DISSERTATION

OPTIMIZING TEXT ANALYTICS AND DOCUMENT AUTOMATION

WITH META-ALGORITHMIC SYSTEMS ENGINEERING

Submitted by

Arturo N. Villanueva, Jr.

Department of Systems Engineering

In partial fulfillment of the requirements

For the Degree of Doctor of Engineering

Colorado State University

Fort Collins, Colorado

Summer 2023

Doctoral Committee:

Advisor: Steven J. Simske

Rick D. Hefner
Nikhil Krishnaswamy
Erika Miller
Nicholas Roberts

ABSTRACT

OPTIMIZING TEXT ANALYTICS AND DOCUMENT AUTOMATION WITH META-ALGORITHMIC SYSTEMS ENGINEERING

Natural language processing (NLP) has seen significant advances in recent years, but challenges remain in making algorithms both efficient and accurate. In this study, we examine three key areas of NLP and explore the potential of meta-algorithmics and functional analysis for improving analytic and machine learning performance and conclude with expansions for future research.

The first area focuses on text classification for requirements engineering, where stakeholder requirements must be classified into appropriate categories for further processing. We investigate multiple combinations of algorithms and meta-algorithms to optimize the classification process, confirming the optimality of Naïve Bayes and highlighting a certain sensitivity to the Global Vectors (GloVe) word embeddings algorithm.

The second area of focus is extractive summarization, which offers advantages to abstractive summarization due to its lossless nature. We propose a second-order meta-algorithm that uses existing algorithms and selects appropriate combinations to generate more effective summaries than any individual algorithm.

The third area covers document ordering, where we propose techniques for generating an optimal reading order for use in learning, training, and content sequencing. We propose two main methods: one using document similarities and the other using entropy against topics generated through Latent Dirichlet Allocation (LDA).

ACKNOWLEDGMENTS

My sincere gratitude goes to my doctoral committee, especially to my advisor Professor Steve Simske, for his unwavering support and guidance throughout my doctoral journey. I am also grateful to Dr. Rick Hefner of Caltech, my industry sponsor, for his support, especially during the flux in my employment.

Colorado State University, the Systems Engineering Department, Ingrid Bridge, and Dr. Debra Dandaneau were incredibly helpful in my journey. Of course, I would not have made it this far without my undergraduate education at the Department of Mathematics at UCLA, the Master of Advanced Studies program at UCSD, and Philippine Science High School.

Dr. Volodymyr Verba deserves recognition for his invaluable assistance in the tedious task of extracting relevant information from PDF files for my document ordering efforts. I would like to acknowledge him and the important services he provided as he and his countrymen and countrywomen endure Russia's war in Ukraine.

I am grateful to Mr. Daniel Carroll, my manager, and Dell Technologies, my employer, for hiring me and giving me an opportunity to do what I love that dovetails nicely with my research.

Of course, I express my immense gratitude towards my parents, Edna and Arturo Sr., for their sacrifices and the unending support I have been so lucky to receive, both directly and indirectly, throughout my life. I thank my brother, Vernon, for being there in my growth.

I want to thank my ever supportive and lovely wife, Mary, who has been patient with me being locked up in my home office, taking classes, and working on my dissertation nights and weekends. The understanding that she has provided through these years has been exceptional and has not been lost on me.

Finally, I extend my heartfelt gratitude to my beloved non-human companions — Linus, Luna, Taz, and Fluffy, who have all passed on — and Kiwi, who is right now being a princess sitting on my lap. They have brought immense joy and comfort to my life, and for that, I am truly thankful.

Now enough with my Oscar speech… let’s get down to business…

PREFACE

On Systems vs. Components

An interesting thing happened on the way to completing my research. OpenAI released ChatGPT, spurring a revolution and arms race in generative natural language processing. Had I not chosen my dissertation topic carefully, my years of work would have likely been rendered obsolete even before my dissertation could be published. But I did choose my topic carefully, and rather than potentially having an irrelevant research topic, my work could only benefit from ChatGPT and these similar tools.

In selecting the topic for my dissertation, I prioritized timelessness as a crucial factor. Therefore, I consciously avoided incorporating the development of "state-of-the-art" algorithms. With my extensive experience in the software industry, I knew too well that dedicating my research to cutting-edge algorithms would carry a significant risk. It was highly probable that while I worked on my dissertation, a well-funded team of researchers working for a company would develop a groundbreaking algorithm, rendering my work obsolete and irrelevant even before completion.

Following the principles of systems engineering, I chose to focus on algorithms of algorithms. This approach involved integrating pre-existing components, much like building a physical system, allowing me to select from a variety of options as a system developer. By designing interfaces that enabled the swapping of components, the architecture of the system could remain intact while evolving to enhance its performance. This approach aligned with the concept of utilizing commercial off-the-shelf (COTS) components, wherein the combination of individual pieces produced a complex system with emergent properties that depended on how the components were integrated rather than being supplanted by a single component.

Undoubtedly, the application of “systems thinking” [1] has been instrumental in the success of my research.

On Generalization vs. Specialization

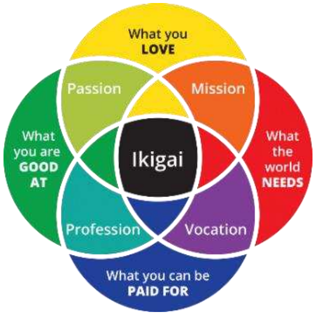
Another decision that has served me well was to pursue this doctorate, even this late in my career. After all, I have always prided myself on being a lifetime generalist. My undergraduate degree in applied mathematics was carefully chosen because I wanted to have a solid basis upon which I could learn almost anything. Even with my earning my master's degree focusing on an engineering area, I chose to specialize in systems engineering. I have, one could say, specialized in generality.

But my pursuit of a doctorate has now taken me somewhat orthogonally as I specialize in meta-algorithmics applied to natural language processing — somewhat — but not quite, as the knowledge I have gained hereon can once again be applied to many areas. Yet, I grin as I complete this dissertation, reminded of the words of Denis Diderot, the 18th-century French philosopher and writer, who famously said, 'I shall very soon know everything about nothing' (‘Je saurai bientôt tout sur rien’ in French).

In the end, perhaps more than ever, I am made keenly aware of how much there is left to discover, understand, and build in uncountably many lifetimes. I remain humbled by the vastness of the unknown and its endless possibilities.

DEDICATION

Dear Zachary,



As your dad, it brings me great joy to dedicate one of the toughest things I've accomplished, to you. With this, I give you some foundational LEGO bricks for success in life — the power of goals, grit, and growth.

- **Goals:** Try many things to find your purpose to give you motivation to achieve your goals. Clarify your goals. Make them specific, measurable, achievable, relevant, and time-bound (SMART) and devise plans for how to reach them. Whether it's financial stability or a healthy life – goals are your North Star. Determine your ikigai and run towards it.
- **Grit:** Perseverance and resilience in the face of obstacles and setbacks are essential. Keep working towards your goals, even against difficulties and discomfort. Always be the Positive Potato. With practice and hard work, your commitment to achieving your goals will serve you well in all aspects of life. Control what you can, and don't worry about those you can't.
- **Growth:** The desire to continuously seek out new knowledge and skills is key. Knowledge is power, so learn like there is no tomorrow. Learn from the experts. Be open to feedback and constructive criticism and be willing to improve with these. Become an expert yourself. Take risks. Embrace your failures as they are lessons for future success.

It is an honor to be a part of your life, and I look forward to continuing to guide you in your journey. As your dad, I will always be here to support you in your endeavors. I am proud of you, and I have no doubt that you will achieve great things in life for yourself and in service to others and the world.

With love, unconditionally,

Dad

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

1 Introduction

1.1 Motivation

Contemporary applications of analytics and machine learning mostly focus on singular algorithms and models. In fewer cases, ensemble algorithms (also called hybrid or combinatorial algorithms) may be used to combine the predictions of multiple individual models to produce more accurate and robust predictions. These ensemble algorithms are typically of the bagging and boosting variety. While useful, they are but a small subset of possible methods of integrating models. Bagging, also known as bootstrap aggregating, is a technique in which several individual models are trained on various subsets of the training data, and their predictions combined to generate a final prediction. In contrast, boosting involves training a sequence of individual models, where each subsequent model is trained to correct the errors of the previous model. Meta-algorithmics, on the other hand, expand on these combinatorial concepts for even more sophisticated and complex constructions.

In his book, *Meta-algorithmics: Patterns for Robust, Low-Cost, High-Quality Systems*, Dr. Steven Simske [2] presents in detail a cornucopia of these tried and tested patterns that go beyond bagging and boosting. It is by way of this publication and Dr. Simske's Colorado State University Systems Engineering course titled "Analytics in Systems Engineering" (SYSE 571) that this dissertation started to take shape.

Equipped with this framework, our objective was to explore practical implementations in natural language processing. For a first effort, we began with a fundamental application in classification, aimed to contribute to alleviating the monotony of requirements engineering by utilizing some of the foundational meta-algorithmic techniques. Our subsequent focus resulted from recognizing the insufficiencies of existing models for extractive summarization, creating an opportunity for the application of meta-algorithms. Lastly, we opted to address a relatively overlooked aspect of natural language processing (NLP): document ordering. The methods described herein could be used in a variety of applications, such as reading comprehension and syllabus creation.

1.2 Background

To fully appreciate this research, three basic ideas must first be clearly defined: systems engineering, meta-algorithms, and functional methods.

1.2.1 Systems Engineering and Natural Language Processing

Systems engineering (SE) is an interdisciplinary field for enabling complex systems. SE can be broken down into a series of steps that includes requirements engineering, design, development, integration, verification and validation, deployment, operation and maintenance, retirement, and the management of this lifecycle. While the term 'systems engineering' conjures up the combined fields of traditional (sometimes called classical) engineering such as mechanical, electrical, software, industrial engineering, and the management of their integration into a physical system, it does not require all these components to be present. The IEEE definition is clear in its generality: "Systems engineering is an interdisciplinary approach and means to enable the realization of successful systems." [3]

It is critical, however, that systems engineering make use of structured and iterative processes to develop and optimize the systems in context, from initial concept to final implementation and maintenance. It is this interpretation that we base our research. And it is with this definition that we put forth our system in context: an integrative combination of algorithms used in natural language processing that result in more complex, often emergent behaviors. This is a result of our desire to take advantage of the component pieces (in this case, algorithms) arranged in a way to gain performance and reliability advantages over these components by themselves. Meta-algorithms and functional methods are thus systems engineering approaches to enabling machine learning or analytic system.

1.2.2 Meta-algorithms and Functional Methods

A meta-algorithm is a high-level algorithm that operates on other algorithms, designed to improve or optimize the performance of these existing algorithms. The simplest of these meta-algorithms are random forests and boosting, used to improve the accuracy of decision trees in machine learning.

Similarly, functional methods are a paradigm that relies heavily on the use of higher-order functions. Often described in the context of certain languages such as Haskell and Scala, functional methods in our context take various analytics methods and combine them into higher level analytics to achieve novel tasks.

1.3 Organization of this Dissertation

This dissertation is organized in the chronological order that individual research was performed.

We start at Chapter 2 with one of the most common applications of supervised learning: text classification in NLP. And what would be a more appropriate application of systems engineering techniques than using them on a systems engineering problem? This chapter focuses on a sub-area called requirements engineering (RE). RE begins with discovery at the outset of project acquisition [1]. Documents typically used during this phase include statements of work (SOWs) and requests for proposals (RFPs) [2]. One of the first challenges of a systems engineer is to carefully classify requirements into appropriate bins for further processing. This manual process, fundamental to understanding stakeholder needs and architecting and designing the system(s) of interest, is often tedious, particularly for large projects that start with thousands of requirements embedded in these documents, making the task ripe for automation. For this research, we investigate multiple combinations of algorithms and meta-algorithms to glean insight into how well they perform on one of the more mundane aspects of requirements engineering. By running various training corpora representing multiple industries through our pipelines of (meta-)algorithms, we obtain some understanding of what works best and how they could be improved.

In Chapter 3, we increase the complexity of our methods, employing a second-degree meta-algorithmic method for abstractive summarization. While much work on abstractive summarization has been conducted in recent years, extractive summarization's lossless nature continues to provide advantages, preserving the style and often key phrases of the original text as meant by the author. Libraries for extractive summarization abound, with a wide range of efficacy. Some do not perform much better or perform even worse than a random sampling of sentences extracted from the original text. This study proposes an implementation of a second-order meta-algorithm in the form of the Tessellation and Recombination with Expert Decisioner pattern, taking advantage of the plethora of already-existing algorithms and dissociating their individual performance from the implementer's biases. It does this by using all of the results obtained by running all the algorithms and by a chosen metric, the Jaccard similarity coefficient, picking appropriate combinations to arrive at summaries better than those generated by any of the component algorithms themselves.

In Chapter 4, we round up our systems engineering methods, focusing on an area that has not seen a lot of research: document ordering. In this chapter, we propose multiple techniques for automatic document order generation for creating optimal reading order for use in learning, training, and other content-sequencing applications. Such techniques could potentially be used to improve comprehension, identify areas that need expounding, generate curricula, and improve search engine results. We advance two main techniques: The first uses document similarities through various methods. The second uses entropy against the backdrop of topics generated through Latent Dirichlet Allocation (LDA). In addition, we try the same methods on the summarized documents and compare them with the results obtained using the unabridged documents. Testing is done using textbook chapters, courses, journal papers, and articles obtained from Wikipedia.

In each of the above-described research activities, we detail the goals, describe the processes and tasks we undertook, provide detailed analyses, and summarize our results. As each of these areas provides tremendous opportunities for further research, we suggest some, though certainly not comprehensively, avenues for investigation and expansion.

We complete this dissertation with Chapter 5, where we discuss a sampling of areas of application from knowledge we gleaned throughout the three areas: meta-algorithmics for classification, meta-algorithmics for summarization, and functional analytics for document ordering. Finally, as this dissertation is being completed shortly after the release of ChatGPT and similar generative AI tools, we consider and reflect on how this research has not been supplanted pre-publication, but rather how our methods could be used to enhance the results we get from these new, innovative tools.

2 Meta-algorithmics for Text Classification[1]

2.1 Introduction

Requirements engineering (RE), the discipline within systems engineering that deals with developing, analyzing, and managing requirements that define a system at successive levels of abstraction [4], can be laborious. One of the most tedious aspects of this area commences at the onset of the project, or even before, with the contractor's receipt of a request for proposal (RFP) or statement of work (SOW) (Sainani *et al.* [5], for example, report experts classifying a mean of 17 requirements per hour by hand.) While small projects may consist of only a few top-level requirements, large enterprise-scale endeavors such as a public transportation system or a new communications infrastructure for the U.S. Navy's fleet may have thousands. According to [6], requirements development alone constitutes 7.0% of a project's cost for commercial projects and 10% for military software, translating to 22.7 and 17.5 person-months in requirements development, respectively.

One such tedium in RE is the classifying of requirements, which is particularly important in the beginning, for various reasons. Various ways of classifying have been proposed and used for diverse applications:

1. Classification by contract obligations, whether governance or architectural [5]

[1] A. N. Villanueva, Jr. and S. J. Simske, "Algorithmic and meta-algorithmic machine learning natural language processing approaches for stakeholder requirements classification," *International Journal of Computational Systems Engineering,* vol. 7, no. 1, pp. 41-56, 2023.

2. Classification by hierarchy or detail [7]
3. Classification by functionality or non-functionality [8]
4. Classification by types (functional, performance, design constraints, quality attributes) [9]
5. Classification by quality attribute and expertise needed (cybersecurity, reliability, etc.) [10] [11]
6. Classification by importance or urgency [4].

The systems engineering "Vee" model, regarded as the standard process for systems engineering [12], starts on the left side to address requirements decomposition starting from stakeholder needs and requirements. This chapter describes, via machine learning and meta-algorithmic patterns, one of the Vee model's earliest needs for organization by taking as input, customer-provided SOWs written in natural language and classifying requirements as governance or system requirements, similar to #1 above. We, however, did not want to separate contract obligations into governance and architecture because architecture has a narrower meaning [11] in the context of a solution.

Design constraints and functional requirements also need to be considered. As such, we opted to separate statements into the following: 1) systems requirements, those that are levied on the system being developed and delivered, including functional requirements and non-functional requirements (constraints, performance, and quality attributes); and 2) governance requirements, which are requirements that are not system requirements, but those levied on the project team and other support, and includes project delivery, compliance, execution, training, operation, maintenance, and other services. This separation is often necessary so that the project management and support teams can focus on the support activities and the engineering team can focus on the technical areas.

However, such a dichotomy is often not cut and dried, as we will see that some requirements straddle the line. Requirements such as "the contractor shall produce the information model of the system and its components" are indeed levied on the contractor yet are targeted towards the engineering team, and "the contractor shall implement a zero-trust architecture" appears to be levied on the contractor but describes a constraint requirement levied on the system. The training corpora reflect the binary classification bins as two documents are used – a document that is purely programmatic (such as a performance work statement, or PWS) and a document that describes a system [13], such as a specification.

For the purposes of this research, we adhere to standard contract language [14] and universally accepted convention that requirements have the imperative "shall" following the subject [7] [13] [15] [16] [17], loosely of the form "<subject> shall <action verb clause> <object clause> <optional qualifying clause>." Some examples include "the vendor shall provide a monthly-updated integrated master schedule within 30 days of award date," "the system shall be accessible as per the Americans with Disabilities Act (ADA)," and "the contractor shall utilize model-based systems engineering (MBSE) principles." With the convention, we distinguish requirements from needs, which are typically not written in the "shall" structure. Capabilities, operational and mission threads, needs, use cases, and user stories may be used to derive requirements but do not qualify as requirements themselves. A formal conversion to the "shall" structure is necessary not only for consistency with a standard form, but also to vet the stakeholders' wants and expectations [18].

2.2 Related Work

In recent years, some work has been done in classifying requirements using machine learning techniques, though many have focused on software engineering projects. There is notable work on the subject:

Sainani *et al.* [5] started with 20 software contracts, extracted obligations (requirements) from them, and classified those obligations using Naïve Bayes, Random Forest, and Support Vectors Machine (SVMs), as well as using a Bidirectional Long Short-term Memory (BiLSTM) deep learning method and

Google's BERT for comparison. Similarly, Canedo and Mendes [19] studied multiple algorithms (Logistic Regression (LR), SVMs, Multinomial Naïve Bayes (MNB), and k-Nearest Neighbors (kNN)) for their accuracy and precision for classifying. Earlier work by Mahmoud and Williams [20] used word similarity and clustering techniques.

Abad *et al.* [21] looked at software requirements from Software Requirements Specifications (SRSs) and classified them as either functional requirements (FRs) or non-functional requirements (NFRs). They investigated the effect of preprocessing the dataset by applying grammatical, temporal, and sentimental characteristics of sentences using parts of speech (POS) tagging to standardize the dataset requirements for simpler processing. Finally, they classified NFRs into quality attributes using multiple algorithms, with Naïve Bayes taking the trophy.

Sabir *et al.* [22] tackle misclassification of NFRs by assigning multiple tags to requirements with the premise that requirements often straddle a grey area when correctly categorizing them.

Giannakopoulou *et al.* [23] describe FRETISH, and Lucio *et al.* [24] describe EARS, structured natural languages for formally writing requirements, useful for reducing conflicting interpretations and improving analysis.

However, besides the work by Sainani *et al.* [5], not much has been conducted on ingesting raw SOW data and splitting them into a set that needs to be consumed by project management and a set that needs to be consumed by the systems engineering and technical team. The work led by Sainani focused on investigating software (not complex systems) contracts. What they considered architectural contract obligations were somewhat limited to architectural constructs specific to software. Complex systems, on the other hand, are a generalized collection of interconnected and interrelated parts, and add another dimension beyond bits and bytes [25]. Contractual obligations for software projects as these systems typically uniquely include physical quality attributes such as reliability and availability and constraints such as size, weight, and power (SWaP) [26].

2.3 Context and Research Goals

2.3.1 Context

The general research area is deemed a natural application of machine learning and automation [27], as the vast majority of RFPs and SOWs do not follow a universal structured natural language such as EARS [28] or FRETISH [23], or at best follow a semi-restricted format that begins with either "the system shall" or "the contractor shall." Such restrictions are unlikely to be enforced by every SOW writer. The PROMISE database itself has only 455 of its 604, or roughly 0.75 [29], requirements compliant with the latter restrictions.

We do make at least one assumption: For a statement to be called a requirement, it must contain a "shall" as per what is mentioned in the Introduction above. Other keywords could be used, including "will" or "should," but following standard RE practices [12] [13] [16], we stick with "shall." In this scheme, requirements that have bullets or lists, such as "the system shall comply with (a) Standard A, (b) Standard B, and (c) Standard C," counts as a single requirement. This, of course, violates the rule that requirements need to be singular [16].

2.3.2 Research Goals

Given the above context, we endeavor to discover how some fundamental algorithms (Naïve Bayes, TF*IDF with Cosine Similarity, logistical regression) compare with each other as well as against a simple pattern match and a more complex unsupervised learning algorithm in Stanford's GloVe [30]. The first three will also be subjected to two first-order meta-algorithms [2] in the form of weighted voting and predictive selection.

In addition, using GloVe, we determine the effect of various parameters on classification accuracy, both as a standalone algorithm and as the initial categorizer for predictive selection. In particular, we use the most prevalent words found in the training corpora, starting with the single most common word for the Governance bin and similarly for the System bin, and increasing number of words to the second most common, and so on until 15 of the most common words (Table 4) are represented.

2.4 Process / Tasks

We executed the following process tasks:

2.4.1 Data Collection

The process of collecting appropriate datasets for analysis was long and arduous, mostly for the lack of publicly accessible ground-truthed requirements sets from which to train and test our engines. The vast majority of publicly available SOWs either do not come with ground truth labels, are limited to software, small (under 50 requirements), heavily governance-based statements, or a combination of some or all of these. To get around this, for training purposes, we settled on substituting five reasonably large, expired SOWs and specifications, each containing nothing but governance statements or system statements, specifications, and technical descriptions. Because some corpora, such as specification documents, do not have many "shall" statements but are otherwise relevant to training because of the vocabulary used for those corpora, the number of words (and their frequencies) were deemed more relevant than the actual number of "shall" statements. However, for completeness, these numbers are also included for reference (Table 1).

Table 1. Training Corpora.

Type	**Corpus**	**Industry / Type**	**“Shall” Statement Count**	**Word Count**
Governance	TrainG01	Defense / Enterprise Communications Network	696	163,437
Governance	TrainG09	Construction / Laboratory	199	81,235
System	TrainS02	Agriculture / Conservation Management System	8	55,369
System	TrainS05	Defense / Comms and Data Management System	6,251	668,912
System	TrainS12	Defense / Electronic Warfare System	910	200,622

Table 2. Training/Validation Corpora Combinations

Training ID	Validation ID	Corpora Combination
TR01	V01	TrainG01+TrainS02
TR02	V02	TrainG01+TrainS05
TR03	V03	TrainG01+TrainS12
TR04	V04	TrainG01+TrainS02+TrainS05
TR05	V05	TrainG01+TrainS02+TrainS12
TR06	V06	TrainG01+TrainS05+TrainS12
TR07	V07	TrainG01+TrainS02+TrainS05+TrainS12
TR08	V08	TrainG09+TrainS02
TR09	V09	TrainG09+TrainS05
TR10	V10	TrainG09+TrainS12
TR11	V11	TrainG09+TrainS02+TrainS05
TR12	V12	TrainG09+TrainS02+TrainS12
TR13	V13	TrainG09+TrainS05+TrainS12
TR14	V14	TrainG09+TrainS02+TrainS05+TrainS12
TR15	V15	TrainG01+TrainG09+TrainS02
TR16	V16	TrainG01+TrainG09+TrainS05
TR17	V17	TrainG01+TrainG09+TrainS12
TR18	V18	TrainG01+TrainG09+TrainS02+TrainS05
TR19	V19	TrainG01+TrainG09+TrainS02+TrainS12
TR20	V20	TrainG01+TrainG09+TrainS05+TrainS12
TR21	V21	TrainG01+TrainG09+TrainS02+TrainS05+TrainS12

All 21 combinations with at least one Governance corpus and one System corpus were used for training. For reference, see Table 2 for the various combinations and their IDs.

For the test phases for each algorithm, despite the tedium of ground-truthing, we had little choice but to do so. We collected and manually processed a set of thirteen SOW corpora (Table 3).

Extraction of the “shall” statements was trivial. Classifying them as either governance or system requirements for ground-truthing was time-consuming but provided some insight. Note that some engineering-related requirements such as requiring the use of model-based systems engineering (MBSE) were best classified as governance requirements as these are levied on engineers rather than the system in context. In addition, it was decided that, given a choice between more false positives in the classification of system requirements and more false positives in the classification of governance requirements, the former was more tolerable, as some governance requirements could often be better satisfied by system capabilities. For example, if the SOW required the contractor to provide weekly data dumps from the system, it might be best to implement a feature in the system to automatically send the required data automatically at the required intervals. Or if training was required, a system that was developed to be user-friendly would reduce the amount of manual training involved.

Four critical infrastructure sectors are represented — transportation, defense industrial base, energy, and commercial facilities — each of whose representative SOWs weigh more heavily towards governance-related requirements.

Table 3. Test Corpora

SOW #	Industry / Type	Requirement Counts		
		Governance	System	Total
Test01	Transportation System	23	15	38
Test02	Transportation Services	19	0	19
Test03	Defense / C4ISR Installation Services	381	8	389
Test04	Energy / Solar System	331	25	356
Test05	Defense / Communications System	53	0	53
Test06	Defense / Inventory Mgmt System Prototype	50	11	61
Test07	Defense / Cybersecurity Services	40	2	42
Test08	Civil Engineering / Construction	174	71	245
Test09	Defense / Communications System	141	4	145
Test10	Defense / Application Infrastructure	9	15	24
Test11	Energy / Microgrid	8	55	63
Test12	Defense / Cybersecurity System (Software)	20	314	334
Test13	Defense / Communications System	11	315	326
	TOTAL	1,260	835	2,095

2.4.2 Preparation and Preprocessing

Preprocessing the training datasets involved the following:

- A typical first step, every word in the corpora was converted to lowercase.
- A total of 337 stop words were removed, starting with scikit-learn's [31] English stop words combined with the words *annex*, *appendix*, *diagram*, *example*, *fig*, *figure*, *handbook*, *may*, *mil*, *must*, *page*, *requirement*, *shall*, *table*, *unless*, *use*, *used*, *will*, *within*, and *would*. These additional common words, a few corpus-specific words (*steward*, *watershed*), and the name of the project proved to be prevalent and added no value to the corpora. We used scikit-learn's set of stop words as our basis as it was one of the most extensive sets to remove the 318 most common words in the English language.
- Punctuation marks were removed.

- Next, words were combined to prepare for counting by passing all of them through NLTK's [32] lemmatization function three times; first, treating everything as nouns, followed by verbs, then adjectives.
- The final step involved removing all words that were not in NLTK's set of 235,892 English words.

2.4.3 Vectorization

The next step was vectorization. Using scikit-learn's CountVectorizer with a maximum of 500 features and a minimum of one mention, we created vectors for each document or combined set of documents representing a classification (Governance or System). Table 4 shows the probability distribution for the first 15 most common words associated with each of the classification corpora and Figure 1 and Figure 2 show the complete results of the normalized vectors following Zipf's Law. The resulting vectors were then used for both Naïve Bayes and TF*IDF/Cosine Similarity training. Note that training (and later validation for GloVe) used the documents or combination of documents as a comprehensive "bag of words" whole to generate the vectors since some documents, such as specification documents, do not have "shall" statements but still contain valuable information on the types of words that is useful for classifying requirements statements.

Table 4. Top 15 Word Probabilities for Training Corpora

Governance Training Corpora				**System Training Corpora**					
TrainG01.pdf		**TrainG09.pdf**		**TrainS02.pdf**		**TrainS05.pdf**		**TrainS12.pdf**	
contractor	0.0593	contract	0.0667	data	0.1129	display	0.0352	provide	0.0367
test	0.0237	construction	0.0286	user	0.0364	section	0.0256	data	0.0277
provide	0.0221	contractor	0.0239	search	0.0281	provide	0.0229	operator	0.0257
support	0.0206	service	0.0217	site	0.0226	function	0.0199	capability	0.0245
plan	0.0205	officer	0.0207	component	0.0185	trim	0.0170	track	0.0188
government	0.0185	review	0.0192	specification	0.0169	control	0.0152	display	0.0185
report	0.0174	draw	0.0187	access	0.0159	drain	0.0148	control	0.0152
technical	0.0139	require	0.0176	design	0.0156	accordance	0.0125	channel	0.0151
train	0.0136	project	0.0172	model	0.0143	data	0.0123	interface	0.0127
management	0.0134	government	0.0159	time	0.0140	alarm	0.0116	distribution	0.0116
include	0.0123	clause	0.0153	provide	0.0124	indication	0.0108	support	0.0108
design	0.0122	work	0.0151	interface	0.0124	mode	0.0108	increment	0.0106
follow	0.0107	business	0.0148	support	0.0121	test	0.0102	print	0.0099
engineer	0.0106	design	0.0142	server	0.0112	operator	0.0099	revision	0.0097
service	0.0106	document	0.0140	management	0.0108	refer	0.0091	equipment	0.0095

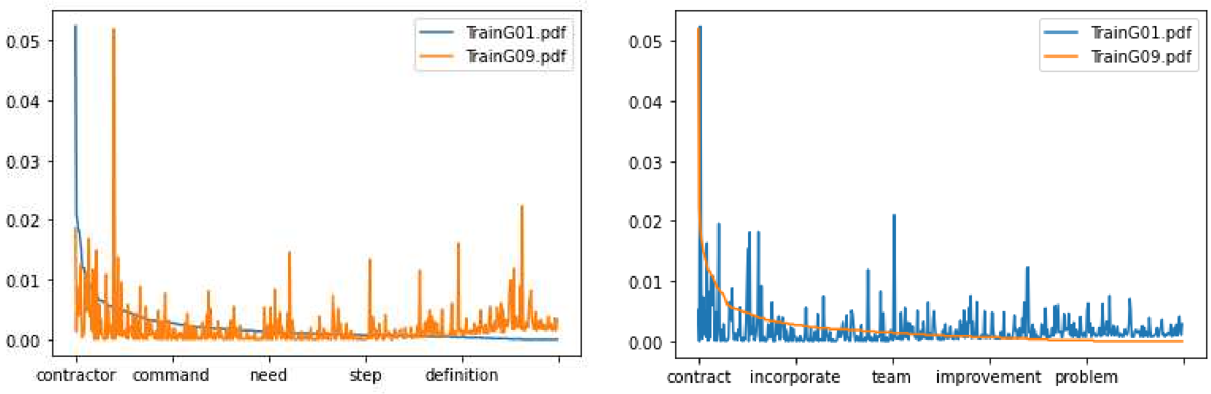


Figure 1. Probability Distribution Showing Zipf's Law Normalized for the Two Training Corpora Representing Governance

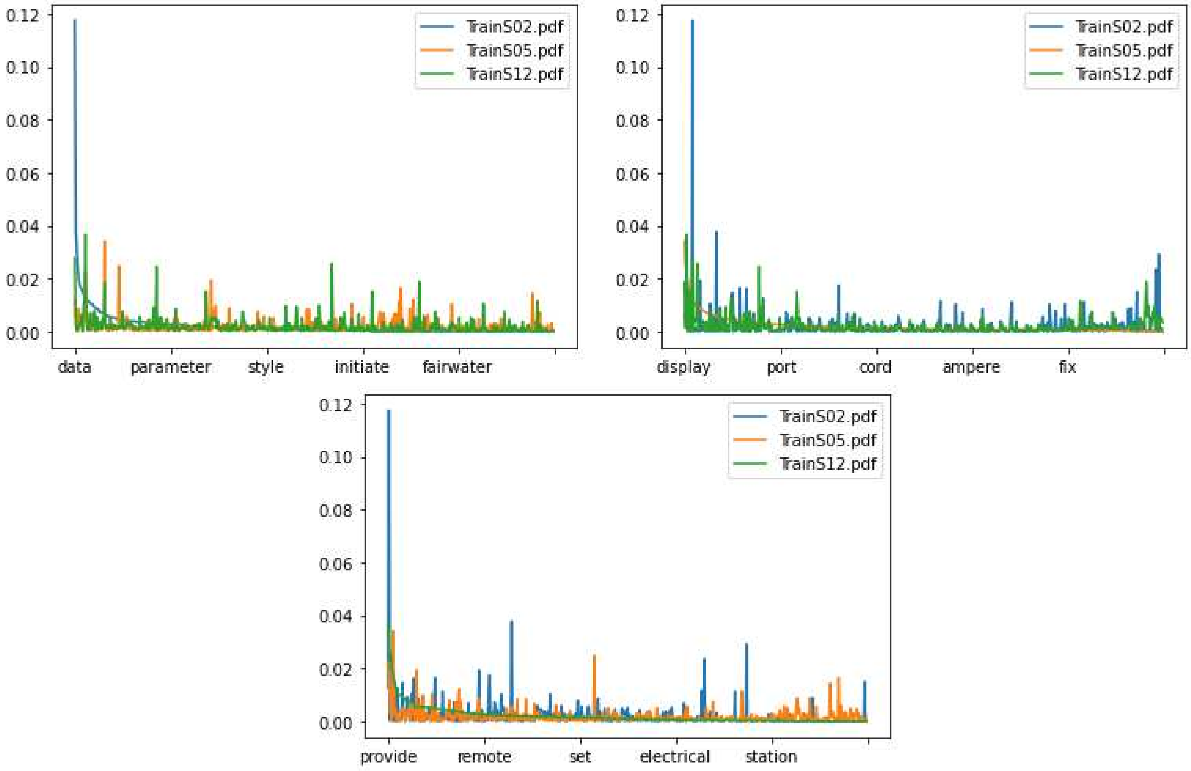


Figure 2. Probability Distribution Showing Zipf's Law Normalized for the Three Training Corpora Representing System

2.5 Analysis and Results

The comprehensive set of results, using TR21/V21, are depicted in Figure 3 and Figure 4. The other training/validation corporate resulted in similar outcomes. What is interesting to note here is the enormous variation among the GloVe variants.

