# MECA: MATHEMATICAL EXPRESSION BASED POST PUBLICATION CONTENT ANALYSIS

A Dissertation

by

XING WANG

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 (Steve) Liu |
| Committee Members, | Thomas R. Ioerger |
| | Ruihong Huang |
| | Nick Duffield |
| Head of Department, | Dilma Da Silva |

December 2018

Major Subject: Computer Science

# ABSTRACT

Mathematical expressions (ME) are critical abstractions for technical publications. While the sheer volume of technical publications grows in time, few ME centric applications have been developed due to the steep gap between the typesetting data in post-publication digital documents and the high-level technical semantics. With the acceleration of the technical publications every year, word-based information analysis technologies are inadequate to enable users in discovery, organizing, and interrelating technical work efficiently and effectively.

This dissertation presents a modeling framework and the associated algorithms, called the *mathematical-centered post-publication content analysis* (MECA) system to address several critical issues to build a layered solution architecture for recovery of high-level technical information. Overall, MECA is consisted of four layers of modeling work, starting from the extraction of MEs from Portable Document Format (PDF) files. Specifically, a weakly-supervised sequential typesetting Bayesian model is developed by using a concise font-value based feature space for Bayesian inference of ME vs. words for the rendering units separated by space. A Markov Random Field (MRF) model is designed to merge and correct the MEs identified from the rendering units, which are otherwise prone to fragmentation of large MEs.

At the next layer, MECA aims at the recovery of ME semantics. The first step is the ME layout analysis to disambiguate layout structures based on a Content-Constrained Spatial (CCS) global inference model to overcome local errors. It achieves high accuracy at low computing cost by a parametric lognormal model for the feature distribution of

typographic systems. The ME layout is parsed into ME semantics with a three-phase processing workflow to overcome a variety of semantic ambiguities. In the first phase, the ME layout is linearized into a token sequence, upon which the abstract syntax tree (AST) is constructed in the second phase using probabilistic context-free grammar. Tree rewriting will transform the AST into ME objects in the third phase.

Built upon the two layers of ME extraction and semantics modeling work, next we explore one of the bonding relationships between words and MEs: ME declarations, where the words and MEs are respectively the qualitative and quantitative (QuQn) descriptors of technical concepts. Conventional low-level PoS tagging and parsing tools have poor performance in the processing of this type of mixed word-ME (MWM) sentences. As such, we develop an MWM processing toolkit. A semi-automated weakly-supervised framework is employed for mining of declaration templates from a large amount of unlabeled data so that the templates can be used for the detection of ME declarations.

On the basis of the three low-level content extraction and prediction solutions, the MECA system can extract MEs, interpret their mathematical semantics, and identify their bonding declaration words. By analyzing the dependency among these elements in a paper, we can construct a QuQn map, which essentially represents the reasoning flow of a paper. Three case studies are conducted for QuQn map applications: differential content comparison of papers, publication trend generation, and interactive mathematical learning. Outcomes from these studies suggest that MECA is a highly practical content analysis technology based on a theoretically sound framework. Much more can be expanded and improved upon for the next generation of deep content analysis solutions.

# DEDICATION

To my family.

ACKNOWLEDGEMENTS

I want to thank my committee chair, Dr. Liu, and my committee members, Dr. Ioerger, Dr. Huang, and Dr. Duffield, for their guidance and support throughout this research.

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

Thanks to Drs. Luciana Barroso, Mary Margaret Capraro, Robert M. Capraro, and AggieSTEM team for providing the data used for this study.

Finally, thanks my wife and daughter for her patience and love and my mother and father for their encouragement.

## CONTRIBUTORS AND FUNDING SOURCES

The outcome of this dissertation is a result of significant teamwork. This work was initiated and shaped with supports from my advisor Professor Jyh-Charn Liu (advisor). Jason Lin and Ryan Vrecenar contributed to the annotation of LaTex annotation for ME in the Elsevier dataset for first version ME-PoS tagging. Ryan Vrecenar helped develop the prototype of the QuQn map based on the sub-string matching of ME LaTex. Donald Beyette made a significant contribution to the frontend interface development for the QuQn map. The case study for interactive learning experience based on QuQn map is in collaboration with the AggieSTEM program, including Drs. Robert Capraro, Mary Capraro, Luciana Barroso, and a graduate student Michael Rugh.

There are no funding sources related to this dissertation.

# NOMENCLATURE

| | |
|---|---|
| AST | Abstract Syntax Tree |
| Bbox | Bounding Box |
| CCS | Content-constrained Spatial Model |
| CMML | Content MathML |
| CRF | Conditional Random Field |
| EG | Essence Graph |
| EME | Embedded Mathematical Expression |
| FP | False Positive |
| FN | False Negative |
| GD | Ground Truth |
| GUI | Graphical User Interface |
| HMM | Hidden Markov Model |
| HOR | Horizontal |
| HR | Height Ratio |
| HTML | Hypertext Markup Language |
| IME | Isolated Mathematical Expression |
| LSA | Latent Semantic Analysis |
| LDA | Latent Dirichlet Analysis |
| MAG | Microsoft Academic Graph |
| MathML | Mathematical Markup Language |
| MDS | Multi-Dimensional Scaling |

| | |
|---|---|
| ME | Mathematical Expression |
| MECA | Mathematical Expression centered publications Content Analysis |
| MIP | Mixed Integer Programming |
| MIR | Mathematical Information Retrieval |
| MRF | Markov Random Field |
| MWM | Mixed Word-ME |
| NLP | Natural Language Processing |
| NME | Non- Mathematical Expression |
| NML | Noun modifier |
| NP | Noun Phrase |
| NPMI | Normalized Pointwise Mutual Information |
| NSCS | Non-Separable Character Sequence |
| NTCIR | NII Testbeds and Community for Information access Research |
| NVCD | Normalized Vertical Center Difference |
| OCR | Optical Character Recognition |
| PCFG | Probabilistic Context Free Grammar |
| PDF | Portable Document Format |
| PHN | Parametric modeling of the HR and NVCD |
| PoS | Part-of-Speech |
| PG | Percentage Gain |
| PMML | Presentational MathML |
| PPC | Projective Profiling Cutting |
| PPES | Pre-Post Effect Size |

| | |
|---|---|
| QuQn | Qualitative-Quantitative |
| RC | Relation Chain |
| Regex | Regular Expression |
| RL | Reference Line |
| RU | Rendering units |
| REV | Reversed |
| RSBS | Relative Size and Baseline shift for super-/sub-script |
| SAWS | Semi-Automated Weakly-Supervised |
| SQL | Structured Query Language |
| STEM | Science, Technology, Engineering, and Mathematics |
| SUP | Superscript |
| SUB | Subscript |
| SVM | Support Vector Machine |
| TF*IDF | Term frequency, inversed document frequency |
| TN | True Negative |
| TP | True Positive |
| TSB | TypeSetting-based Bayesian Model |
| TFIDF | Term Frequency - Inverse Document Frequency |
| VSM | Vector Space Model |
| XML | eXtensible Markup Language |

TABLE OF CONTENTS

Page

ABSTRACT ..... ii

DEDICATION ..... iv

ACKNOWLEDGEMENTS ..... v

CONTRIBUTORS AND FUNDING SOURCES ..... vi

NOMENCLATURE ..... vii

TABLE OF CONTENTS ..... x

LIST OF FIGURES ..... xiv

LIST OF TABLES ..... xviii

CHAPTER I INTRODUCTION ..... 1

I.1 Background on the importance of mathematics, publishing, and automation ..... 1
I.2 State of the arts and challenges in the automated processing of mathematical documents ..... 6
I.2.1 Digital document analysis ..... 6
I.2.2 ME extraction ..... 9
I.2.3 ME layout analysis ..... 11
I.2.4 ME semantics analysis ..... 13
I.2.5 Declaration extraction ..... 16
I.2.6 High-level Application of Mathematical Analysis ..... 19
I.3 Overview of the dissertation ..... 23

CHAPTER II ME EXTRACTION FROM PDF FILES ..... 27

II.1 Overview of the chapter ..... 27
II.2 Document layout analysis ..... 29
II.2.1 Document layout model ..... 29
II.2.2 Line-Column Generator ..... 31
II.3 Typesetting-based Bayesian model for ME extraction ..... 33
II.3.1 Heuristic rules to identify ME/NME characters and their quality ..... 34
II.3.2 Bayesian Inference for EME identification ..... 36
II.4 An MRF-based sequential modeling for EME extraction ..... 38
II.4.1 Problem formulation of MRF-TSB model ..... 40
II.4.2 How MRF-TSB model works and the parameter setting ..... 41

II.4.3 Solver design ..... 43
II.5 Performance and analysis for ME extraction ..... 43
II.5.1 Dataset and evaluation criteria ..... 43
II.5.2 Experiment settings for comparison CRF based method ..... 45
II.5.3 Performance ..... 45
II.5.4 Case study for CRF model to show its drawback ..... 47
II.5.5 Computational cost ..... 48
II.6 Conclusion ..... 49

CHAPTER III CONTENT CONSTRAINED SPATIAL MODEL FOR ME LAYOUT ANALYSIS ..... 50

III.1 Overview of the chapter for ME layout analysis ..... 50
III.2 Typographic System ..... 55
III.2.1 Typographic lines ..... 55
III.2.2 Categorization of characters and recovery of the normalized height ..... 56
III.3 Hierarchical ME layout taxonomy ..... 58
III.3.1 ME layout taxonomy ..... 59
III.3.2 Common interface for ME layout blocks ..... 62
III.4 Two-phase ME layout analysis architecture ..... 63
III.5 Rule-based MEBlock identification ..... 66
III.5.1 Accent Structure ..... 66
III.5.2 Pre-merging of consecutive alphabets on the same baseline ..... 68
III.5.3 Fraction ..... 69
III.5.4 Big operator structure ..... 70
III.5.5 Fence, matrix, piecewise processing ..... 71
III.5.6 Element with both superscript and subscript ..... 71
III.5.7 General Upper/Under ..... 72
III.5.8 Performance of non-horizontal structure identification ..... 73
III.6 Global inference for super/subscript resolution ..... 75
III.6.1 Searching Space Enumeration ..... 76
III.6.2 Global probabilistic inference and features ..... 80
III.6.3 Parametric modeling of Height ratio and Normalized vertical center difference (PHN) ..... 82
III.7 ME layout analysis experiment and results ..... 90
III.7.1 Dataset and evaluation criteria ..... 90
III.7.2 ME layout prediction evaluation at the character level ..... 92
III.7.3 ME layout prediction evaluation by exact matching ..... 93
III.7.4 ME layout prediction evaluation by EMERS tree edit distance ..... 93
III.7.5 Post Checking ..... 95
III.7.6 Execution Speed ..... 96
III.8 Conclusion of CCS-PHN model ..... 97

CHAPTER IV ME SEMANTICS ANALYSIS ..... 98

IV.1 Overview of the chapter ..... 98

IV.2 ME semantic taxonomy ..... 99
IV.3 The parsing algorithm and ambiguities resolution ..... 102
IV.3.1 Consecutive alphabets ambiguity resolution ..... 103
IV.3.2 Probabilistic Context Free Grammar ..... 104
IV.3.3 ME objects generation ..... 110
IV.4 Experiment and Result Analysis ..... 111
IV.4.1 Dataset collection ..... 111
IV.4.2 Evaluation criteria and ME semantics recognition performance ..... 113
IV.5 Conclusion for ME-semantics parsing ..... 114

