MODELING OF REASONING FLOWS IN SCIENTIFIC PUBLICATIONS

A Dissertation

by

JASON LIN

Submitted to the Office of Graduate and Professional Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

Chair of Committee,	Jyh-Charn Liu
Committee Members,	Thomas R. Ioerger
	Anxiao Jiang
	Adam Pickens
Head of Department,	Scott Schaefer

May 2020

Major Subject: Computer Engineering

ABSTRACT

Mathematical language plays an essential role in conceptualizing the technical contents of scientific publications. It applies words, symbols, and rules to constitute any sophisticated technical discussion. Existing technologies have achieved the recognition of mathematical objects (MOs) from digital documents, as well as the use of MOs and keywords to locate relevant resources. However, very few successful applications are on computer-based content analysis due to the obscured boundaries and semantics of technical contents. In this dissertation, we introduce the concept of reasoning block (RB) to mimic the divide-and-conquer of human writing and reading process. The RB model develops MO-based foundational solutions to address the challenges of reversing the original linear descriptions back to their logical non-linear structure.

A system model requires both the annotations of constraint expressions and textual declarations to enhance the mapping of problem settings and physical semantics. These two components highlight the information the readers need to know for the proposed system model of a paper. Reliable indicators such as mathematical symbols, stop words, and punctuations are used as features to distinguish constraint expressions from any other MO. We have investigated both a greedy approach based on the local optimal and a probabilistic approach based on Bayes' theorem in this study. As for mining the textual declarations of MOs, it requires to overcome the challenges of tagging, chunking, and pairing on the sentences mixed with words and MOs (MWM). We propose a second-order hidden Markov model and a frequent pattern mining toolkit for tagging and chunking the

MWM sentence, respectively. The final pairing of MOs and their declarations depend on the three-layer information (spatial, semantic, and syntactic) of the intermediate tokens that connect them.

Finally, the above analytical products are integrated and transform each publication into a hierarchical structure known as the MO reasoning (MOR) graph that consists of RBs in logical flows. Redundant MOs and their dependencies are removed based upon the minimum information required to cover all relations of MOs and words. The MOR graph is used as the technical essence to discover new forms of document fingerprint based on different writing styles in various domains.

DEDICATION

To My Family

ACKNOWLEDGEMENTS

I would first like to thank my advisor Dr. Jyh-Charn Liu, and my committee members, Dr. Thomas R. Ioerger, Dr. Anxiao Jiang, and Dr. Adam Pickens, for their support and guidance throughout this research.

I would also like to express the gratitude to all my colleagues in Real-Time Distributed Systems Laboratory at Texas A&M University, particularly to Dr. Xing Wang, Mr. Zelun Wang, Mr. Donald Beyette, and Mr. Colton Riedel for their assistance to my Ph.D. research. Both Xing and Zelun provided useful inputs to help me refine my proposed prediction models in this study. Donald implemented a web platform to help me visualize my research outcomes, which are present in parts of this dissertation. Colton helped proofread and provided useful comments to improve the clarity of the content. Besides, I would like to thank my friends Ryan Vrecenar and Kathy Pai, for helping me annotate the pilot datasets used in this dissertation. I sincerely appreciate their time in helping me with my research.

Thanks to all my friends and the department faculty and staff for making my time at Texas A&M University a great experience.

Finally, thanks to my father Chi-Nan Lin, my mother Meng-Chee Tan, and my younger brother Jared Lin for their love and encouragement.

CONTRIBUTORS AND FUNDING SOURCES

This work was supervised by a dissertation committee consisting of Professor Jyh-Charn Liu (advisor), Thomas R. Ioerger, and Anxiao Jiang of the Department of Computer Science and Engineering and Professor Adam Pickens of the Department of Environmental and Occupational Health.

All other work conducted for the dissertation was completed by the student independently.

There are no funding sources related to this dissertation.

NOMENCLATURE

CFG	Context Free Grammar
CMML	Content MathML
DMO	Displayed Mathematical Object
FP	False Positive
FN	False Negative
HMM	Hidden Markov Model
IID	Independent and Identically Distributed
IMO	Inline Mathematical Object
LHS	Left-Hand Side
LDA	Latent Dirichlet Analysis
LSA	Latent Semantic Analysis
MathML	Mathematical Markup Language
MIR	Mathematical Information Retrieval
MO	Mathematical Object
MOR	Mathematical Object Reasoning
MWM	Mixed with Words and MOs
NCS	Non-separated Character Sequence
NLP	Natural Language Processing
NML	Noun Modifier
NP	Noun Phrase

NTCIR	NII Testbeds and Community for Information access Research
OCR	Optical Character Recognition
PCFG	Probabilistic Context Free Grammar
PDF	Portable Document Format
PMI	Pointwise Mutual Information
PMML	Presentation MathML
POS	Part-of-Speech
QuQn	Qualitative-Quantitative
RB	Reasoning Block
RHS	Right-Hand Side
STEM	Science, Technology, Engineering, and Mathematics
SVM	Support Vector Machine
TF-IDF	Term Frequency-Inversed Document Frequency
TN	True Negative
TP	True Positive
XML	Extensible Markup Language

TABLE OF CONTENTS

Page

ABSTRACT ii

DEDICATION iv

ACKNOWLEDGEMENTS v

CONTRIBUTORS AND FUNDING SOURCES vi

NOMENCLATURE vii

TABLE OF CONTENTS ix

LIST OF FIGURES xii

LIST OF TABLES xvi

CHAPTER I INTRODUCTION 1

Motivation 1
Objective and Challenges 2
System Model 6
Organization of the Dissertation 11

CHAPTER II PRELIMINARIES 12

Electronic Documents 12
Extraction of Technical Elements 13
Technical Content Analysis 16
Overview of the Datasets 18

CHAPTER III CONSTRAINT EXPRESSION OF MATHEMATICAL OBJECT 23

Overview 23
Related Works 25
Features of MOC Expressions 25
Feature Extraction 26
Analysis of Mathematical Symbol Features 27
Analysis of Contextual Word Features 29
Comparison of Mathematical and Textual Features 31

Prediction of Constraint Expression 32
Experimental Results and Discussion 34
Summary 35

CHAPTER IV TEXTUAL DECLARATION OF MATHEMATICAL OBJECT 37

Overview 38
Related Works 41
Mathematical Objects and Words Mixed Tagger 43
The Syntactic Role of Mathematical Object 43
The Syntactic Tagger for MWM Sentences 49
Noun Phrase Extraction in MWM Sentences 51
Features of MOD Predictions 52
Feature Extraction 53
Analysis of Spatial Feature 55
Analysis of Word Feature 58
Bonding of Mathematical Objects and Noun Phrases 63
Template Pattern Matching 64
Naïve Bayesian Inference 65
Majority Vote Ensemble 67
Experimental Results and Analysis 67
Dataset and Evaluation Criteria 68
Result and Discussion 68
Summary 70

CHAPTER V REASONING GRAPH OF MATHEMATICAL OBJECT 72

Overview 72
Related Works 74
Revisit the QuQn Map System 76
Mathematical Object Processing 76
Notation Definitions 77
Denotation Extraction 77
Skeleton Graph Construction and Pruning 78
Visualization of the QuQn Map 80
The Limitation of QuQn Map 82
Construction of the MOR Graph 83
Segmentation of Reasoning Blocks 84
Dependency Analysis of Mathematical Objects 87
Reduction of the MOR Graph 92
Visualization of the MOR Graph 96
Applications of the MOR Graph 97
The Cross-paper Analysis Approach 99
Use Case 1: Detail Analysis based on Technical Elements 100

Use Case 2: Text-based Document Clustering 101
Use Case 3: Structure-level Differential Analysis 107
Summary 114

CHAPTER VI CONCLUSIONS 115

REFERENCES 119

APPENDIX A PART-OF-SPEECH (POS) TAGS 133

APPENDIX B THE DATA STRUCTURE OF MATHEMATICAL SEMANTICS 134

APPENDIX C A CASE STUDY OF PARAGRAPH EXTRACTION 135

APPENDIX D SOME COMMON LATEX CODES 136

APPENDIX E SOME COMMON MATHEMATICAL TERMS 137

APPENDIX F HISTOGRAMS OF MATHEMATICAL OBJECT PROPERTIES 138

LIST OF FIGURES

Page

Figure 1: The growing trend of academic publishing. 1

Figure 2: The conceptual reasoning block model for a segment of content in [8]. 3

Figure 3: The characteristics of the boundary of reasoning blocks and the main equations. 7

Figure 4: The system architecture of the reasoning block model and the four research tasks (RT1-RT4). 8

Figure 5: The three-tier analytical framework for human reading process. 9

Figure 6: The information layer and their key features required to model and predict mathematical object related entities..10

Figure 7: The data structures of mathematical objects and words obtained from the PDF parser. 14

Figure 8: The labeling of metadata in a research paper [35]..17

Figure 9: The construction of QuQn map on a paragraph in arXiv document 1605.02019. Reprinted with permission from [33]. 18

Figure 10: An example of the dataset OA-STM Corpus for syntactic role and constraint annotation of mathematical objects (MOs) in STEM fields. 20

Figure 11: An example of the NTCIR-10 dataset for the textual declaration of mathematical object. 21

Figure 12: An example for the mathematical symbol extraction. Reprinted with permission from [50]. 26

Figure 13: The statistical likelihood of strong indicative symbols used for recognizing the constraint expressions. 28

Figure 14: The statistical likelihood of weak indicative symbols used for recognizing the constraint expressions. 28

Figure 15: The statistical likelihood of word stems for introducing the constraint expressions in a sentence. Reprinted with permission from [50]. 29

Figure 16: The statistical likelihood of POS tags for introducing the constraint expressions in a sentence. Reprinted with permission from [50]. 30

Figure 17: The sample density of MOC expression likelihood. Reprinted with permission from [50]. 32

Figure 18: The three-level framework for MOD prediction. Reprinted with permission from [58]. 39

Figure 19: The system architecture for the bonding prediction of mathematical objects (MOs) and their related noun phrases (NPs). Reprinted with permission from [58]. 41

Figure 20: The error propagation from POS tagging in L1 to constituent parsing (i.e., Fail to identify “prime” as the declaration of MO_1). Reprinted with permission from [58]. 45

Figure 21: An example of LaTeX code and its corresponding Presentation MathML. ...46

Figure 22: Histograms of syntactic roles on the (a) length and (b) depth of mathematical object (MO) in MathML structure. 47

Figure 23: Histograms of syntactic roles on the number of (a) mathematical identifiers and (b) operators. 48

Figure 24: The visualization of process model in Disco [79] using POS tag series of MOD (top 10% frequent nodes and links from the data). Reprinted with permission from [58]. 51

Figure 25: The modeling of spatial, semantic, and syntactic features in a Halo. Reprinted with permission from [58]. 53

Figure 26: An example sentence (highlighted in blue) for negative sampling (green bold line connects MO to its true declaration, while red dashed line connects to a false declaration). Reprinted with permission from [58]. 55

Figure 27: Histograms of the positive (a) and the negative (b) instances for MOD based on the spatial distances. MO is a mathematical object where Dec is its declaration while Non-Dec is not. Reprinted with permission from [58]. 56

Figure 28: The statistical likelihood of spatial distance between the mathematical object (MO) and its related textual declaration (Dec). Reprinted with permission from [58]. 57

Figure 29: Histograms of the positive (a) and the negative (b) instances for MOD based on the word stems. Notice that -NONE- represents an empty token, MO is a mathematical object where Dec is its declaration while Non-Dec is not. Reprinted with permission from [58]. 59

Figure 30: The statistical likelihood of word stem for intermediate words that connect the mathematical object (MO) to its related textual declaration (Dec). Reprinted with permission from [58]. 60

Figure 31: Histograms of the positive (a) and the negative (b) instances for MOD based on the POS tags. Notice that -NONE- represents an empty token, MO is a mathematical object where Dec is its declaration while Non-Dec is not. Reprinted with permission from [58]. 62

Figure 32: The statistical likelihood of POS tag for intermediate words that connect the mathematical object (MO) to its related textual declaration (Dec). Reprinted with permission from [58]. 63

Figure 33: The graph skeleton of QuQn map on arXiv document 1605.02019. 80

Figure 34: The graph visualization of QuQn map on arXiv document 1605.02019 using the color wheel to highlight the components. 82

Figure 35: A conceptualized structure of reasoning graph for arXiv document 1605.02019 (red dashed lines denote the blocks and flows). 84

Figure 36: The boundary of reasoning blocks (purple dashed line) with a threshold of allowing four consecutive sentences without MO. 86

Figure 37: The threshold $\boldsymbol{\tau}$ that impacts (a) the number of reasoning blocks created; and (b) the number of mathematical objects in the first reasoning block. 87

Figure 38: The identifier set of a mathematical object. 88

Figure 39: The occurrences of identifier sets in arXiv document 1605.02019. 90

Figure 40: The two types of dependencies link based on (a) the left-hand side of MO or (b) the whole MO, is a subcomponent of the MO pointed. 91

Figure 41: Histograms of the four properties of main equation from arXiv document 1605.02019. 94

Figure 42: The visualization of reasoning graph for arXiv document 1605.02019. 97

Figure 43: Case study on cross paper analysis for one of the paragraphs in paper [35] and [112]. 101

Figure 44: The bipartite graph for arXiv research fields and mathematical text clustering......103

Figure 45: The bipartite graph for arXiv research fields and pure text clustering. 104

Figure 46: The bipartite graph for pure text clustering and mathematical text clustering......106

Figure 47: The process of serializing the MOR graph into Prüfer sequence based on its spanning tree. 108

Figure 48: The process of serializing the dependency graph of mathematical objects into degree sequence based on the number of associated mathematical constraints. 109

Figure 49: Similarity matrices of documents in three research fields (‘L’, ‘G’, ‘E’) based on the structural differences of (a) dependency graph and (b) reasoning graph of mathematical objects. 111

Figure 50: Dendrograms of document clustering results in three research fields (‘L’, ‘G’, ‘E’) based on the structural differences of (a) dependency graph and (b) reasoning graph of mathematical objects. 112

Figure 51: The bipartite graph for research fields and mathematical text clustering.....113

Figure 52: The bipartite graph for research fields and pure text clustering. 114

Figure 53: The semantic taxonomy of mathematical object [88]......134

Figure 54: The preliminary results of paragraph extraction based on the content layout (in green rectangles) 135

Figure 55: The histograms of the frequency of MOs and the MO density sampled from 50 documents of the KDD Cup 2003 dataset......138

Figure 56: The histograms of the number of associated MOs and the MO length sampled from 50 documents of the KDD Cup 2003 dataset 139

LIST OF TABLES

Page

Table 1: List of existing datasets for technical content analysis. 19

Table 2: The average performance of recognizing the constraint expressions from mathematical objects. Reprinted with permission from [50]...........................35

Table 3: The syntactic role of mathematical object (MO-SR) and examples. Reprinted with permission from [58]. 44

Table 4: POS tagging prediction performance on MWM sentences in the dataset OA-STM Corpus [36]. Reprinted with permission from [58]. 50

Table 5: A list of common template patterns collected from arxiv.org to relate mathematical objects (MOs) to their textual declaration (Dec). Reprinted with permission from [58]. 65

Table 6: The average performance of mining the textual declarations of mathematical objects. Reprinted with permission from [58]. 69

Table 7: The assessment score of identifying the consecutive multiple identifiers. 90

Table 8: A primitive evaluation of equations rank on arXiv document 1605.02019. 96

Table 9: The English Penn Treebank POS tags [116]..133

Table 10: A list of reserved words in LaTeX for different types of mathematical symbols 136

Table 11: A list of mathematical terms used in different mathematical fields 137

CHAPTER I

INTRODUCTION

Motivation

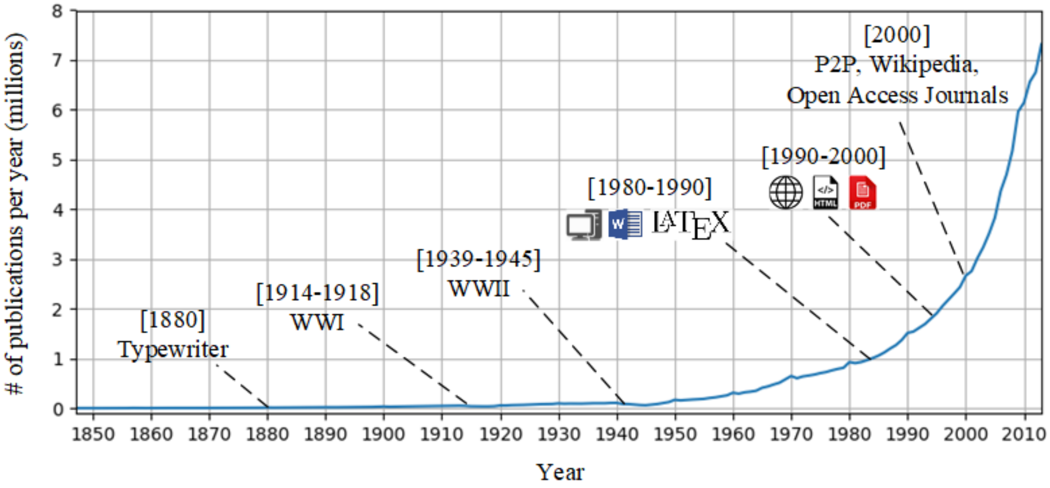


Figure 1: The growing trend of academic publishing.

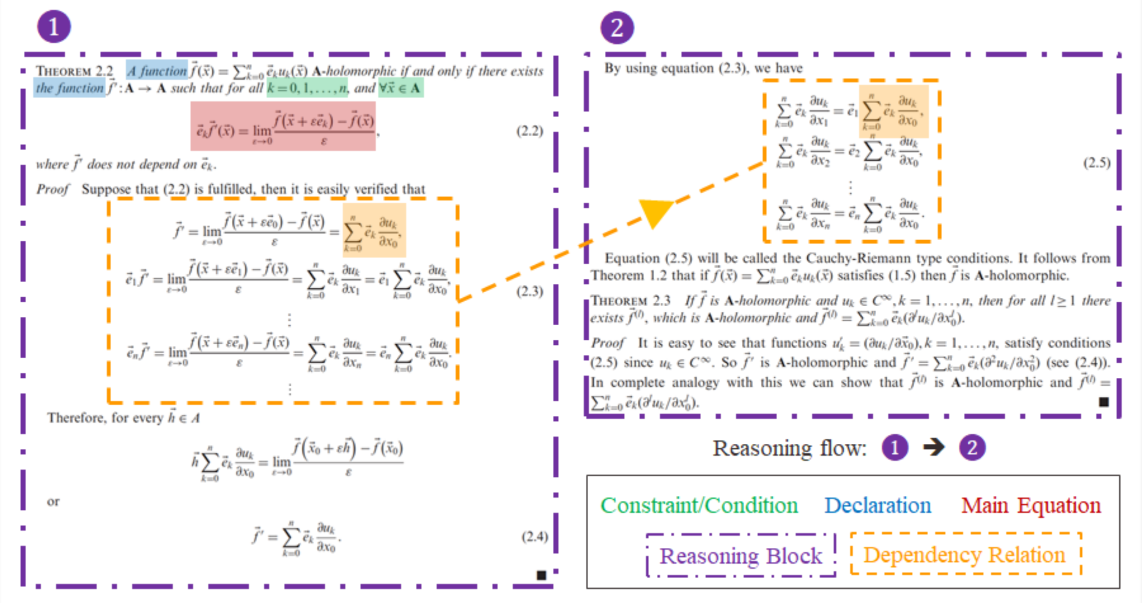
In recent decades, the volume of academic publications has grown rapidly [1] as shown in Figure 1, largely due to advances in information technology. It has become a burden for scientists and researchers to effectively consume the vast amount of papers within a limited time frame. Approximately 62% of publications (79 million out of 127 million) are in science, technology, engineering and mathematics (STEM) fields according to the investigation of Microsoft Academic database [2]. Unlike consumer market-based information indexing and retrieval systems [3], [4], STEM publications are written to cover major aspects regarding technical issues being studied, and deliver focused technical themes in a semi-structured manner based on common practice of the respective technical community. Technologies such as search engines [5] and citation tools [6] have led to very large-scale digital library systems for indexing, searching, and locating of intellectual work. With the maturity of information management solutions [7], research in deep content analysis have also gained more attentions.

Mathematical objects (MOs) and words are two major components that constitute the technical contents of scientific publications. MO, once defined, are treated as a new form of token similar to other words in a paper, and its constraints and declarations which characterizing the quantitative and qualitative attributes, respectively, are quite often expressed by surrounding tokens. These MO-to-MO and MO-to-words tuples are often of technical significance to the community and can be viewed as a description of the work's technical elements. By mining of these tuples at large scale, the most significant technical elements can be extracted as a new form of high-level concise abstractions for the technical content.

Objective and Challenges

Writing a scientific paper can be characterized as a process of organizing sophisticated technical issues into self-contained reasoning blocks (RBs), and with aid of sectioning, the complex relationships among technical elements are presented in a set of linear sequences. Conversely, in reading the same paper, a reader must identify RBs, and use explicit and implicit links among elements in RBs to understand technical issues. We observe that most RBs are built upon MOs, using them to quantify system behaviors based on rigorous notations and elaborations. As such, in this research, an MO centric content analysis framework is proposed, based on the hypothesis that symbols and MOs often carry the most prominent and sophisticated abstractions, and authors must carefully construct their reasoning flows around them.

It is relatively easy for a human reader to manually delineate RBs such as the handcrafted example ❶ and ❷ illustrated in Figure 2, though individuals may have different interpretations regarding scopes and boundaries of RBs. Yet, this is only the beginning of a very complex process in order to understand the technical essence of a paper. Through a divide-and-conquer approach and the use of RBs to scope and interrelate algorithmic analysis and representations of technical contents, this dissertation explores fundamental issues in automating aspects of deep content analysis.



THEOREM 2.2 *A function* $\vec{f}(\vec{x}) = \sum_{k=0}^{n} \vec{e}_k u_k(\vec{x})$ **A**-*holomorphic if and only if there exists the function* $\vec{f}' : \mathbf{A} \to \mathbf{A}$ *such that for all* $k = 0, 1, \ldots, n$, *and* $\forall \vec{x} \in \mathbf{A}$

$$\vec{e}_k \vec{f}'(\vec{x}) = \lim_{\varepsilon \to 0} \frac{\vec{f}(\vec{x} + \varepsilon \vec{e}_k) - \vec{f}(\vec{x})}{\varepsilon}, \tag{2.2}$$

where $\vec{f}'$ *does not depend on* $\vec{e}_k$.

Proof Suppose that (2.2) is fulfilled, then it is easily verified that

$$\begin{aligned} \vec{f}' &= \lim_{\varepsilon \to 0} \frac{\vec{f}(\vec{x} + \varepsilon \vec{e}_0) - \vec{f}(\vec{x})}{\varepsilon} = \sum_{k=0}^{n} \vec{e}_k \frac{\partial u_k}{\partial x_0}, \\ \vec{e}_1 \vec{f}' &= \lim_{\varepsilon \to 0} \frac{\vec{f}(\vec{x} + \varepsilon \vec{e}_1) - \vec{f}(\vec{x})}{\varepsilon} = \sum_{k=0}^{n} \vec{e}_k \frac{\partial u_k}{\partial x_1} = \vec{e}_1 \sum_{k=0}^{n} \vec{e}_k \frac{\partial u_k}{\partial x_0}, \\ &\vdots \\ \vec{e}_n \vec{f}' &= \lim_{\varepsilon \to 0} \frac{\vec{f}(\vec{x} + \varepsilon \vec{e}_n) - \vec{f}(\vec{x})}{\varepsilon} = \sum_{k=0}^{n} \vec{e}_k \frac{\partial u_k}{\partial x_n} = \vec{e}_n \sum_{k=0}^{n} \vec{e}_k \frac{\partial u_k}{\partial x_0}. \\ &\vdots \end{aligned} \tag{2.3}$$

Therefore, for every $\vec{h} \in A$

$$\vec{h} \sum_{k=0}^{n} \vec{e}_k \frac{\partial u_k}{\partial x_0} = \lim_{\varepsilon \to 0} \frac{\vec{f}\left(\vec{x}_0 + \varepsilon \vec{h}\right) - \vec{f}(\vec{x}_0)}{\varepsilon}$$

or

$$\vec{f}' = \sum_{k=0}^{n} \vec{e}_k \frac{\partial u_k}{\partial x_0}. \tag{2.4}$$

■

By using equation (2.3), we have

$$\begin{aligned} \sum_{k=0}^{n} \vec{e}_k \frac{\partial u_k}{\partial x_1} &= \vec{e}_1 \sum_{k=0}^{n} \vec{e}_k \frac{\partial u_k}{\partial x_0}, \\ \sum_{k=0}^{n} \vec{e}_k \frac{\partial u_k}{\partial x_2} &= \vec{e}_2 \sum_{k=0}^{n} \vec{e}_k \frac{\partial u_k}{\partial x_0}, \\ &\vdots \\ \sum_{k=0}^{n} \vec{e}_k \frac{\partial u_k}{\partial x_n} &= \vec{e}_n \sum_{k=0}^{n} \vec{e}_k \frac{\partial u_k}{\partial x_0}. \end{aligned} \tag{2.5}$$

Equation (2.5) will be called the Cauchy-Riemann type conditions. It follows from Theorem 1.2 that if $\vec{f}(\vec{x}) = \sum_{k=0}^{n} \vec{e}_k u_k(\vec{x})$ satisfies (1.5) then $\vec{f}$ is **A**-holomorphic.

THEOREM 2.3 *If* $\vec{f}$ *is* **A**-*holomorphic and* $u_k \in C^\infty, k = 1, \ldots, n$, *then for all* $l \geq 1$ *there exists* $\vec{f}^{(l)}$, *which is* **A**-*holomorphic and* $\vec{f}^{(l)} = \sum_{k=0}^{n} \vec{e}_k (\partial^l u_k / \partial x_0^l)$.

Proof It is easy to see that functions $u_k' = (\partial u_k / \partial \vec{x}_0), k = 1, \ldots, n$, satisfy conditions (2.5) since $u_k \in C^\infty$. So $\vec{f}'$ is **A**-holomorphic and $\vec{f}' = \sum_{k=0}^{n} \vec{e}_k (\partial^2 u_k / \partial x_0^2)$ (see (2.4)). In complete analogy with this we can show that $\vec{f}^{(l)}$ is **A**-holomorphic and $\vec{f}^{(l)} = \sum_{k=0}^{n} \vec{e}_k (\partial^l u_k / \partial x_0^l)$. ■

Figure 2: The conceptual reasoning block model for a segment of content in [8].

Succinctly put, this study identifies the following technical challenges and propose new modeling techniques and associated algorithms.

1. The boundary detection of RB: there are no explicit tag defined thus far in any document system to support such boundary and existence of RB.
2. The classification of technical elements in RB: what are the necessary types of technical elements in RB for understanding the content of RB?
3. The relational properties of the technical elements in RB: what are the features we can use from the relations of technical elements to study the characteristics of different technical elements?
4. The technical elements that are important compared to others in the RB: how do we reduce the amount of information needed from the technical elements to efficiently grasp the technical essence of the raw content?
5. The potential of reasoning flows in document fingerprinting: can we leverage the proposed reasoning block model to characterize the writing styles of the scientific publications for applications in the cross-paper analysis?

To address the above five challenges, useful observations and facts are identified to aid in modeling.

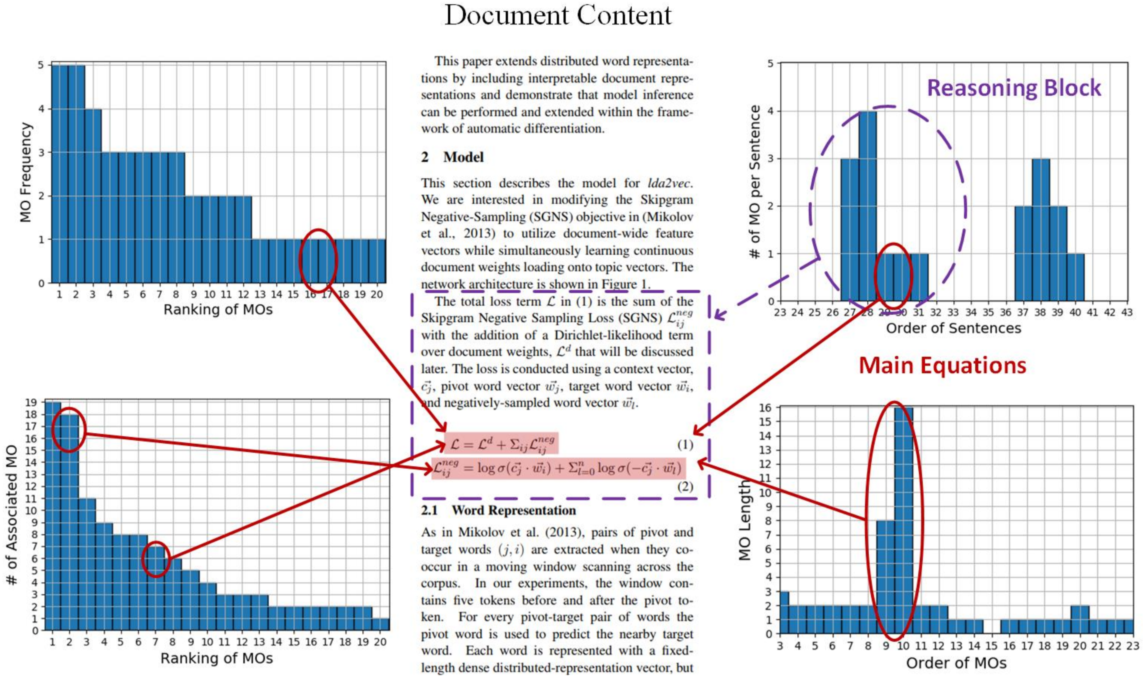
1. In technical writing, MOs are highly expressive, compact, and precise; so that researchers can effectively use them to covey sophisticated concepts and relationships. Even when the sentence sequences are not explicitly tagged, they can be used to segment the semantics of the reasoning flows in scientific papers.

2. It is necessary to use the textual declarations of MOs to elaborate sophisticated concepts in a particular technical domain and followed by constraint expressions to assign them domain-specific semantics. Otherwise, MO alone is purely mathematical abstractions without semantic significance for a technical subject. Being able to automatically detect these semantic bonding of MO-to-MO and MO-to-words is of great importance to deep content analysis.
3. MOs are implicitly connected through their common identifiers such as variables, indices, or function names. The associated MOs provide intermediate results or constraints/conditions of the MO that they pointed to.
4. The importance of MO can be inferred by criteria used in different weights that defines the ranking of importance such as the spacing in the context, the size of MO, the number of associated MOs, and the type of associativity such as the convergence/divergence point.
5. A graphical structure can be defined through the RBs with its technical elements and their relations. The topological properties of the graph are a way to reflect the style of the author composing technical contents.

The primary goal of this dissertation is to address the following issues, which are critical for the computer-based detections and the modeling of reasoning flows: (1) Segmentation on the original content of a paper into reasoning blocks; (2) Recognizing symbols and/or connecting words that are strongly indicative for the MO constraint expressions; (3) Detection of MOs and their coreference words as the textual declarations of MOs; (4) Establishing the connectivity and dependency between technical elements. We will leverage the final product to perform large-scale studies in STEM fields.

System Model

By definition, an RB is a region of MOs surrounded by closely related words to form a localized, self-contained technical concept. Typically, RBs are centered around one or a few large equations, together with their constraint expressions, and optionally additional explanation of details. Most RBs are approximately aligned with paragraphs, yet some others are segmented by the sentence semantics. It is reasonable to assume that RBs are clustered around MOs so that the local density of MOs, can be readily used to segment MOs, as shown in the purple dash lines of Figure 3. The main equations of RB are defined based on four MO features, including the occurrence of MO throughout the content, the local density of MOs, the dependency on other MOs, and the MO length as addressed in the red bold lines of Figure 3.



This paper extends distributed word representations by including interpretable document representations and demonstrate that model inference can be performed and extended within the framework of automatic differentiation.

2 Model

This section describes the model for *lda2vec*. We are interested in modifying the Skipgram Negative-Sampling (SGNS) objective in (Mikolov et al., 2013) to utilize document-wide feature vectors while simultaneously learning continuous document weights loading onto topic vectors. The network architecture is shown in Figure 1.

The total loss term $\mathcal{L}$ in (1) is the sum of the Skipgram Negative Sampling Loss (SGNS) $\mathcal{L}_{ij}^{neg}$ with the addition of a Dirichlet-likelihood term over document weights, $\mathcal{L}^d$ that will be discussed later. The loss is conducted using a context vector, $\vec{c_j}$, pivot word vector $\vec{w_j}$, target word vector $\vec{w_i}$, and negatively-sampled word vector $\vec{w_l}$.

$$\mathcal{L} = \mathcal{L}^d + \Sigma_{ij}\mathcal{L}_{ij}^{neg} \quad (1)$$

$$\mathcal{L}_{ij}^{neg} = \log\sigma(\vec{c_j}\cdot\vec{w_i}) + \Sigma_{l=0}^{n}\log\sigma(-\vec{c_j}\cdot\vec{w_l}) \quad (2)$$

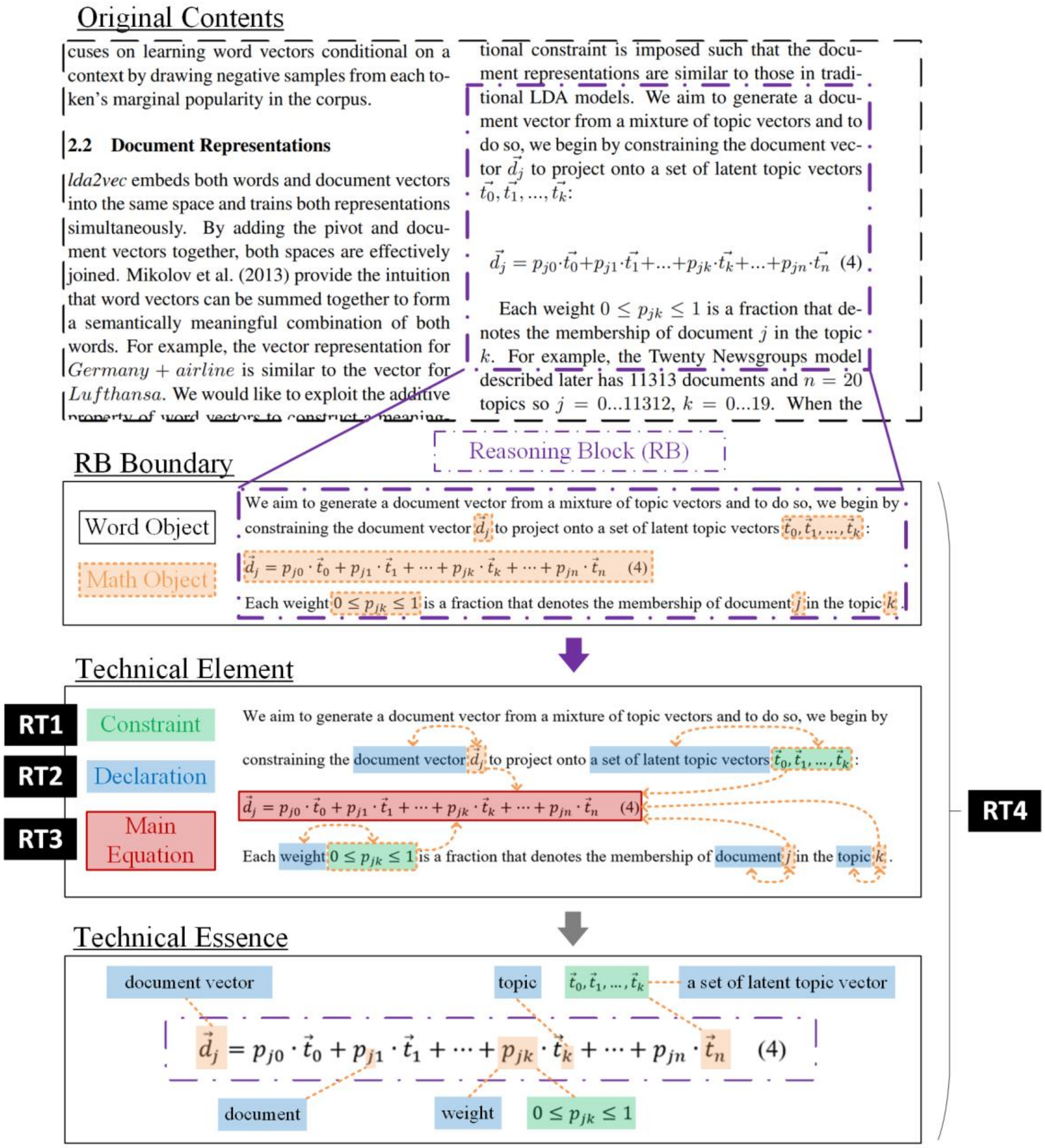
2.1 Word Representation

As in Mikolov et al. (2013), pairs of pivot and target words (j, i) are extracted when they co-occur in a moving window scanning across the corpus. In our experiments, the window contains five tokens before and after the pivot token. For every pivot-target pair of words the pivot word is used to predict the nearby target word. Each word is represented with a fixed-length dense distributed-representation vector, but

Figure 3: The characteristics of the boundary of reasoning blocks and the main equations.

On the basis of the RB model, this dissertation is organized to perform four major research tasks (RT1 to RT4), which are illustrated in Figure 4, with respect to constraint expressions, textual declarations, and main equations of MOs. Specifically, RT1 focuses on developing the prediction model for MOs that contain constraint semantics, and RT2 deals with the prediction model for mapping MOs to their related words. RT3 ranks the equations based on the content flow and the associated MO that formulate significant technical discussions of a paper. A graphical skeleton can be constructed to highlight the technical essence of an RB based on the results derived in RT1-RT3. The main focus of

RT4 is a case study on cross-paper analysis by using graph structures of MOs and their related words as the document fingerprints.



cuses on learning word vectors conditional on a context by drawing negative samples from each token's marginal popularity in the corpus.

2.2 Document Representations

lda2vec embeds both words and document vectors into the same space and trains both representations simultaneously. By adding the pivot and document vectors together, both spaces are effectively joined. Mikolov et al. (2013) provide the intuition that word vectors can be summed together to form a semantically meaningful combination of both words. For example, the vector representation for $Germany + airline$ is similar to the vector for $Lufthansa$. We would like to exploit the additive property of word vectors to construct a meaning-

tional constraint is imposed such that the document representations are similar to those in traditional LDA models. We aim to generate a document vector from a mixture of topic vectors and to do so, we begin by constraining the document vector $\vec{d_j}$ to project onto a set of latent topic vectors $\vec{t_0}, \vec{t_1}, ..., \vec{t_k}$:

$$\vec{d_j} = p_{j0} \cdot \vec{t_0} + p_{j1} \cdot \vec{t_1} + ... + p_{jk} \cdot \vec{t_k} + ... + p_{jn} \cdot \vec{t_n} \quad (4)$$

Each weight $0 \le p_{jk} \le 1$ is a fraction that denotes the membership of document j in the topic k. For example, the Twenty Newsgroups model described later has 11313 documents and $n = 20$ topics so $j = 0...11312$, $k = 0...19$. When the

We aim to generate a document vector from a mixture of topic vectors and to do so, we begin by constraining the document vector $\vec{d_j}$ to project onto a set of latent topic vectors $\vec{t_0}, \vec{t_1}, \ldots, \vec{t_k}$:

$$\vec{d_j} = p_{j0} \cdot \vec{t_0} + p_{j1} \cdot \vec{t_1} + \cdots + p_{jk} \cdot \vec{t_k} + \cdots + p_{jn} \cdot \vec{t_n} \quad (4)$$

Each weight $0 \le p_{jk} \le 1$ is a fraction that denotes the membership of document j in the topic k.

We aim to generate a document vector from a mixture of topic vectors and to do so, we begin by constraining the document vector $\vec{d_j}$ to project onto a set of latent topic vectors $\vec{t_0}, \vec{t_1}, \ldots, \vec{t_k}$:

$$\vec{d_j} = p_{j0} \cdot \vec{t_0} + p_{j1} \cdot \vec{t_1} + \cdots + p_{jk} \cdot \vec{t_k} + \cdots + p_{jn} \cdot \vec{t_n} \quad (4)$$

Each weight $0 \le p_{jk} \le 1$ is a fraction that denotes the membership of document j in the topic k.

RT4
Technical Essence
document vector
topic
$\vec{t_0}, \vec{t_1}, \ldots, \vec{t_k}$
a set of latent topic vector
$\vec{d_j} = p_{j0} \cdot \vec{t_0} + p_{j1} \cdot \vec{t_1} + \cdots + p_{jk} \cdot \vec{t_k} + \cdots + p_{jn} \cdot \vec{t_n}$ (4)
document
weight
$0 \le p_{jk} \le 1$

Figure 4: The system architecture of the reasoning block model and the four research tasks (RT1-RT4).