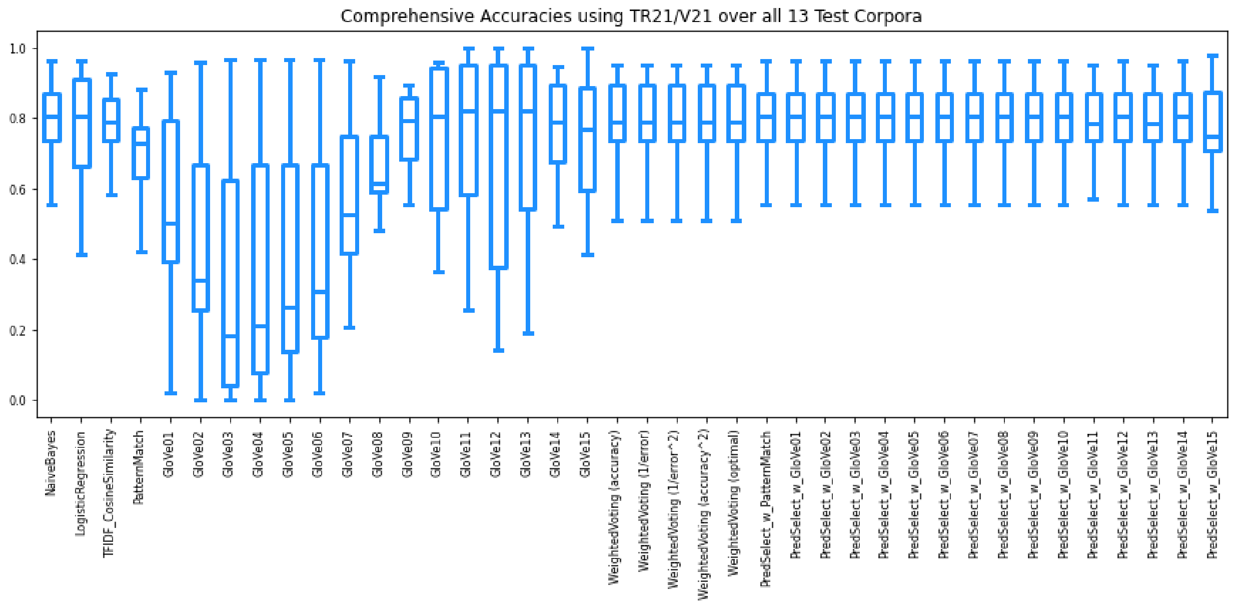


Figure 3. Accuracy Distribution of Classification using all 40 Algorithms and Meta-algorithms over 13 Test Corpora for the TR21 Training Corpora / V21 Validation Corpora

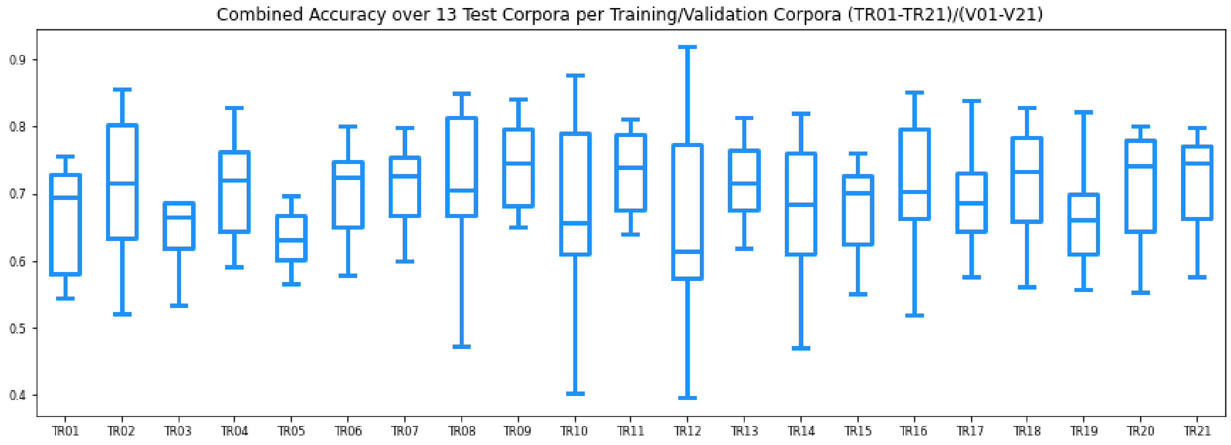


Figure 4. Accuracy Distribution of Classification using all 40 Algorithms and Meta-algorithms over 13 Test Corpora for each Training Corpus Combination

We dive deeper into the results in the following subsections:

2.5.1 Traditional Algorithms

2.5.1.1 Simple Pattern Match

To have a good idea of how well our chosen machine learning natural language processing (NLP) algorithms work, it is useful to set an extremely simple baseline. For our test dataset, we simply checked to see if certain words existed in a statement. For the Governance classification, we picked the words *contractor, vendor, offeror, provide,* and *support.* Statements that did not have these keywords were classified as System. As expected, performance was poor (training accuracy = 0.687) and can partially be attributed to SOW diversity, such as substituting the actual name of the vendor for the terms *vendor* or *contractor*.

The next step was to compare accuracy with the results taken from three fundamental NLP classification methods:

2.5.1.2 Naïve Bayes

Using the vectorized documents described above, our first attempt using the NLTK implementation appeared to be ultra-sensitive to the small training dataset and produced unusable results. The second attempt involved going back to basics for an implementation [33] that resulted in much more productive classifications. For this run, we obtained a mean accuracy of 0.929 across all the training corpora.

*2.5.1.3 TF*IDF/Cosine Similarity*

Again, using the vectorized documents, we employed term frequency – inverse document frequency (TF*IDF), calculating the weighted Cosine Similarity between the requirements and each of the two classification documents. For this run, we obtained a mean accuracy of 0.903 across all training corpora.

2.5.1.4 Logistic Regression

Finally, of the traditional algorithms, we applied Logistic Regression. For the gradient descent portion of this algorithm, we used $\propto = 10^{-8}$ and limited iterations to 500. For this run, we obtained a mean accuracy of 0.890 across all the training corpora.

All three NLP ML algorithms performed considerably better than the simple pattern match and showed slightly better results for both Naïve Bayes and TF*IDF/Cosine Similarity compared to logistic regression. It exhibited the expected comparative accuracy between the two classifiers that use the same word vectors as a basis.

Figure 5 summarizes the results of these first four algorithms using all training corpora TR01-TR21.

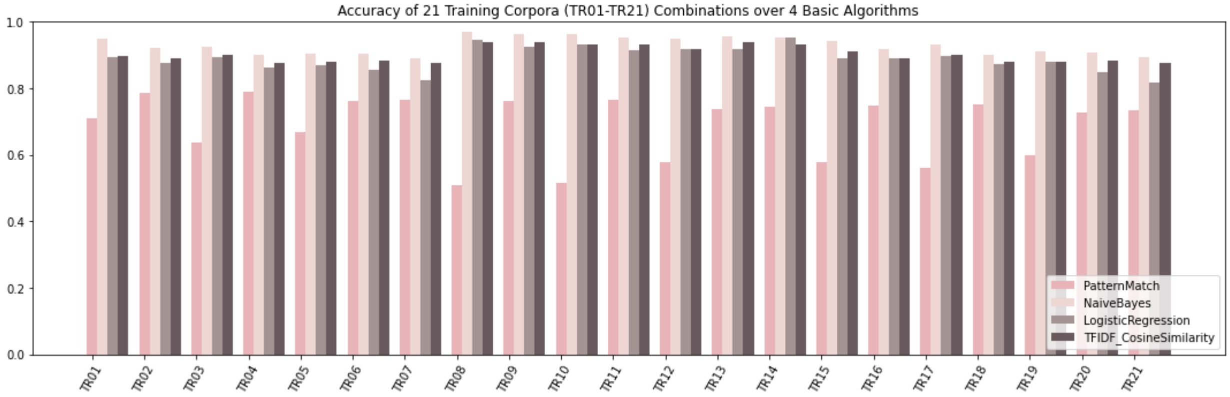


*Figure 5. Comparing Basic Algorithms (Pattern Match, Naïve Bayes, TF*IDF/Cosine Similarity, and Logistic Regression) Accuracies for Training Corpora (TR01-TR21)*

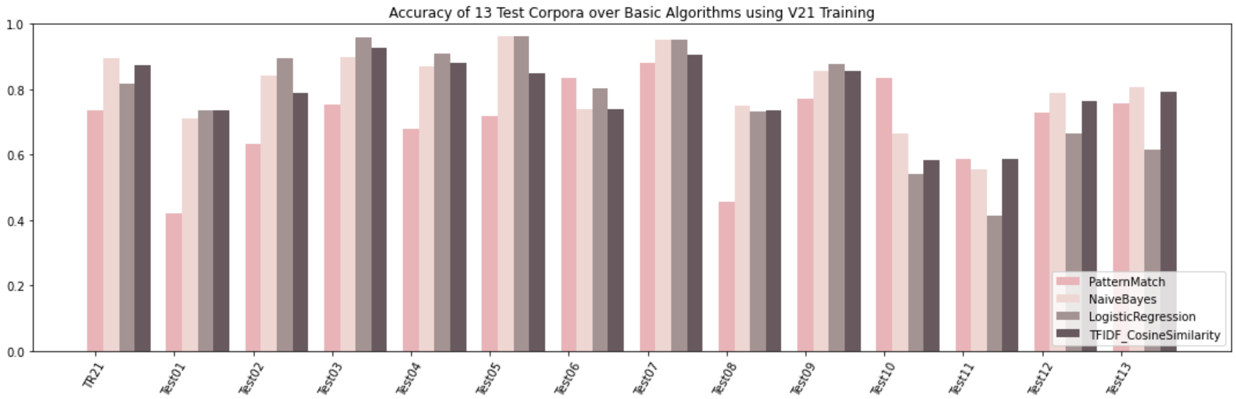


*Figure 6. Comparing basic algorithms (Pattern Match, Naïve Bayes, TF*IDF/Cosine Similarity, and Logistic Regression) accuracies for test corpora (TR01-TR13) using TR21 training corpora*

2.5.1.5 GloVe

Next, we implemented a more sophisticated algorithm using Stanford's GloVe. The algorithm takes advantage of global distances between words using word embeddings or multi-dimensional vector representations. Unlike with CountVectorizer, GloVe word representations may number in the hundreds of dimensions. Our first implementation created a model from the training corpora described above but resulted in a low accuracy of <0.60. With the lack of large requirements-specific ground-truthed documents, we looked to use a 100-dimensional word representation database based on a combination of the entirety of Wikipedia from 2014 and the fifth edition of the English Gigaword [30] [34].

This second run used the words in Table 4 as anchor words to measure proximity to the bins and consisted of 15 sub-runs, each corresponding to the number of words from most common to least common. For example, in the TrainG01 and TrainS02 combination, GloVe1 was fed the words *contractor* and *data* to represent the Governance and System bins, respectively. In the TrainG09 and TrainS05 combination, GloVe3 was fed the words *contract*, *construction*, and *contractor* to represent the Governance bin, and *display*, *section*, and *provide* to represent the System bin. Combined documents used the combined normalized word probabilities of those documents, with Figure 8 detailing the results of the GloVe variations on the different test corpora. Figure 5 includes the best validation values of the GloVe variations in Figure 8, which were trained with the Wikipedia+Gigaword combination.

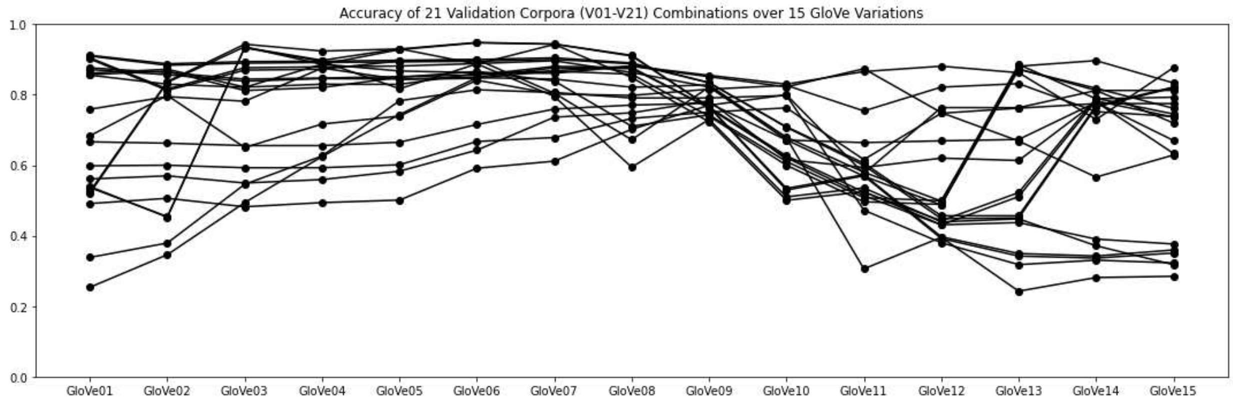


Figure 7. Comparing GloVe Accuracies for Validation Corpora (V01-V21)

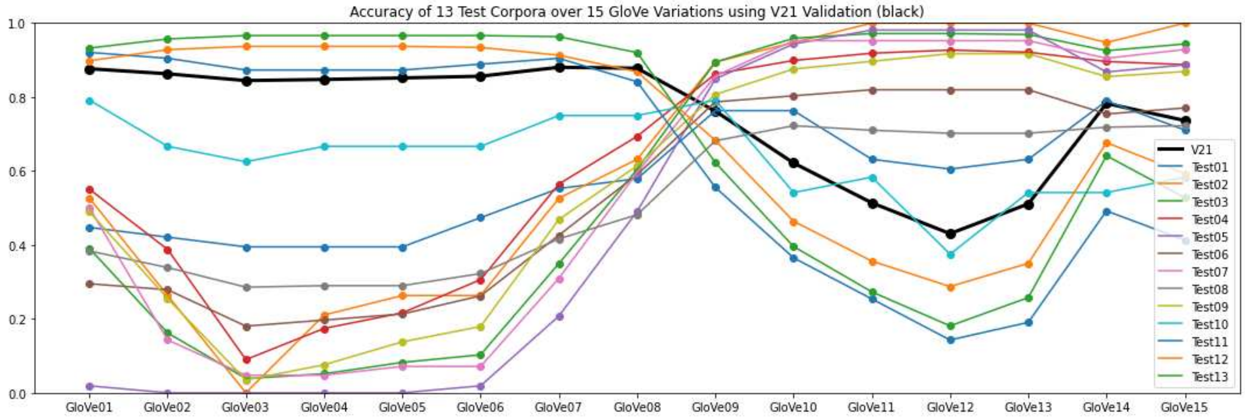


Figure 8. Comparing GloVe Variations on each Test Corpus using V21 Validation Corpora

Figure 7 and Figure 8 suggest GloVe is highly sensitive to the number of words used as anchors and generally peaked in accuracy (for the validation corpora) using the top-ranked 6-8 common words and dropped again as more words are used. Yet, GloVe09 performed the best and gave the most consistent results (i.e., lowest coefficient of variance) with the test corpora. For some test corpora, GloVe performed extremely well, but it could likely be attributed to a bias of the System-oriented nature of those corpora (Table 3). In general, the best performing GloVe variations did not perform any better than the traditional algorithms, and consistency of performance was rather undependable (see Figure 9 and Figure 10).

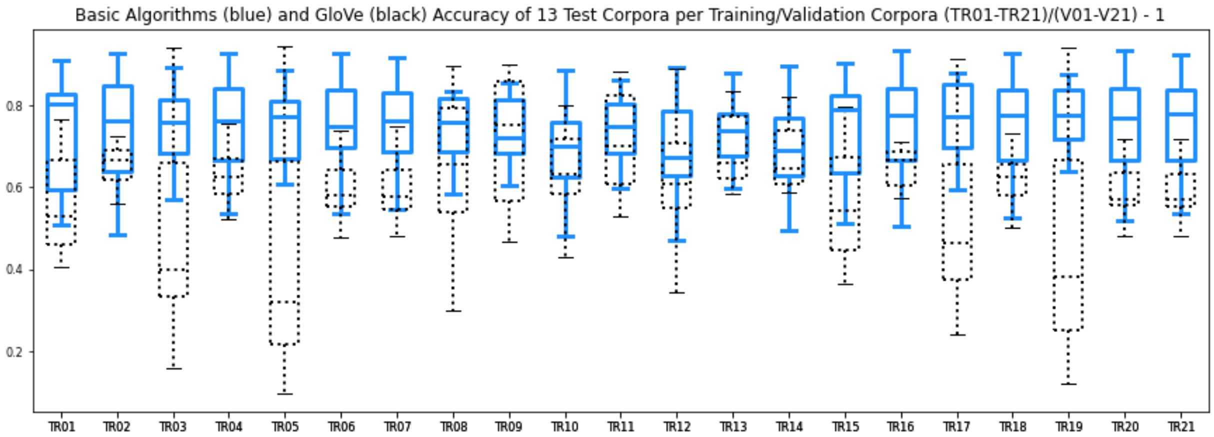


Figure 9. Basic Algorithms (blue) Compared with GloVe (dotted-black) Accuracy across the 13 Test Corpora for each of the TR01-TR21 Training Corpora (V01-V21 for GloVe)

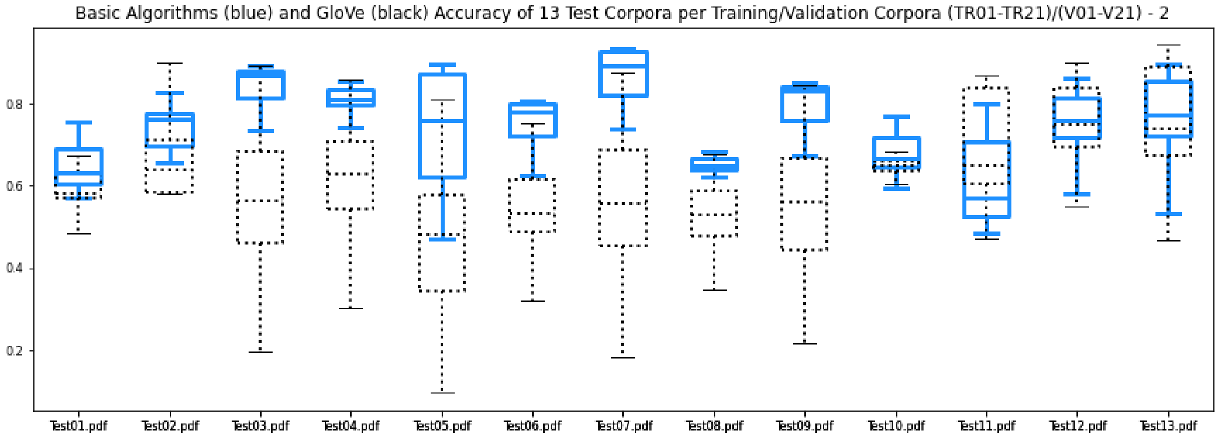


Figure 10. Basic Algorithms (blue) Compared with GloVe (dotted-black) Accuracy across the TR01-TR21 Training Corpora (V01-V21 Validation Corpora for GloVe) for each of the 13 Test Corpora

2.5.2 Meta-algorithmic Approaches

The next step involved applying meta-algorithmic approaches to see if any advantages are obtained through such advanced consensus approaches. A meta-algorithm is a higher-level algorithm that combines more fundamental algorithms to obtain results that are as good or better than the original basis algorithms. For this research, we looked to [2] for descriptions of a library of these meta-algorithmic patterns and picked two first-order meta-algorithms: weighted voting and predictive selection.

2.5.2.1 Weighted Voting

The weighted voting pattern is often an improvement over the simple voting pattern [2]. While the voting pattern weights each component algorithm equally, the weighted voting pattern assigns proportional weights to each component algorithm based on their performances in the training corpora. Figure 11 depicts the weighted voting meta-algorithm.

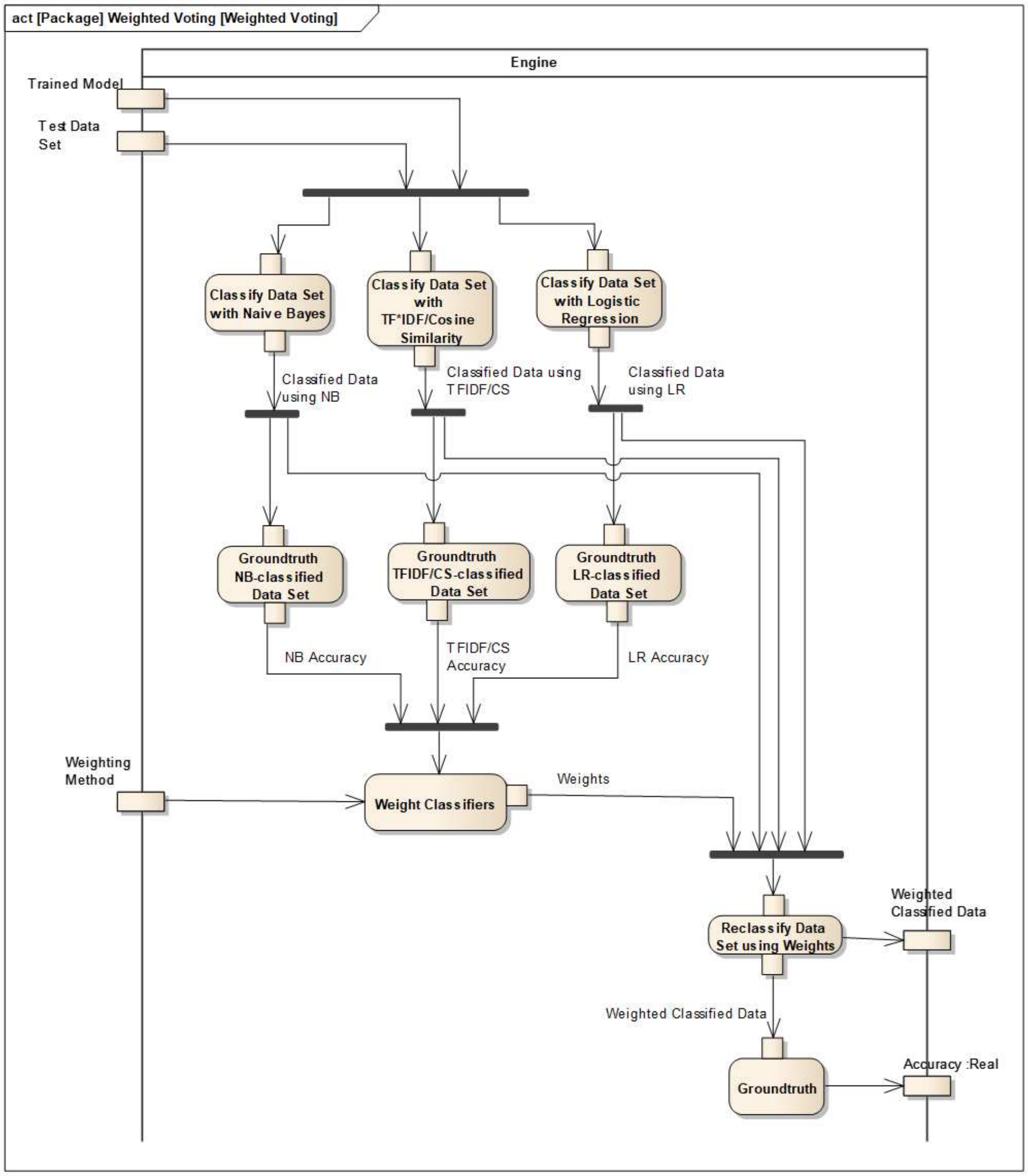

act [Package] Weighted Voting [Weighted Voting]
Engine
Trained Model
Test Data Set
Classify Data Set with Naive Bayes
Classify Data Set with TF*IDF/Cosine Similarity
Classify Data Set with Logistic Regression
Classified Data using NB
Classified Data using TFIDF/CS
Classified Data using LR
Groundtruth NB-classified Data Set
Groundtruth TFIDF/CS-classified Data Set
Groundtruth LR-classified Data Set
NB Accuracy
TFIDF/CS Accuracy
LR Accuracy
Weighting Method
Weight Classifiers
Weights
Reclassify Data Set using Weights
Weighted Classified Data
Weighted Classified Data
Groundtruth
Accuracy :Real

Figure 11. Weighted Voting

Weighting was performed using five methods: $accuracy$, $\frac{1}{error}$, $accuracy^2$, $\frac{1}{\sqrt{error}}$, and an information-theory-based optimal approach [35] to weighting of the form

$$W_j = \ln\left(\frac{1}{N_{classes}}\right) + \ln\left(\frac{p_j}{e_j}\right),$$

Equation 1

where

$$e_j = \frac{1-p_j}{N_{classifiers}-1}.$$

Equation 2

In all cases, we obtained little variation in the results among the five weighting methods. However, they did provide mostly improved results when compared to the component algorithms taken together (i.e., non-weighed voting). In some, weighted voting performed worse than the best of the three component algorithms. This can be attributed to the small differences in performance among Naïve Bayes, TF*IDF/Cosine Similarity, and Logistic Regression by themselves (Figure 5). For the weighted voting algorithm, in many cases, the two relatively inferior component algorithms agree on the classification and, therefore, overpower the superior component algorithm, which was consistently the Naïve Bayes algorithm. Figure 12 and Figure 13 summarize the results comparing the two approaches.

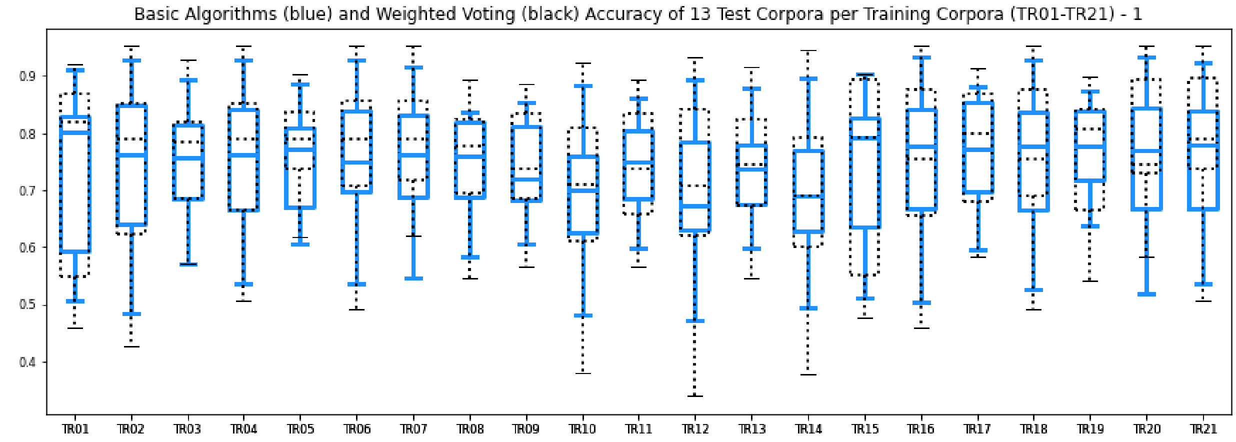


Figure 12. Basic Algorithms (blue) Compared with Weighted Voting (dotted-black) Accuracy across the 13 Test Corpora for each of the TR01-TR21 Training Corpora

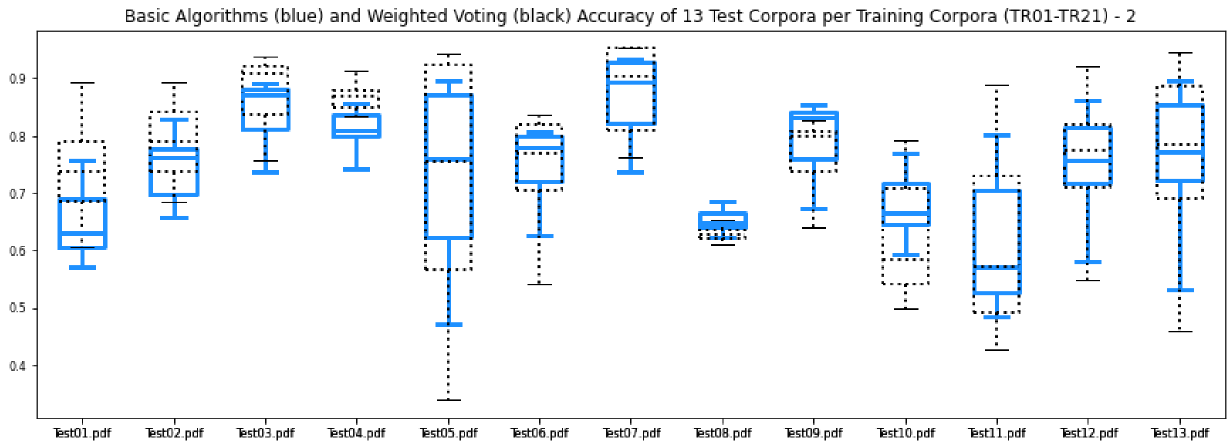


Figure 13. Basic Algorithms (blue) Compared with Weighted Voting (dotted-black) Accuracy across the 21 Training Corpora for each of the 13 Test Corpora

2.5.2.2 Predictive Selection

Predictive selection, like weighted voting, is a first-order meta-algorithm [2]. For this method, a separate preliminary categorizer is introduced to bin the input dataset, and individual component classifiers are chosen to operate on each of those bins. The idea is to select a single component algorithm for each category that provides the best precision for the category. Predictive selection is comprised of two phases: the statistical learning phase (Figure 14) and the run-time phase (Figure 15).

For the first phase, we applied 16 preliminary categorizers. The first involved using a simple pattern match algorithm (described above) on the training corpora, and the rest used the fifteen GloVe variations already calculated (Figure 7). The statistical learning phase for each trial provided us with the data to generate the category-scoring matrices.

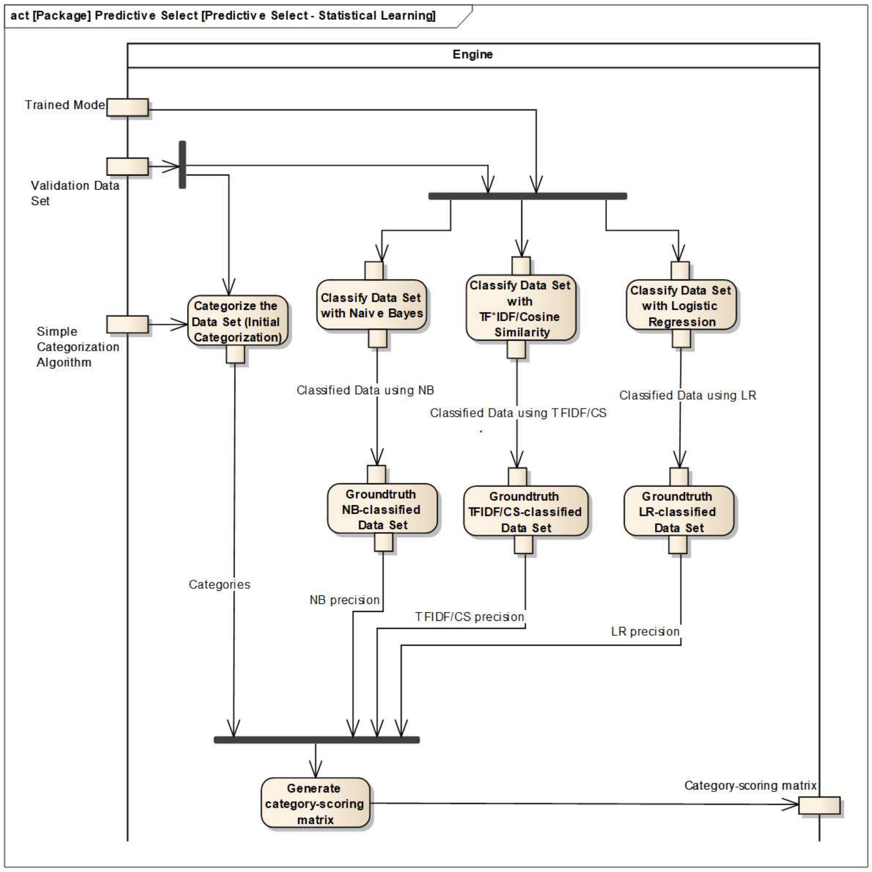

act [Package] Predictive Select [Predictive Select - Statistical Learning]
Engine
Trained Mode
Validation Data Set
Simple Categorization Algorithm
Categorize the Data Set (Initial Categorization)
Classify Data Set with Naive Bayes
Classify Data Set with TF*IDF/Cosine Similarity
Classify Data Set with Logistic Regression
Classified Data using NB
Classified Data using TFIDF/CS
Classified Data using LR
Groundtruth NB-classified Data Set
Groundtruth TFIDF/CS-classified Data Set
Groundtruth LR-classified Data Set
Categories
NB precision
TFIDF/CS precision
LR precision
Generate category-scoring matrix
Category-scoring matrix

Figure 14. Predictive Selection Meta-algorithm: Statistical Learning Phase

The second phase of predictive selection, the run-time phase, uses the learned best algorithms from the training phase (i.e., using the category-scoring matrices). Categorization during the run-time phase is then performed on the test corpora the same way as the training / validation steps during the statistical learning phase. Naïve Bayes was the overwhelming choice for our runs regardless of the initial categorization, followed by Logistic Regression.

