the challenge of topicality, subjectivity, and aboutness is demonstrated by the wide variance in what people believe to be a document's subject. He provides additional evidence in that there are many different approaches to determining what the subject of a document is—none of which is sufficient on its own. He cites the work of Patrick Wilson who enumerates the following methods for determining a documents subject [20]:

> 1) […] identify the author's purpose in writing the document
>
> 2) […] weigh the relative dominance and subordination of different elements in the picture given by reading the document

> 3) […] group or count the documents’ use of concepts and references, and
>
> 4) […] invent a set of rules of selection for what are the “essential” elements (in contrast to the inessential) of the document in its entirety

Wilson concludes that the subject of a document is undeterminable. Hjorland disagrees and interprets Wilson’s methods in the following ways. The first method is addressed by hermeneutics, which studies the characteristics of the author to determine the meaning of the text [20]. Wilson’s second method is likened to efforts made by “modern psychological, cognitive, and user-oriented approaches” [20]. The third method analyzes documents according to statistical, bilbiometric and positivistic strategies. Hjorland, considers the fourth method to be related to “text linguistics and compositional methods” [20]. Hjorland summarizes this analysis by stating that the subject of a document is “related to theories of meaning, interpretation, and epistemology” [20].

Where Bruza et al. attempt to formalize aboutness within a framework, and then discuss methodologies for determining completeness, soundness and consistency of the system [21], Hjorland considers the role that aboutness plays in theories of IS and IR. In particular, Hjorland highlights IR systems biases toward a particular type of IR system based on their retrieval model. Therefore, any common-sense (rules-based) approaches to building these systems favor certain types of *aboutness* over and above other types [20]. The point of departure in his theory is that Bruza et al. assume that there exists universal structures that cause intersubjectivity[10] agreement, while he believes that discourse, and theoretical presumptions cause the agreement [20].

Countering Bruza et al., Hjorland defines subject in two ways. First, he defines it as “that ‘something’ that subject analysis and retrieval are supposed to identify” [20]. Second, he calls it, “the epistemological or informative potentials of

[10] Intersubjectivity in many NLP contexts means the understanding that is shared between people who are communicating. Hjorland simply means the subject that is common to a set of documents.

documents” [20]. These two definitions demonstrate that he believes there exists some knowledge that is true with respect to the people using the documents. From these definitions, he proceeds to discuss ideas similar to subject by way of distinctions. He distinguishes subject from field in that field is a social and cognitive concept. He describes topic as being related to field, but narrower in scope. The theme of a document is a description of the “broad pattern[s] of similarity” [20] in a document.

He performs the same type of analysis with domain, which he considers a cognitive and social construction. He defines information, which is also connected to subject, as a subjective concept that informs a person of “something” within a social context, requiring that different information systems treat documents differently according to the community of interest [20]. Finally, he analyzes the role of relevance in IR systems, arguing that a document may be relevant even if it is not of the desired subject. The affect skews IR system outputs, allowing reported incompetence and bias of annotators to negatively alter the results of such systems [20].

Defining and describing a topic is challenging and topics possess a subjective component as demonstrated by Bruza et al.’s work and Hjorland’s analysis. Whatever the underlying topic may be, it lays hidden, yet somehow shared. In this work, we construct models that do not consider cognitive, social, or psychological constructs explicitly. We define topic as the model that best fits what people consider topics. We consider only the surface aspects of language to be indicative of the latent topics. Any distinctions between topic, field, theme, or domain that may exist are ignored to the extent that the annotator ignored them. By taking this position, we acknowledge that the social context of annotators will affect our results. However, to some degree, by using probabilistic methods, we overcome the variance with large amounts of data. Essentially, we acknowledge Hjorland’s position, but leave such granular distinctions to the future work of others.

Unlike Bruza et al., who used a system of complex rules to develop a framework, we considered particular probabilistic models, which we train to develop robust topic models particular to chat. We discuss various probabilistic topic models that have been used to detect topics in chat, but focus on the model we use in our experiments—Latent Dirichlet Allocation (LDA).

F. TOPIC MODELS FOR TEXT

There are many ways to create topic models for textual documents. Many of these models leverage co-occurrence of terms in documents. Latent Semantic Analysis (LSA) (also Latent Semantic Indexing (LSI)), probabilistic Latent Semantic Analysis (pLSA) (also probabilistic Latent Semantic Indexing (pLSI)), and LDA all take advantage of the collocation of terms in documents. The differences in the models depend largely upon the probabilistic assumptions each model considers. LSA models each document as a linear projection of its term frequencies [22]. pLSA [23] and LDA [6] are probabilistic generative mixture models that consider each document a mixture of topics.

LSA and its IR complement LSI examine vector representations of documents (usually only term frequencies), which have a high dimension, and create a low dimension linear projection. Singular Value Decomposition (SVD) is a common method for creating such mappings. This model has improved IR tasks because documents that share co-occurring words should have a similar representation even though they may not share the particular word used to create the query [23]. Ostensibly, LSA is simultaneously performing a type of noise reduction [23]. Indeed, LSA has been helpful in detecting synonymy amongst words of the same topic and many of its applications have resulted in improved word processing [23].

pLSA and LDA are generative mixture models. A generative probability model assumes that outputs (documents, in this case) are produced according to a set of probabilistic rules. In these models, a topic is simply a distribution of words, where words with higher probability assumed to occur commonly together when the (unnamed) topic is being discussed. pLSA and LDA models have a more robust concept of a topic than LSA because they assume more than one topic generates each document. These models share two multinomial distributions: the document-topic distribution; and the topic-word distributions. Documents are assumed to accumulate words by randomly selecting them from topics. For each word position, a topic is first selected according to the document-topic distribution, and then a word is randomly selected from the corresponding topic-word distribution. These models use what is commonly referred to as the *bag-of-words* assumption, which disregards word sequence [6]. The interest of most literature on the subject is the relation of these models to the semantics of corpora and how to efficiently and effectively determine the models using statistical inference.

For pLSA, Hoffman used a modification of Expectation Maximization (EM) called tempered EM (TEM) in order to create estimates of the document-topic multinomial and the topic-word multinomial [23]. LDA takes a Bayesian approach, and places a Dirichlet prior on each topic. It places a conjugate prior over the multinomial document-topic distribution. This enables the use of estimation methods such as variational inference [6] and Gibbs sampling [24]. The pLSI model has parameters k multinomial distributions and the mixtures M. The multinomial distributions are the size of the vocabulary V. And there are k number of mixtures M. There are as many mixtures as there are documents, therefore the number of parameters grows linearly with the number of documents (size of the corpus). LDA, which treats "the topic mixture weights as a k-parameter hidden random variable" [6], has parameters that do not grow with the size of the corpus. Because of this, LDA does not have the same overfitting problems as pLSA [6].

G. LATENT DIRICHLET ALLOCATION (LDA)

We describe LDA in a more formal way because it is the topic model in view for the rest of this thesis. LDA has received much acclaim by the NLP community, and, as of this writing, represents the state-of-the-art for topic detection in textual documents. Adams used LDA in the context of chat for performing thread extraction [13]. We were unable to find experiments that use LDA as the central means of determining the topic of chat. LDA, in addition to the aforementioned advantages, provides the benefit of feature reduction for use in classification tasks [6].

We follow Griffiths and Steyvers' notation in our examination of LDA. As mentioned before LDA models documents as mixtures of topics, which are in turn distributions over words. For ease of explanation, we introduce notation to describe the ideas previously discussed:

z—a topic

w_i—the ith word token in a document

$P(z)$—the distribution of topics in a document

$P(w \mid z)$—the distribution over words given topic

T—number of topics

$P(z_i = j)$—probability the jth topic was sampled for the ith word token

$P(w_i \mid z = j)$—probability of word w_i under topic j

$\phi^{(j)} = P(w \mid z = j)$—multinomial distribution over words for topic j

$\theta^{(d)} = P(z)$—multinomial distribution over topics for document d

D—number of documents in the corpus

N_d—number of words in document d

$N = \sum N_d$—total number of word tokens

$\alpha = (\alpha_1 ..., \alpha_T)$—Dirichlet prior hyperparameter for θ

β—Dirichlet prior hyperparameter for ϕ

$p = (p_1 ..., p_T)$ —Dirichlet distribution over the topic multinomial

Given this notation the word distribution for a document is:

$$P(w_i) = \sum_{j=1}^{T} P(w_i \mid z_i = j) P(z_i = j)$$

Blei et al.'s innovation was to place a Dirichlet prior (α) on θ. Dirichlet distribution is a conjugate prior to the multinomial distribution. This simplifies the problem of statistical inference. $p = (p_1 ..., p_T)$ is the Dirichlet distribution for the topic multinomial and its probability density is:

$$Dir(\alpha_1, ..., \alpha_T) = \frac{\Gamma\left(\sum_j \alpha_j\right)}{\prod_j \Gamma\left(\alpha_j\right)} \prod_{j=1}^{T} p_j^{\alpha_j - 1}$$

where α is the set of all hyperparameters on θ—α can be individually interpreted as the expected count of a particular topic in a document before observing any of the words in the document. By changing these parameters, one may change the amount of smoothing amongst topics [24]. Increasing the α parameter increases the smoothing. This means that higher settings of α indicates the belief that the documents contain a greater mixture of topics rather than a smaller mixture containing fewer more concentrated topics [24].

Blei et al. introduced a similar strategy for smoothing the word-topic distribution by the placing a Dirichlet prior over the multinomial [6]. β like α is a hyper-parameter, but for ϕ multinomial distribution. β can be individually interpreted as the count of a particular word is sampled from a topic before any words have actually been observed [24]. This has the same smoothing affect that α has on the θ distribution. Hyperparameters α and β depend upon the particular corpus vocabulary and the number of topics selected for the model. Previous research conducted by Steyvers et al. shows that α values of $50/T$ and β values of 0.01 seem to work with various corpora [24]. In our experiments, we use these parameter settings as a proof of concept.

With the framework of LDA in place we now turn to the subject of constructing the model distributions ϕ and θ. Many scientists have used to Gibb sampling as a means of constructing LDA models of texts. It offers scientists the following advantages: easy of implementations, memory efficiency, and competitive speed and performance as compared to existing algorithms [25]. This thesis makes extensive use of the Mallet toolkit, which uses Gibbs sampling in constructing LDA models and inferring topic distributions of unseen texts.

H. GIBBS SAMPLING

Griffiths and Steyvers create estimates of ϕ and θ by evaluating the posterior distribution of words to topics, $P(z|w)$. $P(z|w)$ cannot be computed directly [25], so they estimate $P(z|w)$ using Gibbs sampling which is a Markov chain Monte Carlo (MCMC). MCMC is a method commonly employed in physics applications to sample large discrete probability distributions. Griffiths and Steyvers, describe it thusly [25]:

> In Markov chain Monte Carlo, a Markov chain is constructed to converge to the target distribution, and samples are then taken from that Markov chain. Each state of the chain is an assignment of values to the variables being sampled, in this case z, and transitions between states follow a simple rule. We use Gibbs sampling, known as the heat bath algorithm in statistical physics, where the next state is reached by sequentially sampling all variables from their distribution when conditioned on the current values of all other variables and the data.

The process of constructing the posterior of words to topics using Gibbs sampling can be seen as keeping track of two matrices. The first matrix is the counts of words per topic C^{WT} with dimensions $W \times T$, which contains the number of times a words is assigned to topics. The second matrix is the count of times a topics is assigned to a word in a documents: C^{DT}. Its dimensions are $D \times T$. The probability distribution below is used for assigning a word token to each topic, given the topic assignments of all other words after a sample has been executed [24]:

$$P(z_i = j \mid z_{-i}, w_i, d_i, \bullet) \propto \frac{C^{WT}_{w_i j} + \beta}{\sum_{w=1}^{W} C^{WT}_{wj} + W\beta} \frac{C^{DT}_{d_i j} + \alpha}{\sum_{t=1}^{T} C^{DT}_{d_i t} + T\alpha}$$

$z_i = j$ – is the topic assignment of token i to topic j

z_{-i} – is the topic assignments of all other word tokens

$\bullet$ – all known or observed information (e.g., other word and document indices w_{-i}, d_{-i} and hyperparameters α, and β.)

C^{DT}_{dj} – contains the number of times topic j is assigned to some word token in document d, not including the current instance i

Gibbs sampling would proceed as follows in order to construct the posterior [24]:

1. Randomly assign each token to a topic, which initializes C^{WT} and C^{DT}
2. For each word token

   2a. In C^{WT} and C^{DT} decrement the entries that correspond to the current topic assignment by one

   2b. Sample a new topic from the $P(z_i = j \mid z_{-i}, w_i, d_i, \bullet)$ and increment C^{WT} and C^{DT} with the new topic assignment by one

3. Step 2 continues without saving topic assignments until after a certain number of trips through the corpus called burn-in.
4. After burn-in, sampling continues and estimates of ϕ and θ (equations below) are collected periodically according to a predetermined interval of iterations, preventing correlations between samples—a process called thinning.
5. The process continues until a predetermined number of iterations execute at which point the process ends.

Determining appropriate values for burn-in, thinning, and the total iterations is the subject of open research. This thesis uses values from research concerning large corpora, but future work should give a better understanding of appropriate values for chat corpora. Gibbs sampling provides the means to construct an LDA model. Estimates of ϕ' and θ' of the word-topic distributions and topic-document distributions respectively, may be obtained by the following equations [24]:

$$\phi'^{(j)}_i = \frac{C^{WT}_{ij} + \beta}{\sum_{k=1}^{W} C^{WT}_{kj} + W\beta} \qquad \theta'^{(d)}_j = \frac{C^{DT}_{dj} + \alpha}{\sum_{k=1}^{T} C^{DT}_{dk} + T\alpha}$$

I. SAMPLED TOPIC DISTRIBUTIONS USING GIBBS SAMPLING

After, ϕ and θ were discovered using an LDA training set, we needed to determine the topic distribution of a new unseen document. Obtaining such a sampled distribution is a matter of performing the same procedure (Gibbs sampling), but holding the posterior distribution static and allowing the entry in the document topic counts for that document to be accumulated. Again, appropriate values of burn-in, thinning, and total iterations must be provided.

Given Gibb sampling as a method for constructing LDA models, we now discuss a method for determining the appropriate number of topics to create LDA models. Many methods for determining the number of topics in a given corpus have been proposed. Generally, these methods fall into two categories: objective and performance based. Performance based approaches evaluate the appropriateness of the number of topics selected by using the performance of a resultant classifier as a metric [24]. Conversely, Griffiths and Steyvers in *Finding Scientific Topics* propose an objective method for determining the appropriate number of topics by using the log-likelihood of the data used to build the model as an objective measure of the appropriateness of a particular number of topics [25]. We use the same method, but use the implementation provided with the Mallet toolkit. Griffiths and Steyvers, use the following equation to evaluate the likelihood of the data, $P(\mathbf{w}\,|\,\mathbf{z})$ [25]:

$$P(\mathbf{w}\,|\,\mathbf{z}) = \left(\frac{\Gamma(W\beta)}{\Gamma(\beta)^{W}}\right)^{T} \prod_{t=1}^{T} \frac{\prod_{w} n_j^{(w)} + \beta}{\Gamma(n_j^{(\cdot)} + W\beta)},$$

where $n_j^{(w)}$ is the number of times w has been assigned to topic j in the vector assignment $\mathbf{z}$, and Γ is the gamma function.

J. SUPPORT VECTOR MACHINES—SVM

After determining the appropriate number of topics for the chat corpus, we will use the LDA model and additional data to construct feature vectors, which represent the data. The feature vectors are the sampled topic probabilities that result from performing Gibbs sampling with the addition of an unseen document (described above). The resultant feature vectors are then used to construct a Support Vector Machine (SVM) classification system. After presenting test data, in the form of feature vectors, we use various metrics to evaluate the performance of the resultant classification systems. One of the advantages of using a feature vector created from an LDA model is that its dimensionality is the same as the number of topics chosen for the model.

An SVM is a maximum margin classifier that generates a separating hyperplane between sets of data distinguished by their class [26]. Classes of data are always separable, given an appropriate nonlinear mapping of the data to some higher dimensional space [27]. Figure 1 provides an example of a set of data that is linearly separable and demonstrates that SVMs create a maximum margin between the two sets.

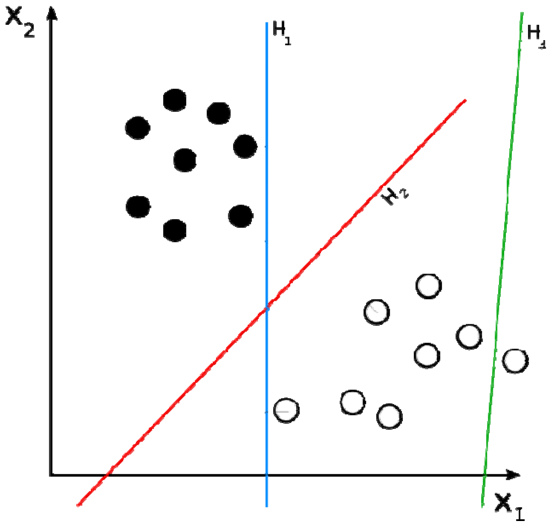


Figure 1. Trivial example of linearly separable data. From [28]

The linear SVM will seek the maximum margin between the two classes; it will find the hyperplane that creates the greatest distance between the closest data points of the opposing classes. These points define the support vectors. Figure 2 shows a hypothetical data set and its accompanying hyperplane, and support vectors.

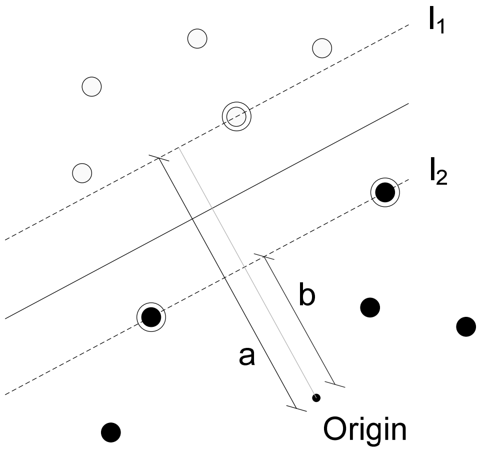


Figure 2. Linear Separating Hyperplanes. From [12]. The support vectors are circled. The maximum margin is the distance between the two dashed lines (l_1 and l_2). The hyperplane is the solid line between the two dashed lines.

We use the nomenclature used in *A Short SVM (Support Vector Machine) Tutorial*, by J. P. Lewis, in the following discussion of how to obtain the support vectors for a given data set. We consider a situation where there exists only to classes of data. Lewis uses the following set of mathematical relationships [26]:

$\vec{w}^T\vec{x}+b=0$—the general equation for the hyperplane

b —a constant

$\vec{x}$ —a data point using vector notation

$\vec{w}$ —a vector of weights

$\vec{w}^T\vec{x}_j+b<0$—the equality statement for data of class one

$\vec{w}^T\vec{x}_k+b>0$—the equality statement for data of class two

$y_i \in \{-1,1\}$—set of labels for the classes

$y_i(\vec{w}^T \vec{x}_i + b) > 0$—holds for all points

It can be demonstrated that the hyperplane is offset from the origin along $\vec{w}$ by $\frac{b}{\|\vec{w}\|}$. Additionally, $\vec{w}$ and b are scalable without altering the hyperplane. In order to prevent such scaling Lewis adds the following constraint [26]:

$y_i(\vec{w}^T \vec{x}_i + b) \geq 1 \ \ \forall i$

Next, Lewis describes how to obtain "an expression for the distance between the hyperplane and the closest points" [26]. In Figure 2, this is the distance between the l_1 and l_2, which pass through the support vectors. Lewis refers to them as supporting hyperplanes. Their formula's are $y_j(\vec{w}^T \vec{x}_j + b) = 1$ for the positive support vector(s) and $y_k(\vec{w}^T \vec{x}_k + b) = 1$ for the negative support vector(s). Lewis demonstrated how KKT (Karush Kuhn Tucker) [26] methods are used to determine $\vec{w}$ and b to maximize the distance.