CHAPTER V DECLARATION EXTRACTION AND MIXED WORD-ME PROCESSING ..... 116

V.1 Overview of the chapter ..... 116
V.2 Mixed word-ME processing ..... 118
V.2.1 New ME-PoS tag and In-sufficiency of existing NLP toolkit ..... 118
V.2.2 MWM PoS tagger ..... 119
V.3 Declaration extraction system description ..... 120
V.4 Weakly-supervised learning ..... 123
V.5 Experiment Result and Analysis ..... 127
V.5.1 Dataset and evaluation criteria ..... 127
V.5.3 Experiment design ..... 127
V.5.3 Result and Analysis ..... 128
V.6 Conclusion ..... 130

CHAPTER VI QUALITATIVE-QUANTITATIVE MAPPING OF SCIENTIFIC PUBLICATIONS ..... 131

VI.1 Overview of the chapter ..... 131
VI.2 QuQn abstraction and essence graph construction ..... 133
VI.2.1 QuQn abstraction construction ..... 133
VI.2.2 essence graph construction ..... 136
VI.3 Essence graph visualization ..... 138
VI.4 Summary ..... 140

CHAPTER VII APPLICATIONS OF MECA ..... 141

VII.1 User study in AggieSTEM summer camp ..... 141
VII.1.1 Experiment settings ..... 142
VII.1.2 Questionnaire and evaluation metrics ..... 143
VII.1.3 Quantitative results ..... 144
VII.2 Knowledge mapping and evolution analysis ..... 145
VII.2.1 Declaration-based document clustering ..... 146
VII.2.2 Evolution visualization and analysis ..... 148
VII.2.3 Weak citation linked analysis based on declaration similarity ..... 150
VII.3 Differential publication analysis by QuQn map ..... 152

VII.4 Summary of MECA applications ..... 153

CHAPTER VIII CONCLUSION ..... 154

VIII.1 Summary of research findings ..... 154
VIII.2 Lessons learned and implication for future work ..... 158

REFERENCES ..... 160

APPENDIX A PCFG PRODUCTOIN RULES FOR ME SEMANTICS PARSING ..... 180

A.1 A complete table of the terminal tokens. ..... 180
A.2 A complete list of the production rules ..... 182
A.2.1 Digits ..... 182
A.2.2 Algebra ..... 183
A.2.3 Binding operators ..... 185
A.2.4 Relation ..... 186
A.2.5 Spatial layout ..... 186
A.2.6 Calculus ..... 187
A.2.7 Functions ..... 187
A.2.8 Probability ..... 188
A.2.9 Set ..... 189
A.2.10 Units ..... 190
A.2.11 Incomplete ..... 191

APPENDIX B. MANUAL FILTERING FOR DECLARATION PATTERNS COLLECTION ..... 192

B.1 PoS Table ..... 192
B.2 Manual filter process ..... 193

APPENDIX C. MECA SOFTWARE SYSTEM ..... 195

LIST OF FIGURES

Page

Figure 1 Trends and driving force for the booming of academic publishing ..... 1

Figure 2 Publication Statistics ..... 2

Figure 3 The writing-reading process: conceptual graph model <=> sequential paper, illustrated with the event co-reference resolution paper, parts of this figure are adopted from [11] ..... 3

Figure 4 Research framework of MECA ..... 4

Figure 5. Research Scope for MECA: Analytical framework ..... 21

Figure 6 Research scope for MECA: major software architecture and workflow ..... 22

Figure 7 Document layout analysis from the Typesetting and font resources of PDF file, parts of this figure are adapted from "Lecture Notes 11: The Good-Turing Estimation" [98] ..... 29

Figure 8 Column Detection Illustration, parts of this figure are adopted from paper 10.1.1.58.6850 from CiteseerX [99] ..... 32

Figure 9 Heuristics to identify partial ME/NME, parts of this figure are adapted from the "residual transfer networks [100]" ..... 34

Figure 10 The workflow for the Typesetting-based Bayesian model, reprinted with permission from[102] ..... 37

Figure 11 The motivation for sequential tagging and the related posterior probability $\boldsymbol{log(P(ME|ni))}$ and $\boldsymbol{log(P(NME|ni))}$, parts of this figure are adapted from 10.1.1.6.2281_9 in Marmot dataset [24] ..... 39

Figure 12 Statistics of the negative log-likelihood ratio for "10.1.1.6.2281_9 [103]" ..... 42

Figure 13 The criteria for the performance evaluation of ME Extraction ..... 44

Figure 14 Example to show the fallacy of the CRF model, parts of this figure are adapted from 10.1.1.6.2308_3 in Marmot dataset [24] ..... 47

Figure 15 The feature weight for CRF based EME identification ..... 48

Figure 16 Example of ME layout ..... 50

Figure 17 The challenge from the glyph of the characters ..... 51

Figure 18 Degraded discrimination performance ..... 52

Figure 19 Examples to illustrate the ME layout and challenges, parts of the figure are adapted from Infty-CDB [47]..... 53

Figure 20 The typographic reference lines ..... 55

Figure 21 Recover the ascender/descender line and the normalized height for height stable or width stable characters..... 57

Figure 22 Histogram of the vertical adjustment upper/under ratio for “A” and “sum” and horizontal adjustment upper/under ratio for “-“ and “,”..... 58

Figure 23 The taxonomy of ME Layout ..... 60

Figure 24 Illustration of attacher and attached object, parts of this figure are adapted from InftyCDB [47] ..... 62

Figure 25 Two-phase ME layout analysis architecture, parts of this figure are adapted with permission from [108] ..... 64

Figure 26 Merging alphabetic MEHorBlocks after the accent analysis, parts of this figure are adapted from ME 28016825 in InftyCDB [47] ..... 65

Figure 27 Illustration of the iterative accent structure analysis, parts of this figure are adapted from ME 28008501 of InftyCDB [47]..... 67

Figure 28 Accent structure processing..... 67

Figure 29 Iterative expanding procedure ..... 67

Figure 30 The tradeoff between the precision and recall for centerline-based analysis ..... 68

Figure 31 Fraction structure processing..... 69

Figure 32 Example of big operator structures, parts of this figure are adapted from InftyCDB [47] ..... 70

Figure 33 The semantics related with upper/under structure, parts of this figure are adapted from InftyCDB [47]..... 72

Figure 34 General upper/under structure detection ..... 73

Figure 35 Cases study for the rule-based MEBlocks identification, parts of this figure are adapted from InftyCDB [47] ..... 74

Figure 36 Intermediate results after the rule-base ME layout structure analysis ..... 75

Figure 37 Four axioms about the layout of horizontally arranged ME blocks, reprinted with permission from [108] ..... 76

Figure 38 Enumeration of the ME layout for horizontally arrange blocks ..... 78

Figure 39 Horizontal layout candidate enumeration ..... 78

Figure 40 Features for inference of super/subscript ..... 80

Figure 41 Feature distribution for different relations with/out filtering ..... 81

Figure 42 Non-parametric estimation of the likelihood, reprinted with permission from [108] ..... 83

Figure 43 Relative sizing and baseline shifting of super/subscript ..... 83

Figure 44 Outline for the inference process for the PHN model ..... 86

Figure 45 The likelihood for each relation chain in HR&NVCD joint space ..... 90

Figure 46 The relationship among ME symbols in InftyCDB ..... 90

Figure 47 Example to show fallacy to evaluate using edit distance of LaTeX ..... 92

Figure 48 The performance of F1 score vs. the number of characters in MEs ..... 93

Figure 49 Example of the flexibility of the presentational MathML ..... 94

Figure 50 ME-level evaluation using MathML representation ..... 94

Figure 51 Speed performance comparison ..... 97

Figure 52 The ME semantics taxonomy ..... 100

Figure 53 Conversion among different standard ..... 101

Figure 54 ME semantics parsing pipeline ..... 102

Figure 55 The interface for the annotation of the ME semantics in STeX ..... 112

Figure 56 Three-phase framework ..... 116

Figure 57 The error propagation from PoS tagging to constituent parsing ..... 119

Figure 58 The system architecture for declaration extraction ..... 121

Figure 59 Semi-automated weakly-supervised process for declaration pattern extraction 123

Figure 60 Statistics of the position of the declaration relative to the ME..... 123

Figure 61 QuQn map architecture, reprinted with permission from [120]..... 132

Figure 62 Examples to illustrate ME decomposition, parts of this figure are adapted from LDA2Vec [122]..... 134

Figure 63 Colored visualization of a (cropped) essence graph pruned from its raw QuQn map, parts of this figure are adapted from paper LDA2Vec [122] ..... 137

Figure 64 Graphical user interface (GUI) for the QuQn visualization and interaction, parts of this figure are adapted from the paper LDA2Vec [122] ..... 139

Figure 65 The final QuQn map created by the teacher, parts of this figure are adapted from [124] ..... 142

Figure 66 Pipeline for declaration-based topic mining..... 146

Figure 67 Full document-based clustering vs. Declaration-based clustering ..... 148

Figure 68 Evolution visualization for the Neural Information Processing System (NIPS) conference papers (2013-2017)..... 149

Figure 69 Weak citations ..... 151

Figure 70 Illustration of differential publication analysis, parts of this figure are adopted from [141] and [142]..... 152

LIST OF TABLES

Page

Table 1 The rules to match document structures ..... 34

Table 2 Performance of the Heuristics to identify ME/NME ..... 35

Table 3 Objective value table for the case [“[”, “1”, “,”, ”T]”] ..... 41

Table 4 Objective value table for the case [“that”, “w”] ..... 41

Table 5 Coarse performance statistics for EME detection ..... 46

Table 6 Detail Performance statistics for EME detection ..... 47

Table 7 Glyph types and categorized of the characters ..... 56

Table 8 Illustration of ME blocks and the baseline character, parts of this table are adapted from InftyCDB [47] ..... 61

Table 9 Performance of Non-horizontal structure analysis ..... 73

Table 10 Parameter estimation of the RSBS ..... 85

Table 11 ME-level evaluation using the EMERS edit distance on MathML ..... 94

Table 12 NPMI score & frequency for the AggieSTEM test case ..... 104

Table 13 Terminal tokens of the PCFG for ME semantic parsing ..... 105

Table 14 Internal states of the PCFG for ME semantic parsing ..... 107

Table 15 Internal states of the PCFG for ME semantic parsing (Cont.) ..... 108

Table 16 Production rules and the probability ..... 109

Table 17 Extended LaTeX tags to annotate the ME semantics ..... 112

Table 18 The performance for ME semantic analysis ..... 113

Table 19 PoS for ME and examples ..... 118

Table 20 TnT-based PoS tagging prediction performance ..... 120

Table 21 Features for declaration extraction ..... 121
Table 22 Patterns of declaration ..... 122
Table 23 Trivial patterns collected simply by frequency..... 125
Table 24 Manual intervention for declaration pattern extraction, round 1 ..... 125
Table 25 Manually constructed patterns from mined skip-bi-gram..... 126
Table 26 Short declaration extraction performance..... 128
Table 27 Long declaration extraction performance ..... 129
Table 28 Mean and Std. statistics for the pre/post ..... 144
Table 29 Comparison of pre-post ..... 145
Table 30 PoS filtering for declaration pattern collection..... 192
Table 31 Manual pattern filtering, round 2 ..... 193
Table 32 Manual pattern filtering, round 3 ..... 194