The results of both phases are depicted in Figure 16 and Figure 17 and show tremendous variation across test corpora. However, predictive selection generally performed better than weighted voting (and non-weighted voting). In some unusual cases, though we saw a slight degradation in performance, we generally observed consistent results regardless of the preliminary categorizer used. For example, using the V21 validation corpora, which shows a dramatic dip beyond using eight anchor points, Figure 18 shows consistently flat accuracy results given any test corpus (Test01-Test13).

This predictive selection performance can be explained by the similar performances exhibited by the three traditional component algorithms from which the predictive selection meta-algorithm chooses: Naïve Bayes, TF*IDF/Cosine Similarity, and Logistic Regression.

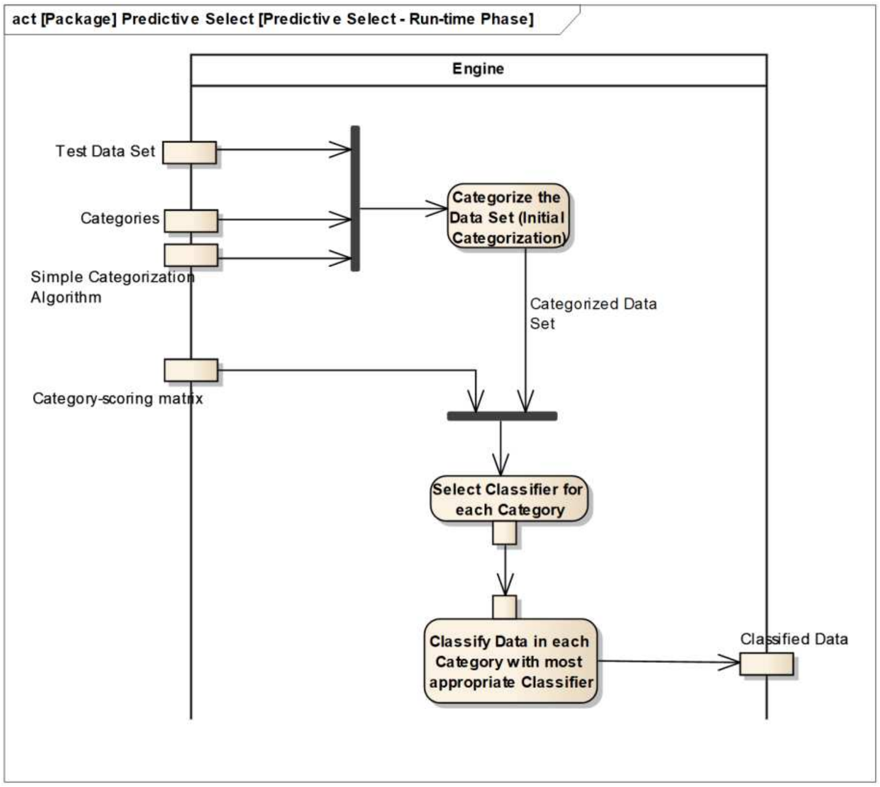

act [Package] Predictive Select [Predictive Select - Run-time Phase]
Engine
Test Data Set
Categories
Simple Categorization Algorithm
Category-scoring matrix
Categorize the Data Set (Initial Categorization)
Categorized Data Set
Select Classifier for each Category
Classify Data in each Category with most appropriate Classifier
Classified Data

Figure 15. Predictive Selection Meta-algorithm: Run-time Phase

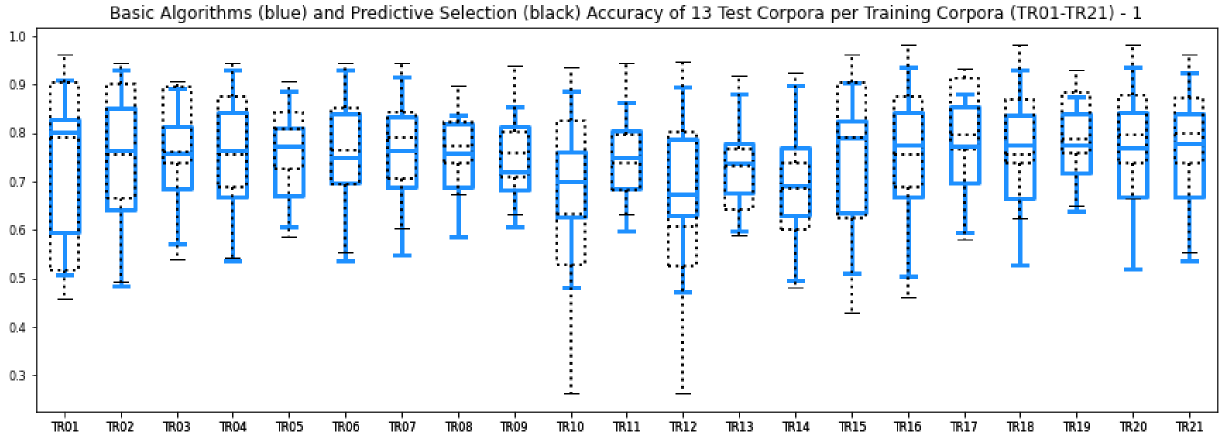


Figure 16. Basic Algorithms (blue) Compared with Predictive Selection (dotted-black) Accuracy across the 13 Test Corpora for each of the TR01-TR21 Training Corpora

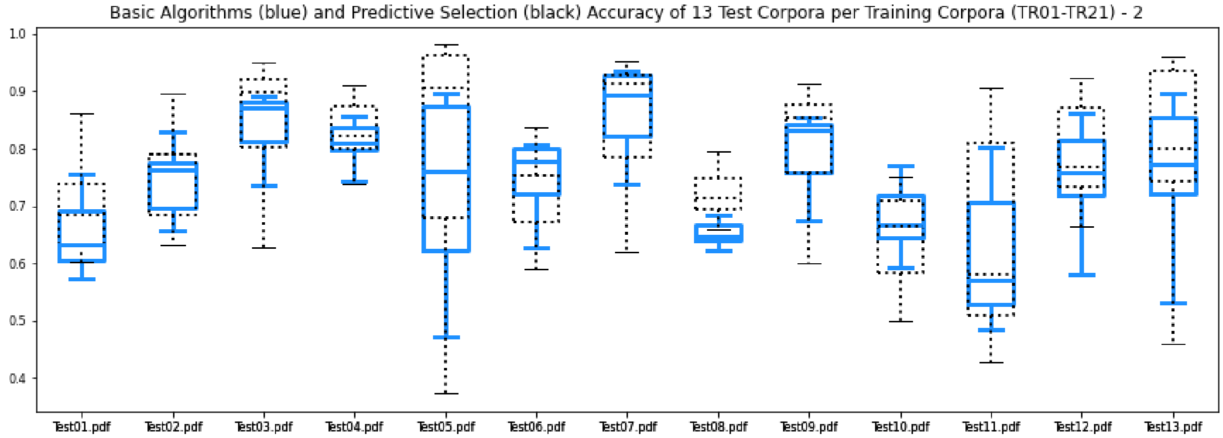


Figure 17. Basic Algorithms (blue) Compared with Predictive Selection (dotted-black) Accuracy across the 21 Training Corpora for each of the 13 Test Corpora

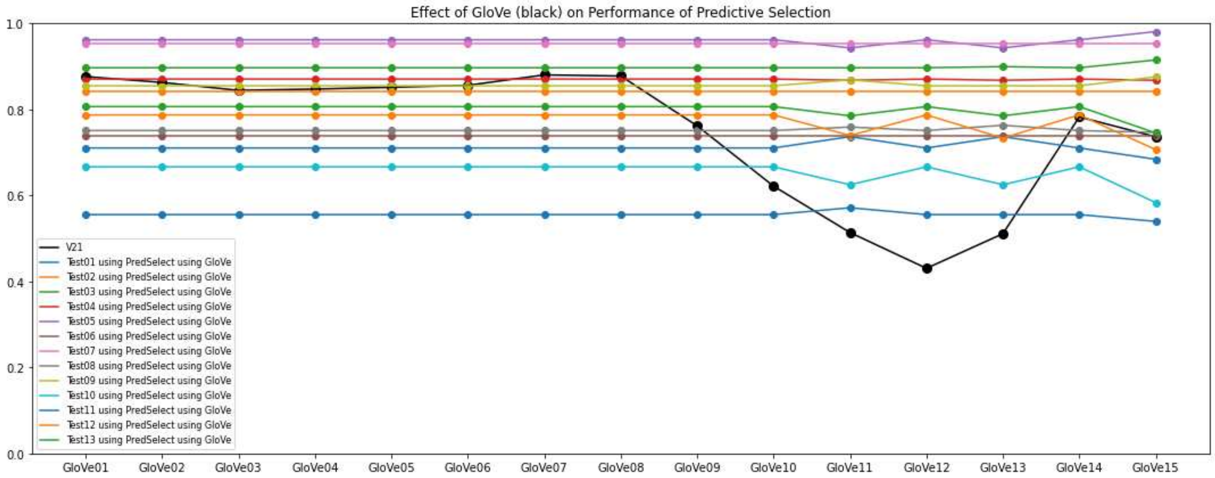


Figure 18. Using the V2 validation corpora, the effect of 15 GloVe variations as a preliminary categorizer on the performance of predictive selection on each of the 13 test corpora remains little-changed

2.5.3 Improvement

To additionally show the advantage of meta-algorithms, we improved the performance of weighted voting by adding a fourth component algorithm. As an illustrative example, we took the worst-performing set of traditional algorithms (TR07, Figure 5) and added GloVe07 to improve the classification of Test 13 from 0.74 to 0.88. Table 5 summarizes the results of Test 13. Note that this improvement is much more pronounced in Test 13 because of the poor results from the traditional algorithms for this Test corpus. Improvements are generally not universal by adding GloVe as a fourth component algorithm since our GloVe implementations were highly inconsistent. A mean improvement of 0.03 (from 0.75 to 0.77) has been observed as the poor-performing GloVe variants performed worse than the traditional algorithms and weighed down the improvements.

Table 5. Accuracies of Using Various Algorithms and the Effect of GloVe on Weighted Voting for Test13 using TR07 only

Algorithm or Meta-algorithm	Accuracy
(1) Naïve Bayes	0.80
(2) TF*IDF/Cosine Similarity	0.79
(3) Logistic Regression	0.64
(4) GloVe07	0.95
Weighted Voting (all variations) using (1), (2), and (3)	0.74
Weighted Voting (all variations) using (1), (2), (3), and (4)	0.88

With predictive selection, even though a potential improvement could be made by picking a much better preliminary categorizer than a simple pattern match or the fifteen GloVe variations, none were identified.

2.5.4 Limitations

We have identified risks to validity due to several limitations:

Limited publicly available ground-truthed data. While some body of work currently exists for stakeholder requirements classification, few publicly-available ground-truthed requirements corpora exist. Perhaps the most referenced is the OpenScience tera-PROMISE repository [29] of 625 labeled FRs and NFRs, with the NFRs broken down into a set of quality attribute requirements. Other works, such as [5], involve private data processed by multiple subject-matter experts over several weeks. For our purposes, every data point used for this study involved initial ground-truthing, which took many hours of review. We eventually ended up with five training corpora with over 1 million words and thirteen test corpora with over 2,000 requirements. For GloVe, we settled on a generic model trained with Wikipedia and newswire text.

Limited variety. One of the effects of limited ground-truthed data is we were limited to the four industries with which we had close ties — defense, energy, transportation, and construction, and heavily weighted to defense contracts. With the variety of system types, we expected that we would have obtained better results with representations from other industries. This is particularly true with classifying system requirements, as industry-specific systems often have industry-specific terms, acronyms, and initialisms.

Parsing imperfections. Parsing of corpora involved ingesting PDF files, which was rudimentary and dependent on properly written requirements with *shall* statements ending in periods. Statements with a *shall* but multiple bullet points were processed only based on the first period. For example, a requirement of the form on Table 6 translates to a single requirement, "*The system shall: statement 1.*" and the rest of the requirements, statement 2 through statement n, are ignored.

Table 6. Multiple Requirements Written as Bullets

```
The system shall:
      statement 1.
      statement 2.
      …
      statement n.
```

Poorly written requirements. While it is not expected that requirements follow all of INCOSE's recommendations for writing requirements [12], a certain level of quality was expected. With the set of corpora at our disposal, the larger projects appeared to have better-written SOWs, presumably because their potentially higher cost and schedule risk necessitated more experienced systems engineers to write the SOWs.

Disguised requirements. This study looked at classifying requirements into Governance and System, the former type levied on the contractor and the latter on the system. While most requirements have clear-cut classifications, a few straddle the line. In particular, some requirements initially appear to be Governance requirements but are really System requirements. For example, "the contractor shall design the module for reliability" is a requirement levied on the contractor/designer, but the implications are on the system being developed. We noticed that a construction SOW we had (Test 08) was rife with these requirements levied on the design-build company, but actually described the design constraints of the project. Other requirements, such as those focusing on cost, are even more blurred, as cost is the responsibility of both the project manager and engineer. One approach to alleviate this problem is multi-label classification similar to that espoused in [22] and using a rating system instead of a binary classification.

2.6 Conclusions

We initially ran 40 algorithms and meta-algorithm variations trained over 21 training corpora combinations and tested over 13 test corpora for a total of 10,920 combinations. The results are summarized as follows and in Table 7.

2.6.1 Training and Validation Corpora

In the absence of publicly-available ground-truthed training corpora, a substitution using the following was useful and provided reasonable results: i) several medium to large statements of work and specifications documents and combinations of them, and ii) Wikipedia + Gigaword combination for GloVe.

2.6.2 Comparison of Algorithms and Meta-algorithms

Table 7 shows a summary of all the results from our research, Figure 3 is a slice of that summary using TR21/V21 only, and Table 5 shows a different slice. These slices provide more pronounced differences in the algorithmic and meta-algorithmic implementations. Nonetheless, Table 7 shows an advantage of meta-algorithmic approaches, regardless of the sub-optimal variations of the component algorithms.

The simple pattern match, as expected, did not perform well, and the results were extremely sensitive to a subject matter expert picking the right words for matching (not robust). Naïve Bayes, despite its simplicity, provides the best results compared with processing time. This is consistent with what has been documented in the past [36] [37].

TF*IDF/Cosine Similarity and Naïve Bayes, despite using the same frequency vectors, varied in the training corpora, but performed similarly on the test corpora. Naïve Bayes, however, generally outperformed the former.

The number of anchor words heavily influenced GloVe and produced highly sensitive results, depending on the validation corpora (Figure 7). However, we did notice that fewer anchor words (GloVe01-Glove08) appeared to highly favor governance-heavy corpora, while more anchor words (GloVe09-GloVe15) favored system-heavy corpora (e.g., see Figure 8). We did see a “tightness” or convergence on GloVe09 which may indicate that the GloVe variation would be the best for future research.

Weighted voting was good but depended slightly on how the inferior algorithms fared. In some cases, the two inferior algorithms overwhelmed the best one and resulted in a misclassification, whereas the individual best classifier would have picked the correct one.

Predictive selection, heavily dependent on the component algorithms, improved upon weighted voting even with preliminary categorizers that were not ideal. A lower bound on classification accuracy can be obtained even with these non-ideal preliminary categorizers (Figure 18). GloVe is extremely heavy (i.e., it uses a lot of computational resources), and while it can be used to improve results in many cases, one must be cognizant of the resources needed for using GloVe. Perhaps better training and validation corpora would make GloVe the hands-down choice for classification, but for this study, we did not find using GloVe very compelling.

Table 7. Algorithm/Meta-algorithm Performance Summary Across All Test Corpora

	Algorithm	Type	Training	Validation	Test	Results Across All Test Corpora
(1)	Simple Pattern Match	Exact Match	N/A	N/A	Table 3	0.69
(2)	Naïve Bayes	ML Algorithm	Table 2	N/A	Table 3	0.76
(3)	TF*IDF/Cosine Similarity	ML Algorithm	Table 2	N/A	Table 3	0.76
(4)	Logistic Regression	ML Algorithm	Table 2	N/A	Table 3	0.75
(5)	GloVe09 (best variant of Glove01-GloVe15)	ML Algorithm (Deep Learning) [30]	[30] [34]	N/A	Table 3	Best (GloVe09): 0.74 All (GloVe01-15): 0.61 Highly inconsistent per variant.
(6)	Weighted Voting using (2), (3), and (4)	Meta-algorithm with components (2), (3), & (4)	Table 2	N/A	Table 3	0.75 Only as good as (2), (3), & (4) will allow
(7)	Predictive Selection using (1) as prelim. categorizer	Meta-algorithm with components (2), (3), & (4)	Table 2	N/A	Table 3	0.75 Only as good as (2), (3), & (4) will allow.
(8)	Predictive Selection using (5) variants as prelim. categorizer	Meta-algorithm with components (2), (3), & (4)	[30] [34]	Table 2	Table 3	0.76 Highly consistent for a particular test corpus regardless of GloVe performance as a preliminary categorizer. Dependent on quality of (2), (3), & (4)
(9)	Weighted Voting using (2), (3), (4), and (5)	Meta-algorithm with components (2), (3), (4), and (5)	Table 2	N/A	Table 3	0.77 Only as good as (2), (3), (4), & (5) will allow

The comprehensive results of all these combinations showed how meta-algorithms could stabilize results to improve the lower bound on accuracies given certain basic or traditional component algorithms. In addition, meta-algorithms have the advantage of remaining fresh and relevant no matter what new singular component algorithms are devised in the future. Meta-algorithms can and often become more powerful as these new components are employed. For our study, our GloVe implementation was not compelling by itself because of the enormous variations in the results. While

GloVe implementations did somewhat improve our meta-algorithmic results, a 60-fold increase in processing time negates a reason for including GloVe as a component unless a superior set of training corpora is identified.

2.7 Further Experimentation and Research

The 10,920 combinations of training, validation, test, and algorithms provided some insight into how classifying stakeholder requirements could be performed and improved. Because of a wide variety of parameters that could be substituted for those that we used (such as word anchors for GloVe, the training and validation corpora, various other mixes of component algorithms, other weighting methods, etc.), an exhaustive investigation was not feasible. However, the insights gleaned could be used to drive future similar efforts.

There is a lack of publicly accessible ground truth documents. Efforts to provide such documents to the community could start. We also found that parsing needed some work. Basic parsing of PDF files was performed and provided reasonably good results, but as described in the Limitations section above, classification results could benefit from a more robust parsing algorithm.

Classifying requirements not just on words but also on the context headers could prove useful. For example, requirements under the heading “Reliability Requirements” could provide additional weight on classifying those requirements as System and not Governance.

Realizing that the classification of SOW requirements is not always singular (that is, each document belongs to exactly one class), using multi-label classification [22] and providing weights on those could prove useful. Finally, the methods used for this research could be tailored for three or more classifications that are even fuzzier, such as classifying for quality attributes such as availability, security, sustainability, usability, and others.

3 Meta-Algorithmics for Extractive Summarization

3.1 Introduction

The claim that 90% of the world's data has been generated in the last two years has been a running trope in articles, blogs, and papers for over a decade. However, the claim has not been scientifically verified [38]. Nonetheless, the enormity of data being generated daily is mind-boggling. Regardless of the actual number, whether audio, video, multimedia, or text, the need for summarization has never been more relevant than it is now.

Text summarization can be either extractive or abstractive. While extractive summarization picks representative sentences or phrases in the context of the text verbatim, abstractive summarization attempts to generate novel sentences that do not necessarily exist in the text. Extractive summarization results in a subset of existing sentences or phrases, while abstractive summarization does not guarantee such a set since the generated sentences are not directly extracted from the sample. It is not unusual for words that do not exist in the text to appear in the summary. For better or for worse, extractive methods somewhat preserve the author's style while abstractive summarization does not necessarily do so. Simske and Vans [39] characterize extractive summarization as lossless and abstractive summarization as lossy, referring to the compression that either method performs.

This research aims to use a second-order meta-algorithm referred to as Tessellation and Recombination with Expert Decisioner [2] to determine if combining existing methods produces better results compared to their individual outcomes. While [40] argues that extractive summarization has mostly given way to abstractive summarization, the authors believe that meta-algorithmic techniques could be used to continue making advances in the field and be bases for use with new algorithms for both extractive and abstractive summarization. And even a hybrid approach where both techniques are combined to get even better results. Additional benefits for extractive summarization include retaining keywords and phrases suitable for indexing and preserving behavior [39] [41].

3.2 Related Work

3.2.1 Extractive Summarization

There already exists a plethora of work in text summarization, both extractive [42] and abstractive [43]. Extractive methods, more than abstractive methods, have been embedded in common libraries such as PyTextRank in spaCy [44] (an implementation of TextRank [45]), Gensim's TextRank [46], PySummarizer [47] (an implementation of Long Short-Term Memory (LSTM)), Stanford's CoreNLP Summarizer [48], and the Natural Language Toolkit (NLTK) [49]. Some unique work has also been done using particle swarm optimization (PSO) [50] which improved on MS Word-auto-summarized articles by 10-19%, depending on the metric. (The MS Word AutoSummarizer feature was discontinued in Word 2010 [51].)

3.2.2 Abstractive Summarization

Much work has also been done with abstractive summarization as described in [43]. More recently, with the explosive interest in generative adversarial networks (GANs) [52] for image processing and generating deep fakes, Bhargava *et al*. [53] have applied GANs methods to summarization.

3.2.3 Evaluation

Bilingual Evaluation Understudy (BLEU) [54], the SacreBLEU [55] standardization, and several variations of Recall-Oriented Understudy for Gisting Evaluation (ROUGE) [56] are common evaluation metrics for summarization due to their ease of implementation and efficiency [57]. But many other scoring methods exist; Ferreira *et al*. [58] describe, implement, and assess 15 of them. Wolyn and Simske [41] took a different approach by using functional methods. Specifically, they evaluated summaries based on both document classification and document query ranking compared to the original complete text. They did this with the CNN news corpora (the same corpora used in this paper) and a smaller set of novels from Project Gutenberg [59].

3.3 Research Goals

The overarching goal of this research was to investigate how meta-algorithmic techniques could be used to improve existing and future component algorithms. In doing so, we also compared metrics for measuring the appropriateness of the generated summaries.

3.4 Process/Tasks

We executed the following process tasks:

3.4.1 Dataset

The data used for this study was the Cable News Network (CNN) set used in [41] and first introduced in [60], a set of 3,000 English language articles from early 2015 spanning business, health, justice, living, opinion, politics, showbiz, sports, technology, travel, United States, and world news. Articles contained anywhere from 10 to 197 sentences with a mean of 38.4. The “Gold Standard” summaries ranged from 4 to 13 sentences with a mean of 7.2.

The dataset had been prepared by [60] in XML format which has the advantage of having many parsers available, including ElementTree [61], BeautifulSoup [62], and the standard XML Document Object Model (DOM) API [63]. We opted to use BeautifulSoup because of its maturity, having been available since 2004 [64].

3.4.2 Preparation and Initial Processing

The process of preparing and processing the articles is illustrated in Figure 11, taking advantage of the results of seven extractive summarizer algorithms prepackaged in NLTK’s Sumy and an algorithm that simply picks random sentences from the article. The seven algorithms include:

- Basic, or SumBasic [65], is the simplest of the algorithms we used, based on the premise that frequently occurring words are given more weight than those words that are not, and therefore drive the algorithm to pick sentences with those heavily weighted words.
- LexRank [66] is a graph algorithm that relies on the similarity of sentences with others. The idea is that a sentence with many similarities to other sentences weighs more than sentences that do not.
- The Luhn algorithm [67] is a heuristic extractive summarization algorithm with its roots in TF*IDF (Term Frequency-Inverse Document Frequency) and, while similar to SumBasic, discounts words that are too frequent (stop words).
- Latent Semantic Analysis, or LSA [68] is based on the idea that words that are more closely related to each other (such as *foot* and *shoe*) carry a higher weight than pairs that are remotely related (such as *foot* and *book*); and therefore, semantically more relevant.
- The TextRank [45] algorithm is similar to the PageRank algorithm used by Google and other search engines, with the difference that instead of web pages, TextRank ranks sentences based on similarities.
- Edmundson [69] is an algorithm that weighs sentences using word position, word frequency, usage of cue words (e.g., superlatives), and document structure (titles, sub-titles, etc.)
- The Kullback–Leibler algorithm, or KL [70], picks sentences based on entropy [71] [72] and can be very computationally heavy.

3.4.3 Tessellation and Recombination with Expert Decisioner

The Tessellation and Recombination with Expert Decisioner pattern (T&R) is a second-order meta-algorithmic pattern that first utilizes multiple generators (for this research, the algorithms identified above), tessellates the results, and recombines these tessellations into the desired results.

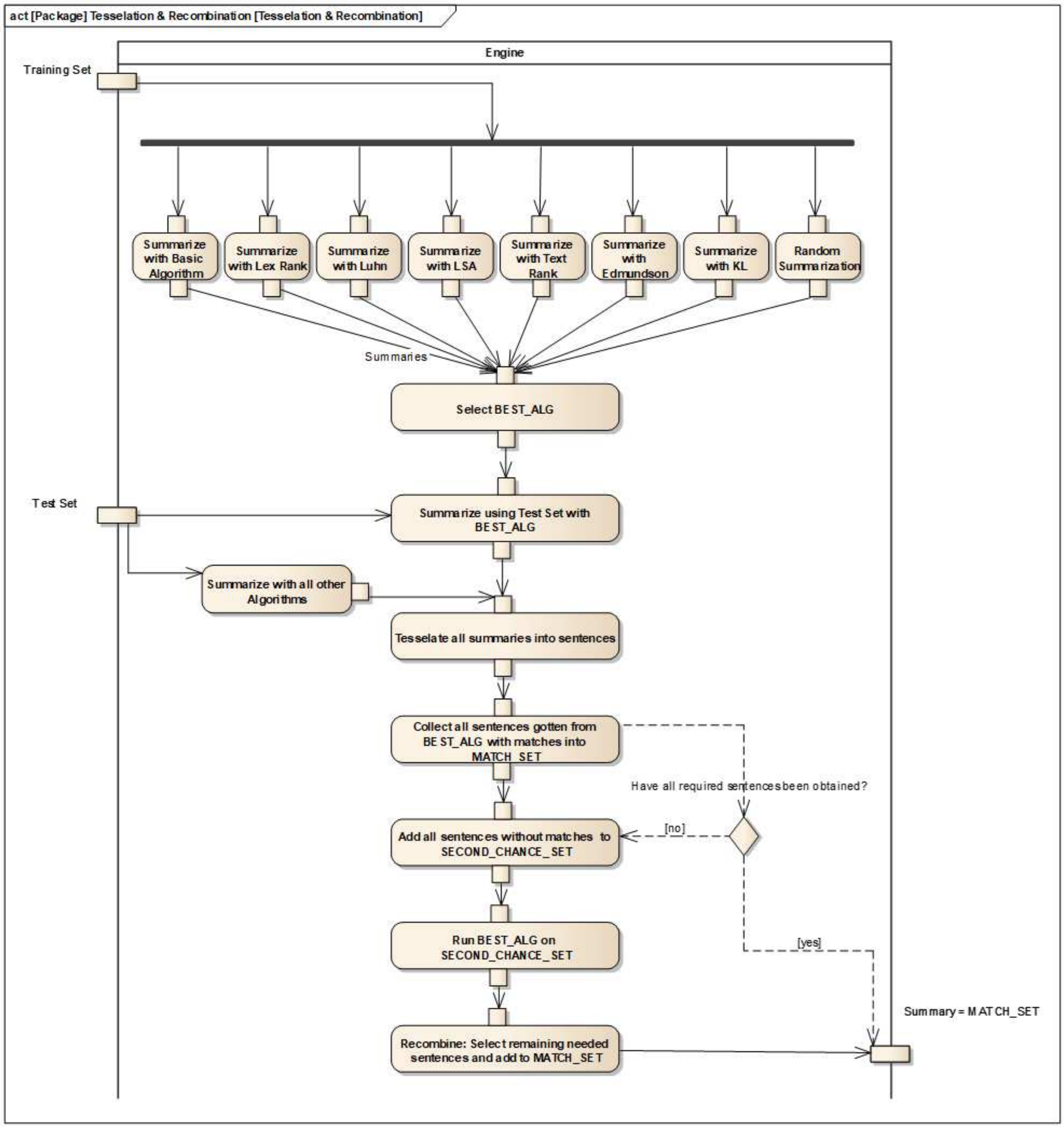

act [Package] Tesselation & Recombination [Tesselation & Recombination]
Engine
Training Set
Summarize with Basic Algorithm
Summarize with Lex Rank
Summarize with Luhn
Summarize with LSA
Summarize with Text Rank
Summarize with Edmundson
Summarize with KL
Random Summarization
Summaries
Select BEST_ALG
Test Set
Summarize using Test Set with BEST_ALG
Summarize with all other Algorithms
Tesselate all summaries into sentences
Collect all sentences gotten from BEST_ALG with matches into MATCH_SET
Have all required sentences been obtained?
[no]
[yes]
Add all sentences without matches to SECOND_CHANCE_SET
Run BEST_ALG on SECOND_CHANCE_SET
Recombine: Select remaining needed sentences and add to MATCH_SET
Summary = MATCH_SET

Figure 19. Tessellation and Recombination with Expert Decisioner Meta-algorithm Applied to the Extractive Summarization Task

Prior to tessellation, a *best component algorithm* (BEST_ALG) is chosen using the training set. This BEST_ALG becomes the basis by which the other (non-best) algorithms are compared and later used for the second pass at summarization. The test set is then evaluated with each of the algorithms and the summaries are tessellated by breaking them down (except for the random summary) into their constituent sentences. Because we are extracting sentences from the articles, there is a good probability that the sentences extracted (rather than abstracted) using one algorithm overlaps with sentences extracted using the other algorithms. In fact, using BEST_ALG as the primary algorithm, we measured a mean overlap of 85%, not including random sentence selection. We took advantage of this and used those overlapping sentences as part of the final summary.

The pattern is complete for those summaries with an overlap of 100%, and we are left with the desired summary with the desired number of sentences. For those with less than 100%, we collect all the sentences left over (those that did not match with sentences extracted by BEST_ALG) and ran them through the BEST_ALG one last time to generate the remainder of the summary. This indicates that since after the first pass, the number of sentences in the generated summary is less than our required number, we take the remaining sentences (that were not chosen during the first pass) and once again run them through our preferred summarizer (chosen during the first pass) to complete the summary. Note that we have a guarantee of only requiring a maximum of two passes for our method.

3.5 Analysis and Results

3.5.1 Metric Selection

For comparing the efficacy of the algorithms, we opted to use the Jaccard index (also called the Jaccard similarity coefficient), calculated simply as

$$J(A,B) = \frac{|A \cap B|}{|A \cup B|}$$

Equation 3

A represents the words used in the Gold Standard summary and B represents the words used in the hypothesis summary. The Jaccard similarity coefficient has been shown to be effective and efficient for keyword similarity [73] and in expanded form to measure text similarity [74]. We have verified its appropriateness by comparing results to those reported for the BLEU and ROUGE metrics which, for our CNN set, tended to score random sentence selection favorably over most of the Sumy algorithms.

For example, while Luhn's ROUGE recall results were superior to all others, Luhn's precision scores were unremarkable, weighing down Luhn's F1 score towards the bottom, at roughly half of a random selection's F1 score. Furthermore, using the Jaccard similarity coefficient, we verified that the Luhn algorithm's advantage is consistent with results from functional analyses (in the form of querying) performed in [41].

Originally developed as a metric to measure the appropriateness of language translations, the BLEU metric is a precision metric that measures how many of the n-grams in the hypothesis appear in the Gold Standard. BLEU precision is different from ROUGE precision in that BLEU includes a "brevity penalty" (which penalizes short hypotheses) that does not exist in ROUGE. ROUGE, by comparison, measures precision, recall, and the resulting F1 score rather than just precision.

3.5.2 Analysis

Results of our research are summarized in Figure 18 and Table 8, which reflect the mean values observed with ten randomized collections of training and test articles selected from the set of 3,000.