For algorithm design, we follow a three-tier analytical framework including symbol, layout, and semantics to mimic the human reading process as shown in Figure 5. When reading a scientific paper, we as human readers first identify the symbol values such as MOs and words in the content. Then, based on the layout of these symbols, we sequentially analyze the writing intent and relationships among them to infer semantics of the technical discourse.

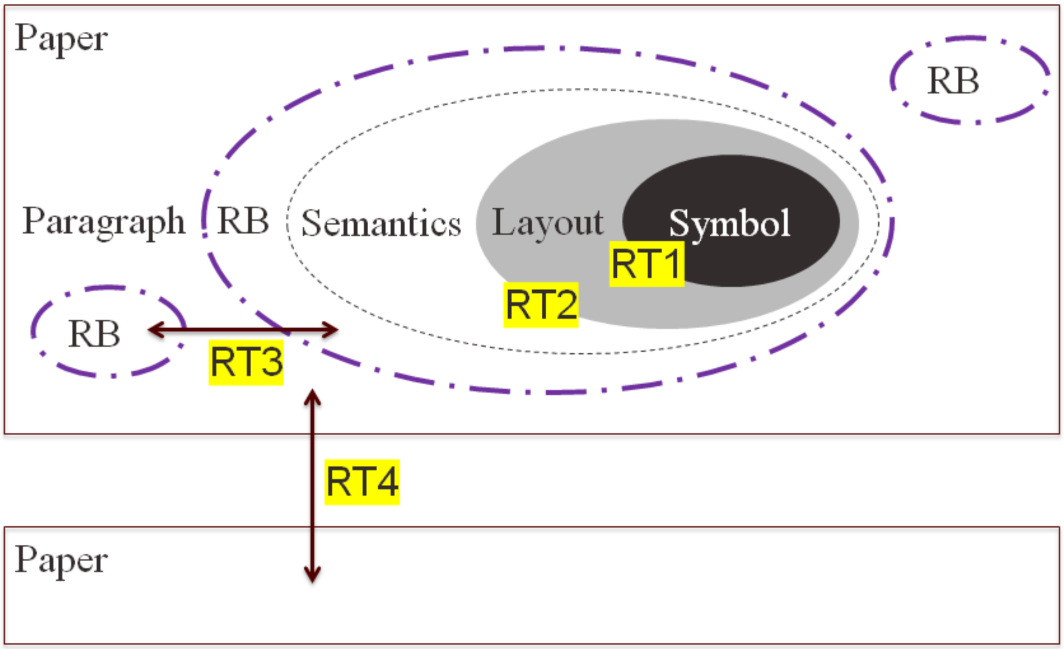


Figure 5: The three-tier analytical framework for human reading process.

For feature analysis, three-layers of information (spatial, syntactic, and semantic) are used to model the classification of technical elements as shown in the natural language interface of Figure 6. Spatial analysis is how the linguistic unit distance (e.g., number of words) is used to reason about the coreference relation between tokens based on their relatively position (i.e., left-hand side and right-hand side). The syntactic analysis refers to the grammatical functions of tokens which imply their usage in influencing neighboring tokens. The semantic analysis is the property of stop/root words, punctuation, and strong reserved mathematical symbols that carry significant meaning in deriving or introducing a technical element. The mathematical language interface in Figure 6 has manifest the three types of technical elements: constraints, declarations, and main equations, with respect to their associated words/MOs in the context.

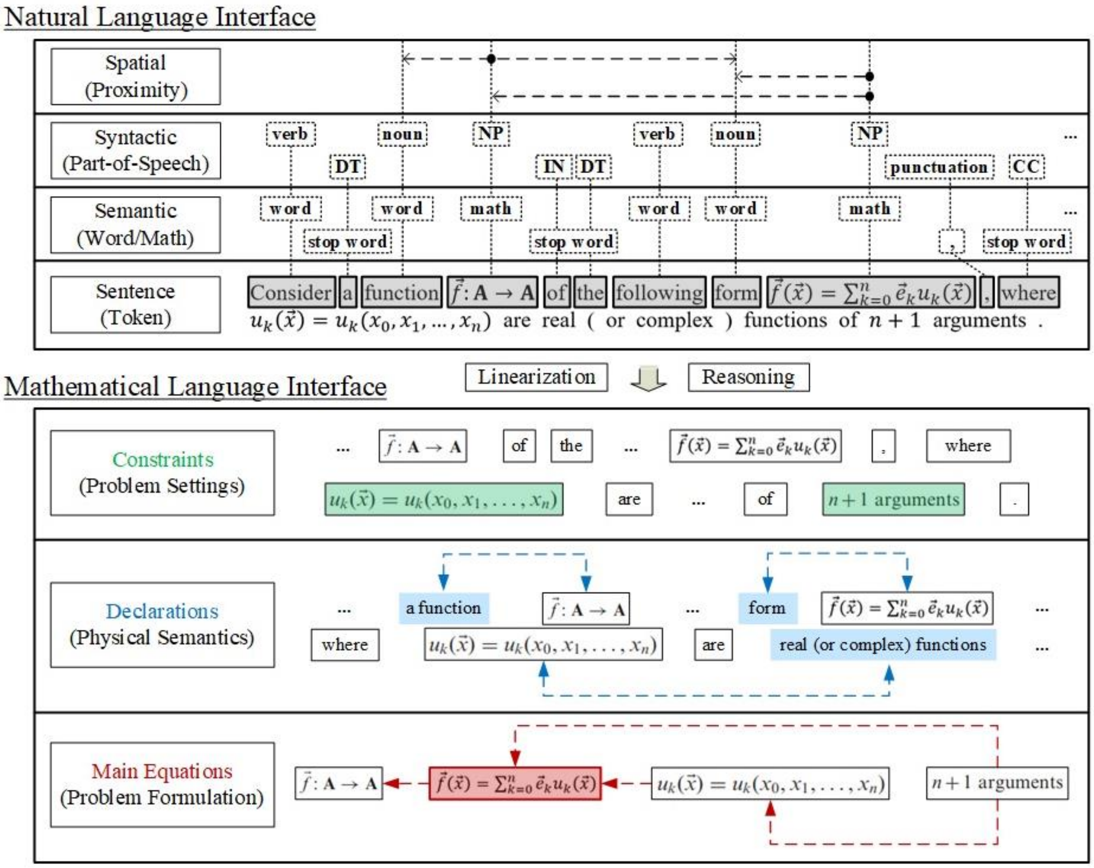

Natural Language Interface
Spatial (Proximity)
Syntactic (Part-of-Speech)
verb
noun
NP
DT
IN
DT
verb
noun
NP
punctuation
CC
...
Semantic (Word/Math)
word
word
math
stop word
stop word
word
word
math
,
stop word
...
Sentence (Token)
Consider a function $\vec{f}: \mathbf{A} \to \mathbf{A}$ of the following form $\vec{f}(\vec{x}) = \sum_{k=0}^{n} \vec{e}_k u_k(\vec{x})$, where
$u_k(\vec{x}) = u_k(x_0, x_1, \ldots, x_n)$ are real (or complex) functions of $n+1$ arguments .
Mathematical Language Interface
Linearization
Reasoning
Constraints (Problem Settings)
... $\vec{f}: \mathbf{A} \to \mathbf{A}$ of the ... $\vec{f}(\vec{x}) = \sum_{k=0}^{n} \vec{e}_k u_k(\vec{x})$, where
$u_k(\vec{x}) = u_k(x_0, x_1, \ldots, x_n)$ are ... of $n+1$ arguments .
Declarations (Physical Semantics)
... a function $\vec{f}: \mathbf{A} \to \mathbf{A}$ ... form $\vec{f}(\vec{x}) = \sum_{k=0}^{n} \vec{e}_k u_k(\vec{x})$ ...
where $u_k(\vec{x}) = u_k(x_0, x_1, \ldots, x_n)$ are real (or complex) functions ...
Main Equations (Problem Formulation)
$\vec{f}: \mathbf{A} \to \mathbf{A}$
$\vec{f}(\vec{x}) = \sum_{k=0}^{n} \vec{e}_k u_k(\vec{x})$
$u_k(\vec{x}) = u_k(x_0, x_1, \ldots, x_n)$
$n+1$ arguments

Figure 6: The information layer and their key features required to model and predict mathematical object related entities.

Organization of the Dissertation

The remainder of this dissertation is organized as follows. First, Chapter II presents prior knowledge on the problem setting in this research. Chapter III addresses the issue of identifying the MOs which contain the semantics of constraints. Chapter IV deals with coreference mining of MOs and context words to bridge the gap from quantitative abstraction to the physical world. Chapter V introduces heuristics to construct a graphical representation that expresses the technical essence of a scientific paper, which is a collective process of segmentation, interrelation, and reduction for the original contents. The final products are used for cross-paper analysis to cluster various documents based on different writing styles derived from the reasoning flows. Finally, a summary of contributions and expectations of near future works is given in Chapter VI.

CHAPTER II

PRELIMINARIES

This chapter provides some background knowledge for the research proposed in this dissertation study, which includes the introduction of the commonly used types of input documents, the existing works of technical element extraction, the current status of technical content analysis, and the existing datasets we used to evaluate our models.

Electronic Documents

TeX [9] and portable document format (PDF) [10] are the two most common electronic document formats that are distributed in academic publishing. TeX is a typesetting system originated by Donald Knuth in 1977 and its initial version is released in 1978. It has been extensively used in academia, especially for authors who are in science, technology, engineering and mathematics (STEM) disciplines. TeX has the advantages of cross-platform and can handle the typesetting complex mathematical formulae. It is also used to support other forms of typesetting tasks such as the LaTeX macro packages [11]. PDF, on the other hand, is a general printing format developed by Adobe in 1990s to include text, fonts, vector graphics, raster images and other information needed to display in a document. Most of the popular word processors such as Microsoft Word, LyX, and Google Docs can support PDF as the output document format. Other TeX editing systems like TeXmaker, TeXnicCenter, and TeXworks also support PDF as output files and preserve the original TeX sources. Our system mainly focuses on the TeX document, where mathematical objects (MOs) are labeled in the contents. However, since

PDF is the de facto standard for scientific publishing, we have also developed a pipeline to extract and parse MOs from the PDF files using API tools: PDFMiner [12] and PDFBox [13].

Extraction of Technical Elements

Any technical discussion can be constructed by two major technical elements: MOs and words. MO is a finite combination of symbols that is well-organized according to rules and the semantics can be formally defined in mathematics. Word, on the other hand, is a set of non-separable character sequence (NCS) that carry human understandable meanings. In LaTeX files, MOs exist in syntax such as '$' for users to define math zones in their editing and words are simply separated by spaces. However, PDF files have excluded the information of which part is MO and which part is word since their objective is to easily archive documents. Hence, there is a significant amount of research [14], [15], [16], [17], [18], [19], [20], [21], [22], [23], [24] focusing on the extraction of MOs from PDF files.

PDFMiner [12] and PDFBox [13] are two major tools for parsing the resources of PDF files such as font set. They applied physical layout lines to scan through the whole document, as shown in Figure 7. Each physical layout line contains a list of character objects separated by their built-in tokenizer based on spacing. PDFBox creates fonts and maps a character value to a glyph name such as "alpha" and "beta". A character object includes metadata such as glyph name, font name, Unicode, and the bounding boxes for font and glyph (see Figure 7). A bounding box consists of the left, right, top, and bottom positions of a rectangle with respect to the left-bottom corner of the page as the origin position.

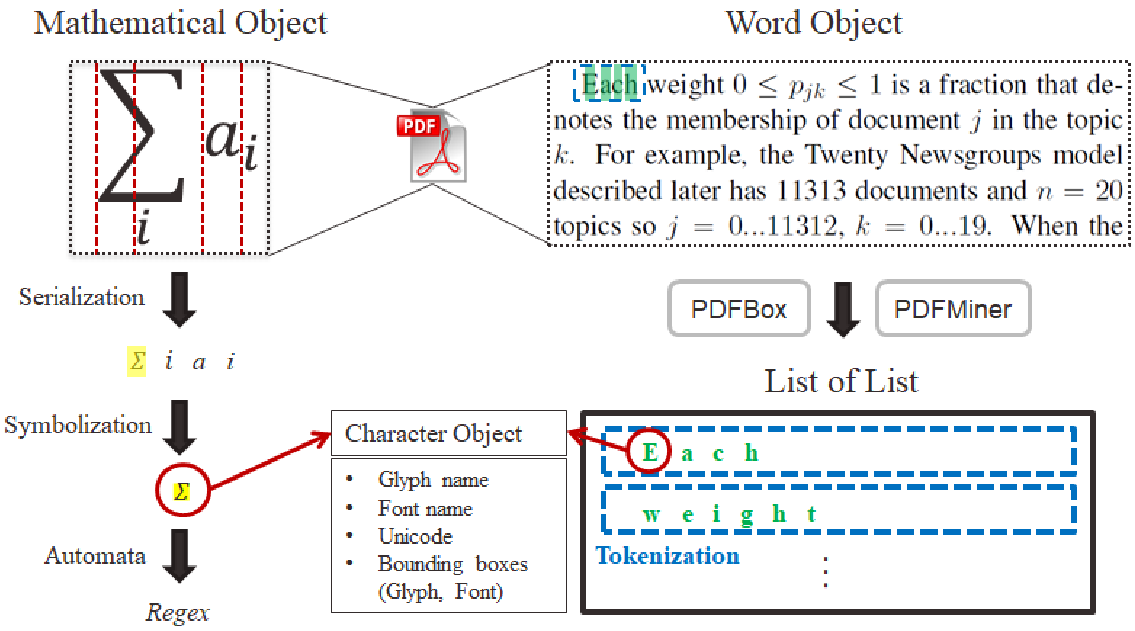

Mathematical Object
Word Object
$\sum_i a_i$
PDF
Each weight $0 \le p_{jk} \le 1$ is a fraction that denotes the membership of document j in the topic k. For example, the Twenty Newsgroups model described later has 11313 documents and $n = 20$ topics so $j = 0...11312$, $k = 0...19$. When the
Serialization
PDFBox
PDFMiner
Σ i a i
List of List
Symbolization
Character Object
Glyph name
Font name
Unicode
Bounding boxes (Glyph, Font)
Σ
E a c h
w e i g h t
Tokenization
⋮
Automata
Regex

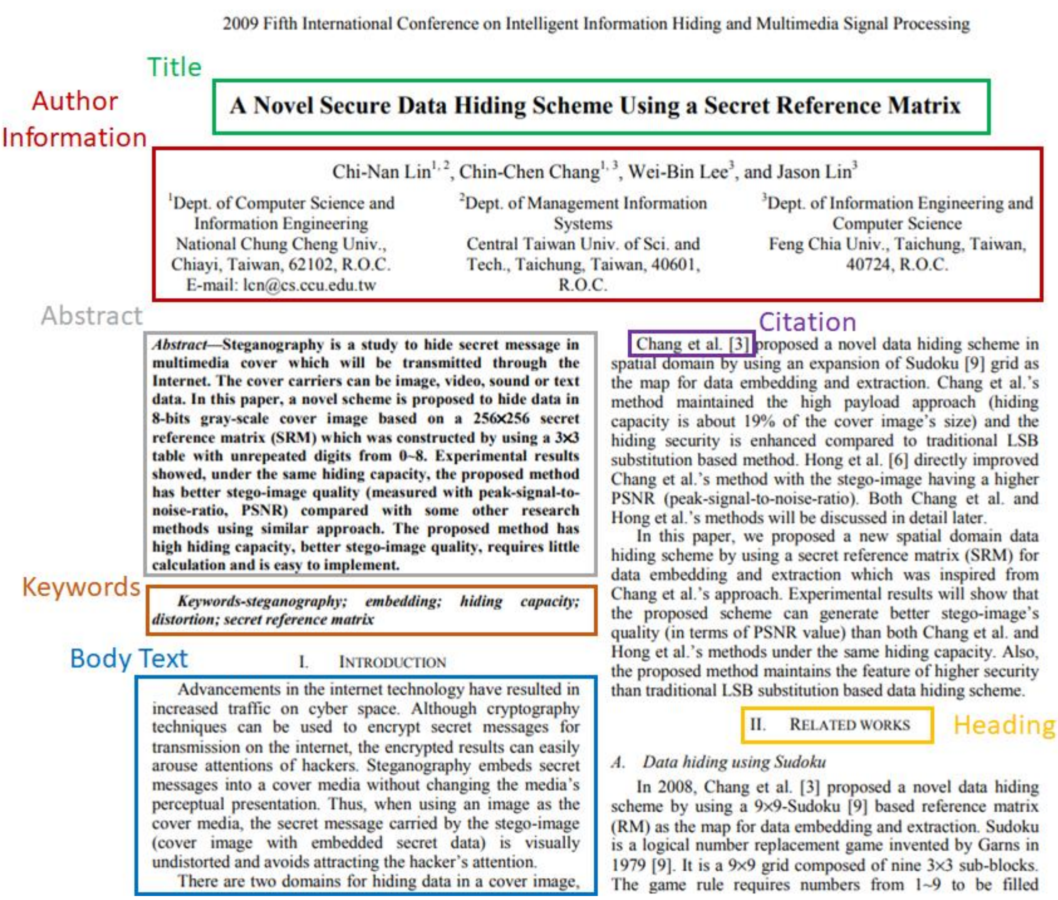
Figure 7: The data structures of mathematical objects and words obtained from the PDF parser.

The earliest work for extracting MO from scientific documents can be traced back to 1995 [14]. MO can be inline among plaintexts (IMO) or displayed as a stand-alone formula (DMO). The DMO is easier to detect as it often has formula serial number with distinct layout. The IMO, on the other hand, is more challenging to extract than DMO due to its unrestricted used of fonts and ambiguous boundaries with words caused by the discrepancy between physical layout analysis and the logical units. Besides spatial layout features, other semantic aspects, including fonts and special characters, are also explored in existing works [17], [18], [23], [24] to distinguish MOs from the text. For example, the italic font and the irregular font size are used to filter out MOs from other texts. Special characters such as function names, fraction/radical structure, relational operators, Greek letters, delimiters, integral symbol, etc. are also used as indicators.

There are two primary techniques for the research regarding MO extraction from PDF documents: the optical character recognition (OCR) [14], [15], [16], and the PDF parser [17], [18], [19], [20], [21], [22], [23], [24]. In the OCR-based research, a PDF document is first converted into render images, and then MOs are detected based on layout analysis. An early work [15] has applied the OCR technique to extract non-Japanese characters as MOs from Japanese documents. A follow-up work [16] has improved the performance based on the character size and position. In the PDF parser-based research, various features such as the attributes of the character object, the geometric layout, and the context were used to train the prediction models. For example, Lin et al. [20] utilized the visual and character features to establish a support vector machine (SVM) to identify IMOs from PDF documents. Several machine learning algorithms were combined with heuristic rules to detect both IMOs and DMOs [21]. In 2017, a weakly-supervised Bayesian model for MO extraction was proposed based on the font set information [23]. Without using any ground truth data for supervised learning, the algorithm first employed heuristic rules for DMO detection. Then, a Bayesian predictor was trained based on the font name and glyph name of the DMO characters to identify the IMO characters relatively. Recently, Wang et al. [24] proposed an unsupervised learning method based on the font size information to achieve a state-of-the-art performance of 93.6% F1 score in extracting MOs from PDF documents.

Technical Content Analysis

Many existing works [25], [26], [27], [28], [29] for content analysis focused on predicting the metadata in the text to recover its logical structure with respect to a document. They analyzed the contents based on layout information and textual features to automatically infer the types of metadata at different positions in the contents like author information, keywords title, abstract, headings, body texts, and citations as shown in Figure 8. Besides recovering the logical structure of a text, some other research have emphasized on extracting the mathematical logic [30], [31], [32] from text. Their approaches are mostly heuristics based on special mathematical terms ("theorem", "proof", "lemma"), layout information (space, position), and word fonts (style, size). Also, their model requires the input documents to be well-formatted in mathematical writing practices. In 2018, we developed the first work that studied on recovering the technical essence of linear displayed contents [33]. A graphical structure Qualitative-Quantitative (QuQn) map is created as the technical essence of any scientific document, as shown in Figure 9. The recovering process of the QuQn map is as follows. First, the digital files are parsed as rendering blocks or markup units according to the format specification. Then, the rendering blocks are transformed into layout structures (columns, lines) and grouped into logical structures (body texts and DMOs). For the MOs, their semantics of the internal components are labeled through layout analysis [34] and their external meanings of MOs are recovered through the bonding words. Finally, the reasoning logic flows are discovered through MO-based dependency analysis.



2009 Fifth International Conference on Intelligent Information Hiding and Multimedia Signal Processing

A Novel Secure Data Hiding Scheme Using a Secret Reference Matrix

Chi-Nan Lin[1, 2], Chin-Chen Chang[1, 3], Wei-Bin Lee[3], and Jason Lin[3]

[1]Dept. of Computer Science and
Information Engineering
National Chung Cheng Univ.,
Chiayi, Taiwan, 62102, R.O.C.
E-mail: lcn@cs.ccu.edu.tw

[2]Dept. of Management Information
Systems
Central Taiwan Univ. of Sci. and
Tech., Taichung, Taiwan, 40601,
R.O.C.

[3]Dept. of Information Engineering and
Computer Science
Feng Chia Univ., Taichung, Taiwan,
40724, R.O.C.

***Abstract*—Steganography is a study to hide secret message in multimedia cover which will be transmitted through the Internet. The cover carriers can be image, video, sound or text data. In this paper, a novel scheme is proposed to hide data in 8-bits gray-scale cover image based on a 256×256 secret reference matrix (SRM) which was constructed by using a 3×3 table with unrepeated digits from 0~8. Experimental results showed, under the same hiding capacity, the proposed method has better stego-image quality (measured with peak-signal-to-noise-ratio, PSNR) compared with some other research methods using similar approach. The proposed method has high hiding capacity, better stego-image quality, requires little calculation and is easy to implement.**

Keywords-steganography; embedding; hiding capacity; distortion; secret reference matrix

I. INTRODUCTION

Advancements in the internet technology have resulted in increased traffic on cyber space. Although cryptography techniques can be used to encrypt secret messages for transmission on the internet, the encrypted results can easily arouse attentions of hackers. Steganography embeds secret messages into a cover media without changing the media's perceptual presentation. Thus, when using an image as the cover media, the secret message carried by the stego-image (cover image with embedded secret data) is visually undistorted and avoids attracting the hacker's attention.

There are two domains for hiding data in a cover image,

Chang et al. [3] proposed a novel data hiding scheme in spatial domain by using an expansion of Sudoku [9] grid as the map for data embedding and extraction. Chang et al.'s method maintained the high payload approach (hiding capacity is about 19% of the cover image's size) and the hiding security is enhanced compared to traditional LSB substitution based method. Hong et al. [6] directly improved Chang et al.'s method with the stego-image having a higher PSNR (peak-signal-to-noise-ratio). Both Chang et al. and Hong et al.'s methods will be discussed in detail later.

In this paper, we proposed a new spatial domain data hiding scheme by using a secret reference matrix (SRM) for data embedding and extraction which was inspired from Chang et al.'s approach. Experimental results will show that the proposed scheme can generate better stego-image's quality (in terms of PSNR value) than both Chang et al. and Hong et al.'s methods under the same hiding capacity. Also, the proposed method maintains the feature of higher security than traditional LSB substitution based data hiding scheme.

II. RELATED WORKS

A. Data hiding using Sudoku

In 2008, Chang et al. [3] proposed a novel data hiding scheme by using a 9×9-Sudoku [9] based reference matrix (RM) as the map for data embedding and extraction. Sudoku is a logical number replacement game invented by Garns in 1979 [9]. It is a 9×9 grid composed of nine 3×3 sub-blocks. The game rule requires numbers from 1~9 to be filled

Figure 8: The labeling of metadata in a research paper [35].

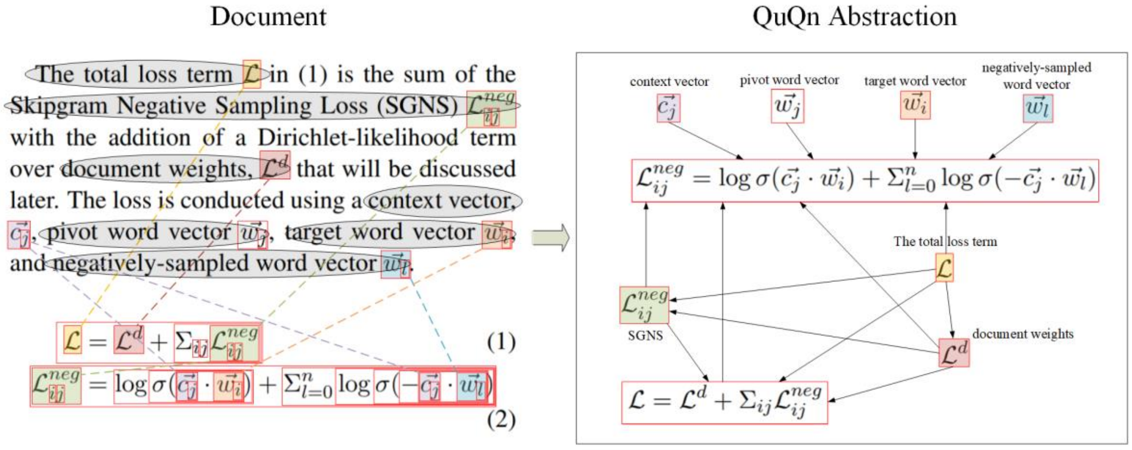

Document
QuQn Abstraction
The total loss term $\mathcal{L}$ in (1) is the sum of the Skipgram Negative Sampling Loss (SGNS) $\mathcal{L}_{ij}^{neg}$ with the addition of a Dirichlet-likelihood term over document weights, $\mathcal{L}^d$ that will be discussed later. The loss is conducted using a context vector, $\vec{c_j}$, pivot word vector $\vec{w_j}$, target word vector $\vec{w_i}$, and negatively-sampled word vector $\vec{w_l}$.
$\mathcal{L} = \mathcal{L}^d + \Sigma_{ij}\mathcal{L}_{ij}^{neg}$ (1)
$\mathcal{L}_{ij}^{neg} = \log\sigma(\vec{c_j}\cdot\vec{w_i}) + \Sigma_{l=0}^{n}\log\sigma(-\vec{c_j}\cdot\vec{w_l})$ (2)
context vector
$\vec{c_j}$
pivot word vector
$\vec{w_j}$
target word vector
$\vec{w_i}$
negatively-sampled word vector
$\vec{w_l}$
$\mathcal{L}_{ij}^{neg} = \log\sigma(\vec{c_j}\cdot\vec{w_i}) + \Sigma_{l=0}^{n}\log\sigma(-\vec{c_j}\cdot\vec{w_l})$
The total loss term
$\mathcal{L}$
$\mathcal{L}_{ij}^{neg}$
SGNS
document weights
$\mathcal{L}^d$
$\mathcal{L} = \mathcal{L}^d + \Sigma_{ij}\mathcal{L}_{ij}^{neg}$

Figure 9: The construction of QuQn map on a paragraph in arXiv document 1605.02019. Reprinted with permission from [33].

Overview of the Datasets

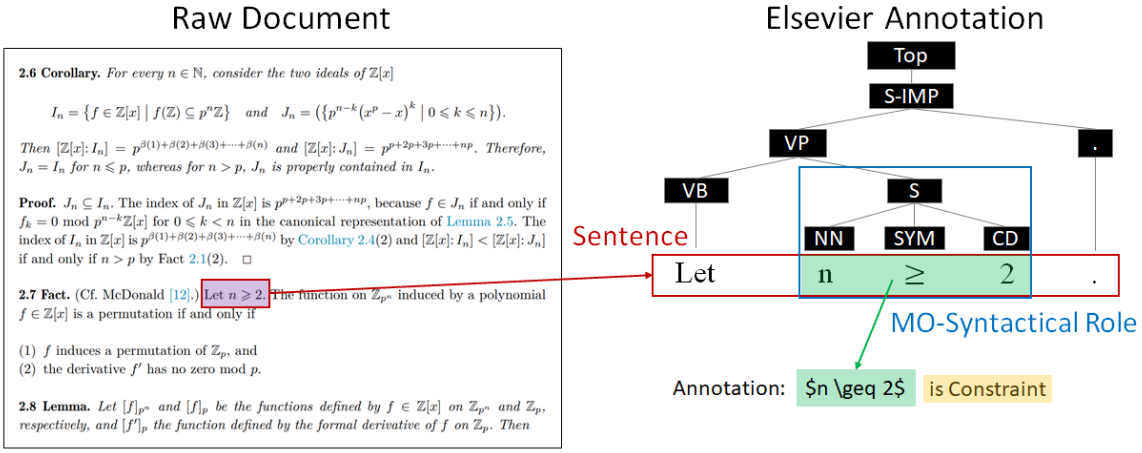
We introduce in this subsection the publicly available datasets used in this dissertation for evaluating the performance results of the research tasks RT1-RT4 in Figure 4 of Chapter I. There are four major datasets we used for our research: OA-STM Corpus [36], NTCIR-10 [37], KDD Cup 2003 [38], and arXiv.org [39], listed in Table 1, provided by the Elsevier Labs, National Institute of Informatics, Cornell University, and the arXiv e-print services, respectively.

Table 1: List of existing datasets for technical content analysis.

Name	**Year**	**Details**
OA-STM Corpus [36]	2015	→ 10 annotated documents in 10 different STEM disciplines. → Documents are provided in both txt and xml formats. → There are a total of 346 mathematical objects and words mixed sentences. → 600 mathematical objects are given in Penn Treebank formats along with their syntactic roles.
NTCIR-10 Math Understanding [37]	2012	→ 35 annotated documents with a total of 9172 mathematical objects. → The annotations include 3076 short declarations and 3053 full declarations of mathematical objects. → All annotations are provided in xml and txt formats.
KDD Cup 2003 [38]**,** [39]	2003	→ A collection of 29,000 papers with 352,807 citations in High Energy Particle Physics (HEP) from 1992 to 2003. → The TeX sources are downloaded from the arXiv.org.
RTDS Collections [40]**,** [41]**,** [42]	Active	→ A collection of 180 papers from arXiv.org in Physics, Mathematics, Computer Science, Statistics, and Economics. → A dataset MOP that combines the TeX and PDF files from KDD Cup 2003 to label the ground truth of the mathematical objects. → The LaTeX codes and constraint annotations of the mathematical objects in the OA-STM Corpus.

The OA-STM Corpus was released for the FORCE-2015 Hackathon event. It contained the annotations of MOs versus words and the syntactic roles of each MO as shown in Figure 10. There were 10 papers from 10 different fields annotated in this dataset.

Since the format of MOs and words were all in plaintext, some further annotations for the LaTeX code of MOs [40] and the constraint label of MOs [41] were made and released on behalf of RTDS Lab at Texas A&M University, College Station. This dataset was applied to both RT1 and RT2.



2.6 Corollary. *For every $n \in \mathbb{N}$, consider the two ideals of $\mathbb{Z}[x]$*

$$I_n = \{f \in \mathbb{Z}[x] \mid f(\mathbb{Z}) \subseteq p^n\mathbb{Z}\} \quad \textit{and} \quad J_n = (\{p^{n-k}(x^p - x)^k \mid 0 \leqslant k \leqslant n\}).$$

Then $[\mathbb{Z}[x]\colon I_n] = p^{\beta(1)+\beta(2)+\beta(3)+\cdots+\beta(n)}$ and $[\mathbb{Z}[x]\colon J_n] = p^{p+2p+3p+\cdots+np}$. Therefore, $J_n = I_n$ for $n \leqslant p$, whereas for $n > p$, J_n is properly contained in I_n.

Proof. $J_n \subseteq I_n$. The index of J_n in $\mathbb{Z}[x]$ is $p^{p+2p+3p+\cdots+np}$, because $f \in J_n$ if and only if $f_k = 0 \bmod p^{n-k}\mathbb{Z}[x]$ for $0 \leqslant k < n$ in the canonical representation of Lemma 2.5. The index of I_n in $\mathbb{Z}[x]$ is $p^{\beta(1)+\beta(2)+\beta(3)+\cdots+\beta(n)}$ by Corollary 2.4(2) and $[\mathbb{Z}[x]\colon I_n] < [\mathbb{Z}[x]\colon J_n]$ if and only if $n > p$ by Fact 2.1(2). □

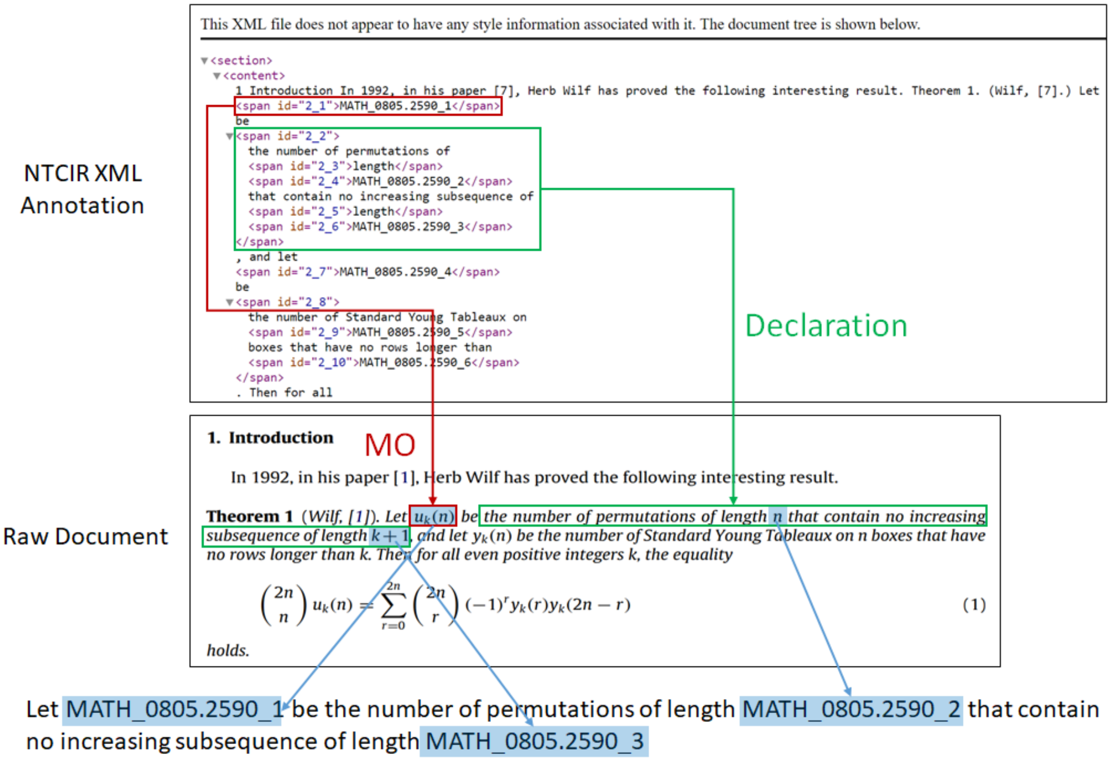
2.7 Fact. (Cf. McDonald [12].) Let $n \geqslant 2$. The function on $\mathbb{Z}_{p^n}$ induced by a polynomial $f \in \mathbb{Z}[x]$ is a permutation if and only if

(1) f induces a permutation of $\mathbb{Z}_p$, and
(2) the derivative f' has no zero mod p.

2.8 Lemma. *Let $[f]_{p^n}$ and $[f]_p$ be the functions defined by $f \in \mathbb{Z}[x]$ on $\mathbb{Z}_{p^n}$ and $\mathbb{Z}_p$, respectively, and $[f']_p$ the function defined by the formal derivative of f on $\mathbb{Z}_p$. Then*

Figure 10: An example of the dataset OA-STM Corpus for syntactic role and constraint annotation of mathematical objects (MOs) in STEM fields.

The NTCIR-10 was a collaborative annotation task from various institutions in the world [37]. There were 35 documents from arXiv.org annotated in XML format as shown in Figure 11. The main goal for this annotation task was to understand the meaning of mathematical formulae in scientific publications. The annotations include the mapping of MOs to their declarations (see Figure 11). There were two types of declarations: short and full declarations, representing the level of semantic details for the annotated MOs. A declaration can be either in a sequence of words or a mix of MOs and words. This dataset was applied to RT2 for extracting textual declarations of MOs.



This XML file does not appear to have any style information associated with it. The document tree is shown below.

```
▼<section>
  ▼<content>
     1 Introduction In 1992, in his paper [7], Herb Wilf has proved the following interesting result. Theorem 1. (Wilf, [7].) Let
     <span id="2_1">MATH_0805.2590_1</span>
     be
   ▼<span id="2_2">
       the number of permutations of
       <span id="2_3">length</span>
       <span id="2_4">MATH_0805.2590_2</span>
       that contain no increasing subsequence of
       <span id="2_5">length</span>
       <span id="2_6">MATH_0805.2590_3</span>
     </span>
     , and let
     <span id="2_7">MATH_0805.2590_4</span>
     be
   ▼<span id="2_8">
       the number of Standard Young Tableaux on
       <span id="2_9">MATH_0805.2590_5</span>
       boxes that have no rows longer than
       <span id="2_10">MATH_0805.2590_6</span>
     </span>
     . Then for all
```

1. Introduction

In 1992, in his paper [1], Herb Wilf has proved the following interesting result.

Theorem 1 (*Wilf, [1]*). *Let $u_k(n)$ be the number of permutations of length n that contain no increasing subsequence of length $k+1$, and let $y_k(n)$ be the number of Standard Young Tableaux on n boxes that have no rows longer than k. Then for all even positive integers k, the equality*

$$\binom{2n}{n} u_k(n) = \sum_{r=0}^{2n} \binom{2n}{r} (-1)^r y_k(r) y_k(2n-r) \quad (1)$$

holds.

Let MATH_0805.2590_1 be the number of permutations of length MATH_0805.2590_2 that contain no increasing subsequence of length MATH_0805.2590_3

Figure 11: An example of the NTCIR-10 dataset for the textual declaration of mathematical object.

The KDD Cup 2003 was a competition event held by the Ninth Annual ACM SIGKDD Conference. It included 29000 documents of High Energy Particle Physics published in arXiv.org during 1992 to 2003. The dataset was crawled and organized into single TeX sources. The RTDS Lab also developed an MOP tool [42] to convert and combine these TeX sources along with their corresponding PDF documents to obtain a large-scale dataset with MO annotated in LaTeX code. A small portion of these documents are used for cross-paper analysis in RT4.

In this dissertation, we also sample some scientific documents in various disciplines from the arXiv.org e-print platform [39] to construct a pilot dataset for the study of use cases in RT3 and RT4. There are 180 documents sampled from 6 research fields including Computation and Language, Graph Theory, Machine Learning, Quantum Cryptography, Steganography, and Theoretical Economics. An additional 30 documents are sampled from the KDD Cup 2003 dataset in High Energy Particle Physics to constitute a medium-size dataset with 210 documents in 7 fields.

CHAPTER III

CONSTRAINT EXPRESSION OF MATHEMATICAL OBJECT[1]

This chapter presents two models to extract the constraint expressions of mathematical objects (MOs) in scientific publications. With the assumption of the features are independent and identically distributed (IID), we apply two types of features: the mathematical symbols (F_S) and the words adjacent to MOs (F_W), for analysis. The first prediction model is based on a greedy approach to iteratively optimize the performance goal. The second scheme is based on naïve Bayesian inference of the two different types of feature considering the likelihood of the training data. The first model achieved an average F1 scores of 69.5% (based on the tests made on an Elsevier dataset OA-STM Corpus). The second prediction model using F_S achieved 82.4% for F1 score and 81.8% accuracy. Furthermore, the second model achieved similar yet slightly higher F1 scores as that of the first model for the word stems of F_W, but slightly lower F1 score for the Part-of-Speech (POS) tags of F_W.

Overview

In technical writings, MOs and carefully placed adjacent words are used to characterize the technical substance within specific disciplines. The constraint expression of MO (MOC) refers to MOs (and their adjacent words) that are meant to describe the

[1] Reprinted with permission from "Prediction of Mathematical Expression Constraints (ME-Con)" by Jason Lin, Xing Wang, and Jyh-Charn Liu, 2018. *Proceedings of the 18th ACM Symposium on Document Engineering*, Pages 1-4, Copyright 2018 by ACM.

constraints or conditions of any mathematical or technical subject. Being a form of semantic abstraction, human readers can readily differentiate them from other semantic abstractions such as definitions and reasoning flow transitions, but the boundaries between these different types can be fluid.

In this chapter, we propose and compare two different optimization models for prediction of MOC. Constraints are an integrated part of every MO, and they are often the attachments of the main discussion threads. Automatic detection of MOC is useful in tracking the evolution of a reasoning process, delineation of similar works, among other modeling tasks. Using the annotated dataset in [36], [40], [41] as the ground truth, we develop the models by two classes of features: F_S, based on mathematical symbols, and F_W, based on the word stems or the POS tags, at the left-hand side, as well as at the right-hand side of MO. Although natural language processing (NLP) technologies makes significant progress in low level content processing [43], [44], they do not take into account the mathematical semantics of MOs and their relationship with adjacent words.