CHAPTER I

INTRODUCTION

## I.1 Background on the importance of mathematics, publishing, and automation

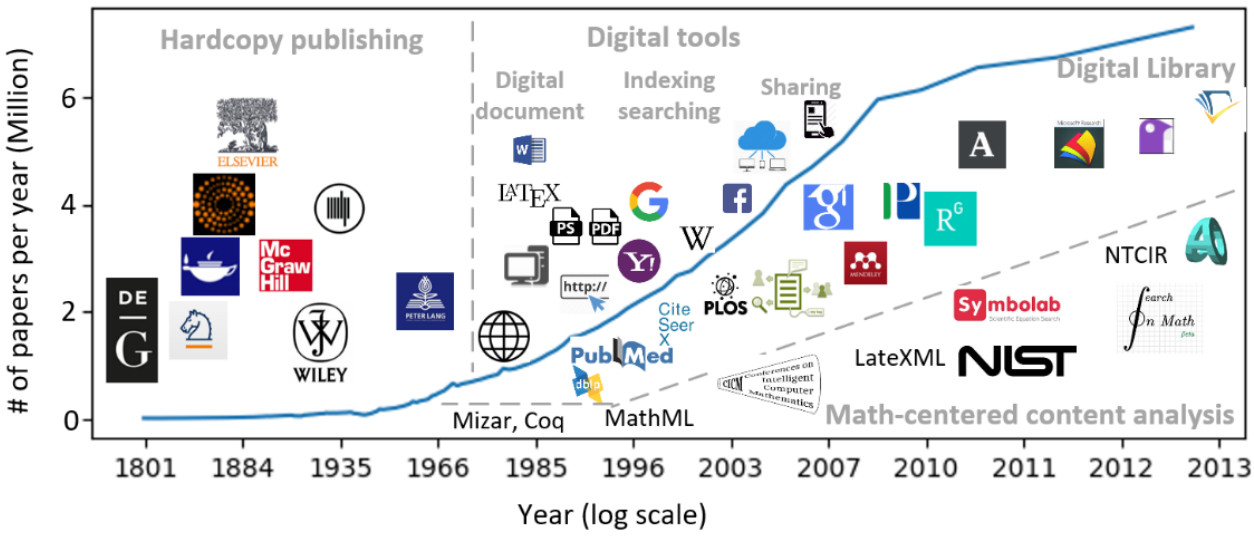
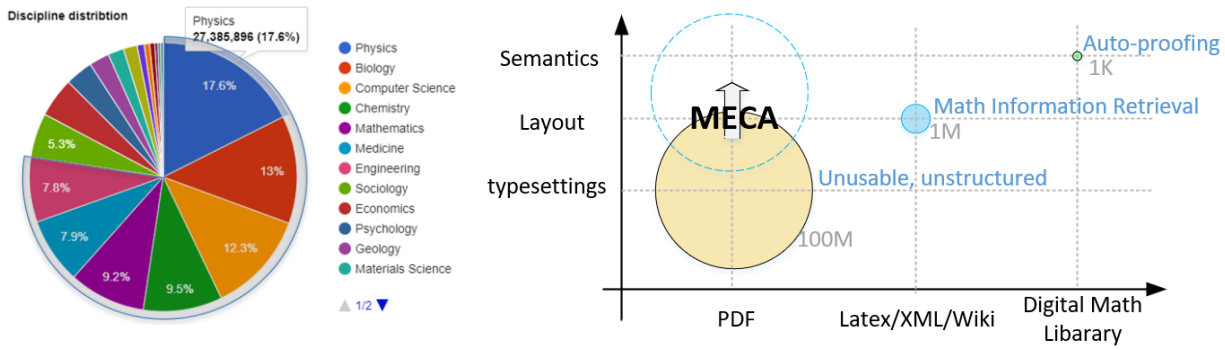



**Figure 1 Trends and driving force for the booming of academic publishing**

Empowered by information technology, the publishing industry has experienced an exponential accumulation of knowledge in the past few decades as shown in Figure 1. Authoring tools and the Internet allow authors to produce sophisticated content and publish/transmit conveniently. Information technologies such as search engines and automated citation extraction tools lead to very large-scale digital library systems for the indexing and searching of intellectual work. The existing systems are mostly based on the plaintext words. However, the large number of mathematical expressions (MEs) are less studied with few successful applications.




(a) Publication ratio by disciplines

(b) The scale of publications in different formats

**Figure 2 Publication Statistics**

Though there are a large number of mathematical contents, they are mostly unstructured and could not be processed by automated computer algorithms. Statistics in Figure 2.a from Microsoft Academic Graph (MAG) [1] show that over three-quarters of the publications are from the Science, Technology, Engineering, and Mathematics (STEM) domains. In STEM, mathematical expressions (MEs) are widely adopted, because they provide a standard medium for formalizing, exchanging, and accumulating knowledge concisely and efficiently. Even though MEs are composed using alphanumerical and special symbols, their content analysis does not enjoy the same level of automation as that of the plaintext-based content. The existing efforts for ME analysis primarily covers two aspects: formal symbolic computing such as auto-proofing systems [2], [3] and formula search engines [4], [5], [6], [7], [8]. The inputs are in a structured format such as presentational and content MathML [9]. However, as shown in Figure 2.b, only 1.4M files in Latex/XML format [10] are annotated for their ME content in a structured format. Over 100M articles are only available in Portable Document Format (PDF), where the MEs are not explicitly marked, and their layout and semantic structure are unavailable. ME-based data

models and their processing algorithms are highly valuable for researchers to facilitate access to the vast number of technical articles in STEM (about 79 million (M) out of 127M in MAG).

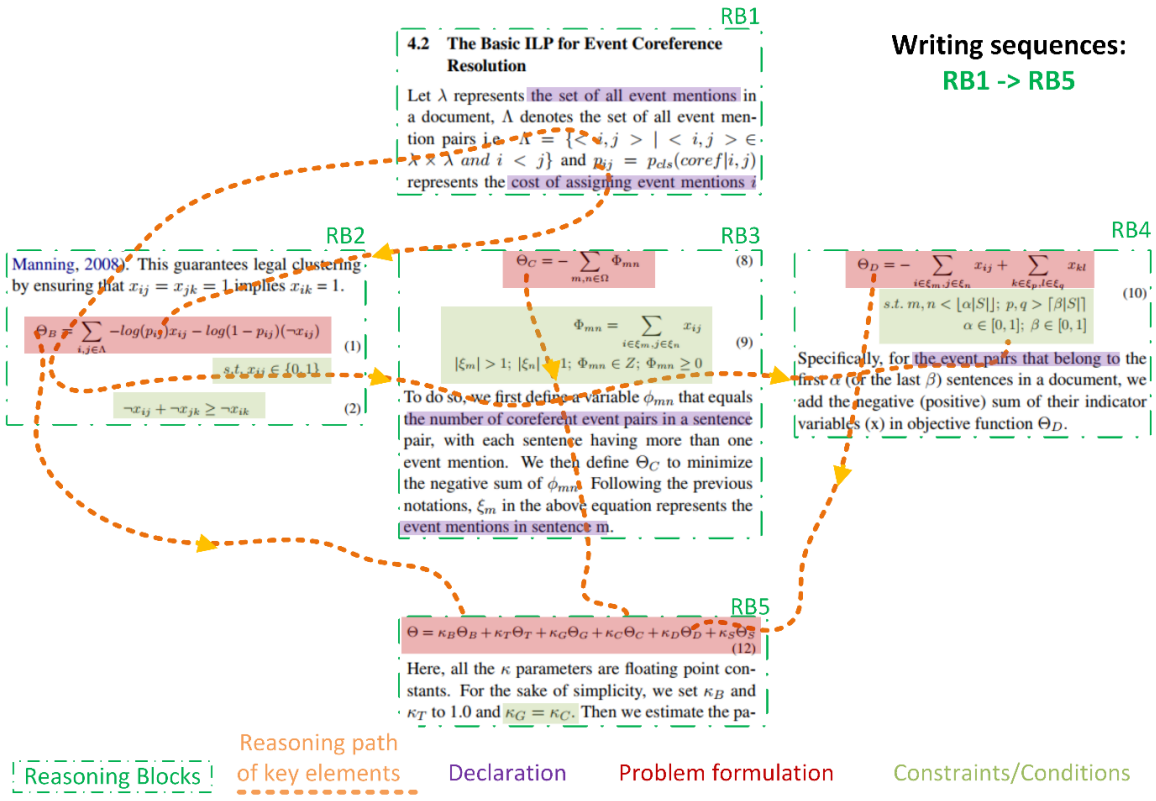



**Figure 3 The writing-reading process: conceptual graph model <=> sequential paper, illustrated with the event co-reference resolution paper, parts of this figure are adopted from [11]**

Besides the extraction and analysis of the ME content, the neighbor words also play important roles in enhancing the semantics of MEs and connecting the logic among MEs. Together, the ME and words serve as the bridge to connect the writers and authors. Technical writing can be characterized as a collaborative divide-and-conquer process between authors and readers. Authors divide the complex technical concepts into a sequence of self-contained yet

interconnected reasoning blocks (RB), and readers conquer the RBs in the reading process to rebuild the technical concept. Authors and readers rely on a combination of community-specific *technical dialects* ("jargon," "terminology") and MEs, as well as lower level gluing words, to ensure the correct understanding of the substance.

It is challenging to recover high-level semantics of technical materials from low-level digit files using computer algorithms, which involve PDF parsing, document layout analysis, ME layout/semantics parsing, and mixed word-ME mining. First, the typesetting information in PDF files is transformed into a layout structure such as columns and lines and grouped into logical structures such as paragraphs and MEs. For the MEs, their semantics will be understood through layout and semantic analysis. Further, the external meaning of MEs is recovered through the bonding with words. Finally, the logic flow should be discovered through dependency analysis at the semantic level as the technical essence, which will be the basis for high-level applications.

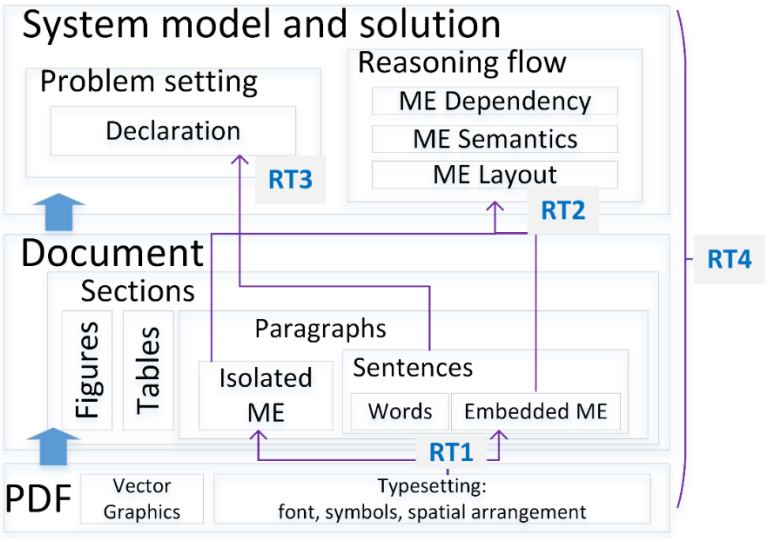



**Figure 4 Research framework of MECA**

A long chain of solutions is required to recover high-level semantics of technical materials from low-level digit files using computer algorithms. In this dissertation, a Mathematical Expression Content Analysis (MECA) system is proposed to support layered ME content extraction and recovery of their semantics. MECA is organized into the following layers: ME extraction, ME layout/semantics parsing, and mixed word-ME (MWM) mining. As shown in Figure 4, the dissertation is organized into five research tasks (RT), RT1-RT4 to implement and test modeling work and their associated algorithms for these layers. In RT1, the typesetting information in PDF files is transformed into layout structure such as columns and lines and grouped into logical structures such as paragraphs and MEs. In RT2, the ME semantics is recovered through layout analysis and then represented by an ME semantics taxonomy data structure. RT3 focuses on the prediction of the bonding between MEs and their word based declarations. RT4 uses the outputs generated from RT1-R3 to recover the reasoning flow of a paper based on the dependency analysis of declarations (qualitative descriptors) of MEs (quantitative descriptors). And the result is a novel abstraction called the QuQn map[1] to represent the technical essence of a paper. At last, three different case studies were conducted to validate the usefulness of QuQn graphs for real-world applications. The first use case is for supporting the mathematical learning of a high school summer camp. The second use case is for publication trend analysis based on the ME declarations. The last use case is for differential content analysis of technical papers.