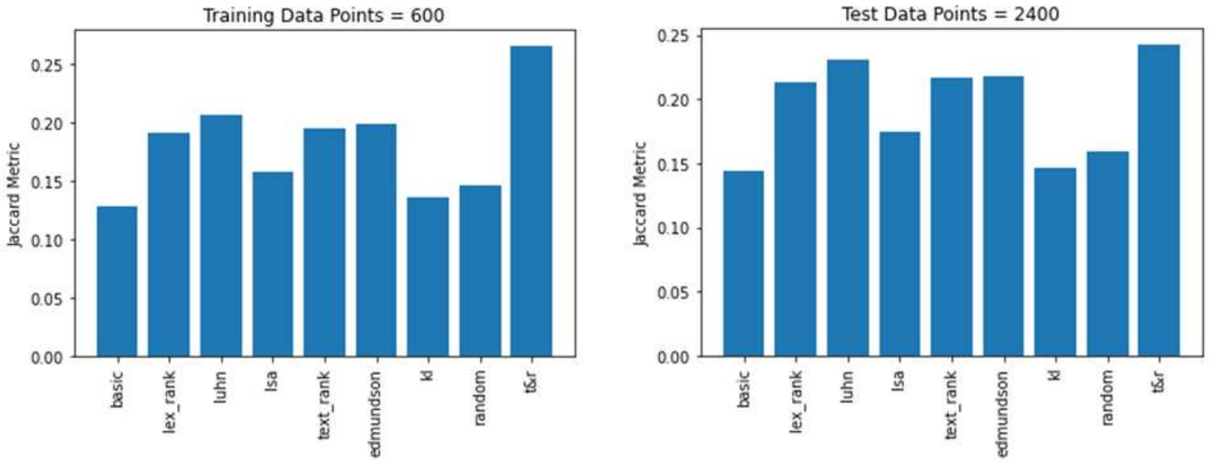


Figure 20. Results of using Tessellation and Recombination with Expert Decisioner

Using the Jaccard coefficient index, it is evident that the T&R approach produces improvements over all the individual component algorithms, including, on the extremes, a 4.9% improvement over Luhn and a 69% improvement over SumBasic.

Table 8. Jaccard Similarity Coefficient Results when using Tessellation and Recombination with Expert Decisioner

Algorithm	**Jaccard Similarity Coefficient**	**T&R Improvem ent**
Basic	0.1436	0.6877
LexRank	0.2131	0.1376
Luhn	0.2310	0.0492
LSA	0.1741	0.3922
TextRank	0.2168	0.1179
Edmundson	0.2185	0.1093
KL	0.1463	0.6566
Random	0.1599	0.5160
T&R	0.2424	N/A

Below is a comparison sample of the Gold Standard with several algorithm-generated summaries, including Basic, Luhn, and T&R from an article titled “Airport charges passengers for cool air.” A random extraction is also included. The Luhn and T&R summaries are identical except for the last sentence.

Gold Standard:

> *Venezuela's Simon Bolivar International Airport of Maiquetia demands $18 for air conditioning. Airport says its new air-con unit protects passenger health and adds ozone to the atmosphere. People have taken to Twitter to question new measure, criticize other facilities at the airport. Anyone departing from the Simon Bolivar International Airport of Maiquetia in Caracas now faces a levy of 127 bolivars ($18) to pay for a new air conditioning unit installed earlier this year, according to a statement on the airport's website. The airport says its air conditioning system "eliminates contaminants" and injects ozone into the atmosphere to improve the environment and protect the health of passengers. The "breathing tax" which came into force on July 1 and must be paid by all domestic and international passengers to airlines at check in, has generated bemusement in Venezuela, with many taking to Twitter to criticize the measure.*

Random:

> *The "breathing tax" which came into force on July 1 and must be paid by all domestic and international passengers to airlines at check in, has generated bemusement in Venezuela, with many taking to Twitter to criticize the measure. "Maiquetia airport is 36 C, it must be the injection of ozone that makes the temperature shoot up," they write. Another, Chipopo (@ELvDav) comments that whatever task the air conditioning unit performs, it doesn't seem to include lowering the temperature. It boasts that the system is the first of its kind in an airport anywhere in South America and the Caribbean. Earlier this week the country's air transport minister, Luis Caraballo Graterol, insisted in an interview that Venezuela had no debts to foreign air carriers. Several international airlines, including Air Canada and Alitalia have suspended or reduced flights to Venezuela in recent months, blaming strict government currency controls that they say have left them billions of dollars out of pocket.*

Basic:

> *Plane passengers are used to paying for check-in baggage or priority boarding, but one airport in Venezuela is now charging for the ultimate hidden extra -- air. The airport says its air conditioning system "eliminates contaminants" and injects ozone into the atmosphere to improve the environment and protect the health of passengers. Daniel Martinez (@DanielMartinezD), a radio presenter, tweeted his gripes about the measure. Several international airlines, including Air Canada and Alitalia have suspended or reduced flights to Venezuela in recent months, blaming strict government currency controls that they say have have left them billions of dollars out of pocket. Built in 1945, Maiquetia airport -- as it is known by locals -- is the main international air hub serving Venezuela. It's currently undergoing a major overhaul, according to Worldtravelguide.net.*

Luhn:

> *Anyone departing from the Simon Bolivar International Airport of Maiquetia in Caracas now faces a levy of 127 bolivars ($18) to pay for a new air conditioning unit installed earlier this year, according to a statement on the airport's website. The airport says its air conditioning system "eliminates contaminants" and injects ozone into the atmosphere to improve the environment and protect the health of passengers. It boasts that the system is the first of its kind in an airport anywhere in South America and the Caribbean. The "breathing tax" which came into force on July 1 and must be paid by all domestic and international passengers to airlines at check in, has generated bemusement in Venezuela, with many taking to Twitter to criticize the measure. "While the stench of the toilets asphyxiates me ... they have started to charge 127 bolivars for breathing the ozone," wrote Vero (@VeronicaTorresA). "Maiquetia airport is 36 C, it must be the injection of ozone that makes the temperature shoot up," they write.*

T&R:

> *Anyone departing from the Simon Bolivar International Airport of Maiquetia in Caracas now faces a levy of 127 bolivars ($18) to pay for a new air conditioning unit installed earlier this year, according to a statement on the airport's website. The airport says its air conditioning system "eliminates contaminants" and injects ozone into the atmosphere to improve the environment and protect the health of passengers. It boasts that the system is the first of its kind in an airport anywhere in South America and the Caribbean. The "breathing tax" which came into force on July 1 and must be paid by all domestic and international passengers to airlines at check in, has generated bemusement in Venezuela, with many taking to Twitter to criticize the measure. "While the stench of the toilets asphyxiates me ... they have started to charge 127 bolivars for breathing the ozone," wrote Vero (@VeronicaTorresA). The toilets have no water, the air-con is broken, there are stray dogs inside the airport, but there's ozone?"*

For comparison, BLEU and ROUGE-1 (unigram) F1 metrics were computed after performing T&R and obtained the results in Table 9.

Table 9. Jaccard Results Compared to BLEU and ROUGE-1.

Algorithm /	**Jaccard**		**BLEU**		**ROUGE-1**	
Meta-algorithm	**Index**	**Rank**	**Index**	**Rank**	**F1**	**Rank**
Basic	0.1436	9	0.00238	1	0.1499	1
LexRank	0.2131	5	0.00157	5	0.1014	4
Luhn	0.2310	2	0.00126	8	0.0696	8
LSA	0.1741	6	0.00164	4	0.0930	5
TextRank	0.2168	4	0.00122	9	0.0675	9
Edmundson	0.2185	3	0.00182	2	0.0886	6
KL	0.1463	8	0.00143	6	0.1295	2
Random	0.1599	7	0.00174	3	0.1158	3
T&R	0.2424	1	0.00133	7	0.0731	7

A cursory look into these summaries reveals that, arguably, the BLEU and ROUGE-1 scores and rankings do not accurately reflect the relative summary representation of the complete text unlike those suggested by the Jaccard rankings. This is supported specifically by the relatively high score of the “Random” algorithm for both BLEU and ROUGE-1 (Rank = 3 for both). If a random summarization is evaluated as good or better than most of the summarization techniques, there is either something amiss with the documents or with the evaluation approach. It appears that BLEU and ROUGE-1 both rank random summarizations too highly, undermining their overall credibility. This is quite different from the Jaccard metric. In fact, Figure 21 reveals that while both BLEU and ROUGE-1 trend toward a positive correlation for their rankings in Table 9 ($R^2 = 0.593, p > 0.05$), Jaccard and BLEU trend toward negative correlation ($R^2 = 0.365, p > 0.05$), and Jaccard and ROUGE-1 are strongly negatively correlated ($R^2 = 0.799, p = 0.0099$).

The apparent accuracy of Jaccard over BLEU and ROUGE may, ironically, lie in its simplicity. Jaccard looks only at word sets (minus stop words) and should be considered for evaluating any summarization technique, perhaps as the main metric, as we have done here, or as a supplement for evaluating the goodness of summaries. We surmise that as long as the sentences are extracted, there is a presumption that the sentences are well-formed. Therefore, focusing on word set intersections and unions only (as in the definition of the Jaccard index) is sufficient in providing accurate measures.

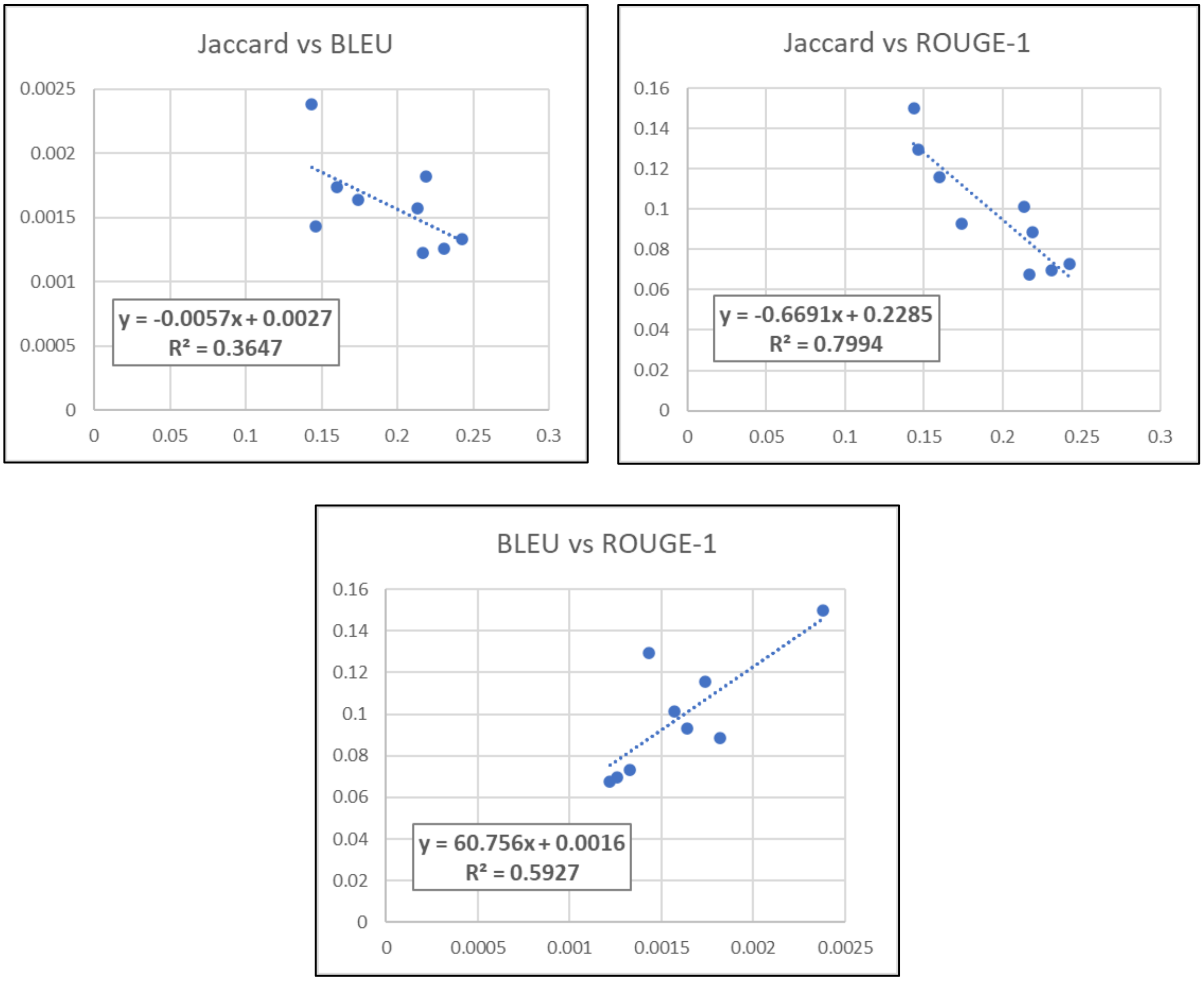


Figure 21. Correlation Among All Three Considered Metrics: Jaccard, BLEU, and ROUGE-1

3.5.3 Limitations

Because of how the SECOND_CHANCE_SET is obtained and tagged at the end of the MATCH_SET, there is a probability that the summary will contain sentences that are not in the same order as the original text. This phenomenon is most likely to be a relevant consideration for summarizing timeline-critical corpora such as novels. New articles, such as in our study, are less affected by small perturbations in sentence order.

3.6 Conclusion

We have shown that, given existing algorithms and regardless of their individual efficacy, we are able to obtain a 5% improvement in summarization results of news articles using the second-order meta-algorithmic pattern Tessellation and Recombination with Expert Decisioner.

We have also demonstrated that careful consideration of metrics is important when evaluating summarization algorithms. Neither BLEU nor ROUGE-1 F1 results, for example, sufficiently reflected the appropriateness of the summaries generated by our algorithms. Both BLEU and ROUGE ranked random selection higher than six of the eight algorithms. For the CNN corpus, the Jaccard similarity method is posed as a more germane means of assessment.

3.7 Further Experimentation and Research

As described in the *Limitations* section above, the readability of summaries could potentially be improved by the accurate ordering and placement of the SECOND_CHANCE_SET sentences as they relate to the rest of the generated summary. Since this would not improve the summary's Jaccard similarity coefficient as we have used it in this research, a modified version of the metric would have to be constructed or an entirely different metric would have to be considered.

Also, to further verify our research findings, we propose repeating the querying and classification of functional tests described in [41] to study the effects of our proposed methods.

Finally, T&R, in the form we have described above, would not be suitable for abstractive summarization as their component algorithms are likely to construct sentences that are unique to the algorithm. As such, picking MATCH_SET (see Figure 11) would likely result in a null set. However, there is good potential that combining both abstractive and extractive methods with another meta-algorithmic pattern would result in even better results than what we have achieved here.

4 Functional Analytics for Document Ordering

4.1 Introduction

This research provides, through our investigated algorithms, various reading orders for (1) curriculum development and (2) optimal learning. Curriculum ordering involves sequencing documents such that the transition between topics covered is smooth and builds upon previous material. On the other hand, optimal learning aims to maximize functional measurements such as proficiency testing.

4.2 Related Work

Some work exists in the area of document ordering, but it is not extensive. We list here some of the more recent research.

Ramya Thinniyam [75], in her doctoral dissertation "On Statistical Sequencing of Document Collections," focused on automated statistical techniques for determining the chronological order of corpora using only the words contained in them. With the presumption that documents composed closer in time will be more comparable in their substance, she proposes (1) calculating the distance between document pairs using only their word features and (2) estimating the optimal document order based on these distances. The research examined different types of distances that can be determined between document pairs and introduced methods for sequencing a set of documents based on their pairwise distances.

Devi *et al*. [76], in "A Novel Approach for Curriculum Ordering of Course Topics Using Data Mining," propose a novel approach for ordering curriculum topics of a course using data mining techniques. The method involves extracting a set of relevant metadata features from the course content (such as learning objectives, course materials, and assessments), identifying the relationships between these features, and generating a weighted graph model of the course structure. The graph is then analyzed using data mining algorithms to identify the optimal ordering of course topics.

Kahani *et al*. [77], in "A new algorithm for optimal curriculum ordering using genetic algorithm and local search," propose to create an effective curriculum structure that enhances student learning outcomes and engagement. Their study showed that their approach identified an optimal ordering of course topics that significantly improved student learning outcomes and engagement.

4.3 Research Goals

Informational material is designed to provide knowledge on a particular subject and can be presented in various forms, such as lectures, training modules, or book chapters. Depending on the purpose, readers may choose to consume this material in different orders or subsets. For example, someone who is already familiar with a particular topic may choose to read only specific chapters or sections of a book or training module rather than starting from the beginning.

However, the default consumption order for informational material is usually in the order of chapter or lecture number. The author, editor, or publisher often chooses this order to provide a logical progression of ideas and ensure that readers have the necessary background knowledge before moving on to more advanced concepts.

That said, not all orders are equal, and some may be better suited to certain purposes. For example, if the goal is to gain a general overview of a subject, reading the introduction and conclusion chapters of a book or training module may be sufficient. On the other hand, if the goal is to develop a deep understanding of a subject, reading all the chapters in a logical sequence may be necessary.

In some cases, orders may not be predetermined, such as when assembling a curriculum from existing documents. This can involve selecting the most relevant chapters or sections from multiple sources and organizing them in a logical sequence.

The goal of our research is to develop ordering algorithms that improve comprehension and facilitate the creation of effective training curricula. By understanding the most effective order in which to present information, educators can improve student learning outcomes and ensure that key concepts are fully understood.

Additionally, predicting (or suggesting) the order of chapters in a textbook, lecture, or other sets of documents may help readers plan their study schedule better and provide a framework for understanding how the different pieces of information fit together.

4.4 Process/Tasks

While predicting the order of chapters as composed by the author may be an interesting exercise and offshoot of our research, the main intended use is generating a proposed reading order. Figure 22 summarizes the many combinations of the algorithms we used to investigate their effectiveness.

4.4.1 Overview

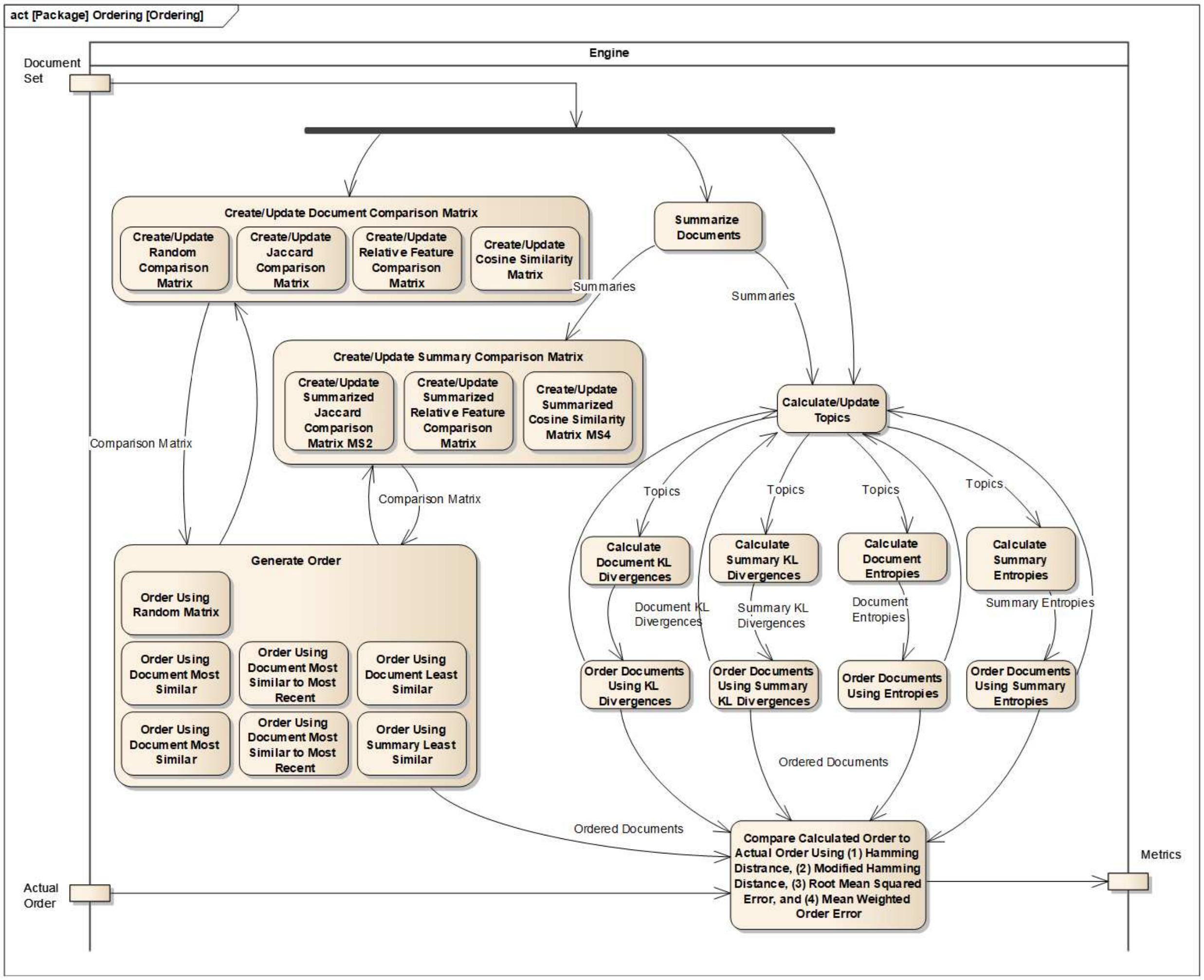

act [Package] Ordering [Ordering]
Engine
Document Set
Create/Update Document Comparison Matrix
Create/Update Random Comparison Matrix
Create/Update Jaccard Comparison Matrix
Create/Update Relative Feature Comparison Matrix
Create/Update Cosine Similarity Matrix
Summarize Documents
Summaries
Summaries
Create/Update Summary Comparison Matrix
Create/Update Summarized Jaccard Comparison Matrix MS2
Create/Update Summarized Relative Feature Comparison Matrix
Create/Update Summarized Cosine Similarity Matrix MS4
Comparison Matrix
Comparison Matrix
Calculate/Update Topics
Topics
Topics
Topics
Topics
Calculate Document KL Divergences
Calculate Summary KL Divergences
Calculate Document Entropies
Calculate Summary Entropies
Document KL Divergences
Summary KL Divergences
Document Entropies
Summary Entropies
Order Documents Using KL Divergences
Order Documents Using Summary KL Divergences
Order Documents Using Entropies
Order Documents Using Summary Entropies
Generate Order
Order Using Random Matrix
Order Using Document Most Similar
Order Using Document Most Similar to Most Recent
Order Using Document Least Similar
Order Using Document Most Similar
Order Using Document Most Similar to Most Recent
Order Using Summary Least Similar
Ordered Documents
Ordered Documents
Compare Calculated Order to Actual Order Using (1) Hamming Distrance, (2) Modified Hamming Distance, (3) Root Mean Squared Error, and (4) Mean Weighted Order Error
Actual Order
Metrics

Figure 22. Overall Ordering Process

4.4.2 Dataset

To have a good understanding of how our algorithms perform, it is necessary to employ several datasets and types of datasets for testing. The test corpora we used are summarized in Table 34 and control corpora are listed in Table 33. In choosing the test corpora (specifically the textbooks and possibly more so, courses and dissertations), we expected that we could generally detect order in the documents. For those that do not appear to require or have a specific order, we would attempt to explain away.

In choosing these test corpora, we have endeavored to limit the minimum number of chapters to six, giving us at least 6! = 720 sequence variations, reducing perfect ordering simply by randomly guessing. (e.g., three would be too easy to guess with only six different sequence variations.)

- **Courses** – While course lectures are often meant to be progressive and order-dependent (i.e., lectures built upon previous lectures), skipping lectures, depending on the course, may or may not significantly affect comprehension. Prof. Simske of the Department of Systems Engineering donated most of the courses we analyzed.
- **Dissertations** – Dissertations are great for ordering exercises because usually, there is a single focused topic, and each chapter of a 100-300-page doctoral dissertation is sufficiently long for analysis. The dissertations we analyzed were retrieved from the Colorado State University repository for doctoral dissertations and master's theses. However, we limited our selection to doctoral dissertations which tend to be longer and much richer in content.
- **Journals** – Journals are periodicals that accept authors from various organizations that do not necessarily work on research related to each other. Therefore, papers published in journals are self-contained and do not rely on other papers within that publication. Related subject areas may be grouped within chapters while no real required reading order is suggested, either explicitly or implicitly. However, we surmised that papers have a loose order in which more general topics are covered earlier and more specific ones are covered later.
- **Textbooks** – While textbooks can and are often presented in the order in which they are meant to be read, instructors using those textbooks for instruction often jump from chapter to chapter in an alternative order, sometimes skipping chapters altogether. This indicates that chapters can and do often stand on their own. The collection of textbooks we used was obtained from the Colorado State University library, donated by professors, or sourced from the public domain.

In addition to the above, we use several control documents to provide contrast.

- **Biographies** – While biographies rely on the progression of a story, and thus reading a middle chapter before reading the introduction typically would not make for a sensible narrative, their order is time-based and not information-based. That is, in general, the prerequisite for understanding a later chapter relies on the understanding of previous chapters because something happened earlier in time that caused something later in time. Foundational knowledge in earlier chapters relies on time but does not necessarily add depth to the whole. On the other hand, our algorithms are depth-based in that entropy of words and similarity of word structures are the pieces examined for relationships. All biographies were retrieved from Project Gutenberg [78].
- **Novels** – While fiction in nature, novels are time-based narratives as with biographies and thus have similar limitations. Novels often have characters that appear periodically and may have several interwoven sub-narratives, each of which might be sequential but not necessarily. All novels were retrieved from Project Gutenberg. [78]
- **Wikipedia Articles** – Wikipedia articles are, by design, meant to stand independent of one another, even if hyperlinking to other articles is encouraged. With millions of authors, this is also the de facto structure. In principle, however, it is impossible for all articles to present all requisite knowledge within the articles themselves. For example, the article on Petri Nets assumes that the reader has an understanding of set and graph theory and can read mathematical notation pertinent to them.

4.4.3 Data Preparation

Each dataset described above (genres) was treated as a collection of documents split into logical documents. Dissertations, textbooks, novels, and biographies were split into author-identified chapters. Dissertations contained a minimum and usually not more than six chapters each, and textbooks, novels, and biographies had a much wider range of numbers of chapters. Course sets were split into weekly lectures and were thirteen or fifteen weekly lectures long, depending on whether they occurred during the summer or a regular semester. Wikipedia article sets were split into individual articles. And journals were split into their component articles.

Preparatory processing steps were first performed, including the removal of stop words (using the SciKit-learn library [31] plus [*abstract*, *introduction*, *chapter*, *figure*, *fig*, *table*]) and lemmatization of each document. We then created a document-term frequency matrix from the remaining terms.

4.4.4 Sequencing Method

4.4.4.1 Similarity Matrix-based Content Sequencing Method Using Complete Documents

4.4.4.1.1 Step One: Generate Similarity Matrix

Given an array of n documents D, we define its similarity matrix $\boldsymbol{S}_D = (\gamma_{ij})$ which is an nxn symmetric matrix. If each γ_{ij} represents how the row document d_i compares with the column document d_j (or vice versa), we have

$$\gamma_{ij} = \begin{cases} 1 & \text{if } i = j \\ c_{ij} & \text{if } i \neq j \end{cases}$$

Equation 4

All the values c_{ij} depend on how comparisons are handled. We use three methods for comparison with varying results: Cosine Similarity, Jaccard similarity, and a metric we named the Relative Probability Similarity.

4.4.4.1.1.1 Cosine Similarity

Cosine similarity is a common method of comparing two documents based on their content. The Cosine Similarity of two documents a and b is calculated:

$$s_C(a,b) = \cos(a,b) = \frac{a \cdot b}{\|a\|\|b\|} = \frac{\sum_{i=1}^{n} a_i b_i}{\sqrt{\sum_{i=1}^{n} a_i^2}\sqrt{\sum_{i=1}^{n} b_i^2}}$$

Equation 5

For our example, the similarity of the chapters using Cosine Similarity may look like Table 10.

Table 10. Document Similarity Matrix, $\boldsymbol{S}_D$, using Cosine Similarity

	d_1	d_2	…	d_n
d_1	1	$s_C(1,2)$	…	$s_C(1,n)$
d_2	$s_C(1,2)$	1	…	$s_C(2,n)$
…	…	…	…	…
d_n	$s_C(1,n)$	$s_C(2,n)$	…	1

4.4.4.1.1.2 Jaccard Similarity

Jaccard similarity is calculated by using the overlap of the words that comprise the documents:

$$s_J(a,b) = J(A,B) = \frac{|A \cap B|}{|A \cup B|}$$

Equation 6

where A and B are the sets of words (features) contained in the corresponding a and b documents. The matrix using Jaccard similarity thus looks like Table 11.

Table 11. Document Similarity Matrix, $\boldsymbol{S}_D$, using Jaccard similarity

	d_1	d_2	…	d_n
d_1	1	$s_J(1,2)$	…	$s_J(1,n)$
d_2	$s_J(1,2)$	1	…	$s_J(2,n)$
…	…	…	…	…
d_n	$s_J(1,n)$	$s_J(2,n)$	…	1

For our calculations, we restricted ourselves to single words and not multi-n-grams.

4.4.4.1.1.3 Relative Probability Similarity

Our third way of comparing documents is by a feature-wise comparison. We define the similarity of two documents by:

$$s_R(a, b) = 1 - \sum_{i=1}^{n} |f_i(a) - f_i(b)|$$

Equation 7

where $f_i(d)$ is the ith feature (word) of the document vector d.

We considered but opted against an alternate definition,

$$s_R(a, b) = \sum_{i=1}^{n} \alpha\beta$$

Equation 8

where

$$\alpha = \begin{cases} 0 & if\ f_i(a) = 0 \\ \log\big(f_i(a)\big) & if\ f_i(a) \neq 0 \end{cases} \quad \text{and} \quad \beta = \begin{cases} 0 & if\ f_i(b) = 0 \\ \log\big(f_i(b)\big) & if\ f_i(b) \neq 0 \end{cases}$$

as s_R did not accurately represent similarity when documents were subjectively evaluated.

For our chosen method (Equation 7), our document similarity matrix is represented as in Table 12.

Table 12. Document Similarity Matrix, $\boldsymbol{S}_D$, using Relative Probability Similarity

	d_1	d_2	…	d_n
d_1	1	$s_R(1,2)$	…	$s_R(1,n)$
d_2	$s_R(1,2)$	1	…	$s_R(2,n)$
…	…	…	…	…
d_n	$s_R(1,n)$	$s_R(2,n)$	…	1

4.4.4.1.2 Step Two: Generate the Sequence

Multiple ways of generating sequences were investigated. In all sequence-generating schemes, it was important to have a baseline comparison sequence. For this purpose, we generated 100 random sequences and calculated the metrics for each. The means of these metrics for each of the different sequences are compared with the results from the following three methods.

4.4.4.1.2.1 Sequence Most Similar to All Selected Documents

This method of generating a sequence assumes that the first document is the most general and therefore most similar to all the other documents in the set. Ensuing (sequentially following) documents chosen to build the sequence are chosen for their similarity to the already chosen documents, relying on the concept that transitions are smoothly bridged between the documents.

The sequencing of the documents is initially based on the matrix as described in one of the methods in 4.4.4.1.1 and is iterative. Operating on $\boldsymbol{S}_D$ as defined above and referring to Equation 4, we select the next document

$$d_{next} = \left\{ d_{max} \middle| \{d_{max} \in D\} \wedge \left\{ d_{max} \hat{=} \min_{d_i}(\bar{\gamma}_{ij}) \, , i \neq j \right\} \right\}.$$

Equation 9

as the first element in the computed sequence. d_{max} is removed from $\boldsymbol{S}_D$, which is then recalculated as an $(n-1)x(n-1)$ $\boldsymbol{S'}_D$ matrix with the remaining documents using the chosen Step 1 matrix similarity method.

The next document is chosen similarly from the $\boldsymbol{S'}_D$ matrix with the exception that each of the remaining documents is compared with the mean of the values of the already chosen documents. Iterations are repeated until the original $\boldsymbol{S}_D$ is reduced to a $1x1$ matrix. At which point, a complete sequence will have been generated.

4.4.4.1.2.2 Sequence Most Similar to Most Recent Document

This method shares a lot of similarities with 4.4.4.1.2.1 *Sequence Most Similar to All Selected Documents*, with a deviation in the way the iterative matrices are calculated. Rather than the second and ensuing documents chosen from the resulting $\boldsymbol{S'}_D$ matrix with each of the remaining documents compared with the mean of the values of the already chosen documents, they are compared with only the last chosen document. As with 4.4.4.1.2.1, iterations are repeated until the original $\boldsymbol{S}_D$ is reduced to a $1x1$ matrix. While this method of sequencing also picks the most general document as the starting point, ensuing documents are chosen to be most similar to the last previously chosen document only, attempting to create a smooth transition from document to document.

4.4.4.1.2.3 Sequence Least Similar to All Selected Documents

The third way of sequencing using the similarity matrix method can be thought of as the opposite of the 4.4.4.1.2.1. This method attempts to generate order with the idea that the more specific topics are necessary building blocks to grasp the information presented by the document that is the culmination of the introductory material.