The first prediction model is a greedy approach based on an iterative heuristic rule to optimize the prediction goal, and the second adopts the Bayesian theorem. Empirical results suggest that certain mathematical symbols from F_S directly dictates the semantics of an MO as an MOC or not. To less extent, certain words or syntactical forms from F_W have similar designating power for the adjacent MO (not) to be a constraint. Moreover, evidences show that the use of word stem is more indicative than its syntactic role (i.e., the POS tag) to convey the intention of an MO. An observation consistent with the fact that MOC is a semantic interpretation of an MO, not a particular class of presentation structures expressed in POS.

Related Works

The word “constraint” has been defined in different terms such as “properties”, “attributes” of a subject, and relational properties between subjects. It is related to the information extraction field, where constraint analysis is aimed at the retrieval of properties and relations in or between subjects from computer-generated contents or well-defined language such as SQL [45], [46], [47]. Resources, structure, hierarchy, and dependency are considered the main constraints in system development documents [48]. For these applications, constraint extraction techniques are largely based on a mix of grammar, keyword matching, and NLP tools. For example, the work in [49] presented a patent for constraint extraction based on template matching for generation of testing data. It utilized handcrafted rules on POS tags to capture three sentence elements: subject, object, and condition. To date, no known work done specifically for MOC extraction.

Features of MOC Expressions

A human reader asserts an MOC based on the combination of symbols (e.g., ‘|’ within ‘{’ and ‘}’), attached words or their syntactic role, e.g., “where” WRB, “for” IN), in simple or compounded paragraphs (e.g., “where … {…|…}”). Those cue words or symbols that give human strong hint about the presence of MOC are called constrainators. We use statistic to rank the likelihood of different mathematical symbols, word stems and POS tags of words adjacent to MO.

Feature Extraction

The training of the MOC prediction rules starts with extraction of F_S and F_W. Here, we employed a regular expression (regex) parser to parse the LaTeX annotated MOs [40] to extract mathematical symbols represented by reserved words, such as “\sub”. The regex “(\\[A-Za-z]+)” is used to extract the LaTeX reserved words for mathematical symbols so that “\ldots” in the example shown in Figure 12 can be identified to represent the symbol of a long dot string ‘…’. Other symbols that can be directly entered using common keyboards are extracted from the regex “([-!%&*()+|~=\[\]\' :;<>?,.\/]|\\[{}])”. An example on the set of symbols parsed by the two extractors is illustrated in Figure 12.

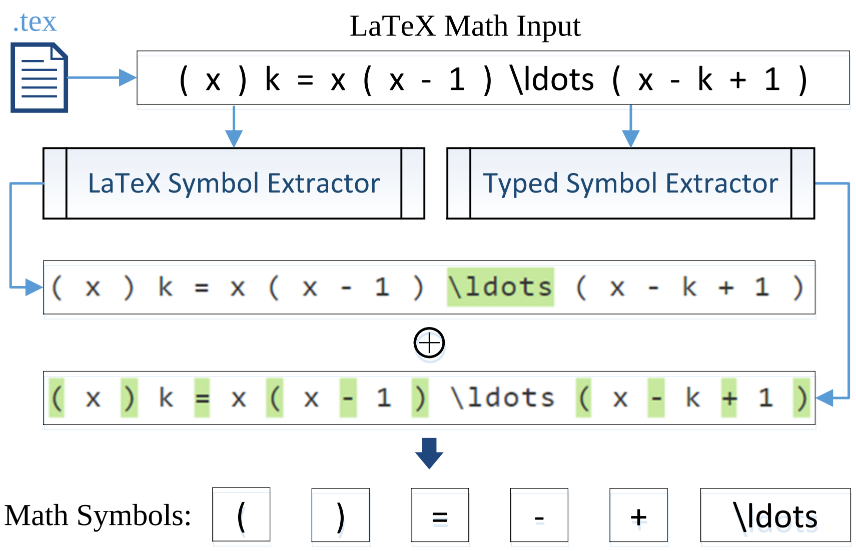

.tex
LaTeX Math Input
(x) k = x (x - 1) \ldots (x - k + 1)
LaTeX Symbol Extractor
Typed Symbol Extractor
(x) k = x (x - 1) \ldots (x - k + 1)
⊕
(x) k = x (x - 1) \ldots (x - k + 1)
Math Symbols:
(
)
=
-
+
\ldots

Figure 12: An example for the mathematical symbol extraction. Reprinted with permission from [50].

It is obvious that some words adjacent to an MO are highly likely meant to describe its semantics. That being said, to cope with the very large number of random word forms being used in MOC, we opted to the hypothesis that the MOC assertion could be related to the word stems or the syntactic roles (POS tag) of words adjacent to the MO. To test this theory, we used up to two adjacent words of an MO to construct F_W. Only one F_W feature entry has value when the MO is adjacent to another MO or is located at the beginning/ending of a sentence. In the latter case the type of adjacent punctuation is treated as a stem (or a POS tag). For the word stem, we apply the stemming process via the API in NLTK [43], which is based on the algorithm in [51]. The POS tags entries in the annotated dataset OA-STM Corpus [36] are used for training of F_W.

Analysis of Mathematical Symbol Features

MOC can be classified into five major types: (1) condition type ("$x \geq 1$", "Let $x = 1, \ldots$"), (2) index (range) type (e_i for $i \in [LHS, RHS]$), (3) set type ($S' \subseteq S$), (4) enumerative type (the number of permutations of n elements is n!), and (5) complexity type $O(n)$.

Based on the IID assumption of individual symbols and words, we first measured the relative likelihood of each symbol being used in MOC instances in the training dataset. The statistic on the MOC likelihood of different symbols is shown in Figure 13 and Figure 14. Within the space limit, only symbols with higher than 2% of occurrence ratio over the overall symbol counts were plotted. Symbols like "≥, $>$, $<$, ≤, ∈, Ω" are almost always meant for MOC, while symbols like "∘, ∗, ←, ρ, lim" almost never meant for MOC. The ubiquitous symbol "$=$" can be used for comparison or assignment. Different paired braces "$\{\ldots\}$", "[…]" when embedded with the bar '|' or colon ':' are most likely being used for MOC.

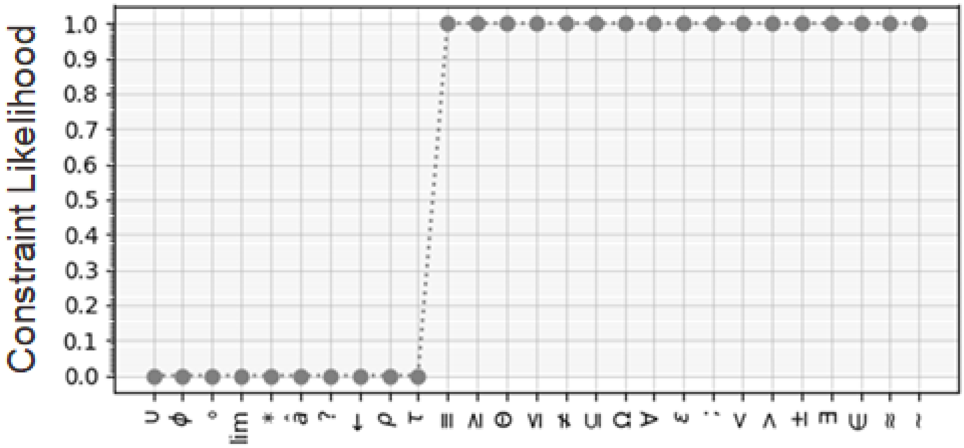


Figure 13: The statistical likelihood of strong indicative symbols used for recognizing the constraint expressions.

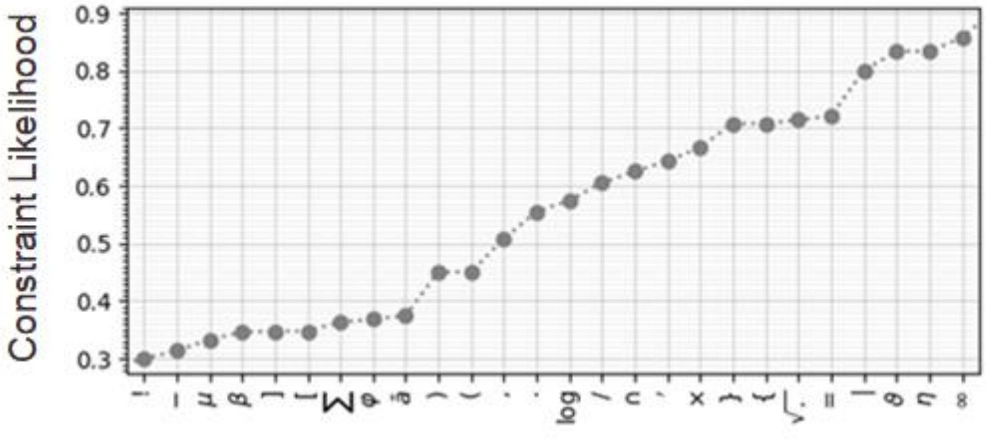


Figure 14: The statistical likelihood of weak indicative symbols used for recognizing the constraint expressions.

Analysis of Contextual Word Features

For the analysis of F_W, we took the words at the left and right sides of each MO and computed their relative likelihood of being associated with an MOC. Upon analyzing the words attach to MOs in the dataset OA-STM Corpus [36], a total of 28 POS tags and 155 (131) words with (without) stemming were used. To reduce the sampling space, all words are transformed into their word stems to be compared against POS tags as features for analysis. The results are summarized in Figure 15 for the pairs of word stems with at least a word stem is not strongly indicative for MOC when it is positioned at one side of the MO. The results for all possible POS tags are also summarized in Figure 16.

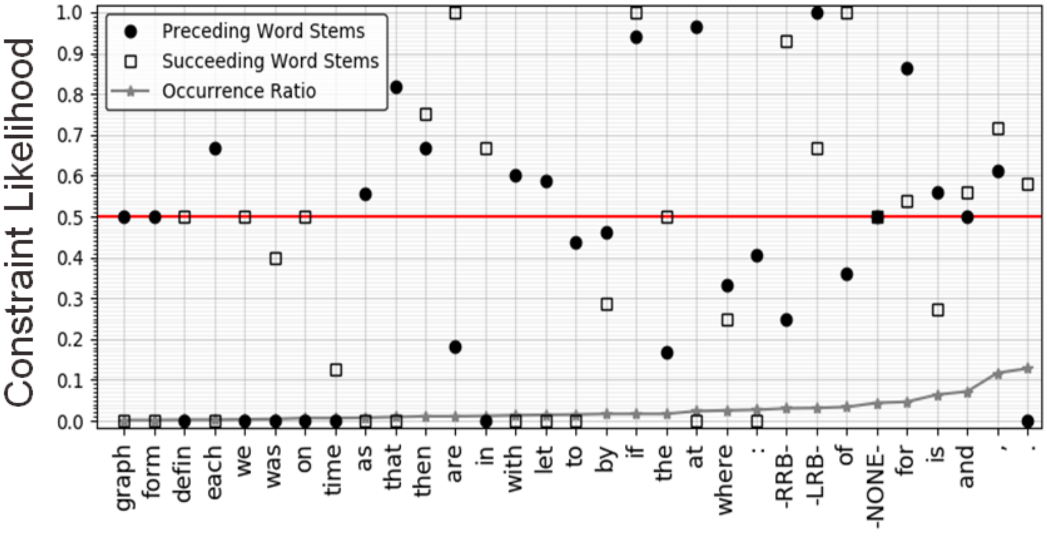


Figure 15: The statistical likelihood of word stems for introducing the constraint expressions in a sentence. Reprinted with permission from [50].

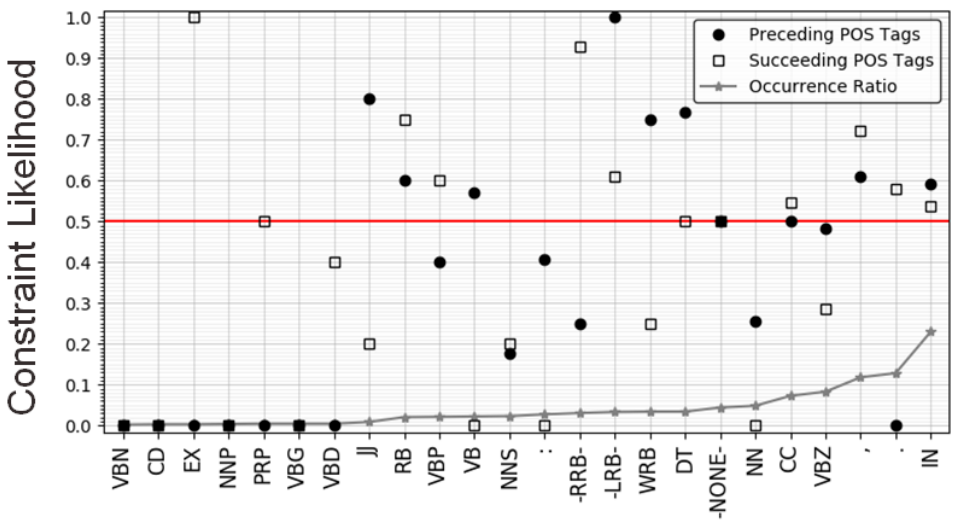


Figure 16: The statistical likelihood of POS tags for introducing the constraint expressions in a sentence. Reprinted with permission from [50].

Among word stems that appeared more than 1.5% in the dataset, certain word stems preceding an MO, like {“because”, “some”, “everi”, “get”, “now”, “relat”, “therefor”, “have”, “polynomi”, “between”, “when”, “all”}, are always meant for an MOC. Other word stems like {“while”, “sinc”, “there”, “mod”} after an MO are always meant for an MOC. Alone, the remaining word stems such as {“span”, “order”, “equivale”, “simpli”, “use”, “write”, “contain”, ‘[’, “onto”, “project”, “choic”, “group”, “pair”, “sylow”, “take”, “epimorph”, “transit”, “function”} were not strongly tied to MOC.

Regarding the preceding word, an MO following a left bracket -LRB- is nearly certain to be an MOC, because it is a common writing practice to place a condition/constraint in the bracket pair as a supplement to the main description. MOs following Wh-adverb WRB such as “where” and “when” has a relatively high likelihood to be an MOCs for the common phrasings of “where/when MOC …”. An MO following a noun (with POS tag NN and NNS for words such as “time” and “graph”) is unlikely an MOC because the MO is usually the apposition of the noun phrases.

With respect to the succeeding word, we find that MO followed by punctuations such as ‘.’ and ‘,’, or right bracket RRB are more likely to be MOC, because many constraints are positioned to follow a main statement. A noun NN word or a verb VB word does not usually follow an MOC, where the MO plays the role of noun modifier or subject.

Compared with word stem, no POS tag clearly stands out to be considered as constrainators. Taking the relatively high co-occurrence syntactic patterns (preposition-MO-punctuation) like “… [for] [$x > 0$] [.] …” as an example, the preceding POS tag IN alone has 19.2% of true positive cases, and yet also 15.2% of false positive cases for non-MOC cases. Overall, word stems performed better than POS tag, because MOC is a semantic level expression to indicate the intended purposes of an MO.

Comparison of Mathematical and Textual Features

Figure 17 summarizes the ratios of instances for different ranges of MOC likelihood. Results show that F_S features are more uniformly distributed than their F_W counterparts. Unlike MO symbols, F_W are much more pronounced in the midrange of likelihoods, suggesting that they tend to be more random, especially the POS distributions, than the F_S in expressing the semantic intent. As a result, increasing the sensitivity for F_W based MOC detection also increases the false positive rate resulting lower precisions. A general conclusion is that F_W alone has relatively low discriminating power for prediction of MOC.

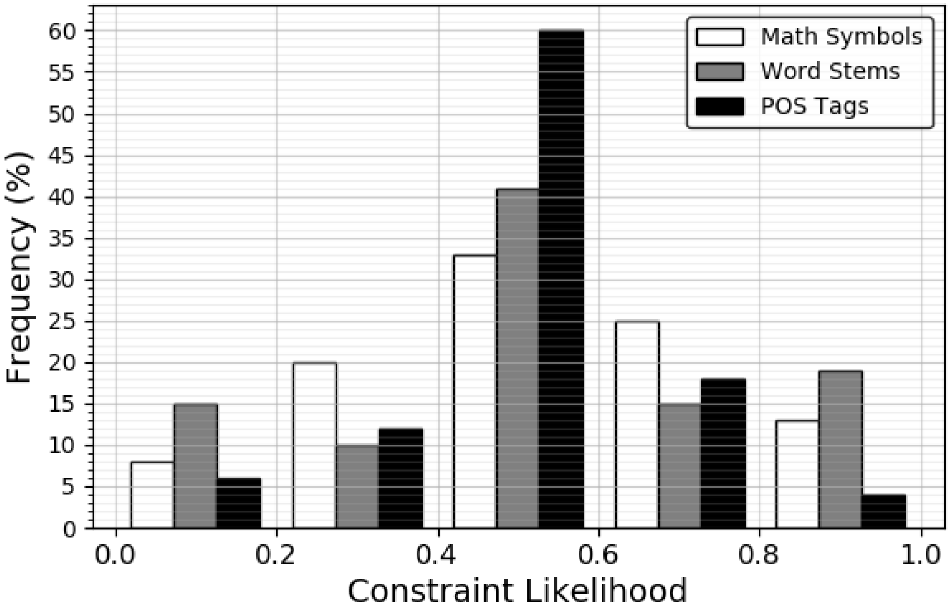


Figure 17: The sample density of MOC expression likelihood. Reprinted with permission from [50].

Prediction of Constraint Expression

An iterative heuristic rule and a naïve Bayesian decision rule are designed for MOC prediction using F_S and F_W as the inputs.

The greedy model iteratively refines the prediction rule for F_S and F_W based on the true positive (TP), false positive (FP), true negative (TN), and false negative (FN). Given a set of MO-words mixed phrases $P = \{p_1, \dots, p_{|P|}\}$, where p consists of an MO x and its attached words $\{LHS(x), RHS(x)\}$ at the left-hand side (LHS) and the right-hand side (RHS). We use a binary array $\vec{s}_x$ to represent the $|N|$ distinct symbols $\langle b_1, \dots, b_{|N|}\rangle$ in x. Similarly, the binary array $\vec{w}_x$ represents $|M|$ adjacent word stems or POS tags

$\langle b_1, \dots, b_{|M|} \rangle$. A binary variable $\mathcal{L}(x)$ represents the prediction outcome of x so that $\mathcal{L}(x) = 1$ when x is MOC.

In the training phase, each entry in $\vec{s}_x$ ($\vec{w}_x$) is assigned one of the four cases $k \in \{TP, FP, TN, FN\}$ in confusion matrix by comparing the ground truth and the prediction result. Let the binary vector $\vec{u}_k$ ($\vec{v}_k$) represents F_S (F_W) for the case k. The bit position in $\vec{u}_k$ ($\vec{v}_k$) is 1 if the corresponding position in $\vec{s}_x$ ($\vec{w}_x$) is 1. That is, $\vec{u}_k = \bigvee_{j=k} \vec{s}_x^{\,j}$ and $\vec{v}_k = \bigvee_{j=k} \vec{w}_x^{\,j}$ where V denotes the logic OR operation. A binary array y of F_S or F_W is trained from the process based on an iterative heuristic rule: at the *i*-th iteration, $y^i = \left(y_{TP}^{i-1} \vee y_{FN}^{i-1}\right) - \left(y_{TN}^{i-1} \wedge y_{FP}^{i-1}\right)$. That is, features appeared in both TN and FP instances are eliminated from the set of TP or FN instances. The iteration stops when there is no more training data to fit the model. The positive predictions of MOC are made upon y ∧ z ≠ $\vec{0}$ where z is the feature array of F_S or F_W.

Next, we discuss the naïve Bayesian model. Empirical results in Figure 13 and Figure 14 suggests that relative likelihoods of certain mathematical symbols being used in MOC expressions are higher than others. To test their discriminating power, we propose a naïve Bayesian inference model based on the likelihood of individual features without taking into account of the potential dependency between F_S and F_W.

Let $\theta = 1$ (0) denote the case that x is (not) in MOC. Here, for an MO x with word $LHS(x)$ and $RHS(x)$, the posterior probability is $Pr\left(\theta | e(x)\right) = \frac{Pr(e(x)|\theta)Pr(\theta)}{Pr\left(e(x)\right)}$, where $\theta \in \{0,1\}$ and $e(x)$ is an evidence set of F_S or F_W, where F_W is derived by the POS tag of $LHS(x)$ and $RHS(x)$. Based on the assumption of conditional independence, we have

$Pr(e(x)|\theta) = \prod_i Pr(e(x)_i|\theta)$ for the i-th symbol being evaluated. We propose a ratio function $\frac{Pr(e(x)|\theta=1)}{Pr(e(x)|\theta=0)}$ to assess the likelihood of x being in MOC or not. Based on this formula, one can ignore the common term $Pr\big(e(x)\big)$ at the denominator for cases $\theta = 1$ versus $\theta = 0$, so that $\frac{Pr\left(\theta=1|e(x)\right)}{Pr\left(\theta=0|e(x)\right)} = \frac{Pr(e(x)|\theta)Pr(\theta=1)}{Pr(e(x)|\theta)Pr(\theta=0)}$.

Experimental Results and Discussion

We compare the performance of the two proposed models based on precision (P), recall (R), F-measure (F1), accuracy (ACC), based on experiments performed on the dataset OA-STM Corpus [36], which provides 10 papers from different fields with 2757 sentences annotated in Penn tree format [52]. Each MO is labeled with FRM in the tag. 346 of the 2757 sentences contain about 600 MOs in the dataset. All MOs are manually annotated with “yes”, “no”, or “uncertain” that whether they are expressing a constraint semantic. Both the mathematical symbols and the local word features were applied to all 600 MOs out of 346 sentences.

We adopted the 10-fold cross validation (9:1 ratio for the sizes of the training and testing data sets) to evaluate the performance of the two models, and the results are given in Table 2. The high recall rate 92.7% in average using F_W indicates that the greedy model can capture nearly all combinations of word stems or POS tag patterns. However, the two prediction models achieve different levels of both precision and F1 score. As shown in Table 2, the heuristic rule-based greedy approach obtained an average precision of 58.1% and 69.5% of average F1 score for the three features F_S, F_W (stem), and F_W (POS). In contrast to the heuristic prediction model, the naïve Bayesian based prediction model obtained a precision of 80.1% and the F1 score of 82.4% for F_S, and an average precision and F1 score of 63.5% and 69.6% for F_W, which is higher than the overall average of the first model.

Table 2: The average performance of recognizing the constraint expressions from mathematical objects. Reprinted with permission from [50].

	P	R	F1	ACC
Greedy Model (F_S)	61.4%	85.3%	70.3%	63.5%
Naïve Bayesian (F_S)	80.1%	87.4%	82.4%	81.8%
Greedy Model (F_W, stem)	58.4%	90.2%	70.1%	61.4%
Naïve Bayesian (F_W, stem)	65.8%	82.8%	72.3%	68.1%
Greedy Model (F_W, POS)	54.5%	95.2%	68.0%	55.2%
Naïve Bayesian (F_W, POS)	61.1%	78.2%	66.9%	60.1%

Summary

In this chapter, we propose two supervised MOC prediction models based on heuristic optimization and naïve Bayesian decision inference, respectively. Multiple factors may contribute to the performance of the prediction models and the data features. The outcomes suggested that the F_S alone based prediction obtained overall good prediction scores. On the other hand, F_W was found to have much lower prediction powers, fluctuating sharply with respect to different word stems and POS tags. By intentionally keeping F_S and F_W separate, our experiments examine the semantic power of individual words and symbols. Mathematical symbols alone are found to carry significant semantic expression power, so that in many cases a human reader can readily assert an MO to be MOC (or not) with the presence of some symbols. Of course, words still carry some weights in such assertions, but its exact nature requires employment of more effective word-based features. With this work being the first of its kind in MOC inference, many open questions remain to be answered to understand the natures of F_W, in terms of issues such as phrase bonding with MO, other NLP generated primitives that may work better for MOC prediction.

CHAPTER IV

TEXTUAL DECLARATION OF MATHEMATICAL OBJECT[2]

Mathematical objects (MOs) and words are carefully bonded together in most science, technology, engineering and mathematics (STEM) documents. They respectively give quantitative and qualitative descriptions of a system model under discussion. In this chapter, we will introduce a general model for finding the coreference relations between words and MOs, based on which we developed a novel algorithm for predicting the natural language declarations of MOs--the MOD. The prediction algorithm is applied in a three-level framework, where the first level is a customized tagger to identify the syntactic roles of MOs and the Part-of-Speech (POS) tags of words in the MO-word mixed sentences. The second level screens the MOD candidates based on the hypothesis that most MOD are noun phrases (NP). A shallow chunker is trained from the fuzzy process mining algorithm, which uses the labeled POS tag series in the NTCIR-10 dataset as input to mine for the frequent syntactic patterns of NP. In the third level, using distance, word stem, and POS tag respectively as the spatial, semantic, and syntactic features, the bonding model between MOs and MOD candidates is trained on the NTCIR-10 training set. The final prediction results are made upon the majority votes of an ensemble of naïve Bayesian classifiers based on the three features. Evaluation of the model on the NTCIR-10 test set,

[2] Reprinted with permission from "Prediction of Mathematical Expression Declarations Based on Spatial, Semantic, and Syntactic Analysis" by Jason Lin, Xing Wang, Zelun Wang, Donald Beyette, and Jyh-Charn Liu, 2019. *Proceedings of the 19th ACM Symposium on Document Engineering*, Pages 1-10, Copyright 2019 by ACM.

the proposed algorithm achieved 75% and 71% average F1 score in soft matching and strict matching, respectively, which outperforms the state-of-the-art solutions by a margin of 5-18%.

Overview

In scientific documents, MOs are abstract notations of a complex system. It requires readers to map back to physical concepts through human-readable texts. A common practice for technical writing is that a mathematical notation must be introduced or declared before it is used for further discussions. A study showed that 58% of simple MOs come with declarations in the first occurrence in articles [53]. Since the way people declared MO followed limited patterns, it leads to a potential automation for extracting the textual declaration of MO (MOD). The automatically extracted MOD can act as a notation table to help the reader navigate between MOs and their physical meanings, and in addition, it have potential usage for cross-paper analysis [54], [55]. It has been shown that words and phrases used in MOD could help enhance the performance of mathematical information retrieval (MIR) [56] (i.e., combining the query of MOs and words to rank the relevance of searched documents).

The mining of MOD belongs to the domain of information extraction. We propose to solve this problem from three aspects: the spatial, semantic, and syntactic relations of MOs and words. In general, we as human writer use these by instinct to place words and MOs in a way such that the contextual information is derived in multiple levels which supports the bonding of the two. The challenges of this problem lies in three parts: (1) comprehensive solutions on sentences with mixed use of words and MOs (MWM) for lower level document processing such as POS tagging; (2) higher level constituent parsing for noun phrase (NP) to locate the possible chunks (i.e., group of words) of MOD; (3) mapping the MOs to their corresponding NP-chunks in the sentence. For the first part of challenges, we observed a degradation of the word-level POS tagging when applying the conventional natural language processing (NLP) toolkit to the MWM sentences. The syntactic role of MO (MO-SR) does not exist in any conventional POS annotation schema. Here, the syntactic role is defined as an umbrella term for the grammatical function of linguistic unit such as word, phrase, and expression. The errors occurred in word-level POS tagging could propagate to later constituent parsing on phrase-level annotation. The second part of the challenges requires a strategy to find the most frequent pattern of NP that covers most variances of MOD. Finally, the selection of which NP and MO have the same referent (i.e., the coreference) in a sentence remains a challenging problem. Naïve approach such as merely using the spatially nearest NP to MO as the MOD has resulted in a high false positive rate [57].

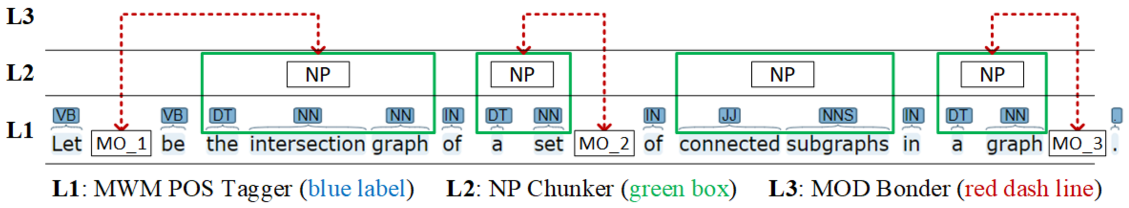

L3
L2
L1
NP
NP
NP
NP
VB
VB
DT
NN
NN
IN
DT
NN
IN
JJ
NNS
IN
DT
NN
Let MO_1 be the intersection graph of a set MO_2 of connected subgraphs in a graph MO_3 .
L1: MWM POS Tagger (blue label)
L2: NP Chunker (green box)
L3: MOD Bonder (red dash line)

Figure 18: The three-level framework for MOD prediction. Reprinted with permission from [58].

We propose a three-level framework as in Figure 18 to address the above three challenges. First, a customized POS tagger for the MWM sentences is proposed using the second order hidden Markov model [59]. Then, the NP-chunks are extracted as the potential MOD candidates by a shallow parser learned from the result of fuzzy process mining [60] using the human annotated dataset NTCIR-10 [37]. Finally, a predictive decision procedure determines whether an NP coreference to a designated MO in the sentence. The system architecture is depicted in Figure 19. Besides the pipeline of three-level predictive analysis (L1 to L3), we also apply a weakly supervised learning approach to semi-automatic identify rules of template patterns using anchor words for our model to filter out obvious cases of MOD. The system will consolidate the results with the ones from the statistical inference step (L3) to make the final selections.

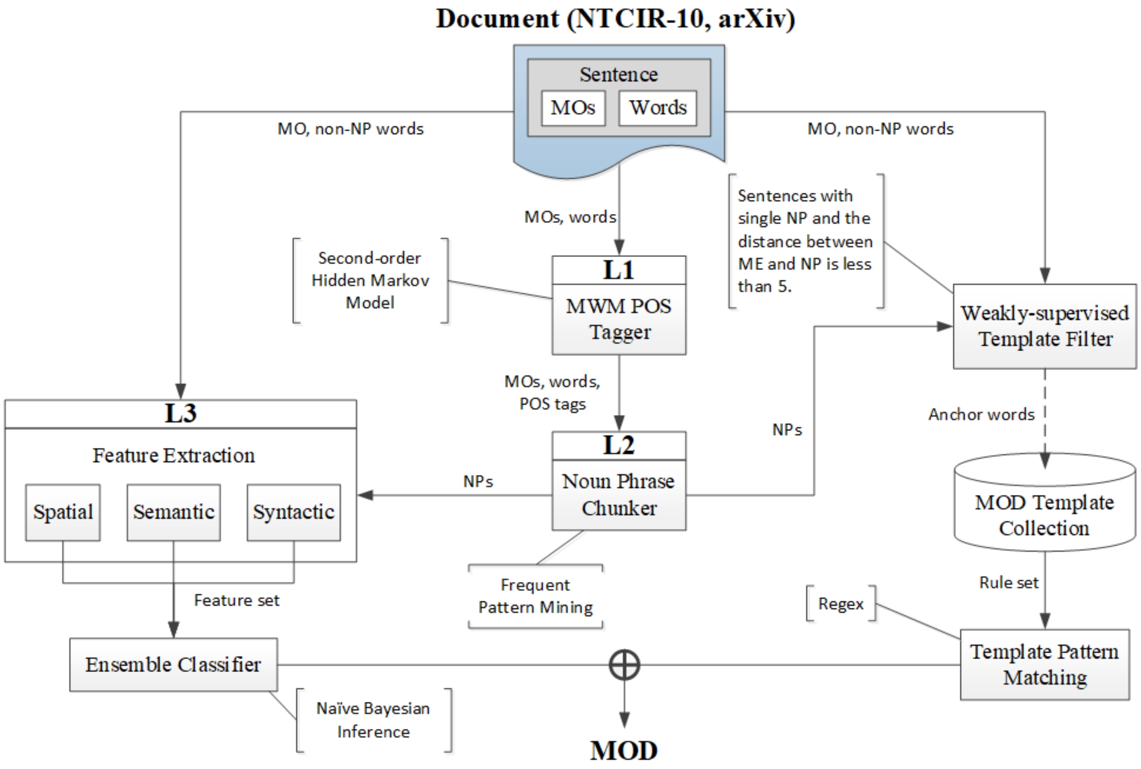


Figure 19: The system architecture for the bonding prediction of mathematical objects (MOs) and their related noun phrases (NPs). Reprinted with permission from [58].

Related Works

For automated declaration extraction, the first paper [61] in 2010 attempts to find coreference relation between formulas and their surrounding text on Wikipedia documents. They proposed a triplet tuple $\langle C, F, D\rangle$ of potential candidates representing the chunks of Concept, Formula, and Description in the contents, respectively. Their heuristics are based upon that description always follow a concept after the verb “be”, and such description belongs to the nearest formula that have overlaps with its concept. They achieved a performance of 68.33% on precision but only limited to a more organized contents like Wikipedia. Other existing works follow a two-phase framework [57], [62], [63]. First, the NPs are extracted as the candidates of the MOD based on traditional constituent parsing. Then, a prediction is made upon each pair of MO and NP about whether the NP is the MOD using a binary classifier. The classifier is trained using the features concerning the common declaration patterns, the values/POS of neighboring words, and structural features. However, none of them have considered the problem of MWM sentences that affect the prediction performance of the POS tagger, and propagate the errors to an upper layer analysis. They treat MOs as ordinary words and directly apply the existing solutions of POS tagger.

In the NLP community, POS tagging tasks are considered a nearly solved problem using statistical machine learning models [64], [59], [65], [66]. Common features include the value, the preceding (succeeding) of the current word and its neighbors [64]. Due to the difference in the interaction of MO with word, a traditional constituent or dependency parser failed to analyze the syntactical structure of the MWM sentence. As for MO specific syntactic tagging, the work in [67] proposed the first MO-SR tagger using a mixture of naïve Bayesian models based on the format complexity of MO, neighbors POS prediction, and the syntactic properness of the sentence reached a 69% F1 score for the three-class classification of MO-SR. However, their model is unable to predict the POS tags of other words.

The existing solutions for parsing MWM sentences are based on brittle grammar, including the combinatorial category grammar [68] and the typed probabilistic context-free grammar (PCFG) [69]. They both require the semantic analysis of MO, which itself is still a challenge. On the other hand, a data-driven training approach might not be feasible due to scarcity of dependency parsing tree data for MWM sentences. Though it is reasonable to directly extract relation using the dependency parsing structure as done in the protein interaction extraction [70], the errors accumulate at both the POS and parsing steps result in wrong relation catch in natural language. Besides, the performance of the dependency/constituent parsing still face challenges in the multi-word expression [71], the special punctuation [72], and the ambiguity of prepositional phrase attachment and coordinate conjunction attachment [73], [74], [75] even for normal language.

Mathematical Objects and Words Mixed Tagger

POS tagging is an important fundamental work in NLP, which is the foundation of high-level tasks such as phrase extraction and dependency analysis. However, the type of sentences in scientific documents which mixed with unknown words like MOs introduce new usage patterns compared with our daily used natural language. These patterns lead to the degradation of the existing POS taggers [65], [66], which further propagates to high-level analysis such as phrase extraction and syntactic structure parsing. To address this problem, we propose a customized MWM tagger (L1 in Figure 19) to accurately label the words with respect to the syntactic roles of MOs (MO-SR) in the sentence.

The Syntactic Role of Mathematical Object

The mathematical notation system itself could be treated as a language. This implies that one MO could be very complex and even correspond to a sentence or subordinate clause in the contents. Follow the conventions provided in an Elsevier open access dataset OA-STM Corpus [36], there are three categories of MO-SR as shown in Table 3. The existing conventions for the three MO-SR are: S for sentence (main clause) or subordinate clause, NP for noun phrase, and NML for noun modifier.

Table 3: The syntactic role of mathematical object (MO-SR) and examples. Reprinted with permission from [58].

MO-SR	***Example (MO is in bold font)***
S	"Note that $[\boldsymbol{f}]_{\boldsymbol{p}} = [\boldsymbol{f_0}]_{\boldsymbol{p}}$ and $[\boldsymbol{f'}]_{\boldsymbol{p}} = [\boldsymbol{f'_0} - \boldsymbol{f_1}]_{\boldsymbol{p}}$."
NP	"We are given a graph $\boldsymbol{G} = (\boldsymbol{V}, \boldsymbol{E})$."
NML	"This happens $\boldsymbol{lgn}$ times by repeating squaring."

The special syntactic role of MO could not be covered by the conventional POS taggers [43], [44], and about 10% degradation of the POS tagging for other words was also observed. The F1 scores of 0.868 and 0.882 is obtained using the Stanford maximum entropy tagger [44], [66] and the NLTK averaged perceptron tagger [43], [65] in comparison with their 0.973 and 0.971 F1 score, respectively, for non-MWM corpus according to the report of the Association for Computational Linguistics (ACL) in [76]. For example, in Figure 20, the word "prime" is supposed to be the textual declaration of MO_1.

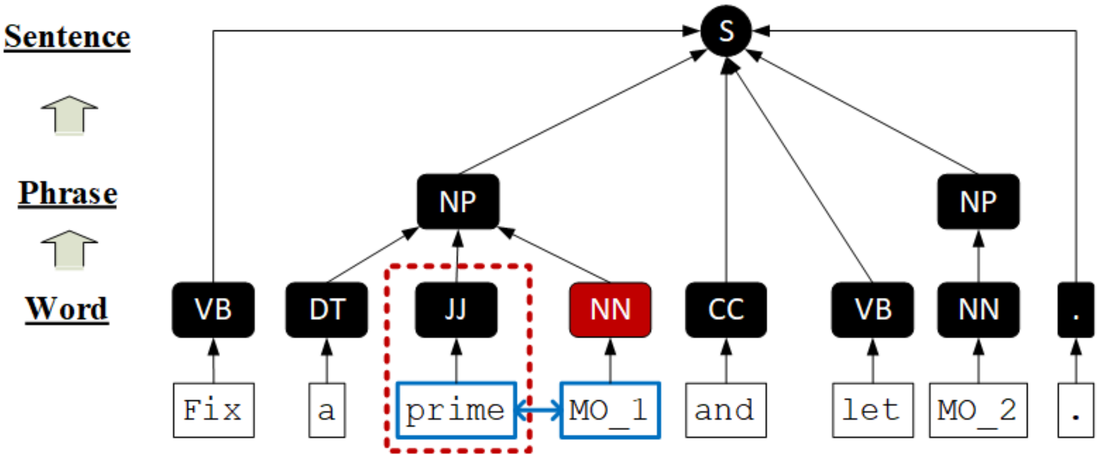
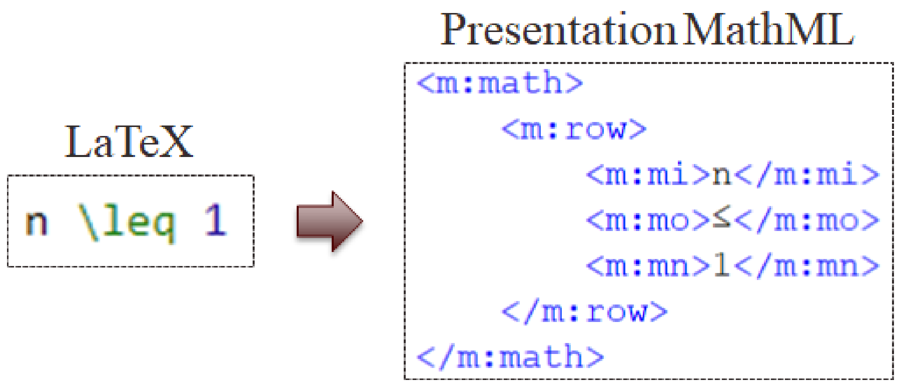


Figure 20: The error propagation from POS tagging in L1 to constituent parsing (i.e., Fail to identify “prime” as the declaration of MO_1). Reprinted with permission from [58].

Given that MOs which correspond to sentences can be very complex, failure to identify their syntactic roles not only leads to mislabeling of POS tags for other words but also propagate the errors to parsing phrase, affecting the NP candidate generation for MOD [57]. For example in Figure 20, the MOD “prime” cannot be detected if the POS of “prime” is mislabeled as adjective (JJ) from the context of the left neighboring word (‘a’) is determiner (DT) and the right neighbor MO (MO_1) is noun (NN), which will mislead the later structural analysis of the sentence in capturing the whole term “a prime MO_1” as an NP rather than relating “MO_1” to “prime” in bold blue line.