It is advantageous to allow some data points to lie on the opposite class side of the separating hyperplane or between the margin and the separating hyperplane in order to prevent overfitting in classification [26]. Using non-linear mappings into higher dimensional spaces will separate the data, but these mappings may make a poor classifier for unseen examples (i.e. test data). This problem is overcome by providing a mechanism for allowing deviation from the margin—"slack variables" (s_k) [26]. The slack variable is introduced by modifying the margin constraint:

$y_k(\vec{w}^T \vec{x}_k + b) \geq 1$ (original) $\quad$ $y_k(\vec{w}^T \vec{x}_k + b) \geq 1 - s_k$ (modified)

This allows the creation of a better hyperplane by loosening the margin constraint Figure 3 demonstrates slack variables in use.

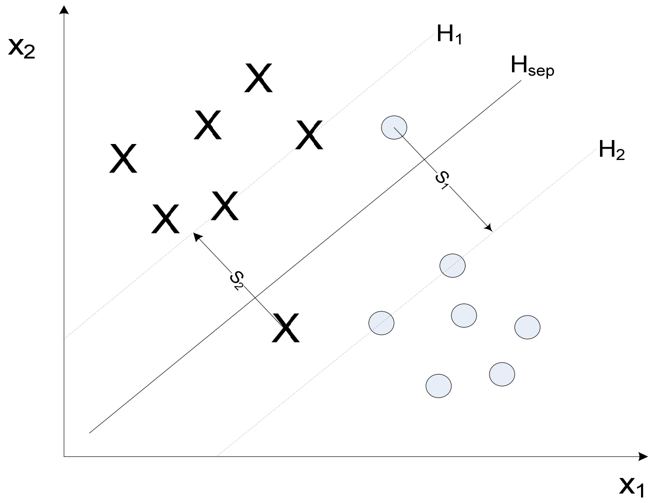


Figure 3. This figure shows the relationship of slack variables to the margin and data points that are found on the opposite class side of the separating hyperplane. s_1 and s_2 are distances from the margin to the point in question. After [37]. H_{sep} is the separating hyperplane and H_1 and H_2 are the margins.

In order to prevent simply selecting a large slack variable that allows any hyperplane to separate the data, Lewis shows how a penalizing term is added to the KKT set used to determine the maximal margin [26]. The slack variable and corresponding penalization help to create SVMs that are more general. A more robust treatment may be found in the reference material ([26]), for the interested reader. We now turn to the matter of how to map data into higher dimensional space using the various kernel functions.

The kernel function is a function that maps the original feature space into a new feature space [29]. Generally, a researcher should use a kernel that has some meaningful relation to the data domain in view, or convert the data to a domain where other kernels have some demonstrated relevance [29]. However, in our study we thought it wise to begin with well-know and simple kernels, as SVM applications in the chat domain are relatively unexplored. Borrowing Hsu et al.'s notation from *A Practical Guide to Support Vector Classification* [30], we use the following kernels as the basis for the SVMs in this research: linear, polynomial, radial basis function, and sigmoid. Their respective functions are as follows [30]:

Linear: $K(x_i, x_j) = x_i^T x_j$

Polynomial: $K(x_i, x_j) = (\gamma x_i^T x_j + r)^d, \gamma > 0$

Radial Basis Function (RBF): $K(x_i, x_j) = \exp(-\gamma \| x_i - x_j \|^2), \gamma > 0$

Sigmoid: $K(x_i, x_j) = \tanh(\gamma x_i^T x_j + r)$

Here, γ, r, and d are kernel parameters. These kernels are a reasonable first step for SVM users because they are simple and are often used to build other more complex kernels [30].

You, Lee and Li proposed a kernel, which may hold more relevance for features taken from a probability distribution [31]. Past research using a Kullback-Leibler (KL) kernels have been applied classification systems for evaluating voice recognition using Gaussian Mixture Model-supervectors[11] (GMM-supervectors) SVM classifiers [31]. They propose a kernel that uses the Bhattacharyya distance measure instead of a KL kernel. Both Bhattacharyya and KL distances are means by which to evaluate the relation of one probability distribution to another [31]. Either a KL or Bhattacharyya may produce better results than the four kernels used in this research; since using SVM's for LDA model chat data is relatively new territory, we will leave evaluation of other kernels to future research.

An SVM classifier takes one set of data to build its model. This set is called the training set. It is composed of feature vectors from the positive and negative classes, which are labeled as such. In our case, the positive label is 1 and the negative label is -1. Then a separate set (the test set) of feature vectors from both classes are placed in the newly constructed feature space. If a

[11] GMM-supervectors are vectors that include GMM parameters such as mean vectors, covariance matrices, and mixture weights.

positive test feature vector falls on the positive class's side of the hyperplane it is counted as a "true positive." If it falls on the negative class's side of the hyperplane it is counted as a "false negative." If a negative feature vector falls on the positive class's side of the hyperplane it is counted as a "false positive." The counts are used to evaluate the performance of the classifier.

K. EVALUATION METHODS

We use precision, recall, and F-score to measure the performance of the various classifiers discovered in the research. These are common, easy to implement and easily interpreted measures of performance. Each measure relies on the three following values:

TP—true positives

TN—true negatives

FP—false positives

FN—false negatives

TP is the count of all the test examples that were from the positive class, and were classified as such by the classifier. TN is the count of all the test examples that were from the negative class and were classified as such. FP is the count of all the test examples that were from the negative class but were classified as being a part of the positive class. FN is the count of all the test example that were from the positive class but were classified as being a part of the negative class. We provide the equation and short description of each measure below.

1. Precision

Precision is the proportion of correctly classified positive examples to the total of those that were classified as positive examples. Colloquially, it might be stated thusly, "Of all the ones I called positive, what percent actually were positive." It can be interpreted as the classifier's ability to identify the positive class while at the same time not calling negative examples positive, for as FP decreases, precision increases. Its equation is a follows:

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

2. Recall

Recall is the proportion of correctly classified positive examples to the total of those that were actually positive examples. Colloquially, it might be stated thusly, “Of all the actual positive examples, what percent actually were classified as positive.” It can be interpreted as the classifier’s ability to identify the positive class while at the same time not missing positive examples, for as FN decreases, recall increases. Its equation is:

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

3. F-score

F-score is the harmonic mean of precision and recall. It can be interpreted as the balance between precision and recall. Its equation is:

$$F - score = \frac{2}{\frac{1}{P} + \frac{1}{R}}$$

Generally, the F-score functions as the means to ensure the integrity of a classifier. Those wishing to obtain high levels of recall need only classify all examples as positive. In this case, no examples would be labeled as negative, causing FN to equal 0, which in turn causes recall to equal 1—its highest possible value. F-score prevents such a scheme from going unnoticed by favoring the lower of the two. Were the case in point to actually occur, FP would equal the number of negative examples in the test set, because they were all classified as positive. This would in turn drive precision to its lowest possible value. The lower precision value would increase the overall size of the denominator in the F-score equation driving the F-score value to its lowest possible value. For our purposes, high values of F-score are considered more favorable, followed closely by high values of recall.

THIS PAGE INTENTIONALLY LEFT BLANK

III. EXPERIMENTAL DESIGN AND METHODOLOGY

A. SOURCE OF DATA

We explored four different types of classification tasks for our research. Each of the experiment sets included the construction of SVM classifiers. We construct LDA models based on different document notions from which we derived feature vectors for SVM classifiers. The first set of experiments considered the collective posts of an author as a document. The second set considered an individual post as a document. The third and fourth considered a textbook paragraph as a document.

1. NPS Chat Corpus

We use two distinct sections of the NPS Chat Corpus. First, age-oriented chat posts from twenty-year-olds collected by Lin in 2006 [13]. Second, Freenode IRC posts for physics collected by Adams [13]. Lin collected chat posts from five different age-segregated socially oriented rooms. There are more than 475,000 posts created by over 3,200 users. Lin's collection was from a non-IRC chat site—Talkcity. The chat rooms were not oriented by topic, but were socially oriented [13]. We will focus on a small segment of the corpus; namely, those posts created by self-reported age group users. The age groups collected were teens, twenties, thirties, forties, and fifties. We ignore their age in order to focus on methods for topic detection. We refer to this set of chat posts hereto forward as *all-ages-chat*.

Adams collected the second corpus, also a part of the NPS Chat Corpus, from the Freenode IRC server during July 2008. Using the open source pidgin [32] client, the data was collected over twelve days and is composed of 7,803 posts [13]. Over 280 authors contributed to the physics chat. The conversations involved people seeking help with physics related problems, seeking out physics textbook recommendations, and reflecting on physics in general. Conversations about religion and confrontations often occurred, despite the general focus on physics. Additionally, the system generates posts created as a direct result of user interaction. For instance, the system performs a Wikipedia lookup if a user posts a message with certain syntax. Physics chat, compared to all-ages-chat, has more focused conversations. Adams attributes this characteristic to three factors: 1) the chat room's topic assignment, 2) usage rules, and 3) enforcement of the rules [13]. The usage rules were posted in the room automatically and periodically. In two instances, users were ejected from the chat room because their conversations were off-topic and confrontational. We use this physics chat, because physics has a known and particular theme that is readily identifiable by people and distinct from other types of conversation.

2. Newtonian Physics Textbook

We use a Newtonian Physics Textbook [33] made available under the Creative Commons License as a basis for LDA models in two set of experiments. The physics text is a textbook dedicated to the small segment of physics know as "Newtonian" physics. This book covers the following subjects as stated in its table of contents: Velocity and Relative Motion, Acceleration and Free Fall, Force and Motion, Analysis of Forces, Motion in Three Dimensions, Vectors, Circular Motion, and Gravity. We decompose the textbook into paragraphs for use in the LDA model as documents. There are 488 paragraphs in the book. Paragraphs are reasonable size compared to the documents they will be use to predict, namely short chat posts and the collective posts of individual authors.

3. Use and Annotation of the Corpora

We use the corpora in various ways throughout our experiments. Generally, the physics chat forms the core of the positive class in all of our classification experiments. All-ages-chat forms the core of our negative class in author experiments, because we assume that all the authors contributed to at least one physics conversation. In two sets of experiments, we use non-physics chat posts within the physics chat corpus as the negative class. In this case, we hand annotated all of the posts from the physics chat as either "physics" or "non-physics." We only use the physics textbook to construct LDA models with which to build feature vectors for training and testing the SVM classifier. We never use the textbook as a source of training and or testing data for the SVM.

B. GENERAL APPROACH

1. Four Types of Experiment Sets

We perform four different types of experiments. In each experiment set, we first create an LDA model from a source of topic information then generate SVM testing and training sets based on data from a positive and negative class of the class data. In the experiments, we exclude the negative class from the LDA model in one subset and then add the negative class for the second subset. The SVM classifier requires testing and training data in the form of feature vectors. In our case, the feature vectors are the sampled topic distributions created by Gibbs sampling as described in Chapter II, Section J of this thesis. After generating the feature vectors using the LDA model, we train and test the SVM classifier. Each set of feature vectors is classified using a combination of the four previously mentioned kernels, and a range of slack variables. Figure 4 shows a diagram of the mechanics of the experiments.

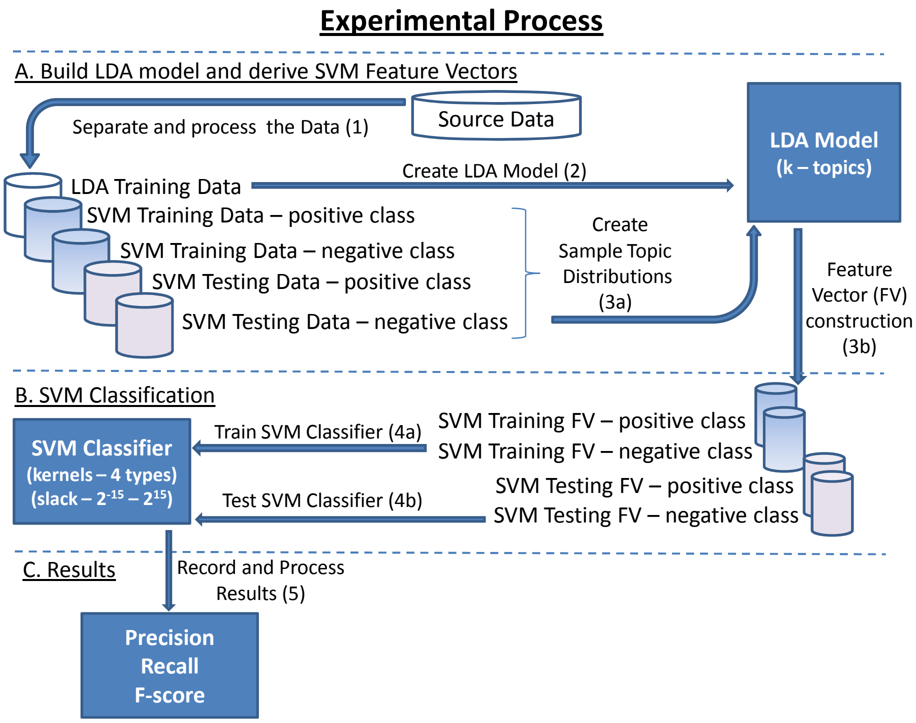
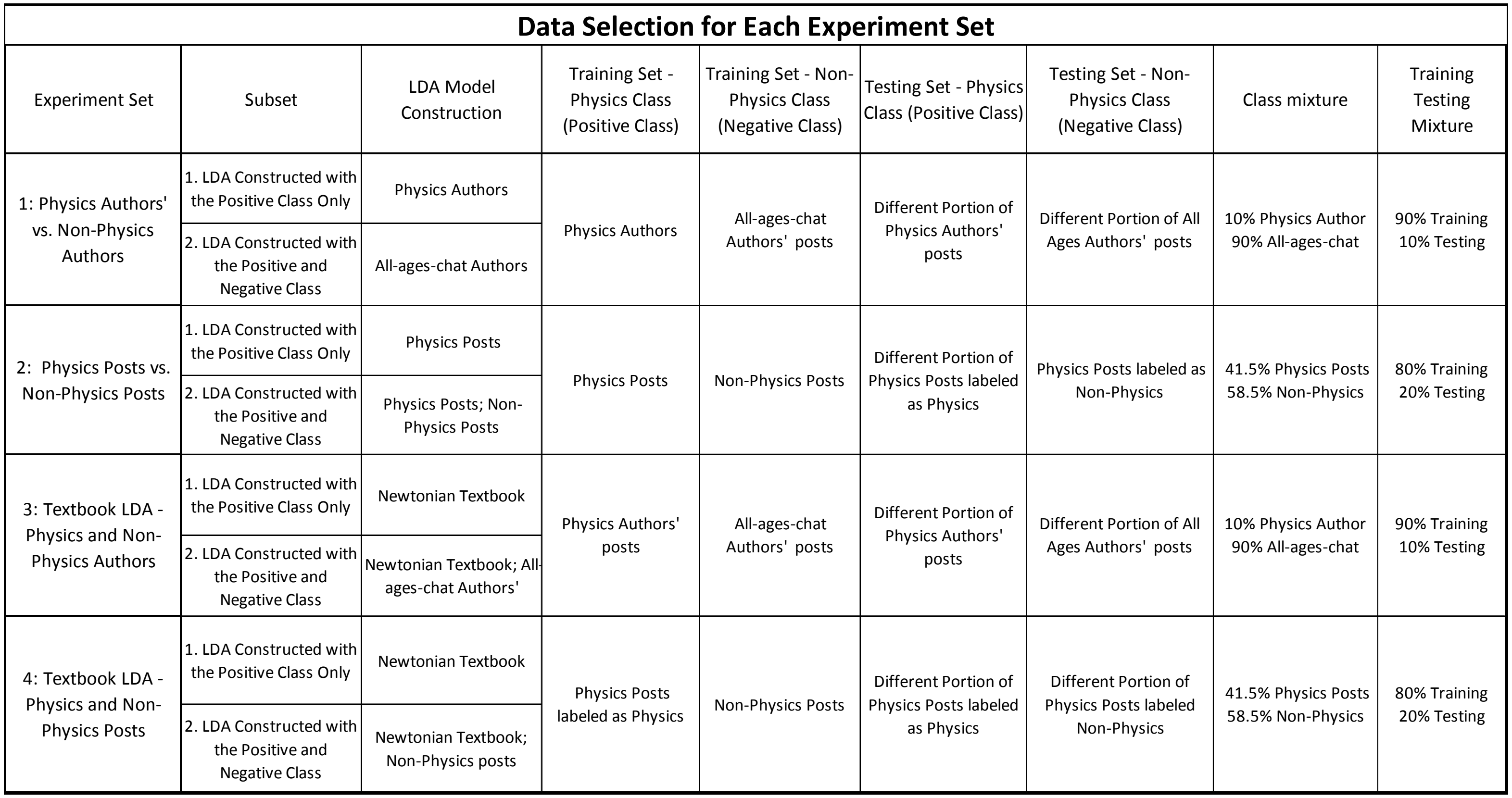

Experimental Process
A. Build LDA model and derive SVM Feature Vectors
Source Data
Separate and process the Data (1)
LDA Training Data
Create LDA Model (2)
LDA Model
(k – topics)
SVM Training Data – positive class
SVM Training Data – negative class
SVM Testing Data – positive class
SVM Testing Data – negative class
Create
Sample Topic
Distributions
(3a)
Feature
Vector (FV)
construction
(3b)
B. SVM Classification
SVM Classifier
(kernels – 4 types)
(slack – 2^{-15} – 2^{15})
Train SVM Classifier (4a)
SVM Training FV – positive class
SVM Training FV – negative class
SVM Testing FV – positive class
SVM Testing FV – negative class
Test SVM Classifier (4b)
C. Results
Record and Process
Results (5)
Precision
Recall
F-score

Figure 4. This figure outlines the general experimental process. Note, in the LDA Model, "k-topics" refers to a different setting for the number of topics calculated per data set (Discussed in Chapter III, Section C). Note also, in the SVM classifier, "kernels" refers to the four different kernel types (linear, radial basis function, polynomial and sigmoid) that will be used in separate SVM models of the same training and testing set. Note also, the range of slack variables to be investigated for each kernel type.

Each experiment hypothesizes that topic distributions of LDA documents in some way map to what humans consider topics. First, we consider the author's posts as documents, deriving all the testing and training sets from the collective posts by an author from the physics chat (positive class) and all-ages-chat (negative class). In the second set of experiments, we consider only the posts within the physics chat, using those items labeled physics as our positive class and non-physics as our negative class. Third, we consider the posts collected by author as they relate to an LDA model constructed by the physics textbook. As in the first experiment set, we use the physics authors as the positive class and all ages of authors as the negative class. In the fourth set of experiments, we again use the physics text as the basis for constructing the LDA model and then classify physics posts versus non-physics posts. In all experiment sets, we construct LDA models based on the positive class alone and then construct the LDA models using both the positive and negative classes. Table 1, outlines how the different data are used in each experiment set.

For experiments that compared authors (1 and 3), our training to testing ratio was 90%-10%. The class split was 10% physics authors to 90% all-ages-chat authors to simulate rarity of the positive class. For experiments that compared individual posts (experiments 2 and 4), our training to testing ratio was 80%-20%; the class split was 41.5% physics and 58.5% non-physics as this is the naturally occurring split. In all experiments, we performed random sampling from each class 10 times and evaluate the average, maximum and minimum of the results. When splitting the classes, it was necessary to leave some data out in order to precisely obtain the correct training and testing ratios. We chose to leave out the smallest author and post documents in the corpus. Table 1 shows the data settings for each experiment subset.