Again, sequencing the documents is initially based on the matrix as described in one of the methods in 4.4.4.1.1 and is iterative. Operating on $\boldsymbol{S}_D$, we select the next document

$$d_{next} = \left\{ d_{min} \middle| \{d_{min} \in D\} \wedge \left\{ d_{min} \mathrel{\widehat{=}} \min_{d_i}\left(\bar{\gamma}_{ij}\right), i \neq j \right\} \right\}.$$

Equation 10

as the first element in the computed sequence. d_{min} is removed from $\boldsymbol{S}_D$, which is then recalculated as an $(n-1)x(n-1)$ $\boldsymbol{S'}_D$ matrix with the remaining documents using the chosen Step 1 matrix similarity method.

4.4.4.1.3 Variation: Similarity Matrix-based Content Sequencing Method Using Summaries

While the above methods were first employed on the full texts of the documents, we were also interested in whether operating on the summaries of these texts would provide similar results. A strong positive correlation using reasonably-sized summaries could provide enhanced processing speed at worst. And at best, operating on summaries could provide even better sequence ordering.

We opted for an extractive summarizer to guarantee compact verbatim extractions that used words that existed in the original text. Such was necessary so that 1) topic extraction would operate on identical words to those that exist in the full text and 2) similarity comparisons such as Jaccard and Cosine Similarity could similarly operate on the summaries as with the full documents without the use of expensive word embeddings such as Word2Vec [79] or GloVe [30]. The Luhn heuristic extractive summarization algorithm [67] was chosen from the many algorithms provided in the Sumy library [80] based on its performance in Chapter 3. Luhn is based on TF*IDF (Term Frequency-Inverse Document Frequency), and discounts stop words.

4.4.4.2 Entropy Ordering Method of Complete Documents

Entropy can be used in at least two potential ways as ordering methods. The first is the idea that introductory corpora are more general and provide more general, superficial treatise of the topics covered by the rest of the corpora. The second is the idea that less entropic corpora are prerequisite material for understanding the most entropic corpora.

4.4.4.2.1 Step One: Generate the Topics

For topic generation, we used Latent Dirichlet Allocation (LDA), described in the seminal paper by Blei *et al.* [10] [81] and implemented in the Gensim library [11] [82]. LDA is highly dependent on the number of topics passed as a parameter. But in order to limit the number of variables for this research, we standardized by choosing the number of topics to be 20% of the total number of documents as a 6reasonable number (see 4.5.1.1). For example, an introductory systems engineering textbook may cover the topics of requirements engineering, architecture and design, development and integration, verification and validation, and maintenance. A text on chess may cover history, rules of chess, openings, middle game, and end game. Undoubtedly, this is a very rough approximation; different corpus collections have various numbers of topics. Regardless, optimizing for the ideal number of topics for use in entropy calculations is left as a topic for future research.

We define the topic array as

$$T = \begin{bmatrix} t_1 \\ t_2 \\ \dots \\ t_p \end{bmatrix}, \text{ and each } t_i = [w_1 \quad w_2 \quad \cdots \quad w_m].$$

Equation 11

where w_i is the ith keyword-probability tuple of the m keyword-probability tuples that represent the document. For illustrative purposes, Table 13 lists topics generated by LDA for the ‘Simske Functional Applications’ book.

p is the number of topics chosen and t_i is the ith topic vector with m word tuples. Table 13 shows a sample matrix T (extracted from our test dataset ‘Simske Functional Applications’ book from Table 34. Each column represents a topic defined by a set of tuples of words and their percentage contribution to defining the topic.

Table 13. Sample topic matrix, T, from corpus ‘Simske Functional Applications’ book

Topic 1 (t_1)	**Topic 2 (t_2)**	**Topic 3 (t_3)**
(cluster, 0.03477)	(text, 0.01935)	(summarization, 0.01954)
(class, 0.02063)	(learn, 0.01473)	(weight, 0.01823)
(data, 0.01848)	(accuracy, 0.01276)	(sentence, 0.01678)
(classification, 0.01524)	(translation, 0.01266)	(word, 0.01666)
(equation, 0.01086)	(language, 0.01191)	(text, 0.01565)
(distance, 0.01021)	(document, 0.01093)	(document, 0.01518)
(train, 0.009958)	(data, 0.009477)	(count, 0.01059)
(categorization, 0.009648)	(order, 0.009465)	(example, 0.00941)
(example, 0.008913)	(read, 0.009278)	(use, 0.008444)
(score, 0.008463)	(example, 0.008173)	(language, 0.008234)

Based on these definitions, each document (chapter) in the corpus collection is represented best by a topic, referred to as the dominant topic. Table 14 illustrates this.

Table 14. Sample Document-Topic assignment of corpus 'Simske Functional Applications' book

Document	Chapter Title	(Dominant) Topic
Chapter 1	Linguistics and NLP	Topic 3
Chapter 2	Summarization	Topic 3
Chapter 3	Clustering, Classification, and Categorization	Topic 1
Chapter 4	Translation	Topic 2
Chapter 5	Optimization	Topic 2
Chapter 6	Learning	Topic 2
Chapter 7	Testing and Configuration	Topic 2

4.4.4.2.2 Step Two: Generate the Sequence

Calculate Corpus Entropy Given Topics: After evaluating all the corpora documents through Gensim to generate the topics $(0..5)$, with each topic, the next step was to calculate the makeup of each chapter

$$D_W = \begin{bmatrix} d_{W_1} \\ d_{W_2} \\ \cdots \\ d_{W_n} \end{bmatrix},$$

Equation 12

where d_{Wi} is the document-term vector for each document i of the collection of n documents.

Using the document-term matrix D_W and Topics table T, we generate the document-topic matrix D_T, which is comprised of the percentage of the topics that are discussed in every given document in D. The Document topic matrix explains how k topics are distributed in the n documents. From our D (Table 1) and T (Table 2) matrices, we generate the following document topic matrix D_T.

$$D_T = \begin{bmatrix} d_{T_1} \\ d_{T_2} \\ \cdots \\ d_{T_n} \end{bmatrix},$$

Equation 13

where d_{T_i} is the document vector with five topics as features (the number of topics assumed for this research as described above) for each document i of the collection of n documents. A sample matrix is shown in Table 15.

Table 15. Sample Document-Topic Matrix, D_T

	t_1	t_2	t_3	t_4	t_5
d_1	0.2314	0.1250	0.1340	0.1002	0.0956
d_2	0.1235	0.1852	0.1478	0.1085	0.1521
…	…	…	…	…	…
d_n	0.1391	0.1245	0.0891	0.1557	0.1436

4.4.4.2.2.1 Sequence Most to Least Entropic

Entropy, defined in Equation 14, measures the 'amount of disorder' of a system, or in this case, a document [71] [72], defined conceptually, ordering a set of documents starting from the most entropic to the least entropic relies on the idea that a document is more entropic when more topics are covered. The first documents are considered more entropic because they are more introductory and, therefore, cover broader topics. The later documents cover more specific topics.

$$H = -\sum_{i=1}^{N} p(x_i) \log p(x_i)$$

Equation 14

This step of sequencing using entropy relies on the topics generated in 4.4.4.2.1 and more specifically, the Document-Topic Matrix, D_T, and as illustrated in Equation 13 and Table 15.

$$D_H = \begin{bmatrix} H(X_{T1}) \\ H(X_{T2}) \\ \dots \\ H(X_{Tn}) \end{bmatrix},$$

Equation 15

The first document chosen is the most entropic (Equation 17) as measured against the topics that documents have been clustered using LDA. Once this first document has been removed, LDA is re-applied to the remaining documents. The second document is then chosen to be the most entropic of the remaining set. The process is repeated until all the documents from the first set have been consumed to produce the ordered set. In general, we have

$$d_{next} = \left\{ d_{max} \middle| \{d_{max} \in D\} \wedge \left\{ d_{max} \widehat{=} \max_{d_i}(H_i) \right\} \right\}.$$

Equation 16

Note that the generated ordered set of documents $\{d'_1, d'_2, \ldots, d'_2\}$ does not guarantee that $H_{d'_1} > H_{d'_2} > \cdots > H_{d'_n}$ because in each iteration, the LDA and entropy are recalculated amongst the remaining document peers.

4.4.4.2.2.2 Sequence Least to Most Entropic

This sequence varies from the preceding sequence generator only in that the new sequence is generated such that at each iteration, the minimum entropy document is chosen (Equation 17). Conceptually, the documents with the least entropy are more specific and therefore may serve as satisfying prerequisite information before those documents with higher entropy.

$$d_{next} = \left\{ d_{min} \middle| \{d_{min} \in D\} \wedge \left\{ d_{min} \widehat{=} \min_{d_i}(H_i) \right\} \right\}.$$

Equation 17

And similarly, here, we do not have a guarantee that $H_{d'_1} < H_{d'_2} < \cdots < H_{d'_n}$.

4.4.4.2.2.3 Sequence Most to Least KL-Divergent

The Kullback-Leibler divergence (KL divergence), also known as relative entropy, is defined as

$$KLD(p \parallel q) = \sum_{i=1}^{N} p(x_i) \log \frac{p(x_i)}{q(x_i)}$$

Equation 18

and correspondingly, for the array of document D,

$$D_{KL} = \begin{bmatrix} KLD(X_{T1}) \\ KLD(X_{T2}) \\ \dots \\ KLD(X_{Tn}) \end{bmatrix}$$

Equation 19

The KL divergence calculates the amount of information lost by approximating one distribution with another [83] [84]. In our case, we approximate with a uniform distribution q with the same number of elements as p, corresponding to the number of topics.

We follow the same procedures as described in 4.4.4.2.2.1 *Sequence Most to Least Entropic* but using the KLD formula

$$d_{next} = \left\{ d_{max} \middle| \{d_{max} \in D\}^\wedge \left\{ d_{max} \hat{=} \max_{d_i}(KLD_i) \right\} \right\}.$$

Equation 20

4.4.4.2.2.4 Sequence Least to Most KL-Divergent

Finally, we follow the same procedure as described in 4.4.4.2.2.2 *Sequence Least to Most Entropic* with the following formula:

$$d_{next} = \left\{ d_{min} \middle| \{d_{min} \in D\}^\wedge \left\{ d_{min} \hat{=} \min_{d_i}(KLD_i) \right\} \right\}.$$

Equation 21

4.4.4.2.3 Variation: Entropy Ordering Method of Summarized Documents

In their paper, “Summarization Assessment Methodology for Multiple Corpora Using Queries and Classification for Functional Evaluation,” Wolyn and Simske [41] demonstrated that summaries can be good substitutes for complete texts. We expand on this by experimenting with summaries with our analyses and comparing the results with those obtained from analyzing their complete document counterparts.

4.4.5 Metric Selection

While our generated sequences provide suggestions for ordering of corpora depending on the purpose, where possible (such as with chapters in a textbook, chapters in a dissertation, and lectures in a course), we compare them with the author’s or instructor’s (or curriculum designer’s) intent as the gold standard. However, since ‘goodness’ of order, even with comparisons to the gold standard, is at least partially subjective, we use a variety of metrics to add some objectivity. For illustration purposes, we define the following example:

$$A_1 = [1\ 2\ 3\ 4\ 5],\ B_1 = [2\ 3\ 4\ 5\ 1],\text{ and } B_2 = [5\ 4\ 1\ 2\ 3]$$

Normalized Hamming Distance (NHD). The Hamming Distance between two sequences of equal length is the number of positions that differ in value and, thus, the minimum number of substitutions needed on either sequence to make them identical to the other [85]:

$$HD(A,B) = \sum c_i\ ,$$

Equation 22

$$\text{where } c_i = \begin{cases} 0\ if\ a_i = b_i \\ 1\ if\ a_i \neq b_i \end{cases}$$

A shortcoming of this metric is it uses pairwise comparisons for each position in the two sequences, resulting in a comprehensive comparison that does not indicate how far apart they are. A small perturbation that shifts the order forward or backward by one position produces inordinate errors. For our example, $HD(A_1, B_1) = 5$, which is the same as $HD(A_1, B_2)$. Clearly, however, we can see that B_1 is a better ordering approximation of A_1 than B_2 is to A_1. We nonetheless include this metric for comparison with other metrics. For our purposes, we normalize the Hamming Distances to get a value between $[0, 1]$.

Normalized Modified Hamming Distance (NMHD). A Modified Hamming Distance was included in our research measurements to account for the shortcoming of the Hamming Distance. Our implementation takes into consideration the distances between each feature in the two sequences in question [39]:

$$MHD(A, B) = \sum |a_i - b_i|$$

Equation 23

For our example, $MHD(A_1, B_1) = 1 + 1 + 1 + 1 + 4 = 8$, but $MHD(A_1, B_2) = 4 + 2 + 2 + 2 + 2 = 12$, capturing the superiority of B_1 over B_2. As with the HD, we normalize the Modified Hamming Distance to get a value between $[0, 1]$.

Normalized Root Mean Square Error (NRMSE). The third metric we used is the Root Mean Square Error, which is a second-order version of the Normalized Hamming Distance and is defined as:

$$RMSE(A, B) = \sqrt{\frac{\sum_{i=1}^{n} (a_i - b_i)^2}{n}}$$

Equation 24

Our example yields $RMSE(A_1, B_1) = \sqrt{1 + 1 + 1 + 1 + 16} = 4.47$ and $RMSE(A_1, B_2) = \sqrt{16 + 4 + 4 + 4 + 4} = 5.66$. Again, we normalize this Root Mean Square Error to get a value between $[0, 1]$.

Normalized Mean Weighted Order Error (NMWOE). Next, we calculate the Mean Weighted Order Error [39] by linearly adjusting the weights depending on how far from the beginning of the gold standard sequence the predicted sequence is. For example, the weight vector for A_1 is $W_{A_1} = [5\ 4\ 3\ 2\ 1]$, indicating that the further the position is from the beginning of the sequence, the less significant the actual placement is. The MWOE is thus defined as:

$$MWOE(A, B) = \frac{1}{n}\sum_{i=1}^{n} w_i|a_i - b_i|,$$

Equation 25

where $w_i = n - i + 1 \quad \forall w_i \in W$ and $\mathrm{n}(W) \doteq \mathrm{n}(A)$

As with the first three metrics, the MWOE values are normalized to $[0, 1]$.

Normalized Clustering Order Error (NCOE). Finally, we define a metric that acknowledges that documents may form sub-sets within the entire set, but the ordering of those sub-sets may not matter as much as the order within those sub-sets. As a “more forgiving” version of the Levenshtein distance [86], to take this clustering or ‘chunking’ into consideration, we have:

$$COE(A, B) = \frac{1}{n-1}\sum_{i=1}^{n-1} w_i$$

where

$$w_i = \begin{cases} 1 & \text{if } b_i + 1 = b_{i+1} \\ 0 & \text{if } b_i + 1 \neq b_{i+1} \end{cases} \qquad \text{if } A = \{1, 2, \dots, n\}$$

and

$$w_i = \begin{cases} 1 & \text{if } b_i - 1 = b_{i+1} \\ 0 & \text{if } b_i - 1 \neq b_{i+1} \end{cases} \qquad \text{if } A = \{n, n-1, \dots, 1\}$$

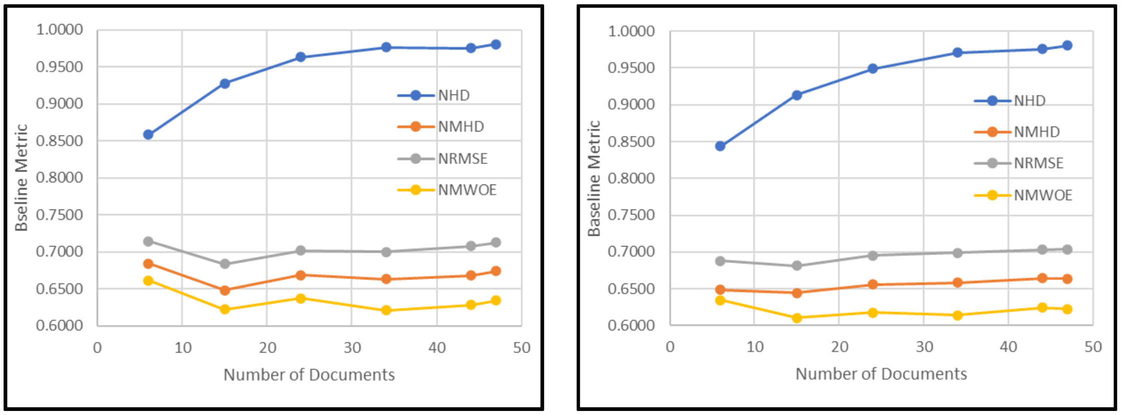
Equation 26

Using our sample-defined vectors above, we have $COE(A_1, B_1) = \frac{1}{4}(1 + 1 + 1 + 0) = 0.75$ and $COE(A_1, B_2) = \frac{1}{4}(0 + 0 + 1 + 1) = 0.5$.

Randomization. For the above metrics, we are required to have baselines to be able to compare certain ordering results. For these purposes, we generated 100 random matrices for the Similarity Matrix Method and 100 random sequences for the Entropy Sequencing Method. The results are in Table 16 and Figure 23. They illustrate the sensitivity of the Normalized Hamming Distance (NHD) to the number of documents to be ordered. The NMHD, NRMSE, and NMWOE metrics, on the other hand, tended to be consistent.

Table 16. Random Sequences Comparison Metrics for the Similarity Matrix Method and the Entropy Sequencing Method

	Similarity Matrix Method				**Entropy Sequencing Method**			
# Documents	**NHD**	**NMHD**	**NRMSE**	**NMWOE**	**NHD**	**NMHD**	**NRMSE**	**NMWOE**
6	0.8433	0.6489	0.6879	0.6345	0.8433	0.6489	0.6879	0.6345
15	0.9133	0.6444	0.6810	0.6104	0.9133	0.6444	0.6810	0.6104
24	0.9492	0.6560	0.6951	0.6180	0.9492	0.6560	0.6951	0.6180
34	0.9706	0.6581	0.6990	0.6140	0.9706	0.6581	0.6990	0.6140
44	0.9755	0.6645	0.7029	0.6245	0.9755	0.6645	0.7029	0.6245
46	0.9804	0.6639	0.7038	0.6226	0.9804	0.6639	0.7038	0.6226



(a)

(b)

Figure 23. Graphical Representation of both (a) the Similarity Matrix Method and (b) the Entropy Sequencing Method to Choose a Reasonable Number of Random Matrices and Entropy Sequences

4.5 Analysis and Results

4.5.1 Assumptions and Initial Parameter Selections

4.5.1.1 Topics Number Selection: 20% num documents

Methods have been proposed to determine the optimal number of topics fed into LDA algorithms, including [87], which includes a comprehensive evaluation using perplexity, isolation, stability, and coincidence. However, for simplicity, and based on having a small set of documents (mean < 15) on which to apply LDA, we picked 20% the number of documents as the number on which to standardize. Some variations in the results of ordering were observed, but for our purposes, not enough to necessitate a rigorous application of optimality calculations. 20% the number of documents provided heuristically similar-enough results as other numbers of topics less than the total number of documents. We also considered choosing a number of topics to equal the number of documents, but we observed that the Gensim LDA sometimes returned a maximum number of topics fewer than this specified number.

4.5.1.2 Summarizer Selection: Luhn

We chose Luhn as the extractive summarizer as it provided the best results as observed in Chapter 3 on the CNN dataset and Wolyn and Simske [41] with the same dataset applying functional analytics. Luhn is based on TF*IDF and relies on word frequency after the removal of stop words and the application of stemming and/or lemmatization. We applied only lemmatization in our case. Other available and considered summarizers include SumBasic, LexRank, Latent Semantic Analysis (LSA), TextRank, Edmundson, and Kullback–Leibler (KL) from the NLTK Sumy library.

4.5.1.3 Summary Percentage Selection: 20%

To realize the usefulness of summarization, it was important to choose a percentage that was not too large as to be too close to being the complete document and small enough but still be suitable for the summary to be a good representation of the complete text. For this parameter, we settled on 20%, a reasonable-length summary for a document. Figure 24 shows the effect of various lengths of Luhn summarization on the entropy on four of our test datasets. Each data point corresponds to the difference of entropy calculated for the full document (chapter) and the relative entropy calculated for the summary. A 100% summary indicates an extraction of the complete text and therefore results in zero difference (i.e., identical entropy).

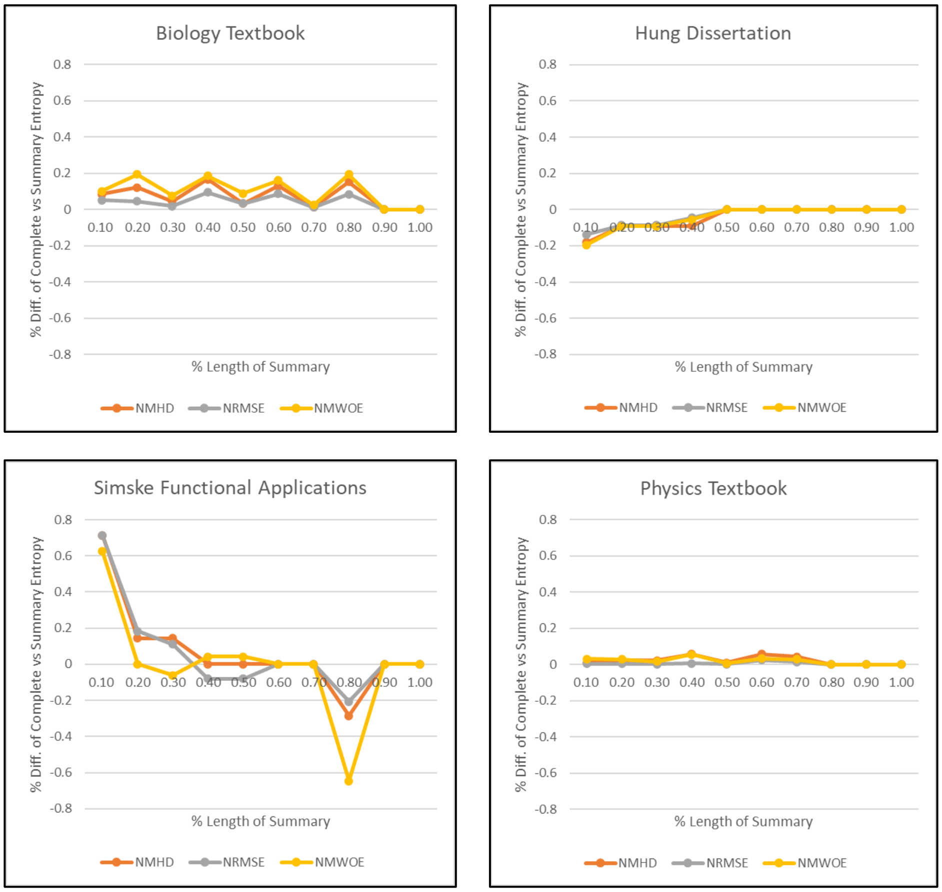


Figure 24. % Difference of Entropies vs. % Length of Summary for Four Datasets Showing that 20% Summarization is a Reasonable Representation

4.5.2 Document Sequence Ordering

To describe our findings, we first define our corpora based on the two levels. Summarized in Table 17, collections are any of the seven genres of corpora we have under analysis. Each collection is comprised of multiple documents, named depending on the genre. Biographies, novels, dissertations, and textbooks are split into chapters; the Wikipedia collection is comprised of individual articles; courses are divided into lectures; and journals are collections of papers.

Table 17. Corpus Definitions

Level 1 Corpus: Collection (Genre)	**Level 2 Corpus: Document**
Biography	Chapter
Novel	Chapter
Wikipedia	Article
Course	Lecture
Dissertation	Chapter
Journal	Paper
Textbook	Chapter

Table 18 through Table 22 show the various metrics applied to each of the ordering schemes described in 4.4.4 (Sequencing Method) applied to the full documents. In each of the metrics, there is a preponderance of the dissertation collections being sequenced by the algorithms much more effectively than random matrices and random entropies. The tables highlight, tiered by the green saturation of the cells, the lower p-values in comparison with random ordering. p-values under our threshold of 0.05 are enclosed in thickened borders for easier identification.

The journal and textbook test collections did not perform as well as dissertations but appear to perform better overall than any of the control collections (biographies, novels, and Wikipedia). The journal collection, in particular, was interesting. We did not anticipate that the order of papers to be given much thought since each paper is typically written by distinct authors and covers a topic that is not dependent on any of the other papers.

Finally, the course collection appeared to not perform any better than the control collections.

Table 23 through Table 27 provide us with similar comparisons as was performed with the complete documents but applied to their 20% Luhn summaries. The results indicate behavior that somewhat mirrored those gotten using the complete documents.

Table 18. Normalized Hamming Distance (NHD) Using Complete Documents: biographies

	Sequence Most Similar			Sequence Most Similar to Most Recent Document			Sequence Least Similar			Sequence Most to Least Entropic		Sequence Least to Most Entropic	
	Document Cosine Similarity	Document Jaccard Similarity Coefficient	Document Relative Feature Probability	Document Cosine Similarity	Document Jaccard Similarity Coefficient	Document Relative Feature Probability	Document Cosine Similarity	Document Jaccard Similarity Coefficient	Document Relative Feature Probability	Document Entropy	Document Relative Entropy	Document Entropy	Document Relative Entropy
biographies	0.228356	0.084538	0.350774	0.908754	0.497787	0.870804	0.643875	0.286121	0.691716	0.256162	0.769261	0.371098	0.821014
novels	0.590040	0.738924	0.469460	**0.026721**	0.080533	0.297326	**0.013106**	0.466437	0.140686	0.575800	0.635461	0.271837	0.618457
wikipedia	0.312741	0.889410	0.568801	**0.021324**	0.264398	0.524038	0.266696	0.760781	0.244544	0.602492	0.206627	0.825222	0.912171
courses	0.551021	0.917483	0.524137	0.362253	0.657978	0.953088	0.495507	0.853647	0.853647	0.671785	0.147572	0.112637	0.684796
dissertations	**0.034821**	**0.000137**	0.575014	0.086579	**0.000363**	0.863361	0.270982	**0.000000**	0.117417	0.141028	**0.000000**	0.256623	0.100360
journals	0.732850	0.724387	0.743743	0.787778	0.573654	0.681589	0.367072	0.626801	0.456308	0.749170	0.768452	0.378631	0.095258
textbooks	0.246814	0.649529	0.366325	0.490561	0.781579	0.474684	0.554516	0.576440	0.462811	0.808024	0.484612	0.744095	0.981504

Table 19. Normalized Modified Hamming Distance (NMHD) Using Complete Documents

	Sequence Most Similar			Sequence Most Similar to Most Recent Document			Sequence Least Similar			Sequence Most to Least Entropic		Sequence Least to Most Entropic	
	Document Cosine Similarity	Document Jaccard Similarity Coefficient	Document Relative Feature Probability	Document Cosine Similarity	Document Jaccard Similarity Coefficient	Document Relative Feature Probability	Document Cosine Similarity	Document Jaccard Similarity Coefficient	Document Relative Feature Probability	Document Entropy	Document Relative Entropy	Document Entropy	Document Relative Entropy
biographies	0.577178	0.229963	0.527912	0.844348	0.546910	0.609918	0.463006	0.794145	0.542756	0.317511	0.665922	0.813272	**0.025434**
novels	0.050010	0.767241	0.129098	0.118995	0.906349	0.555365	0.377355	0.772027	0.439216	0.921002	0.768506	0.821398	0.907359
wikipedia	0.367222	0.343003	0.351803	0.293245	0.051232	**0.037103**	0.824927	0.107483	0.844224	0.393550	0.792485	0.790497	0.730644
courses	0.919490	0.790422	0.262578	0.972144	0.651563	0.305402	0.631501	0.654018	0.468010	0.726507	0.547774	0.581131	0.835920
dissertations	0.069082	**0.000689**	0.799823	0.067731	**0.004370**	0.689430	0.457865	**0.000040**	0.361081	**0.000395**	**0.000000**	**0.020272**	**0.000010**
journals	0.253908	0.819187	0.485483	0.053389	0.911371	0.878143	0.324233	0.837441	**0.000454**	**0.013785**	0.434353	0.997746	**0.049427**
textbooks	0.240665	0.539553	0.310824	0.114284	0.462848	0.231719	0.444043	0.176097	0.079835	0.942386	0.587001	0.850544	0.891166

Table 20. Normalized Root Mean Square Error (NRMSE) Using Complete Documents

	Sequence Most Similar			Sequence Most Similar to Most Recent Document			Sequence Least Similar			Sequence Most to Least Entropic		Sequence Least to Most Entropic	
	Document Cosine Similarity	Document Jaccard Similarity Coefficient	Document Relative Feature Probability	Document Cosine Similarity	Document Jaccard Similarity Coefficient	Document Relative Feature Probability	Document Cosine Similarity	Document Jaccard Similarity Coefficient	Document Relative Feature Probability	Document Entropy	Document Relative Entropy	Document Entropy	Document Relative Entropy
biographies	0.629546	0.593452	0.583256	0.657920	0.650499	0.530071	0.401098	0.688608	0.380862	0.084535	0.944152	0.749576	0.431796
novels	**0.022363**	0.683533	0.126024	0.171244	0.888029	0.472315	0.164350	0.897380	0.350093	0.895741	0.804430	0.963095	0.823963
wikipedia	0.217307	0.250867	0.267704	0.165286	0.051389	0.109846	0.420006	0.076596	0.782539	0.753182	0.678128	0.766324	0.659689
courses	0.916245	0.921162	0.269017	0.857575	0.805199	0.248095	0.702533	0.757103	0.337917	0.796598	0.547329	0.688651	0.639868
dissertations	**0.038183**	0.079912	0.500013	**0.024958**	0.114457	0.795054	0.519246	**0.002069**	0.675743	**0.000026**	**0.000001**	**0.000060**	**0.000008**
journals	0.511914	0.735528	0.382957	0.091737	0.855715	0.770522	0.331624	0.989397	**0.001657**	**0.026938**	0.172289	0.907287	**0.006459**
textbooks	0.365759	0.411945	0.192528	0.224734	0.286839	0.290017	0.330666	0.192804	0.098408	0.782222	0.729438	0.812027	0.902739

Table 21. Normalized Mean Weighted Order Error (NMWOE) Using Complete Documents

	Sequence Most Similar			Sequence Most Similar to Most Recent Document			Sequence Least Similar			Sequence Most to Least Entropic		Sequence Least to Most Entropic	
	Document Cosine Similarity	Document Jaccard Similarity Coefficient	Document Relative Feature Probability	Document Cosine Similarity	Document Jaccard Similarity Coefficient	Document Relative Feature Probability	Document Cosine Similarity	Document Jaccard Similarity Coefficient	Document Relative Feature Probability	Document Entropy	Document Relative Entropy	Document Entropy	Document Relative Entropy
biographies	0.520535	0.131520	0.358576	0.755539	0.610948	0.553560	0.471696	0.841336	0.399909	0.147576	0.788441	0.976977	0.267344
novels	0.234209	0.713077	0.153740	0.107851	0.828441	0.707925	0.597385	0.886057	0.667136	0.695895	0.668951	0.852990	0.845456
wikipedia	0.458392	0.413507	0.069224	0.316018	0.051647	0.081763	0.794329	0.323296	0.902392	0.210286	0.627129	0.575531	0.418319
courses	0.534807	0.905641	0.489133	0.512207	0.790435	0.524558	0.464827	0.613111	0.300514	0.836410	0.663377	0.739354	0.830061
dissertations	**0.002446**	**0.001542**	0.854322	**0.003471**	**0.013677**	0.465288	**0.014562**	**0.000016**	0.135995	**0.045265**	**0.000000**	0.300952	**0.000124**
journals	0.449563	0.757333	0.717591	0.067936	0.903031	0.660533	0.501776	0.759535	**0.005140**	**0.010404**	0.238627	0.694174	**0.008224**
textbooks	0.276074	0.518869	0.197041	0.095879	0.416982	0.211701	0.496457	0.212056	0.209770	0.900966	0.802489	0.388781	0.967295

Table 22. Normalized Chunking Order Error (NCOE) Using Complete Documents

	Sequence Most Similar			Sequence Most Similar to Most Recent Document			Sequence Least Similar			Sequence Most to Least Entropic		Sequence Least to Most Entropic	
	Document Cosine Similarity	**Document Jaccard Similarity Coefficient**	**Document Relative Feature Probability**	**Document Cosine Similarity**	**Document Jaccard Similarity Coefficient**	**Document Relative Feature Probability**	**Document Cosine Similarity**	**Document Jaccard Similarity Coefficient**	**Document Relative Feature Probability**	**Document Entropy**	**Document Relative Entropy**	**Document Entropy**	**Document Relative Entropy**
biographies	**0.049335**	0.665037	**0.006233**	0.224233	0.236676	0.112889	0.558462	0.218909	0.578023	0.967360	0.553383	0.533325	0.932521
novels	0.595820	0.305372	0.647867	0.110024	**0.029919**	**0.043882**	0.392585	0.873744	0.757568	0.962240	0.524080	0.280498	0.280498
wikipedia	0.233267	**0.022088**	**0.000238**	0.050465	**0.004672**	0.200739	0.355860	0.205555	**0.018141**	0.396207	0.396125	0.218738	0.656982
courses	0.332580	0.212584	**0.004908**	0.304836	0.059078	0.173106	0.931238	0.553257	**0.001716**	0.247937	0.200896	0.453939	0.742125
dissertations	**0.001588**	**0.000026**	**0.000835**	**0.000012**	**0.000000**	**0.000000**	**0.019259**	0.128562	**0.026373**	**0.000005**	**0.000000**	0.084246	0.253126
journals	0.609809	0.731109	0.979748	0.365448	0.779815	0.371775	0.587767	0.443589	0.819172	0.759521	0.602131	0.694422	0.123096
textbooks	0.160006	0.987545	0.223400	0.098754	0.247237	0.192475	0.162837	0.492460	**0.032190**	0.851619	0.627335	0.544095	0.661581