[1]QuQn graph is an alternative name.

## I.2 State of the arts and challenges in the automated processing of mathematical documents

As the MECA analytical framework in Figure 4 shows, the MECA system involves multiple components ranging from document processing, layout analysis, semantic analysis, natural language processing, and high-level math-centered applications. The related work and open challenges will be introduced individually in the following sections.

### *I.2.1 Digital document analysis*

#### I.2.1.1 State of the arts of digital document analysis

Digital files are mostly designed for the ease of editing and dissemination. There is a trend of machine-readable publishing [12], and open document standard supporting semantic tags such as Office Open XML [13]. But PDF [14] is still the de facto standard for publishing. PDF files only contain a sequence of rendering units (RU) containing the typesetting information. Digital document analysis aims at recovering the document layout and logical structure from the rendering units [15].

The document layout structures refer to the hierarchy of documents, including pages, columns, lines, and tokens separated by space. There is no one-to-one correspondence between RUs and layout structures. One token might be split into multiple RUs. The RUs could be merged into higher-level layout elements based on overlapping in a bottom-up fashion or a top-down split based on Projection Profiling Cutting (PPC) could be applied to identify the high-level structures such as column first. The PPC technique is a widely adopted technique for document layout analysis [16] and mathematical analysis [17]. The PPC works by projecting the pixels or shapes onto either the vertical or the horizontal direction and detect the change of element distribution for the segmentation boundary. Regardless of processing in a bottom-up or a top-down fashion, it is crucial to obtain the exact position of each character.

It has already been observed and verified that the raw bounding box (bbox) of the characters read from the PDF file is not accurate for all the existing PDF processing toolkits, including PDFMiner [18], PDFBox [19], Multivalent [20]. Additional processing is required to account for the extra space and shifting of the bounding box for the big operators. Baker [21] tried to overcome this problem by matching the bbox with the pixels. However, this is computationally costly and cannot resolve the shifting of the parsed bbox comparing with the perceived bbox. TextStripper [22] from PDFBox is the best at the recovery of the tight bounding box. Further, the accurate bbox is also crucial to improve the discrimination ability of features for ME layout analysis. Finally, some big math characters such as the fence for matrices are composed of multiple glyphs, and additional pre-merging is necessary [21].

In addition to the accurate position estimation of each character, layout analysis gets even more challenging for documents containing complex MEs due to the two-dimensional nature of ME, where one ME is commonly split into multiple rendering blocks and might split into multiple vertical ranges. It has been previously observed that the quality of text line segmentation has a direct impact on the performance of Isolated ME (IME) detection [23]. Special processing is needed to merge the accent and under/upper parts of big operators, where the semantic information of the character values is required. Normalization of the character values in PDF is highly desired as the value might be ASCII, Unicode, or manifested as the glyph name in the font resources.

On the other side, the logical structures refer to semantic meaning segments, such as title, header, paragraph, figure, table, sentence, word and ME. The target of this dissertation, ME, will be elaborated in the next section. Only the identification of other logical structures will be introduced here, which also play an important role in filtering out negative candidates. Similar to

the discrepancy between rendering units and physical layout, one major challenge for logical structure analysis is the discrepancy between physical layout units and logical structures. A figure might be composed of multiple vector graphics, and an ME might be separated into multiple layout units, causing partial matching and over matching issues in existing ME Extraction systems [24]. Due to their simple layout structure, the heading information (title, authors, and abstract) and references/citations are accurately extracted based on the conditional random field (CRF) [25]. The Parcit system [26] has reached production level and to be deployed in digital libraries such as Citeseerx [27], Google Scholar, Microsoft academic search [1], semantic scholar. The figure/table [28] and their caption/reference metadata [29] could also be extracted by regular expression (regex) patterns.

Most existing PDF analysis tools do not have special processing for MEs, including the official Adobe Acrobat DC and the Phantom PDF editor from Foxit. Maxtract [30] is the first and only attempt to convert PDF to Latex, which could be considered recovery at the logical structure level. However, Maxtract has limited applicability as it only uses the font to discriminate between ME and words.

Though publications in markup languages are much less in comparison to PDF, they are still large in quantity. The Arxiv [10] pre-print service hosts about 1.4 million documents as of August 2018, occupying about 1% of all publications based on statistics from the Microsoft Academic Graph [1]. Much more insights about the nature of technical expressions could be discovered even if a fraction of the papers could be analyzed. The Arxiv data has been successfully used in the KDD cup 2003 for citation prediction [31] and the NTCIR mathematical information retrieval task [5]. LaTeX source can be converted to a semantic level by LateXML [32] for both text and MEs with limited accuracy. For transformation among markup languages,

the Pandoc Project [33] is the most active and mature, covering most existing file formats, including Latex and Docx.

#### I.2.1.2 Challenges of digital document processing

In summary, the critical challenges for digital document processing arise from two discrepancies: the discrepancy between the rending units with layout structure and the discrepancy between the layout structures with logical units. Accurate recovery of the physical layout structures and logical structures is the foundation for all later steps. Also, there are two engineering challenges in the normalization of the character information: character value normalization and tight bounding box recovery. The normalized values play crucial roles in the layout and semantical analysis for MEs.

### *I.2.2 ME extraction*

#### I.2.2.1 State of the arts for ME extraction

ME is a particular type of logical structure, which faces the same challenge of the discrepancy between the physical layout and the logical structure as elaborated in the document logical structure analysis and the identification of ME. Additionally, the ME extraction task has its unique properties and challenges.

ME extraction has been studied since 2011 [34], [35], [23], [24]. An ME can be embedded (EME) among plaintexts or isolated ME (IME) from them in a standalone line. The IMEs are easier to detect as they often have formula serial number [34] with distinct layout [36], [34], [23]. Spatial layout features include line height, space above/below, left/right indent [21], line centeredness, the variation of line width [34], the sparseness of chars, the variance of baseline and the bounding box size [37].

EME extraction is still an open problem due to the unrestricted use of fonts and the fuzzy boundary with words caused by the discrepancy between physical layout analysis and logical units. Besides the above-mentioned spatial layout features for IME detection, the following aspects are also explored: 1) math element, 2) fonts, 3) linguistics. Math elements include named functions [34], fraction/radical structure [37], and special characters for operations, relations, Greek, delimiters, integrals, etc. [34], [38]. The italic font and the irregular size are also indicator [37], [39] and [21] also used the particular font name to extract MEs. Linguistics features include the purity of words [37] and letters ratio [40]. Past methods mostly model the EME identification problem as a classification problem using the Support Vector Machine to train the discriminant model. For non-ME (NME) detection, a set of customized regular expressions to detect figure, table and equation references are developed based on [29].

There is a trend of using adaptive features besides the general features mentioned above. To accommodate the writing habits of each user, [40] proposed to use the local features based on the identified isolated mathematical expression (IME). However, the mixed usage of general features and customized features still hinder the correct decision as will be elaborated in this dissertation.

In addition to typesetting, neighbors of MEs may also provide useful detection clues. For example, [35] used the label of neighbors as a feature and [38] used the context as semantic constraints and made an assessment of the relationship between connected characters [38]. Iwatsuki [40] is the only work which systematically models the neighbor information for decision making based on the conditional random field [25].

#### I.2.2.2 Challenges for ME extraction

As a particular type of logical structure, the ME extraction module inherits all the challenges of digital document processing. More specifically, the errors in layout analysis cause partial matching and over-matching issues. The variety of writing habits might violate the assumption of global training, leading to degradation of performance. There is a need for the design of adaptive feature to capture the writer habits.

### *I.2.3 ME layout analysis*

#### I.2.3.1 State-of-the-arts for ME layout analysis

Given the identified MEs, represented as typesetting, i.e., a collection of characters with value, font, and positional information, the ME layout must be recovered to understand the semantics. The existing methods for ME layout analysis can be grouped into divide-and-conquer approaches and integrated methods based on the survey by Chan [41] and Zanibbi [42], [43].

Characters are atomic building units. The character value and bounding box (bbox) are critical information in predicting the ME layout. The bbox must be accurately adjusted to reflect the tight bounding box, as elaborated in the digital document analysis section. Many characteristics only apply to a subset of the characters. First, the value of accent, radical, and binding operators are reliable indicators of possible affiliated children [44], [45]. Second, for alphabets and digits, the baseline can be identified to organize the characters into a recursive structure which is then transformed into ME layout using tree transformation [46]. Besides the baseline, the normalized height, i.e., the distance between the ascender line and the descender line can be recovered for more accurate super/subscript classification [47], [48]. Third, besides the characters mentioned above, there are large quantities of characters remaining, including operators, relations, arrows, etc. Typically, the tight bounding box top/bottom boundary of their

glyphs does not align with the typographic reference lines such as baseline or midline. However, some are vertically asymmetric and have the vertical center estimated reliably for the assessment of vertical relationship [45].

For the divide-and-conquer approach, decision rules for different structures are proposed based on the aforementioned character dominance [45] and the relative spatial position [42]. Since the super/subscript relationships are widely used, many studies focused on them alone. Okamoto [49] used fixed thresholds to search for the SUP/SUB. Aly [48] used relative size and relative position features calculated from normalized bounding box to predict the relation between a pair of alphanumeric character as HOR/SUP/SUB. But alphanumeric characters only cover about 57% of all characters and 26.5% of all pairs of characters in dominance relationship. Ling [50] and Zanibbi [51] proposed features in the log-polar space and PCA is adopted for dimension reduction and improved discriminant ability for layout recovery of hand-written MEs. A similar feature was introduced by Fotini [52] to capture the angle. Generally, if the characters are not correctly processed to recover their normalized height and vertical center, there will a significant overlapping on the distribution of the feature, leading to a degradation of the discrimination ability. The methods mentioned above only apply to each pair of characters locally, but the local decision might introduce error and also lead to inconsistency globally.