Data Selection for Each Experiment Set

Experiment Set	Subset	LDA Model Construction	Training Set - Physics Class (Positive Class)	Training Set - Non-Physics Class (Negative Class)	Testing Set - Physics Class (Positive Class)	Testing Set - Non-Physics Class (Negative Class)	Class mixture	Training Testing Mixture
1: Physics Authors' vs. Non-Physics Authors	1. LDA Constructed with the Positive Class Only	Physics Authors	Physics Authors	All-ages-chat Authors' posts	Different Portion of Physics Authors' posts	Different Portion of All Ages Authors' posts	10% Physics Author 90% All-ages-chat	90% Training 10% Testing
	2. LDA Constructed with the Positive and Negative Class	All-ages-chat Authors						
2: Physics Posts vs. Non-Physics Posts	1. LDA Constructed with the Positive Class Only	Physics Posts	Physics Posts	Non-Physics Posts	Different Portion of Physics Posts labeled as Physics	Physics Posts labeled as Non-Physics	41.5% Physics Posts 58.5% Non-Physics	80% Training 20% Testing
	2. LDA Constructed with the Positive and Negative Class	Physics Posts; Non-Physics Posts						
3: Textbook LDA - Physics and Non-Physics Authors	1. LDA Constructed with the Positive Class Only	Newtonian Textbook	Physics Authors' posts	All-ages-chat Authors' posts	Different Portion of Physics Authors' posts	Different Portion of All Ages Authors' posts	10% Physics Author 90% All-ages-chat	90% Training 10% Testing
	2. LDA Constructed with the Positive and Negative Class	Newtonian Textbook; All-ages-chat Authors'						
4: Textbook LDA - Physics and Non-Physics Posts	1. LDA Constructed with the Positive Class Only	Newtonian Textbook	Physics Posts labeled as Physics	Non-Physics Posts	Different Portion of Physics Posts labeled as Physics	Different Portion of Physics Posts labeled Non-Physics	41.5% Physics Posts 58.5% Non-Physics	80% Training 20% Testing
	2. LDA Constructed with the Positive and Negative Class	Newtonian Textbook; Non-Physics posts						

Table 1. Data Selection for each Experiment Set

For all the experiments, we divided the data into 5 different sets then preprocess them. The data sets are: 1) the LDA training data 2) the SVM training data—positive class 3) the SVM training data—negative class, 4) the SVM testing data—positive class, and 5) the SVM testing data—negative class. We perform simplistic stop word selection in all of the experiments, but leave the exhaustive search of that space to future work. We used the stop word list provided by the classification toolkit: Mallet [34].

C. LDA MODEL SELECTION USING MALLET

Mallet (MAchine Learning for LanguagE Toolkit) performs all of the functions we required for creating LDA models. It allows the user to adjust all the LDA hyperparameters α , β , and k (discussed in Chapter II, Section G), and the Gibbs Sampling hyperparameters: thinning, burn-in and total iterations (discussed in Chapter II, Section H). Mallet offers more than one LDA implementation. We use the ParallelTopicModel class of Release Candidate 4 [34], in our experiments because it allows the user the ability to obtain sampled distributions of unseen documents.

1. LDA Hyperparameter Selection

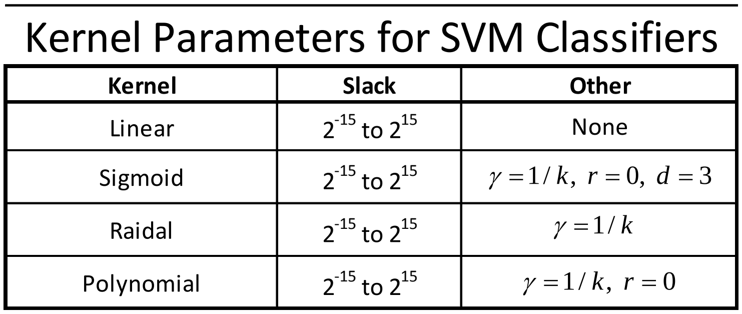
Hyperparameter optimization with regard to LDA and Gibbs Sampling is the subject of current research. We use hyperparameters used in previous research, except in the case for k. α and β were obtained from Griffiths and Steyvers in [25]. As stated in Chapter II, Section G we use values of 50/k for α and 0.01 for β in all experiments.

2. Empirically Derived Topic Number Selection

Like Griffiths and Steyvers we empirically derive the number of topics for each Experiment Set. Using Griffiths and Steyvers' method, we use the objective measure of log-likelihood of $P(\mathbf{w} | \mathbf{z})$ as described in Chapter II, Section K to determine the number of topics each data set requires. Since Gibbs Sampling is a stochastic process, there exists some variation between the results of running the Gibbs Sampler over the same data, using the same settings. Mallet uses the following Gibbs Sampling hyperparameters for finding the log-likelihood of a data set and accompanying settings: 1000 for total iterations, 200 for burn-in, and 50 for thinning. In order to determine optimal value of k, we execute the process of constructing the LDA model and then measure the log-likelihood of $P(\mathbf{w} | \mathbf{z})$ ten times for each setting of k topic number. This is performed for each data set. We chose the k that generated the highest average log-likelihood for the models used in classifying the data set under investigation. The results of these experiments are provided in this chapter, Section E.

3. SVM Parameter Selection

There are several settings associated with SVM classifiers. The parameter explored most in this work is the "slack" or cost parameter. We use the LIBSVM implementation for SVM and all experiments we use the default values native to the implementation [35]. However, we range over all values of 2^{-15} to 2^{15} increasing by powers of two for the slack parameter. We use four different kernels: linear, radial, sigmoid, and polynomial. The linear kernel ($K(x_i, x_j) = x_i^T x_j$) has no hyperparameters to adjust. For the polynomial kernel ($K(x_i, x_j) = (\gamma x_i^T x_j + r)^d, \gamma > 0$), we use the default setting of the γ parameter of $\frac{1}{\text{Feature Vector Size}} = \frac{1}{\text{Number of Topics}}$, 0 for the r parameter, and 3 for the d parameter. The RBF kernel ($K(x_i, x_j) = \exp(-\gamma \| x_i - x_j \|^2)$) requires one parameter γ, which we set to $\frac{1}{\text{Feature Vector Size}} = \frac{1}{\text{Number of Topics}}$ (the default). For the sigmoid kernel ($K(x_i, x_j) = \tanh(\gamma x_i^T x_j + r)$) we use the default values for γ and r. Each kernel and parameter settings are applied to each experiment set. Table 2 shows the setting of the kernel and their parameters that we use for each experiment set.



Kernel Parameters for SVM Classifiers

Kernel	Slack	Other
Linear	2^{-15} to 2^{15}	None
Sigmoid	2^{-15} to 2^{15}	$\gamma = 1/k,\ r = 0,\ d = 3$
Raidal	2^{-15} to 2^{15}	$\gamma = 1/k$
Polynomial	2^{-15} to 2^{15}	$\gamma = 1/k,\ r = 0$

Table 2. This table summarizes the kernel parameters and the slack variables explored in our experiments.

The combination of LDA model parameter choices and SVM classifier parameter choices creates a very large set of experiments to explore. Instead of focusing on any one type, we explore the broad number of possibilities as a way of demonstrating the validity of these types of experiments. This provides a solid “first-step” into the this type of classification system for chat.

D. RESULTS OF TOPIC NUMBER DETERMINATION

As stated afore, the number of topics determined in this set of experiments, also determines the k number of topics to be used in the subsequent experiments. The three different divisions of the data sets demand an empirical estimate of the optimal values of k. For our use, the optimal k is the number of topics that contain the largest average log-likelihood of each model. This k provides the basis for the constructing subsequent LDA models. In each of the experiments, we used different document types for the LDA model. We chose three different document types, which form the basis for the four different experiment sets. We chose the documents to be 1) all of an individual author’s post, 2) an individual post, and 3) a paragraph from the physics textbook.

1. Number of Topics for Author Level Documents

The author-divided chat data is the collection of the physics chat posts that were created by a single author. One may think of this data set as the composition of each author's half of the conversation. More precisely, it is the composition of each of the author's contribution to multiple conversations. This data set, model, and accompanying classifier, should be able to identify whether or not an author had discussed a particular topic.

For this division of the data, we removed all the authors that had zero posts. We also removed all the authors that had only system level posts such as entering and exiting declarations. We did not remove punctuation, emoticons, but removed stop words provided in the Mallet toolkit. Each LDA model was constructed ten times with the same settings, and the log-likelihood was determined. We conducted ten iterations of each k ranging from 4 to 200, 300, 400, 500, 600, 700, 800, 900, and 1000 on the entire data set. There were 283 author documents extracted from the physics chat data.

Figure 5 shows the average log-likelihood of each k topics explored. For these experiments $\alpha = 50/k$ and $\beta = 0.01$. The maximum average log-likelihood was -171078, for the k value 13.

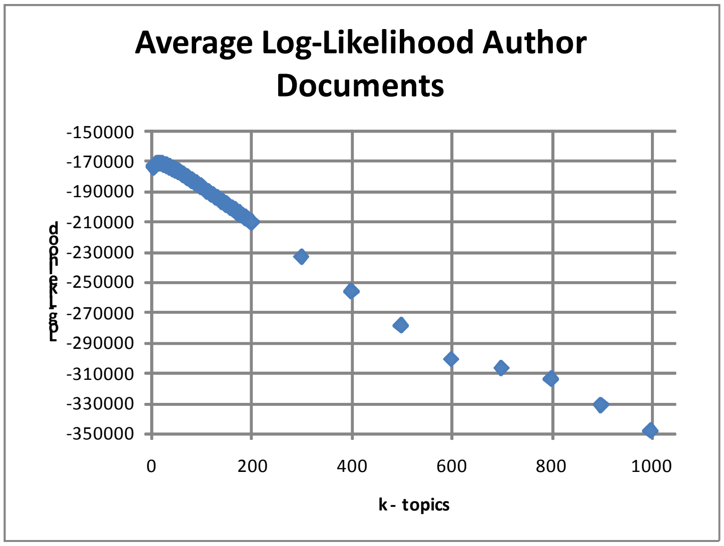


Figure 5. This figure shows the average of 10 samples of each value of k. The LDA settings were: $k \in \{4,...,200,300,400,500,600,700,800,900,1000\}$, $\beta = 0.01$, and $\alpha = 50/k$. The document selections were the collection of individual author posts.

Figure 6 displays the all log-likelihood results for k values 4 to 60. It demonstrates the difficulty of concluding an optimal value of k. In each of the topic selection experiments, the variation between log-likelihood samples for each value of k, was significant. So much so, that concluding which k values would yield the most probable model based on log-likelihoods alone made little sense.

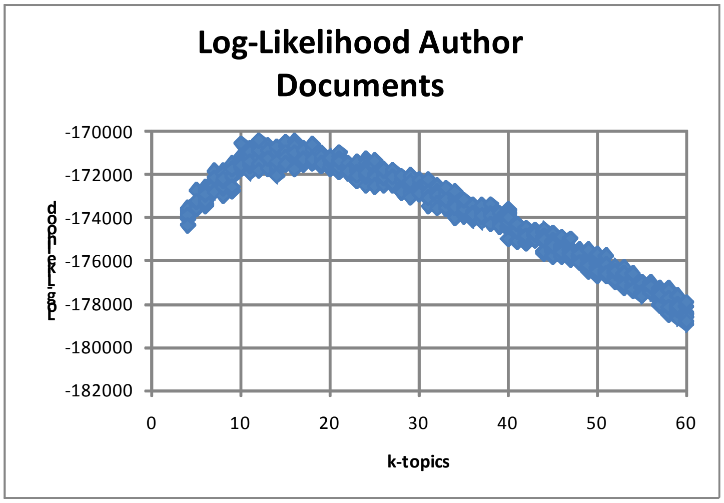


Figure 6. This figure shows all 10 samples of a subset of k values for Author Level Documents. The LDA settings were: $k \in \{4,...,60\}$, $\beta = 0.01$, and $\alpha = 50/k$. The document selections were the collection of individual author posts.

2. Number of Topics for Post Level Documents

The post-divided chat data considers each document of the LDA model to be a post. This type of design decision supposes that each post can be identified with a topic. We removed all the posts that had only system created messages such as entering and exiting declarations, except Wikipedia lookups. We did not remove punctuation, emoticons, or stop words. We conducted ten iterations for each topic number ranging from 4 to 200, 300, 400, 500, 600, 700, 800, 900, and 1000 on the entire data set. There were 5037 post documents extracted from the 7803 post corpus.

Figure 7 shows the average log-likelihood of each k topics explored. For these experiments $\alpha = 50/k$ and $\beta = 0.01$. The maximum average log-likelihood was -171129, for the k value 16. Figure 8 displays the all log-likelihood results for k values 4 to 60.

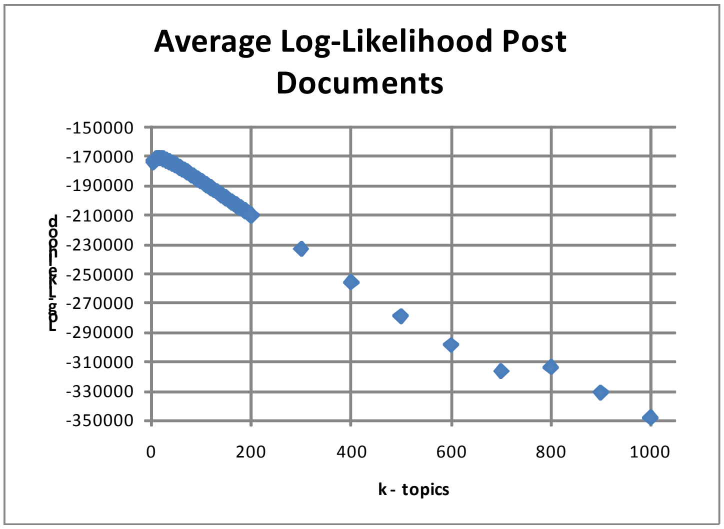


Figure 7. This figure shows the average of 10 samples of each value of k for Post Level Doucuments. The LDA settings were: $k \in \{4,...,200,300,400,500,600,700,800,900,1000\}$, $\alpha = 50/k$, and $\beta = 0.01$. The document selections were the collection of individual posts.

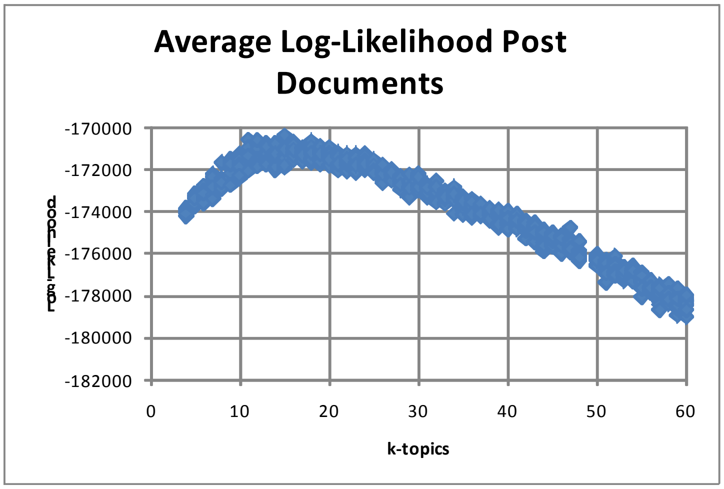


Figure 8. This figure shows all 10 samples of a subset of k values for Post Level Documents. The LDA settings were: $k \in \{4,...,60\}$, $\beta = 0.01$, and $\alpha = 50/k$. The document selections were the collection of individual author posts.

3. Number of Topics for Textbook Paragraph Level Documents

The Newtonian Physics textbook data document division considers each paragraph of the textbook an LDA document. In all the other models, under consideration, the model of the physics topic is constructed based on chat data. Using textbook data, the model is constructed based on a traditional written language example. In the experiment sets involving this model, we test whether LDA is suitable model for cross-domain topic modeling. This data set, model, and accompanying classifiers should be able to identify whether or not an author or a post relates to the physics text.

We removed no punctuation, but removed the stop words provided by the Mallet Toolkit. We conducted 10 iterations of each topic number ranging from 4 to 200, 300, 400, 500, 600, 700, 800, 900, and 1000, holding out no data. There were 488 paragraphs extracted from the Newtonian Physics textbook.

Figure 9 shows the average log-likelihood of each k topics explored. For these experiments $\alpha = 50/k$ and $\beta = 0.01$. The maximum average log-likelihood was -171043, for the k value 15. Figure 10 displays the all log-likelihood results for k values 4 to 60.

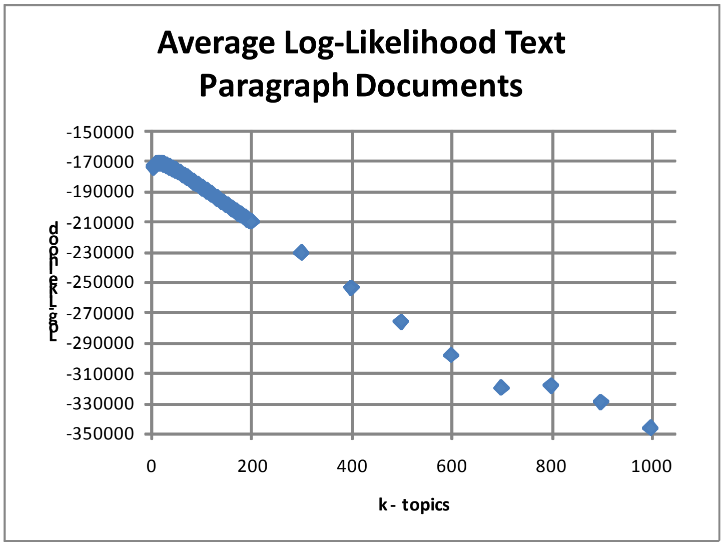


Figure 9. This figure shows the average of 10 samples of each value of k for Paragraph Level Documents. The LDA settings were: $k \in \{4,...,200,300,400,500,600,700,800,900,1000\}$, $\alpha = 50/k$, and $\beta = 0.01$. The document selections were the collection of paragraphs from the Newtonian Physics textbook.

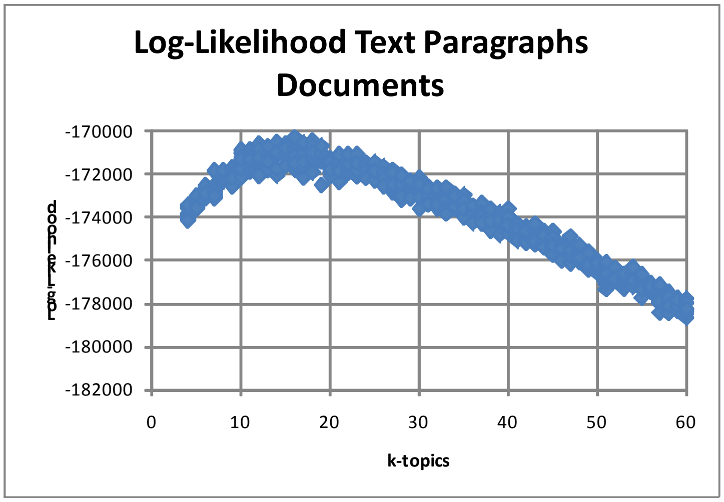


Figure 10. This figure shows all 10 samples of a subset of k values for Paragraph Level Documents. The LDA settings were: $k \in \{4,...,60\}$, $\beta = 0.01$, and $\alpha = 50/k$. The document selections were the collection of paragraphs from the Newtonian Physics textbook.