Table 23. Normalized Hamming Distance (NHD) Using Summaries

	Sequence Most Similar			Sequence Most Similar to Most Recent Document			Sequence Least Similar			Sequence Most to Least Entropic		Sequence Least to Most Entropic	
	0.2 Luhn Summary Cosine Similarity	**0.2 Luhn Summary Jaccard Similarity**	**0.2 Luhn Summary Relative Feature**	**0.2 Luhn Summary Cosine Similarity**	**0.2 Luhn Summary Jaccard Similarity**	**0.2 Luhn Summary Relative Feature**	**0.2 Luhn Summary Cosine Similarity**	**0.2 Luhn Summary Jaccard Similarity**	**0.2 Luhn Summary Relative Feature**	**0.2 Luhn Summary Entropy**	**0.2 Luhn Summary Relative Entropy**	**0.2 Luhn Summary Entropy**	**0.2 Luhn Summary Relative Entropy**
biographies	0.878620	0.836961	0.988024	0.971836	0.593015	0.885322	0.568464	0.657124	0.894224	**0.022359**	0.638684	0.970991	0.641896
novels	0.879380	0.469460	0.415174	0.980406	0.206233	0.925464	0.265326	0.498021	0.631428	0.618695	0.378003	0.075435	0.097447
wikipedia	0.765394	0.747738	0.984109	0.232724	0.240344	0.977839	0.997488	0.107976	0.686464	**0.007914**	0.839332	0.302532	0.396132
courses	**0.047471**	0.438847	0.524137	0.475508	0.797012	0.735614	0.720048	0.155822	0.801615	**0.029801**	0.740475	0.845185	0.354822
dissertations	0.639205	**0.000962**	0.756733	**0.010198**	**0.001830**	0.961145	0.387957	**0.000723**	**0.013289**	0.155984	**0.000004**	**0.019682**	**0.005499**
journals	0.674682	0.401070	0.215598	0.469914	0.510082	0.147423	0.798679	0.437674	0.451458	0.582110	0.968790	0.762069	0.977757
textbooks	0.324939	0.687289	0.900728	0.690002	0.818773	0.461865	0.364087	0.501989	0.200708	0.577508	0.732715	0.794645	0.452393

Table 24. Normalized Modified Hamming Distance (NMHD) Using Summaries

	Sequence Most Similar			Sequence Most Similar to Most Recent Document			Sequence Least Similar			Sequence Most to Least Entropic		Sequence Least to Most Entropic	
	0.2 Luhn Summary Cosine Similarity	**0.2 Luhn Summary Jaccard Similarity**	**0.2 Luhn Summary Relative Feature**	**0.2 Luhn Summary Cosine Similarity**	**0.2 Luhn Summary Jaccard Similarity**	**0.2 Luhn Summary Relative Feature**	**0.2 Luhn Summary Cosine Similarity**	**0.2 Luhn Summary Jaccard Similarity**	**0.2 Luhn Summary Relative Feature**	**0.2 Luhn Summary Entropy**	**0.2 Luhn Summary Relative Entropy**	**0.2 Luhn Summary Entropy**	**0.2 Luhn Summary Relative Entropy**
biographies	0.721221	0.586343	0.975201	0.897959	**0.010668**	0.973446	0.472357	0.885947	0.685698	**0.009736**	0.639743	0.907427	0.080054
novels	0.582587	0.683917	0.674639	0.126923	0.118924	0.128866	0.781686	0.200204	0.560732	0.788122	0.775337	0.878769	0.825104
wikipedia	0.825073	0.057249	0.934773	0.518708	0.078973	0.202960	0.909645	0.448156	0.975874	0.900361	0.555280	0.789894	0.466545
courses	0.999153	0.626424	0.245697	0.938251	0.543558	0.106044	0.708308	0.894809	0.359208	0.430468	0.662670	0.637829	0.655176
dissertations	0.525654	**0.024735**	0.484026	0.213903	0.117246	0.233654	0.119689	0.109529	0.066725	**0.002381**	**0.000017**	0.354755	**0.000020**
journals	0.594038	0.543224	0.448912	**0.000563**	0.750873	0.997694	**0.005629**	0.782818	**0.018113**	0.625335	0.854827	0.392640	0.337213
textbooks	0.133551	0.426099	0.106180	0.179806	0.778926	0.145418	0.912203	0.095709	**0.017267**	0.881626	0.734900	0.794727	0.369737

Table 25. Normalized Root Mean Square Error (NRMSE) Using Summaries

	Sequence Most Similar			Sequence Most Similar to Most Recent Document			Sequence Least Similar			Sequence Most to Least Entropic		Sequence Least to Most Entropic	
	0.2 Luhn Summary Cosine Similarity	**0.2 Luhn Summary Jaccard Similarity**	**0.2 Luhn Summary Relative Feature**	**0.2 Luhn Summary Cosine Similarity**	**0.2 Luhn Summary Jaccard Similarity**	**0.2 Luhn Summary Relative Feature**	**0.2 Luhn Summary Cosine Similarity**	**0.2 Luhn Summary Jaccard Similarity**	**0.2 Luhn Summary Relative Feature**	**0.2 Luhn Summary Entropy**	**0.2 Luhn Summary Relative Entropy**	**0.2 Luhn Summary Entropy**	**0.2 Luhn Summary Relative Entropy**
biographies	0.785417	0.869559	0.967650	0.861476	0.260865	0.611780	0.467467	0.441759	0.468163	0.051874	0.998514	0.814924	0.189585
novels	0.575726	0.530386	0.638932	0.155057	0.057842	0.114418	0.977514	0.492293	0.914711	0.653310	0.681910	0.904659	0.848685
wikipedia	0.803316	**0.020045**	0.996358	0.366598	0.052074	0.273325	0.810176	0.394955	0.734786	0.825215	0.572901	0.653748	0.449798
courses	0.751233	0.847236	0.285134	0.684702	0.608057	0.137415	0.714386	0.937891	0.326870	0.695654	0.530863	0.715642	0.543906
dissertations	0.629176	0.388466	0.364827	0.627675	0.517977	0.142285	0.215263	0.329027	0.192889	**0.000036**	**0.000245**	**0.031539**	**0.000156**
journals	0.285996	0.764831	0.633055	**0.023548**	0.750807	0.783101	**0.003648**	0.874690	0.063661	0.597804	0.712747	0.798161	0.750693
textbooks	0.128349	0.294633	0.064532	0.173001	0.398213	0.121787	0.869191	0.101932	**0.031210**	0.989664	0.731015	0.855912	0.814925

Table 26. Normalized Mean Weighted Order Error (NMWOE) Using Summaries

	Sequence Most Similar			Sequence Most Similar to Most Recent Document			Sequence Least Similar			Sequence Most to Least Entropic		Sequence Least to Most Entropic	
	0.2 Luhn Summary Cosine Similarity	**0.2 Luhn Summary Jaccard Similarity**	**0.2 Luhn Summary Relative Feature**	**0.2 Luhn Summary Cosine Similarity**	**0.2 Luhn Summary Jaccard Similarity**	**0.2 Luhn Summary Relative Feature**	**0.2 Luhn Summary Cosine Similarity**	**0.2 Luhn Summary Jaccard Similarity**	**0.2 Luhn Summary Relative Feature**	**0.2 Luhn Summary Entropy**	**0.2 Luhn Summary Relative Entropy**	**0.2 Luhn Summary Entropy**	**0.2 Luhn Summary Relative Entropy**
biographies	0.901756	0.327847	0.574296	0.961695	**0.009398**	0.910288	0.509936	0.625609	0.518414	**0.003668**	0.531164	0.991973	0.139054
novels	0.937086	0.641853	0.539971	0.354887	0.202449	0.428176	0.958824	0.193855	0.793131	0.531997	0.809782	0.770972	0.736612
wikipedia	0.873789	0.102692	0.663987	0.974825	0.111181	0.274073	0.715512	0.889525	0.997698	0.795587	0.692361	0.826526	0.604108
courses	0.839984	0.684733	0.423926	0.780187	0.619645	0.316626	0.459726	0.859683	0.117283	0.568753	0.607987	0.736786	0.662181
dissertations	0.294352	**0.040376**	0.181801	0.086314	0.202926	0.196338	**0.027014**	**0.015424**	**0.010484**	0.108029	**0.000397**	0.725805	**0.000416**
journals	0.614625	0.417016	0.327998	**0.019466**	0.826411	0.834138	**0.012418**	0.871094	**0.036306**	0.594913	0.435329	0.114845	0.527728
textbooks	0.089379	0.460573	0.088094	0.144979	0.743817	0.078046	0.824229	0.121796	**0.039998**	0.765711	0.880805	0.358245	0.178997

Table 27. Normalized Chunking Error (NCOE) Using Summaries

	Sequence Most Similar			**Sequence Most Similar to Most Recent Document**			**Sequence Least Similar**			**Sequence Most to Least Entropic**		**Sequence Least to Most Entropic**	
	0.2 Luhn Summary Cosine Similarity	**0.2 Luhn Summary Jaccard Similarity**	**0.2 Luhn Summary Relative Feature**	**0.2 Luhn Summary Cosine Similarity**	**0.2 Luhn Summary Jaccard Similarity**	**0.2 Luhn Summary Relative Feature**	**0.2 Luhn Summary Cosine Similarity**	**0.2 Luhn Summary Jaccard Similarity**	**0.2 Luhn Summary Relative Feature**	**0.2 Luhn Summary Entropy**	**0.2 Luhn Summary Relative Entropy**	**0.2 Luhn Summary Entropy**	**0.2 Luhn Summary Relative Entropy**
biographies	0.159368	0.219156	0.159368	0.220029	0.242024	**0.007630**	0.578023	**0.011191**	0.966244	0.289127	0.464743	0.750348	0.828576
novels	0.561523	0.668506	**0.044186**	0.151366	0.063304	0.187955	0.204422	0.543899	0.537511	0.315999	0.820426	0.546991	0.960021
wikipedia	**0.034906**	0.288369	0.232869	**0.000923**	0.078506	0.119946	0.089860	0.050071	0.633398	0.313696	0.772402	0.597734	0.255847
courses	0.371926	0.178589	**0.031249**	0.141829	0.570075	**0.045856**	0.156249	0.432276	**0.025880**	0.480473	0.524509	0.894819	0.562668
dissertations	**0.034149**	**0.000008**	**0.000027**	**0.000700**	**0.000000**	**0.000000**	**0.004225**	0.153878	0.153553	**0.004777**	**0.000000**	**0.000014**	0.902559
journals	0.722225	0.668402	0.491036	0.297695	0.770598	0.253137	0.704238	0.992243	0.834821	0.463024	0.327784	0.590369	0.534577
textbooks	0.282720	0.758413	0.238230	0.091861	0.050987	**0.009411**	**0.032190**	0.757809	0.597434	0.728625	0.473357	0.664821	0.881501

To better visualize comparative differences and similarities, a next-level roll-up of the metrics gathered from these tables provides a more concise and clearer comparison. We capture these results by summarizing the results in Table 28 and Table 29. Here, we observe a strong indication that of all the different collection genres, dissertation document sequences are best predicted by just about any of our five metrics, whether applied to the documents in their entirety (Table 18 – Table 22) or applied to their summaries (Table 23 – Table 27). Table 28 shows the percentage of p-values less than 0.05 and Table 29 shows the mean of the p-values.

Table 28. Percentage of p-values > 0.05

	Full Documents					Summaries				
	NHD	NMHD	NRMSE	NMWOE	NCOE	NHD	NMHD	NRMSE	NMWOE	NCOE
biographies	1.000	0.923	1.000	1.000	0.846	0.923	0.846	1.000	0.846	0.846
novels	0.846	1.000	0.923	1.000	0.846	1.000	1.000	1.000	1.000	0.923
wikipedia	0.923	0.923	1.000	1.000	0.692	0.923	1.000	0.923	1.000	0.846
courses	1.000	1.000	1.000	1.000	0.846	0.846	1.000	1.000	1.000	0.769
dissertations	0.615	0.462	0.462	0.308	0.231	0.385	0.692	0.692	0.538	0.231
journals	1.000	0.769	0.769	0.769	1.000	1.000	0.769	0.846	0.769	1.000
textbooks	1.000	1.000	1.000	1.000	0.923	1.000	0.923	0.923	0.923	0.846

Table 29. Mean of p-values

	Full Documents					Summaries				
	NHD	NMHD	NRMSE	NMWOE	NCOE	NHD	NMHD	NRMSE	NMWOE	NCOE
biographies	0.522	0.535	0.563	0.525	0.434	0.734	0.604	0.599	0.539	0.377
novels	0.379	0.580	0.559	0.612	0.446	0.495	0.548	0.580	0.608	0.431
wikipedia	0.492	0.456	0.400	0.403	0.212	0.560	0.589	0.535	0.656	0.267
courses	0.599	0.642	0.653	0.631	0.324	0.513	0.601	0.598	0.591	0.340
dissertations	0.188	0.190	0.212	0.141	0.040	0.227	0.173	0.265	0.145	0.096
journals	0.591	0.466	0.445	0.444	0.605	0.569	0.489	0.542	0.433	0.588
textbooks	0.586	0.452	0.432	0.438	0.406	0.578	0.429	0.429	0.367	0.428

4.5.3 Effect of Summarization

Another observation is that summaries provide reasonable stand-ins for their complete document counterparts, a verification of the work by Wolyn and Simske [41]. We note here that generally, the test collections were better approximated by their summaries for the purposes of ordering compared to the those of the test collections (Table 30).

Table 30. Difference Between p-values Using Full Documents and Summaries

	Summaries				
	NHD	**NMHD**	**NRMSE**	**NMWOE**	**NCOE**
biographies	-0.213	-0.068	-0.036	-0.014	0.057
novels	-0.117	0.031	-0.022	0.005	0.015
wikipedia	-0.068	-0.134	-0.135	-0.252	-0.055
courses	0.086	0.041	0.054	0.041	-0.015
dissertations	-0.039	0.017	-0.053	-0.004	-0.057
journals	0.022	-0.023	-0.097	0.011	0.017
textbooks	0.009	0.023	0.004	0.071	-0.022

Furthermore, close inspection shows a tight correlation of results as indicated by our whisker plots in Figure 25. Consolidated distributions of differences in p-values in Figure 26 indicate tightness that approximates a normal distribution in the case of Figure 26-(c) and Figure 26-(d). Figure 26-(a), Figure 26-(b), and Figure 26-(e) approximate the upper half of a sinc function.

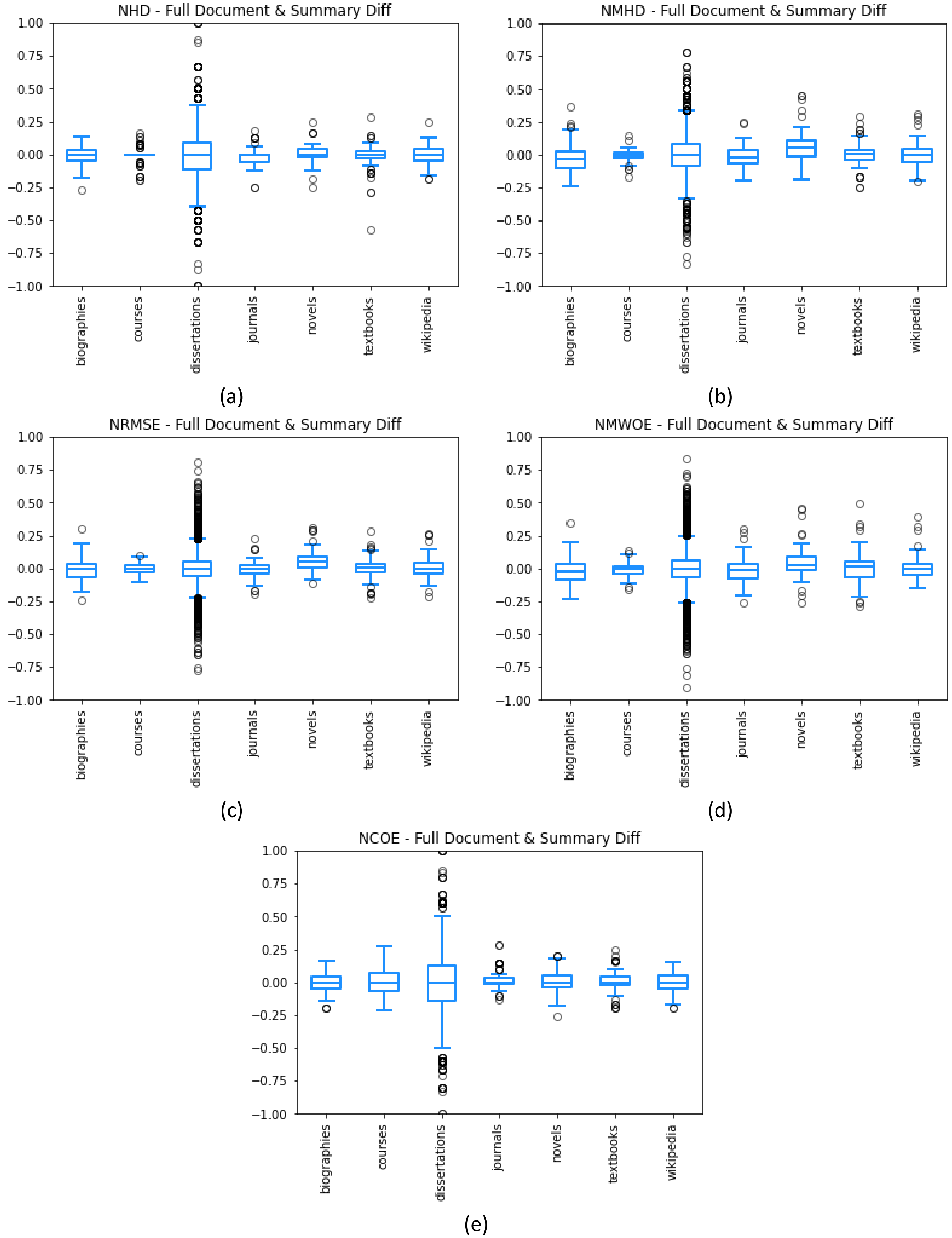
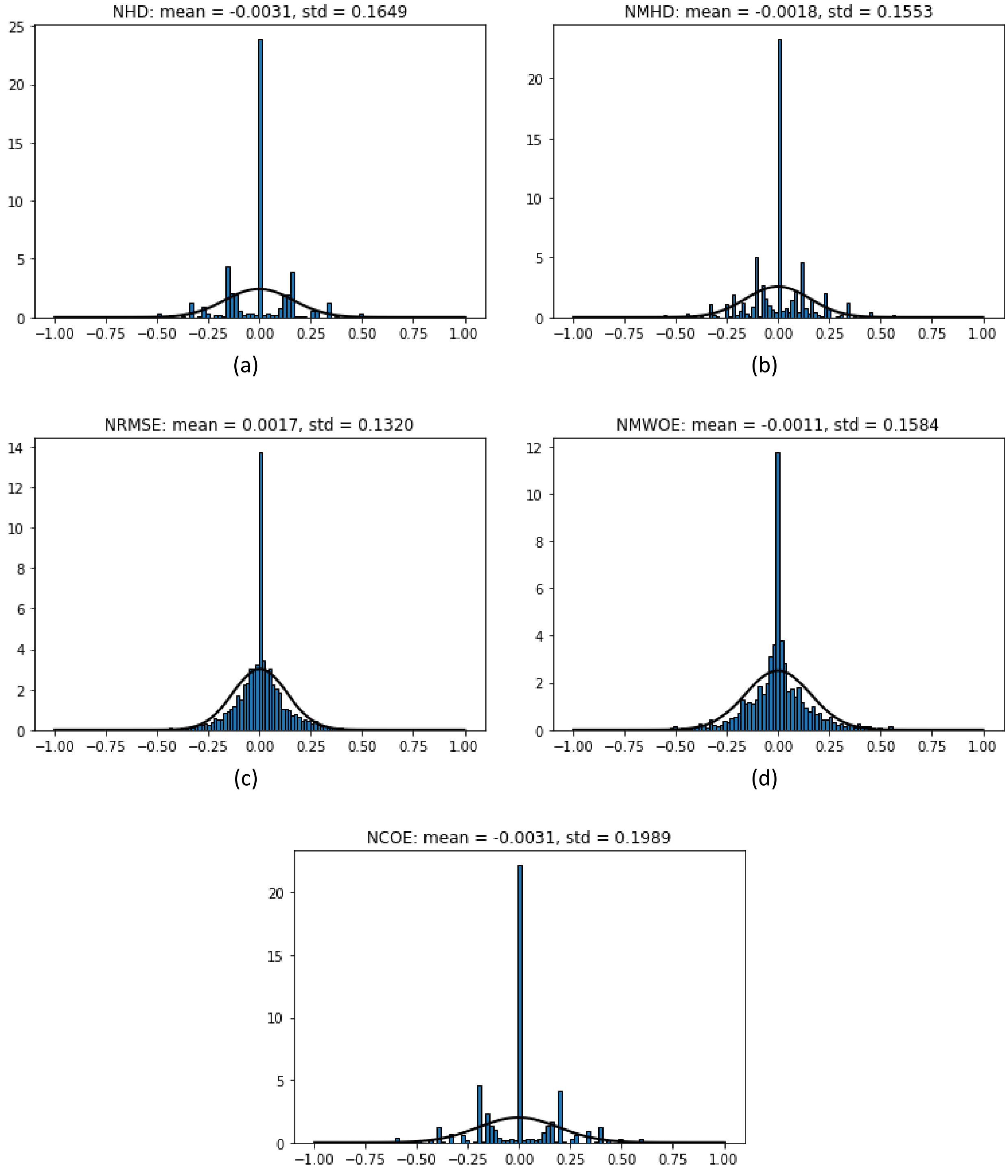


Figure 25. Distribution of Differences Between Complete Document p-values and Summary p-values by Collection



(a)

(b)

(c)

(d)

(e)

Figure 26. Distribution of Consolidated p-values of Full Documents minus p-values of Summary Documents

4.5.4 Analysis of Various Departmental Dissertations

Compared to the other genres, because of their availability and accessibility from Colorado State University, we have collected more dissertations than all the others combined. But because many departments are represented (we have 32 that had five or more dissertations that had six or more chapters), we were also interested in seeing if predictability varied across them. We used ANOVA tests for all the sequencing schemes (including complete documents and summaries) to quantify this. We investigated measurements with all our metrics and determined that, indeed, there were variations. Rather than showing all the results, we present a sample set of results for the metric NMWOE for *Sequence Most to Least Entropic* using *Document Relative Entropy* with only the pairs in which the p-value ≤ 0.5.

Table 31. Sample ANOVA Test Results: NMWOE, Sequence Most to Least Entropic, Document Relative Entropy; *p-value: 0.314339*

Department 1	Department 2	p-value
Agricultural Biology and Economics	Atmospheric Science	0.010100
Agricultural Biology and Economics	Biochemistry _ Molecular Biology	0.016900
Agricultural Biology and Economics	Environmental and Radiological Health Sciences	0.047600
Agricultural Biology and Economics	Human Dimensions of Natural Resources	0.046200
Atmospheric Science	Computer Science	0.004200
Atmospheric Science	Economics	0.019500
Atmospheric Science	Food Science and Human Nutrition	0.047800
Atmospheric Science	Mathematics	0.024000
Atmospheric Science	Statistics	0.014200
Biochemistry _ Molecular Biology	Computer Science	0.006700
Biochemistry _ Molecular Biology	Economics	0.029800
Biochemistry _ Molecular Biology	Mathematics	0.034500
Biochemistry _ Molecular Biology	Statistics	0.022300
Civil and Environmental Engineering	Computer Science	0.019000
Civil and Environmental Engineering	Statistics	0.044400
Civil and Environmental Engineering	Systems Engineering	0.048500
Computer Science	Environmental and Radiological Health Sciences	0.021800
Computer Science	Forest and Rangeland Stewardship	0.021900
Computer Science	Human Dimensions of Natural Resources	0.016200
Computer Science	Physics	0.045400
Computer Science	Political Science	0.031100
Economics	Human Dimensions of Natural Resources	0.048500
Human Dimensions of Natural Resources	Statistics	0.037400

Using the Bonferroni Correction [88] and calculating the t-statistic, the p-values are sorted and summarized in Table 32. For this set of results, we note that Computer Science dissertations are most different from the rest of the departments.

Table 32. Sorted t-static p-values after a Bonferroni Correction based on Table 31

Department	Mean p-value Compared to Other Departments
Computer Science	0.235975
Civil and Environmental Engineering	0.250743
Atmospheric Science	0.267396
Human Dimensions of Natural Resources	0.278377
Biochemistry _ Molecular Biology	0.295775
Agricultural Biology and Economics	0.296260
Forest and Rangeland Stewardship	0.384956
Environmental and Radiological Health Sciences	0.394341
Political Science	0.408809
Statistics	0.432523
Mathematics	0.456586
Economics	0.458738
Physics	0.472531
Biomedical Sciences	0.514174
Systems Engineering	0.535665
Chemistry	0.550397
Mechanical Engineering	0.568558
Food Science and Human Nutrition	0.571663
Journalism _ Media Communication	0.579678
Geosciences	0.581272
Electrical and Computer Engineering	0.586312
Biology	0.590579
Soil _ Crop Sciences	0.599033
Animal Sciences	0.606219
Psychology	0.610589
Chemical and Biological Engineering	0.614384
Clinical Sciences	0.615356
Horticulture _ Landscape Architecture	0.617040
Health and Exercise Science	0.621763
Sociology	0.629826
Microbiology, Immunology, and Pathology	0.637283
Fish, Wildlife, and Conservation Biology	0.656991

4.5.5 Limitations

In Section 4.4.4.2.2.3 Sequence Most to Least KL-Divergent, we made an assumption that there was topic balance, and therefore used a uniform distribution q for comparison. It is quite possible that the topic distributions followed a normal, Zipf, or some other distribution.

4.6 Conclusions

4.6.1 Predicting Existing Order

Employing an extensive comparison of results using educational material control document sets and multiple metrics, we have demonstrated through this research that we can predict the existing order of these documents (i.e., the order intended by the author or editor) better than randomly (and in some instances much better than randomly). As expected, predicting the orders of control documents within the sets (biographies, novels, and randomly picked Wikipedia articles on a broad subject) were not successful using our methods, but predicting the orders of the test documents (courses, dissertations, journals, and textbooks) were successful, especially with dissertations.

Perhaps most surprising to the author, journal article order could be somewhat predicted, indicating that journal editors do not simply publish articles in an issue in random order. The results of our experiments on textbook chapters can be thought of as mirroring what textbook ordering should be. Though an explicit ordering of chapters from Chapter 1 through Chapter n is often implied, chapters can be consumed in random fashion. After all, the author does not remember when he last read a textbook from cover to cover. Perhaps the same can be said of a university course, which often mirrors the chapters in a textbook.

4.6.2 Proposing Order for Comprehension

As an extension to Section 4.6.1, if order was not correctly predicted, we submit that it is feasible that more suitable orders exist. Therefore, our algorithms could be used to generate proper sequences automatically. For example, we would use the *Sequence Most to Least Entropic* (Section 4.4.4.2.2.1) algorithm to order reading or present a curriculum based on a textbook or series of textbooks by acknowledging that more generic materials (i.e., materials with more concepts covered) are prerequisites to more in-depth topics. For example, an introductory book on quantum computing may cover qubits, a brief history of quantum mechanics, post-quantum cryptography, quantum supremacy, and Cuda, while a more advanced book may simply cover the mathematics of post-quantum cryptography but in depth.

4.6.3 Understanding the Results of Summarization

Though it didn't come much as a surprise, as extensive work by Simske [39] on summarization and text analytics has demonstrated the effectiveness of functional analytics, it is of note that our work in document ordering is also applicable using summarized documents. At least for the fixed hyper-parameters that we have chosen in Section 4.5.1, we settled on the number of topics at 20% of the number of documents, using Luhn as the extractive summarizer of choice, and limiting the size of the summary to 20% of the length (number of sentences) of the full document.

4.7 Further Experimentation and Research

Our research, while extensive, is but a potential launch point for many avenues. While certain hyper-parametric assumptions were made with reasonable justifications, they are not substitutes for extensive research and experimentation on the ideal parameters. The following are some areas in which additional work is warranted:

- Using different summarizers. We used Luhn as a summarizer based on the results obtained in Chapter 3. However, because Chapter 3 worked on news articles (an entirely different genre of text from the ones for this section), using Luhn may not have been optimal. It behooves us to try other various extractive summarizers available and even use abstractive summarizers for comparison. In addition, since Chapter 3 showed improvement beyond single summarizers, a meta-algorithmic summarizer such as the one we developed using Tessellation and Recombination with Expert Decisioner may provide superior results.

- Optimizing the number of topics. There are well-known methods of picking the optimal number of topics for K-means clustering. While we settled on a number that was 20% the size of the number of documents, the ideal number would depend on the text itself and would likely be different depending on the genre and contents of the text.
- Optimizing summary percentage. Here, we reasoned that summaries that were about 20% of the full text were reasonable. The ideal percentage may be different.
- Instead of only lecture slides, add the transcripts. When we originally conceptualized using course lecture slides, we desired to include the professor's transcribed lecture as part of the lecture slides. Unfortunately, those were not readily available, and we settled for just the lecture slides.
- Ordering song tracks. While CDs are now mostly relegated to the Smithsonian, artists still do release most of their songs as parts of "albums." It would be interesting to see if ordering song tracks in an album using lyrics and/or chords would be feasible. Artists can be particular about these sequences [89].
- Using Predictive Selection meta-algorithm for Section 4.5.4. Since we noticed differences among department dissertations, we could use Predictive Selection to "pre-classify" the dissertations by department and run the best (for that class) reading order approach to detect if the overall system reading order accuracy improves.

5 Summary and Conclusion

In all our experiments, from classification to summarization to document ordering, we have demonstrated that even proven and older algorithms can be improved with systems thinking [1] and systems engineering approaches. The Gang of Four's publication, *Design Patterns: Elements of Reusable Object-Oriented Software* [90], became a must-read for any software engineer wanting to take advantage of proven arrangements of software. Simske's publications, including *Meta-algorithmics: Patterns for Robust, Low Cost, High Quality Systems* [2], *Meta-analytics: Consensus Approaches and System Patterns for Data Analysis* [91], and *Functional Applications of Text Analytics Systems* [39], apply systems engineering methods to analytics and machine learning, and provide proven patterns for component algorithms integration. This dissertation takes concepts from Simske's publications and lectures and applies them to real-world problems involving natural language processing, covered in Chapters 2, 3, and 4 above.

For a classification problem, we decided to apply a "meta-" approach – applying systems engineering to help solve a systems engineering problem – requirements engineering and classification. Focused on existing models and using several first-order meta-algorithms as patterns, we provided improvement and consistency in results compared to those obtained from simply using the component algorithms by themselves.

For summarization, we took existing, well-known models, and used a second-order meta-algorithmic pattern called Tessellation and Recombination with Expert Decisioner to develop a hybrid model that resulted in higher-quality summaries. These results could be applied to generating better, more accurate summaries for news articles and other documents. Within the system engineering community, summarizing documents such as SOWs, PWSs, RFPs, RFIs, and responses to them would be useful in speeding up the acquisition process.

And finally, the document ordering methods we investigated are useful not just for developing curricula, but also for evaluating existing ones and providing recommended reading orders. In the context of systems engineering, our document ordering methods can be used for sequencing training material whether for general knowledge, for specific systems, or other educational resources.

What we have demonstrated here is but a minuscule subset of what is possible. With the unprecedented adoption of ChatGPT (and Microsoft Bing) as this dissertation is being completed and published, competition from Google, Meta, and others is heating up. Yet, meta-algorithmic and functional methods don’t care which ones win out. The results from integrating a subset of these heavy hitters with others will prove to take advantage of all of them. Without any presumption of the impact of our work, I am reminded of Sir Isaac Newton’s note to Robert Hooke in 1676, “If I have seen further, it is by standing on the shoulders of giants.”

References

[1] D. H. Meadows, Thinking in Systems: A Primer, Chelsea Green Publishing, 2008.

[2] S. J. Simske, Meta-algorithmics: Patterns for Robust, Low Cost, High Quality Systems, New York, NY: John Wiley & Sons, Ltd., 2013.

[3] IEEE, "IEEE Standard 1220-2019: Standard for Application and Management of the Systems Engineering Process," IEEE, 2019.

[4] J. Dick, E. Hull and K. Jackson, Requirements Engineering, London: Springer International Publishing AG, 2017.