Integrated model-based approaches [53], [47] are proposed to overcome local decision error. Wang [53] treated the layout of ME together as the event space, and the dominance relationships of all the characters are inferred simultaneously to reach global optimality. Suzuki [47] formulated the layout identification problem as a minimal cost spanning tree problem. However, the cost/score for each local linkage is set manually, which might not attain the best performance. Alvaro [48] expanded the stochastic context-free grammar and incorporated spatial

relationship assessment into the grammar. Although the incorporation of the semantic grammar brings some benefits, it also limits the applicability because of the difficulty in capturing the flexible representation and customization of ME fonts. Okamoto [49] used projection profiling cutting to produce a hierarchical grouping of symbols, which is then traversed and transformed into a mathematical layout using re-writing rules. The PPC method is sensitive to the overlapping of the characters, and there are no systematical solutions about the order to apply vertical/horizontal cutting, which will affect the final results. Raja [54] adopted graph grammar rewriting over the neighbor graph of symbols by minimizing conflicts.

#### I.2.3.2 Challenges for ME layout analysis

First, the recovery of the hierarchical ME layout faces the ambiguity in create blocks. Further, the identification of the characters on the main baseline is required rather than treating the ME block as a whole unit, and the characters must be normalized concerning the reference lines to precisely assess the relative spatial relation between ME blocks. Second, there are many rules to recover a portion of the ME structures. It is necessary to recover the partial structure in the correct order so that the partial structures do not interfere with each other and can cover all situations. At last, the local greedy decision method suffers from error propagating, but the global inference is computationally costly. Further, given that new layout conventions might be introduced, a generative model is preferred over the discriminative model, since generative models have a clear system boundary.

### *I.2.4 ME semantics analysis*

#### I.2.4.1 State-of-the-arts for ME semantics analysis

The “ME semantics” and the “semantic taxonomy” mentioned in this work are similar to the concepts Operator Tree [42], OpenMath [55] and Content MathML [9]. However, the

operator tree does not adequately express the semantics yet. The superscript might be represented as a character '^' in operator tree, but it can have different meanings, such as superscript, function inverse, exponential, function differentiation. It is also a non-trivial task to convert between different standards [56].

Different ME-semantics parsing systems have different assumptions about input. Some works [57], [58] assume the inputs as images with the need of an OCR module. Some make the assumptions that the layout is correctly recovered and only the semantics is left to resolve [59], [32]. The second approach with a modularized design is adopted in this work, which is also suggested in the survey paper [42].

Early works on ME semantics parsing are mostly rule-based systems. Andrea [60] used a top-down syntax grammar to build the operator tree. This top-down way has the advantage of using the target tag as the context to guide the meaning of the dominated symbols. But the top-down schema has the disadvantages of exponential complexity and could not pinpoint the error when parsing failed. This early work only showed the feasibility with a limited grammar for basic algebra. Similarly, a recursive descent method is adopted with the assumption that the ME is already segmented into meaningful semantic blocks [57].

Graph re-written is another popular rule-based approach [61]. Spatial and content type conditions trigger rules to re-write the graph. These rules are applied to the graph iteratively until the stop conditions are met. One challenge for the graph-based method is the rule selection when multiple rules are satisfied. Lavirotte [59] used the context to make sure there is no ambiguity. One equivalent explanation for adding the context is to enforce the order of execution. Practically, graph search is time-consuming for the subgraph matching.

Miller from NIST developed the LateXML [32] system to convert from Latex into the Content MathML representation using Context Free Grammar. It is the state-of-the-art for ME semantic parsing. However, it has been reported above 41% of the notations did not have their semantic role resolved [62], where the role attribute is set to 'unknown.'

The rule-based parsing mainly uses the context and manually defined rules for the resolution of ambiguity. Another direction for the ambiguity resolution is using stochastic grammar to resolve the ambiguity statistically, where the probability could either be trained using the unsupervised Inside/Outside algorithm [63] or supervised probability estimation from ground truth data [64].

Except for the LaTeXML [32], the works above only cover the basic math concepts. As more math dialects are considered, more ambiguities will be introduced, which is the main challenge for ME semantics understanding. Youssef [65] enumerate five types of ambiguities that might happen during the semantic analysis, which could be grouped into three major categories: tokenization, scoping, and interpretation. The tokenization refers to the process of segmenting an ME into atomic building units such as operators, relations, and identifiers (which might be a single character or multiple characters). As such, there is an ambiguity that the consecutive characters could either mean an identifier or the multiplication of multiple variables. Second, for the convenience of writing, the grouping fences could be omitted, causing various possible ways to interpret the operation order. For the last interpretation layer, one needs to resolve the actual meaning given the same physical layout structure. The accent might mean conjugation for complex number or differentiation of a function. The superscript could indicate an exponent component or an index.

#### I.2.4.2 Challenges for ME semantics analysis

First, to cover a wide spectrum of applicability for different math dialects, a general extendable framework is necessary to add new rules when necessary. Second, during the parsing phase, the semantic analysis faces the ambiguities for tokenization, abstract syntax tree (AST) construction, and the AST interpretation. The tokenization challenges come from two aspects. On the one hand, one character might have multiple meaning, which will lead to entirely different ASTs. On the other hand, the consecutive alphabets might mean multi-character identifiers or multiplication by omitting the operators. Second, the AST might not be correctly recovered. At last, the same structure might also different interpretation. For example, the superscript component could be index or exponents.

### *I.2.5 Declaration extraction*

#### I.2.5.1 State-of-the-arts for declaration extraction

ME-declaration extraction belongs to the domain of information extraction, but it differs from the traditional natural language processing (NLP) due to the elaboration of mixed Word-ME (MWM) sentences. Additional taxonomy and customization are necessary to analyze the syntactic role of ME and its interaction with neighbor plaintext.

First, the ME could be more complicated than simple plaintext words, acting as a sentence or subordinate clause. The existing convention [66] for the part-of-speech (PoS) of ME contains three categories: S for a sentence or subordinate clause, NP for a noun or noun phrase, NML for a noun modifier. None of the existing PoS taggers pays particular attention to the ME. Current works [67], [68], [69] process MWM sentences by treating the MEs as ordinary words and directly apply the existing PoS tagger [70], [71]. The special syntactic role of ME could not be covered, and the degradation of the PoS tagging for other words was also observed. An F1

score of 0.936 is obtained using the Stanford MaxEnt tagger in comparison with 0.96 F1 score non-MWM corpus in our study.

In traditional NLP domain, the PoS tagging task has been considered an almost solved problem using statistical machine learning models. Features are the most critical aspect of machine learning based methods. The standard features for PoS tagging include the value/prefix/suffix of the current token or its neighbors [70]. The machine learning methods that capture the interaction among neighbors also helped improve the performance such as the Tri-gram HMM model [71]. One challenge issue in PoS tagging is the parameter estimation for the out of dictionary words, which is commonly attacked by back off interpolation [71]. As for ME specific PoS tagging, our previous statistical ME-PoS tagging model [72] based on the format complexity of ME, neighbors PoS prediction, and the syntactic properness of the sentence reached an accuracy of 75% for three classes classification of the PoS of the MEs. However, it did not predict the PoS of other words, which is not accurately predicted by existing toolkit because of the ME neighbors.

Due to the particular PoS of ME and the difference in the interaction of ME with plaintext, a traditional constituent or dependency parser will fail to analyze the syntactical structure of the MWM sentence related to ME part and even propagate the error to the plaintext parts. The existing solution for parsing MWM sentences are based on brittle grammar, including the combinatorial category grammar [73] and the typed PCFG [74]. They both require the semantic analysis of ME, which itself is still a challenge. On the other hand, a data-driven training approach might not be feasible due to the 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 [75], the errors accumulate at both

the PoS and parsing steps. Besides, the dependency/constituent parsing face the challenge in the multi-word expression [76], the special punctuation [77], and prepositional phrase attachment and coordinate conjunction attachment ambiguity [78], [79], [80] even for the regular languages. Nevertheless, features will be extracted from the dependency parsing tree generated from the existing dependency parser and the training process to determine weight assignments to the dependency tree related features.

The declaration extractor will be built on the information from the above low-level processing. The declaration extraction gets attention starting from the NTCIR competition of math understanding [4]. Existing work [81], [67], [68] formulated the declaration extraction problem into two phases: NP candidate pair generation and ME-NP pair classification. From the view of the candidate generation, these existing methods are all using the traditional NLP tools for PoS tagging and NP extraction, where errors were introduced for the MWM sentences processing. From the view of feature engineering for the classification, the features of the classification cover: common declaration patterns, punctuation, word distance, occurrence order of ME vs. the declaration candidate, surface text/PoS of two previous/subsequent words of declaration candidate and ME, uni/bi/tri-gram of the definition candidate, and the surface text of the verb between the ME and candidates. Among all the features, the declaration patterns play the most critical roles. However, the patterns manually enumerated are not complete and it is highly desirable to have an automated or semi-automated method to collect the declaration patterns.

I.2.5.2 Challenges for declaration extraction

In summary, the challenges for declaration extraction comes from two aspects. First, the MEs in MWM sentences have special PoS tags that do not fit into existing categories. The

special PoS tags lead to degradation of the NP extraction as the declaration candidates. Second, the declaration patterns are the features with the most significant weight, but the manual enumeration process might miss many patterns. It is necessary to train a customized MWM processing toolkit and have a (semi-)automated way to collect the declaration patterns.

### *I.2.6 High-level Application of Mathematical Analysis*

#### I.2.6.1 State-of-the-art for math-centered applications

Similar with the search engine to query by keywords, there have been more than ten years of research and many online systems [82], [83] on the retrieval of mathematical expression using mathematical expressions and a mixture of words as inputs. The layout structures of ME variables and operators can support novel approaches for presentation-based IR systems [84], [85], [86], [87], and the semantic structures of MEs, as well as the declaration words, will support semantic-level IR systems [6], [88], [89], [90], [8]. Normalization and approximation of polymorphic forms of MEs are critical to the performance outcomes [7]. Common normalization procedures include the removal of structures (mrow, parentheses, attachment, right-hand side ME), and case normalization. The notation differences are also alleviated by matching MEs with explicit declaration [8]. There are two standard techniques for the indexing term generation: vector space model (VSM) and the suffix tree path. VSM treat the symbols in the MEs are tokens and build a vector space model, while the substation tree indexing [91] will transform each ME into a set of paths. After the term generation, traditional information retrieval technology could be applied for indexing and retrieval, including the language model, the binary model, the BM25 [92]. There are some other MathIR techniques are also design for tree/graph matching. However, as pointed in [7], the systems that support querying by formulae are “perceived as not very useful yet?” Traditional search engines mostly depend on the word matching to locate specific topics or

questions. On the other side, the users need a math search engine are solving problems which require the transformation and derivation from some facts to others. The symbolic computing and proving assistant might be what they want on this aspect.

The MathIR is also highly related with the proving assistant system Mizar [2], theorem prover Coq [3], and mathematical knowledge management system such as Mathematica [93]. Started 45 years ago, the Mizar system is the largest collection strictly formalized mathematical knowledge, containing more than 12, 274 definitions and 59, 706 theorems [94]. Though the formalization is very helpful in organizing the mathematical knowledge for abstract inference, they are less useful for applied mathematics and engineering.

From the view of improving the readability of mathematical intensive papers/books, there is limited research work on ME. There have been attempts to recover the structure of the mathematical discussion within a paper through extraction of the math block and links them using explicit reference based on pattern matching for math terms such as definition, theorem, lemma [95], [96]. But many implicit linkages among the ME are still not recovered yet. For non-mathematical content, the Utopia project [97] enhanced the reading experience of the medical domain by matching external resources such as terminology dictionaries.