4. Topic Number Selection Analysis

In each of the topic-number experiments, the graphs reach a maximum value around the 13 to 16, k topic mark. The curves demonstrate remarkable consistency in shape. This type of result was expected for the author documents and the post documents, as they were exactly the same data. However, the Newtonian Physics textbook data exhibited similar shape and maximum values. These phenomena may have resulted from the limited scope of α and β values examined. Since we did not vary these parameters, it is difficult to determine whether they would have any effect. Were we to vary these parameters it may demonstrate that the resultant k topics are not a good measure of the topic models for the corpora. However, the consistent shape and topic number similarities between data sets may indicate that humans discuss matters of physics in very similar ways; regardless of the mode of language employed.

The range of log-likelihood for each ten iterations of the k topics explored is also troubling. Each k values' log-likelihood varies so greatly that it is difficult to distinguish the quality of its model from those surrounding it. That is to say, any number of the surrounding models may be as good or better than the one with the highest log-likelihood. Ten samples may not be sufficient to determine which k topic is best. Regardless, the maximum average meets the immediate need of providing our four experiment sets with an initial value for k that holds some meaningful relationship to the corpora.

E. CROSS VALIDATION

Cross validation is methodology used to prevent anomalous statistical results from driving scientific conclusions in small data sets. Two major factors drive this motivation. First, the small data sets may contain divisions of the data, which create classification examples that are not representative of the whole: they are exceptions [36]. Secondly, the results may be based on random factors that affect classifier generalization [37]. In our case, Gibbs sampling contains such a stochastic process.

We did not use 10-fold cross validation in our experiments. Consider a popular method for performing 10-fold cross validation. The data set is divided into ten smaller data sets, whose elements are chosen randomly. Then, nine divisions are used to train the model or classifier and the remaining division is used for testing. This is done 10 times each time, exchanging one of the training divisions with the testing division. Then, the classification results are analyzed. Often, they are averaged.

A simpler version of this process randomly samples 10% of the data ten times. In this case, quality of generalization will depend on the amount of overlap in the training set. We instead use random sampling, where we randomly select 90% (or 80%) of the data for training, and use the remaining 10% (or 20%) for testing. We perform this procedure ten times and then analyze the results.

IV. RESULTS AND ANALYSIS

In this chapter, we present the results of the experiments described in Chapter III. We provide the results of the four different LDA leveraged SVM experiments. We explored four different types of classification tasks. We construct LDA models based on different document notions from which we derived feature vectors for SVM classification classifiers. The first set of experiments considered the collective posts of an author as a document. The second set considered an individual posts as a document. The third and fourth considered a textbook paragraph as a document.

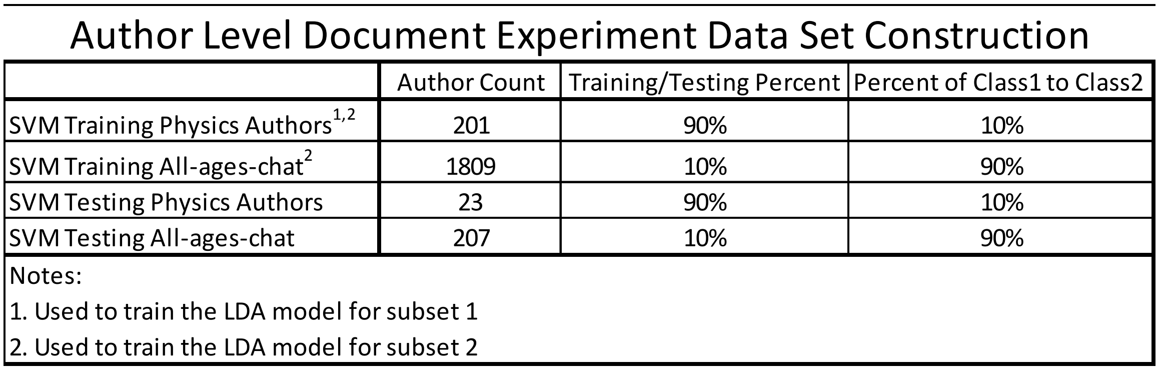
We describe the data set and each experiment's setup as each experiment is expounded. We analyze the results of the experiments as they are presented, and then conclude with collective analysis of the experiments.

A. AUTHOR LEVEL DOCUMENT TRAINED LDA TO PREDICT PHYSICS AUTHORS EXPERIMENTS RESULTS

1. Further Setup

In this set of experiments, we examine the classifier's ability to identify physics related authors amongst a large set of authors who are chatting about various different non-physics related topics. The document concept for this experiment set uses the collective posts produced by individual author to generate the LDA model. In the first subset, we use only authors from physics chat to construct the LDA model. In the second subset, we use authors from both physics and all ages authors to generate the LDA model.

Originally, we started with 283 authors, but left out 69 of the authors with the smallest number of tokens in their chat. This was done to get the proper 90%/10% training-testing proportions leaving us with 224 authors from physics. As mentioned before, we simulate the rarity of physics conversations by creating a 90% all-ages-chat class to 10% physics class split. In all, 201 physics authors were used for training and 23 for testing. We left out eight of the 2,023 all-ages-chat authors, resulting in 1,809 for training and 207 for testing. Table 3 displays the data set configuration for the SVM classifiers for this experiment set. Table 4 displays the Mallet, and LIBSVM configurations used.

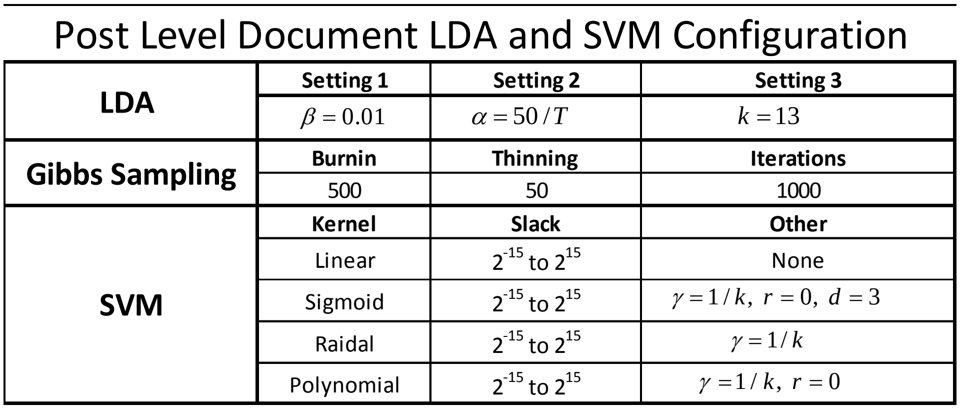


Author Level Document Experiment Data Set Construction

	Author Count	Training/Testing Percent	Percent of Class1 to Class2
SVM Training Physics Authors[1,2]	201	90%	10%
SVM Training All-ages-chat[2]	1809	10%	90%
SVM Testing Physics Authors	23	90%	10%
SVM Testing All-ages-chat	207	10%	90%

Notes:
1. Used to train the LDA model for subset 1
2. Used to train the LDA model for subset 2

Table 3. This table shows the data configuration for the Author Level Document Experiments.



Post Level Document LDA and SVM Configuration

LDA	**Setting 1**	**Setting 2**	**Setting 3**
	$\beta = 0.01$	$\alpha = 50/T$	$k = 13$
Gibbs Sampling	**Burnin**	**Thinning**	**Iterations**
	500	50	1000
SVM	**Kernel**	**Slack**	**Other**
	Linear	2^{-15} to 2^{15}	None
	Sigmoid	2^{-15} to 2^{15}	$\gamma = 1/k,\ r = 0,\ d = 3$
	Raidal	2^{-15} to 2^{15}	$\gamma = 1/k$
	Polynomial	2^{-15} to 2^{15}	$\gamma = 1/k,\ r = 0$

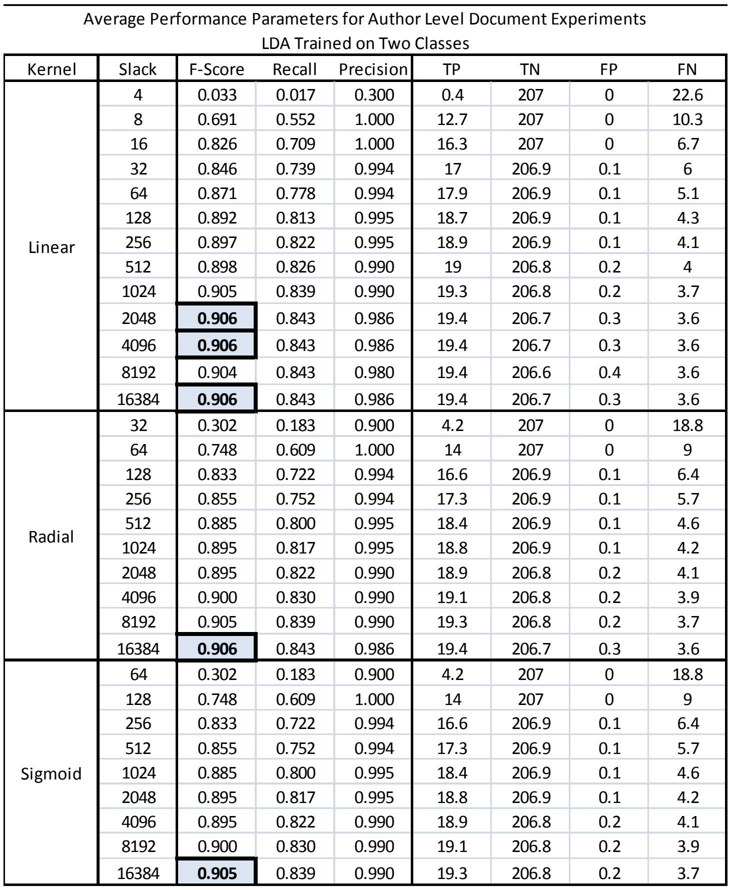
Table 4. This table shows the configuration of LDA, Gibbs Sampling and the SVM model used in constructing the classification system for the Author Level Documents Experiments.

2. First Subset Results

The first subset of experiments for author level document experiments left the negative class out of the LDA model creation process. We performed 10 random samples for all SVM classifier settings of kernel and slack variables, resulting in 1,200 distinct classifiers. The classifiers were unable to distinguish physics authors from non-physics authors. The classifiers identified each test document as belonging to the negative class—all-ages-chat.

3. Second Subset Results

The second subset of experiments for author level document experiments used the positive and negative class documents for LDA model creation. We performed 10 random samples for all SVM classifier settings of kernel and slack variables. The classifier performance from these experiments showed marked performance improvement over the LDA model created in subset one. All of the kernel types, except polynomial, generated multiple classifiers that were able to distinguish physics author documents from all ages author documents. Many of them achieved F-scores above 90%. We provide Table 5, which shows all the results of this experiment subset that attained an F-score above zero.



Average Performance Parameters for Author Level Document Experiments
LDA Trained on Two Classes

Kernel	Slack	F-Score	Recall	Precision	TP	TN	FP	FN
Linear	4	0.033	0.017	0.300	0.4	207	0	22.6
	8	0.691	0.552	1.000	12.7	207	0	10.3
	16	0.826	0.709	1.000	16.3	207	0	6.7
	32	0.846	0.739	0.994	17	206.9	0.1	6
	64	0.871	0.778	0.994	17.9	206.9	0.1	5.1
	128	0.892	0.813	0.995	18.7	206.9	0.1	4.3
	256	0.897	0.822	0.995	18.9	206.9	0.1	4.1
	512	0.898	0.826	0.990	19	206.8	0.2	4
	1024	0.905	0.839	0.990	19.3	206.8	0.2	3.7
	2048	**0.906**	0.843	0.986	19.4	206.7	0.3	3.6
	4096	**0.906**	0.843	0.986	19.4	206.7	0.3	3.6
	8192	0.904	0.843	0.980	19.4	206.6	0.4	3.6
	16384	**0.906**	0.843	0.986	19.4	206.7	0.3	3.6
Radial	32	0.302	0.183	0.900	4.2	207	0	18.8
	64	0.748	0.609	1.000	14	207	0	9
	128	0.833	0.722	0.994	16.6	206.9	0.1	6.4
	256	0.855	0.752	0.994	17.3	206.9	0.1	5.7
	512	0.885	0.800	0.995	18.4	206.9	0.1	4.6
	1024	0.895	0.817	0.995	18.8	206.9	0.1	4.2
	2048	0.895	0.822	0.990	18.9	206.8	0.2	4.1
	4096	0.900	0.830	0.990	19.1	206.8	0.2	3.9
	8192	0.905	0.839	0.990	19.3	206.8	0.2	3.7
	16384	**0.906**	0.843	0.986	19.4	206.7	0.3	3.6
Sigmoid	64	0.302	0.183	0.900	4.2	207	0	18.8
	128	0.748	0.609	1.000	14	207	0	9
	256	0.833	0.722	0.994	16.6	206.9	0.1	6.4
	512	0.855	0.752	0.994	17.3	206.9	0.1	5.7
	1024	0.885	0.800	0.995	18.4	206.9	0.1	4.6
	2048	0.895	0.817	0.995	18.8	206.9	0.1	4.2
	4096	0.895	0.822	0.990	18.9	206.8	0.2	4.1
	8192	0.900	0.830	0.990	19.1	206.8	0.2	3.9
	16384	**0.905**	0.839	0.990	19.3	206.8	0.2	3.7

Table 5. Table displays all SVM classifiers from Author Level experiments where the LDA model was created using both classes of data. All classifiers with an F-score value greater than zero are provided. Maximums are bolded.

These results provide evidence that LDA models of chat data must use both classes of data in order to provide sample distributions of unseen documents suitable for use as feature vectors for SVM classification systems, as the experiments in the first subset performed much worse than those in the second subset. Figure 11 shows the trend of increasing average F-scores with increasing slack values across kernel types. However promising these results may be, in general, the classifiers showed a strong bias for the negative class. Of the 1,200 classifiers built, 889 classified all the documents as all-ages-chat, and 44 classifiers misclassified only one of the all-ages-chat documents.

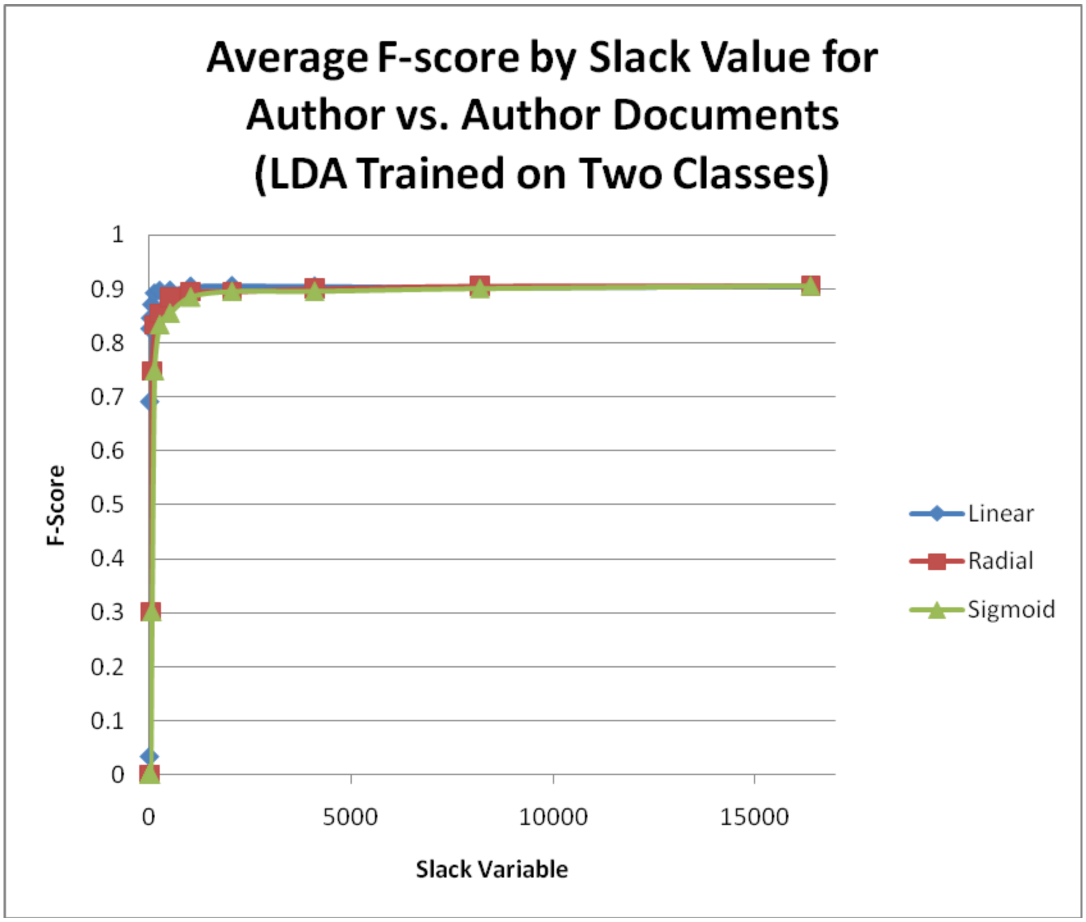
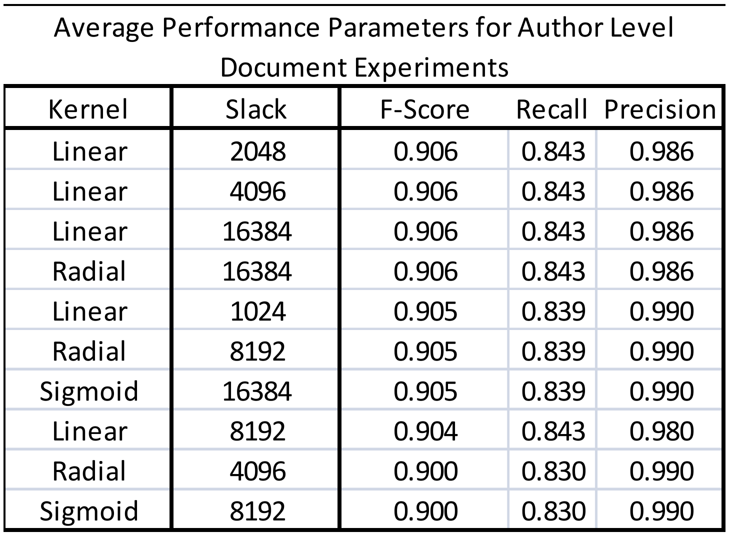


Figure 11. This figure demonstrates the increase average values of F-score with an increase in the slack value for three of the four kernel types. The polynomial kernel (not displayed) did not attained F-scores over zero.

The best performing classifier in this experiment set achieved an F-score of 0.906, Recall of 0.843, and Precision of 0.986. Table 7 provides the 10 best performing classifiers based on average F-score. Five of the ten top-performing classifiers had linear kernels, their slack variables ranged from 1024 to 16384.