A preliminary study was demonstrated on the MO properties with respect to its syntactic role, including the presentation features (i.e., the structural length and depth) in Figure 22 and the content features (i.e., the number of variables and operators) in Figure 23. We applied 600 MOs with LaTeX code annotations from the dataset OA-STM Corpus

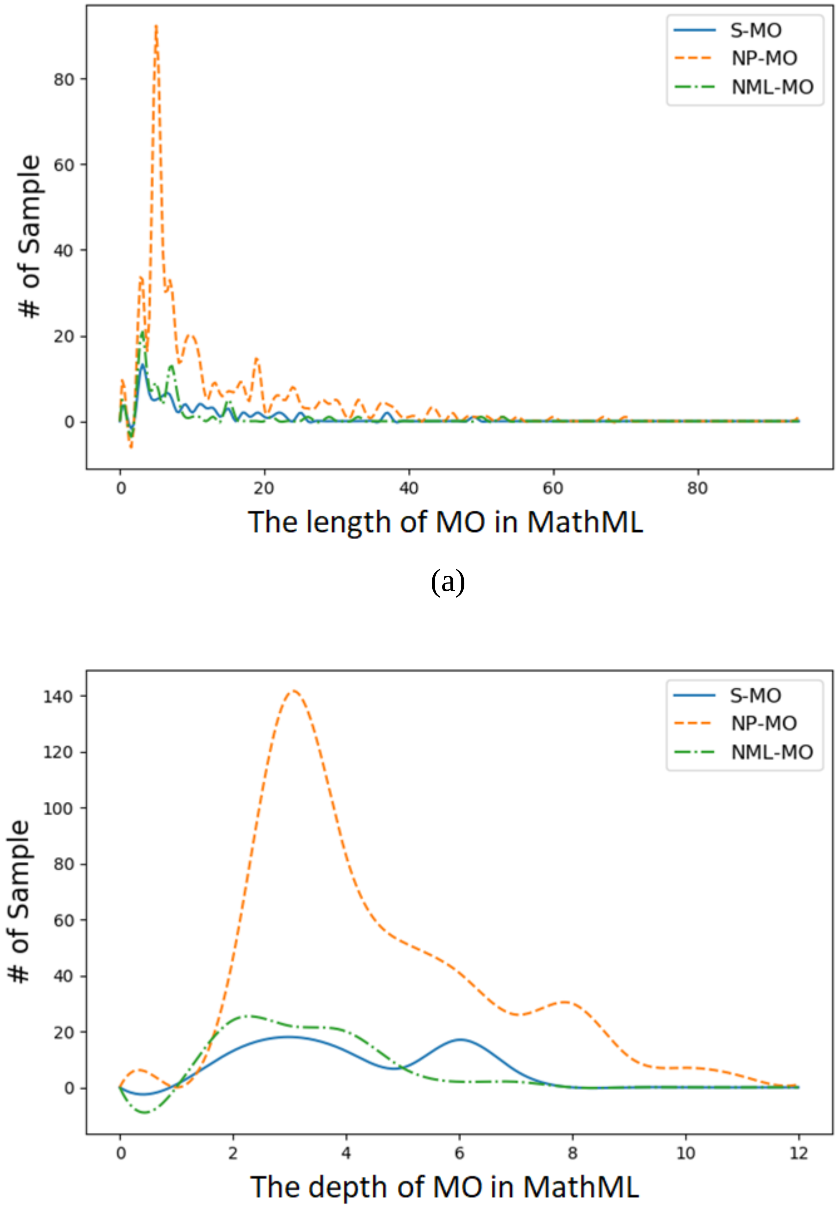
[36] in this study. All MOs are first converted into Presentation MathML (PMML) format via the LaTeXML toolkit [77] as an example given in Figure 21. The presentation features are then calculated based on the PMML structure of MO, where the length is the number of leaf nodes, and the depth is the number of layers starting from the root node. The content features are further obtained based on the number of "mi" node and "mo" node in the PMML structure. As a result, we observed that the curves of the three MO-SR classes are highly overlapped, suggesting that the feature of using MO properties has less discriminant power. Also, our MWM tagger needs to consider the POS tags of other words in context as well. Hence, we proposed a sequential classifier using the Markov property.



```
n \leq 1
```

```
<m:math>
    <m:row>
        <m:mi>n</m:mi>
        <m:mo>≤</m:mo>
        <m:mn>1</m:mn>
    </m:row>
</m:math>
```

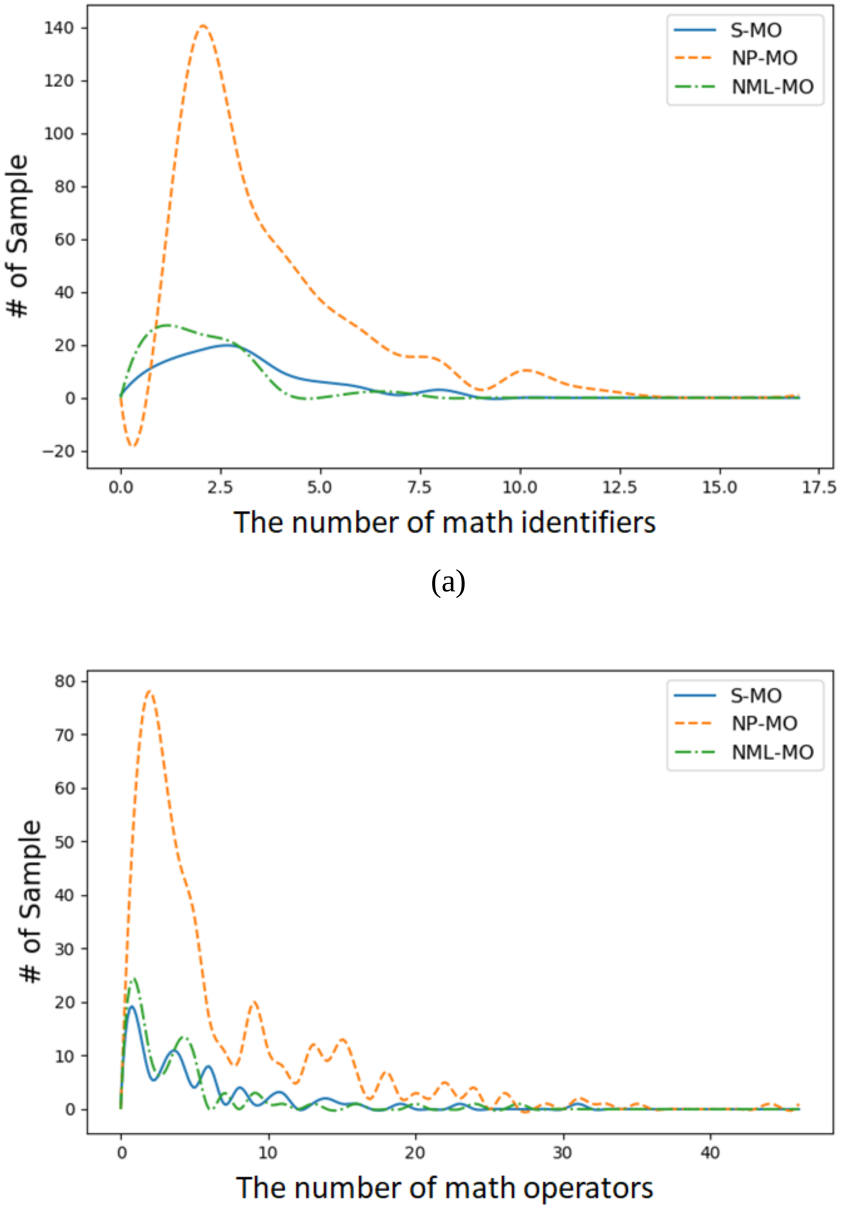
Figure 21: An example of LaTeX code and its corresponding Presentation MathML.



(a)

(b)

Figure 22: Histograms of syntactic roles on the (a) length and (b) depth of mathematical object (MO) in MathML structure.



(a)

(b)

Figure 23: Histograms of syntactic roles on the number of (a) mathematical identifiers and (b) operators.

The Syntactic Tagger for MWM Sentences

The task of POS tagging is to predict the syntactic label t_i for each token x_i in a sentence $s = \{x_1, \dots, x_n\}$ where x_i can be a word or an MO. For a word, the label candidates are the 36 Penn Treebank POS tags such as noun (NN), adjective (JJ), and verb (VB). See Table 9 of Appendix A for more information. When the token is an MO, there are three possible labels: $\{S, NP, NML\}$. The POS tagging is formulated as the optimization goal as follows:

$$\arg\max_t \left[\prod_{i=1}^{n} Pr(t_i | t_{i-1}, t_{i-2}) \prod_{i=1}^{n} Pr(x_i | t_i) \right] Pr(t_{n+1} | t_n)$$

based on the second-order hidden Markov model [59]. Three additional labels x_{-1}, x_0, and x_{n+1} are added to the beginning and end of each sequence. Since trigram instances are sparse, we need to have a smoothing paradigm on estimating $Pr(t_i | t_{\mathrm{i}-1}, t_{\mathrm{i}-2})$ to prevent zero count on trigram cases that never occur in the corpus. Similar to [59], we apply the linear interpolation of unigram, bigrams, and trigrams for smoothing technique to resolve scarcity of the trigram cases in the dataset, which is estimated as:

$$Pr(t_i | t_{\mathrm{i}-1}, t_{\mathrm{i}-2}) = \lambda_1 \widehat{p}\,(t_i) + \lambda_2 \widehat{p}\,(t_i | t_{i-1}) + \lambda_3 \widehat{p}\,(t_i | t_{i-2}, t_{i-1})$$

where $\lambda_1 + \lambda_2 + \lambda_3 = 1$. The $\widehat{p}$ indicates the unsmoothed probability and all λ are estimated using a global context-independent smoothing [59]. For rare words with frequency less than 5, their suffixes (i.e., last m letters) are used to estimate the probability $Pr(x_i | t_i)$. The Viterbi algorithm [78] is a dynamic programming approach used for the efficient prediction of tagging based on tokens.

Table 4: POS tagging prediction performance on MWM sentences in the dataset OA-STM Corpus [36]. Reprinted with permission from [58].

Model	P (%)	R (%)	F1 (%)
CoreNLP [44]	86.56	86.96	86.76
NLTK [43]	90.03	86.52	88.24
Proposed	95.91	95.46	95.68

The MWM POS tagging is evaluated on the Elsevier open access dataset [36] which consists of 10 papers from different disciplines in STEM fields. There are 346 MWM sentences containing 545 MOs mixed in these sentences. A 10-fold cross-validation experiment is designed to test the generalization of the performance. In each fold, we pick one file as the test data set. The other 9 files together with the CoNLL2000 and the Penn Treebank from NLTK [43] are used for training. We achieved a performance of over 0.95 for precision (P), recall (R), and F1 score as shown in Table 4. The performance result has improved the existing POS taggers by 7-9%. Furthermore, the performance of the MO-SR tagging has significantly improved from 0.69 [67] to 0.91 of F1 score.

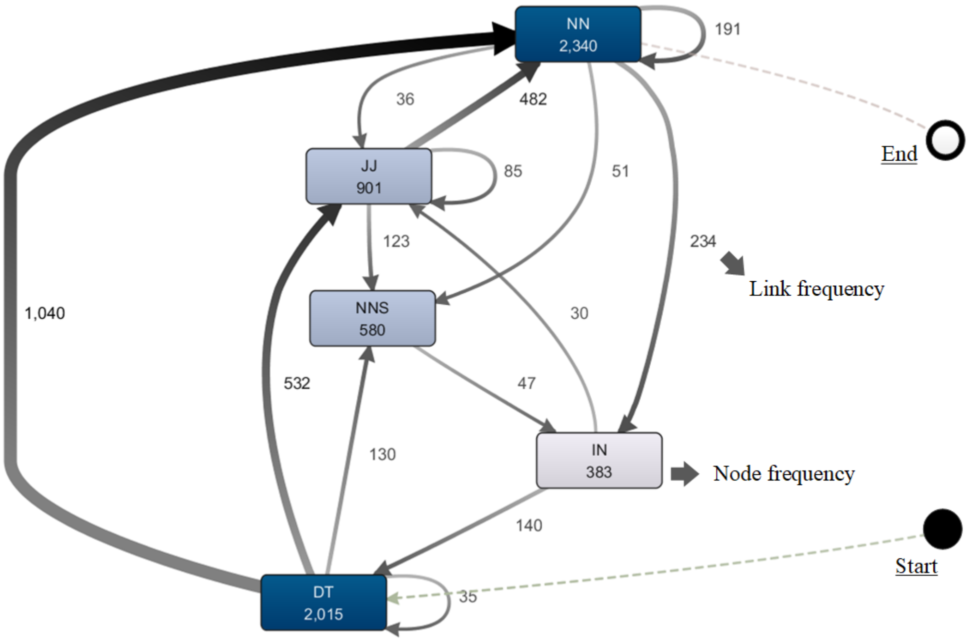


Figure 24: The visualization of process model in Disco [79] using POS tag series of MOD (top 10% frequent nodes and links from the data). Reprinted with permission from [58].

After the POS tag is accurately labeled on each word, we next identify NPs as candidates of MOD using the context-free grammar (CFG) learned from the ground truths of human annotated dataset NTCIR-10 [37] (L2 in Figure 19). We take each word in the MOD as a unit of event and apply the fuzzy process mining algorithm [60] to construct an automata-like structured process model in 3-tuple $T = (V, R, S)$ that simplifies the unstructured sequence of data. The finite set V contains all 50 POS tags of word, and the finite relations R contains any directed link of V to $(V \cup S)$ based on the observation of their order in the MOD. S contains the two terminal states that represent the beginning and the ending of a declaration, respectively (i.e., the Start and End in Figure 24). We display the process model T using the visualization tool Disco [79] as shown in Figure 24. Note that the thickness of link indicates the frequency of the bigram pattern, and the black dot and the concentric circle nodes are the beginning and the ending of the sequence, respectively. From the graph, it shows majority of the MOD (i.e., top 10% frequency of nodes and links) is consists of the POS tags DT (determiner) and NN (noun) such as “a graph”, “the matrix”, etc. It is understandable that some NPs have tags JJ (adjective) before NN and RB (adverb) before NN or JJ. Also, some rare cases connect two NP-chunks with the tag IN (subordinating conjunction) like “the set of vertices” and “the element of matrix”. Hence, we construct an NP shallow chucker by the following CFG using regex:

$$NP \leftarrow \langle NBAR\rangle\langle IN\rangle\langle NBAR\rangle$$
$$NP \leftarrow \langle NBAR\rangle$$
$$NBAR \leftarrow \langle DT\rangle?\,\langle RB.* \,|JJ.* \,|VB.* \,|NN.*\rangle * \langle NN.*\rangle$$

The CFG we propose above has covered around 94.67% of ground truth in the NTCIR-10 dataset [37] we used for evaluating the performance of our proposed MOD model.

Features of MOD Predictions

Human readers can readily recognize MOD based on their prior knowledge on how the authors express in writing practice to relate an MO to its corresponding declaration. To mimic how humans, make this kind of connection, we propose five main features that are considered holding the key to determine whether an NP is referring to an MO (L3 in

Figure 19). As depicted in Figure 25, the first two features are the relative position of the MO with respect to an NP in the sentence. That is, the NP is either at the left-hand side (❶) or the right-hand side (❷) of an MO. The last three features: distance (❸), word stem (❹), and POS tag (❺) are all aim for words in between MOs and NPs, so we put the three together as one level of discussion. Note that the different color-coding zones of the Halo annulus in Figure 25 imply the spatial confidence in making an assertion of any MO-NP pair. The closer distance they are, the more likely they are an MOD pair.

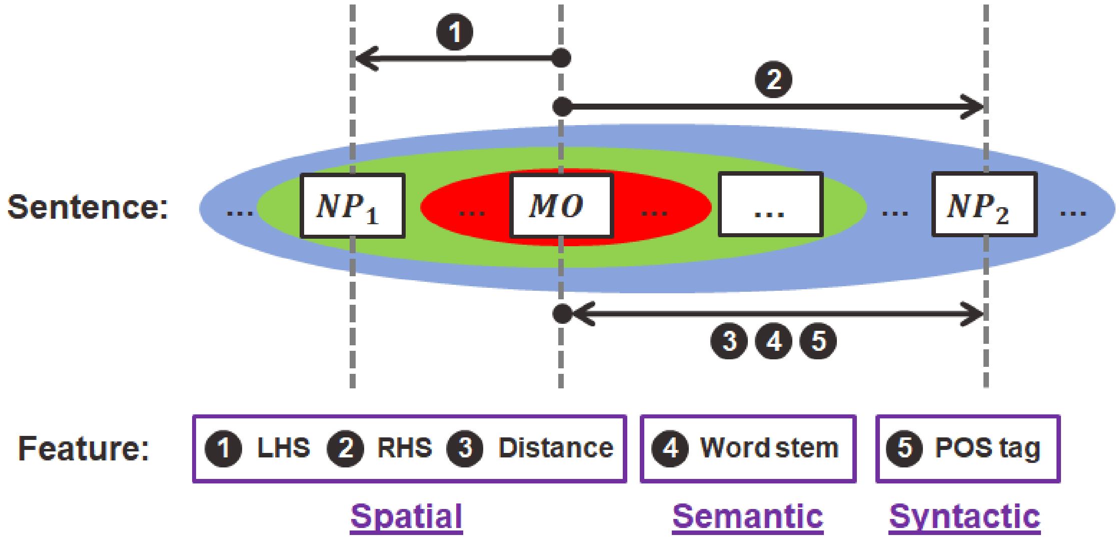


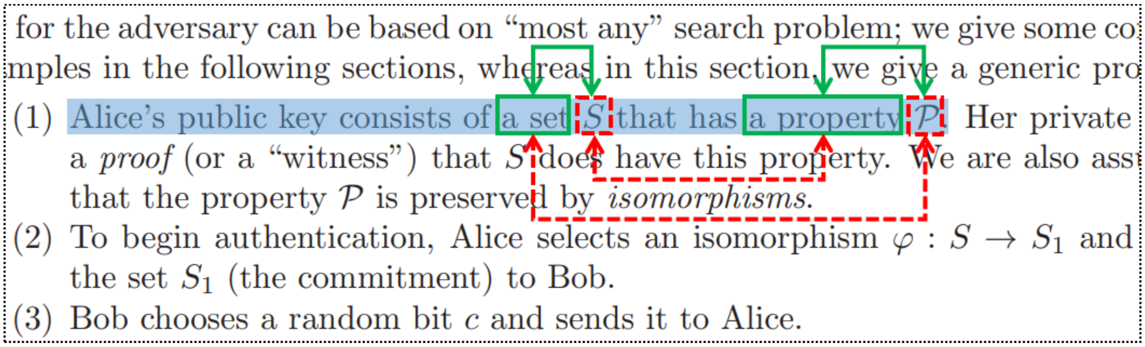
Figure 25: The modeling of spatial, semantic, and syntactic features in a Halo. Reprinted with permission from [58].

Feature Extraction

To decide an MO m is at the left or right of an NP w, we can simply retrieve the starting positions $min(I(w))$ and the ending positions $max(I(w))$ of w where the function $I(w)$ represents the set of indices of each token in w in the sentence, and compare them with respect to m's position $I(m)$. If $max\big(I(w)\big) < I(m)$ ($min\big(I(w)\big) > I(m)$), then NP w is at the left (right) of the MO m in the sentence. Otherwise, the MO are overlapped with its declaration, and this is considered a false case since self-declaration is meaningless.

The distance feature is based upon the number n of words and/or MOs between a given MO and NP, which is $n + 1$ intervals from all n units. As for the word feature, a word can have various forms under different contexts in the content. For example, words like "denote" can have forms "denoted", "denotes", "denoting" based on the tense of verb and the subject/object in a sentence. Another example will be the uppercase/lowercase according to English grammar rule. A word can also have different usages based on its syntactic context (the POS tag). Therefore, we applied the snowball stemming algorithm [51] and the proposed MWM tagger to extract the word stem and the POS tag of each connecting word of MOD, respectively. The word stemming not only helps us normalize the word to its original form to aggregate the words that belong to the same meaning but also captures the stop words that usually have unique form. The POS tag of the word will preserve the word syntactic property and help us identify significant syntactic roles and/or punctuations that connects an MO to its declaration.

One thing that is not brought to attention from the existing works [57], [62], [63] is the enumeration of negative side of instances. By merely studying the positive side of instances will not result in an objective likelihood assertion of the fact. Rather, we face a case that the assertion is based on whether something exist or not, no spectrum information of how the existence of one evidence can lead to the final decision. In this dissertation, the approach we used for sampling the negative instance is by enumerating the complement connections of MOD in each sentence. That is, given a graph G, we examine all the edges in the complement graph $\bar{G}$. To avoid oversampling of negative instances, we select the cases that have the shortest distance to MO. As in Figure 26, the green bold line connects the original ground truth of MOD, and we can obtain the negative samples by studying the complement of the ground truth links, which is, the red dashed line.

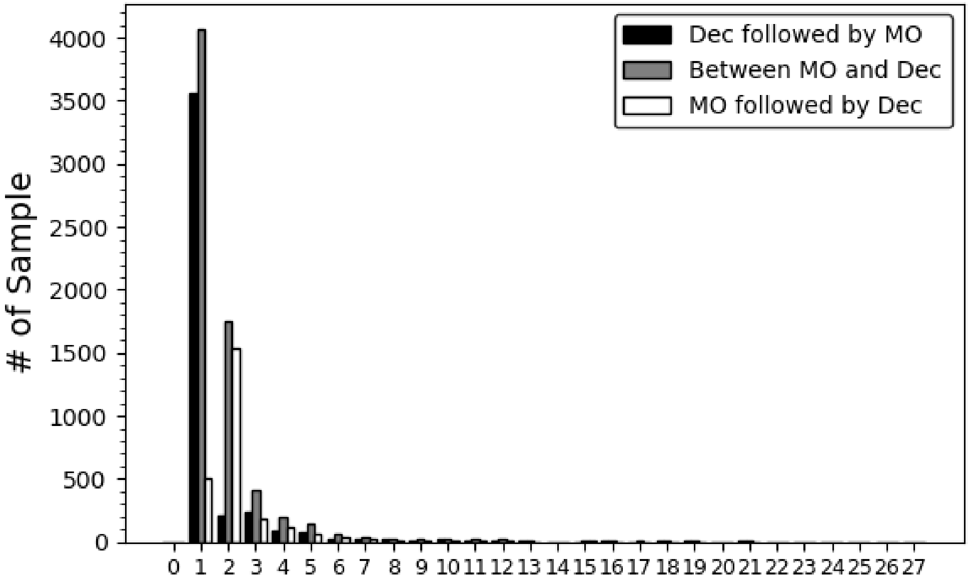


for the adversary can be based on "most any" search problem; we give some co
mples in the following sections, whereas in this section, we give a generic pro
(1) Alice's public key consists of a set S that has a property $\mathcal{P}$. Her private
a *proof* (or a "witness") that S does have this property. We are also ass
that the property $\mathcal{P}$ is preserved by *isomorphisms*.
(2) To begin authentication, Alice selects an isomorphism $\varphi : S \rightarrow S_1$ and
the set S_1 (the commitment) to Bob.
(3) Bob chooses a random bit c and sends it to Alice.

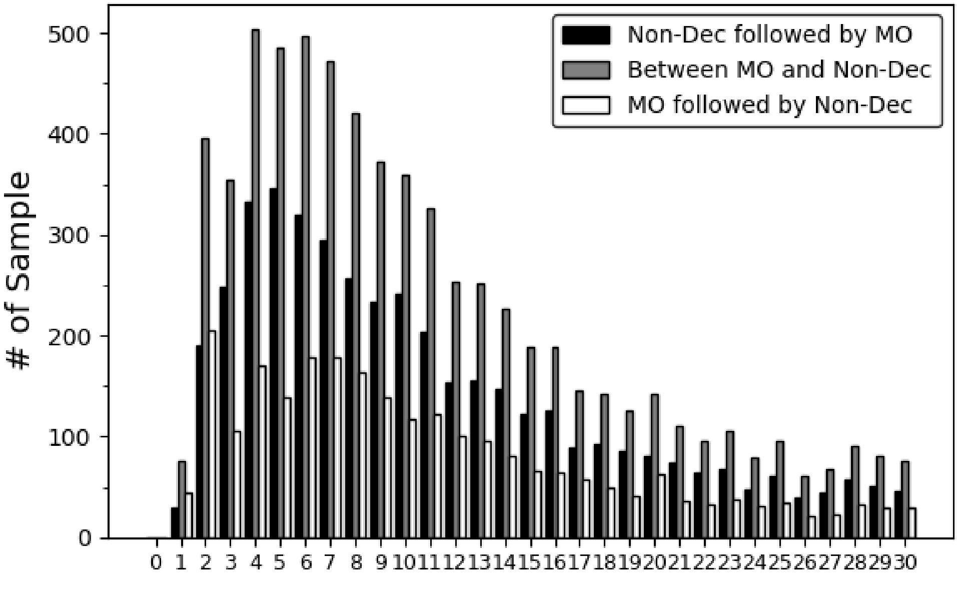
Figure 26: An example sentence (highlighted in blue) for negative sampling (green bold line connects MO to its true declaration, while red dashed line connects to a false declaration). Reprinted with permission from [58].

Analysis of Spatial Feature

Based on the statistics from the annotated NTCIR-10 dataset, the distances between an MO and its declaration are mostly within 6 as shown in Figure 27(a). Some rare cases with distance longer than 10 might be noises in data, or special cases that implicitly and indirectly referring an MOD. On the other hand, the distribution on the negative samples shown in Figure 27(b) also suggested that different distances still exist possible false cases.



(a)



(b)

Figure 27: Histograms of the positive (a) and the negative (b) instances for MOD based on the spatial distances. MO is a mathematical object where Dec is its declaration while Non-Dec is not. Reprinted with permission from [58].

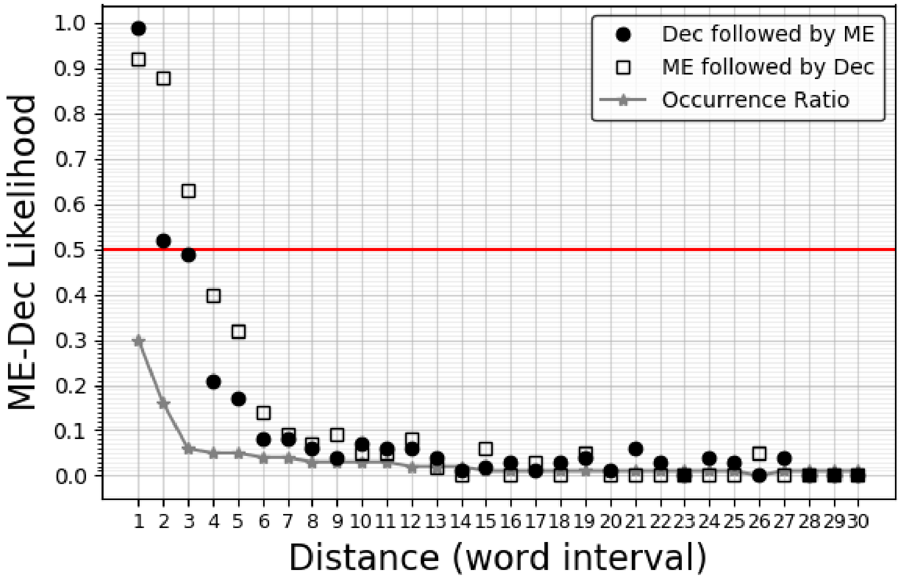


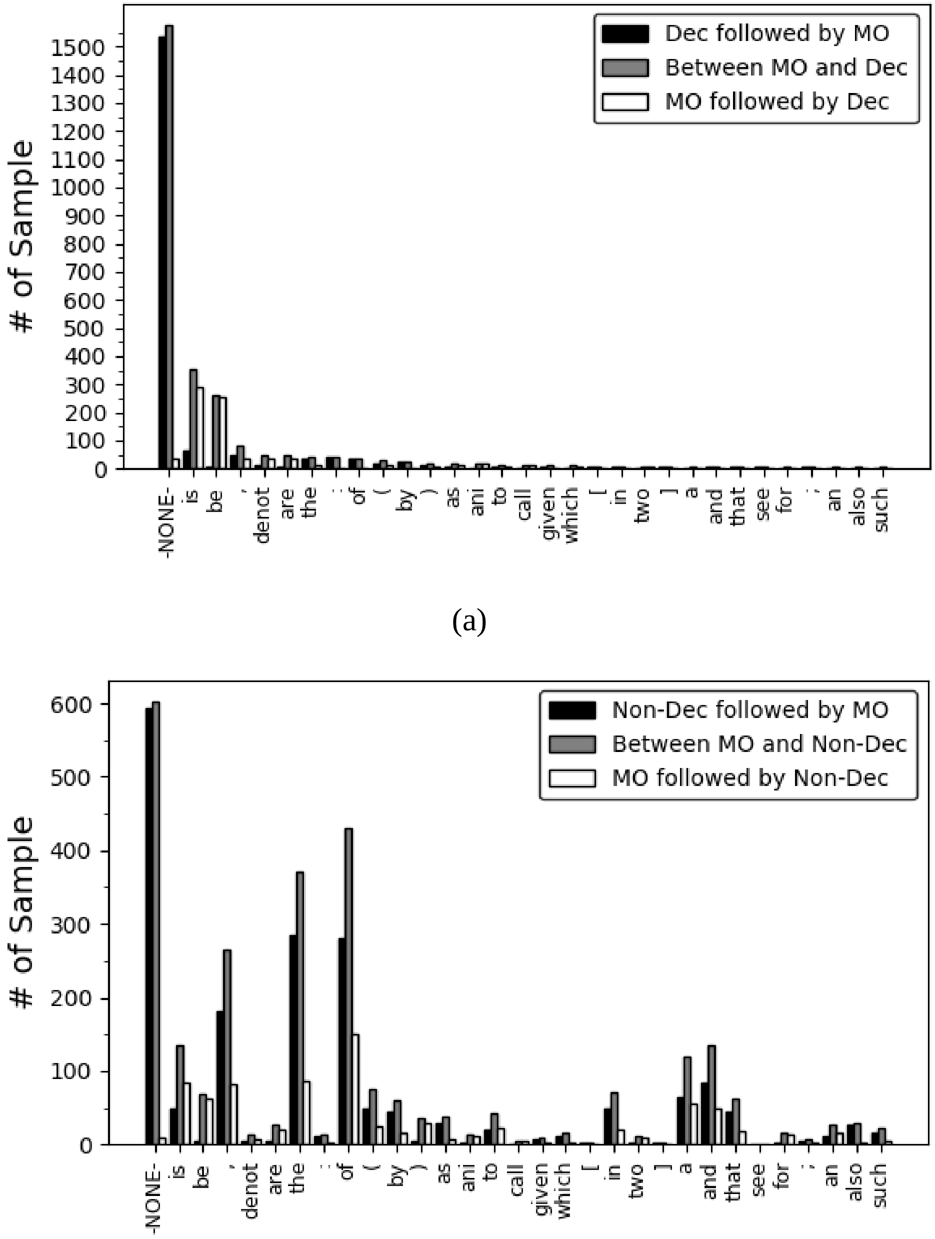
Figure 28: The statistical likelihood of spatial distance between the mathematical object (MO) and its related textual declaration (Dec). Reprinted with permission from [58].

The likelihood spectrum for the distance feature is shown in Figure 28, which is derived from the two histograms Figure 27(a)(b) via $\frac{|P|}{|P|+|N|}$ where P is a set of all positive instances and N is the set of all negative instances. We observed that when an MO and an NP are attached to each other in a sentence, no matter the MO is at which side of that NP, it is nearly certain that they are a match with more than 90%. However, when it comes to distance of 2 words between MO and NP, it is more likely that the MO is at the left than at the right of NP to result in an MOD. This is due to human writing practice that if the declaration follows its MO in a sentence, we usually use active verb phrase to address their coreference relation. For example, “Let MO denotes the Euclidean distance.” where

“Euclidean distance” is the MOD. On the contrary, if the declaration followed by its MO in the sentence, we must use passive verbs to coreference them, which can be more than two words like “The eigenvector is defined as MO.” where “eigenvector” is the MOD.

Analysis of Word Feature

On the analysis of the word feature, both word stem and POS tag have high frequency on the empty set (-NONE-), which support the observation in distance feature that zero-distance has the highest indicating power for positive assertion of MOD. For those declarations that followed by their MOs, the top rank frequent word stems include stop words like “is”, “the”, “of”, “by” as shown in Figure 29(a). These words include POS tags like verbs (VBZ, VBP, VBN), determiner (DT), noun (NN), and preposition (IN) as in Figure 31(a). Some punctuations like ‘,’, ‘:’, and ‘(’ appear also very frequent, which reflects human writing practice in using those punctuations to apply MO for denotation of words. On the other hand, for those MOs that followed by their declarations, the top frequent words are be-verb (“is”, “are”, “be”) and verb words like “denote”. It is interesting to note that only comma has noticeable frequency on the histogram. This suggests a type of writing practices for MOD is to use a comma ‘,’ to connect words to MO such as the way we introduce an acronym in natural language.



(a)

(b)

Figure 29: Histograms of the positive (a) and the negative (b) instances for MOD based on the word stems. Notice that -NONE- represents an empty token, MO is a mathematical object where Dec is its declaration while Non-Dec is not. Reprinted with permission from [58].

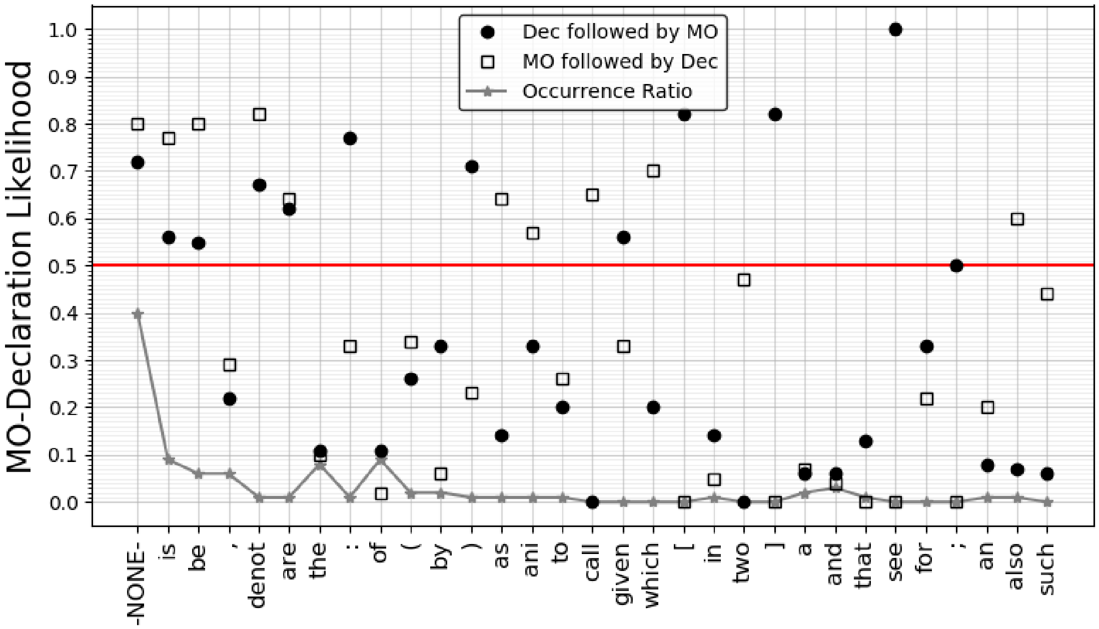
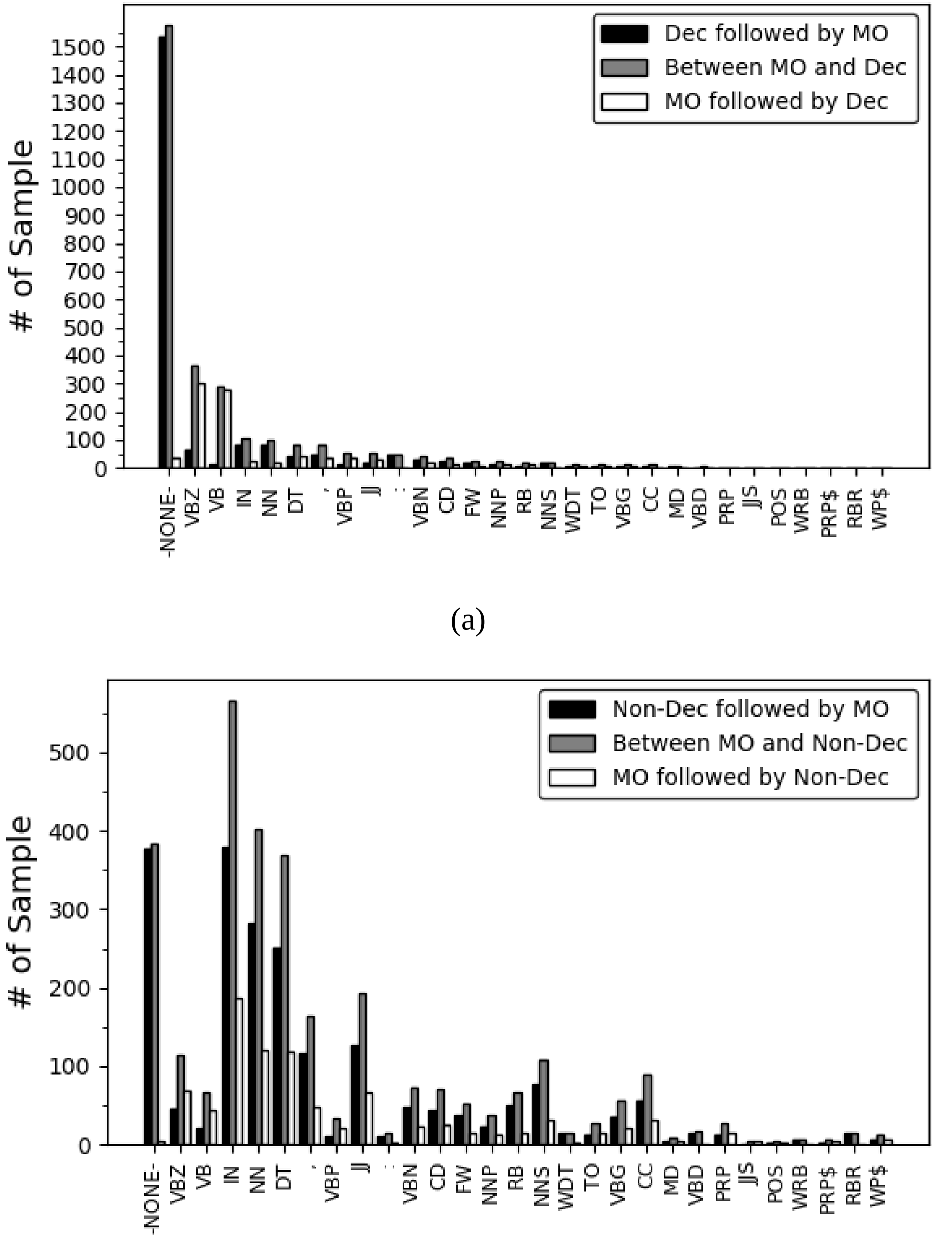


Figure 30: The statistical likelihood of word stem for intermediate words that connect the mathematical object (MO) to its related textual declaration (Dec). Reprinted with permission from [58].

We now discuss the discriminative features that make the decision between the NP at the left and/or right of MO. These require examining the likelihood of both word stem (Figure 30) and POS tag (Figure 32), which are derived from Figure 29 and Figure 31, respectively. When the declaration is followed by MO, it tends to use verb words like "are", "see", "given" to connect an MO. The word "are" is obvious to understand, but "see" and "given" are actually verbs used to attached an MO such as "given $x \geq 1$". There are also cases which use an NP to declare several MOs at the same time. An example sentence would be "The trees MO_1, MO_2, and MO_3" where all three MOs are declared simultaneously. Punctuations like the right bracket ')' is used when more than one NP are denoting the MO, and the colon ':' is directly used for denotation. As for the declaration at the right of MO, the most likely used word stems are “as”, “ani”, “call”, and “also”. Other word like “which” is used for relative clause to further define the subjects in the main clause. The usage of these words can also be seen in top likelihood of POS tags: VB, VBD, and WDT.



(a)

(b)

Figure 31: Histograms of the positive (a) and the negative (b) instances for MOD based on the POS tags. Notice that -NONE- represents an empty token, MO is a mathematical object where Dec is its declaration while Non-Dec is not. Reprinted with permission from [58].