#### I.2.6.2 Challenges for math-centered applications

The desired math-centered user experiences are still under exploration. The systems that support searching by MEs are perceived as not very useful [7]. Auto-proving [3] and proof-checker [2] could not scale up due to the massive manual labor efforts and only targeting at pure mathematics. Recovery of logic flow by the reference could not cover the implicit dependency.

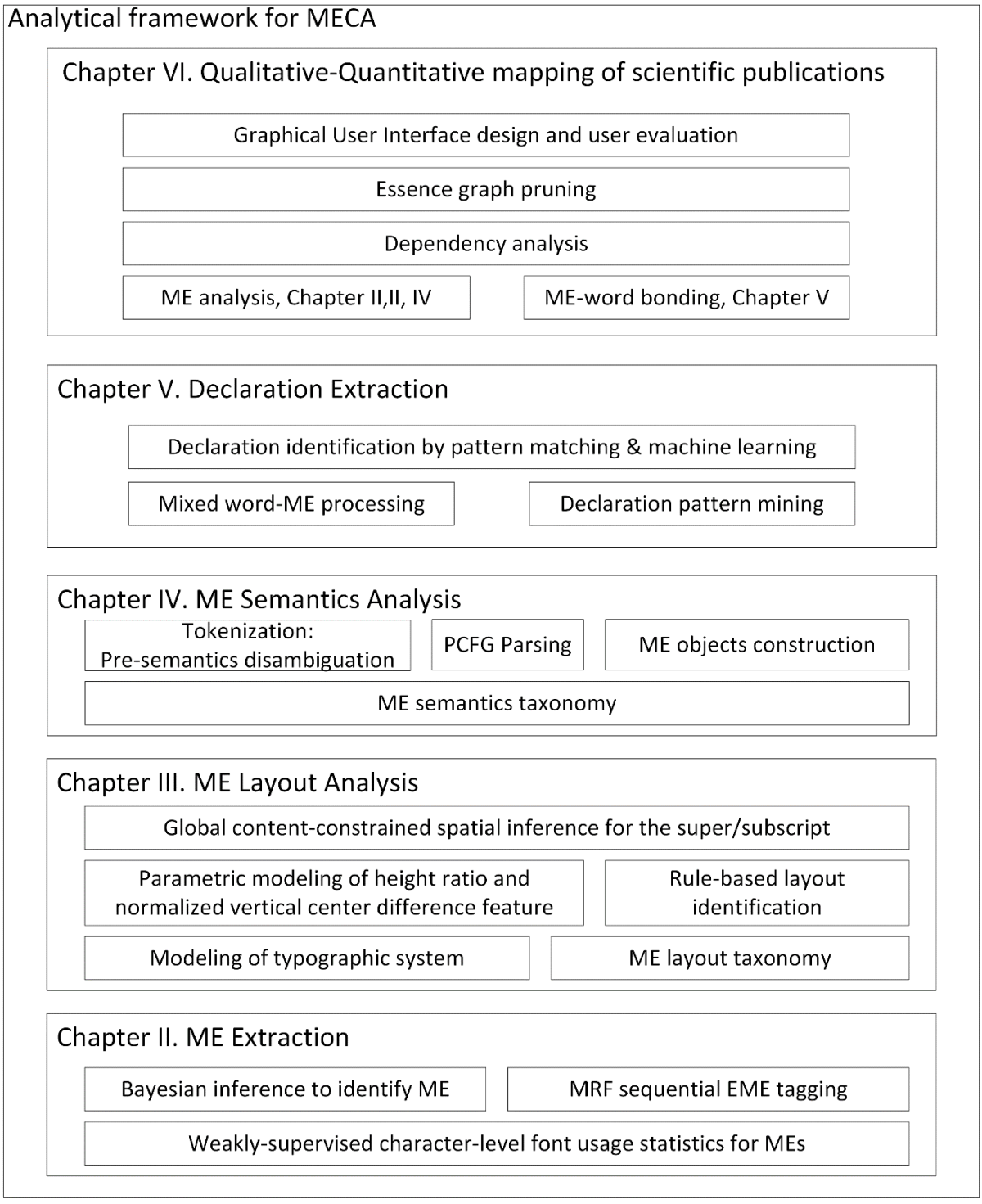



**Figure 5. Research Scope for MECA: Analytical framework**

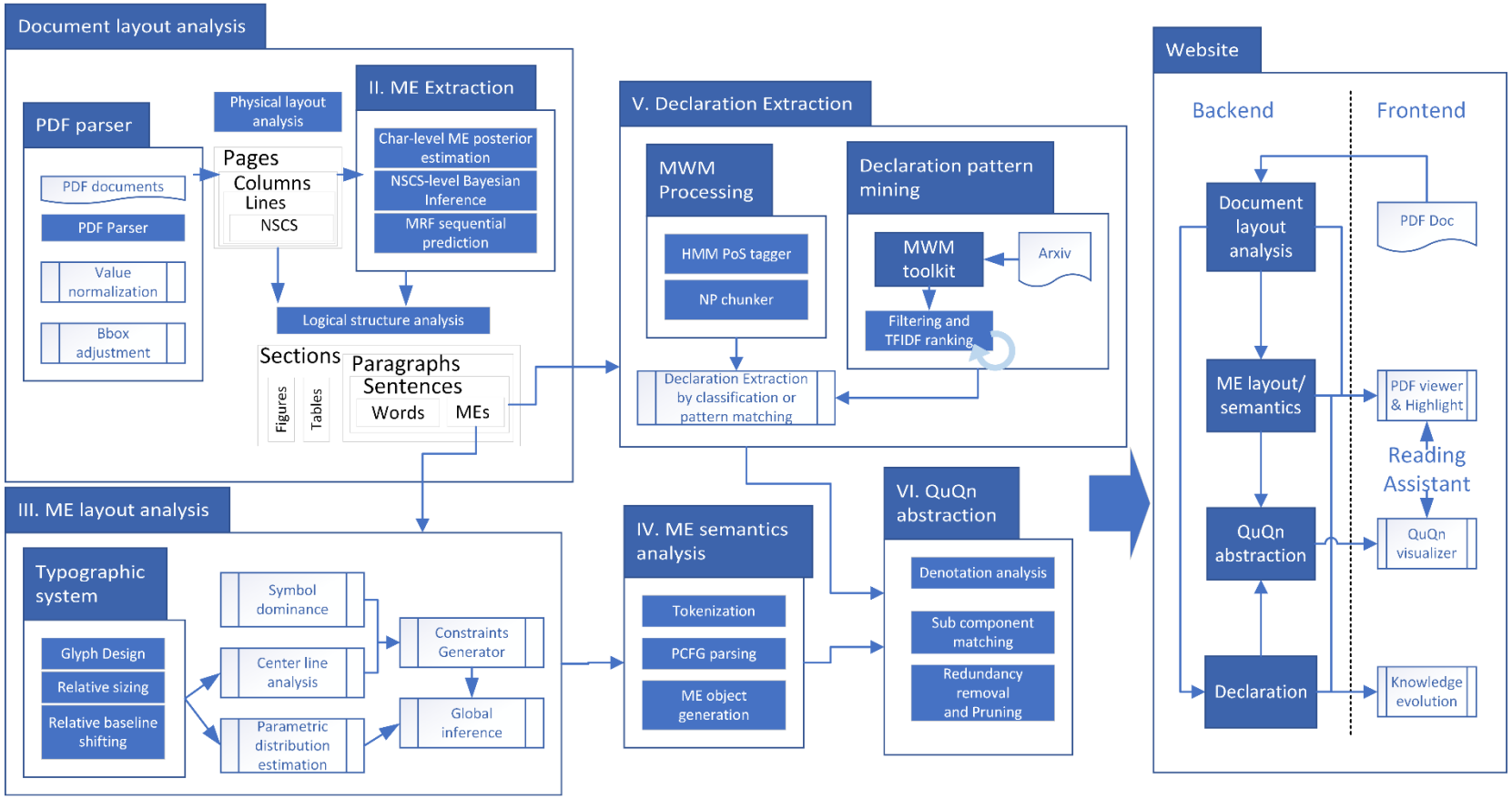



**Figure 6 Research scope for MECA: major software architecture and workflow**

### I.3 Overview of the dissertation

In this dissertation, the Mathematical Expression centered publications Content Analysis system (MECA) is proposed for the large-scale post-publication technical material analysis. Elements of the MECA system, organized by the chapters, are illustrated in Figure 5. Correspondingly, the software architecture and workflow is shown in Figure 6. A complete elaboration of the software system could be found in Appendix C.

Our study starts with Chapter II, which analyzes the logical structure of documents and identify the MEs. A weakly-supervised sequential model to extract MEs from the typesetting of PDF files is proposed to overcome the discrepancy between physical layout and the logical structure and alleviate the difference in writing habits. The essence of this typesetting-based modeling is the consistency of the font usage patterns for MEs and NMEs, either explicit selected by the author or implicitly chosen by the document processing system. Based on the weakly-supervised heuristic rules using the particular symbol values or external dictionary, a significant portion of the ME and NME characters could be identified with high precision. The recognized high confident ME/NME characters could build a reliable estimation of the posterior probability of the character label as ME/NME given its font-value pair. Then, the char-level posterior probability is used for the inference of each physical layout unit (non-separable character sequence) to identify potential EME segments. At last, a Markov Random Field based sequential modeling is applied to remove the local errors to reach global optimality. This weakly-supervised approach based on typesetting provide a simple yet efficient way for the adaptation to the font usage of each writer. The MRF based sequential tagging offers a systematical way to overcome the discrepancy between the physical layout analysis and the logical structure identification.

After the identification of ME represented as typesetting, the next task elaborated in Chapter III is the recovery of the layout structure of ME, which is crucial for the understanding of ME semantics. The ME layout organizes the ME characters into a hierarchical of MEblocks with specific relative spatial relationships. The key to accurate ME layout analysis is the modeling of the typographical system for precise decision-making. A systematic categorization of the characters based on their glyph design is summarized to estimate their normalized height and vertical center reliably. Further, parametric modeling for the height ratio and the normalized vertical center difference (PHN) could be used reliably for the identification of the relative spatial relationships, sub/superscript. The typographic and PHN model provide a solid foundation for the tradeoff between the precision and recall for predictive analysis. The above foundations are deployed into a divide-and-conquer content-constrained spatial (CCS) layout for MEs. First, rules are applied to identify MEBlocks based on the symbol value indicator and the dominated regions. Second, a global inference model is applied for the super/subscript identification that could overcome local errors. The typographic and PHN model are succinct with powerful discriminating ability. The CCS ME layout analysis module on top of them outperforms state of the art with fast execution speed.

The ME layout already encodes lots of semantics manifested as the hierarchical grouping of characters into blocks. But more semantics information is left to explore. The chapter IV presents the systematic modeling and ambiguity resolution techniques to recover the ME semantics. First, a semantic taxonomy of ME is summarized according to the current standard OpenMath [55] and MathML [9]. The ME semantics taxonomy provides a guideline for the semantics parsing and a convenient framework to operate on the MEs. Second, a systematical review of the ambiguity during the ME semantics understanding process is presented. Then, a tri-

phase ME semantics understanding framework is proposed. The first phase is the preprocessing for character semantics disambiguation and characters grouping. The second phase is the PCFG parsing tree construction to find the correct hierarchical scoping. The last phase is the context-dependent ME object generation through tree rewriting. Experiments on a preliminary dataset show that the proposed method could achieve similar ME Semantics to the ground truth.