Average Performance Parameters for Author Level Document Experiments

Kernel	Slack	F-Score	Recall	Precision
Linear	2048	0.906	0.843	0.986
Linear	4096	0.906	0.843	0.986
Linear	16384	0.906	0.843	0.986
Radial	16384	0.906	0.843	0.986
Linear	1024	0.905	0.839	0.990
Radial	8192	0.905	0.839	0.990
Sigmoid	16384	0.905	0.839	0.990
Linear	8192	0.904	0.843	0.980
Radial	4096	0.900	0.830	0.990
Sigmoid	8192	0.900	0.830	0.990

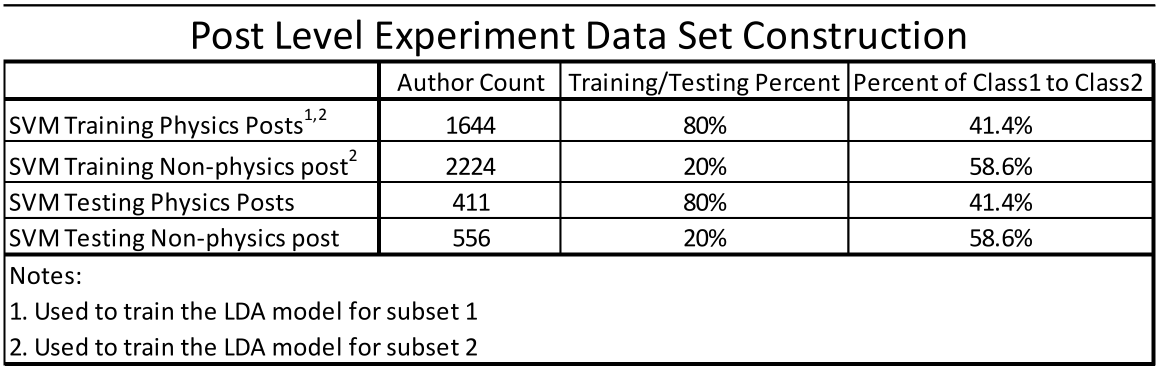
Table 6. The Top 10 Best Performing Classifier for Author Level Documents.

B. POST LEVEL DOCUMENT TRAINED LDA TO PREDICT PHYSICS POSTS EXPERIMENTS RESULTS

1. Further Setup

In this set of experiments, individual posts in physics chat room form the basis for generating the LDA model and SVM classifier. We examine the classifier’s ability to identify physics related posts from a physics chat room amongst posts from the same chat room, but were not about physics. As noted in Chapter III, the physics chat corpus is composed of 7803 posts. After removing the system generated posts, the corpus retrained 4966 post of which we removed six of the smallest posts to obtain the correct training and testing mixture: 80% training-20% testing. In the first subset, we train the LDA model using only the posts labeled as physics; in the second subset, we use both posts hand labeled as physics and posts hand labeled as non-physics.

Using the 4960 posts, we split the data into testing and training groups. We used 20% for training and 80% for testing. Within both of these groups, we retained a class split of 41.4% physics class to 58.6% non-physics class (this is approximately the same as the inherent 41.5% physics class to 58.5% non-physics class that naturally occurs in the original data set). Table 7 displays the data set configuration for the SVM classifiers for this experiment set. Table 8 displays the Mallet, and LIBSVM configurations.

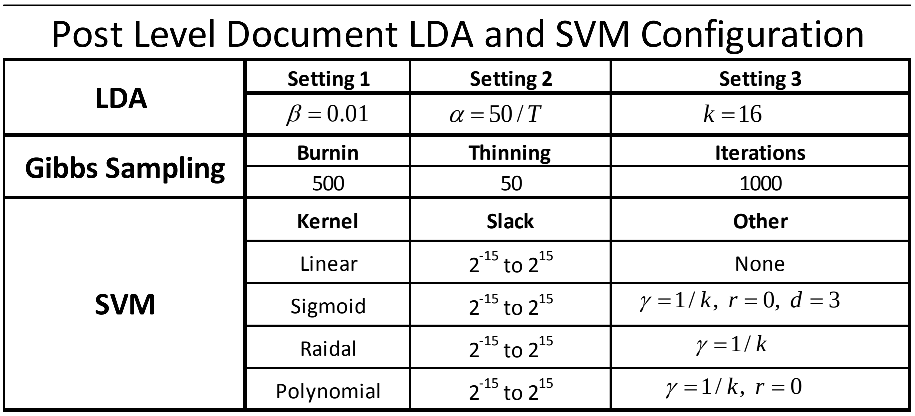


Post Level Experiment Data Set Construction

	Author Count	Training/Testing Percent	Percent of Class1 to Class2
SVM Training Physics Posts[1,2]	1644	80%	41.4%
SVM Training Non-physics post[2]	2224	20%	58.6%
SVM Testing Physics Posts	411	80%	41.4%
SVM Testing Non-physics post	556	20%	58.6%

Notes:
1. Used to train the LDA model for subset 1
2. Used to train the LDA model for subset 2

Table 7. This table shows the data configuration for the Post Level Document Experiments.



Post Level Document LDA and SVM Configuration

LDA	Setting 1	Setting 2	Setting 3
	$\beta = 0.01$	$\alpha = 50 / T$	$k = 16$
Gibbs Sampling	Burnin	Thinning	Iterations
	500	50	1000
SVM	Kernel	Slack	Other
	Linear	2^{-15} to 2^{15}	None
	Sigmoid	2^{-15} to 2^{15}	$\gamma = 1/k,\ r = 0,\ d = 3$
	Raidal	2^{-15} to 2^{15}	$\gamma = 1/k$
	Polynomial	2^{-15} to 2^{15}	$\gamma = 1/k,\ r = 0$

Table 8. This table shows the configuration of LDA, Gibbs Sampling and the SVM model used in constructing the classification system for the Post Level Document Experiments.

2. First Subset Results

In the first subset, we created SVM classifier varying the settings of kernel and slack variables for 10 random samples of the data, resulting in 1,200 distinct classifiers. The classifier performance from these experiments shows limited ability to distinguish physics posts from non-physics posts. Figures 12, 13 and 14 display the classifier F-scores by slack value. They are arranged by linear, radial, and sigmoid kernels respectively. Each figure includes the maximum, minimum, and average F-scores.

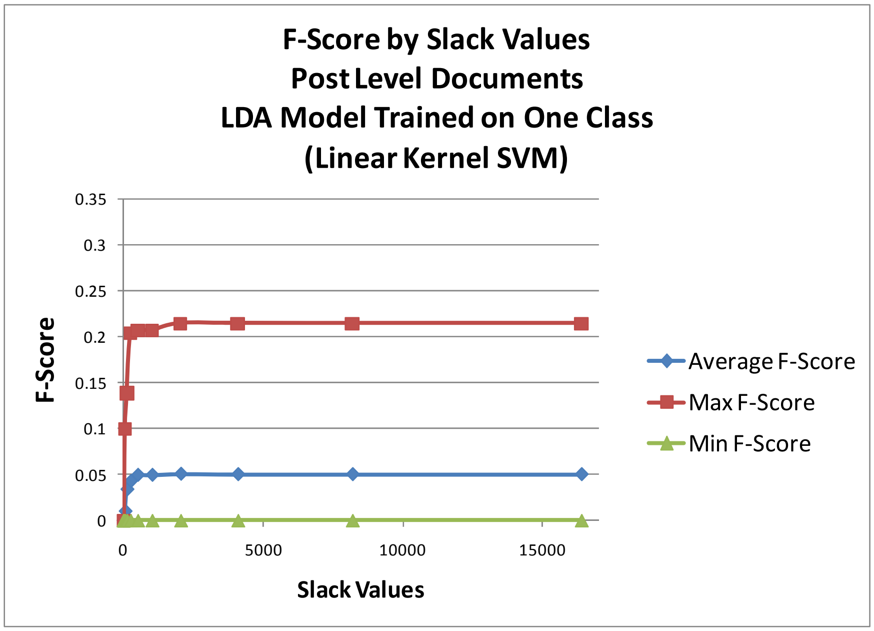


Figure 12. Post Level Document SVM Classifier F-scores by Slack Value: Linear kernel.

Figure 12 shows that the linear kernel performance is highly dependent on the particular sample used for the experiment. The minimum F-score line remains zero for all slack values, indicating that there was at least one random sample that yielded a classifier that could not identify a single physics post level document. In fact, a random sample that creates an F-score greater than zero is the exception. This pattern holds for the radial and sigmoid kernels as well.

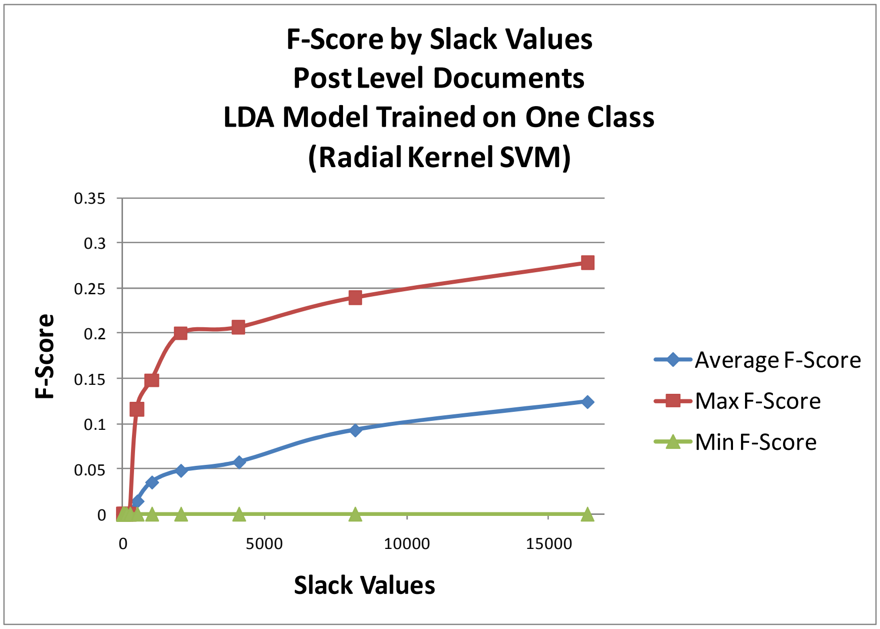


Figure 13. Post Level Document SVM Classifier F-scores by Slack Value: Radial kernel.

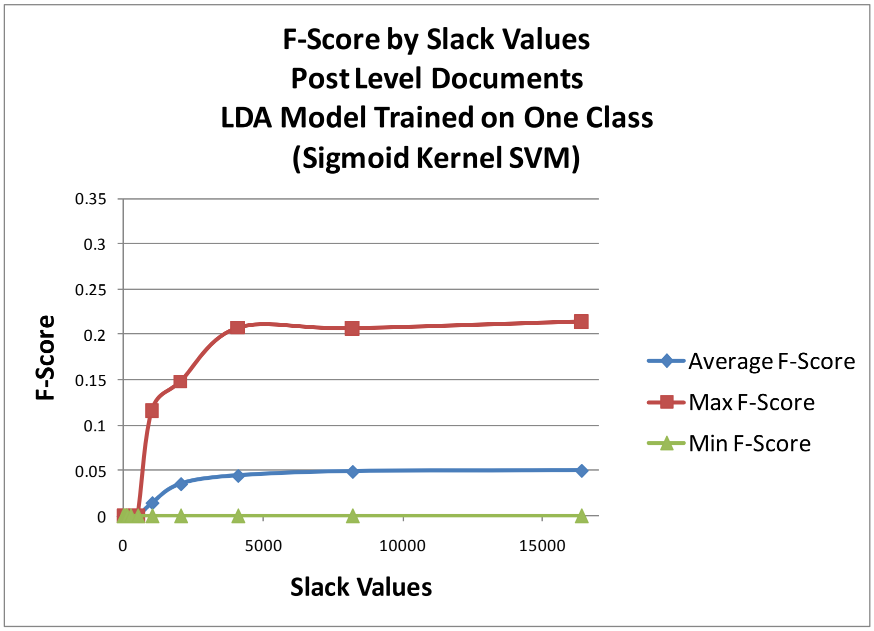


Figure 14. Post Level Document SVM Classifier F-scores by Slack Value: Sigmoid kernel.

Although, these results are rather low, the classifiers performed better than the author level document experiments whose LDA models were created with only one class. This may have been caused by the more equitable mixture between the positive and negative classes (41.4% - 58.6% for posts; and 90% - 10% for authors). It may have also been a result of using data from two radically different sources. That is, in the author level document experiments we used authors in a physics chat room and compared them to the posts of authors in several different socially oriented chat rooms. While in the post level document experiments, we used posts that came from the exact same chat room.

On the other hand, it may be the case that post level documents have characteristics that are more discriminating than the collective posts of an individual author, e.g., an author is more "noisy" than any one of his or her posts. This experiment subset also shows a general trend of increasing F-score performance as the slack value is increased, regardless of the kernel type. Table 9 shows all of the non-zero average performance metrics for the linear, radial and sigmoid kernels. Figure 15 combines the average F-scores of linear, radial and sigmoid kernels.

Average Performance Parameters for Post Level Document Experiments
LDA Trained on One Class

Kernel	Slack	F-score	Recall	Precision	TP	TN	FP	FN
Linear	64	0.010	0.005	0.071	2.2	555.1	0.9	408.8
	128	0.034	0.018	0.193	7.6	551.9	4.1	403.4
	256	0.042	0.024	0.285	9.9	550.5	5.5	401.1
	512	0.049	0.028	0.250	11.4	549.5	6.5	399.6
	1024	0.049	0.028	0.250	11.4	549.5	6.5	399.6
	2048	0.050	0.028	0.252	11.7	549.5	6.5	399.3
	4096	0.050	0.028	0.249	11.6	549.4	6.6	399.4
	8192	0.050	0.028	0.249	11.6	549.4	6.6	399.4
	16384	**0.050**	0.028	0.249	11.6	549.4	6.6	399.4
Radial	512	0.015	0.008	0.270	3.2	554.9	1.1	407.8
	1024	0.035	0.020	0.190	8.1	551.5	4.5	402.9
	2048	0.048	0.027	0.351	11.2	549.7	6.3	399.8
	4096	0.058	0.032	0.376	13.3	548.3	7.7	397.7
	8192	0.093	0.053	0.602	21.7	544.3	11.7	389.3
	16384	**0.124**	0.072	0.613	29.7	541.4	14.6	381.3
Sigmoid	1024	0.015	0.008	0.270	3.2	554.9	1.1	407.8
	2048	0.035	0.020	0.190	8.1	551.5	4.5	402.9
	4096	0.045	0.025	0.249	10.4	550.1	5.9	400.6
	8192	0.049	0.028	0.250	11.4	549.5	6.5	399.6
	16384	**0.050**	0.028	0.252	11.7	549.5	6.5	399.3

Table 9. Average Performance metrics: Post Level Document Experiments—LDA trained on one class. Maximum for each kernel type is bolded.

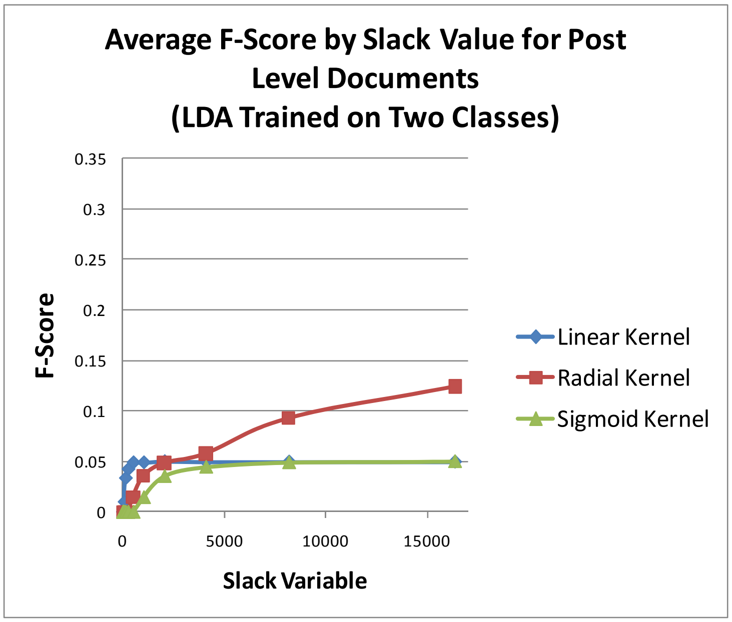


Figure 15. Average F-score by Slack Value: Post Level Documents—LDA trained on One Class.

3. Second Subset Results

The second subset of the post level document experiment set included both classes to create the LDA model. It performed better than the first subset. We used the same random samples, slack and kernel settings used for subset one, again creating 1,200 distinct classifiers. In general, the classifiers in this subset out-performed those in the previous subset. The best performing classifier used a linear kernel with a slack value of 16384. It obtained an F-score of 0.481, precision of 0.375, and recall of .677. This maximum performance is better than the previous experiment subset. Table 10 records the average values of each classifier whose average F-score is greater than zero.

Average Performance Parameters for Post Level Document Experiments
LDA Trained on Two Classes

Kernel	Slack	Recall	Precision	F-Score	TP	TN	FP	FN
Linear	8	0.054	0.029	0.705	11.9	553.3	2.7	399.1
	16	0.315	0.205	0.732	84.4	523.9	32.1	326.6
	32	0.421	0.302	0.702	124.3	502.7	53.3	286.7
	64	0.449	0.333	0.695	137	495.5	60.5	274
	128	0.465	0.352	0.688	144.8	490	66	266.2
	256	0.474	0.365	0.684	149.9	485.9	70.1	261.1
	512	0.477	0.368	0.684	151.3	485.5	70.5	259.7
	1024	0.479	0.373	0.678	153.4	481.9	74.1	257.6
	2048	0.479	0.373	0.677	153.5	481.7	74.3	257.5
	4096	**0.481**	0.375	0.677	154.1	481.6	74.4	256.9
	8192	0.480	0.375	0.676	154.1	481	75	256.9
	16384	0.480	0.374	0.677	153.9	481.5	74.5	257.1
Radial	32	0.001	0.000	0.100	0.2	556	0	410.8
	64	0.126	0.071	0.709	29.3	546.4	9.6	381.7
	128	0.359	0.243	0.710	99.8	514.7	41.3	311.2
	256	0.430	0.313	0.699	128.5	500.2	55.8	282.5
	512	0.454	0.340	0.690	139.8	492.9	63.1	271.2
	1024	0.467	0.357	0.686	146.6	488.2	67.8	264.4
	2048	0.474	0.364	0.685	149.6	486.6	69.4	261.4
	4096	0.476	0.367	0.681	151	484.8	71.2	260
	8192	0.476	0.368	0.681	151.1	484.7	71.3	259.9
	16384	**0.478**	0.370	0.682	152	484.4	71.6	259
Sigmoid	64	0.001	0.000	0.100	0.2	556	0	410.8
	128	0.126	0.071	0.709	29.3	546.4	9.6	381.7
	256	0.359	0.243	0.710	99.7	514.8	41.2	311.3
	512	0.431	0.313	0.700	128.7	500.2	55.8	282.3
	1024	0.454	0.340	0.690	139.8	492.9	63.1	271.2
	2048	0.469	0.358	0.685	147.3	487.6	68.4	263.7
	4096	0.475	0.366	0.685	150.5	486	70	260.5
	8192	0.477	0.369	0.681	151.8	484.1	71.9	259.2
	16384	**0.479**	0.373	0.677	153.2	481.9	74.1	257.8

Table 10. Average Performance Parameters Results: Post Level Document Experiments—LDA Trained on Two Classes

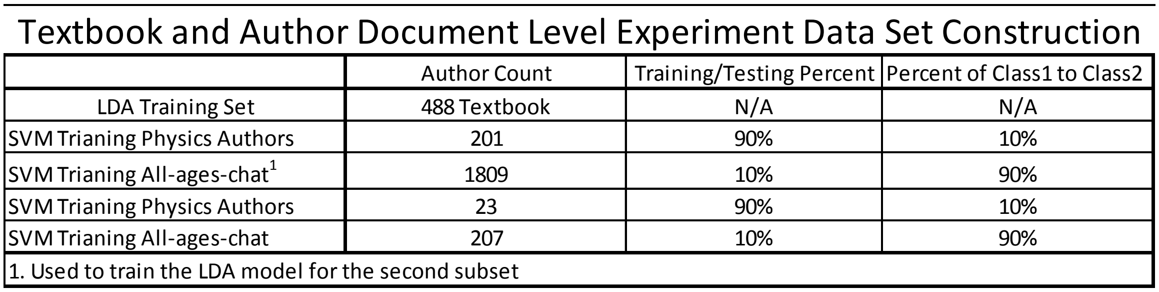
While, these set of experiments perform better than the previous subset, they fail to outperform their author level document analog. In fact, the F-score values are approximately half for each corresponding setting of kernel and slack value. This may be due difference class mixtures. It may have been caused by the fact that the physics authors chat was compared to all-ages-chat, which from a completely different set of chat rooms, while the physics posts and non-physics posts came from the same chat room making them more difficult to distinguish from one another.