[5] A. Sainani, P. R. Anush, V. Joshi and S. Ghaisas, "Extracting and Classifying Requirements from Software Engineering Contracts," in *2020 IEEE 28th International Requirements Engineering Conference*, 2020.

[6] C. Jones, Software Assessments, Benchmarks, and Best Practices, New Jersey: Addison Wesley Longman, Inc., 2000.

[7] INCOSE, INCOSE Systems Engineering Handbook : A Guide for System Life Cycle Processes and Activities, San Diego, CA: John Wiley & Sons, Inc., 2015.

[8] M. Glinz, "CPRE Glossary 2.0," 1 October 2020. [Online]. Available: www.ireb.org/en/downloads/. [Accessed 19 November 2021].

[9] R. Hefner, *Requirements Management Tutorial,* San Diego: Caltech, 2019.

[10] P. Clements, F. Bachmann, L. Bass, D. Garlan, J. Ivers, R. Little, R. Nord and J. Stafford, Documenting Software Architectures: Views and Beyond, Boston: Pearson Education, Inc., 2003.

[11] N. Rozanski and E. Woods, Software Systems Architecture: Working With Stakeholders Using Viewpoints and Perspectives, 2nd Edition, Upper Saddle River, NJ: Addison-Wesley Professional, 2011.

[12] INCOSE, Guide for Writing Requirements, San Diego, CA: INCOSE, 2017.

[13] DAU, "Defense Acquisition Guidebook," [Online]. Available: www.dau.edu. [Accessed August 11 2021].

[14] Black's Law Dictionary Free Online Legal Dictionary 2nd Ed., "What is SHALL?," [Online]. Available: https://thelawdictionary.org/shall/. [Accessed 4 February 2022].

[15] B. P. Douglass, Agile Systems Engineering, Waltham, MA: Elsevier, Inc., 2016.

[16] "ISO/IEC/IEEE International Standard - Systems and software engineering -- Life cycle processes -- Requirements engineering," *ISO/IEC/IEEE 29148:2018(E),* 2018.

[17] NASA, "Appendix C: How to Write a Good Requirement," [Online]. Available: www.nasa.gov/seh/appendix-c-how-to-write-a-good-requirement. [Accessed 11 November 2021].

[18] INCOSE, Needs, Requirements, Verification, Validation Lifecycle Manual, San Diego, CA: INCOSE, 2022.

[19] E. D. Canedo and B. C. Mendes, "Software Requirements Classification Using Machine Learning Algorithms," *Entropy,* 2020.

[20] A. Mahmoud and G. Williams, "Detecting, classifying, and tracing non-functional software requirements," *Requirements Engineering,* vol. 21, no. 3, pp. 357-381, 2016.

[21] Z. S. H. Abad, O. Karras, P. Ghazi, M. Glinz, G. Ruhe and K. Schneider, "What Works Better? A Study of Classifying Requirements," in *2017 IEEE 25th International Requirements Engineering Conference*, 2017.

[22] M. Sabir, C. Chrysoulas and E. Banissi, "Multi-label Classifier to Deal with Misclassification in Non-functional Requirements," *International Journal of Information Technology,* vol. 12, no. 1, pp. 101-110, 2020.

[23] D. Giannakopoulou, T. Pressburger, A. Mavridou and J. Schumann, "Generation of Formal Requirements from Structured Natural Language," in *Proceedings of the 26th International Working Conference on Requirements Engineering: Foundation for Software Quality (REFSQ)*, Pisa, Italy, 2020.

[24] L. Lucio, S. Rahman, C.-H. Cheng and A. Mavin, "Just Formal Enough? Automated Analysis of EARS Requirements," in *NASA Formal Methods*, Moffet Field, CA, 2017.

[25] J. Ladyman and K. Wiesner, What is a Complex System?, New Haven, CT: Yale University Press, 2020.

[26] A. Kossiakoff, S. M. Biemer, S. J. Seymour and D. A. Flanigan, Systems Engineering Principles and Practice, 3rd Edition, Hoboken, NJ: Wiley, 2020.

[27] G. Kotonya and I. Sommerville, Requirements Engineering: Processes and Techniques, Chichester: John Wiley & Sons, Inc, 1998.

[28] A. Mavin, P. Wilkinson, A. Harwood and M. Novak, "EARS (Easy Approach to Requirements Syntax)," in *RE09*, Atlanta, Georgia, 2009.

[29] J. Cleland-Huang, "Exploración de base de datos de Atributos de Calidad," 2007. [Online]. Available: github.com/Manolomon/tera-promise. [Accessed 19 November 2021].

[30] J. Pennington, R. Socher and C. D. Manning, "GloVe: Global Vectors for Word Representation," 2014. [Online]. Available: nlp.stanford.edu/projects/glove. [Accessed 15 September 2021].

[31] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot and E. Duchesnay, "Scikit-learn: Machine Learning in {P}ython," *Journal of Machine Learning Research,* vol. 12, pp. 2825-2830, 2011.

[32] S. Bird, E. Loper and E. Klein, Natural Language Processing with Python, Sebastapol, CA: O'Reilly Media Inc., 2009.

[33] H. Lane, C. Howard and H. M. Hapke, Natural Language Processing in Action, Shelter Island, NY: Manning Publications Co., 2019.

[34] R. Parker, D. Graff, J. Kong, C. Ke and M. Kazuaki, "English Gigaword Fifth Edition," LDC: Linguistic Data Consortium, 17 June 2011. [Online]. Available: catalog.ldc.upenn.edu/LDC2011T07. [Accessed 15 September 2021].

[35] X. Lin, S. Yacoub, J. Burns and S. Simske, "Performance analysis of pattern classifier combination by plurality voting," *Pattern Recognition Letters,* vol. 24, no. 12, pp. 1959-1969, 2003.

[36] H. Zhang, "The Optimality of Naive Bayes," in *FLAIRS Conference*, Miami Beach, Florida, 2004.

[37] O. Kupervasser, "The Mysterious Optimality of Naive Bayes: Estimation of the Probability in the System of “Classifiers”," *Mathematical Theory of Pattern Recognition,* vol. 24, no. 1, pp. 1-10, 2014.

[38] R. J. Moore, "Eric Schmidt’s “5 Exabytes” Quote is a Load of Crap," 7 February 2011. [Online]. Available: https://blog.rjmetrics.com/2011/02/07/eric-schmidts-5-exabytes-quote-is-a-load-of-crap/. [Accessed 5 June 2022].

[39] S. Simske and M. Vans, Functional Applications of Text Analytics Systems, Gistrup: River Publishers, 2021.

[40] P. Mehta, "From Extractive to Abstractive Summarization: A Journey," in *Proceedings of the ACL 2016 Student Research Workshop*, Berlin, Germany, 2016.

[41] S. Wolyn and S. J. Simske, "Summarization Assessment Methodology for Multiple Corpora Using Queries and Classification for Functional Evaluation," 2022.

[42] V. Gupta and G. S. Lehal, "A Survey of Text Summarization Extractive Techniques," *Journal of Emerging Technologies in Web Intelligence,* vol. 2, no. 3, pp. 258-268, 2010.

[43] S. Gupta and S. K. Gupta, "Abstractive summarization: An overview of the state of the art," *Expert Systems With Applications,* vol. 121, pp. 49-65, 2019.

[44] P. Nathan, "PyTextRank v3.2.3," [Online]. Available: https://spacy.io/universe/project/spacy-pytextrank/. [Accessed 16 June 2022].

[45] R. Mihalcea and P. Tarau, "TextRank: Bringing Order into Texts," in *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing*, Barcelona, Spain, 2004.

[46] R. Rehurek, "summarization.summarizer - TextRank Summariser," [Online]. Available: https://radimrehurek.com/gensim_3.8.3/summarization/summariser.html. [Accessed 16 June 2022].

[47] "Automatic Summarization Library: pysummarization," [Online]. Available: https://pypi.org/project/pysummarization/. [Accessed 16 June 2022].

[48] Stanford NLP Group, "CoreNLP," [Online]. Available: https://stanfordnlp.github.io/CoreNLP/. [Accessed 10 June 2022].

[49] NLTK Project, "NLTK," [Online]. Available: https://www.nltk.org/. [Accessed 1 April 2022].

[50] M. S. Binwahlan, N. Salim and L. Suanmali, "Swarm Based Text Summarization," in *International Association of Computer Science and Information Technology - Spring Conference*, Singapore, 2009.

[51] Microsoft, "Changes in Word 2010 (for IT pros)," Microsoft, July 22 2014. [Online]. Available: https://docs.microsoft.com/en-us/previous-versions/office/office-2010/cc179199(v=office.14). [Accessed 3 July 2022].

[52] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu and D. Warde-Farley, "Generative Adversarial Nets," *Proceedings of the International Conference on Neural Information Processing Systems,* pp. 2672-2680, 2014.

[53] R. Bhargava, G. Sharma and Y. Sharma, "Deep Text Summarization using Generative Adversarial Networks in Indian Languages," in *International Conference on Computational Intelligence and Data Science (ICCIDS)*, 2019.

[54] K. Papineni, S. Roukos, T. Ward and W.-J. Zhu, "Bleu: a method for automatic evaluation of machine translation," *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics,* 2002.

[55] M. Post, "A Call for Clarity in Reporting BLEU Scores," *Proceedings of the Third Conference on Machine Translation,* 2018.

[56] C.-Y. Lin, "ROUGE: A Package for Automatic Evaluation of Summaries," *Text summarization branches out,* pp. 74-81, 2004.

[57] L. Ermakova, J. V. Cossu and J. Mothe, "A survey on evaluation of summarization methods," *Information Processing and Management,* vol. 56, pp. 1794-1814, 2019.

[58] R. Ferreira, L. d. S. Cabral, R. D. Lins, G. Pereira e Silva, F. Freitas, G. D. Cavalcanti, R. Lima, S. J. Simske and L. Favaro, "Assessing sentence scoring techniques for extractive text summarization," *Expert Systems with Applications,* vol. 40, no. 13, pp. 5755-5764, 2013.

[59] M. Hart, "The Project Gutenberg Mission Statement," 20 June 2004. [Online]. Available: www.gutenberg.org/about/background/mission_statement.html. [Accessed 4 June 2022].

[60] R. Lins, H. Oliveira, L. Cabral, J. Batista, B. Tenorio, R. Ferreira, R. Lima, G. de Franca Pereira e Silva and S. J. Simske, "The CNN Corpus: A large textual corpus for single-document extractive summarization," *Proceedings of the ACM Symposium on Document Engineering,* pp. 1-10, 2019.

[61] Python Software Foundation, "xml.etree.ElementTree - The ElementTree XML API," 2022. [Online]. Available: https://docs.python.org/3/library/xml.etree.elementtree.html. [Accessed May 2022].

[62] Python Software Foundation, "Beautiful Soup 4," 2022. [Online]. Available: https://pypi.org/project/beautifulsoup4/. [Accessed April 2022].

[63] Python Software Foundation, "xml.dom - The Document Object Document API," 2022. [Online]. Available: https://docs.python.org/3/library/xml.dom.html. [Accessed April 2022].

[64] L. Richardson, "Beautiful Soup," [Online]. Available: www.crummy.com/software/BeautifulSoup/. [Accessed 1 April 2022].

[65] A. Nenkova and L. Venderwende, "The impact of frequency on summarization," 2005.

[66] G. Erkan and D. R. Radev, "LexRank: Graph-based Lexical Centrality as Salience in Text Summarization," *The Journal of Artificial Intelligence Research,* vol. 22, p. 457–79, 2004.

[67] P. Verma, S. Pal and H. Om, "A Comparative Analysis on Hindi and English Extractive Text Summarization," *ACM Trans. Asian Low-Resour. Lang. Inf. Process.,* vol. 18, no. 3, 2013.

[68] "LSA Based Text Summarization," *International Journal of Recent Technology and Engineering,* vol. 9, no. 2, pp. 150-156.

[69] H. P. Edmundson, "New Methods in Automatic Extracting," *Journal of the ACM,* vol. 16, no. 2, pp. 264-285, 1969.

[70] B. Chen, H.-C. Chang and K.-Y. Chen, "Sentence modeling for extractive speech summarization," in *IEEE International Conference on Multimedia and Expo (ICME)*, San Jose, CA, 2013.

[71] C. E. Shannon, "A Mathematical Theory of Communication," *Bell System Technical Journal,* vol. 27, no. 3, pp. 379-423, 1948.

[72] h. S. community, "scipy.stats.entropy 1.9.1," The SciPy community, [Online]. Available: https://docs.scipy.org/doc/scipy/reference/generated/scipy.stats.entropy.html. [Accessed September 2022].

[73] S. Niwattanakul, J. Singthongchai, E. Naenudorn and S. Wanapu, "Using of Jaccard Coefficient for Keywords Similarity," in *Proceedings of the International MultiConference of Engineers and Computer Scientists*, Hong Kong, 2013.

[74] E. Y. Puspaningrum, B. Nugroho, A. Setiawan and N. Hariyanti, "Detection of Text Similarity for Indication Plagiarism Using Winnowing Algorithm Based K-gram and Jaccard Coefficient," in *International Conference of Science and Technology*, 2019.

[75] R. Thinniyam, On Statistical Sequencing of Document Collections, ProQuest Dissertations Publishing, 2014.

[76] A. Devi T., R. R. and G. R., "A novel approach for curriculum ordering of course topics using data mining," *Journal of Ambient Intelligence and Humanized Computing,* vol. 11, no. 3, pp. 1127-1136, 2020.

[77] M. Kahani, A. Ghorbani and M. R. Meybodi, "A new algorithm for optimal curriculum ordering using genetic algorithm and local search," *Journal of Educational Technology & Society,* vol. 18, no. 1, pp. 207-221, 2015.

[78] Project Gutenberg, "Project Gutenberg," Project Gutenberg, [Online]. Available: https://www.gutenberg.org/. [Accessed October 2022].

[79] R. Řehůřek, "Word2Vec Model," Gensim, [Online]. Available: https://radimrehurek.com/gensim/auto_examples/tutorials/run_word2vec.html. [Accessed November 2022].

[80] Python Software Foundation, "sumy 0.11.0," The Python Package Index (PyPI), October 2022. [Online]. Available: https://pypi.org/project/sumy/. [Accessed October 2022].

[81] D. M. Blei, A. Y. Ng and M. I. Jordan, "Latent Dirichlet Allocation," *Journal of Machine Learning Research,* vol. 3, pp. 993-1022, 2003.

[82] R. Řehůřek, G. Mohr, M. Penkov and I. Menshikh, "Gensim," Gensim, [Online]. Available: https://radimrehurek.com/gensim/auto_examples/tutorials/run_lda.html. [Accessed November 2022].

[83] W. Kurt, "Kullback-Leibler Divergence Explained," Count Bayesie, 10 May 2017. [Online]. Available: https://www.countbayesie.com/blog/2017/5/9/kullback-leibler-divergence-explained. [Accessed November 2022].

[84] W. Kurt, Bayesian Statistics the Fun Way, San Francisco, CA: No Starch Press, 2019.

[85] R. Hamming, "Classical Error-Correcting Codes," in *Classical and Quantum Information*, Elsevier Inc., 2012, pp. 345-454.

[86] V. I. Levenshtein, "Binary codes capable of correcting deletions, insertions, and reversals," *Soviet Physics Doklady,* vol. 10, no. 8, pp. 707-710, 1966.

[87] J. Gan and Y. Qi, "Selection of the Optimal Number of Topics for LDA Topic Model—Taking Patent Policy Analysis as an Example," *Entropy,* vol. 23, no. 1301, 2021.

[88] C. E. Bonferroni, Teoria statistica delle classi e calcolo delle probabilità, Pubblicazioni del R Istituto Superiore di Scienze Economiche e Commerciali di Firenze, 1936.

[89] J. V. Ruiz, "Grammy winner explains why Adele is right – album tracks should not be shuffled," Big Think, 17 December 2021. [Online]. Available: https://bigthink.com/high-culture/dont-shuffle-albums/. [Accessed 7 March 2023].

[90] E. Gamma, R. Helm, R. Johnson and J. Vlissides, Design Patterns: Elements of Reusable Object-Oriented Software, Boston, MA: Addison-Wesley Professional, 1994.

[91] S. Simske, Meta-analytics: Consensus Approaches and System Patterns for Data Analysis, San Rafael: Morgan Kaufmann Publishers, 2019.

List of Abbreviations

Acronym / Initialism	Definition
ACL	Association for Computational Linguistics
ACM	Association for Computing Machinery
ADA	Americans with Disabilities Act
AI	Artificial Intelligence
ANOVA	Analysis of Variance
API	Application Programming Interface
BERT	Bidirectional Encoder Representations from Transformers
BLEU	Bilingual Evaluation Understudy
CD	Compact Disc
CEO	Chief Executive Officer
CNN	Cable News Network
COTS	Commercial Off-The-Shelf
CPRE	Certified Professional for Requirements Engineering
DAU	Defense Acquisition University
DOM	Document Object Model
EARS	Easy Approach to Requirements Syntax
EMNLP	Conference on Empirical Methods in Natural Language Processing
FLAIRS	Florida Artificial Intelligence Research Society
FRETISH	Formal Requirements Elicitation Tool Language
GloVe	Global Vectors
HD	Hamming Distance
ICCIDS	International Conference on Computational Intelligence and Data Science
ICME	International Conference on Multimedia and Expo
ID	Identification
IEC	International Electrotechnical Commission
IEEE	Institute of Electrical and Electronics Engineers
INCOSE	International Council on Systems Engineering
ISO	International Organization for Standardization
KL	Kullback-Leibler
KLD	Kullback-Leibler Divergence
LDA	Latent Dirichlet Allocation
LDC	Linguistic Data Consortium
LEGO	A Danish toy company
LR	Logistic Regression
LSA	Latent Semantic Analysis
LSTM	Long Short-Term Memory

Acronym / Initialism	**Definition**
MBSE	Model-Based Systems Engineering
ML	Machine Learning
MNB	Multinomial Naive Bayes
MS	Microsoft
MWOE	Mean Weighted Order Error
NASA	National Aeronautics and Space Administration
NCOE	Normalized Clustering Order Error
NHD	National Hamming Distance
NLP	Natural Language Processing
NLTK	Natural Language Toolkit
NMHD	Normalized Modified Hamming Distance
NMWOE	Normalized Mean Weighted Order Error
NRMSE	Normalized Root Mean Square Error
PDF	Portable Document Format
POS	Part of Speech
PROMISE	Predictive Models in Software Engineering
PSO	Particle Swarm Optimization
PWS	Performance Work Statement
RE	Requirements Engineering
RFI	Request for Information
RFP	Request for Proposal
ROUGE	Recall-Oriented Understudy for Gisting Evaluation
SMART	Specific, Measurable, Achievable, Relevant, and Time-bound
SOW	Statement of Work
SVM	Support Machine Vectors
SWaP	Size, Weight, and Power
TF*IDF	Term Frequency * Inverse Document Frequency
UCLA	University of California, Los Angeles
UCSD	University of California, San Diego
XML	Extensible Markup Language

Appendix A: Test Corpora for Chapter 4

Table 33. Control Corpora (Novels, Biographies, Wikipedia)

Document Sets (Clustered in Genres)	Num of Component Documents	Number of Sentences / Document
Novels		
Alice in Wonderland	12	134.2
Crime & Punishment	39	366.6
Frankenstein	24	128.4
The Time Machine	16	121.1
Wonderful Wizard of Oz	24	92.3
Biographies		
12 Years a Slave	23	169.0
Autobiography of Andrew Carnegie	29	189.8
Autobiography of Benjamin Franklin	19	114.2
Edison, His Life and Inventions	29	258.2
Johann Sebastian Bach	11	120.4
Wikipedia Article Sets		
Chess-related articles	37	566.8
Homo sapiens-related articles	21	1274.8
Circular Economy articles	16	967.9
Plate Tectonics-related articles	19	505.5
Systems Engineering-related articles	24	238.6

Table 34. Test Corpora (Textbooks, Courses, Journals)

Document Sets (Clustered in Genres)	Num of Component Documents	Number of Sentences / Document
Textbooks		
Biology	47	596.8
Borky - Effective MBSE	15	535.7
Camastra - ML for Audio, Image and Video Analysis	16	753.8
College Physics	34	1106.1
Kutz - Handbook of Env Engineering	24	789.5
Ling - Physics	44	1053.8
Robertazzi - Introduction to Computer Networking	11	251.6
Simske - Functional Applications of Text Analysis	7	583.6
Jurafsky - Speech & Language Processing	26	425.4
van der Aalst - Process Mining	16	705.8
Courses		
Simske - Analytics Class	15	320.9
Simske - Cybersecurity Class	12	132.1
Simske - IP Class	12	558.3
Simske - Sensing & Imaging Class	15	211.0
Manning & Socher – Natural Language Processing	19	130.3
Journals		
2014 Transactions on Petri Nets	5	557.5
2016 Usability & Accessibility	11	357.6
2017 NASA Formal Methods	31	388.4
2017 Information Science and Applications	96	189.9
2018 Natural Language Processing	53	216.5

Table 35. Test Corpora (Dissertations)

Author	Document Set (Dissertation)	Year	Number of Component Documents (Chapter Count)	Number of Sentences / Document
Agricultural and Resource Economics / Agricultural Biology				
Bird, Tharina Louise	Cheliceral morphology of Solifugae (Arachnida): primary homology, terminology, and character survey	2015	6	543.0
Boateng, Charles Osei	Physiological responses of onion germplasms to Iris yellow spot virus and onion thrips (Thrips tabaci)	2012	6	295.8
Gilligan, Todd Michael	Advances in tortricid systematics and identification (Lepidoptera: Tortricidae)	2012	9	699.3
Küpper, Anita	Molecular genetics of herbicide resistance in Palmer amaranth (Amaranthus palmeri): metabolic tembotrione resistance and geographic origin of glyphosate resistance	2018	6	178.8
Richardson, Leslie A.	Quantifying the economic health cost of exposure to wildfire smoke: four essays in non-market valuation, methodological comparisons, and econometric methods to address endogeneity	2011	6	206.3
Animal Sciences				
Atkins, Colton A.	Investigation of an embedded-optical-base system's functionality in detecting signal events for gait measurements	2018	8	169.8
Belk, Aeriel D.	Microbiome surrounding death and decay, The: microbial ecology of food processing, meat spoilage, and human decomposition environments	2021	6	273.3
Crawford, Natalie Faye	Calcium signaling genes in association with altitude-induced pulmonary hypertension in Angus cattle	2019	7	213.0
da Costa Santos, Hugo F.	Circulating micro RNA in insulin resistant horses	2018	6	124.5
Jennings, Kaysie Jean	Characterization of pulmonary hypertension status and utilization of multi-omics analyses to discover variants that may inform selection against high mean pulmonary arterial pressure in Angus cattle	2020	8	192.5
Kline, Helen Carter	Carcass bruising location and bruise trim loss in finished steers, cows, and bulls at five commercial slaughter facilities	2018	6	127.5
Paudyal, Sushil	Evaluation of novel strategies for improving health and wellbeing of dairy cattle	2018	7	142.9
Reyes, Arquimides A.	In vitro system evaluation of the rumen microbiome and rumen fermentation characteristics as a result of differing feed additives, An	2019	6	174.8
Shen, Cangliang	Control of Escherichia coli O157:H7 and Listeria monocytogenes in meat and poultry products with chemicals and heating treatments	2010	10	144.7
Zeng, Xi	Angus cattle at high altitude: pulmonary arterial pressure, estimated breeding value and genome-wide association study	2016	7	280.7

Author	Document Set (Dissertation)	Year	Number of Component Documents (Chapter Count)	Number of Sentences / Document
Atmospheric Science				
Brewer, Jared F.	Ketones in the troposphere: studies of loss processes, emissions, and production	2020	6	229.5
Casas, Eleanor G.	Investigation of relationships between tropical cyclone structure and intensity change	2022	6	180.0
Childs, Samuel J.	Projecting end-of-century human exposure to eastern Colorado tornadoes and hailstorms: meteorological and societal perspectives	2020	6	386.7
Dougherty, Erin M.	Characteristics of current and future flood-producing storms in the continental United States	2020	7	194.7
Freeman, Sean William	Examining the impacts of convective environments on storms using observations and numerical models	2022	6	216.8
Jenney, Andrea M.	Quantifying and understanding current and future links between tropical convection and the large-scale circulation	2020	6	184.5
Lee, Yoonjin	Using GOES-16 ABI data to detect convection, estimate latent heating, and initiate convection in a high resolution model	2021	6	168.8
Lindaas, Jakob	Investigating emissions and evolution of reactive nitrogen in western U.S. wildfire smoke plumes	2020	6	250.7
Naegele, Alexandra Claire	Influence of cloud radiative effects on hydrologic sensitivity and variability, The	2021	6	112.3
Trabing, Benjamin	On intensity change and the effects of shortwave radiation on tropical cyclone rainbands	2020	6	305.8
Biochemistry & Molecular Biology				
Carter, Megan	Structure energy relationship of biological halogen bonds	2012	6	291.5
Ford, Melissa Coates	Development of computational tools to model molecular interactions for medicinal chemistry	2017	7	209.9
Hartje, Rhianon Kay Rowe	Relationships between hydrogen bonds and halogen bonds in biomolecular engineering	2019	6	167.8
Lammers, Lindsay	Regulation of dynein activity during spindle positioning in budding yeast	2020	6	266.2
Lyon, Kenneth Ray, Jr.	Multi-color visualization and quantification of single RNA translation and HIV-1 programmed ribosomal frameshifting in living cells	2019	6	170.3
Marzo, Matthew G.	Dynein mutagenesis reveals the molecular basis for dynein regulation in broad spectrum neurological diseases	2020	6	318.0
Russell, Theresa Michelle Tidd	Surface protease of Lyme disease bacteria degrades host extracellular matrix components and induces inflammatory cytokines in vitro, A	2012	8	170.1
Shattuck, Jenifer Elizabeth	Investigating the roles prion-like domains play in cellular stress responses	2018	6	219.8
Thurston, Alison K.	Chromatin binding factor Spn1 contributes to genome instability in Saccharomyces cerevisiae, The	2018	6	172.8

Author	Document Set (Dissertation)	Year	Number of Component Documents (Chapter Count)	Number of Sentences / Document
Wyka, Stephen Andrew	From fields to genomes: towards a comprehensive understanding of the lifestyle and evolution of Claviceps purpurea the ergot fungus	2020	6	373.0
Biology				
Healy, Jessica	Hormonal controls of obesity in feeding and fasting hibernating mammals	2010	6	125.1
Kroh, Gretchen Elizabeth	Initiation and regulation of iron economy in Arabidopsis thaliana chloroplasts	2020	6	332.0
Miller, Ryan S.	Interaction among societal and biological drivers of policy at the wildlife-agricultural interface	2017	6	251.2
Quinn, Colin Francis	Ecological interactions involving plant selenium hyperaccumulation	2010	7	199.6
Stapp, Paul T.	Determinants of habitat use and community structure of rodents in northern shortgrass steppe	1996	6	484.2
Stoerger, Vincent	Charactarization of a nitrate responsive MYB transcription factor in Arabidopsis	2013	7	165.0
Womack, Molly Corinne	Evolution of 'earlessness' in the true Toad family (Bufonidae), The	2016	6	152.8
Biomedical Sciences				
Benham, Hayley Marie	Investigation of assisted reproductive technologies (ART) for conservation of Bovidae	2022	6	290.0
da Silveira, Juliano Coelho	Role of cell-secreted vesicles in equine ovarian follicle development, The	2013	6	182.0
Heck, Ashley L.	Sex-dependent function and regulation of the hypothalamic pituitary adrenal axis	2019	6	308.0
Heise, Natascha	Evaluating curricular implementation techniques to enhance anatomy education	2021	8	269.4
Johnson, Ben	Molecular mechanisms regulating Kv2.1-induction of endoplasmic reticulum / plasma membrane contact sites	2019	6	217.3
Magee, Christianne	Evaluation of kisspeptin in the mare	2010	6	186.3
Meyers, Jacob	Characterizing the target of ivermectin, the glutamate-gated chloride channel, and other insecticide targets as candidate antigens for an anti-mosquito vaccine	2015	6	229.3
Romero, Jared Jerome	Endocrine actions of IFNT during early ruminant pregnancy	2013	6	294.3
Schwerdtfeger, Luke A.	Anatomic plasticity and functional impacts of neural – immune and neural – epithelial signaling in the intestine	2021	10	165.8
Vallejos, Maximiliano Jose	Age-dependent decline in Kv4 channels, underlying molecular mechanisms, and potential consequences for coordinated motor function	2019	6	233.6
Chemical and Biological Engineering				
Adkins, Nadine C.	Framework for development of data analysis protocols for groundwater quality monitoring systems	1992	6	237.2

Author	Document Set (Dissertation)	Year	Number of Component Documents (Chapter Count)	Number of Sentences / Document
Farr, Anne McCormack	Optimal design of groundwater quality monitoring networks	1992	11	157.4
Gujarathi, Ninad	Phytoremediation of tetracycline and oxytetracycline	2005	8	170.5
Harcum, Jonathan Brooks	Water-quality data analysis protocol development	1990	7	293.9
Lorentz, Simon A.	Dependence of the formation factor on the unsaturated hydraulic properties of porous media	1995	7	88.7
Mirjalili, Noushin	Taxol productivity and physiological relationships in suspension cultures of Taxus Cuspidata	1995	8	159.4
Peterson, Thomas Charles	Transport of copiotrophic bacteria in oligotrophic coarse soils : a Monte Carlo analysis	1987	7	154.1
Phisalaphong, Muenduen	Metabolic manipulation of Taxus canadensis for taxol production	1999	9	169.9
Stephens, Matthew David	Thin film integrated optical waveguides for biosensing using local evanescent field detection	2010	6	192.0
Wang, Yan	Novel applications of advanced integral-equation theories to various polymeric systems	2021	8	128.5
Chemistry				
Bhattacharya, Atanu	Part 1, executed electronic state decomposition of energetic molecules. Part 2, conformation specific reactivity of radical cation intermediates of bioactive molecules	2010	11	246.8
Corbin, Daniel Andreas	Advancements in organocatalyzed atom transfer radical polymerization by investigation of key mechanistic steps	2022	6	662.9
Farah, Yusef Rodney	Accessing molecular structure and dynamics of photoelectrochemical systems with nonlinear optical spectroscopy	2022	8	179.6
Gordon, Jenna Leigh	Anticancer potential of nitric oxide-based therapeutics for pediatric and adult cancers	2021	7	159.1
Gray, Chandele Ramsey	Asperparaline A: biosynthetic studies and synthetic efforts	2008	6	272.8
Gubler, Daniel A.	Mitomycin alkaloids: synthetic studies	2009	7	250.9
Halligan, Kathleen Marie	Synthetic and biosynthetic studies of the brevianamides	2000	6	212.0
Kudisch, Max	Towards elucidating photochemical reaction pathways in nickel catalyzed cross coupling and organocatalyzed Birch reduction	2021	6	512.7
Newkirk, Tenaya L.	Towards the total synthesis of 14-acetoxygelsenicine and synthesis of largazole analogs	2009	6	366.2
Stocking, Emily M.	Studies on the biosynthesis of paraherquamide A and the total synthesis of (±) VM55599	2001	7	325.6
Civil and Environmental Engineering				
Akmalah, Emma	Integrated flood management model: a socio-technical systems approach to overcome institutional problems in Jakarta	2010	6	238.5

Author	Document Set (Dissertation)	Year	Number of Component Documents (Chapter Count)	Number of Sentences / Document
Baker, Jessica L.	Innovative application of random packing material to enhance the hydraulic disinfection efficiency of small scale water systems, The	2021	7	218.3
Caruso, Brian S.	Watershed-based methodology for assessment of nonpoint source pollution from inactive mines	1995	9	199.8
Coelho Maran, Ana Carolina	Multicriteria decision support system to delineate water resources planning and management regions	2010	7	335.2
Dao, Thang Nguyen	Development of performance-based wind engineering for residential structures: from concept to application	2010	8	145.6
Hafez, Youssef Ismail	K-ϵ turbulence model for predicting the three-dimensional velocity field and boundary shear in closed and open channels, A	1995	6	169.2
Hotto, Harvey P.	Framework for evaluating water quality information system performance	1994	7	252.9
Liu, Hongyan	Performance-base seismic design of woodframe buildings using non-linear time history analysis	2010	8	139.8
Oke, Oluwatobi Olamiposi	Systems-based approaches for evaluating residential-based hazards to inform environmental exposure intervention design	2022	6	183.5
Yin, Yiming	Elucidating the mechanisms and developing mitigation strategies of mineral scaling in membrane desalination	2022	7	177.4
Clinical Sciences				
Colbath, Aimee	Evaluation of allogeneic bone marrow-derived mesenchymal stem cells for use in equine joints: in vitro to preclinical evaluation	2019	7	184.3
Contreras, Elena T.	Assessment of novel strategies for the prevention and treatment of feline upper respiratory tract infections in shelters and feline herpesvirus-1 in laboratory settings	2019	8	189.1
Erales Villamil, José Alberto	Silvopastoral system for sustainable cattle production in the tropics of Mexico	2017	6	148.3
Kradangnga, Krishaporn	Development of an ex vivo pulsatile heart model of functional mitral regurgitation to facilitate posterior papillary muscle geometric studies and subvalvular surgical strategy	2018	7	107.7
Miller, David Steven	Oropharyngeal bacteria, with respect to animal health classification, and viral serology of Montana bighorn sheep (Ovis canadensis) and domestic (Ovis aries) near to and distant from the wildlife/domestic animal interface	2010	7	210.1
Nelson, Bradley B.	Investigation of cationic contrast-enhanced computed tomography for the evaluation of equine articular cartilage	2017	6	222.7
Pezzanite, Lynn M.	Use of immune activated cellular therapy and risks with antibiotic administration in treatment of septic arthritis	2021	8	177.9