Besides the MEs, the bonding words around also play important roles. In chapter V, the extraction of declaration for MEs is elaborated, which is very important in linking the mathematical abstraction with the physical worlds. The core for successful identification of ME-declaration is at two aspects: the low-level processing of the mixed Word-ME (MWM) sentences, and the high-level features/patterns for declaration. A customized PoS tagger and NP chunker for the MWM sentences are trained to avoid the degradation that harms the declaration candidate enumeration. Further, a semi-automated weakly-supervised method is developed to gather a variety of patterns for ME declaration. Experiment results showed a significant improvement in the F1 score for ME-declaration identification.

At last, given the rich analytics of the ME semantics from the quantitative aspect and the ME-declaration from the qualitative perspective, these metadata are integrated to create a unified qualitative-quantitative (QuQn) mapping by recovering the dependency and pruning redundancy. The QuQn mapping of a publication provides a concise representation of the technical essence of a publication, with redundant information consolidated and dependency highlighted. A high reduction ratio of around 1:4 is reached. The QuQn map is integrated into a web-based reading assistant system as the graphical organizer of the technical essence with rich interactive features to explore the dependency among factors. The synchronization between the QuQn map and the

original materials make it very easy to switch between the high-level abstraction and low-level detail.

Three application scenarios concerning education and knowledge mapping are explored. A user study during a high school summer camp shows that the QuQn map could help the students understand the dependency among different factors and boost their confidence to learn complex systems. The declaration-based topic clustering captures the technical essence behind the variety of the research task. The paper difference analysis shows the potential of MECA for cross-paper analysis.

# CHAPTER II

# ME EXTRACTION FROM PDF FILES*

## II.1 Overview of the chapter

The ability to locate Mathematical Expressions (ME) from digital files is the entry for math-centered publication analysis. Given that over 90% papers published in Portable Document Format (PDF) according to the statistics from Microsoft Academic Graph [1], this chapter focuses on the extraction of ME from PDF files, which only contains typesetting information. MEs can be further divided into Isolated MEs (IME), which are explicitly separated from the plaintext part, and the Embedded MEs (EME), which are usually treated as a form of technical entity being blended into plaintext sentences for reasoning, explanation, or association of the mathematical notions with the subject under discussion. It is relatively easy to extract IME because of their highlighted spacing. On the other hand, EME extraction is much more challenging due to its resemblance with words and the customize font style outside of the training dataset. The best performance for EME extraction has a false negative rate of 15.9% and a false positive rate over 20% [24].

As IMEs are particular types of physical layout lines and the EMEs are embedded into lines, the accurate physical layout analysis, especially the recovery of the lines, is the foundation for the ME extraction. In this chapter, the document layout analysis is first presented. The Projection Profiling Cutting (PPC) based algorithm for the Line-Column Generation (LCG)

according to the bounding box position of the characters in PDF. Each physical layout line is tokenized based on spacing into non-separable character sequence (NSCS) using the built-in tokenizer of PDFBox and PDFMiner. But when apparent errors are detected, a word matching based tokenizer will be applied.

Then, the IME and EME will be identified from the lines and NSCSes. We observe that authors tend to express MEs in particular styles repeatedly in a paper. This observation leads to a succinct feature space for the labeling of NSCS for EME. Multiple semantic resources that include natural language corpus, citation style, headings, highlighting words, math symbols, and math function names, are used to construct heuristic rules for detecting anchoring MEs and non-MEs (NME), which represent the entities that can be recognized with negligible error. The anchoring ME and NME are used to estimate the probability of a character as ME conditional on its font name and value, which will then be used to extract ME for NSCSes based on the Bayesian inference technique. This weakly-supervised EME identification method is called typesetting-based Bayesian (TSB) model.

Though the TSB model provides a succinct representation and outperforms the state-of-art [24], it could not discriminate well on the characters that are commonly used in both ME and non-ME (NME), such as digits and punctuations. Further, the discrepancy between the physical layout and the logical structure might split one ME into multiple rendering units. These two factors together cause the partial matching problem. Given that these ambiguous characters have a similar probability as ME or NME, their label might be able to be corrected by the label of neighbor NSCSes. This idea is formalized into a Markov Random Field (MRF-TSB) based sequential tagging problem.

The TSB and MRF-TSB models are evaluated on the public dataset Marmot [24]. The TSB outperforms state of the art by 10% for both the miss and false rate. Results show that the proposed sequential techniques could reduce the incorrect split by 1/3, together with a slight improvement on the miss and false rate.

In the following of this chapter, the document layout analysis module is first introduced. Then the TSB and MRF-TSB model will be elaborated. Experiment and result analysis will be given at last.

## II.2 Document layout analysis

### *II.2.1 Document layout model*

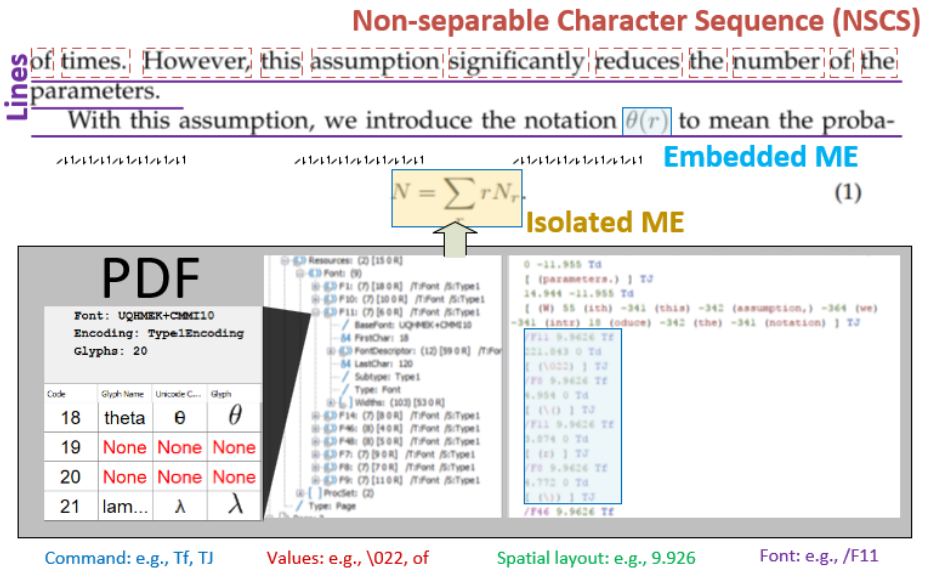



**Figure 7 Document layout analysis from the Typesetting and font resources of PDF file, parts of this figure are adapted from "Lecture Notes 11: The Good-Turing Estimation" [98]**

ME identification is a particular type of document logical structure. The accurate logical structure analysis depends on the precise physical layout recovery as shown in Figure 7. In this work, our Line-Column Generator (LCG) module is designed to produce columns and lines of a page layout in academic publications, which are mostly formatted into single or double columns. For double-column pages, they might also have a single-column header, footer or images/tables. Unlike general document layout analysis where the page orientation can be skewed, in this work, It is assumed that the page orientations of technical papers are either vertical or horizontal. Based on this observation, columns and lines are detected using the concept of Projection Profiling Cutting (PPC) on the converted binary image $I$ from PDF shown in the lower part of Figure 7. A pixel is black if it lies in the character bounding boxes extracted from the typesetting of the PDF files. Formally, for each character $c \in n$, it is associated with the glyph name value $v_c$, font $f_c$, a tight bounding box rectangle $b_c$. Note that some big visual elements such as the open fence for matrix might be split into multiple characters and a pre-merging based on the character value is required [21].

After the LCG processing, a document $D_i$ consists of pages $\{P_{i,j}\}$, where the page $P_{i,j}$ is composed of columns $\{C_{i,j,k}\}$. A column $C_{i,j,k}$ contains lines $\{L_{i,j,k,l}\}$, where each line could stand for an IME or mixed Word-EME line. Each line $L_{i,j,k,l}$ is composed of characters which could be organized as a sequence of non-separable character sequences (NSCS), $(n_{i,j,k,l,1}, n_{i,j,k,l,2}, ...)$, which could be either a plaintext word or part of an embedded ME. IMEs are identified by a classification of the lines. EMEs are identified from the sequence of NSCS separated by space (marked by red dashed rectangles) from the PDF parsing system.

### *II.2.2 Line-Column Generator*

The procedure for Line-Column-Generator (LCG) is illustrated in Figure 8. The PDF files are first fed into the PDF parser [19], [18] to get the tight bounding box for each character for better column/line detection. The TextStripper function in PDFBox could correctly segment lines so that each NSCS corresponds to a word most of the time. Failures are detected when long words are observed. The failure cases will be processed by the PDFminer and a customized tokenizer to maximize the matching of words. After the characters and NSCSes are obtained, a top-down procedure first segments the page into columns as illustrated in Figure 8. After the columns are detected, a bottom-up approach will merge the NSCSes into lines for each column based on the vertical overlapping. Special procedures are designed for the merging of a decorative structure such as the accent and upper/under parts of binding operators.

The column detection procedure follows a two-step approach based on the projection profiling ($pp$), which first decides whether double columns exist, then identifies the double column region. A $pp$ is obtained by projecting the black pixels onto an axis and do a cumulative counting on each position on the axis. The horizontal and vertical profiles for a PDF page are shown in Figure 8 using the test document 10.1.1.58.6850_6 in [24].

A page is detected as double column format if there are at least five lines for the double column region between row pixel index $i_r^l$ and $i_r^h$, s.t. $i_r^h - i_r^l > \delta_{5line}$, and there exist a central gap in the corresponding horizontal PP $pp_h(I[i_r^l{:}\, i_r^h,\ :])$, where $I[i_r^l{:}\, i_r^h,\ :]$ means cropping the image between low boundary row $i_r^l$ to high boundary row $i_r^h$. The center gap is defined as an empty region of at least $\delta_{\mathrm{er}}$ pixels around the center of the horizontal pp of text body region.