C. TEXTBOOK AND AUTHOR LEVEL DOCUMENT TRAINED LDA TO PREDICT PHYSICS AUTHORS EXPERIMENTS RESULTS

1. Further Setup

In this set of experiments, we examined the classifier's ability to identify authors discussing physics from authors discussing socially oriented topics after training the LDA topic model on the textbook paragraph level documents. In the first subset, the LDA model was created using the textbook paragraphs. The SVM model was then trained and tested using the same author level documents as used in the first experiment set. In the second subset, the all-ages-chat author level documents are combined with the textbook paragraph documents for LDA model creation.

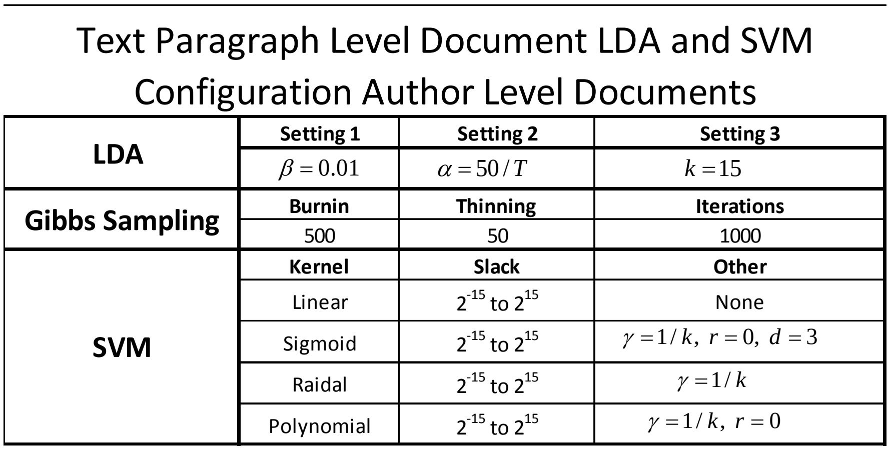
Table 11 displays the data set configuration for the SVM classifiers for this experiment set. Table 12 displays the Mallet, and LIBSVM configurations.



Textbook and Author Document Level Experiment Data Set Construction

	Author Count	Training/Testing Percent	Percent of Class1 to Class2
LDA Training Set	488 Textbook	N/A	N/A
SVM Trianing Physics Authors	201	90%	10%
SVM Trianing All-ages-chat[1]	1809	10%	90%
SVM Trianing Physics Authors	23	90%	10%
SVM Trianing All-ages-chat	207	10%	90%
1. Used to train the LDA model for the second subset			

Table 11. This table shows the data configuration for the Textbook and Author Level Document Experiments.



Text Paragraph Level Document LDA and SVM Configuration Author Level Documents

LDA	Setting 1	Setting 2	Setting 3
	$\beta = 0.01$	$\alpha = 50 / T$	$k = 15$
Gibbs Sampling	Burnin	Thinning	Iterations
	500	50	1000
SVM	Kernel	Slack	Other
	Linear	2^{-15} to 2^{15}	None
	Sigmoid	2^{-15} to 2^{15}	$\gamma = 1/k,\ r = 0,\ d = 3$
	Raidal	2^{-15} to 2^{15}	$\gamma = 1/k$
	Polynomial	2^{-15} to 2^{15}	$\gamma = 1/k,\ r = 0$

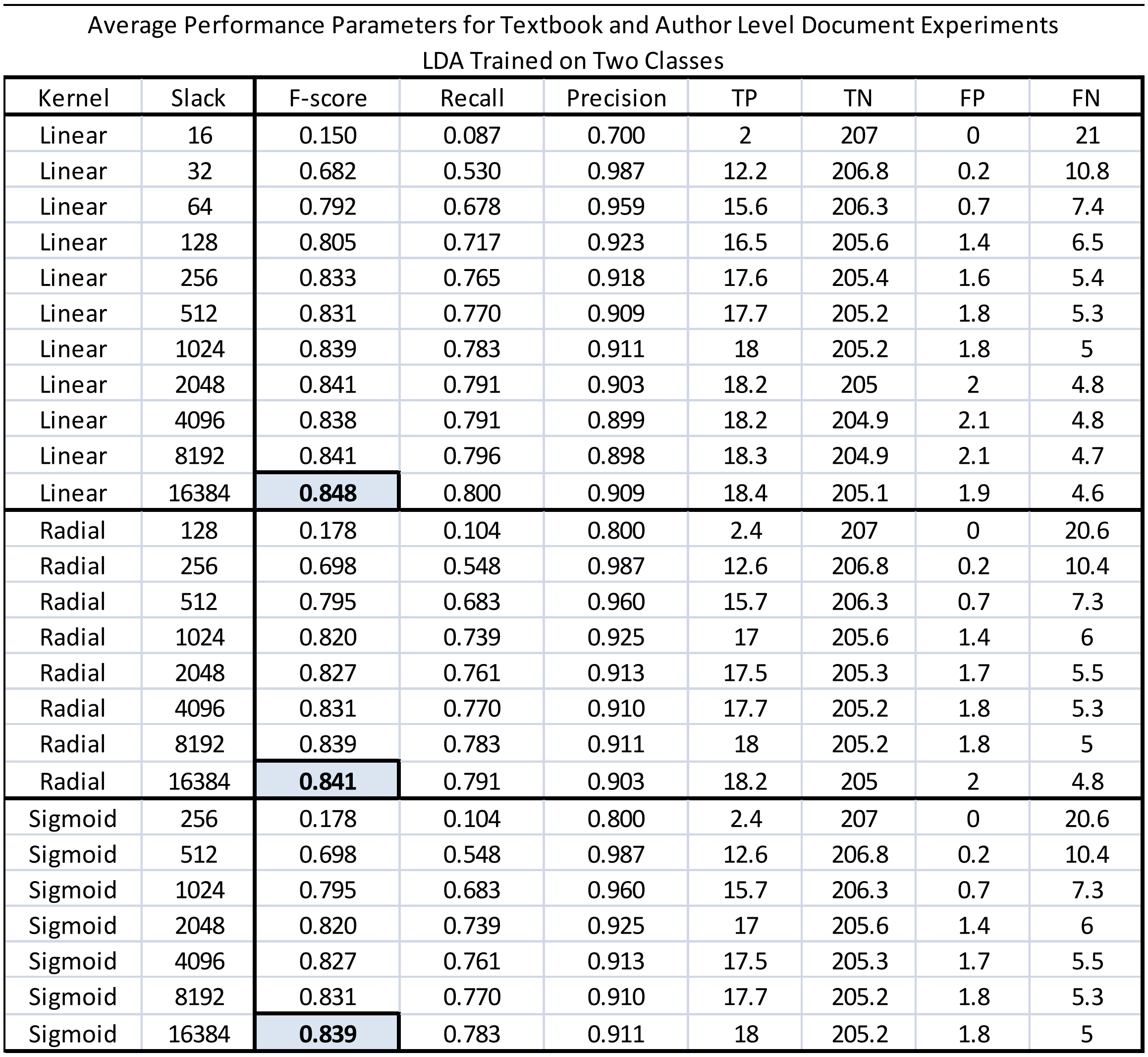
Table 12. This table shows the configuration of LDA, Gibbs Sampling and the SVM model used in constructing the classification system for the Textbook Paragraph and Author Level Document Experiments.

2. First Subset Results

The classifiers in the first subset of these experiments failed to identify a single positive class author document; instead, it classified each document as an all-ages-chat document. No table is provided for these results.

3. Second Subset Results

The second subset, however, performed better. Table 13 shows the average major performance metrics for classifiers that obtained F-scores greater than zero. The best performing classifier used a linear kernel with a slack value of 16384. It obtained an F-score of 0.848, recall of 0.800, and precision of 0.909. This maximum performance is better than the previous experiment subset. Table 13 records the average values of each classifier whose average F-score is greater than zero.



Average Performance Parameters for Textbook and Author Level Document Experiments
LDA Trained on Two Classes

Kernel	Slack	F-score	Recall	Precision	TP	TN	FP	FN
Linear	16	0.150	0.087	0.700	2	207	0	21
Linear	32	0.682	0.530	0.987	12.2	206.8	0.2	10.8
Linear	64	0.792	0.678	0.959	15.6	206.3	0.7	7.4
Linear	128	0.805	0.717	0.923	16.5	205.6	1.4	6.5
Linear	256	0.833	0.765	0.918	17.6	205.4	1.6	5.4
Linear	512	0.831	0.770	0.909	17.7	205.2	1.8	5.3
Linear	1024	0.839	0.783	0.911	18	205.2	1.8	5
Linear	2048	0.841	0.791	0.903	18.2	205	2	4.8
Linear	4096	0.838	0.791	0.899	18.2	204.9	2.1	4.8
Linear	8192	0.841	0.796	0.898	18.3	204.9	2.1	4.7
Linear	16384	**0.848**	0.800	0.909	18.4	205.1	1.9	4.6
Radial	128	0.178	0.104	0.800	2.4	207	0	20.6
Radial	256	0.698	0.548	0.987	12.6	206.8	0.2	10.4
Radial	512	0.795	0.683	0.960	15.7	206.3	0.7	7.3
Radial	1024	0.820	0.739	0.925	17	205.6	1.4	6
Radial	2048	0.827	0.761	0.913	17.5	205.3	1.7	5.5
Radial	4096	0.831	0.770	0.910	17.7	205.2	1.8	5.3
Radial	8192	0.839	0.783	0.911	18	205.2	1.8	5
Radial	16384	**0.841**	0.791	0.903	18.2	205	2	4.8
Sigmoid	256	0.178	0.104	0.800	2.4	207	0	20.6
Sigmoid	512	0.698	0.548	0.987	12.6	206.8	0.2	10.4
Sigmoid	1024	0.795	0.683	0.960	15.7	206.3	0.7	7.3
Sigmoid	2048	0.820	0.739	0.925	17	205.6	1.4	6
Sigmoid	4096	0.827	0.761	0.913	17.5	205.3	1.7	5.5
Sigmoid	8192	0.831	0.770	0.910	17.7	205.2	1.8	5.3
Sigmoid	16384	**0.839**	0.783	0.911	18	205.2	1.8	5

Table 13. Average Performance Parameters Results: Textbook and Author Paragraph Level Document Experiments—LDA Trained on Two Classes

Table 13 highlights some interesting differences between the classifiers produced by LDA models created by positive and negative class author documents as opposed to those created by the textbook and the author negative class. Although the physics author trained classifiers performed better than the textbook-trained classifier, the difference between the two in most cases is less than 10%. Figures 16, 17, and 18 highlight this observation.

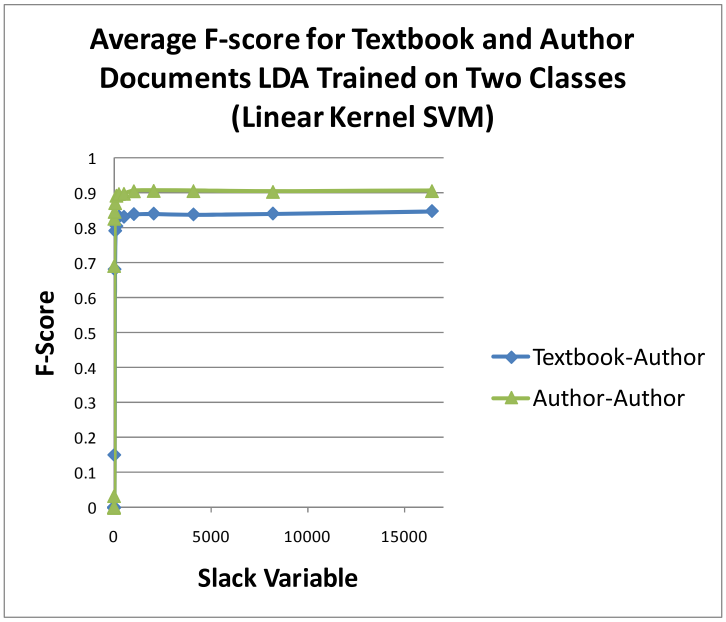


Figure 16. Classifiers Results: LDA Models Created by Textbook-Author Documents and Author-Author Documents—Linear Kernel.

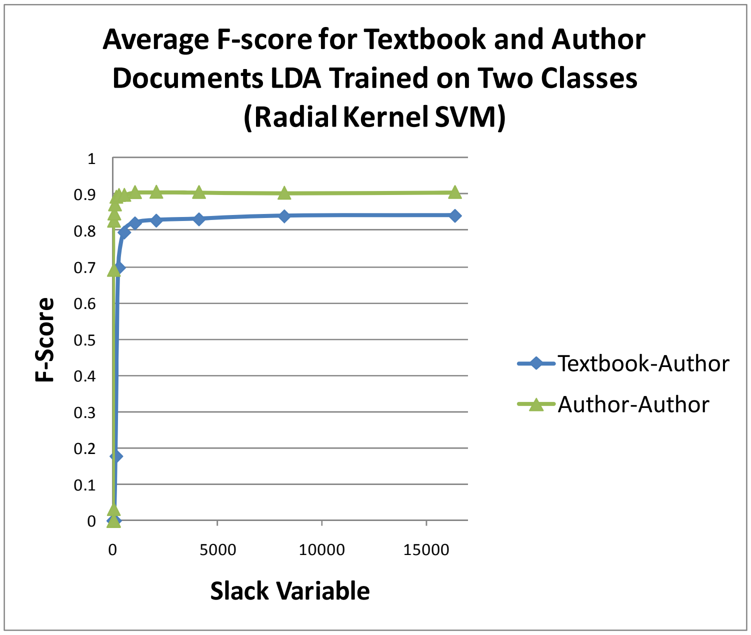


Figure 17. Classifiers Results: LDA Models Created by Textbook-Author Documents and Author-Author Documents—Radial Kernel.

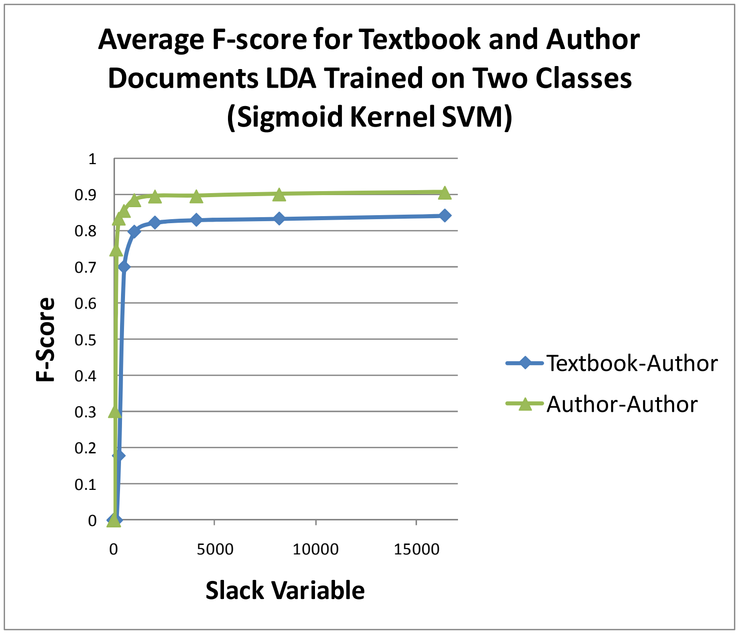


Figure 18. Classifiers Results: LDA Models Created by Textbook-Author Documents and Author-Author Documents—Sigmoid Kernel.

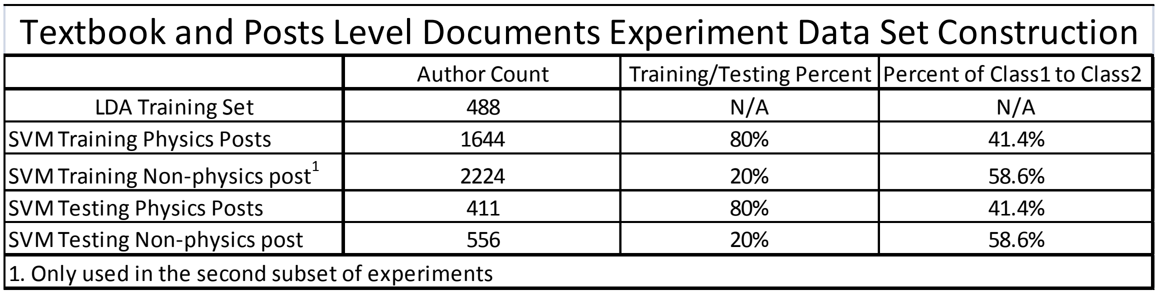
These results indicate that there may be some degree of interchangeability between the traditional written language domain and the chat domain across subjects.

D. TEXTBOOK AND POST LEVEL DOCUMENT TRAINED LDA TO PREDICT PHYSICS POSTS EXPERIMENTS RESULTS

1. Further Setup

In this set of experiments, we examine the classifier’s ability to identify posts about physics in a physics chat room after training the LDA topic model on the textbook paragraph level documents. The first subset used no negative class for LDA model creation and the second subset used post level documents from the non-physics chat posts along with the textbook paragraph documents to create the LDA model—the positive class.

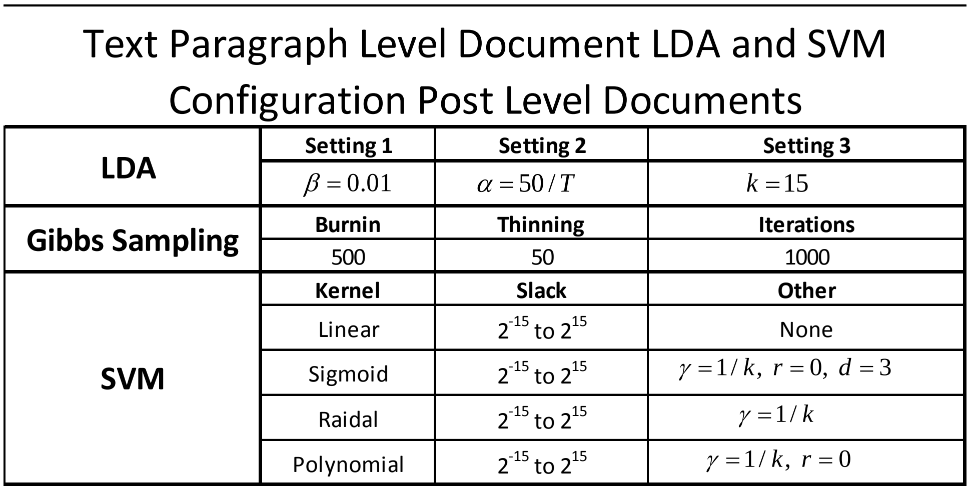
Table 14 displays the data set configuration for the SVM classifiers for this experiment set. Table 15 displays the Mallet, and LIBSVM configurations.



Textbook and Posts Level Documents Experiment Data Set Construction

	Author Count	Training/Testing Percent	Percent of Class1 to Class2
LDA Training Set	488	N/A	N/A
SVM Training Physics Posts	1644	80%	41.4%
SVM Training Non-physics post[1]	2224	20%	58.6%
SVM Testing Physics Posts	411	80%	41.4%
SVM Testing Non-physics post	556	20%	58.6%
1. Only used in the second subset of experiments			

Table 14. This table shows the data configuration for the Textbook Paragraph and Post Level Document Experiments.