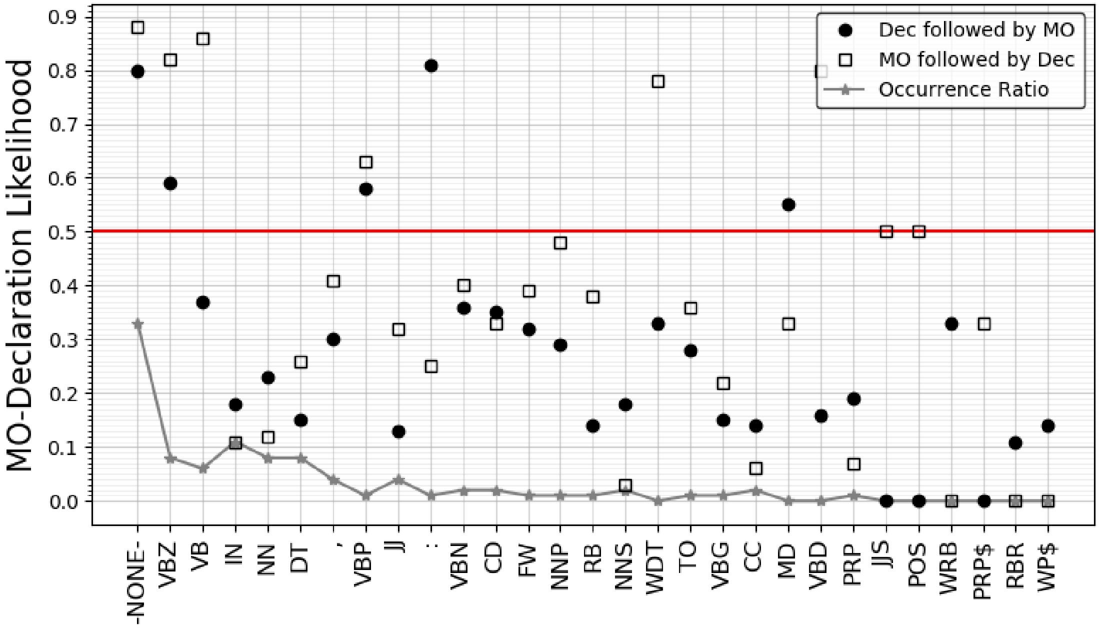


Figure 32: The statistical likelihood of POS tag for intermediate words that connect the mathematical object (MO) to its related textual declaration (Dec). Reprinted with permission from [58].

Bonding of Mathematical Objects and Noun Phrases

In this section, we describe the proposed decision model used for making the inference based on the bonding strength between MO and any NP. After we locate the MOs and the potential MOD candidates of NPs, we first use a template matching that are collected from 1K documents of arXiv using a weakly supervised approach. Then, we apply the naïve Bayesian approach on three features: distance, word stem, and POS tag, based on the MO position with respect to an NP. An ensemble of classifiers based on the majority vote rule will be used to make the final decision.

Template Pattern Matching

Before making any inference, we first apply some golden rule templates in identify the first-hand matching of MOD. There are several steps based on weakly supervised approach to gather the list of declaration templates. The first step is to collect the possible pairs of MOs and their associated NPs from arXiv.org to learn the template patterns that guarantee a match. The type of sentences we considered for learning the template is the ones that contain a single NP with either single or multiple MO(s). The MOs of concern are similar to the works proposed in [54] and [55] that focus on simple variables such as identifiers or identifiers with superscript, subscript or accent. If there is an NP nearby, we commit to a ground truth case for MOD. According to the statistics from the NTCIR-10 dataset, the distance between MO and its declaration are most likely within a distance of 6 as shown in Figure 27(a). When preparing the MOD pairs for the unsupervised arXiv dataset, we set an even stricter threshold that only considers the pair with a distance of less than and equal to 4. In Table 5, we show 8 templates which are at the top of the list of frequency and are manually confirmed to perfectly guarantee the bonding exist when there is a match. Some of these golden rule templates are also introduced in the existing works [57], [62].

Table 5: A list of common template patterns collected from arxiv.org to relate mathematical objects (MOs) to their textual declaration (Dec). Reprinted with permission from [58].

Pattern	Template
1	[MO] {denote(s) | mean(s) | represent(s)} (the) [Dec]
	[MO] stand(s) for (the) [Dec]
2	[MO] {is | are} (the) [Dec]
3	[MO] {is | are} {denoted | defined | given} {as | by} (the) [Dec]
4	let [MO] be denoted by (the) [Dec]
5	denote {as | by} [MO] [Dec]
6	{let | set} [MO] {denote | denotes | be} [Dec]
7	[MO] {, | and | or | [MO]} {are | be} [Dec]
	[Dec] [MO] {, | [MO]}* {and | or} [MO]
8	[Dec] {[MO]}* [MO]

*Braces "{…}" indicate must select one from the given set of items; Parentheses "(…)" indicate with or without; Brackets "[…]" indicate a chunk that must be at that position.

Naïve Bayesian Inference

In this subsection, we introduce the Bayesian inference model to make assertions on the three major types of features (distance, word stem, and POS tag). Based on the assumption of conditional independent probability distributions, the features in each type are separately used for making likelihood assertion of MOD.

Let m be an MO in an MWM sentence and w is a set of words that given as one of the MOD candidates. We would like to estimate the posterior probability of MOD condition on the likelihood of the feature set $e(m,w)$ extracted between m and w. The estimated function can be expressed as follows:

$$Pr\big(\theta|e(m,w)\big) = \frac{Pr(e(m,w)|\theta)P(\theta)}{Pr\big(e(m,w)\big)}$$

where $\theta \in \{0,1\}$ represents positive (1) and negative (0) assertion of MOD for m and w based on one of the three types of features. With assumption of conditional independence, we can obtain the conditional probability

$$Pr(e(m,w)|\theta) = \prod_i Pr(e(m,w)_i|\theta)$$

where i is the ith feature in $e(m,w)$. Note that the posterior probability $Pr\big(\theta|e(m,w)\big) \propto Pr(e(m,w)|\theta)Pr(\theta)$, and $Pr(\theta)$ is the prior probability of MOD as positive instances in the training data. We can derive the labeling result by the likelihood ratio $\rho = \frac{Pr(e(m,w)|\theta=1)}{Pr(e(m,w)|\theta=0)}$ where the predicted $\theta = 1$ (w is likely to be the MOD of m on feature set e) if $\rho > 1$; Otherwise, $\theta = 0$ (w is unlikely to be an MOD of m on feature set e).

A Laplace smoothing [4] is used to estimate the probability of each feature i condition on $\theta = l$, that is:

$$Pr(e(m,w)_i|\theta = l) = \frac{f(e(m,w)_i|\theta = l) + \alpha}{f(e(m,w)_i|\theta) + \alpha(N+1)}$$

where the function $f(e_i|l)$ counts the number of instances for the evidence e_i with respect to the label l. The variable α is a smoothing parameter greater than 0, and $\alpha = 0$ corresponding to no smoothing. The smoothing can resolve the issue that unknown feature (i.e., first time occurrence) result in a zero-probability estimation.

Majority Vote Ensemble

For the final decision of MOD, we apply the ensemble of classifiers based on the majority vote rule to make the final decision whether an MO m and a group of words w are a match of MOD. The voting mechanism follows the objective function $y = \max_{j=1,\dots,C} \sum_{j=1}^{C} \theta_j\big(e(m,w)\big)$ where θ_j is the voting for feature j given the evidence set $e(m,w)$ on the instance of m and w. The final decision can be obtained by the binary variable $d = \left\lfloor y \cdot \frac{2}{C} \right\rfloor$ which indicates the pair (m,w) is an MOD if $d = 1$, or otherwise it is not an MOD.

The inspiration behind using this voting mechanism is the observation that spatial, semantic, and syntactic properties are independent features which complement each other like a triangular rule in making a more accurate assertion. The rule of thumb is that the words used for connecting an MO to a declaration must carry the semantics of such intention under certain syntactic properties within some spatial constrains. For example, we know that verbs operate the relations from one entity to another in a sentence. However, not all verbs are used for connecting one to another. Hence, a better assessment can be made if we know the type of verbs and words that are used under acceptable placement and distance constraints.

Experimental Results and Analysis

In this section, we first introduce the existing dataset used for evaluation. Then, we discuss the performance metrics that are used for comparing with the existing work. Finally, a discussion on the outcomes is given.

Dataset and Evaluation Criteria

The dataset we used to train our predictive model is from the NTCIR-10 math understanding annotation project [37]. There are approximately 35 papers with a total 9172 MOs sampled from the arXiv website. There are two types of annotation: short declaration and full declaration of MO which are taken as MOD in this paper. For the sentence like “Let MO_1 be a virtual link diagram with minimal genus one.”, “a virtual link diagram with minimal genus one” is called a full declaration, while the core, “a virtual link diagram”, is called a short declaration. There are 3076 short declarations and 3053 full declarations with two evaluation modes: strict matching and soft matching. The strict matching requires exact matching of every word in the annotation, while the soft matching only requires partial overlapping. If our prediction is “a virtual link diagram” for the above example, we get a false positive under the strict evaluation mode for the full declaration and a true positive sample for the other combinations. The evaluation criteria presented on Table 6 are precision (P), recall (R), and F1 score.

Result and Discussion

The performance comparison of the methodologies is made at two aspects: strict matching and soft matching. Each type of matching was assessed on the full declaration and short declaration of ground truth dataset. In general, soft matching could work better than strict matching due to its mechanism of a hit is more flexible than the strict one. From the view of methodologies, the comparison is made between different approaches in [62], including: baseline method (the nearest nouns), pattern matching method (templates), and machine learning based method (the support-vector machine, SVM). The features for the machine learning method have been enumerated over different combinations to examine features which yield the best classification results. For simplicity, we only reported their best results for the machine learning method in Table 6.

Table 6: The average performance of mining the textual declarations of mathematical objects. Reprinted with permission from [58].

	Strict Matching (%)			**Soft Matching (%)**		
	P	**R**	**F1**	**P**	**R**	**F1**
Baseline Method (nearest noun) [62]						
Short Declaration	32.20	26.09	28.82	46.19	37.41	41.34
Full Declaration	27.40	22.20	24.53	46.33	37.53	41.47
Pattern Matching Method (template) [62]						
Short Declaration	17.84	24.03	20.48	46.22	62.36	53.09
Full Declaration	19.80	26.66	22.72	46.39	62.59	53.28
Machine Learning based Method (SVM) [62]						
Short Declaration	**84.25**	52.63	64.79	**91.76**	57.32	70.56
Full Declaration	**76.28**	40.85	53.20	**90.60**	46.57	61.52
Proposed Method (combined Bayesian)						
Short Declaration	73.44	**71.25**	**71.58**	76.31	**75.31**	**75.20**
Full Declaration	73.42	**71.23**	**71.56**	76.35	**75.36**	**75.25**

The experimental results show that both the baseline method and the pattern matching method have low performances on P, R, and F1 score. In Table 6, we found that the results of our proposed combined Bayesian model have a performance boost on both short declaration and full declaration compared to state-of-the-art work using SVM [62]. The short declaration (full declaration) improved by 6.79% (18.36%) and 4.64% (13.73%) F1 score in average for strict matching and soft matching, respectively. The reasons for these performance boosts lie in three aspects: First, the MWM processing has significantly improved word-level tagging performance in general (see Table 4), which leads to higher level analysis improvement on capturing the MOD candidates. Second, though the feature used in this paper has been considered in existing works, they did not consider the negative instance that could exist conflicts with the ground truth. Third, the majority vote of the three features can complement each other's inference in making a more precise decision. Furthermore, it is worth knowing that our model has higher sensitivity (recall) on returning the results that cover more MOD cases, but lower correctness (precision) in locating the actual ones compared to the best existing work [62].

Summary

In this chapter, we have identified two bottlenecks for the current works on the extraction of MOD, i.e., the processing of MWM sentences, and the negative sampling to constitute the likelihood of any indicators in the proposed features. The customized MWM tagger and noun phrase (NP) extractor have been proposed to enhance the preprocessing phase of MOD. Negative instances have been enumerated over the positive cases to learn the likelihood of any possible indicators. Evaluation of the Elsevier dataset OA-STM Corpus has shown that the proposed MWM tagger could significantly enhance the POS tagging performance for MWM sentences. The declaration extraction performance has also greatly improved using the NP candidates generated from the customized processing toolkit. The experimental results have shown a margin of 5-18% performance gain on the F1 score compared to other state-of-the-art works. However, the golden rule template patterns manually enumerated are not complete, and it is desirable to have an automated or at least semi-automated method to collect the declaration patterns, which might help the larger-scale experiment in the near future.

CHAPTER V

REASONING GRAPH OF MATHEMATICAL OBJECT

This chapter introduces a mathematical object (MO) based analytical framework for segmentation, interrelation, and reduction of the technical contents. A hierarchical data structure called the MO reasoning (MOR) graph is proposed as a baseline model to study the potential features of document fingerprints. It is a compact representation of the original content that consists of MOs, words, and their interrelations. The dependencies among MOs shape the reasoning flows of a scientific paper. The goal of this MOR graph is to provide new perspectives and foundational solutions for more large-scale content analysis problems such as plagiarism detection and writing style modeling, etc. Some case studies are also given in this chapter to demonstrate the usefulness of the MOR graph for cross-paper analysis.

Overview

Technical writing is a practice of transforming a set of non-linear interrelated abstractions into a linear sequence of mixed symbols and words based on rules in mathematical language [68]. To digest the complex idea of a scientific paper, researchers sometimes need to study the contents back and forth to reconstruct the non-linear relations of MOs from their original linear elaborations. They even need to look up external materials to fully understand the original idea in depth. Missing a subtle piece of technical point may impede a reader from capturing the technical essence of a paper. Being able to automatically discover highly related technical concepts, which mostly consist of MOs and words from scientific papers, will greatly improve the research productivity. To achieve this goal, we apply our algorithms [50] in Chapter III and [58] in Chapter IV to extract the constraint expressions and textual declarations of MOs from documents to assist in recovery of their functional structures.

Technical writers often exercise a divide-and-conquer process to organize the sectioning and ordering of the contents [80], [81], [82]. They first divide a research problem into several sub-problems that individually forms a self-contained technical discussion called the reasoning block (RB). The findings or conclusions of each RB can later consolidate based on the logic flows of the RBs to conquer the discussion of the main research problem. In this chapter, we apply a math-centric analytical framework using MOs as key indicators along with linguistic and layout constraints to analyze the characteristics of an RB. Furthermore, we analyze the dependencies among MOs to reestablish the reasoning logic flows between RBs. A hierarchical data structure, the MOR graph, will be constructed as the technical essence of a paper to apply in various case studies of cross-paper analysis.

Our experimental outcomes showed that the number of MOs and their associated words could be very dense. Sometimes, the large number of MOs in the visualization may interfere with the understanding of the critical logic flows carried by important MOs that are the minority. In order to optimally capture the breadth and depth of technical substance, a systematical approach is essential to group the MOs in hierarchical layers and unclutter the MO-to-MO and MO-to-words relations suitable for human users to read.

Related Works

Recovering the mathematical logic of publications has already been studied by literature at coarse and fine levels. At the coarse level, MO-centered content analysis adopted string matching to extract the mathematical component such as definition, theorem, lemma, etc. and extract the dependencies among the blocks [30], [31], [32]. However, the coarse-level mathematical structures failed to capture many important elements such as the MOs and the interactions between MOs and words. At the fine level, the current state-of-the-art research [56] used the dependencies among MOs to connect all relevant MOs and words for improving the performance of the search engine.

There are three aspects of criteria need to be considered in designing the MO dependency graph: normalization, interrelation, and information overloading. The first two aspects are both related with the common practice of using representation markup language like LaTeX or Presentation MathML (PMML). The LaTeX and PMML are very flexible to produce the same presentation in various encodings. Furthermore, there is a gap between the layout representations and the semantics. Heuristics have been proposed to normalize PMML [56] such as removal of structures (groups, parentheses, attachments, right-hand side MO) and case normalization. After the normalization, the MO dependencies is mostly constructed over the normalized representations based on string matching or subexpression matching (base form or left-hand side). For the last criterion, the dependency graph constructed by [56] is not designed for human to read, so the information overloading issue still remains in their product.

Recently, we developed the first prototype of graph abstraction called the QuQn map [33] to highlight the technical essence of a scientific paper. The QuQn map is used in educational contexts to interactively render the words and MOs based on their qualitative and quantitative dependencies. Similar projects that enhance reading experiences include Utopia [83] and Math-vis [84]. Utopia is used in medical domain by connecting external resources such as terminology dictionaries during the reading process. Math-vis provide single visualization of MathML and differential analysis of a MathML pair.

The QuQn map proposed a normalization process to convert the MOs into a semantic taxonomy [85] like Content MathML (CMML) to avoid the error-prone and ad-hoc normalization that appeared in [86]. It applied basic pruning strategies on the links and nodes to address the problem of information overloading. However, the spaghetti-like MO dependencies still contain too much information for end-users to consume. Hence, we proposed a new compact representation in this chapter to allow hierarchical structure of MO dependency graph using the segmentation, interrelation, and reduction processes. Through our model, the original complex structure of the graph will be progressively reorganized from fine details into coarse representations.

Revisit the QuQn Map System[3]

QuQn map [33] is an abstraction to describe the technical essence of any scientific paper. It uses spatial layout and color style to highlight the dependency relationship among automatically discovered MOs and words. The first processing step of the QuQn map is MO extraction and normalization. Given an MO expressed in LaTeX, MathML or PDF, MOs are extracted, parsed, and converted into a semantic taxonomy structure [85] to further decomposed into sub-expressions for MO dependency analysis. Each MO paired with its semantics by template matching is compared with other pairs to generate their dependencies.

Mathematical Object Processing

The QuQn mapping system includes a preprocessing normalization step to convert MO representations into the notion of semantic taxonomy [85], which is based on the Content MathML standard [87], with extensions for special fields. See Figure 53 of Appendix B for more information.

Succinctly put, MO can be organized into atomic expressions or compounded expressions. An atomic expression can be a constant, or an identifier with optional subscript, superscript, and accent. A compounded expression can be the form of a relation, a function application, or a binding variable. As defined in semantic taxonomy, a function is a general concept that may include common operations, such as addition, multiplication,

[3] Reprinted with permission from “QuQn Map: Qualitative-Quantitative Mapping of Scientific Papers” by Xing Wang, Jason Lin, Ryan Vrecenar, and Jyh-Charn Liu, 2018. *Proceedings of the 18th ACM Symposium on Document Engineering*, Pages 1-4, Copyright 2018 by ACM.

and others. The compounded expressions include domain-specific expressions, such as DeltaVar (calculus), Function (functional analysis), LogicExpression (logic), and ProbExp (probability). The grammar can be dynamically expanded as needed. Based on the semantic taxonomy structure of MO, the QuQn mapping system implemented a parser which can take MOs in XML [77] or PDF [34] as input formats to map to the category of semantic taxonomy.

Each expression of the taxonomy is composed of one or more components for further decomposition. For example, a function application expression may consist of a function and its arguments. The complete grammar for composition of expression can be found in [88].

Notation Definitions

The QuQn map is represented by a paired set $\langle X, Y \rangle$, where $X = \{x_i\}$ denotes the set of MO nodes and their sub-expressions in a document $\mathcal{D}$, and Y is the set of MO denotations which can be a word description or other alternative MO node. Here, the denotation refers to a semantic level declaration of MO. Extraction of equivalent or related MOs from $\mathcal{D}$ requires an understanding of the semantics of MO. As such, the semantic taxonomy structure of MO and the notion of "equal", "sub-component", and "left-hand side" of MOs are used for analysis. They represented the formulation of MO denotations that link the MOs to other MOs or words.

Denotation Extraction

Denotation refers to anything that has semantic equivalence with an MO. A denotation of MO can be a textual declaration expressed in words, or a quantitative description expressed by another MO. Denotation is critical for detecting the relations between MOs and linking an MO to their related words. The three concepts for MO denotation and MO relation extraction are listed as follows:

- $x_i = x_j$ if the two MOs are the same.
- x_i is a subexpression of x_j, denoted as $x_i \in x_j$.
- The left-hand side (LHS) and right-hand side (RHS) function are used to represent MO types such as relation expression and function declaration. For example, $LHS(x_i) = x_j$.

The set of MO denotations Y_X is constructed so that each element represents an MO x that contains two subexpressions $\mathrm{LHS}(x)$ and $\mathrm{RHS}(x)$ with an equal relation "=" in which the denotation is expressed as $\langle \mathrm{LHS}(x), \mathrm{RHS}(x) \rangle$. Besides MO denotations, every MO x_i in a $\langle X, Y \rangle$ set may be optionally associated with a sequence of words $W_i = \{w_i^j\}$ as the textual declaration. The QuQn map adopted and implemented the rule-based model in [57] to extract the textual declaration of $\langle x, W \rangle$ to form the set of word denotations Y_W for a document.

Skeleton Graph Construction and Pruning

The $\langle X, Y \rangle$ set represents significant reduction of information from its original document. It is called the QuQn map when represented in a graph format. When all elements in $\langle X, Y \rangle$ are included, the graph can become too large with low-level details as well as repetitive occurrences of certain MOs and words.

A skeleton graph is proposed in QuQn map to improve its readability. It consists of MOs as nodes with the optional textual declarations placed alongside as shown in

Figure 33. The links between MOs are their dependency relationships. An edge $\langle x_i, x_j \rangle$ means x_i (or $LHS(x_i)$) is a sub-expression of x_j (or $RHS(x_j)$). The skeleton graph of QuQn map is pruned based on two criteria: (1) Keep only the MOs with denotation for users to understand the semantics of every MO node; (2) Remove duplicate occurrences of an MO. To meet the first criterion, the QuQn map only retain the MO $X_y = \{x: \langle x,* \rangle \in Y_X \cup Y_W\}$. To meet the second criterion, we only keep those MOs once at their first occurrence. For those MOs with multiple equivalent denotations, we only keep the first appeared denotation. After the above process, a dependency graph is created as a skeleton of the QuQn map.

Although the pruning conditions have eliminated a good number of nodes, the dependency graph is still too large and complex for visualization. Hence, we employed a series of post-processing on the graph to reduce its size and complexity.

- Keep the longest path between any two MO nodes: If there exists two paths $\{\langle x_i, x_j \rangle, \langle x_j, x_k \rangle\}$ and $\{\langle x_i, x_k \rangle\}$ from x_i to x_k, the shortest path $\langle x_i, x_k \rangle$ is redundant because the longest path already implies that x_i can reach to x_k via the intermediate node x_j.
- Keep the largest connected components: remove the local discussions that are not connected to the main piece of logic flow.

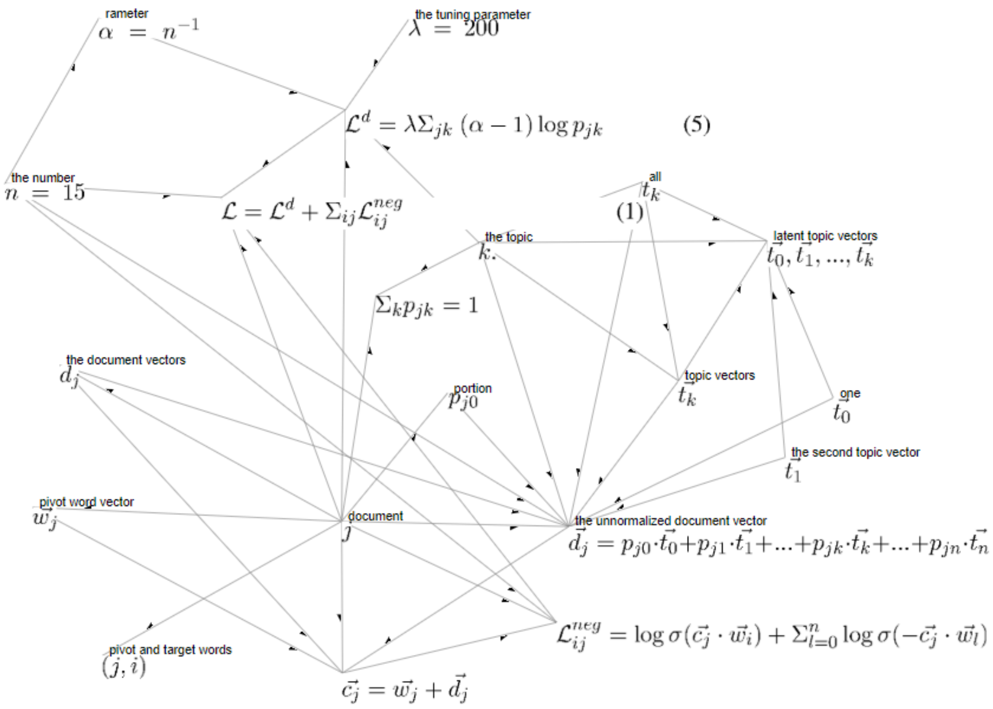

rameter
$\alpha = n^{-1}$
the tuning parameter
$\lambda = 200$
$\mathcal{L}^d = \lambda\Sigma_{jk} (\alpha - 1) \log p_{jk}$
(5)
the number
$n = 15$
$\mathcal{L} = \mathcal{L}^d + \Sigma_{ij} \mathcal{L}_{ij}^{neg}$
all
$\vec{t_k}$
(1)
latent topic vectors
$\vec{t_0}, \vec{t_1}, ..., \vec{t_k}$
the topic
$k.$
$\Sigma_k p_{jk} = 1$
the document vectors
$\vec{d_j}$
portion
p_{j0}
topic vectors
$\vec{t_k}$
one
$\vec{t_0}$
the second topic vector
$\vec{t_1}$
pivot word vector
$\vec{w_j}$
document
j
the unnormalized document vector
$\vec{d_j} = p_{j0} \cdot \vec{t_0} + p_{j1} \cdot \vec{t_1} + ... + p_{jk} \cdot \vec{t_k} + ... + p_{jn} \cdot \vec{t_n}$
$\mathcal{L}_{ij}^{neg} = \log \sigma(\vec{c_j} \cdot \vec{w_i}) + \Sigma_{l=0}^{n} \log \sigma(-\vec{c_j} \cdot \vec{w_l})$
pivot and target words
(j, i)
$\vec{c_j} = \vec{w_j} + \vec{d_j}$

Figure 33: The graph skeleton of QuQn map on arXiv document 1605.02019.

Visualization of the QuQn Map

Given the graph constructed and reduced, the next task is to visualize the QuQn map so that users could quickly identify the essential elements and their details. Following the visual program concept proposed by Tufte [89], we use the spatial and color within a 2D space to make readers quickly identify the dependency relationships among MO nodes.

Spatially, we group the MO nodes into layers based on the depth of the MO in the dependency tree. Since a recursive definition is rarely allowed in scientific elaboration, we remove some edges to make the graph acyclic as a tree, in which the large MO node is composed of smaller MO nodes. The depth of each tree node is a good indicator of how complex the MO is. Furthermore, grouping the MOs into layers could reduce the possibility of crossing edges so that the graph is easier to follow.

From the aspect of color, we use the same color c_i for the same MO x_i across the whole filtered graph to help people quickly identify the same MO in different places. The color c_i will also be used as the color of bounding box for x_i to indicate the first time the MO x_i is presented. Furthermore, the hue contrast of neighbor MOs is maximized to enhance the differentiability for easy identification. The key challenge is that we only have limited colors, so we use the color wheel concept and select the color that is distant from existing color in the angle with smallest standard deviation to enhance the contrast in a limited number of color choices as shown in Figure 34. For example, if an MO only has one neighbor of the blue-violet color, the MO will be assigned the yellow-orange color to maximize the contrast. If an MO is with two neighbor of color red-orange and blue-violet, the MO will be assigned the yellow-green color to maximize the total contrast as well as minimize the standard deviation of contrast.

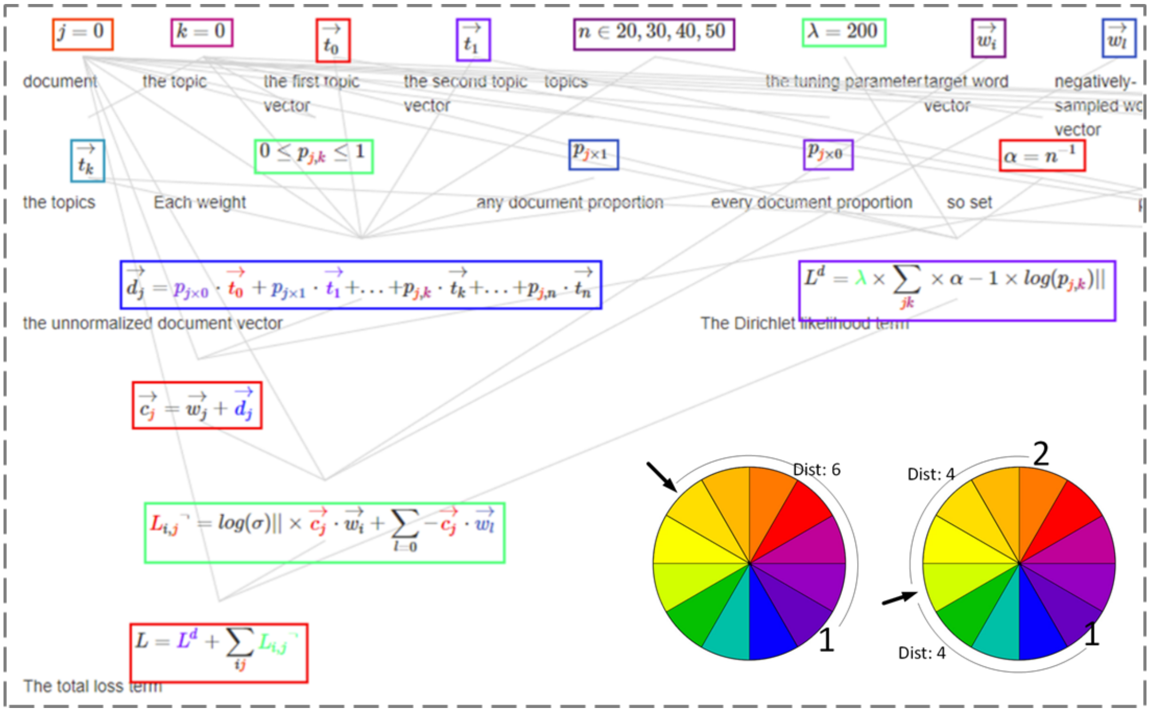

$j = 0$
$k = 0$
$\vec{t_0}$
$\vec{t_1}$
$n \in 20, 30, 40, 50$
$\lambda = 200$
$\vec{w_i}$
$\vec{w_l}$
document
the topic
the first topic vector
the second topic vector
topics
the tuning parameter
target word vector
negatively-sampled wo
vector
$\vec{t_k}$
$0 \leq p_{j,k} \leq 1$
$p_{j\times1}$
$p_{j\times0}$
$\alpha = n^{-1}$
the topics
Each weight
any document proportion
every document proportion
so set
$\vec{d_j} = p_{j\times0} \cdot \vec{t_0} + p_{j\times1} \cdot \vec{t_1} + \ldots + p_{j,k} \cdot \vec{t_k} + \ldots + p_{j,n} \cdot \vec{t_n}$
the unnormalized document vector
$L^d = \lambda \times \sum_{jk} \times \alpha - 1 \times log(p_{j,k})||$
The Dirichlet likelihood term
$\vec{c_j} = \vec{w_j} + \vec{d_j}$
$L_{i,j}^{\,} = log(\sigma)|| \times \vec{c_j} \cdot \vec{w_i} + \sum_{l=0} -\vec{c_j} \cdot \vec{w_l}$
$L = L^d + \sum_{ij} L_{i,j}$
The total loss term
Dist: 6
1
Dist: 4
2
Dist: 4
1

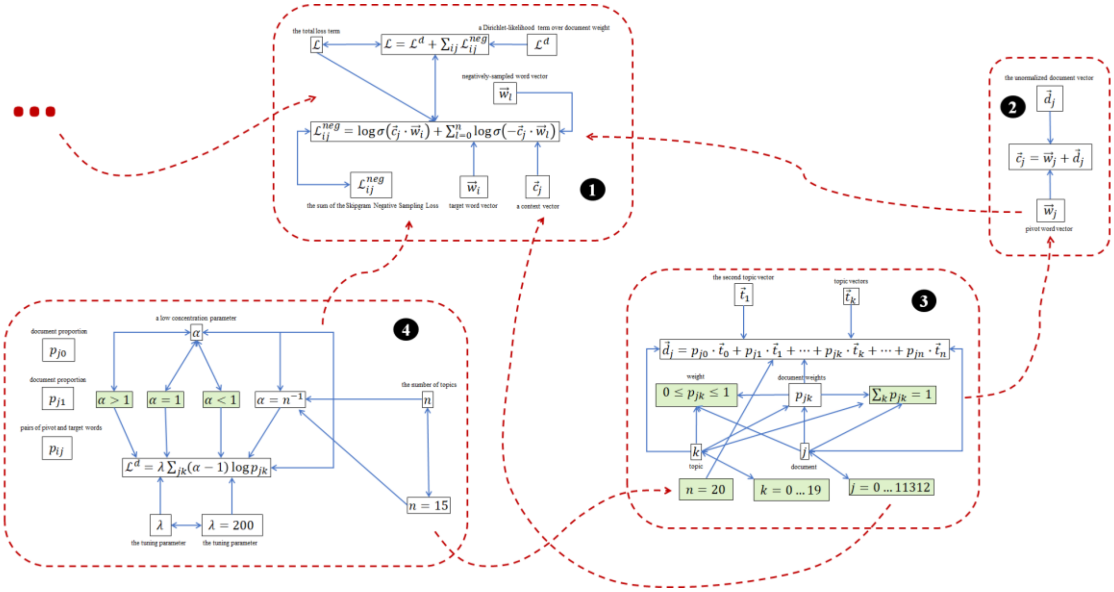
Figure 34: The graph visualization of QuQn map on arXiv document 1605.02019 using the color wheel to highlight the components.

The Limitation of QuQn Map

The construction of QuQn map have largely reduced the complexity of the MO dependency graph proposed in [56]. However, the resulting graph is still complicated in most cases for user to grasp the technical essence. Also, the denotations of MOs only consider MO with limited relations such as equal sign ("="), which have excluded some other possible constraint denotations in the discussion. It is well known in linear programming that the same objective function with different constraints or conditions will result in different solution sets. Besides, the QuQn map does not serve the purpose of modeling the reasoning flows in a scientific paper. The positions of MOs in the contents are lost during the construction processes of the graph. There is no ordering information of how the MOs are originally elaborate in QuQn map. Therefore, we proposed a novel representation, the MOR graph, to address the above issues.

Construction of the MOR Graph

In this section, we introduce the concept of MOR graph to model the reasoning flows in scientific papers. The objective is to offer a compact representation to depict different granularities of technical details for the MO-based scientific documents. On top of the MOR graph is the modularized design so that it could easily be expanded to progressively show different levels of information from the coarsest skeleton graph to the most detailed overlays on the original documents. It has a two-layer hierarchical data structure where the first layer is blocks of self-contained contents known as the RB, and the reasoning logic flows between RBs are based upon the MO dependencies as shown in Figure 35. The underlying skeleton of the MOR graph contains the MO dependencies along with their most related adjacent words similar to the concept of QuQn map. The construction of the MOR graph considers all three aspects of graphic design (normalization, interrelation, and information overloading) to reduce the complexity of the overall structure. We will introduce the segmentation, interrelation, and reduction process for generating the MOR graph in the rest of the subsections.



the total loss term
$\mathcal{L}$
$\mathcal{L} = \mathcal{L}^d + \sum_{ij} \mathcal{L}_{ij}^{neg}$
a Dirichlet-likelihood term over document weight
$\mathcal{L}^d$
negatively-sampled word vector
$\vec{w}_l$
$\mathcal{L}_{ij}^{neg} = \log\sigma(\vec{c}_j \cdot \vec{w}_i) + \sum_{l=0}^{n} \log\sigma(-\vec{c}_j \cdot \vec{w}_l)$
$\mathcal{L}_{ij}^{neg}$
the sum of the Skipgram Negative Sampling Loss
$\vec{w}_i$
target word vector
$\vec{c}_j$
a content vector
1
the unnormalized document vector
2
$\vec{d}_j$
$\vec{c}_j = \vec{w}_j + \vec{d}_j$
$\vec{w}_j$
pivot word vector
the second topic vector
$\vec{t}_1$
topic vectors
$\vec{t}_k$
3
$\vec{d}_j = p_{j0} \cdot \vec{t}_0 + p_{j1} \cdot \vec{t}_1 + \cdots + p_{jk} \cdot \vec{t}_k + \cdots + p_{jn} \cdot \vec{t}_n$
weight
$0 \le p_{jk} \le 1$
document weights
p_{jk}
$\sum_k p_{jk} = 1$
k
topic
j
document
$n = 20$
$k = 0 \ldots 19$
$j = 0 \ldots 11312$
a low concentration parameter
α
4
document proportion
p_{j0}
document proportion
p_{j1}
pairs of pivot and target words
p_{ij}
$\alpha > 1$
$\alpha = 1$
$\alpha < 1$
$\alpha = n^{-1}$
the number of topics
n
$\mathcal{L}^d = \lambda \sum_{jk} (\alpha - 1) \log p_{jk}$
$n = 15$
λ
the tuning parameter
$\lambda = 200$
the tuning parameter

Figure 35: A conceptualized structure of reasoning graph for arXiv document 1605.02019 (red dashed lines denote the blocks and flows).

Segmentation of Reasoning Blocks

RBs is a set of blocks segmented from the original content that contains a set of consecutive sentences with high-density MOs distributed among them. We observed a key feature in the local density of MOs that can be used to determine the start and end of the RB boundaries. Specifically, the density of MOs around main equations often goes high and then progressively drops to a low density of MO when approaching to the end of discussion. The boundaries of RB are likely located in those MO-sparse regions, as shown in Figure 36.

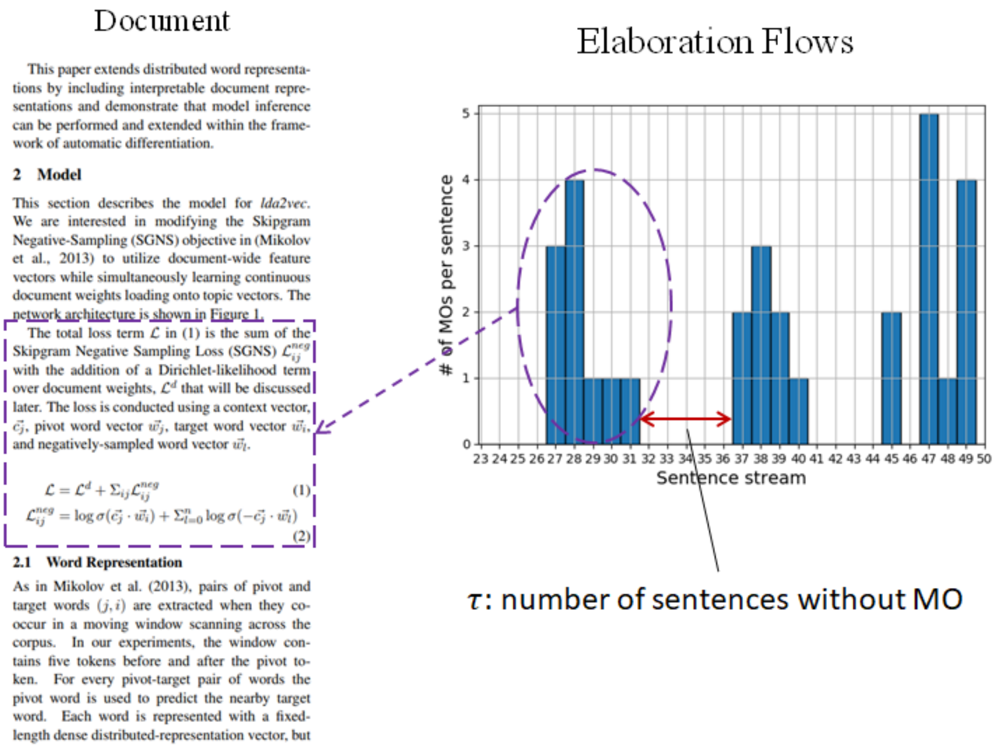
The search of the RB boundary will be based on the continuity of the neighboring related MOs and frequency of MO within a sliding window. We define a parameter τ as the number of sentences without MOs allowed in a block. The value τ is considered as a threshold to define the boundaries of RBs as shown in Figure 36. The pseudo code for the proposed RB segmentation algorithm is described as follows:

(1) Given a set of sentences $S = \{s_1, s_2, \ldots, s_n\}$ in a document and an integer value τ as the threshold of the RB boundaries. We label each sentence $s_i \in S$ to 0 (if it is a sentence without MO) or 1 (if it is a sentence with MO) to obtain a binary string $\{b_1, b_2, \ldots, b_n\}$ where $b_i \in \{0,1\}$.

(2) Iteratively search forward and backward to construct an RB $\{s_i, \ldots, s_j\}$ for $1 \leq i \leq j \leq n$ such that $\langle b_{i-1-\tau}, \ldots, b_{i-1} \rangle = \vec{0}$ and $\langle b_{j+1}, \ldots, b_{j+1+\tau} \rangle = \vec{0}$.

We tested the algorithm on arXiv document 0904.0684 and observed that the higher the value τ is, the less RBs generated as shown in Figure 37(a). We examine the number of MO in the first RB and found out that sudden increase appeared when τ exceeds certain value as shown in Figure 37(b). There is no best value for τ. It is subject to change depending on how MOs are distributed in the document. That is, one might want to consider a lower (higher) τ if the MOs are densely (sparsely) distributed in the content to obtain meaningful segmentations.

Note that linguistic constraints such as the boundaries of paragraph can also be introduced in the segmentation process to obtain better segment points. However, we discovered that there is no general solution in the literature for paragraph extraction because the performance is highly dependent on the style of document. We conducted some preliminary studies and found out that a lot of paragraphs have been detected in fragments due to the use of relative spacing introducing ambiguity for extraction (see Figure 54 of Appendix C). Therefore, for simplicity, we assume our algorithm in this dissertation does not have the paragraph information for alignment with the τ-based boundaries.