Author	Document Set (Dissertation)	Year	Number of Component Documents (Chapter Count)	Number of Sentences / Document
Shropshire, Sarah	Coagulation abnormalities in Ehrlichia canis-infected dogs and detection and dynamics of anti-platelet antibodies in thrombocytopenic dogs	2018	8	145.8
Summers, Stacie	Assessment of novel causes and investigation into the gut microbiome in cats with chronic kidney disease	2020	8	165.5
Wennogle, Sara Anne Jablonski	Clinical, clinicopathologic, histopathologic and immunohistochemical features of dogs with chronic enteropathy with and without concurrent protein-losing enteropathy: focus on the intestinal lymphatic vasculature	2018	7	148.0
Computer Science				
Chaabane, Mohamed	Learned perception systems for self-driving vehicles	2022	6	251.8
Homayouni, Hajar	Anomaly detection and explanation in big data	2021	6	231.5
Kommrusch, Steve	Machine learning for computer aided programming: from stochastic program repair to verifiable program equivalence	2022	8	372.6
Lionelle, Albert	Spiral design, A: redesigning CS 1 based on techniques for memory recall	2021	7	248.1
McNeely-White, David G.	Revealing and analyzing the shared structure of deep face embeddings	2022	6	194.0
Patil, Dhruva Kishor	Something is fishy! - How ambiguous language affects generalization of video action recognition networks	2022	6	265.2
Rammer, Daniel P.	Harnessing spatiotemporal data characteristics to facilitate large-scale analytics over voluminous, high-dimensional observational datasets	2021	8	138.6
Shirazi, Hossein	Phishing detection using machine learning	2021	7	199.3
Weerawardhana, Sachini Situmini	Helping humans and agents avoid undesirable consequences with models of intervention	2021	7	544.3
Xu, Zhisheng	Generalizations of comparability graphs	2022	8	127.6
Economics				
Algarini, Abdullah	Effect of human capital on total factor productivity growth in the Arab Gulf Cooperation Council countries, The	2017	7	190.1
Alkhdour, Rajeh	Estimating the shadow economy in Jordan: causes, consequences, and policy implications	2011	9	118.4
Ardiyanto, Ferry	Foreign direct investment and corruption	2012	6	389.8
Bhattarai, Niroj Kumar	What factors affect school attendance? Quantitative and qualitative study of evidence from Nepal	2017	8	174.2
Chisesi, Lawrence J.	School choice impacts within a local school district	2012	6	271.2
Hannum, Christopher M.	Three applications of regional CGE models	2014	6	153.3

Author	Document Set (Dissertation)	Year	Number of Component Documents (Chapter Count)	Number of Sentences / Document
Khreisat, Mohammad Abdallah	Production function estimations and policy implications	2011	8	277.8
Knight, Tabitha	Gender dynamics of public finance, The: a Chinese and cross-country analysis	2014	6	166.7
Lin, Chun-Wei	Moral hazard in health care: case study of Taiwan's national health insurance	2012	7	181.1
Electrical and Computer Engineering				
Brizuela, Fernando	Table-top, full-field, actinic microscope for extreme ultraviolet lithography mask characterization	2010	6	189.7
Hall, John Joseph	Long-term learning for adaptive underwater UXO classification	2022	9	113.0
Key, Cam	Improvements in computational electromagnetics solver efficiency: theoretical and data-driven approaches to accelerate full-wave and ray-based methods	2020	8	200.8
Kukkala, Vipin Kumar	Robust and secure resource management for automotive cyber-physical systems	2022	8	329.1
Machovec, Dylan	Dynamic resource management in heterogeneous systems: maximizing utility, value, and energy-efficiency	2021	7	308.1
Mahindre, Gunjan S.	Efficient representation, measurement, and recovery of spatial and social networks	2021	9	276.6
Muramudalige, Shashika R.	Automating investigative pattern detection using machine learning & graph pattern matching techniques	2022	11	163.3
Robbiano, Christopher P.	Optimal path planning for detection and classification of underwater targets using sonar	2021	8	121.8
Tiku, Saideep	Secure, accurate, real-time, and heterogeneity-resilient indoor localization with smartphones	2022	8	287.5
Wang, Erkang	Transient absorption imaging of hemeprotein in fresh muscle fibers	2022	9	154.0
Environmental and Radiological Health Sciences				
Bantle, Collin M.	Neuroinflammation and the two-hit hypothesis of Parkinson's disease	2019	7	294.6
Braley, Gerald Scott	Net-risk approach to displacement and reoccupation decision making, A	2019	6	101.2
Brents, Colleen	Occupational injuries among craft brewery workers in Colorado	2021	6	416.2
Burton, Lindsey Hammond	Elucidating the role of iron in the pathogenesis of idiopathic osteoarthritis in the Dunkin-Hartley animal model	2021	6	190.5
Hischke, Molly	Reference values of the distal sensory median and ulnar nerves among newly hired workers	2021	6	168.5
Hoffman, Timothy Edward	Multimethod simulation paradigm for investigating complex cellular responses in biological systems of aging and disease, A	2019	7	206.4

Author	Document Set (Dissertation)	Year	Number of Component Documents (Chapter Count)	Number of Sentences / Document
Hook, Sarah A.	Public health considerations for a potential Lyme disease vaccine in the United States: cost of illness, vaccine acceptability, and net costs of a vaccination program	2021	6	148.0
Martinez, Stephen K.	Evaluation of dose enhancement due to CuATSM uptake in hypoxic environments with external radiation	2019	7	239.9
Schaaf, David Nicholas, Jr.	Skin tissue optical and thermal reactions to pulse sequences of thulium yttrium aluminum garnet laser irradiation	2010	7	188.1
Walker, Ethan Sheppard	Associations between air pollution emitted from cookstoves and central hemodynamics, arterial stiffness, and blood lipids in laboratory and field settings	2019	6	215.3
Fish, Wildlife, and Conservation Biology				
Adeola, Moses Olanrewaju	Utilization of wildlife resources in Nigeria	1987	10	150.4
Davis, Amy Jane	Gunnison sage-grouse demography and conservation	2012	6	245.3
Dergam, Jorge A.	Phylogeography and character congruence within the Hoplias malabaricus Bloch, 1794 (Erythrinidae, Characiformes, Ostariophysi) species complex	1996	6	253.3
Nimir, Mutasim Bashir	Wildlife values and management in northern Sudan	1983	7	313.6
Northrup, Joseph M.	Behavioral response of mule deer to natural gas development in the Piceance Basin	2015	7	295.7
Food Science and Human Nutrition				
Amer, Fauzi Saleh Massoud	Effect of farm to fork operations on bioactive compounds in white-fleshed and color-fleshed potatoes	2015	6	292.0
Bauer, Laura M.	Science of food fermentation: development of a university curriculum and outreach educational materials	2015	6	91.3
Booth-Kalajian, Andrea Deborah	Testing the metabolic sink postulate: subcutaneous adipose tissue the protective depot	2017	6	267.3
Chlipalski, Micheline	Development and evaluation of an online training for paraprofessional nutrition educators from the expanded food and nutrition education program (EFNEP) addressing prenatal nutrition	2016	6	94.3
Hibbs-Shipp, Sarah Katherine	Healthy homes: exploring the quality of the home food environment and maternal health factors	2018	7	161.4
Magnuson, Aaron Mark	Visceral adiposity and pro-inflammation: contributions and consequences of immunity	2017	7	235.0
Murray, Erin K.	Development and testing of measures to assess nutrition behavior change in low income adults participating in the Expanded Food and Nutrition Education Program	2017	6	182.8

Author	Document Set (Dissertation)	Year	Number of Component Documents (Chapter Count)	Number of Sentences / Document
Radhakrishnan, Sridhar	Potato and grape polyphenols, respectively, suppress high-fat diet-elevated oxidative stress/innate inflammation markers in porcine model and induce apoptosis in HCT-116 p53 +/+ and p53 -/- human colon cancer cell lines in vitro	2014	6	243.3
Sheflin, Amy Marie	Supplementing powdered high-fiber foods to alter gut microbial metabolism for colorectal cancer prevention	2016	6	205.8
Smith, Stephanie Laine	It's not healthy if they don't eat it: school lunch plate waste and strategies to increase vegetable consumption	2015	6	112.8
Baival, Batkhishig	Community-based rangeland management and social-ecological resilience of rural Mongolian communities	2012	7	314.1
Bruno, Jasmine Elizabeth	Linked livelihoods, land-use, and identities on transitioning landscapes in northeastern Colorado: a social-ecological study	2021	6	379.7
Bui, Doi The	Patterns of growth dominance and neighborhood effects in eucalyptus plantations and tropical forests	2008	6	131.2
Ex, Seth	Crown characteristics of interior western U.S. conifers with implications for canopy fire hazard evaluation	2014	6	155.8
Franco, Nilson	Three-dimensional finite element model to predict pole strength	1992	7	147.1
Gannon, Benjamin Michael	Wildfire-water supply risk in montane watersheds of Colorado: baseline assessment and evaluation of mitigation strategies	2020	6	324.2
Jablonski, Kevin E.	Skill of managers and the wisdom of herds, The: examining an alternative approach to grazing management in larkspur habitat	2019	6	220.8
Khuc, Quy Van	Integrated eco-socio-economic analysis of forest transition and forest restoration in Vietnam, An	2018	6	242.2
Lenachuru, Clement Isaiah	Ilchamus pastoralists' indigenous knowledge and its use in coping with and adapting to climate change in Marigat, Kenya	2016	6	303.3
Wilmer, Hailey	Cattle ranching on the western Great Plains: a study of adaptive decision-making	2016	6	235.2
Ziegler, Justin Paul	Causes, consequences, and management of tree spatial patterns in fire-frequent forests	2022	6	280.5
Geosciences				
Al Faitouri, Mohamed S. E.	Isotope and noble gas study of three aquifers in central and southeast Libya	2013	7	142.0
Deems, Jeffrey S.	Quantifying scale relationships in snow distributions	2007	7	184.9
Duru, Umit	Modeling sediment yield and deposition using the SWAT model: a case study of Cubuk I and Cubuk II reservoirs, Turkey	2015	8	126.1
Hultstrand, Douglas Michael	Uncertainty in hydrological estimation	2021	6	170.2

Author	Document Set (Dissertation)	Year	Number of Component Documents (Chapter Count)	Number of Sentences / Document
Kramer, Natalie	Great river wood dynamics in northern Canada	2016	7	393.3
Mavor, Skyler	Timing, kinematics, and tectonic significance of strike-slip fault systems in the Atacama Desert of northern Chile and the Lower Colorado River corridor, U.S.A.	2021	6	290.5
Patterson, Glenn G.	Trends in snow water equivalent in Rocky Mountain National Park and the northern Front Range of Colorado, USA	2016	6	145.5
Sutfin, Nicholas A.	Spatiotemporal variability of floodplain sediment and organic carbon retention in mountain streams of the Colorado Front Range	2016	6	204.3
Thomas, Dai B.	Island dynamics and their role in regulating sediment flux in the Middle Snake River, Idaho	2014	7	281.4
Venable, Niah B. H.	Trends and tree-rings: an investigation of the historical and paleo proxy hydroclimate record of the Khangai Mountain Region of Mongolia	2016	6	198.3
Health and Exercise Science				
Bruns, Danielle Reuland	Oxidative and energetic stress: regulation of Nrf2 and mitochondrial biogenesis for slowed aging interventions	2013	6	144.5
Crecelius, Anne Renee	Role of vascular hyperpolarization in muscle blood flow regulation in healthy humans	2013	6	161.8
De Jong, Nathan Paul	Short-term metabolic effects of breaking up sedentary behaviors	2022	7	161.4
Kirby, Brett Sean	On the role of circulating ATP in vascular control at rest and during exercise of aging humans	2010	6	160.7
Nuss, Kayla	Wearable fitness trackers in physical activity research: accuracy assessment and effects on motivation and engagement	2021	6	130.5
Richards, Jennifer Clarke	Role of the sympatho-adrenal system in the regulation of peripheral vascular tone in healthy aging humans	2014	6	132.5
Robinson, Matthew McHutcheson	Protein synthesis rates in response to exercise and β-adrenergic signaling in human skeletal muscle	2011	6	164.3
Scalzo, Rebecca Lynn	Adjusting attitudes about altitude: novel approaches to promote human performance in high-altitude	2014	6	123.8
Horticulture & Landscape Architecture				
Bogs, Jana Dee	Effects of organic, biological and conventional production methods on apple antioxidant levels, sensory qualities and human glycemic response	2009	6	208.7
Castleberry, Henry C.	Development of methods to estimate or reduce pressure flattening of potatoes during storage	2013	6	221.7
Chaparro, Jacqueline Michelle	Manipulating the soil microbiome to increase plant health and productivity	2015	6	242.7
Emargi, Esam	Minimizing the storage losses of potatoes under different storage treatments	2021	6	247.2

Author	Document Set (Dissertation)	Year	Number of Component Documents (Chapter Count)	Number of Sentences / Document
Zuber, Tatiana	Inhibition of HT-29 colon cancer cell cultures by extracts from biodiverse germplasm sources of Solanum tuberosum L.	2012	7	297.9
Human Dimensions of Natural Resources				
Bonfield, Susan B.	Engaging Latino audiences in informal science education	2014	7	186.9
David-Chavez, Dominique M.	Guiding model for decolonizing environmental science research and restoring relational accountability with Indigenous communities, A	2019	6	183.7
DiEnno, Cara Marie	Case study of social capital and collaboration as a communication process in an urban community-based ecological restoration project, A	2009	6	350.3
Knight, David Warner	Tourism, poverty, and development: local perceptions, empowerment, and strategies for change in Peru's Sacred Valley	2015	6	220.3
McGrady, Pavlina Stefanova	Diffusion of sustainability innovation among Colorado ski resorts: a mixed methods approach	2016	7	312.7
Nobre, Ismael	Development and evaluation of an automated multimedia kiosk-based visitor survey system in Iguaçu National Park, Brazil, The	2011	8	176.0
Raadik-Cottrell, Jana	Cultural memory and place identity: creating place experience	2010	7	383.1
Journalism & Media Communication				
Boehm, Nicholas	Presence, what is it good for? Exploring the benefits of virtual reality at evoking empathy towards the marginalized	2020	6	261.8
Humphrey, Michael	Working narrative, The: analysis of linguistic structures and styles in life storytelling on social media	2017	8	204.5
Huntington, Heidi E.	Affect and effect of Internet memes, The: assessing perceptions and influence of online user-generated political discourse as media	2017	7	299.9
Johnson, Emily	Pinning for leisure or labor?: unveiling constructions of wedding planning via Pinterest	2017	8	653.2
Littlefield, Joanne Speirs	Visual rhetoric of U.S. agricultural films: auteurs, actors and assimilation	2016	8	125.0
Mokry, Melissa M.	Analyzing risk-related information seeking behavioral intention and risk perception of wildfires: the High Park Fire Burn Area	2019	7	351.6
Russell, Gregory	Critical analysis of participatory research in the social sciences, A	2022	6	202.2
Stone, Leah	Digitization, innovation, and participation: digital conviviality of the Google Cultural Institute	2018	11	470.8
Zhang, Hui	Conflicting health-related scientific evidence in news reports: effects of presentation format and hedging on perceived issue uncertainty and source credibility	2016	6	268.5

Author	Document Set (Dissertation)	Year	Number of Component Documents (Chapter Count)	Number of Sentences / Document
Zlaten, Rhema	Autonomy in local digital journalism: a mixed-method triangulation exploration of the organizational culture and individual moral psychology factors of digital news workers	2021	7	305.6
Mathematics				
Afandi, Rebecca	Conjugacy extension problem, The	2021	6	239.5
Bush, Johnathan E.	Topological, geometric, and combinatorial aspects of metric thickenings	2021	7	182.1
Heath, Levi Nathaniel	Quantum Serre duality for quasimaps	2022	6	143.0
McBee, Cayla D.	Some topics in combinatorial phylogenetics	2010	7	206.3
McCleary, Alexander J.	Generalizations of persistent homology	2021	6	98.2
Pinckney, Casey M.	Independence complexes of finite groups	2021	8	209.8
Roberts, Colin	Hodge and Gelfand theory in Clifford analysis and tomography	2022	7	261.3
Story, Brittany M.	Molecular configurations and persistence: branched alkanes and additive energies	2022	7	152.6
Story, Dustin	Determining synchronization of certain classes of primitive groups of affine type	2022	6	134.3
Ziliak, Ellen	Arithmetic in group extensions using a partial automation	2010	9	115.9
Mechanical Engineering				
Grauberger, Alex Michael	Experimental investigation of an advanced organic Rankine vapor compression chiller	2022	6	258.8
Hobby, David Ryan	Generalized pressure drop and heat transfer correlations for jet impingement cooling with jet adjacent fluid extraction	2022	8	251.9
Hodgson, David A.	Investigation of a nonlinear controller that combines steady state predictions with integral action	2010	7	111.7
Overton-Katz, Nathaniel D.	Geometry considerations for high-order finite-volume methods on structured grids with adaptive mesh refinement	2022	6	250.3
Polak, Scott E.	Fourth-order finite volume algorithm with adaptive mesh refinement in space and time for multi-fluid plasma modeling, A	2022	7	118.7
Shah, Akash	Experimental and theoretical investigations of selenium graded cadmium telluride-based solar cells	2022	10	148.7
Stansloski, Mitchell	Application of force prediction to rotating equipment using pseudo-inverse techniques	2010	8	95.2
Walters, Sean	Large-eddy simulation of compressible flows using the stretched-vortex model and a fourth-order finite volume scheme on adaptive grids	2022	8	163.6
Wang, Yijun	Bayesian data assimilation for CFD modeling of turbulent combustion	2022	10	124.9

Author	Document Set (Dissertation)	Year	Number of Component Documents (Chapter Count)	Number of Sentences / Document
Zebhi, Banafsheh	Biomechanical analysis of hypoplastic left heart syndrome and calcific aortic stenosis: a statistical and computational study	2021	7	116.0
Microbiology, Immunology, and Pathology				
Baeten, Laurie Ann	Pathogenesis and immunological response of Yersina pestis in carnivores	2019	7	169.6
Butler, Molly	From Retroviridae to Flaviviridae: adventures in molecular virology	2021	6	195.5
Chiu, Elliott S.	Role of endogenous retrovirus in control of feline leukemia virus infection and implications for cross species transmission	2019	7	175.7
Doster, Enrique	Epidemiological investigation of antimicrobial resistance in beef production using metagenomic sequencing	2019	7	145.1
Gatlawi, Hana Bashir	Characterization of grcC1 and grcC2 prenyl diphosphate synthases potentially involved in menaquinone synthesis in Mycobacterium tuberculosis, and a homologous enzyme (ms1133) in Mycobacterium smegmatis	2021	7	147.1
Harris, Lauren	Molecular characterization of canine peripheral T-cell lymphoma	2020	6	234.3
Hoon-Hanks, Laura L.	Use of metagenomic sequencing as a tool for pathogen discovery with further investigation of novel reptilian serpentoviruses, The	2019	6	255.3
Kopanke, Jennifer H.	Characterizing the genetic evolution of endemic bluetongue virus strains	2019	6	160.0
Miller, Megan Rae	Assessment of mosquito and animal model factors in Aedes-borne arbovirus transmission and disease	2021	6	228.7
Rout, Emily	Clinical and molecular characterization of canine small cell B-cell lymphocytosis disorders	2020	6	199.2
Physics				
Bothwell, Alexandra	Development and advancement of thin CdTe-based solar cells for photovoltaic performance improvements	2020	6	264.5
Brandt, Adam D.	New measurement of the 2S1/2-8D5/2 transition in atomic hydrogen, A	2021	7	346.7
Ding, Jinjun	Damping and switching in thin films and hetero-structures of magnetic materials and topological materials	2020	9	190.5
Guthrie, John M.	Off-resonant RF heating of strongly magnetized electrons in ultracold neutral plasma	2021	6	329.8
Hester, Gavin L.	Quantum magnetism in the rare-earth pyrosilicates	2021	6	172.2
Loew, Kevin M.	Modeling and analysis of nanoscale surface patterns produced by broad beam ion bombardment	2020	8	209.4
Maughan, Weston F., II	Vortex rectification and phase slips in superconducting granular aluminum	2020	7	346.3

Author	Document Set (Dissertation)	Year	Number of Component Documents (Chapter Count)	Number of Sentences / Document
Pandey, Ramesh	Metal oxides as buffer layers in polycrystalline CdTe thin-film solar cells	2021	6	151.8
Sarkis, Colin L.	Frustration driven emergent phenomena in quantum and classical magnets	2021	6	243.8
Sutton, Logan	Fabrication and analysis of vanadium oxides and vanadium oxide based magnetic hybrid structures	2021	6	282.7
Political Science				
Angstadt, James Michael	Green courts and global norms: specialized environmental courts and the global governance of environmental challenges	2019	6	299.3
Bork, Nathanial	Failure to communicate how American progressive neoliberal campus policies contribute to conservative mistrust of higher education and skepticism towards research on anthropogenic global warming	2022	10	264.6
Cook, Jeffrey J.	Setting the record straight: interest group influence on climate policy at the Environmental Protection Agency	2017	6	275.8
DeCarlo, Chelsea Loren Welker	Re-imagining the ecological subject: toward a critical materialism of entangled ecologies	2019	7	418.0
Fisk, Jonathan M.	Fracking and Goldilocks Federalism: the too loud, too quiet and just right politics of states and cities	2015	7	339.9
Harwell, Janeane	Impacts of national security and sustainable development, The: comparative study of shared protected areas	2012	7	263.4
Hoffer, Katherine Anne Heriot	Policy innovation and change: the diffusion and modification of the renewable portfolio standard, 1994 – 2014	2018	7	296.3
Jedd, Theresa	Accountability and legitimacy in transboundary networked forest governance: a case study of the Roundtable on the Crown of the Continent	2015	8	356.9
Liebenguth, Julianne	Environmental security: a source of legitimacy and contestation in global environmental governance	2022	7	205.6
Nair, Sharmini	South Africa and India's support for the ILO's green initiatives: a comparative study using the postcolonialism lens	2022	10	420.3
Psychology				
Aeling, Jennifer	Hospice care: nurses' experience and perception of older adult patients' experiences	2018	9	108.9
Blalock, Lisa Durrance	Impact of long-term visual representations on consolidation in visual working memory	2010	6	197.7
Johnson, Ashlie N.	Latent profile analysis of intuitive eating behaviors related to wellbeing, eating behaviors, and physical activity during the early COVID-19 pandemic, A	2022	8	111.8
Manning, Steven G.	Appraising organizational politics and support: challenging employees to engage	2018	6	94.0
McDonald, James Ney	Relational maintenance in mixed-modality romantic relationships	2019	6	84.2

Author	Document Set (Dissertation)	Year	Number of Component Documents (Chapter Count)	Number of Sentences / Document
Pantlin, Lara N.	Mechanisms of timing: an integrative theoretical approach	2019	10	94.3
Raymer, Steven D.	Experiencing information: using systems theory to develop a theoretical framework of information interaction	2021	6	244.2
Sensenig, Amanda E.	Multiple choice testing and the retrieval hypothesis of the testing effect	2010	6	51.8
Stevens, Shalyn C.	Lower-wage workers and work-family social support: a qualitative study	2021	6	206.3
Walters, Kevin M.	Computational model and empirical study of the self-undermining proposition in job demands-resources theory, A	2019	6	148.0
Sociology				
Chesnais, Aude	Wolakota: the face of ReZilience in "post"-colonial America	2017	7	629.3
Gabriel, Jacqulyn S.	Manufacturing precarity: a case study of the Grain Processing Corporation/United Food and Commercial Workers Local 86D Lockout in Muscatine, Iowa	2016	7	446.7
Heller, Andrew	Why organizations matter: certification experiences of coffee producer groups in Guatemala	2010	7	366.4
Mayer, Adam	Risk, place and oil and gas policy preferences among Coloradoans	2017	6	267.7
Moloney, Christopher Jerome	Exploring the cybercrime capacity and capability of local law enforcement agencies in the United States	2021	10	351.9
Mordy, Meghan Katherine	Weighted aspirations: becoming a teenage dropout in El Salvador	2020	8	430.5
Rosty, Claudia Magalhaes	Fair Trade certified coffee estates: can Fair Trade USA promote workers' well-being, empowerment and gender equity in Brazilian and Nicaraguan coffee plantations?	2019	9	273.7
Shan, Yan	How universities participate in agricultural extension: a comparative study of two Chinese agricultural universities	2022	8	391.8
Smith, E. Keith	Beliefs, ideologies, contexts and climate change: the role of human values and political orientations in western European and transition states	2020	6	292.0
Tobin, Jennifer Lynn	Educational continuity following the 2013 Colorado Front Range Floods: a case study of Lyons elementary and middle/senior high schools	2019	7	330.1
Soil & Crop Sciences				
Catlett, Kathryn M.	Role of organic matter and other soil properties in Zn2+ activity and AB-DTPA-extractable Zn in soils, The	2000	6	231.0
Enjalbert, Jean-Nicolas	Integrated approach to local based biofuel development, An	2011	6	234.7

Author	Document Set (Dissertation)	Year	Number of Component Documents (Chapter Count)	Number of Sentences / Document
Magonziwa, Blessing	Understanding the dynamics and management of organic nutrient sources in smallholder farming systems: an interdisciplinary approach	2021	6	191.7
McDaniel, Jacob P.	Soil phosphorus availability and transformations following biosolids applications	2020	6	252.5
Prieksat, Mark Alan	Influence of soil hydraulic property estimation on the predictive accuracy of solute transport modeling, The	1999	6	268.5
Salley, Shawn William	Study of long-term soil moisture dynamics: assessing biologically available water as a function of soil development, A	2015	6	121.2
Sukor, Arina	Organic nitrogen fertilizers influence nutritional value, water use efficiency, and nitrogen dynamics of drip irrigated lettuce and sweet corn	2016	10	153.4
Toonsiri, Phasita	Effects of agricultural management on greenhouse gas emissions, carbon and nitrogen sequestration, and DAYCENT simulation accuracy	2017	6	149.2
Villalobos, Luis Alonso	Annual cool-season forage systems for fall grazing by cattle	2015	7	160.7
Widiastuti, Dwi P.	Azolla biofertilizer growth and utilization for vegetable production	2017	6	310.0
Statistics				
Cao, Meng	Statistical modeling and inference for complex-structured count data with applications in genomics and social science	2020	6	205.5
Chi, Jiarui	Sliced inverse approach and domain recovery for stochastic inverse problems	2021	6	252.5
Fix, Miranda J.	Advances in statistical analysis and modeling of extreme values motivated by atmospheric models and data products	2018	6	285.5
Fout, Alex M.	New methods for fixed-margin binary matrix sampling, Fréchet covariance, and MANOVA tests for random objects in multiple metric spaces	2022	6	226.2
Kim, Soo Young	Improved inference in heteroskedastic regression models with monotone variance function estimation	2018	6	175.2
Liao, Xiyue	Change-Point estimation using shape-restricted regression splines	2016	7	174.7
Roback, Paul J.	Pooling of prior distributions via logarithmic and supra-Bayesian methods with application to Bayesian inference in deterministic simulation models, The	1998	9	261.4
Vollmer, Charles T.	Statistical upscaling of stochastic forcing in multiscale, multiphysics modeling	2019	6	165.7
Weller, Zachary D.	Nonparametric tests of spatial isotropy and a calibration-capture-recapture model	2017	6	371.3
Yang, Lei	Infinite dimensional stochastic inverse problems	2018	7	164.6

Author	Document Set (Dissertation)	Year	Number of Component Documents (Chapter Count)	Number of Sentences / Document
Systems Engineering				
Anderson, Alexander A.	Systems engineering approach to community microgrid electrification and sustainable development in Papua New Guinea, A	2019	7	181.3
Ault, Trevor J.	Modernizing automation in industrial control/cyber physical systems through the system engineering lifecycle	2021	8	204.2
Azevedo, Kurt Milward	Improving construction machine engine system durability in Latin American conditions	2018	6	259.7
Biran, Yahav	Cloud Computing cost and energy optimization through Federated Cloud SoS	2017	7	228.6
Birch, Dustin Scott	Development of a human factors hazard model for use in system safety analysis	2021	11	89.6
Blondheim, David J., Jr.	System understanding of high pressure die casting process and data with machine learning applications	2021	6	653.8
Corrado, Jonathan K.	Technological advances, human performance, and the operation of nuclear facilities	2017	8	231.0
Creary, Andron Kirk	Systems engineering casualty analysis simulation (SE-CAS), The	2019	6	268.0
Davies, Augustus William	Methodology to enhance security of water utility system through RTU hardening	2022	8	137.9
Eaton, Christopher M.	Autonomous UAV control and testing methods utilizing partially observable Markov decision processes	2018	6	198.2
Gallagher, Brian P.	Using operational risk to increase systems engineering effectiveness	2016	9	128.8
Grassian, David	Modelling and analysis of systems on offshore oil and gas platforms	2019	6	262.2
Hung, Benjamin W. K.	Graph-based, systems approach for detecting violent extremist radicalization trajectories and other latent behaviors, A	2017	9	315.6
Jonkers, Raymond Klaas	Integration of systems engineering and project management using a management flight simulator	2020	12	235.0
Katz, Tami E.	Cost optimization in requirements management for space systems	2021	8	282.6
Kimbrough, Hal Reuben	Optimal sensor placement for sewer capacity risk management	2019	6	399.7
Kurtz, Jennifer	Innovative hydrogen station operation strategies to increase availability and decrease cost	2019	7	180.3
Lang, Daniel	Integrated optimization of composite structures	2022	7	160.9
LaSorda, Michael	Applying model-based systems engineering to architecture optimization and selection during system acquisition	2018	6	216.2
Lee, James Y.	System level risk analysis of electromagnetic environmental effects and lightning effects in aircraft -- steady state and transient	2017	8	221.2

Author	Document Set (Dissertation)	Year	Number of Component Documents (Chapter Count)	Number of Sentences / Document
Lunsford, Ian	Aircraft survivability modeling, evaluation, and optimization for multi-UAV operational scenarios	2021	8	124.0
Marzolf, Gregory S.	Systems engineering analysis and application to the Emergency Response System	2021	10	192.9
Meller, Ryan	Voltage reduction and automation on the residential distribution grid	2018	9	49.4
Miller, Andrew R.	Applying model-based systems engineering in search of quality by design	2022	6	396.0
Nelson, Travis J.	Balance of design methodology for enterprise quality attribute consideration in System-of-Systems architecting, A	2019	6	177.8
Othee, Avpreet	Modeling toolkit for comparing AC vs. DC electrical distribution efficiency in buildings, A	2021	9	98.6
Pirani, Badruddin	Combined classification and queuing system optimization approach for enhanced battery system maintainability, A	2022	6	238.8
Polidi, Danny Israel	Linking system cost model to system optimization using a cost sensitivity algorithm	2022	8	236.9
Roberts, Christopher J.	Space communications responsive to events across missions (SCREAM): an investigation of network solutions for transient science space systems	2022	6	297.0
Saripalli, Venkata Ratnam	Scalable and data efficient deep reinforcement learning methods for healthcare applications	2019	6	117.2
Scalco, Aleksandra	Measuring disagreement in segments of the cybersecurity profession as a means of identifying vulnerabilities	2022	13	153.0
Scheibmeir, Jim	Quality attributes of digital twins	2021	8	312.2
Siegel, Barry W.	Spatiotemporal anomaly detection: streaming architecture and algorithms	2020	12	143.7
Speece, Jill E.	Integrating MBSAP with continuous improvement for developing resilient healthcare systems	2021	7	86.3
Sturdivant, Rick L.	Application of systems engineering to complex systems and system of systems	2017	9	269.6
Sugama, Clive	System engineering for radio frequency communication consolidation with parabolic antenna stacking	2020	6	333.2
Vlajnic, Vanja M.	Machine learning and artificial intelligence approaches to the analysis of physical activity from wearables and biosensors in clinical trials: applications of clustering and prediction of clinical outcomes	2022	7	66.0
Walker, Joshua T.	Enhancing the test and evaluation process: implementing agile development, test automation, and model-based systems engineering concepts	2020	7	235.6
White, Wesley Gunnar	Modeling fuzzy criteria preference to evaluate tradespace of system alternatives	2018	9	125.1

Author	Document Set (Dissertation)	Year	Number of Component Documents (Chapter Count)	Number of Sentences / Document
Williams, Haney W.	Continuity of object tracking	2022	6	171.0
Younse, Paulo	Comparative analysis of model-based systems engineering and traditional systems engineering approaches for architecting robotic space systems through knowledge categorization, automatic information transfer, and automatic knowledge processing measures	2021	7	201.9