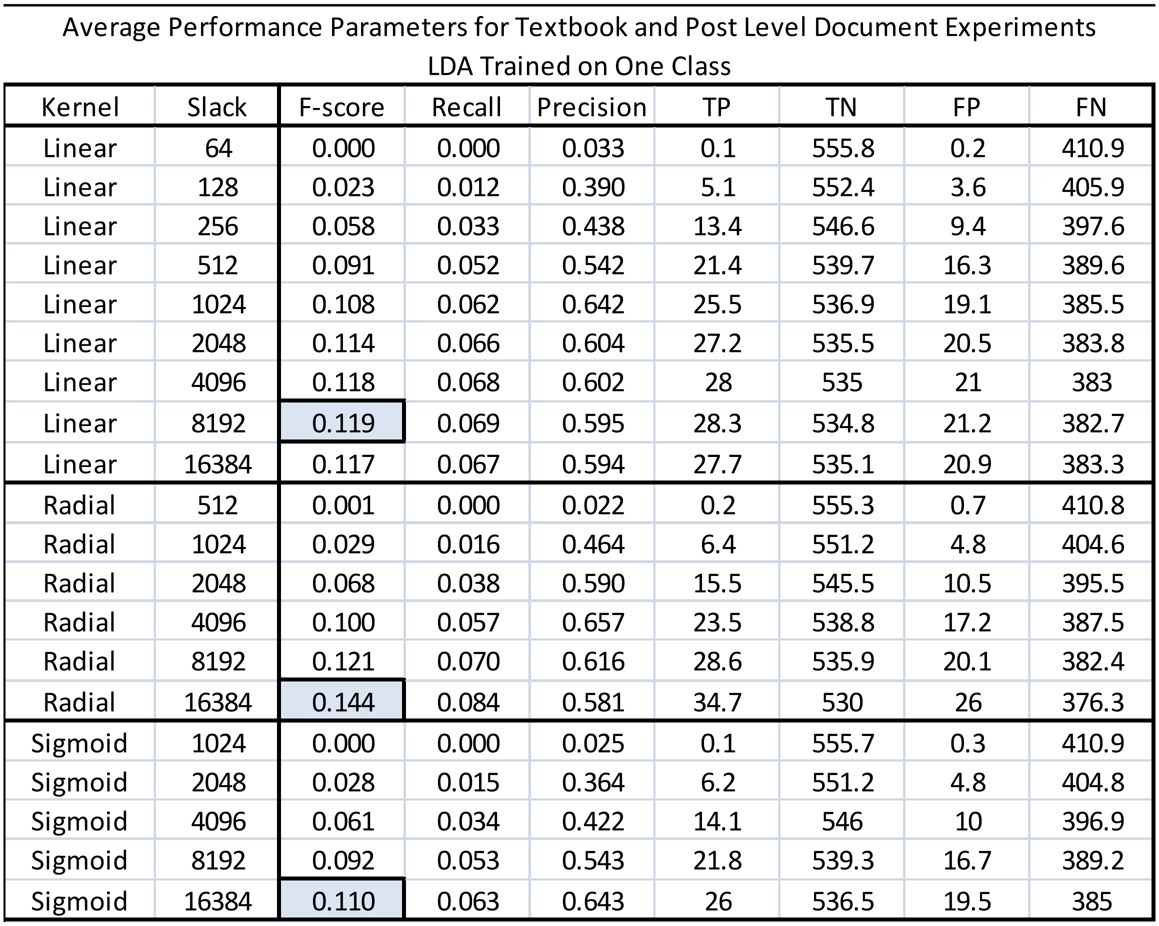
Text Paragraph Level Document LDA and SVM Configuration Post Level Documents

LDA	Setting 1	Setting 2	Setting 3
	$\beta = 0.01$	$\alpha = 50/T$	$k = 15$
Gibbs Sampling	Burnin	Thinning	Iterations
	500	50	1000
SVM	Kernel	Slack	Other
	Linear	2^{-15} to 2^{15}	None
	Sigmoid	2^{-15} to 2^{15}	$\gamma = 1/k,\ r = 0,\ d = 3$
	Raidal	2^{-15} to 2^{15}	$\gamma = 1/k$
	Polynomial	2^{-15} to 2^{15}	$\gamma = 1/k,\ r = 0$

Table 15. This table shows the configuration of LDA, Gibbs Sampling and the SVM model used in constructing the classification system for the Textbook Paragraph and Post Level Documents Experiments.

2. First Subset Results

The first subset of the textbook and post level document experiment set performed better than the post level document experiment set, where the LDA model was trained on physics posts alone. Its best performing classifier used a radial kernel and had a slack value of 16384. The F-score was 0.144, had a recall of 0.084, and a precision of 0.581. Table 16 shows the average major performance metrics for classifiers that obtained F-scores greater than zero.



Average Performance Parameters for Textbook and Post Level Document Experiments
LDA Trained on One Class

Kernel	Slack	F-score	Recall	Precision	TP	TN	FP	FN
Linear	64	0.000	0.000	0.033	0.1	555.8	0.2	410.9
Linear	128	0.023	0.012	0.390	5.1	552.4	3.6	405.9
Linear	256	0.058	0.033	0.438	13.4	546.6	9.4	397.6
Linear	512	0.091	0.052	0.542	21.4	539.7	16.3	389.6
Linear	1024	0.108	0.062	0.642	25.5	536.9	19.1	385.5
Linear	2048	0.114	0.066	0.604	27.2	535.5	20.5	383.8
Linear	4096	0.118	0.068	0.602	28	535	21	383
Linear	8192	0.119	0.069	0.595	28.3	534.8	21.2	382.7
Linear	16384	0.117	0.067	0.594	27.7	535.1	20.9	383.3
Radial	512	0.001	0.000	0.022	0.2	555.3	0.7	410.8
Radial	1024	0.029	0.016	0.464	6.4	551.2	4.8	404.6
Radial	2048	0.068	0.038	0.590	15.5	545.5	10.5	395.5
Radial	4096	0.100	0.057	0.657	23.5	538.8	17.2	387.5
Radial	8192	0.121	0.070	0.616	28.6	535.9	20.1	382.4
Radial	16384	0.144	0.084	0.581	34.7	530	26	376.3
Sigmoid	1024	0.000	0.000	0.025	0.1	555.7	0.3	410.9
Sigmoid	2048	0.028	0.015	0.364	6.2	551.2	4.8	404.8
Sigmoid	4096	0.061	0.034	0.422	14.1	546	10	396.9
Sigmoid	8192	0.092	0.053	0.543	21.8	539.3	16.7	389.2
Sigmoid	16384	0.110	0.063	0.643	26	536.5	19.5	385

Table 16. Average Performance Parameters Results: Textbook and Post Paragraph Level Document Experiments—LDA Trained on One Class

This subset outperformed the post level document experiment set whose LDA model was trained solely on the positive class. Figures 19, 20 and 21 show the improvement. In each case, the F-score nearly doubles.

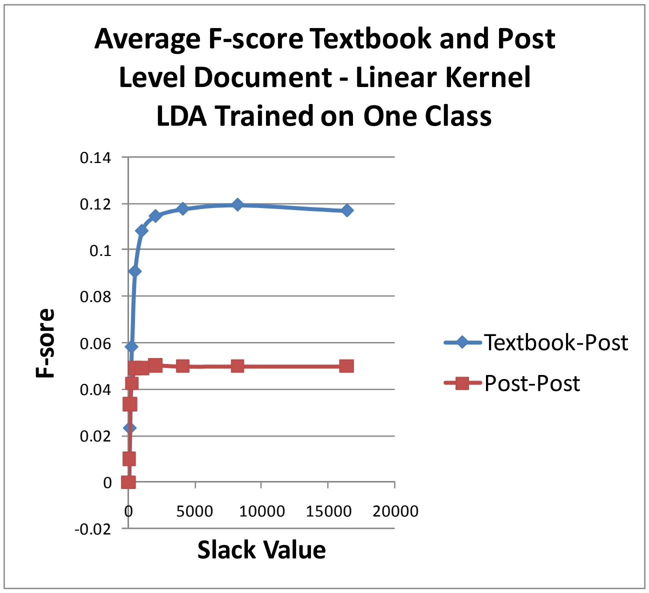


Figure 19. Average F-score Comparison between Post Level Document Classifiers and Textbook Paragraph Level Documents Classifiers. LDA models trained solely on the positive class. Linear Kernel Average

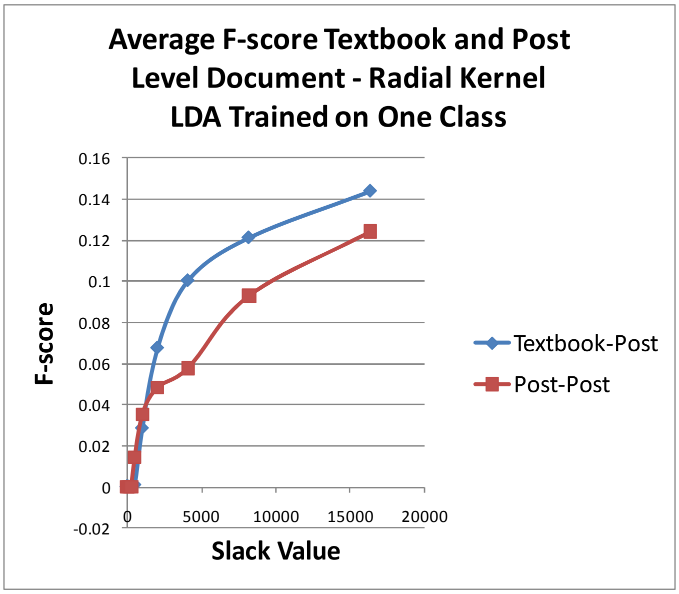


Figure 20. F-score Comparison between Post Level Document Classifiers and Textbook Paragraph Level Documents Classifiers. LDA models trained solely on the positive class. Radial Kernel

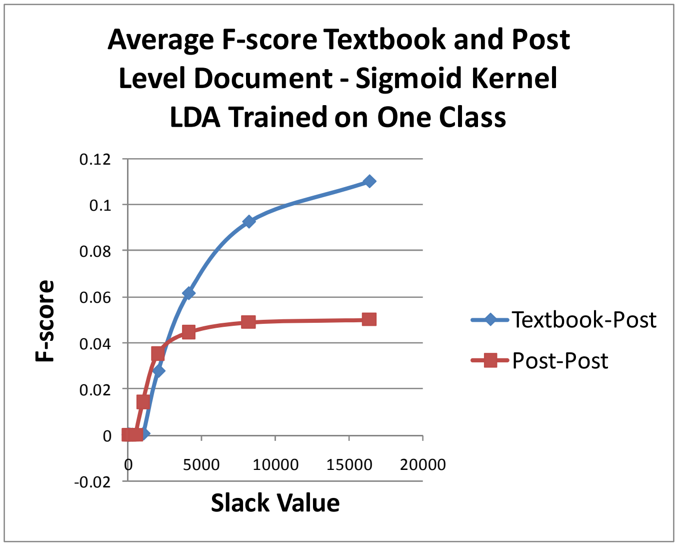
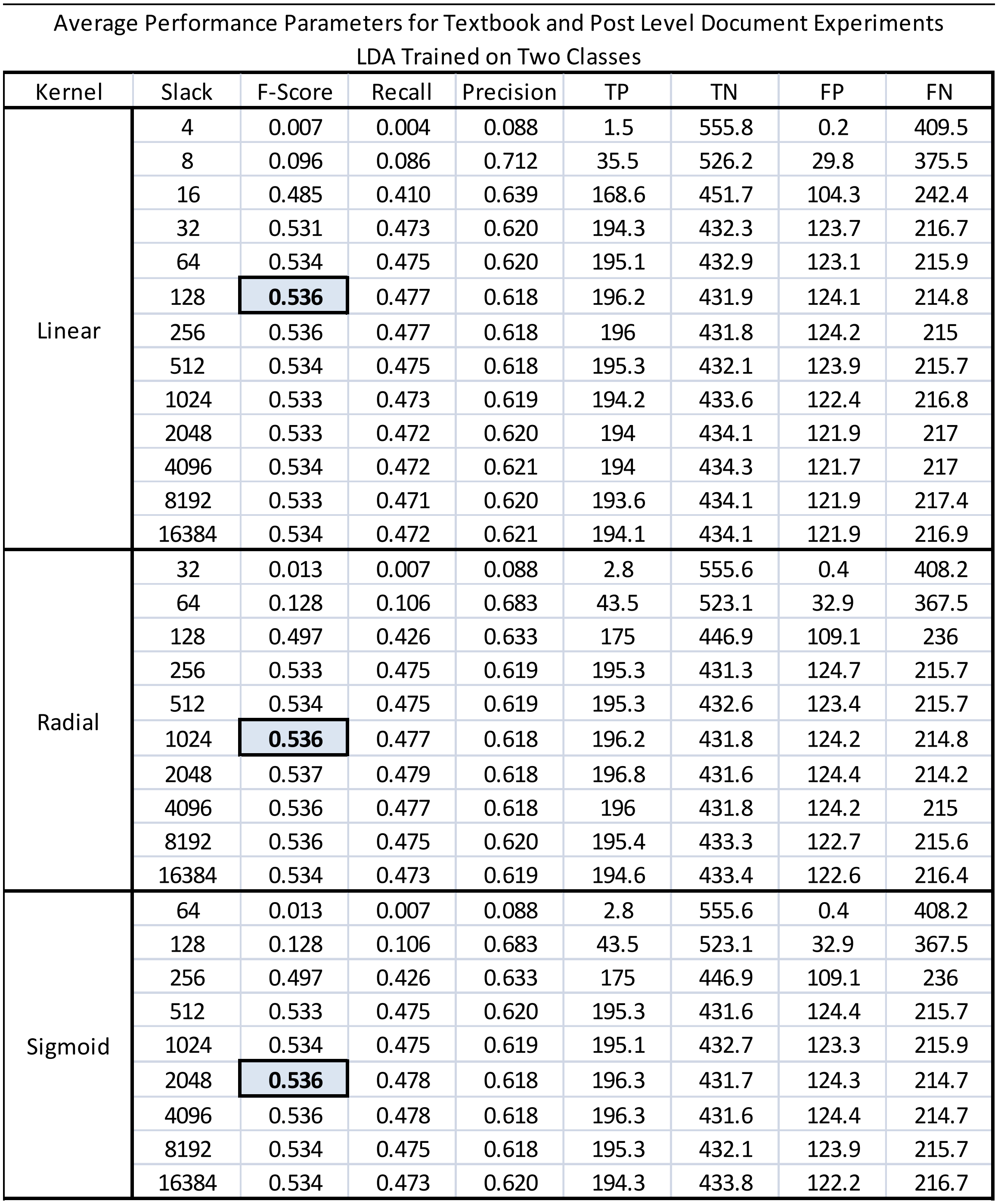


Figure 21. F-score Comparison between Post Level Document Classifiers and Textbook Paragraph Level Documents Classifiers. LDA models trained solely on the positive class. Sigmoid Kernel

3. Second Subset Results

It performed better than the first subset. We used the same random samples, slack and kernel settings used for subset one, again creating 1200 distinct classifiers. Several classifiers generated the highest obtained F-score. Linear, radial and sigmoid kernel classifiers all obtained an F-score of 0.536. Table 17 records the average metrics of each classifier whose average F-score is greater than zero.



Average Performance Parameters for Textbook and Post Level Document Experiments
LDA Trained on Two Classes

Kernel	Slack	F-Score	Recall	Precision	TP	TN	FP	FN
Linear	4	0.007	0.004	0.088	1.5	555.8	0.2	409.5
	8	0.096	0.086	0.712	35.5	526.2	29.8	375.5
	16	0.485	0.410	0.639	168.6	451.7	104.3	242.4
	32	0.531	0.473	0.620	194.3	432.3	123.7	216.7
	64	0.534	0.475	0.620	195.1	432.9	123.1	215.9
	128	**0.536**	0.477	0.618	196.2	431.9	124.1	214.8
	256	0.536	0.477	0.618	196	431.8	124.2	215
	512	0.534	0.475	0.618	195.3	432.1	123.9	215.7
	1024	0.533	0.473	0.619	194.2	433.6	122.4	216.8
	2048	0.533	0.472	0.620	194	434.1	121.9	217
	4096	0.534	0.472	0.621	194	434.3	121.7	217
	8192	0.533	0.471	0.620	193.6	434.1	121.9	217.4
	16384	0.534	0.472	0.621	194.1	434.1	121.9	216.9
Radial	32	0.013	0.007	0.088	2.8	555.6	0.4	408.2
	64	0.128	0.106	0.683	43.5	523.1	32.9	367.5
	128	0.497	0.426	0.633	175	446.9	109.1	236
	256	0.533	0.475	0.619	195.3	431.3	124.7	215.7
	512	0.534	0.475	0.619	195.3	432.6	123.4	215.7
	1024	**0.536**	0.477	0.618	196.2	431.8	124.2	214.8
	2048	0.537	0.479	0.618	196.8	431.6	124.4	214.2
	4096	0.536	0.477	0.618	196	431.8	124.2	215
	8192	0.536	0.475	0.620	195.4	433.3	122.7	215.6
	16384	0.534	0.473	0.619	194.6	433.4	122.6	216.4
Sigmoid	64	0.013	0.007	0.088	2.8	555.6	0.4	408.2
	128	0.128	0.106	0.683	43.5	523.1	32.9	367.5
	256	0.497	0.426	0.633	175	446.9	109.1	236
	512	0.533	0.475	0.620	195.3	431.6	124.4	215.7
	1024	0.534	0.475	0.619	195.1	432.7	123.3	215.9
	2048	**0.536**	0.478	0.618	196.3	431.7	124.3	214.7
	4096	0.536	0.478	0.618	196.3	431.6	124.4	214.7
	8192	0.534	0.475	0.618	195.3	432.1	123.9	215.7
	16384	0.534	0.473	0.620	194.3	433.8	122.2	216.7

Table 17. Average Performance Parameters Results: Textbook and Post Paragraph Level Document Experiments—LDA Trained on Two Classes

These results were surprising because they were better than the results obtained from classifiers whose LDA model were constructed based on post level documents. Figures 22, 23, and 24 show the difference in average F-scores obtained from this subset and the previous experiments that used post level documents.