This paper extends distributed word representations by including interpretable document representations and demonstrate that model inference can be performed and extended within the framework of automatic differentiation.

2 Model

This section describes the model for *lda2vec*. We are interested in modifying the Skipgram Negative-Sampling (SGNS) objective in (Mikolov et al., 2013) to utilize document-wide feature vectors while simultaneously learning continuous document weights loading onto topic vectors. The network architecture is shown in Figure 1.

The total loss term $\mathcal{L}$ in (1) is the sum of the Skipgram Negative Sampling Loss (SGNS) $\mathcal{L}_{ij}^{neg}$ with the addition of a Dirichlet-likelihood term over document weights, $\mathcal{L}^d$ that will be discussed later. The loss is conducted using a context vector, $\vec{c_j}$, pivot word vector $\vec{w_j}$, target word vector $\vec{w_i}$, and negatively-sampled word vector $\vec{w_l}$.

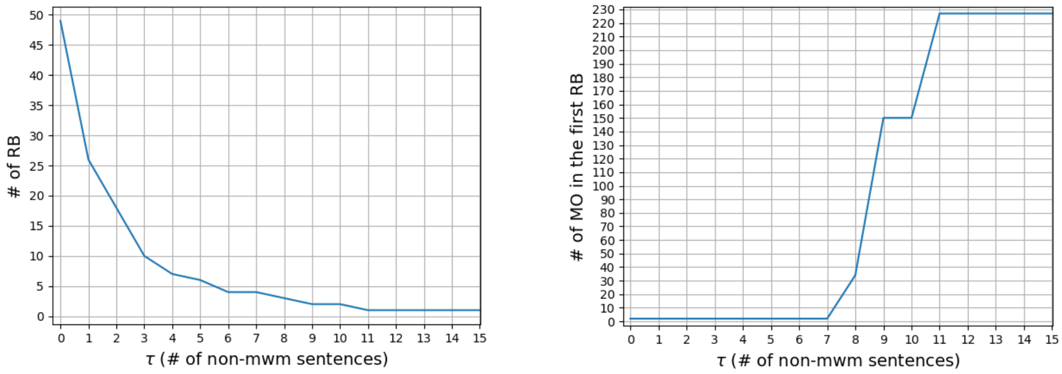
$$\mathcal{L} = \mathcal{L}^d + \Sigma_{ij}\mathcal{L}_{ij}^{neg} \quad (1)$$

$$\mathcal{L}_{ij}^{neg} = \log\sigma(\vec{c_j}\cdot\vec{w_i}) + \Sigma_{l=0}^{n}\log\sigma(-\vec{c_j}\cdot\vec{w_l}) \quad (2)$$

2.1 Word Representation

As in Mikolov et al. (2013), pairs of pivot and target words (j, i) are extracted when they cooccur in a moving window scanning across the corpus. In our experiments, the window contains five tokens before and after the pivot token. For every pivot-target pair of words the pivot word is used to predict the nearby target word. Each word is represented with a fixed-length dense distributed-representation vector, but

Figure 36: The boundary of reasoning blocks (purple dashed line) with a threshold of allowing four consecutive sentences without MO.



(a) (b)

Figure 37: The threshold τ that impacts (a) the number of reasoning blocks created; and (b) the number of mathematical objects in the first reasoning block.

Dependency Analysis of Mathematical Objects

MO dependency is an MO-to-MO relation where one MO is denoted by the other MO. The existence of such relation is defined upon the common identifier set of MOs (i.e., any named entity in MO such as variable and function name) in which a directed edge $\langle x, y\rangle$ $(x \rightarrow y)$ is defined as MO x (or $LHS(x)$) is an identifier set of MO y. However, a challenge for this task lies in the implicit multiplication operation that causes the ambiguity of consecutive multiple identifiers (CMI), which has been summarized in [90] as one of the MO semantic problems. For example, given an MO "Σ_{ij}" as shown in Figure 38, the subscript "ij" is a single or separated identifier(s). The CMI problem requires an assessment function to detect the collocation [91] between characters in the identifier. Any two character are considered collocated if they always appear together side by side in the adjacent sense. Previous work in MO dependency analysis [56] did not consider this issue in their graph construction, which results in a final graph with a lot of redundant edges.

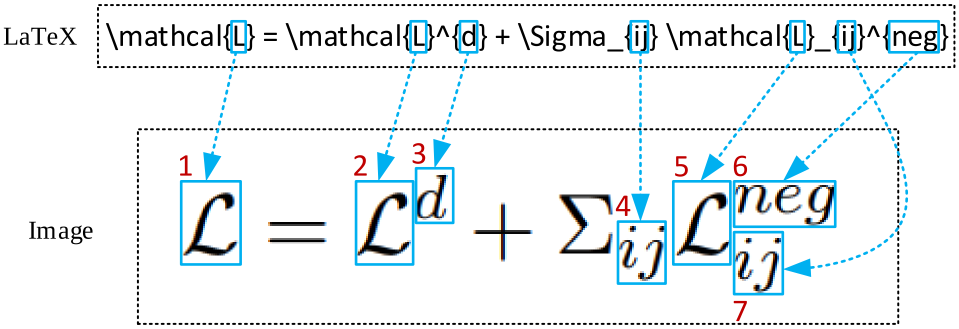

LaTeX
\mathcal{L} = \mathcal{L}^{d} + \Sigma_{ij} \mathcal{L}_{ij}^{neg}
Image
1
2
3
4
5
6
7

Figure 38: The identifier set of a mathematical object.

To address the challenge of recognizing the CMI, we propose to first decompose the MO into a set of identifier strings using a universal list of operators (see Table 10 of Appendix D) as delimiters. Some commonly used mathematical terms (see Table 11 of Appendix E) are excluded from the set of strings. The assessment functions are then applied to each n–gram identifier string by iteratively assess its i-gram and $(n - i)$-gram substrings to obtain the optimum splitting point i. The process will recursively assess on the two partitions split by i until no more new partitions are created.

Pointwise mutual information (PMI) [92] is a popular measurement of association used in the natural language processing (NLP) community to determine the independence of any two words [93]. It is very effective in finding the collocation [94] between words. We proposed to use the PMI as one assessment function for the CMI problem, which is defined as $PMI(x; y) = \log\frac{Pr(x,y)}{Pr(x)Pr(y)}$ where $Pr(s')$ is the occurrence rate of substring s' and the PMI measurement $-\infty \leq PMI(x; y) \leq min(-\log Pr(x), -\log Pr(y))$ for the two substrings resulting in negative if they never occurred together, 0 if they are independence, and positive if they are somewhat co-occurred. Given an identifier string s, for any two substrings $s[0\colon i]$ and $s[i+1\colon n]$, we calculate the optimum splitting point PMI^{*} with the maximum PMI score as follows:

$$PMI^{*}(s,i) = \operatorname*{argmax}_{i} \log \frac{Pr(s[0\colon i], s[i+1\colon n])}{Pr(s[0\colon i])Pr(s[i+1\colon n])}$$

Besides using the PMI, we also proposed another assessment function for the multiple character identifier (MCI) based on the frequency of substring. Given any n-character identifier string s, the MCI is defined as $MCI(x;y) = \frac{2f(x,y)}{f(x)f(y)}$ where $f(s')$ is the frequency of substring s' and the measurement result is normalized as $0 \leq MCI(x;y) \leq 1$ for 0 representing the two substring are not collocated, and 1 representing they are collocated. The optimum splitting point MCI^{*} with the maximum MCI score is then calculated as follows:

$$MCI^{*}(s,i) = \operatorname*{argmax}_{i} \frac{2f(s)}{f(s[0\colon i]) + f(s[i+1\colon n])}$$

The final decision will be based upon a threshold δ such that s is a consecutive multiple identifier if and only if $MCI^{*}(s,i) < \delta$.

As a result, we conducted a preliminary study on arXiv document 1605.02019 to examine the performance of the two assessment functions PMI and MCI, respectively, for recognizing the consecutive multiple identifier. The occurrence of each identifier string extracted from the MOs are shown in Figure 39, and the experiment results of the assessments are listed in Table 7. Both PMI and MCI can successfully identify words like “*king*”, “*queen*”, “*woman*”, “*man*” as inseparable sets. However, the MCI failed to identify “neg” as a MCI and the PMI failed to identify “jk” and “ij” as two CMIs.

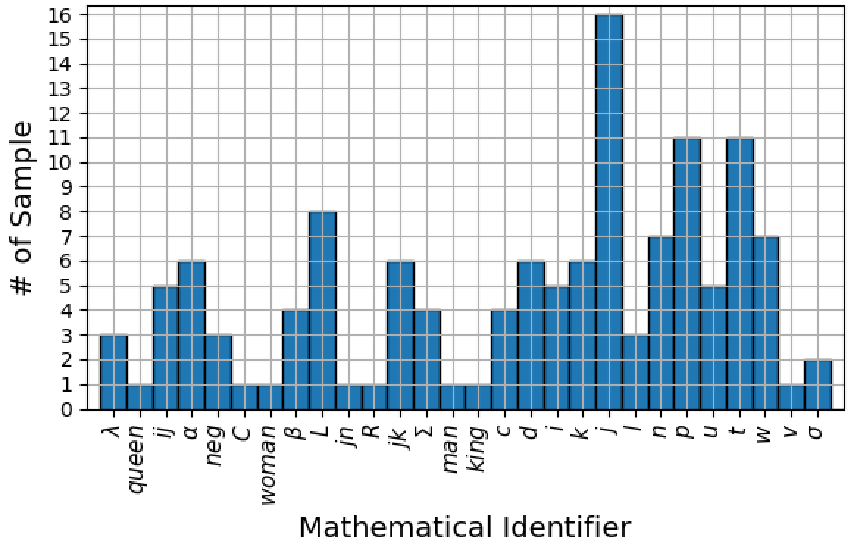
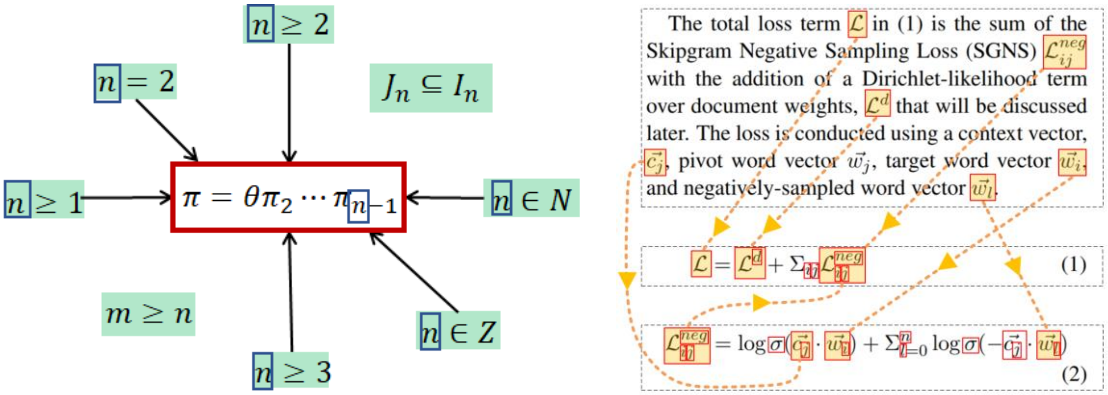


Figure 39: The occurrences of identifier sets in arXiv document 1605.02019.

Table 7: The assessment score of identifying the consecutive multiple identifiers.

Mathematical Identifier Set	MCI	PMI
$king, queen$	1.000	+4.868
$woman$		+4.174
λ		+3.769
β		+3.481
α, σ, Σ		+3.076
man		+2.470
neg	0.462	+2.565
jk	0.293	+0.762
ij	0.222	+0.312
jn	0.047	-1.173

The rest of the process in dependency analysis will be the partial matching of identifier set between the MO pair in building the dependency links. Notice that some MOs can further decompose their components into left-hand side (LHS) and right-hand side (RHS) by a pre-compiled list of relational symbols (see Table 10 of Appendix D). The source MO with a LHS should only consider the identifier set in its LHS when doing the subset matching of identifiers with any target MO as shown in Figure 40(a). The direction of the MO dependency is decided based upon the direction of the matches hit. The direction of an MO node x pointed to an MO node y imply the semantics that x or its LHS is a subcomponent of y. A handcrafted example of connecting MOs based on their internal identifiers is shown in Figure 40(b).



The total loss term $\mathcal{L}$ in (1) is the sum of the Skipgram Negative Sampling Loss (SGNS) $\mathcal{L}^{neg}_{ij}$ with the addition of a Dirichlet-likelihood term over document weights, $\mathcal{L}^d$ that will be discussed later. The loss is conducted using a context vector, $\vec{c_j}$, pivot word vector $\vec{w_j}$, target word vector $\vec{w_i}$, and negatively-sampled word vector $\vec{w_l}$.

$$\mathcal{L} = \mathcal{L}^d + \Sigma_{ij}\mathcal{L}^{neg}_{ij} \qquad (1)$$

$$\mathcal{L}^{neg}_{ij} = \log\sigma(\vec{c_j} \cdot \vec{w_i}) + \Sigma^{n}_{l=0}\log\sigma(-\vec{c_j} \cdot \vec{w_l}) \qquad (2)$$

(a) (b)

Figure 40: The two types of dependencies link based on (a) the left-hand side of MO or (b) the whole MO, is a subcomponent of the MO pointed.

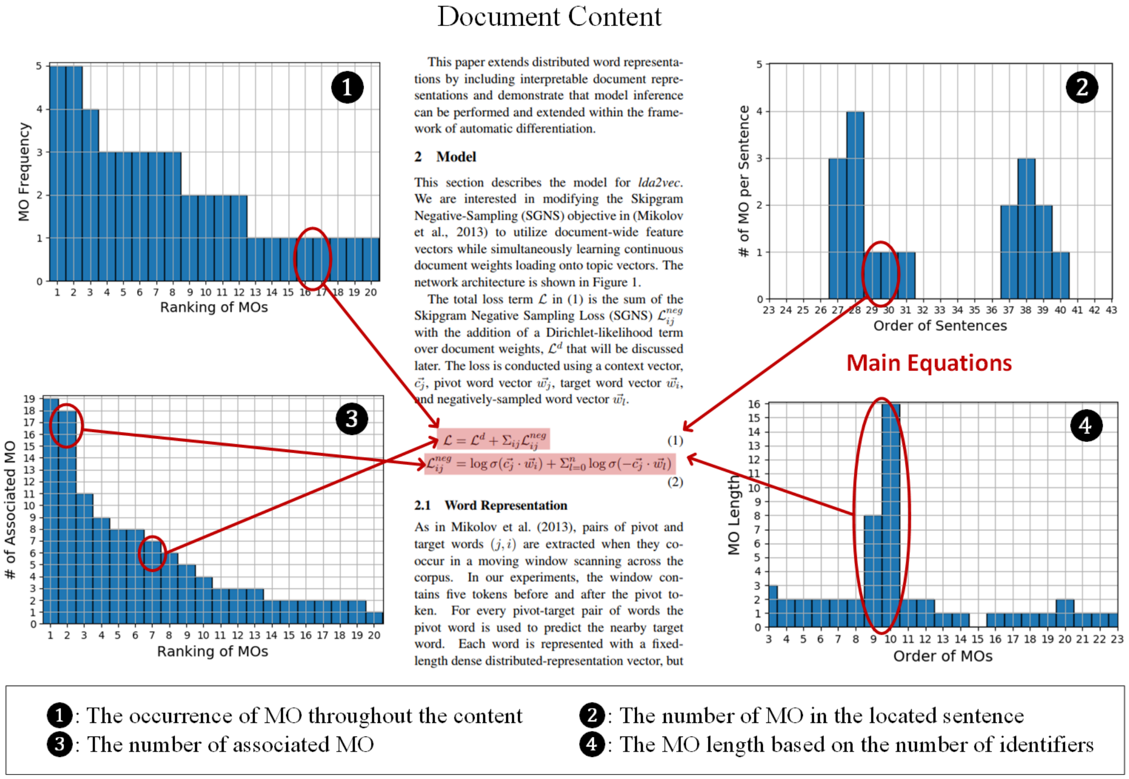
After the interrelation analysis of MOs, the formation of reasoning flows can be recovered based on the dependency of MOs between RBs. The in-link and out-link will decide the direction from one RB to another. No matter how many in-links or out-links are there in an MO, the reasoning flows are constructed based on any dependency link that existed in an MO pair, and the directions are aligned with the dependency. The main reason why the reasoning flow follows the orientation of the dependency is that technical writing tends to first describe the problem formulation and then go into details of problem settings, whereas dependencies were built in the opposite direction of the writing flows from details to abstraction. Based on the original elaboration flow of the paper, the RBs are constructed following the sequence of sentences in the content. However, MO dependencies are non-linearly crossed over the content in a paper, suggesting that the reasoning flows do not necessarily align with the elaboration flows, and hence MOR graph provides the potentials to be applied in document fingerprint.

Reduction of the MOR Graph

The size of the original content has been greatly reduced in MOR graph with an average of 57.88% from an experiment on 210 arXiv documents. However, the MO dependency relationships are still too complex for human to understand. Although the QuQn map [33] has attempted to simplify the dependency structure by removing duplicate nodes and short-cut links in the pruning process, the resulting graph still suffer from spaghetti-like structure when processing papers with dense MOs (see Figure 34). A missing opportunity is to consider the MO-to-MO and MO-to-words relations in the reduction process. To simplify the already complex dependency relationships between MOs in the skeleton of MOR graph, we apply some heuristics on the edge and node pruning based on certain reasoning.

For the node pruning, we merge the duplicate nodes (i.e., same MOs), and eliminate the rooted nodes (i.e., MOs with only out-links) that are neither a constraint nor a declaration. Only the main equations and/or their relevant constraints and declarations will remain in the RB. For the edge pruning, the dependencies between any MOCs are eliminated as a result in a spring graph layout like Figure 40(a) to highlight the subordinate relation of the main equations to their constraints. Note that the MO "$\pi = \theta\pi_2 \cdots \pi_{n-1}$" in Figure 40(a) has various formations of constraints or conditions associated with it, making n an important local variable that often redefined in the contents for different usages. The global variables, however, are expected to be defined consistently in any discussion of the whole document.

After some basic pruning process on the skeleton of MOR graph, we investigate certain filtering criteria to highlight main equations along with or without their constraints and declarations as the building blocks. The main equation here is defined as the MO with equal sign that covers most of the associated MOs in the local content. Some semi-automatic experiments have been conducted to study the properties of main equations as shown in Figure 41. We observed from the preliminary results that main equations: (1) are one-time definitions of their explicit forms and will not be repeatedly mentioned in the content; (2) are often appeared as the displayed mode (i.e., in single line space and/or denoted by equation number); (3) are associated with various MOs based on the common identifiers; (4) are relatively large and contains a lot of identifiers. We further examined the above four properties of all equations on 50 documents randomly sampled from the KDD-2003 dataset [38]. The results of all four histograms present a long tail distribution as shown in Figure 55 and Figure 56 of Appendix F, suggesting that most equations are (1) mostly mentioned once in the content; (2) the only MO in a sentence; (3) associated with a few MOs in the content; (4) of shorter length in their presentations. The contradiction of the third and fourth criteria to the assumption of main equations are due to the existence of short equations that are used for problem settings in the content. For example, the problem setting $c = \sim 3 \times 10^8\ m/s$ (speed of light) for the mass-energy equivalence formula $E = mc^2$.



This paper extends distributed word representations by including interpretable document representations and demonstrate that model inference can be performed and extended within the framework of automatic differentiation.

2 **Model**

This section describes the model for *lda2vec*. We are interested in modifying the Skipgram Negative-Sampling (SGNS) objective in (Mikolov et al., 2013) to utilize document-wide feature vectors while simultaneously learning continuous document weights loading onto topic vectors. The network architecture is shown in Figure 1.

The total loss term $\mathcal{L}$ in (1) is the sum of the Skipgram Negative Sampling Loss (SGNS) $\mathcal{L}_{ij}^{neg}$ with the addition of a Dirichlet-likelihood term over document weights, $\mathcal{L}^d$ that will be discussed later. The loss is conducted using a context vector, $\vec{c_j}$, pivot word vector $\vec{w_j}$, target word vector $\vec{w_i}$, and negatively-sampled word vector $\vec{w_l}$.

$$\mathcal{L} = \mathcal{L}^d + \Sigma_{ij}\mathcal{L}_{ij}^{neg} \quad (1)$$

$$\mathcal{L}_{ij}^{neg} = \log\sigma(\vec{c_j} \cdot \vec{w_i}) + \Sigma_{l=0}^{n} \log\sigma(-\vec{c_j} \cdot \vec{w_l}) \quad (2)$$

2.1 **Word Representation**

As in Mikolov et al. (2013), pairs of pivot and target words (j, i) are extracted when they co-occur in a moving window scanning across the corpus. In our experiments, the window contains five tokens before and after the pivot token. For every pivot-target pair of words the pivot word is used to predict the nearby target word. Each word is represented with a fixed-length dense distributed-representation vector, but

Figure 41: Histograms of the four properties of main equation from arXiv document 1605.02019.

To define the importance of an equation, we first need to aggregate the results of the four criteria into one score. The weights can later be used to reveal or hide an MO node during the visualization process based on its importance with respect to other MO nodes. If we directly use the four criteria as thresholds to filter the MOs, a lot of possibilities will appear that results in a large-scale study. Hence, we proposed to use the z-score normalization to reduce the four measurements of the criteria down to one score. The four metrics are defined as $\theta \in \{freq,\ dens,\ degr,\ leng\}$ where (1) $freq$: the occurrence of MO throughout the whole document; (2) $dens$: the number of MO in the located sentence; (3) $degr$: the number of in-links in an MO; (4) $leng$: the number of mathematical identifiers in an MO. The z-score function of an MO x is defined as $m_\theta(x) = \frac{x-\mu}{\sigma}$ where x is the measurement of metric θ, μ is the mean of the population, and σ is the standard deviation. The final score for the MO x among these four metrics is then aggregated as $score(x) = -m_{freq}(x) - m_{dens}(x) + m_{degr}(x) + m_{leng}(x)$. We tested the scoring function on arXiv document 1605.02019, and the experiment results in Table 8 show that most of the short equations used for problem settings tend to have lower scores and long equations used for problem formulations tend to have higher score from the assessments.

Table 8: A primitive evaluation of equations rank on arXiv document 1605.02019.

Mathematical Equations	*score*	*freq* (−)	*dens* (−)	*degr* (+)	*leng* (+)
$\vec{d}_j = p_{j0} \cdot \vec{t}_0 + p_{j1} \cdot \vec{t}_1 + \cdots + p_{jk} \cdot \vec{t}_k + \cdots + p_{jn} \cdot \vec{t}_n$	8.98	1	1	8	16
$\mathcal{L}_{ij}^{neg} = \log \sigma\left(\vec{c}_j \cdot \vec{w}_i\right) + \sum_{l=0}^{n} \log \sigma\left(-\vec{c}_j \cdot \vec{w}_l\right)$	8.11	1	1	6	16
$\mathcal{L}^d = \lambda \sum_{jk} (\alpha - 1) \log p_{jk}$	7.57	1	1	11	8
$\mathcal{L} = \mathcal{L}^d + \sum_{ij} \mathcal{L}_{ij}^{neg}$	5.38	1	1	6	8
$\alpha = n^{-1}$	3.28	1	2	8	2
$\sum_k p_{jk} = 1$	3.15	1	1	4	4
$\vec{c}_j = \vec{w}_j + \vec{d}_j$	2.95	1	1	2	6
$king - man + woman = queen$	1.40	1	1	0	4
$\lambda = 200$	0.37	1	1	0	1
$\alpha = 1$	0.31	1	2	2	1
$\beta = 0.75$	-0.56	1	2	0	1
$k = 0 \ldots 19$	-1.06	1	3	1	1
$n = 15$	-1.06	1	3	1	1
$j = 0 \ldots 11312$	-1.06	1	3	1	1
$n = 20$	-1.18	2	2	1	1

Visualization of the MOR Graph

We have developed a prototype tool for visualizing the MOR graphs, which is able to support interactive browsing of the technical contents at different granularities of detail. The result of the visualization is tested on the arXiv paper same as the one used for QuQn map in Figure 34. The presentation of MOR graph begins with a coarse level of abstractions, and the user can progressively expand the details if they are interested in certain technical discussions. The first layer of the MOR graph is the RBs in the document. Each RB can expand its main equations and the associated problem settings can be further expanded from those main equations. The layout design follows a multiple level star graph to manifest the subordinating relationships among different levels of technical details. Compared with the QuQn map visualization in Figure 34, the presentation of the MOR graph in Figure 42 has largely decreased the number of links and nodes revealed to the user. By the use of MOR graph, the user can learn the technical details via a progressive way.

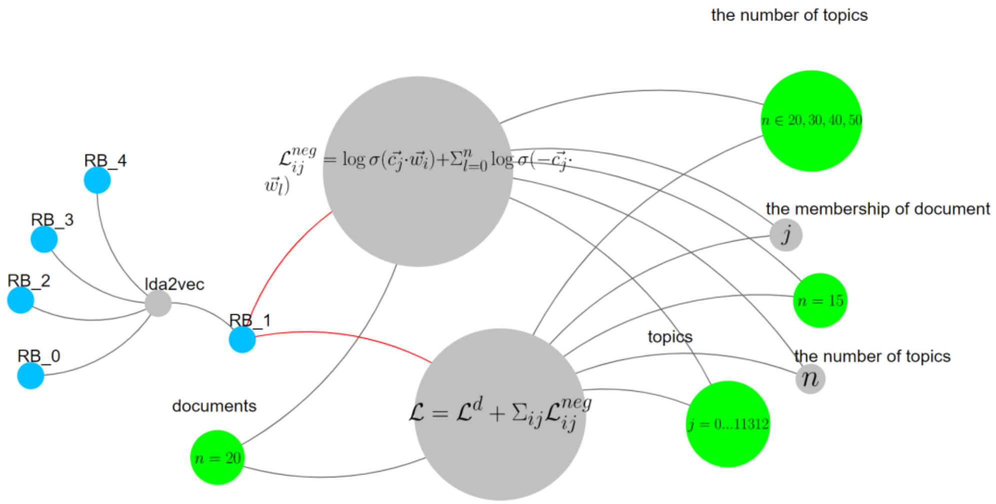

the number of topics
$\mathcal{L}_{ij}^{neg} = \log \sigma(\vec{c_j} \cdot \vec{w_i}) + \Sigma_{l=0}^{n} \log \sigma(-\vec{c_j} \cdot \vec{w_l})$
$n \in 20, 30, 40, 50$
RB_4
RB_3
RB_2
RB_0
lda2vec
RB_1
the membership of document
j
$n = 15$
topics
the number of topics
n
documents
$n = 20$
$\mathcal{L} = \mathcal{L}^d + \Sigma_{ij} \mathcal{L}_{ij}^{neg}$
$j = 0 \ldots 11312$

Figure 42: The visualization of reasoning graph for arXiv document 1605.02019.

Applications of the MOR Graph

The existing cross-paper analyses for mapping a knowledge domain are mostly based on co-occurrence analysis, including the co-author networking [95], co-citation clustering [96], and co-words modeling [97] using latent semantic analysis (LSA) [98] and latent Dirichlet allocation (LDA) [99] methods. Both the author-based and citation-based features are considered very coarse-level analysis which an author might work on various topics and the citations could play various roles such as background, usage, or comparison

[100]. The word-based analysis techniques using bag-of-words are relatively effective for indexing and building cross-links among papers. However, it is very difficult to apply them for deep content analysis of technical essence, most of which are carried by the MO-based information. Some MO-based works have used the properties of MO presentations [101], [102], [103], [104], [105], and MO citations [106], [107], for indexing specific formula in a paper. However, since an MO can carry multiple physical meanings in the same presentation [108], they failed to incorporate the MO semantics in their design of MIR systems. Recently, several research [54], [55], [56] have applied the mixed used of words and MOs for a better performance of MIR task, which is based on the fact that many MOs are bonded to words in scientific documents. Although the above existing works are very effective in MO indexing, they did not consider the importance of MO declarations (MODs) within a document with respect to a corpus of documents. Also, the reasoning structures of how these MOs are elaborated with respect to their localities (RBs) and constraints (MOCs) have not yet been considered in the study of cross-paper analysis.

The cross-paper analysis in this section is confined into three use cases: (1) a manually labeled case for the detail analysis of two papers based on the MOD and MOC through elaboration flows; (2) the unsupervised clustering of MO-based technical dialects ("jargoning", "terminology") across papers in various knowledge domains; (3) the structural-level differential analysis of the MOR graphs in a set of documents. For both aspects, the core research topic is to define the similarity metrics between papers, where each paper will be abstracted as an MOR graph that consists of RBs, MOs, and MO-dependency links. The goal is to utilize the MOR graph and its low-level analysis engines to offer new perspectives and foundational solutions to attack more sophisticated problems like plagiarism detection and writing style modeling, etc.

The Cross-paper Analysis Approach

To model the elaboration flow and enable the cross-paper similarity analysis of the MOR graphs, we propose to develop the similarity metrics that consists of (1) the LSA [98] model to reduce the high dimensionality of feature terms in science, technology, engineering and mathematics (STEM) documents and (2) the string edit distance [109] for assessing the structural differences of the graph abstractions. The analysis aims to answer some of the cross-paper analysis questions as listed below:

- "What are the primary model formulations of MOs in the clustered papers?"
- "Discovery of the common technical essences of papers in a field."
- "Scoring of the technical similarity/difference between two papers."

The first metric starts with constructing a term-document matrix by calculating the TF-IDF score [110], [111] of each term with respect to each document in the corpus. The frequency of each term (TF) extracted from the texts is re-weighted by its inverse document frequency (IDF) value. Given a document set D, the TF-IDF score of each term t extracted from the document $d \in D$ is calculated as follows:

$$tfidf(t, d, D) = tf(t, d) \cdot idf(t, D)$$

where $tf(t, d) = f_{t,d}$ and $idf(t, D) = \log \frac{|D|}{|d \in D : t \in d|}$. The semantics behind the TF-IDF model is that the most representative term for indexing a document is the term that frequently appeared in that document but rarely appeared in other documents in the related field. To avoid the noise from the non-informative words, the model will preprocess the texts using NLP tool to get rid of the stop words, and only consider NP as a term. The term-document matrix will reduce its dimensionality via singular value decomposition (SVD) technique to obtain a lower dimension of similarity matrix with each feature as a linear combination of terms.

The second metric uses the string edit distance to construct a similarity matrix that specify the differences of any two documents. Given any two strings s_1 and s_2, the string edit distance can be calculated as follows:

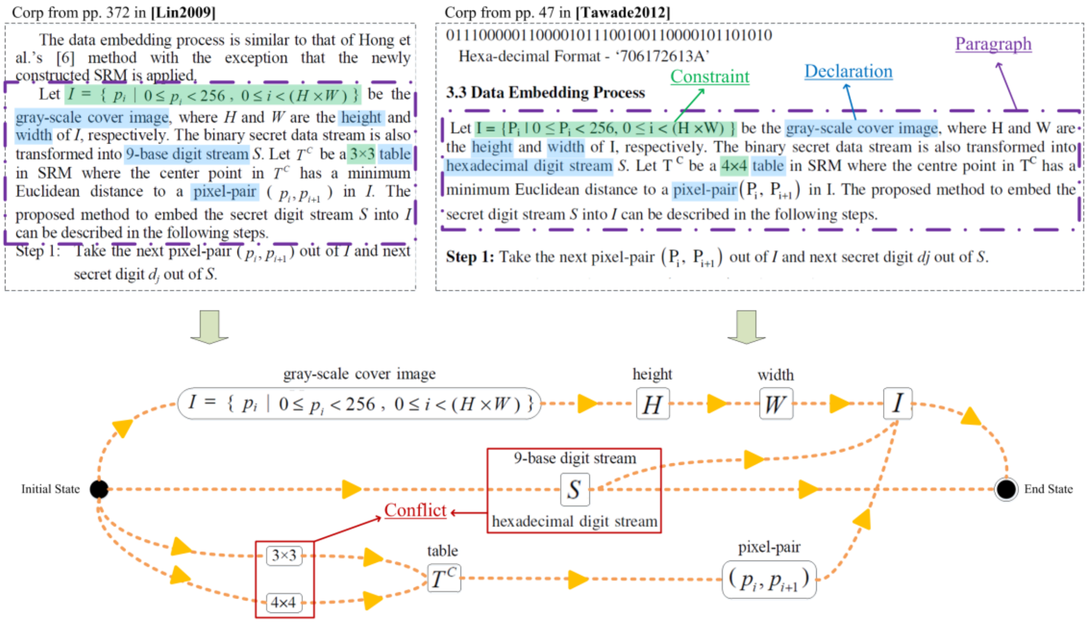
$$ED(s_1, s_2) = \min_{(a_1,\dots,a_n)} \sum_{i=1}^{n} c(a_i)$$

where $\min(|s_1|, |s_2|) \leq n \leq \max(|s_1|, |s_2|)$ and $c(a) \geq 0$ is the cost of each edit operations a (insertion, deletion, and substitution).

Use Case 1: Detail Analysis based on Technical Elements

In this use case, we conduct a simple experiment as shown in Figure 43 by manually annotating two paragraphs from two papers [35] and [112] based on their declarations and constraints to analyze their technical similarities. As a result, the two paragraphs have the same notations and elaboration structures, and the overall content of paper [112] seems to be an extension version of paper [35]. From the analysis of the declarations and constraints, we found two conflicts in the declarations and constraints of notation ‘S’ and ‘T^C’, respectively. However, the declarations of ‘S’: “9-based digit stream” and “hexadecimal digit stream”, refers to the same semantics. In fact, the only difference between the two paragraphs is the constraint of the table ‘T^C’, which is 3×3

and 4 × 4, respectively. The case study results suggest that the technical elements of the RB can be used for differential analysis of technical contents.


Corp from pp. 372 in **[Lin2009]**

The data embedding process is similar to that of Hong et al.'s [6] method with the exception that the newly constructed SRM is applied.

Let $I = \{ p_i \mid 0 \le p_i < 256 , 0 \le i < (H \times W) \}$ be the gray-scale cover image, where H and W are the height and width of I, respectively. The binary secret data stream is also transformed into 9-base digit stream S. Let T^C be a 3×3 table in SRM where the center point in T^C has a minimum Euclidean distance to a pixel-pair (p_i, p_{i+1}) in I. The proposed method to embed the secret digit stream S into I can be described in the following steps.

Step 1: Take the next pixel-pair (p_i, p_{i+1}) out of I and next secret digit d_j out of S.

Corp from pp. 47 in **[Tawade2012]**

011100000110000101110010011000010110101 0

Hexa-decimal Format - '706172613A'

3.3 Data Embedding Process

Let $I = \{P_i \mid 0 \le P_i < 256, 0 \le i < (H \times W) \}$ be the gray-scale cover image, where H and W are the height and width of I, respectively. The binary secret data stream is also transformed into hexadecimal digit stream S. Let T^C be a 4×4 table in SRM where the centre point in T^C has a minimum Euclidean distance to a pixel-pair (P_i, P_{i+1}) in I. The proposed method to embed the secret digit stream S into I can be described in the following steps.

Step 1: Take the next pixel-pair (P_i, P_{i+1}) out of I and next secret digit dj out of S.

Figure 43: Case study on cross paper analysis for one of the paragraphs in paper [35] and [112].

Use Case 2: Text-based Document Clustering

The document clustering is conduct using the MODs and NPs from the MWM sentences and the non-MWM sentences, respectively. The terms appeared in the MWM sentences and/or non-MWM sentences imply their interactions with MOs in the technical discourse. We construct a dataset from arXiv.org with 210 documents uniformly distributed in 7 research fields (30 documents per field). The research fields we selected include Computation and Language (F1), Graph Theory (F2), High Energy Physics (F3), Machine Learning (F4), Quantum Cryptography (F5), Steganography (F6), and

Theoretical Economics (F7). The extracted word-based terms are restricted to three-gram terms (unigram, bigram, and trigram). Through the process of LSA, we fit the resulting feature matrix into k-mean clustering [113] with k set to 7. The clustering results are shown in Figure 44 and Figure 45 for texts in MWM sentences and non-MWM sentences, respectively. The top 6 human understandable terms based on the TF-IDF scores are displayed to represent each cluster. A bipartite graph is generated to map the clusters to the knowledge domains and visualized it using the manifold techniques.

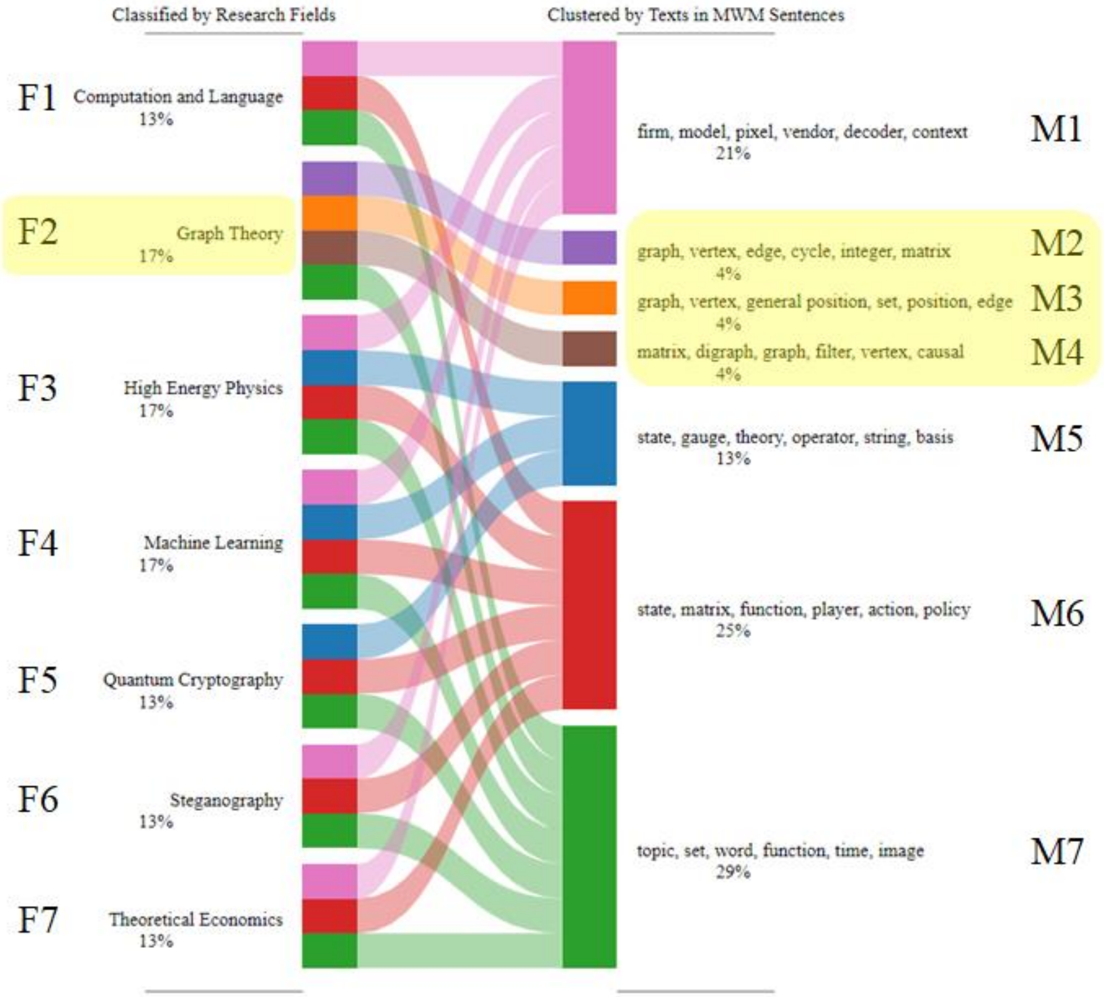


Figure 44: The bipartite graph for arXiv research fields and mathematical text clustering.