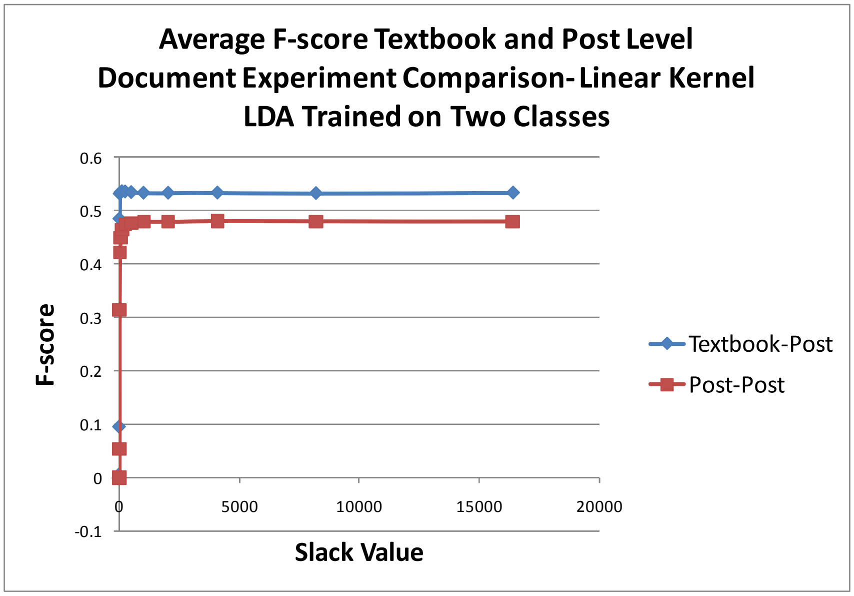


Figure 22. F-score Comparison between Post Level Document Classifiers and Textbook Paragraph Level Documents Classifiers. LDA models trained on Two Classes. Linear Kernel

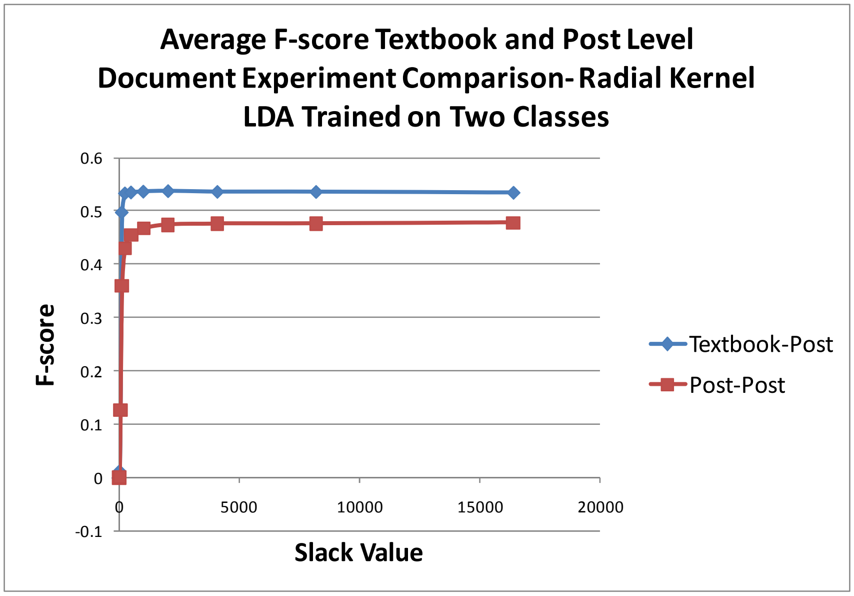


Figure 23. F-score Comparison between Post Level Document Classifiers and Textbook Paragraph Level Documents Classifiers. LDA models trained on Two Classes. Radial Kernel

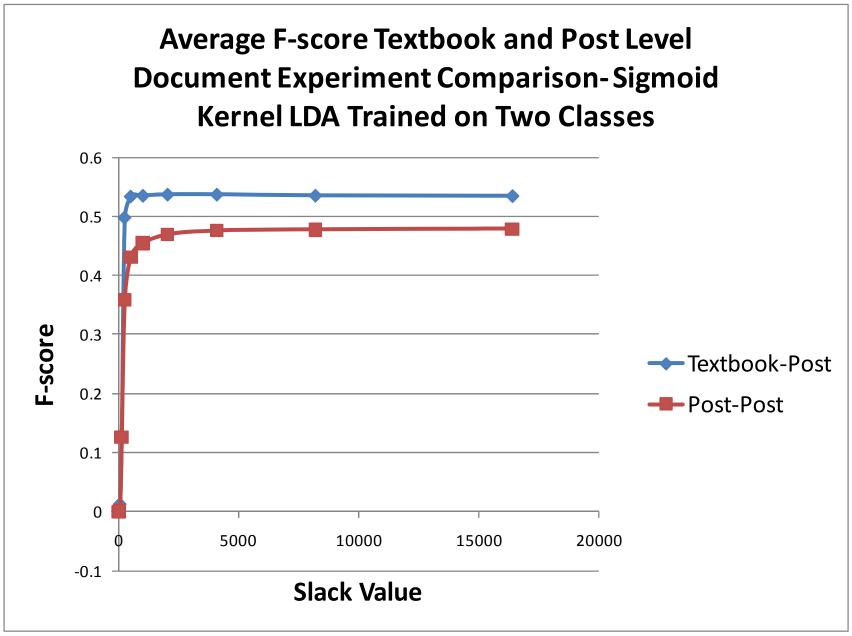
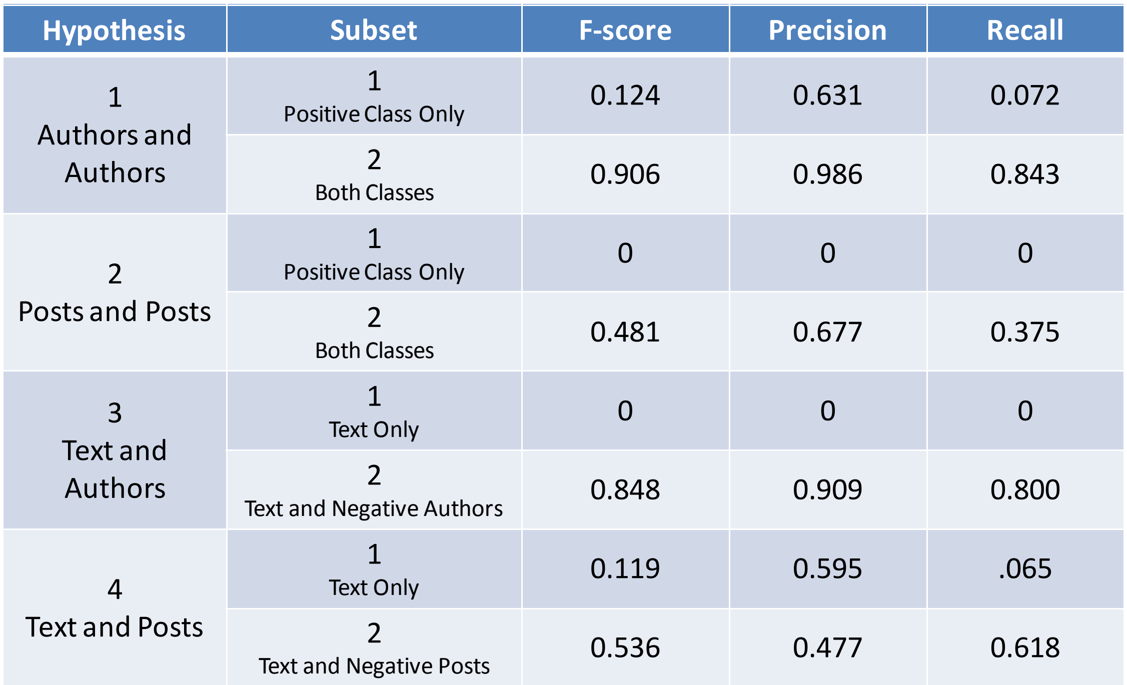


Figure 24. F-score Comparison between Post Level Document Classifiers and Textbook Paragraph Level Documents Classifiers. LDA models trained on Two Classes. Sigmoid Kernel

E. RESULTS SUMMARY

We have constructed four sets of experiments based on three different types of documents: chat posts; the collective posts of authors and; textbook paragraphs. All three were used to construct topic models of physics. Table 18 provides a summary of the maximum classifier results for each of the experiment sets. The author level document classifiers with an LDA model trained on two classes performed the best, followed by textbook paragraph level documents with an LDA model trained on two classes used for classifying authors. Next, the textbook paragraph level documents with an LDA model trained on two classes used to classify posts performed well, followed by post level documents with an LDA model trained on two classes used to classify other posts. Lastly, all the experiments whose LDA model was trained with one class all performed poorly.



Hypothesis	Subset	F-score	Precision	Recall
1 Authors and Authors	1 Positive Class Only	0.124	0.631	0.072
	2 Both Classes	0.906	0.986	0.843
2 Posts and Posts	1 Positive Class Only	0	0	0
	2 Both Classes	0.481	0.677	0.375
3 Text and Authors	1 Text Only	0	0	0
	2 Text and Negative Authors	0.848	0.909	0.800
4 Text and Posts	1 Text Only	0.119	0.595	.065
	2 Text and Negative Posts	0.536	0.477	0.618

Table 18. Maximum Classifier Average Performance for Each Experiment Set

We can derive several generalities about the performance of the classifiers. First, in every experiment set the first subset dramatically under-performed the second subset. LDA models in the first subset were constructed with just the positive class document type. Second, the polynomial kernel was unable to classify a single physics document correctly. Third, linear, and radial kernels generally outperform sigmoid kernels. The highest performing classifiers generally had a linear or radial kernel with slack values 2048 or greater.

Classifiers created with author level documents were better predictors of author documents than post level documents were able to predict other post level documents. Unfortunately, it is difficult to know, whether the higher performance is because authors create better models of physics chat or the better performance is a circumstance of class mixture, testing-training proportion or the difference in source data. For authors, the positive class was taken from a physics chat room from July 2008, and the negative class author documents were taken from an entirely different type of chat room, in the previous year. The data may have been so different that the ability to distinguish them may not have been due characteristics other than topic. Also, the class mixture and testing-training proportions were quite different between those experiments making it difficult to compare author experiments with post level experiments.

The most interesting results were the post level classifiers built with LDA models generated with textbook paragraph documents. The ability to predict post level documents was improved with the use of textbook paragraph LDA model construction versus those made entirely from post level document exemplars. Even more interesting is the fact that the same procedure with author level documents has the opposite effect—predictability decreased in that case. This may lend credence to two ideas. First, that the difference in class sources, mixture, and testing-training proportions affected the predictability of author derived classifiers. Second, it may demonstrate that creating classifiers based on other language modes is a viable way of producing topic classification systems.

Having analyzed the results of our experiments, we now present conclusions and a discussion of future work.

V. CONCLUSIONS AND RECOMMENDATIONS

A. TOPIC NUMBER SELECTION FUTURE WORK

We explored four different types of classification tasks for our research. Each of the experiment sets included the construction of 1,200 SVM classifiers. We construct LDA models based on different document notions from which we derived feature vectors for the SVM classification system. The first set of experiments considered the collective posts of an author as a document. The second set considered an individual post as a document. The third and fourth considered a textbook paragraph as a document.

For the first type of classifier, we compared the collective posts of individual authors in several different socially oriented chat rooms to those in a physics chat room. When we constructed LDA models solely on physics author documents, we achieved best average F-scores of zero. When we trained the LDA model on both physics chat and all-ages-chat, we achieved best average F-scores of 0.906.

For the second type of classifier, we compared the individual physics posts from a physics chat room with non-physics posts from the same chat room. When we trained our LDA model solely on those from the physics chat, we achieved best average F-scores of 0.124. When we trained the LDA model on both physics chat and non-physics chat, we achieved best average F-scores of 0.481.

For the third type of classifier, we compared the topics generated from physics textbook paragraphs to author level documents. We first constructed the LDA models solely on physics textbook paragraphs, resulting in best average F-scores of zero. When we trained the LDA model on both physics textbook paragraphs and all-ages-chat, we achieved best average F-scores of 0.848.

For the fourth type of classifier, we compared the topics generated from physics textbook paragraphs to posts about physics from a physics chat room.

We first constructed LDA models solely on the physics textbook paragraphs, resulting in best average F-scores of 0.144. When we trained the LDA models on both physics textbook and non-physics posts, we achieved best average F-scores of 0.536.

From these results, we conclude the following things. First, when constructing LDA models for classification systems based on sampled distributions, always include documents from both classes. Second, even just a cursory use of SVM classifiers yields reasonably good results when using LDA sampled distributions as feature vectors across multiple kernels. Third, the polynomial classifier in this case performed the worst; however, this may be due to the simplistic settings of its parameters that we selected.

It would seem that detecting physics chat at the author level is reasonably attainable given a large enough portion of known positive chat exemplars. It may be the case that the classifiers are identifying the general “speak” of a chat room rather than topics being discussed. It is clear that correctly classifying physics posts is a truly challenging endeavor, and using a physics textbook to train the LDA model may indeed improve the situation.

Overall, these are good first steps, but leave much room for future work.

B. FUTURE WORK

For each of these experiments there is much space to explore with regard to grooming the data, constructing the LDA models, and configuring SVM classifiers. First, the SVM model should be explored without creating feature vectors based on the LDA model. LDA leveraged experiments should be conducted that predict post level documents after training on author level documents. In addition, research should include testing different stop word lists and text augmentation. Our experiments removed stop words provided by Mallet. There may be improved results were stop words tailored for chat. Techniques employed by Wang [19] and Adams [13] ought to be employed to augment the chat text.

The LDA model should be tested with various values of k, β and α. The SVM classifier should be configured with different parameters for different kernels used. In particular, the polynomial kernel should be tested with various values of γ and r. Additionally, kernels more suitable for probabilities such as those proposed by You et al., which utilize KL and Bhattacharyya distances, should be constructed [31] and tested. In order to test the generalness of this methodology, topics other than physics should be explored. For physics, different textbooks as well as different physics chat rooms should be used as source data.

Finally, we propose using physics texts as augmentation for the LDA model. In our experiments involving the textbook, we use it to replace the positive class. In future experiments, using it in addition to the positive class should be considered.

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF REFERENCES

[1] H. Dong, S. C. Hui, and Y. He, "Structural analysis of chat messages for topic detection," *Online Information Review,* vol. 30, pp. 496–516, 2006.

[2] D. Jurafsky and J. H. Martin, Speech and Language Processing : An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition. London: Prentice Hall, Pearson Education International, 2009.

[3] S. C. Herring, "Computer-Mediated Communication Linguistic, Social, and Cross-Cultural Perspectives," 1996.

[4] E. N. Forsyth, "Improving automated lexical and discourse analysis of online chat dialog," M.S. thesis, Naval Postgraduate School, Monterey, CA, 2007.

[5] M. Zitzen and D. Stein, "Chat and conversation: a case of transmedial stability?" *Linguistics,* vol. 42, pp. 983–1021, 2004.

[6] D. M. Blei, A. Y. Ng, M. I. Jordan, and J. Lafferty, "Latent dirichlet allocation," *Journal of Machine Learning Research,* vol. 3, 2003.

[7] M. R. Freiermuth, "Features of electronic synchronous communication: a comparative analysis of online chat, spoken and written texts," 2002.

[8] C. D. Manning and H. Schütze, *Foundations of Statistical Natural Language Processing*, 6th printing ed. Cambridge, Mass.: MIT Press, 2003.

[9] J. G. Shanahan and N. Roma, "Improving SVM text classification performance through threshold adjustment," in *Machine Learning: ECML 2003*, vol. 2837/2003, Berlin / Heidelberg: Springer Berlin / Heidelberg, 2003.

[10] W. Cohen, V. R. Carvalho and T. M. Mitchell, "Learning to Classify Email into Speech Acts," *EMNLP,* 2004.

[11] J. Lin, "Automatic author profiling of online chat logs," M.S. thesis, Naval Postgraduate School, Monterey, CA, 2007.

[12] J. Tam, "Detecting age in online chat," M.S. thesis, Naval Postgraduate School, Monterey, CA, 2009.

[13] P. H. Adams, "Conversation thread extraction and topic detection in text-based chat," M.S. thesis, Naval Postgraduate School, Monterey, CA, 2008.

[14] T. Dikeos, (July 23, 2009). "Teen's death highlights cyber bullying trend," [Online]. Available: http://www.abc.net.au/news/stories/2009/07/23/2633775.htm (accessed August 20, 2009).

[15] R. Esposito, (May 20, 2009)."Alleged Craigslist Prostitution Ring Busted in New York; Room Service Entertainment Latest Erotic Service Controversy for Classified Ad Site," [Online]. Available: http://a.abcnews.com/Blotter/Story?id=7634021&page=1 (accessed August 20, 2009).

[16] A. Elliott, (July 11, 2009). "A Call to Jihad, Answered in America," [Online]. Available: http://www.nytimes.com/2009/07/12/us/12somalis.html?pagewanted=1&_r=1 (accessed August 20, 2009).

[17] E. Silvestrini, (July 21, 2009). "79-year-old gets 30-year sentence on sex charges," [Online]. Available: http://www2.tbo.com/content/2009/jul/21/79-year-old-gets-30-year-sentence-sex-charges (accessed August 20, 2009).

[18] M. Elsner and E. Charniak, "You talking to me? A corpus and algorithm for conversation disentanglement," in *Proceedings of ACL-08: HLT,* 2008, pp. 834–842.

[19] L. Wang and D. W. Oard, "Context-based message expansion for disentanglement of interleaved text conversations," in Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics, 2009, pp. 200–208.

[20] B. Hjorland, "Towards a theory of aboutness, subject, topicality, theme, domain, field, content ... and relevance," *Journal of the American Society for Information Science and Technology,* vol. 52, pp. 774–778, 2001.

[21] P. D. Bruza, D. W. Song, and K. F. Wong, "Aboutness from a commonsense perspective," *Journal of the American Society for Information Science and Technology,* vol. 51, pp. 1090–1105, 2000.

[22] S. Deerwester, S. T. Dumais, G. W. Furnas, T. K. Landauer, and R. Harshman, "Indexing by Latent Semantic Analysis," *Journal of the American Society for Information Science,* vol. 41, pp. 391–407, 1990.

[23] T. Hofmann, "Probabilistic Latent Semantic Indexing," in SIGIR '99: Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, 1999, pp. 50–57.

[24] T. Griffiths and M. Steyvers, Probabilistic topic models. In T. Landauer, D. McNamara, S. Dennis, and W. Kintsch, editors, Latent Semantic Analysis: A Road to Meaning. 2006.

[25] T. L. Griffiths and M. Steyvers, "Finding scientific topics." *Proc. Natl. Acad. Sci. U. S. A.,* vol. 101, Suppl 1, pp. 5228–5235, April 2004.

[26] J. P. Lewis, (2004), A short SVM (support vector machine) tutorial. [Online]. Available: http://scribblethink.org/Work/Notes/svmtutorial.pdf (accessed September 14, 2009).

[27] R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern Classification (*2nd Edition), Wiley-Interscience, 2000,

[28] Wikipedia, (July 28, 2009). Support vector machine. [Online]. Available: http://en.wikipedia.org/wiki/Support_vector_machine (accessed September 15, 2009).

[29] N. Cristianini and J. Shawe-Taylor, *An Introduction to Support Vector Machines: And Other Kernel-Based Learning Methods.* New York, NY: Cambridge University Press, 2000.

[30] C. Hsu, C. Chang, and C. Lin. (2 October 2008), A practical guide to support vector classification. [Online]. Available: http://www.csie.ntu.edu.tw/~cjlin/papers/guide/guide.pdf (accessed September 15, 2009)

[31] Chang Huai You, Kong Aik Lee, and Haizhou Li, "An SVM kernel with GMM-supervector based on the Bhattacharyya distance for speaker recognition," *Signal Processing Letters, IEEE,* vol. 16, pp. 49–52, 2009.

[32] “Pidgin, the universal chat client,” 2009. [Online]. Available: http://www.pidgin.im (accessed September 22, 2009).

[33] B. Crowell, “Newtonian physics, an online physics textbook,” 2007. [Online]. Available: http://www.lightandmatter.com/area1book1.html (accessed September 22, 2009).

[34] A. K. McCallum, "MALLET: A Machine Learning for Language Toolkit," 2002.

[35] R. Fan, K. Chang, C. Hsieh, X. Wang, and C. Lin, "LIBLINEAR: A library for large linear classification," *Journal of Machine Learning Research,* vol. 9, pp. 1871–1874, 2008.

[36] S. Parsons, "Introduction to Machine Learning by Ethem Alpaydin, MIT Press, 0-262-01211-1," vol. 20, pp. 432–433, 2005.

[37] E. Alpaydin, *Introduction to Machine Learning.* Cambridge, Mass.: MIT Press, 2004.

INITIAL DISTRIBUTION LIST

1. Defense Technical Information Center
   Ft. Belvoir, Virginia

2. Dudley Knox Library
   Naval Postgraduate School
   Monterey, California