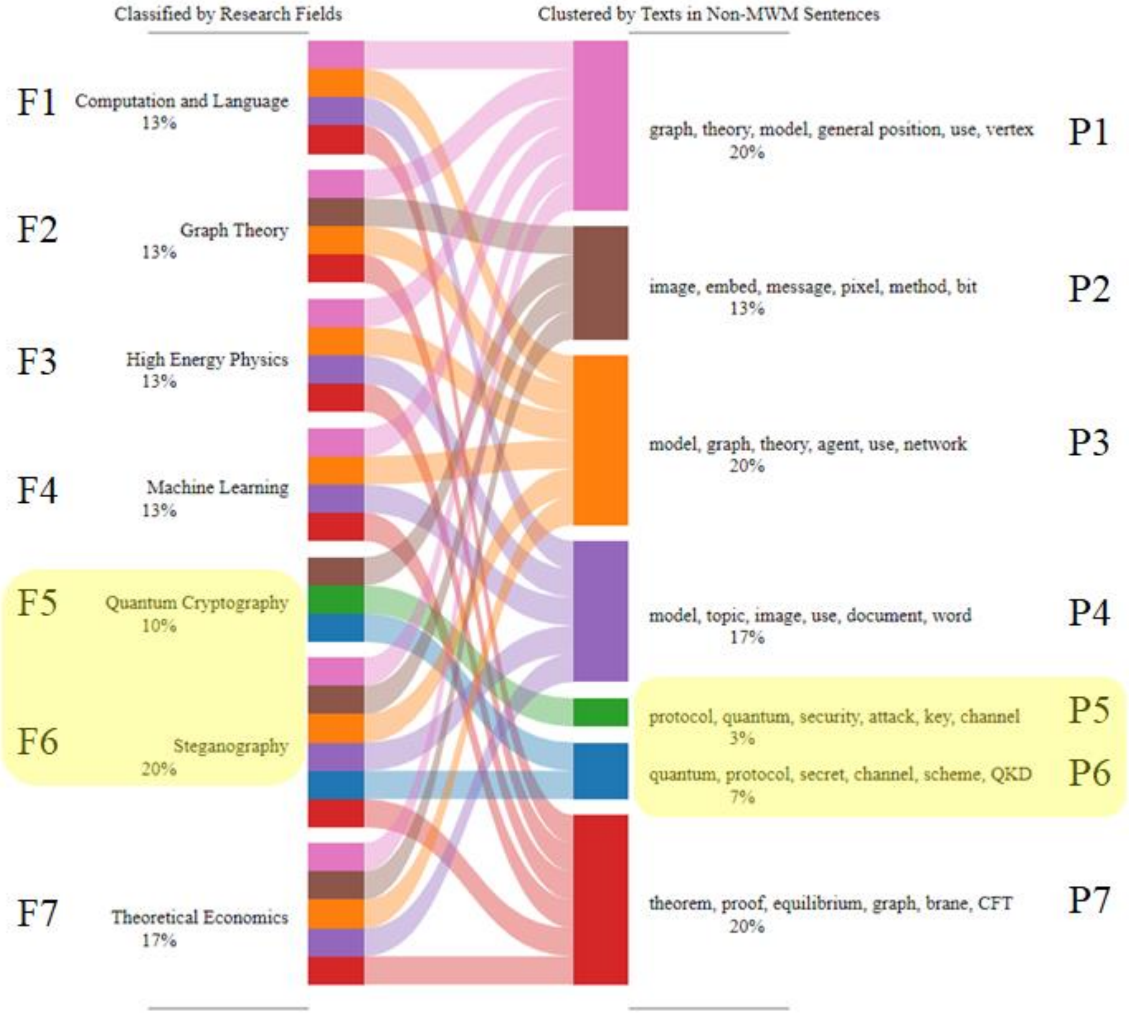


Figure 45: The bipartite graph for arXiv research fields and pure text clustering.

The results based on MOD has 24 links in the field mapping, which is lower than using non-MWM texts with 30 links. Among these links, three of the seven clusters (M1, M6, and M7) based on MOD have overlapped with more than four fields, which is fewer than the clusters using non-MWM texts that have four of the seven clusters overlapped with more than four fields. This indicates that MOD terms has a better performance in distinguishing different fields of research. The qualitative words tend to have higher overlapping in STEM field than quantitative words due to the ambiguity in their semantics. Some research fields like F5 (Quantum Cryptography) and F6 (Steganography) are sufficient to use non-technical terms such as “protocol”, “security”, “attack”, and “key” (from the clusters P5, P6) as shown in Figure 45. This is because most of the research from Quantum Cryptography tend to be protocol designs which are in step-by-step based descriptions like pseudo code with very few MOs defined in the content. However, research fields like F2 (Graph Theory) require the terms to be highly associated with MOs to locate the relevant research. Terms like “vertex”, “matrix”, and “edge” (from the clusters M2, M3, and M4) in Figure 44 are often used to formulate a graph in MOs, and “cycle” and “general position” are classic research problems in graph theory.

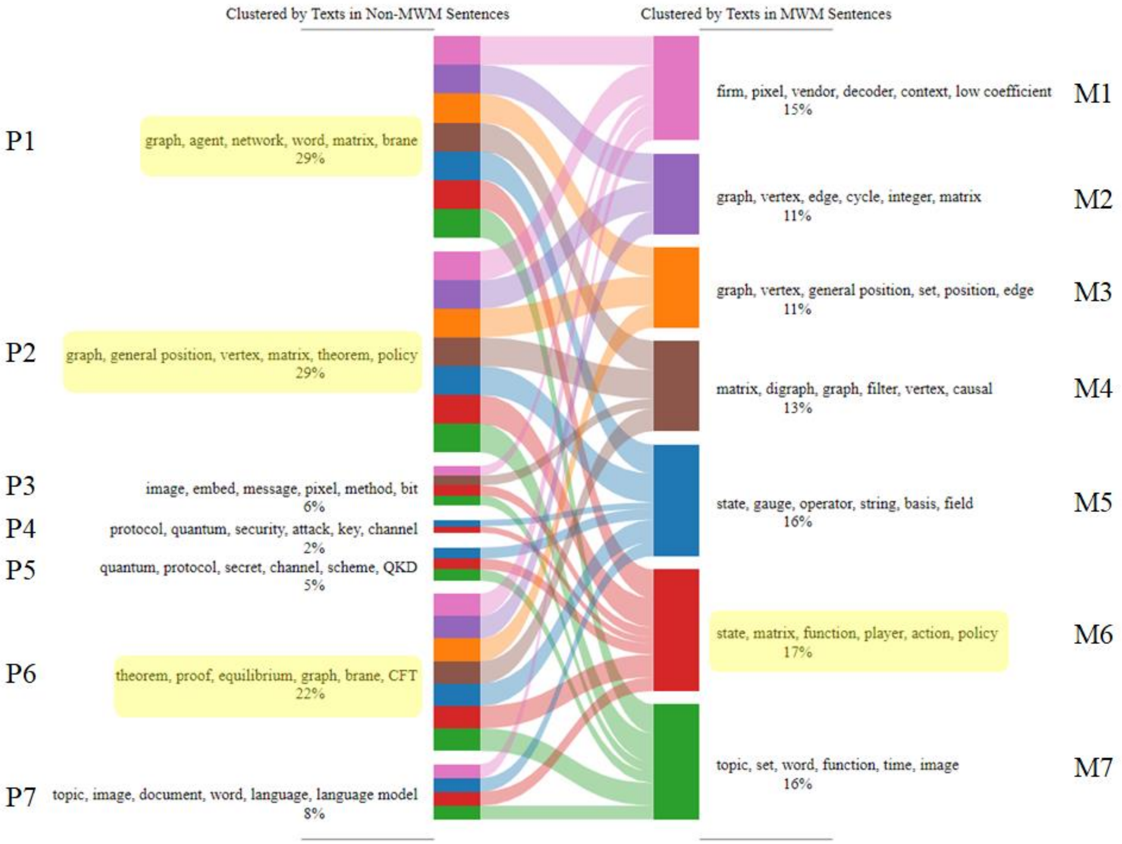


Figure 46: The bipartite graph for pure text clustering and mathematical text clustering.

The mutual relation based on the clustering result of MWM and non-MWM texts are also studied. As the result shown in Figure 44, we discovered that three of the clusters (P1, P2 and P6) at the left-hand side based on the NPs in non-MWM texts are terms that are used in any document cluster at the right-hand side separated by MOD in MWM texts. Most of the clusters at the right-hand side except M6 tend to single out at least one or more clusters at the left-hand side, suggesting that text based on MOD tend to have more distinguishing power to draw the boundaries of the documents clustered by terms that are not closely interact with MO. On the contrary, these terms that are extracted from the non-MWM texts tend to have a hard time locating documents that are clustered by MODs. This finding shows preliminary evidence that terms used in MWM texts for quantitative reasoning tend to have potential strength in delineating documents based on terms that have no interactions with MO.

Use Case 3: Structure-level Differential Analysis

To compare the structural differences between any two MOR graphs, a transformation process is required to convert the 2-dimensional graph into a comparable 1-dimensional sequence. In this case study, we investigate two encoding strategies, the Prüfer sequence [114] and the MOC-based degree sequence, respectively, to normalize the structure of the MOR graph and its underlying skeleton.

For the first strategy, the RBs of the MOR graph are taken as nodes, and the flows between RBs are taken as edges of the input graph. However, the Prüfer sequence required the input structure to be an acyclic labeled tree for transformation. Fortunately, RBs are labeled based upon the ordering of the original content, so the only issue we need to consider is the cycle that exists in the MOR graph. To address the problem, we adopt the breadth-first search (BFS) on the MOR graph to obtain its maximum spanning tree. The process of converting an MOR graph into a unique Prüfer sequence is described in Figure 47. The encoding of the sequence is based upon the order of label to iteratively search for the root node with minimal label to remove and encode the sequence with the label of the adjacent nodes. A Prüfer code has a length of $n-2$ for any n nodes labeled tree. After the encoding of Prüfer sequence, we calculated the string edit distance between the two compared sequences to obtain the similarity between any two MOR graphs.

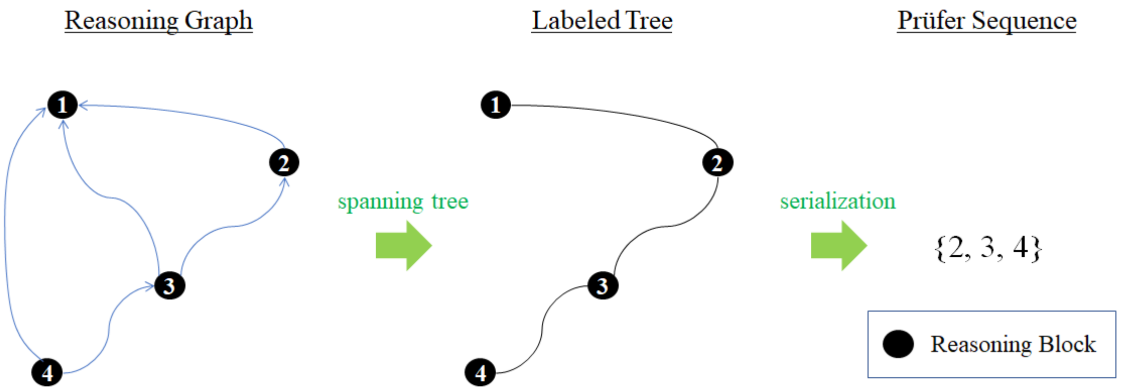


Figure 47: The process of serializing the MOR graph into Prüfer sequence based on its spanning tree.

Similar to the first strategy, we apply another encoding mechanism, the MOC-based degree sequence, on the MO dependency of the MOR graph. The MO is labeled based on its first appearance in an RB, so we can follow the order of the labels to calculate the number of MOCs each MO is associated with as the MOC-based degree. The process is depicted in Figure 48, and the resulting sequence is used as the structural information of the MO dependency graph for similarity analysis using the string edit distance.

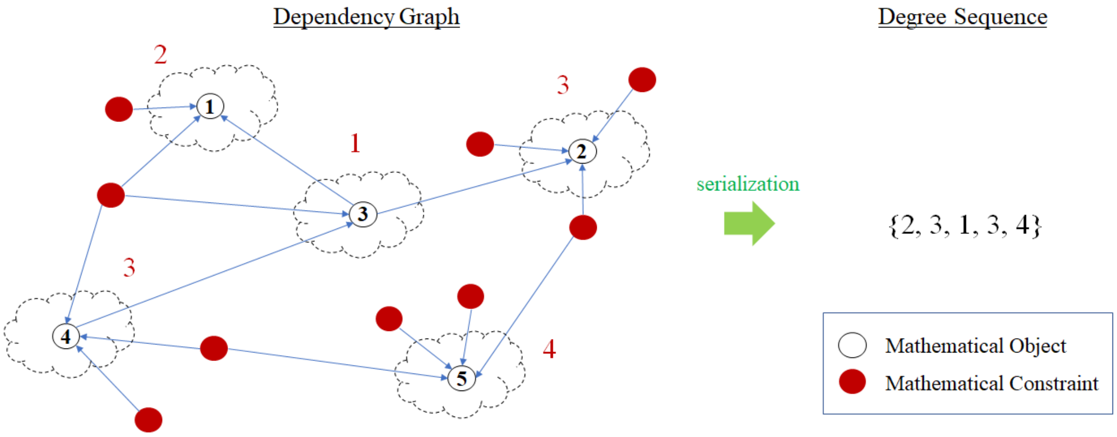
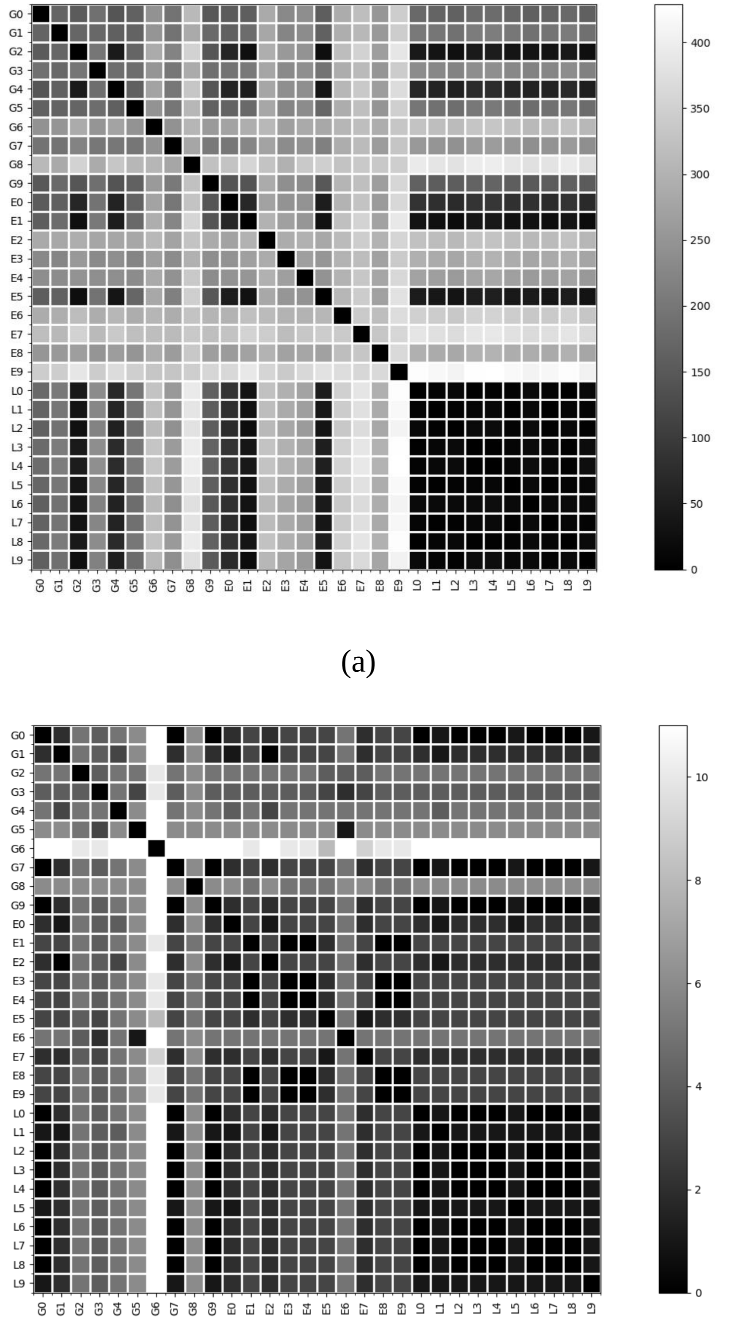


Figure 48: The process of serializing the dependency graph of mathematical objects into degree sequence based on the number of associated mathematical constraints.

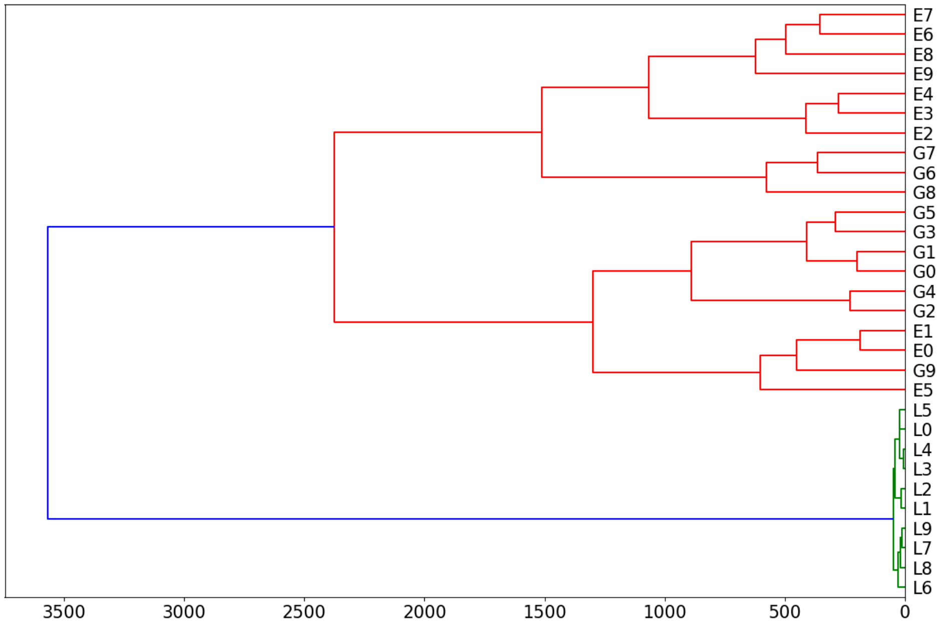
An experiment has been carried on sorting and classifying the structures of the MOR graph based on their differences in the 1-dimnetional encoder. We tested on the pilot dataset with 30 arXiv documents from three research fields: Computation and Language (L), Graph Theory (G) and Theoretical Economics (E). The assumption of this experiment design is that different field experts follow certain common practices in their domains to formulate the research problems and deliver their solutions. We expected our proposed MOR graph model to capture the essence of how they construct the technical contents based on their domain of writing style. We transformed each document into an MOR graph and then calculated its Prüfer sequence and MOC-based degree sequence to construct the similarity matrix between document pair. The results of similarity matrices for the two types of sequences on the arXiv pilot dataset are shown in Figure 49. From the similarity matrix, we found that the documents in the research field L has no significant differences between documents. We manually examine some of these documents and found out that most of their formulation has less dependencies and no constraints annotated to the MO node. To visualize the result, we applied an unsupervised hierarchical clustering on the similarity matrix using the Ward algorithm [115] to hierarchically group the instances based on their structural similarities. The final hierarchical clustering results are as shown in Figure 50. We observed that the documents are roughly group into three knowledge domains that nearly consistent to the three research fields we labeled in our pilot dataset. The results show that the clustering based on the structures of the MOR graph has a perfect clustering in the two research fields G and L, with a small miss rate of 16.67% in field E documents. On the other hand, the clustering based on the dependency structure of MOC has shown that the two research fields G and E both have 30% miss rate in their clusters. The results suggest that the structure of the MOR graph carry more significant insights to cluster the documents based on research fields. However, both approaches have demonstrated a first-hand evidence that both MOC and RB carry some useful insights that can be considered as a type of document fingerprints to differentiate scientific documents in different knowledge domains.



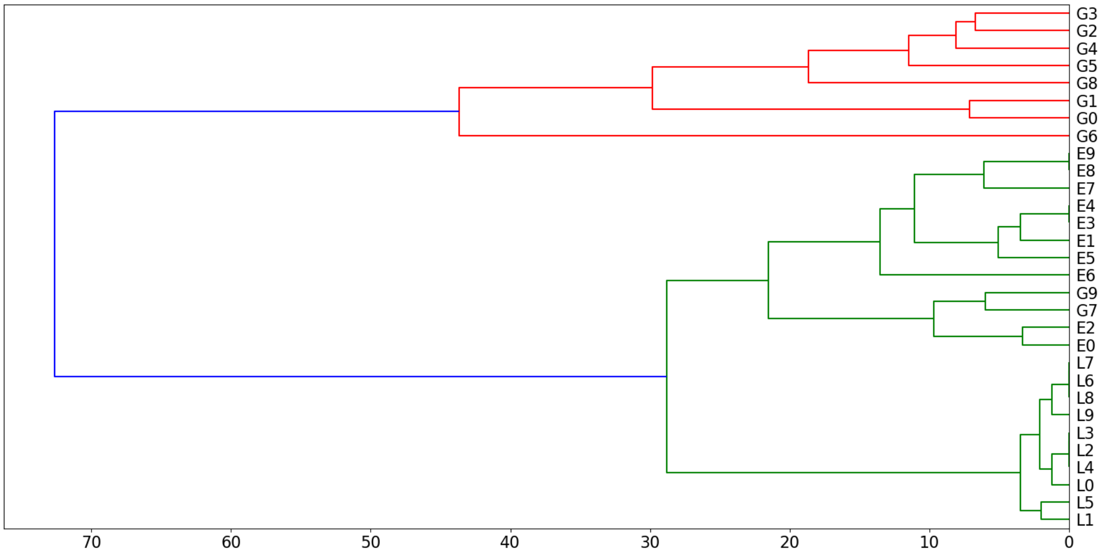
(a)

(b)

Figure 49: Similarity matrices of documents in three research fields ('L', 'G', 'E') based on the structural differences of (a) dependency graph and (b) reasoning graph of mathematical objects.



(a)



(b)

Figure 50: Dendrograms of document clustering results in three research fields (‘L’, ‘G’, ‘E’) based on the structural differences of (a) dependency graph and (b) reasoning graph of mathematical objects.

On the other hand, we applied the document clustering process as in Case 2 on this small dataset to examine the performance of using the word-based terms from the MWM sentences and the non-MWM sentences, respectively, instead of the reasoning structure of the contents. The bipartite relations between the three research fields (L, G, E) of the dataset and the three clusters using the MODs and NPs from the MWM sentences and the non-MWM sentences are shown in Figure 51 and Figure 52, respectively. The top 6 terms within trigram are used to represent each cluster. As a result, only the cluster M3 at the right-hand side in Figure 51 covers 17% of documents that can be precisely classified into the field G. The rest of the overlaps are either multiple-to-one or one-to-multiple relations for both clustering results in Figure 51 and Figure 52. From the clustering results, we found that the MO dependencies and the word-based terms is less distinguishable than the structural-level abstractions extracted based on human writing practice.

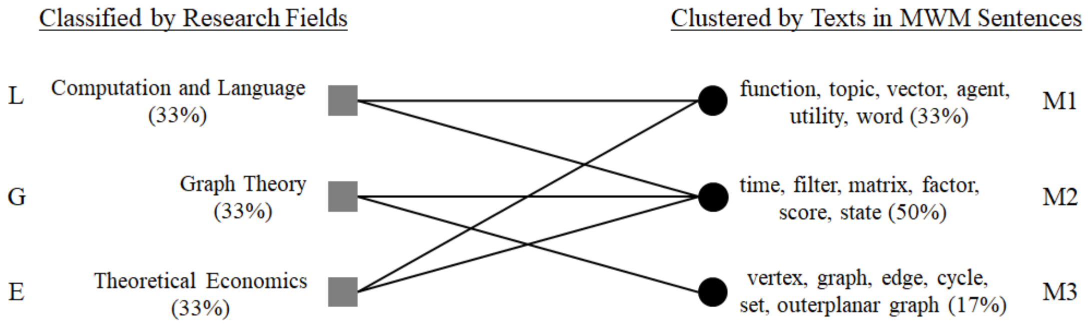

Classified by Research Fields
Clustered by Texts in MWM Sentences
L
Computation and Language (33%)
G
Graph Theory (33%)
E
Theoretical Economics (33%)
function, topic, vector, agent, utility, word (33%)
M1
time, filter, matrix, factor, score, state (50%)
M2
vertex, graph, edge, cycle, set, outerplanar graph (17%)
M3

Figure 51: The bipartite graph for research fields and mathematical text clustering.

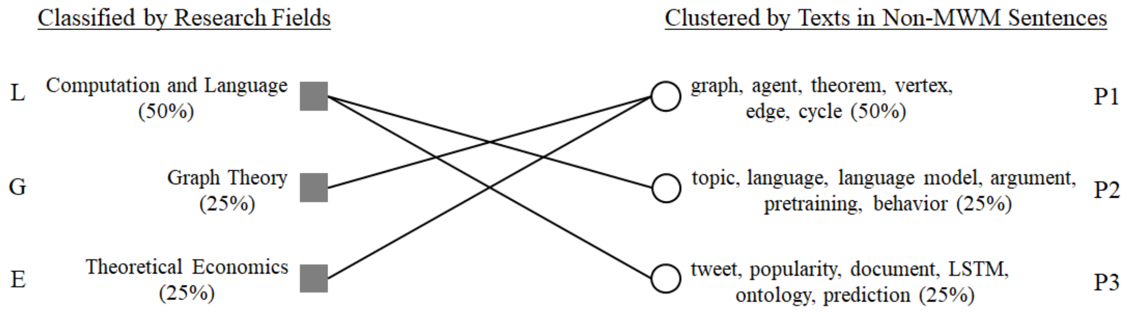

Classified by Research Fields
Clustered by Texts in Non-MWM Sentences
L
Computation and Language (50%)
G
Graph Theory (25%)
E
Theoretical Economics (25%)
graph, agent, theorem, vertex, edge, cycle (50%)
P1
topic, language, language model, argument, pretraining, behavior (25%)
P2
tweet, popularity, document, LSTM, ontology, prediction (25%)
P3

Figure 52: The bipartite graph for research fields and pure text clustering.

Summary

This chapter studies the collective processes of segmentation, interrelation, and reduction for the technical content abstraction of scientific documents. An MOR graph is proposed as a hierarchical data structure consisting of RBs and dependencies of MOs to shape the reasoning flows of the content. We have proposed to use the density of MOs distributed over sentences to set the boundaries of RBs. The center of an RB is a few large main equations associated with some smaller MOs and words. The identification of the main equations in RB are based on the four criteria: (1) the density of MO in the sentence; (2) the complexity of the equation presentation; (3) the frequency of equation in the content; (4) the degree of MOs denoting the equation. For the final product of the MOR graph, we conducted three case studies in cross-paper analysis based on a pilot dataset with 210 documents in 7 fields crawled from the arXiv.org. The results have shown that the proposed MOR graphical model is useful in capturing the technical semantics and the structural representation of the reasoning flows of documents from different research fields.

CHAPTER VI

CONCLUSIONS

This dissertation investigated and developed a mathematical object (MO) based analytical framework through a series of technical content transformations including MO constraint (MOC) classification, MO declaration (MOD) extraction, and MO reasoning (MOR) graph abstraction, to support modeling of reasoning flows in scientific publications. The proposed underlying system is design for a large-scale deep content analysis based on the idea of reasoning flows given the logical order of inductive and deductive process for reader to conquer the technical substance in a bottom up manner. We defined the notion of reasoning blocks (RB) as a self-contained technical discussion consisting of main equations as the problem formulation and their problem settings (i.e., the MOC) and physical semantics (i.e., the MOD) as the technical substances to assist understanding of the technical essence. The technical contributions of this work are summarized as follows.

First, we started with establishing an automated predictive model to identify the MOC expressions used for describing the settings of a problem formulation. Existing technologies are limited to handcrafted ad hoc rules for identifications of the constraint statements. The proposed model is based on the naïve Bayesian approach to consider feature distributions of mathematical symbols and contextual attached words in building the inference engine. It is scalable to allow new data arrival due to the assumption of features' conditional independence. The prediction results have achieved an 81% F1 score on the Elsevier dataset OA-STM Corpus, and we have discovered that mathematical symbols carry stronger indicators than words in determining whether an MO is a constraint expression.

Second, we proposed a predictive framework for finding the MOD pairs between MOs and words based on spatial, syntactic, and semantic information of the intermediate tokens which connect the two. We have addressed three critical issues for the problem: (1) a POS tagger to handle the syntactic parsing of sentences with mixed use of words and MOs (MWM); (2) a higher level constituent parsing built upon the MWM tagger to locate possible groups of words for MOD; (3) predictive mapping of the MOs to their MOD candidates in the sentence. The final products include a customized mathematical language processing toolkit, a shallow parser for MOD candidates, and an ensemble of classifiers to finalize the prediction of MOD. We have achieved a 75% F1 score on short declarations and a 71% F1 score on full declarations from the existing dataset NTCIR-10.

Finally, we created a novel abstraction, the MOR graph, to highlight the technical essence of a scientific paper. The core of the structure is the RBs used to encapsulate the self-contained MO-based technical discussions. RB contains the main equations which highlight the objective of the discussion. The MOD annotated the meaning of the internal components of the main equation, and the MOC are associated with the main equations based on MO dependency analysis. We have observed four criteria for main equations including: (1) they present as stand-alone function in a sentence; (2) they are relatively complex and abstract in their presentations; (3) they are not repeatedly mentioned in the content; (4) they are associated with a lot of MOs to express the depth of the problem. To extract main equations from MOs, we defined a z-score normalization method on measurements of the above four criteria of MO. Preliminary results have shown that the normalization can successfully put main equations in a higher rank compared to other equations. For MO dependency analysis, we have addressed the open problem of multiple-character identifier in finding the common identifier set among MO pairs. An assessment function based on the frequency of collocation is proposed to effectively resolve the problem. The MOR graph is designed as a scalable structure to reduce the complexity of the visualization, which allows progressive revelation of technical details customized by different perspectives of technical importance. We have conducted case studies on properties of the MOR graph including: (1) document clustering based on MODs and (2) structure-level differential analysis based on reasoning flows and MO dependencies in the paper. Results have shown the potential usefulness of applying the MOR graph in cross-paper analysis. Scientific documents can be successfully classified based on the MODs and writing style as modeled by the MOR graph.

For future work, the current datasets used for training the MOC and MOD models are too small, scattered, and error prone. Prediction errors can further propagate and affect performance of the MOR graph. To validate the representativeness of performance results of MOC and MOD, we plan to collaborate with a team to develop a user-friendly web annotation system that collects large-scale ground truths via crowdsourcing. This platform will be the basis to link between user input space and the existing models for MOC and MOD, so that real-time feedback will be provided to help refining current solution models to a convergence point. A comparative study will be carried out between annotations with or without real-time feedback. The measurement includes the number of steps as well as correctness of the annotation. The prototype system is still under development with basic PDF rendering and annotation ability for MOC and MOD labeling including the database design. User's feedback will later be incorporated into the training and inference loop. Last but not least, we will continue exploring the potential features of the MOR graph to formulate useful document fingerprints for cross-paper analysis.

REFERENCES

[1] F. Xia, W. Wang, T. M. Bekele and H. Liu, "Big Scholarly Data: A Survey," *IEEE Transactions on Big Data,* vol. 3, no. 1, pp. 18-35, 2017.

[2] S. B. Roy, M. De Cock, V. Mandava, S. Savanna, B. Dalessandro, C. Perlich, W. Cukierski and B. Hamner, "The Microsoft Academic Search Dataset and KDD Cup 2013," in *Proceedings of the 2013 KDD Cup 2013 Workshop*, Chicago, 2013.

[3] R. Baeza-Yates and B. Ribeiro-Neto, Modern Information Retrieval, vol. 463, New York: ACM, 1999.

[4] C. D. Manning, P. Raghavan and H. Schütze, An Introduction to Information Retrieval, Cambridge, England: Cambridge University Press, 2009.

[5] C. L. Giles, K. D. Bollacker and S. Lawrence, "CiteSeer: An Automatic Citation Indexing System," in *The 3rd ACM Conference on Digital Libraries*, Pittsburgh, PA, 1998.

[6] I. G. Councill, C. L. Giles and M.-Y. Kan, "ParsCit: An Open-Source CRF Reference String Parsing Package," in *The 6th International Conference on Language Resources and Evaluation*, Marrakech, Morocco, 2008.

[7] J. Beel, B. Gipp, S. Langer and C. Breitinger, "Paper Recommender Systems: A Literature Survey," *International Journal on Digital Libraries,* vol. 17, no. 4, pp. 305-338, 2016.

[8] A. A. Pogorui, "Hyperholomorphic functions on commutative algebras," *Complex Variables and Elliptic Equations,* vol. 52, no. 12, pp. 1155-1159, 2007.

[9] D. E. Knuth, TEX and METAFONT: New Directions in Typesetting, Boston, MA: American Mathematical Society, 1979.

[10] ISO, PDF Reference, Adobe, 2006.

[11] L. Lamport, LATEX: A Document Preparation System, Addison-Wesley, 1994.

[12] Y. Shinyama, "PDFMiner: Python PDF Parser and Analyzer," 29 October 2017. [Online]. Available: https://github.com/euske/pdfminer.

[13] "PDFBox: A Java PDF Library," The Apache Software Foundation, 20 September 2019. [Online]. Available: https://pdfbox.apache.org/download.cgi.

[14] H.-J. Lee and J.-S. Wang, "Design of a Mathematical Expression Recognition System," in *Proceedings of the 3rd International Conference on Document Analysis and Recognition*, Montreal, Canada, 1995.

[15] K. Inoue, R. Miyazaki and M. Suzuki, "Optical Recognition of Printed Mathematical Documents," in *Proceedings of the 3rd Asian Technology Conference in Mathematics*, Singapore, 1998.

[16] M. Suzuki, F. Tamari, R. Fukuda, S. Uchida and T. Kanahori, "INFTY: An Integrated OCR System for Mathematical Documents," in *Proceedings of the 10th ACM Symposium on Document engineering*, Grenoble, 2003.

[17] J. B. Baker, A. P. Sexton and V. Sorge, "A Linear Grammar Approach to Mathematical Formula Recognition from PDF," in *International Conference on Intelligent Computer Mathematics*, Grand Bend, Canada, 2009.

[18] J. B. Baker, A. P. Sexton and V. Sorge, "Faithful Mathematical Formula Recognition from PDF Documents," in *Proceedings of the 9th IAPR International Workshop on Document Analysis Systems*, 2010.

[19] X. Lin, L. Gao, Z. Tang, X. Lin and X. Hu, "Mathematical Formula Identification in PDF Documents," in *Proceedings of the 12th International Conference on Document Analysis and Recognition*, Beijing, China, 2011.

[20] X. Lin, L. Gao, Z. Tang, X. Hu and X. Lin, "Identification of Embedded Mathematical Formulas in PDF Documents Using SVM," in *IS&T/SPIE Electronic Imaging*, Burlingame, CA, 2012.

[21] X. Lin, L. Gao, Z. Tang, J. Baker and V. Sorge, "Mathematical Formula Identification and Performance Evaluation in PDF Documents," *International Journal on Document Analysis and Recognition,* vol. 17, no. 3, pp. 239-255, 2014.

[22] L. Gao, X. Yi, Y. Liao, Z. Jiang, Z. Yan and Z. Tang, "A Deep Learning-Based Formula Detection Method for PDF Documents," in *Proceeding of the 14th IAPR International Conference on Document Analysis and Recognition*, Kyoto, Japan, 2017.

[23] X. Wang and J.-C. Liu, "A Font Setting Based Bayesian Model to Extract Mathematical Expression in PDF Files," in *Proceedings of the 14th IAPR International Conference on Document Analysis and Recognition*, Kyoto, Japan, 2017.

[24] Z. Wang, D. Beyette, J. Lin and J.-C. Liu, "Extraction of Math Expressions from PDF Documents Based on Unsupervised Modeling of Fonts," in *International Conference on Document Analysis and Recognition*, Sydney, Australia, 2019.

[25] S. Klink, A. Dengel and T. Kieninger, "Document Structure Analysis Based on Layout and Textual Features," in *Proceedings of the 9th IAPR International Workshop on Document Analysis Systems*, Boston Massachusetts, 2000.

[26] A. Constantin, S. Pettifer and A. Voronkov, "PDFX: Fully-automated PDF-to-XML Conversion of Scientific Literature," in *ACM Symposium on Document Engineering*, Florence, Italy, 2013.

[27] D. Tkaczyk, P. Szostek, M. Fedoryszak, P. J. Dendek and Ł. Bolikowski, "CERMINE: Automatic Extraction of Structured Metadata from Scientific Literature," *International Journal on Document Analysis and Recognition,* vol. 18, no. 4, pp. 317-335, 2015.

[28] M. Singh, B. Barua, P. Palod, M. Garg, S. Satapathy, S. Bushi, K. Ayush, K. S. Rohith, T. Gamidi, P. Goyal and A. Mukherjee, "OCR++: A Robust Framework for Information Extraction from Scholarly Articles," in *Proceedings of the 26th International Conference on Computational Linguistics*, Osaka, Japan, 2016.

[29] C. Soto and S. Yoo, "Visual Detection with Context for Document Layout Analysis," in *Proceedings of the 2019 Conference on EMNLP-IJCNLP*, Hong Kong, China, 2019.

[30] K. Nakagawa, A. Nomura and M. Suzuki, "Extraction of Logical Structure from Articles in Mathematics," in *International Conference on Mathematical Knowledge Management*, Bialowieza, Poland, 2004.

[31] F. Kamareddine and J. B. Wells, "Computerizing Mathematical Text with MathLang," *Electronic Notes in Theoretical Computer Science,* vol. 205, pp. 5-30, 2008.

[32] V. Solovyev and N. Zhiltsov, "Logical Structure Analysis of Scientific Publications in Mathematics," in *Proceedings of the International Conference on Web Intelligence, Mining and Semantics*, Sogndal, Norway, 2011.

[33] X. Wang, J. Lin, R. Vrecenar and J.-C. Liu, "QuQn Map: Qualitative-Quantitative Mapping of Scientific Papers," in *Proceedings of the 18th ACM Symposium on Document Engineering*, Halifax, NS, Canada, 2018.

[34] X. Wang and J.-C. Liu, "A Content-Constrained Spatial (CCS) Model for Layout Analysis of Mathematical Expressions," in *International Conference on Digital Information Management*, Fukuoka, Japan, 2017.

[35] C.-N. Lin, C.-C. Chang, W.-B. Lee and J. Lin, "A Novel Secure Data Hiding Scheme Using a Secret Reference Matrix," in *International Conference on Intelligent Information Hiding and Multimedia Signal Processing*, Kyoto, Japan, 2009.

[36] Elsevier, "OA STM Corpus," Elsevier Labs, 2015. [Online]. Available: https://github.com/elsevierlabs/OA-STM-Corpus.

[37] A. Aizawa, M. Kohlhase and I. Ounis, "NTCIR-10 Math Pilot Task Overview," in *NTCIR*, 2013.

[38] "KDD Cup 2003," Cornell University, 2003. [Online]. Available: https://www.cs.cornell.edu/projects/kddcup/index.html.

[39] arXiv, "arXiv.org," Cornell University, [Online]. Available: https://arxiv.org/.

[40] "LaTeX Math Annotations for Elsevier OA-STM-Corpus," Texas A&M University, 2017. [Online]. Available: http://rtds.cse.tamu.edu/resources/.

[41] "LaTeX Constraint Annotations for Elsevier OA-STM-Corpus," Texas A&M University, June 2018. [Online]. Available: http://rtds.cse.tamu.edu/resources/.

[42] D. Beyette, Z. Wang, J. Lin and J.-C. Liu, "Semi-automatic LaTeX-Based Labeling of Mathematical Objects in PDF Documents: MOP Data Set," in *Proceedings of the 19th ACM Symposium on Document Engineering*, Berlin, Germany, 2019.

[43] S. Bird, "NLTK: The Natural Language Toolkit," in *Proceedings of the COLING/ACL 2006 Interactive Presentation Sessions*, Sydney, Australia, 2006.

[44] C. Manning, M. Surdeanu, J. Bauer, J. Finkel, S. Bethard and D. McClosky, "The Stanford CoreNLP Natural Language Processing Toolkit," in *Proceedings of*

52nd Annual Meeting of The Association for Computational Linguistics: System Demonstrations, Baltimore, Maryland, 2014.

[45] S. Lok and S. Feiner, "A Survey of Automated Layout Techniques for Information Presentations," in *Proceedings of the 2nd International Symposium on Smart Graphics*, Hawthorne, NY, 2001.

[46] M. Ailomaa and M. Rajman, "Natural Language Techniques for Model-Driven Semantic Constraint Extraction," LIA, Lausanne, Switzerland, 2007.

[47] D. T. Wei, J. Wang and Y. Chen, "Extracting Semantic Constraint from Description Text for Semantic Web Service Discovery," in *The 7th International Semantic Web Conference*, Karlsruhe, Germany, 2008.

[48] R. L. Campbell and M. H. Bickhard, "Types of Constraints on Development: An Interactivist Approach," *Developmental Review,* vol. 12, no. 3, pp. 311-338, 1992.

[49] J. Misra, M. Savagaonkar, N. Dubash, S. Podder and S. Hanumantappa Waddar, "Constraint Extraction from Natural Language Text for Test Data Generation". US Patent No. 14/990,051, 7 January 2016.

[50] J. Lin, X. Wang and J.-C. Liu, "Prediction of Mathematical Expression Constraints (ME-Con)," in *ACM Symposium on Document Engineering*, Halifax, NS, Canada, 2018.

[51] M. Porter, "Snowball: A Language for Stemming Algorithms," 2001. [Online]. Available: http://snowball.tartarus.org/.

[52] M. P. Marcus, M. A. Marcinkiewicz and B. Santorini, "Building a Large Annotated Corpus of English: The Penn Treebank," *Computational Linguistics,* vol. 19, no. 2, pp. 313-330, 1993.

[53] M. Wolska and M. Grigore, "Symbol Declarations in Mathematical Writing," in *Towards a Digital Mathematics Library*, Paris, France, 2010.

[54] M. Schubotz, A. Grigorev, M. Leich, H. S. Cohl, N. Meuschke, B. Gipp, A. S. Youssef and V. Markl, "Semantification of Identifiers in Mathematics for Better Math Information Retrieval," in *Proceedings of the 39th International ACM SIGIR Conference on Research and Development in Information Retrieval*, Pisa, Italy, 2016.

[55] M. Schubotz, L. Krämer, N. Meuschke, F. Hamborg and B. Gipp, "Evaluating and Improving the Extraction of Mathematical Identifier Definitions," in *International Conference of the Cross-Language Evaluation Forum for European Languages*, Dublin, Ireland, 2017.

[56] G. Y. Kristianto, G. Topić and A. Aizawa, "Utilizing Dependency Relationships between Math Expressions in Math IR," *Information Retrieval Journal*, vol. 20, no. 2, pp. 132-167, 2017.

[57] G. Y. Kristianto, M. Ngiem, Y. Matsubayashi and A. Aizawa, "Extracting Definitions of Mathematical Expressions in Scientific Papers," in *Proceedings of the 26th Annual Conference of the Japanese Society for Artificial Intelligence*, Tokyo, 2012.

[58] J. Lin, X. Wang, Z. Wang, D. Beyette and J.-C. Liu, "Prediction of Mathematical Expression Declarations Based on Spatial, Semantic, and Syntactic Analysis," in *ACM Symposium on Document Engineering*, Berlin, Germany, 2019.

[59] T. Brants, "TnT: A Statistical Part-Of-Speech Tagger," in *Proceedings of the 6th Conference on Applied Natural Language Processing*, Seattle, Washington, 2000.

[60] C. W. Günther and W. M. Van Der Aalst, "Fuzzy Mining–Adaptive Process Simplification Based on Multi-Perspective Metrics," in *International Conference on Business Process Management*, Brisbane, QLD, 2007.

[61] M.-N. Quoc, K. Yokoi, Y. Matsubayashi and A. Aizawa, "Mining Coreference Relations between Formulas and Text Using Wikipedia," in *The 2nd International Workshop on NLP Challenges in the Information Explosion Era*, Beijing, China, 2010.

[62] G. Y. Kristianto, G. Topić and A. Aizawa, "Extracting Textual Descriptions of Mathematical Expressions in Scientific Papers," *D-Lib Magazine*, vol. 20, no. 11, p. 9, 2014.

[63] U. Schöneberg and W. Sperber, "POS Tagging and Its Applications for Mathematics," in *International Conference on Intelligent Computer Mathematics*, Coimbra, Portugal, 2014.

[64] A. Ratnaparkhi, "A Maximum Entropy Model for Part-Of-Speech Tagging," in *International Conference on Empirical Methods in Natural Language Processing*, Philadelphia, PA, 1996.

[65] M. Collins, "Discriminative Training Methods for Hidden Markov Models: Theory and Experiments with Perceptron Algorithms," in *International Conference on Empirical Methods in Natural Language Processing*, Philadelphia, PA, 2002.

[66] K. Toutanova, D. Klein, C. D. Manning and Y. Singer, "Feature-rich Part-Of-Speech Tagging with a Cyclic Dependency Network," in *Proceedings of NAACL*, Edmonton, Canada, 2003.

[67] X. Wang, J. Lin, R. Vrecenar and J.-C. Liu, "Syntactic Role Identification of Mathematical Expressions," in *International Conference on Digital Information Management,* Fukuoka, Japan, 2017.

[68] M. Wolska and I. Kruijff-Korbayová, "Analysis of Mixed Natural and Symbolic Input in Mathematical Dialogs," in *Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics*, Barcelona, Spain, 2004.

[69] M. Ganesalingam, The Language of Mathematics: A Linguistic and Philosophical Investigation, Cambridge, UK: Springer, 2013.

[70] K. Fundel, R. Küffner and R. Zimmer, "RelEx—Relation Extraction Using Dependency Parse Trees," *Bioinformatics,* vol. 23, no. 3, pp. 365-371, 2006.

[71] I. Sag, T. Baldwin, F. Bond, A. Copestake and D. Flickinger, "Multiword Expressions: A Pain in the Neck for NLP," in *International Conference on Intelligent Text Processing and Computational Linguistics*, Mexico City, Mexico, 2002.

[72] M. Bayraktar, B. Say and V. Akman, "An Analysis of English Punctuation: The Special Case of Comma," *International Journal of Corpus Linguistics,* vol. 3, no. 1, pp. 33-57, 1998.

[73] P. Nakov and M. Hearst, "Using the Web as an Implicit Training Set: Application to Structural Ambiguity Resolution," in *Proceedings of the Conference on Human Language Technology and Empirical Methods in Natural Language Processing,* Vancouver, British Columbia, Canada, 2005.

[74] M. Goldberg, "An Unsupervised Model for Statistically Determining Coordinate Phrase Attachment," in *Proceedings of the 37th Annual Meeting of the Association for Computational Linguistics,* College Park, MD, 1999.

[75] P. Resnik, "Semantic Similarity in a Taxonomy: An Information-Based Measure and Its Application to Problems of Ambiguity in Natural Language," *Journal of Artificial Intelligence Research,* vol. 11, pp. 95-130, 1999.

[76] "POS Tagging (State of the art)," [Online]. Available: https://aclweb.org/aclwiki/POS_Tagging_(State_of_the_art).

[77] B. Miller, "LaTeXML: A LaTeX to XML Converter," NIST, [Online]. Available: https://dlmf.nist.gov/LaTeXML/.

[78] A. Viterbi, "Error Bounds for Convolutional Codes and an Asymptotically Optimum Decoding Algorithm," *IEEE Transactions on Information Theory,* vol. 13, no. 2, pp. 260-269, 1967.

[79] C. W. Günther and A. Rozinat, "Disco: Discover Your Processes," in *International Conference on Business Process Management*, Tallinn, Estonia, 2012.

[80] B. Gastel and R. A. Day, How to Write and Publish a Scientific Paper, Santa Barbara, CA: ABC-CLIO, LLC, 2016.

[81] E. J. Rothwell and M. Cloud, Engineering Writing by Design: Creating Formal Documents of Lasting Value, USA: CRC Press, 2017.

[82] J. Peat, E. Elliott, L. Baur and V. Keena, Scientific Writing: Easy When You Know How, London, UK: BMJ Books, 2002.

[83] T. K. Attwood, D. B. Kell, P. McDermott, J. Marsh, S. R. Pettifer and D. Thorne, "Utopia Documents: Linking Scholarly Literature with Research Data," *Bioinformatics,* vol. 26, no. 18, pp. i568-i574, 2010.

[84] M. Schubotz, N. Meuschke, T. Hepp, H. S. Cohl and B. Gipp, "VMEXT: A Visualization Tool for Mathematical Expression Trees," in *International Conference on Intelligent Computer Mathematics*, Edinburgh, UK, 2017.

[85] "Semantic Taxonomy of the Mathematical Expressions for QuQn Map," RTDS Lab, Texas A&M University, 2018. [Online]. Available: https://rtds.cse.tamu.edu/resources/.

[86] M. Ružicka, P. Sojka and M. Líška, "Math Indexer and Searcher under the Hood: History and Development of a Winning Strategy," in *Proceedings of the 11th NTCIR Conference on Evaluation of Information Access Technologies*, Tokyo, Japan, 2014.

[87] R. Ausbrooks, S. Buswell, D. Carlisle, S. Dalmas, S. Devitt and A. Diaz, "Mathematical Markup Language (MathML) Version 2.0. W3C Recommendation," World Wide Web Consortium, 2003.

[88] X. Wang, MECA: Mathematical Expression Based Post Publication Content Analysis, College Station: Texas A&M University, 2018.

[89] E. R. Tufte, The Visual Display of Quantitative Information, vol. 7.3, Cheshire, CT, 1983.

[90] A. Youssef, "Part-Of-Math Tagging and Applications," in *International Conference on Intelligent Computer Mathematics*, Edinburgh, UK, 2017.

[91] D. Biber, "Co-occurrence Patterns among Collocations: A Tool for Corpus-Based Lexical Knowledge Acquisition," *Computational Linguistics,* vol. 19, no. 3, pp. 531-538, 1993.

[92] K. W. Church and P. Hanks, "Word Association Norms, Mutual Information, and Lexicography," *Computational Linguistics,* vol. 16, no. 1, pp. 22-29, 1990.

[93] F. Role and M. Nadif, "Handling the Impact of Low Frequency Events on Co-occurrence Based Measures of Word Similarity," in *Proceedings of the International Conference on Knowledge Discovery and Information Retrieval*, Paris, 2011.

[94] F. Smadja, "Retrieving Collocations from Text: Xtract," *Computational Linguistics,* vol. 19, no. 1, pp. 143-177, 1993.

[95] F. Janssens, J. Leta, W. Glänzel and B. D. Moor, "Towards Mapping Library and Information Science," *Information Processing & Management,* vol. 42, no. 6, pp. 1614-1642, 2006.

[96] R. N. Kostoff, J. A. del Rio, J. A. Humenik, E. O. García and A. M. Ramírez, "Citation Mining: Integrating Text Mining and Bibliometrics for Research User

Profiling," *Journal of the Association for Information Science and Technology,* vol. 52, no. 13, pp. 1148-1156, 2001.

[97] P. Glenisson, W. Glänzel and O. Persson, "Combining Full-Text Analysis and Bibliometric Indicators: A Pilot Study," *Scientometrics*, vol. 23, no. 3, pp. 163-180, 2005.

[98] S. Deerwester, S. T. Dumais, G. W. Furnas, T. K. Landauer and R. Harshman, "Indexing by Latent Semantic Analysis," *Journal of the American Society for Information Science,* vol. 41, no. 6, pp. 391-407, 1990.

[99] D. M. Blei, A. Y. Ng and M. I. Jordan, "Latent Dirichlet Allocation," *Journal of Machine Learning Research,* vol. 3, no. Jan, pp. 993-1022, 2003.

[100] S. Teufel, A. Siddharthan and D. Tidhar, "Automatic Classification of Citation Function," in *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing*, Sydney, Australia, 2006.

[101] B. R. Miller and A. Youssef, "Technical Aspects of the Digital Library of Mathematical Functions," *Annals of Mathematics and Artificial Intelligence,* vol. 38, no. 1-3, p. 121–136, 2003.

[102] J. Mišutka and L. Galamboš, "Extending Full Text Search Engine for Mathematical Content," in *Towards Digital Mathematics Library*, Birmingham, UK, 2008.

[103] A. Thanda, A. Agarwal, K. Singla, A. Prakash and A. Gupta, "A Document Retrieval System for Math Queries," in *Proceedings of the 12th NTCIR Conference on Evaluation of Information Access Technologies,* Tokyo, Japan, 2016.

[104] R. Zanibbi, K. Davila, A. Kane and F. Tompa, "Multi-Stage Math Formula Search: Using Appearance-Based Similarity Metrics at Scale," in *Proceedings of the 39th International ACM SIGIR Conference on Research and Development in Information Retrieval,* Pisa, Italy, 2016.

[105] K. Davila and R. Zanibbi, "Layout and Semantics: Combining Representations for Mathematical Formula Search," in *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval*, Shinjuku, Tokyo, 2017.

[106] Y. Wang, L. Gao, S. Wang, Z. Tang, X. Liu and K. Yuan, "WikiMirs 3.0: A Hybrid MIR System Based on the Context, Structure and Importance of Formulae in a Document," in *Proceedings of the 15th ACM/IEEE-CS Joint Conference on Digital Libraries*, Knoxville, TN, 2015.

[107] K. Yuan, L. Gao, Z. Jiang and Z. Tang, "Formula Ranking within an Article," in *Proceedings of the 18th ACM/IEEE on Joint Conference on Digital Libraries*, Fort Worth, Texas, 2018.

[108] A. Greiner-Petter, T. Ruas, M. Schubotz, A. Aizawa, W. Grosky and B. Gipp, "Why Machines Cannot Learn Mathematics, Yet," in *The 4th BIRNDL Workshop at 42nd SIGIR*, Paris, France, 2019.

[109] V. I. Levenshtein, "Binary Codes Capable of Correcting Deletions, Insertions, and Reversals," *Soviet Physics Doklady,* vol. 10, no. 8, pp. 707-710, 1966.

[110] H. P. Luhn, "A Statistical Approach to Mechanized Encoding and Searching of Literary Information," *IBM Journal of Research and Development,* vol. 1, no. 4, pp. 309-317, 1957.

[111] K. S. Jones, "A Statistical Interpretation of Term Specificity and Its Application in Retrieval," *Journal of Documentation,* vol. 28, pp. 11-21, 1972.

[112] L. Tawade, R. Mahajan and C. Kulthe, "Efficient & Secure Data Hiding using Secret Reference Matrix," *International Journal of Network Security & Its Applications,* vol. 4, no. 1, p. 43, 2012.

[113] J. MacQueen, "Some Methods for Classification and Analysis of Multivariate Observations," in *Proceedings of the fifth Berkeley Symposium on Mathematical Statistics and Probability,* Berkeley, CA, 1967.

[114] H. Prüfer, "New Proof of A Theorem on Permutations," *Arch d. Math. u. Phys.*, vol. 3, no. 27, pp. 142-144, 1918.

[115] J. H. Ward Jr, "Hierarchical grouping to optimize an objective function," *Journal of the American Statistical Association*, vol. 58, no. 301, pp. 236-244, 1963.

[116] A. Taylor, M. Marcus and B. Santorini, "The Penn Treebank: An Overview," in *Treebanks*, Springer, Dordrecht, 2003, pp. 5-22.

APPENDIX A

PART-OF-SPEECH (POS) TAGS

Table 9: The English Penn Treebank POS tags [116]

Tag	Description	Tag	Description
$	Dollar	NNS	Noun, plural
:	Colon	NNP	Proper noun, singular
,	Comma	NNPS	Proper noun, plural
.	Period	PDT	Predeterminer
“ ‘	Left quote	POS	Possessive ending
” ‘	Right quote	PRP	Personal pronoun
-LRB-	Left bracket	PRP$	Possessive pronoun
-RRB-	Right bracket	RB	Adverb
ADD	Email	RBR	Adverb, comparative
AFX	Affix	RBS	Adverb, superlative
CC	Coordinating conjunction	RP	Particle
CD	Cardinal number	SYM	Symbol
DT	Determiner	TO	To
EX	Existential there	UH	Interjection
FW	Foreign word	VB	Verb, base form
GW	Go with	VBD	Verb, past tense
HYPH	Hyphen	VBG	Verb, gerund or present participle
IN	Preposition or subordinate conjunction	VBN	Verb, past participle
JJ	Adjective	VBP	Verb, non-3rd person singular present
JJR	Adjective, comparative	VBZ	Verb, 3rd person singular present
JJS	Adjective, superlative	WDT	Wh-determiner
LS	List item marker	WP	Wh-pronoun
MD	Modal	WP$	Wh-pronoun, possessive
NFP	Superfluous punctuation	WRB	Wh-adverb

APPENDIX B

THE DATA STRUCTURE OF MATHEMATICAL SEMANTICS

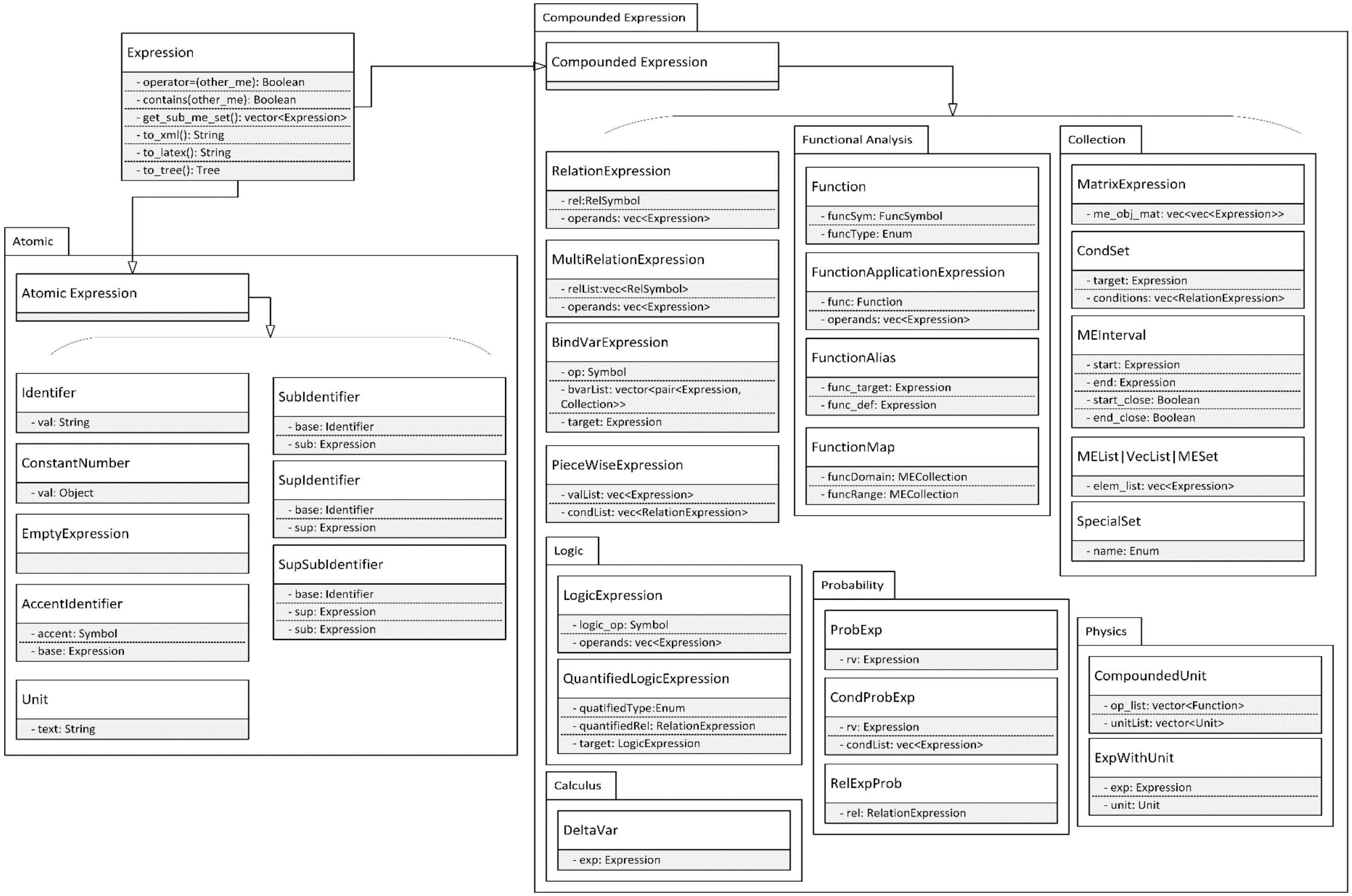
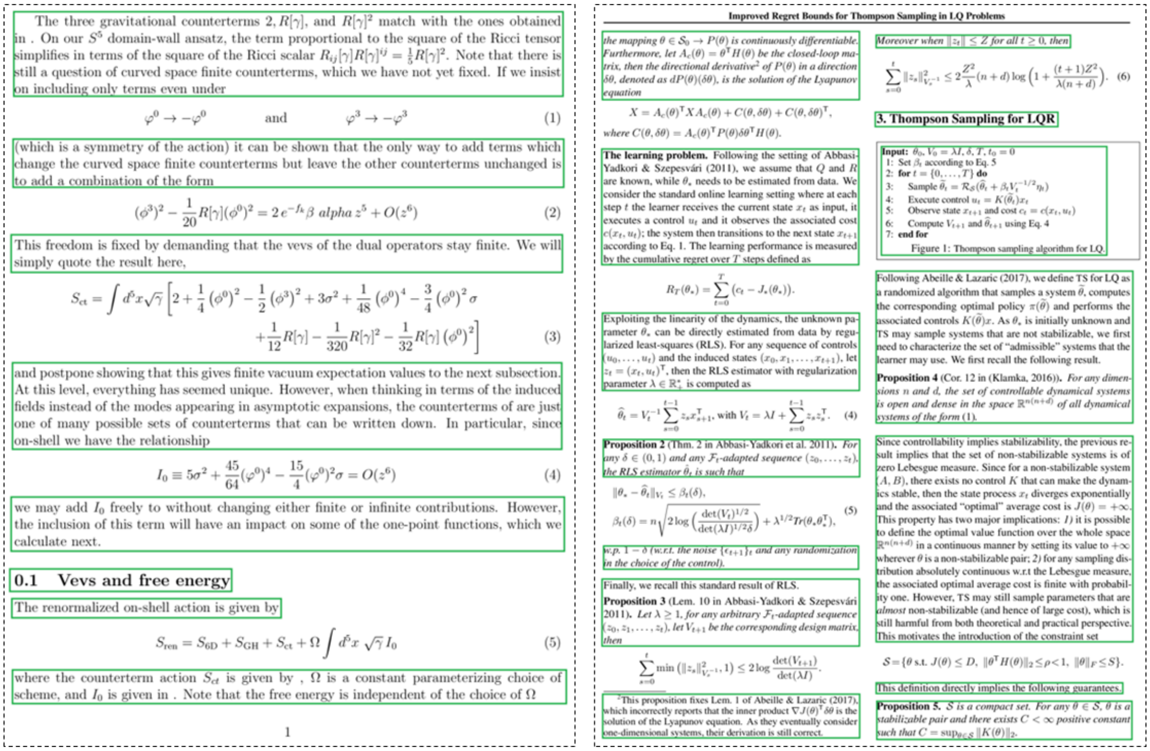


Figure 53: The semantic taxonomy of mathematical object [88]

APPENDIX C

A CASE STUDY OF PARAGRAPH EXTRACTION



The three gravitational counterterms $2, R[\gamma]$, and $R[\gamma]^2$ match with the ones obtained in . On our S^5 domain-wall ansatz, the term proportional to the square of the Ricci tensor simplifies in terms of the square of the Ricci scalar $R_{ij}[\gamma]R[\gamma]^{ij} = \frac{1}{5}R[\gamma]^2$. Note that there is still a question of curved space finite counterterms, which we have not yet fixed. If we insist on including only terms even under

$$\varphi^0 \to -\varphi^0 \qquad \text{and} \qquad \varphi^3 \to -\varphi^3 \tag{1}$$

(which is a symmetry of the action) it can be shown that the only way to add terms which change the curved space finite counterterms but leave the other counterterms unchanged is to add a combination of the form

$$(\phi^3)^2 - \frac{1}{20}R[\gamma](\phi^0)^2 = 2\,e^{-f_k}\beta\ alpha\, z^5 + O(z^6) \tag{2}$$

This freedom is fixed by demanding that the vevs of the dual operators stay finite. We will simply quote the result here,

$$\begin{aligned} S_{\text{ct}} = \int d^5x\sqrt{\gamma}\Big[2 + \frac{1}{4}\left(\phi^0\right)^2 - \frac{1}{2}\left(\phi^3\right)^2 + 3\sigma^2 + \frac{1}{48}\left(\phi^0\right)^4 - \frac{3}{4}\left(\phi^0\right)^2\sigma \\ + \frac{1}{12}R[\gamma] - \frac{1}{320}R[\gamma]^2 - \frac{1}{32}R[\gamma]\left(\phi^0\right)^2\Big] \end{aligned} \tag{3}$$

and postpone showing that this gives finite vacuum expectation values to the next subsection. At this level, everything has seemed unique. However, when thinking in terms of the induced fields instead of the modes appearing in asymptotic expansions, the counterterms of are just one of many possible sets of counterterms that can be written down. In particular, since on-shell we have the relationship

$$I_0 \equiv 5\sigma^2 + \frac{45}{64}(\varphi^0)^4 - \frac{15}{4}(\varphi^0)^2\sigma = O(z^6) \tag{4}$$

we may add I_0 freely to without changing either finite or infinite contributions. However, the inclusion of this term will have an impact on some of the one-point functions, which we calculate next.

0.1 Vevs and free energy

The renormalized on-shell action is given by

$$S_{\text{ren}} = S_{\text{6D}} + S_{\text{GH}} + S_{\text{ct}} + \Omega\int d^5x\,\sqrt{\gamma}\,I_0 \tag{5}$$

where the counterterm action S_{ct} is given by , Ω is a constant parameterizing choice of scheme, and I_0 is given in . Note that the free energy is independent of the choice of Ω

1

Improved Regret Bounds for Thompson Sampling in LQ Problems

the mapping $\theta \in \mathcal{S}_0 \to P(\theta)$ is continuously differentiable. Furthermore, let $A_c(\theta) = \theta^\mathsf{T} H(\theta)$ be the closed-loop matrix, then the directional derivative[2] *of $P(\theta)$ in a direction $\delta\theta$, denoted as $dP(\theta)(\delta\theta)$, is the solution of the Lyapunov equation*

$$X = A_c(\theta)^\mathsf{T} X A_c(\theta) + C(\theta, \delta\theta) + C(\theta, \delta\theta)^\mathsf{T},$$

where $C(\theta, \delta\theta) = A_c(\theta)^\mathsf{T} P(\theta)\delta\theta^\mathsf{T} H(\theta)$.

The learning problem. Following the setting of Abbasi-Yadkori & Szepesvári (2011), we assume that Q and R are known, while θ_* needs to be estimated from data. We consider the standard online learning setting where at each step t the learner receives the current state x_t as input, it executes a control u_t and it observes the associated cost $c(x_t, u_t)$; the system then transitions to the next state x_{t+1} according to Eq. 1. The learning performance is measured by the cumulative regret over T steps defined as

$$R_T(\theta_*) = \sum_{t=0}^{T}\left(c_t - J_*(\theta_*)\right).$$

Exploiting the linearity of the dynamics, the unknown parameter θ_* can be directly estimated from data by regularized least-squares (RLS). For any sequence of controls $(u_0, \ldots, u_t)$ and the induced states $(x_0, x_1, \ldots, x_{t+1})$, let $z_t = (x_t, u_t)^\mathsf{T}$, then the RLS estimator with regularization parameter $\lambda \in \mathbb{R}_+^*$ is computed as

$$\widehat{\theta}_t = V_t^{-1}\sum_{s=0}^{t-1} z_s x_{s+1}^\mathsf{T}, \text{ with } V_t = \lambda I + \sum_{s=0}^{t-1} z_s z_s^\mathsf{T}. \tag{4}$$

Proposition 2 (Thm. 2 in Abbasi-Yadkori et al. 2011)**.** *For any $\delta \in (0,1)$ and any $\mathcal{F}_t$-adapted sequence $(z_0, \ldots, z_t)$, the RLS estimator $\hat{\theta}_t$ is such that*

$$\begin{aligned} &\|\theta_* - \widehat{\theta}_t\|_{V_t} \le \beta_t(\delta), \\ &\beta_t(\delta) = n\sqrt{2\log\left(\frac{\det(V_t)^{1/2}}{\det(\lambda I)^{1/2}\delta}\right)} + \lambda^{1/2} Tr(\theta_*\theta_*^\mathsf{T}), \end{aligned} \tag{5}$$

w.p. $1 - \delta$ (w.r.t. the noise $\{\epsilon_{t+1}\}_t$ and any randomization in the choice of the control).

Finally, we recall this standard result of RLS.

Proposition 3 (Lem. 10 in Abbasi-Yadkori & Szepesvári 2011)**.** *Let $\lambda \ge 1$, for any arbitrary $\mathcal{F}_t$-adapted sequence $(z_0, z_1, \ldots, z_t)$, let V_{t+1} be the corresponding design matrix, then*

$$\sum_{s=0}^{t}\min\left(\|z_s\|^2_{V_s^{-1}}, 1\right) \le 2\log\frac{\det(V_{t+1})}{\det(\lambda I)}.$$

Moreover when $\|z_t\| \le Z$ for all $t \ge 0$, then

$$\sum_{s=0}^{t}\|z_s\|^2_{V_s^{-1}} \le 2\frac{Z^2}{\lambda}(n+d)\log\left(1 + \frac{(t+1)Z^2}{\lambda(n+d)}\right). \tag{6}$$

3. Thompson Sampling for LQR

```
Input: $\theta_0, V_0 = \lambda I, \delta, T, t_0 = 0$
Set $\beta_t$ according to Eq. 5
for $t = \{0, \ldots, T\}$ do
    Sample $\widetilde{\theta}_t = \mathcal{R}_\mathcal{S}(\widehat{\theta}_t + \beta_t V_t^{-1/2}\eta_t)$
    Execute control $u_t = K(\widetilde{\theta}_t)x_t$
    Observe state $x_{t+1}$ and cost $c_t = c(x_t, u_t)$
    Compute $V_{t+1}$ and $\widehat{\theta}_{t+1}$ using Eq. 4
end for
```

Figure 1: Thompson sampling algorithm for LQ.

Following Abeille & Lazaric (2017), we define TS for LQ as a randomized algorithm that samples a system $\widetilde{\theta}$, computes the corresponding optimal policy $\pi(\widetilde{\theta})$ and performs the associated controls $K(\widetilde{\theta})x$. As θ_* is initially unknown and TS may sample systems that are not stabilizable, we first need to characterize the set of "admissible" systems that the learner may use. We first recall the following result.

Proposition 4 (Cor. 12 in (Klamka, 2016))**.** *For any dimensions n and d, the set of controllable dynamical systems is open and dense in the space $\mathbb{R}^{n(n+d)}$ of all dynamical systems of the form* (1).

Since controllability implies stabilizability, the previous result implies that the set of non-stabilizable systems is of zero Lebesgue measure. Since for a non-stabilizable system (A, B), there exists no control K that can make the dynamics stable, then the state process x_t diverges exponentially and the associated "optimal" average cost is $J(\theta) = +\infty$. This property has two major implications: *1)* it is possible to define the optimal value function over the whole space $\mathbb{R}^{n(n+d)}$ in a continuous manner by setting its value to $+\infty$ wherever θ is a non-stabilizable pair; *2)* for any sampling distribution absolutely continuous w.r.t the Lebesgue measure, the associated optimal average cost is finite with probability one. However, TS may still sample parameters that are *almost* non-stabilizable (and hence of large cost), which is still harmful from both theoretical and practical perspective. This motivates the introduction of the constraint set

$$\mathcal{S} = \{\theta \text{ s.t. } J(\theta) \le D,\ \|\theta^\mathsf{T} H(\theta)\|_2 \le \rho < 1,\ \|\theta\|_F \le S\}.$$

This definition directly implies the following guarantees.

Proposition 5. *$\mathcal{S}$ is a compact set. For any $\theta \in \mathcal{S}$, θ is a stabilizable pair and there exists $C < \infty$ positive constant such that $C = \sup_{\theta \in \mathcal{S}}\|K(\theta)\|_2$.*

[2] This proposition fixes Lem. 1 of Abeille & Lazaric (2017), which incorrectly reports that the inner product $\nabla J(\theta)^\mathsf{T}\delta\theta$ is the solution of the Lyapunov equation. As they eventually consider one-dimensional systems, their derivation is still correct.

Figure 54: The preliminary results of paragraph extraction based on the content layout (in green rectangles)

APPENDIX D

SOME COMMON LATEX CODES

Table 10: A list of reserved words in LaTeX for different types of mathematical symbols

Category	Syntax
Display Styles	"\\displaystyle", "\\text", "\\mathcal", "\\textup"
Greek Letters	"\\alpha", "\\theta", "\\eta", "\\tau", "\\beta", "\\vartheta", "\\pi", "\\upsilon", "\\gamma", "\\varpi", "\\phi", "\\xi", "\\delta", "\\kappa", "\\rho", "\\varphi", "\\epsilon", "\\lambda", "\\varrho", "\\chi", "\\varepsilon", "\\mu", "\\sigma", "\\psi", "\\zeta", "\\nu", \\varsigma", "\\omega", "\\Gamma", "\\Lambda", "\\Sigma", "\\Psi", "\\Delta", "\\Xi", "\\Upsilon", "\\Omega", "\\Theta", "\\Pi", "\\Phi", "\\chi", "o"
Relational Symbols	"\\leq", "\\geq", "\\equiv", "\\models", "\\le", "\\ge", "\\prec", "\\succ", "\\sim", "\\perp", "\\preceq", "\\succeq", "\\simeq", "\\mid", "\\ll", "\\gg", "\\asymp", "\\parallel", "\\subset", "\\supset", "\\approx", "\bowtie", "\\subseteq", "\\supseteq", "\\cong", "\\Join", "\\sqsubset", "\\sqsupset", "\\neq", "\\sqsubseteq", "\\sqsupseteq", "\\doteq", "\\frown", "\\in", "\\ni", "\\propto", "=", "\\vdash", "\\dashv", "<", ">"
Big Operators	"\\sum", "\\bigcap", "\\bigodot", "\\prod", "\\bigcup", "\\bigotimes", "\\coprod", "\\bigsqcup", "\\bigoplus", "\\int", "\\bigvee", "\\biguplus", "\\oint", "\\bigwedge"

APPENDIX E

SOME COMMON MATHEMATICAL TERMS

Table 11: A list of mathematical terms used in different mathematical fields

Mathematical Fields	Mathematical Terms
Algebra	"det", "tr", "diag", "rank"
Arithmetic	"exp", "ln", "log", "remainder", "quotient"
Complex Number	"real", "real-part", "imaginary"
Full Name Conventions	"absolute value", "cosine", "sine", "secant", "ker", "kernel", "tangent", "inverse-sine", "maximum", "minimum", "exponential", "hyperbolic-tangent", "vector", "bra", "ket",
Functional Words	"domain", "Id", "range", "image", "domainofapplication", "left_compose", "left_inverse", "right_inverse", "apply_to_list", "kernel", "right_compose", "Laplacian", "curl"
Hyperbolic Function	"sinh", "cosh", "tanh", "coth", "sech", "csch"
Inverse Function	"arccos", "arccosh", "arccot", "arccoth", "arccsc", "arccsch", "arcsec", "arcsech", "arcsin", "arcsinh", "arctan", "arctanh"
Logic	"nand", "xor", "xnor", "nor"
Number	"ceil", "floor", "round", "trunc",
Number Theory	"round", "gcd", "lcm", "mod"
Probability	"logarithm", "Pr", "degree", "argument"
Proposition	"if", "iff", "for", "otherwise", "w.r.t", "s.t.", "i.e.", "e.g."
Set	"size", "make_list",
Statistics	"mean", "median", "mode", "std", "sdev", "variance", "moment", "argmin", "argmax", "min", "max", "lim", "arg", "deg", "dim", "Harr", "Jac", "sgn", "sigmoid",
Trigonometric Function	"sin", "cos", "tan", "cot", "sec", "csc"

APPENDIX F

HISTOGRAMS OF MATHEMATICAL OBJECT PROPERTIES

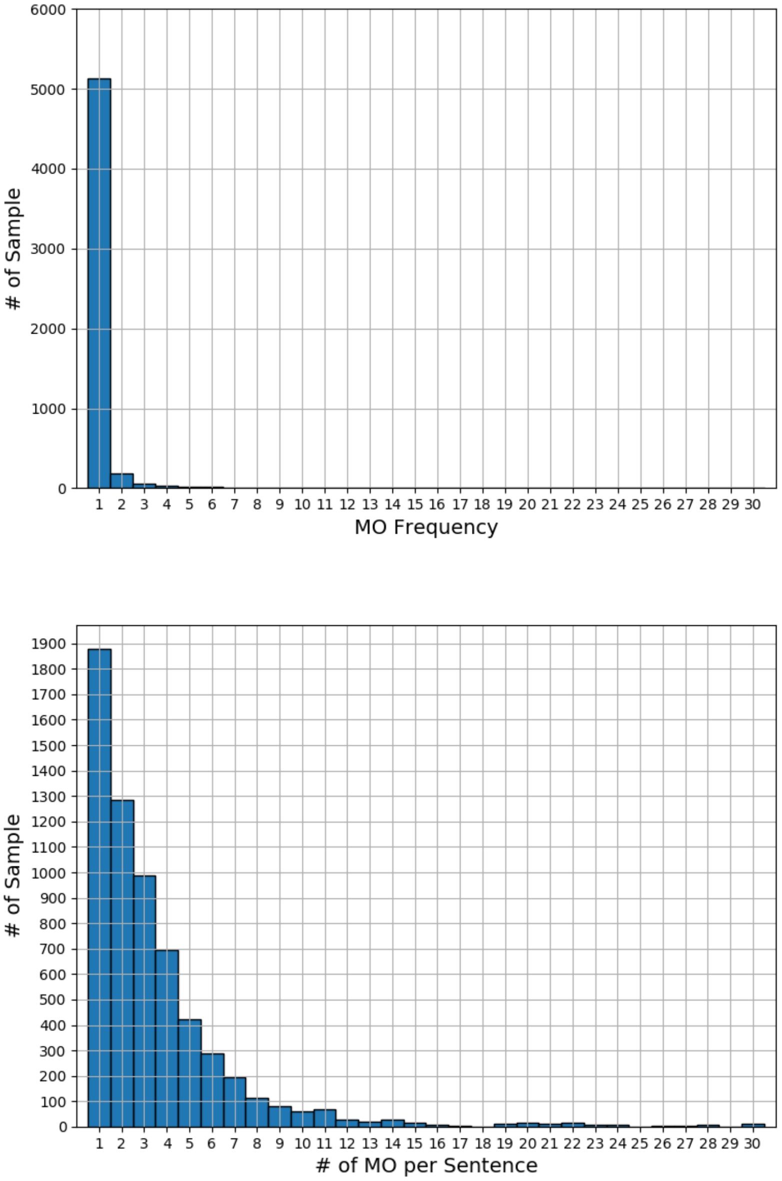


Figure 55: The histograms of the frequency of MOs and the MO density sampled from 50 documents of the KDD Cup 2003 dataset

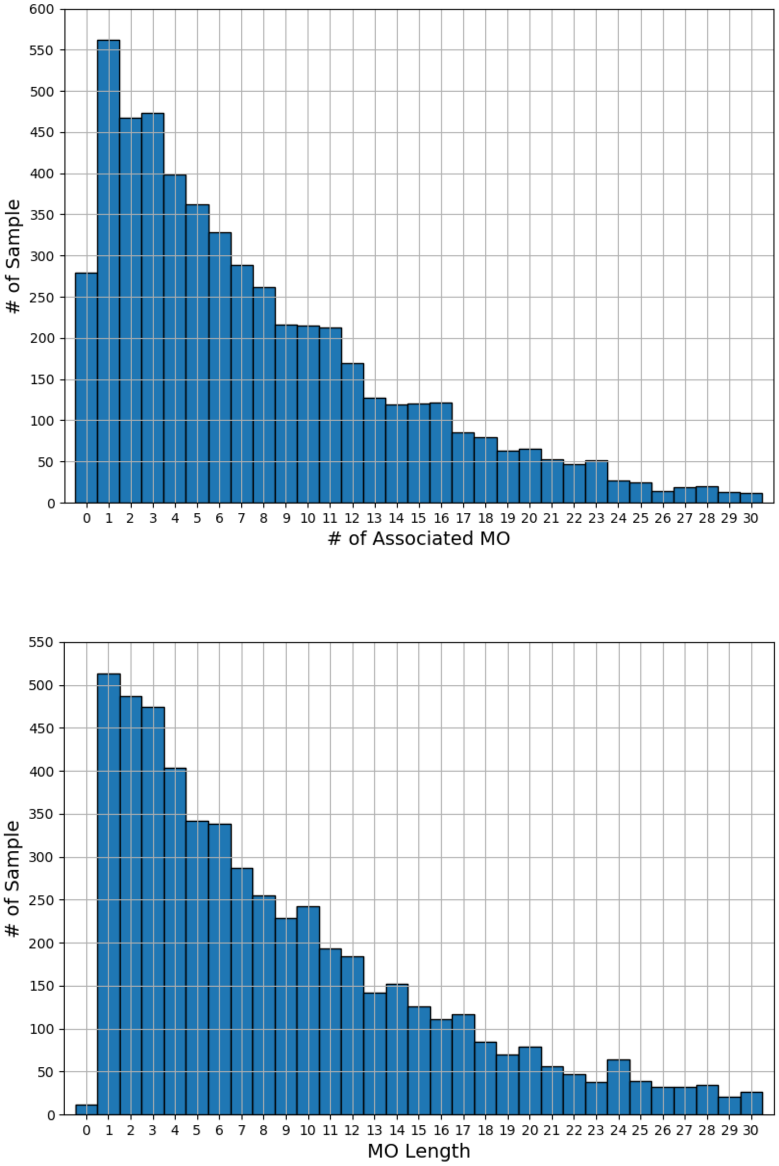


Figure 56: The histograms of the number of associated MOs and the MO length sampled from 50 documents of the KDD Cup 2003 dataset