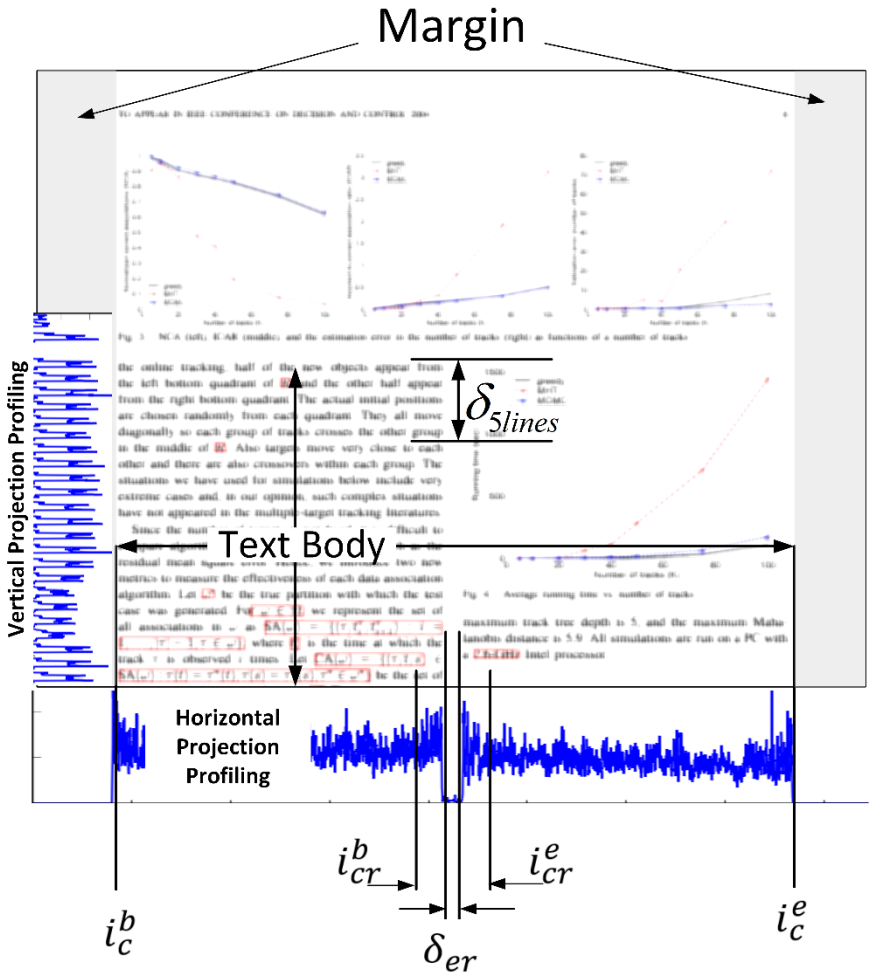



**Figure 8 Column Detection Illustration, parts of this figure are adopted from paper 10.1.1.58.6850 from CiteseerX [99]**

The column range $(i_c^b, i_c^e)$ of the text body is obtained by removing the empty margin. From the column range, the central region $(i_{\mathrm{cr}}^b, i_{\mathrm{cr}}^e)$ is estimated with a width that is in ratio $\alpha$ of the text body. Then, $\delta_{\mathrm{er}}$ consecutive empty pixels are found in the horizontal projection profile of the center region, $pp_h(I[i_r^l : i_r^h, i_{\mathrm{cr}}^b : i_{\mathrm{cr}}^e])$, where $I[i_r^l : i_r^h, i_{\mathrm{cr}}^b : i_{\mathrm{cr}}^e]$ means the cropped image from the beginning column $i_{\mathrm{cr}}^b$ to the ending column $i_{\mathrm{cr}}^e$.

If the double column format is detected, one can find the largest $|i_r^h - i_r^l|$, $i_r^l < i_r^h$, with the constraint that the horizontal PP of $I[i_r^l : i_r^h, :]$ has a central gap. Each column is passed to the line segmentation algorithm, which detects lines based on the zero gaps in the vertical pp. By the end, for each LTTextLine $l_{\text{pdf}}$ extracted from PDF file, a line region $L_{i,j,k}^l$ is detected from PPC such that the overlapping area is at least half of the area of $l_{\text{pdf}}$. Then enumerate through $\{L_{i,j,k}^l\}_{i,j}$ in page $j$ of document $D_i$, and merge the associated $l_{\text{pdf}}$ set to construct the lines.

The center gap ratio $\alpha$ is set to 0.1. The $\delta_{\text{er}}$ is set to 5. And $\delta_{5\text{lines}}$ is set empirically to 400 pixels. By manually checking the line detection results, there is only one failure case where there is an embedded figure. There is one limitation of the PPC based line detection algorithm that it will split the under/over part of IME into separate lines.

Upon the result from document physical layout analysis, the typesetting-based Bayesian model that extracts IMEs from lines and EMEs from NSCSes are introduced in the following subsections.

### II.3 Typesetting-based Bayesian model for ME extraction

Different authors have different document processor environment, and they have free choice in choosing the fonts and layouts for MEs. But the mathematical notations are usually in separate fonts than the words. Given the assumption and observations, a weak-supervised adaptive typesetting-based Bayesian (TSB) model is developed. First, heuristic rules derived from the knowledge of math usage and writing practices are employed to identify the seed set of ME characters $C_{ME}$ and the seed set of NME characters $C_{NME}$ with high confidence. Then, the character-level posterior probability $P_{fv}(L|F, V)$, conditional upon the font $F$ and value

information $V$, is estimated. These posterior probabilities will be used during the inference of the NSCS-level classification as ME or NME.

*II.3.1 Heuristic rules to identify ME/NME characters and their quality*

MEs can be treated as a form of text blended with plaintext words into regular sentences. Some MEs may become reserved, de facto terminologies to represent sophisticated abstractions. In technical writing, important issues are often highlighted in different forms. Several rules are proposed for the partial identification ME and NME characters at the levels of symbols, NSCSes, NSCS sequences, and lines. NME could be the heading of theorems, lemmas and the caption of figures and tables.

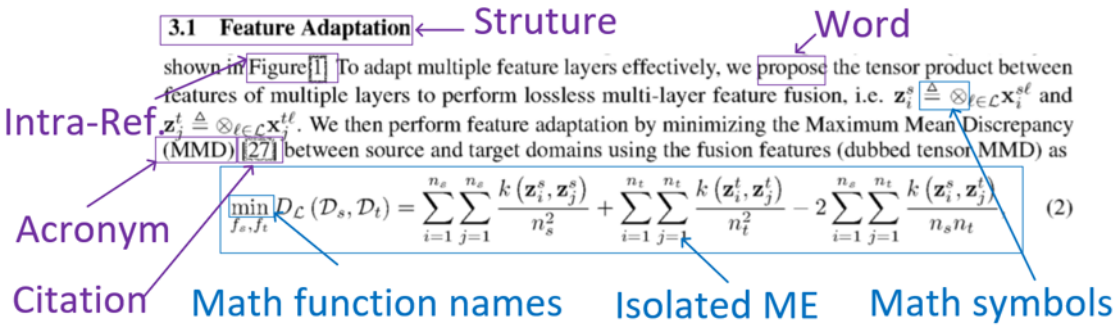



**Figure 9 Heuristics to identify partial ME/NME, parts of this figure are adapted from the "residual transfer networks [100]"**

**Table 1 The rules to match document structures**

| Element | Regex | Example |
|---|---|---|
| Citation | \[\d+(, \d+)*\] | "[1, 17]" |
| | \((\D)*(181920)\d\d\) | (Tracy, 2000) |
| Figure/Table | (figure\|fig.\|table\|tbl.)[ ]* \d(\.\d)*[ ]*(\([a-zA-Z]\)\[[a-zA-Z]\]) | "Fig. 1a", |
| Equation | (equationeqn.eq.formula)[ ]*(\d+(\.\d+)*\(\d+(\.\d+)*\)) | "Equation 1" |
| Theorem | (theorem\|definition\|example\|corollary)[ ]*\d+(\.\d+)* | Theorem 1 |
| Heading | (chapter\|section)[ ]*\d+(\.\d+)* | "Chapter 2" |

**Table 2 Performance of the Heuristics to identify ME/NME**

| | Func. | Math Sym. | IME | Word | Acronym | Citation | Intra | Structure |
|---|---|---|---|---|---|---|---|---|
| NME | 67 | 3849 | 9 | 68570 | 1147 | 687 | 998 | 416 |
| ME | 190 | 26842 | 1409 | 555 | 300 | 40 | 7 | 0 |
| Precision | 0.739 | 0.875 | 0.994 | 0.991 | 0.793 | 0.94 | 0.939 | 1 |

For non-mathematical elements, plaintext words, acronyms, citations, intra document references, and structure indicators such as headings are matched out. The matched words based on natural language corpus covers a lot of characters at a high precision of 0.991. The NSCSes with less than three characters are filtered out to reduce the false positive. The NSCSes are normalized using the Wordnet lemmatizer [101] into its root form and match against the word corpus [101]. The regex rules for matching such elements are summarized in Table 1. An acronym is typically formed from the first letter of multiple word sequence. Acronyms are detected by checking the capitalization and the first letter of surrounding NSCS. Except for the acronym, the other rules for NME all achieve over 90% accuracy as shown in Table 2. As will be discussed later, it is hard to recognized MEs from acronym is hard because the related characters are both used in ME and NME. Further, the human annotation is not consistent either. Based on Unicode value and glyph names, math characters and function names are extracted as MEs. Greek characters, operators, relations [35] are selected as ME symbols. A simple rule is designed for IME detection. If a line contains math elements, but no plaintext words, it will be predicted as Isolated Mathematical Expression (IME), with a precision of 0.994, a recall rate of 0.889 and an F1 score of 0.939. It is slightly better than the best experiment setting of previous work [24]. The first cause of missed IME detection is that the common words for both ME and NME, such as

"for," "and," "otherwise," "super." The other reason is the failure of line extraction and corrupted font resources from PDF parsing.

### *II.3.2 Bayesian Inference for EME identification*

At the NSCS-level where most EME belongs to, there are no silver bullet rules that distinguish ME from NME accurately. Some exception situations include italic fonts for both acronyms and ME and natural language words as variables. But it is observed that authors tend to express MEs in a particular font style repeatedly in a paper. The heuristic rules derived from common writing practice are with high precision at the character level and line level for IME identification. The statistics from the characters identified by heuristic rules will be useful for the likelihood ratio test $L_R(n)$ at the NSCS level as the workflow shown in Figure 10. At last, a post processing step will reject detected EME that overlaps with IME and merge consecutive EME into one ME.

The document elements, characters/NSCS/line, are first pipelined to the rule-based ME/NME identification module, which will produce high confidence character set $C_{\mathrm{ME}}$ and $C_{\mathrm{NME}}$ for ME and NME, respectively. These two sets will be used to estimate the char level posterior probability $P_c(L|C)$ is based on the co-occurrence statistics between font-value and ME/NME label, where $L \in \{\mathrm{ME},\ \mathrm{NME}\}$ is the label and $C \in C$ is the char set. Let $H_{\mathrm{ME}}$ and $H_{\mathrm{NME}}$ respectively denote the font-value co-occurrence matrices, where $H_{\mathrm{ME}}(f,\ v)$ and $H_{\mathrm{NME}}(f,\ v)$ as the count of co-occurrence of font $f$ and value $v$ for ME and NME.

Then $P_C(L = ME|C = c)$ is estimated as:

$$P_{\mathrm{fv}}(L = \mathrm{ME}|F = f_c,\ V = v_c) = \frac{H_{\mathrm{ME}}(f_c, v_c)}{H_{\mathrm{ME}}(f_c, v_c) + H_{\mathrm{NME}}(f_c, v_c)}$$

, where $F \in F$ and $V \in V$ be random variables of font and value defined over the $C$. If the char $c \notin C_{\mathrm{ME}}$, $P_C(L = ME|C = c)$ is estimated by the marginal font conditional probability $P_{\mathrm{f}}(L = \mathrm{ME}|F = f_c)$.

The inference of the label $L$ for a NSCS $N \in N$ is realized through the likelihood ratio test which is transformed using the Bayesian rule as follows:

$$L_R(n) = \frac{P(L = ME|N = n)}{P(L = NME|N = n)} \simeq \frac{P(N = n|L = \mathrm{ME})P(L = \mathrm{ME})}{P(N = n|L = \mathrm{NME})P(L = \mathrm{NME})}$$

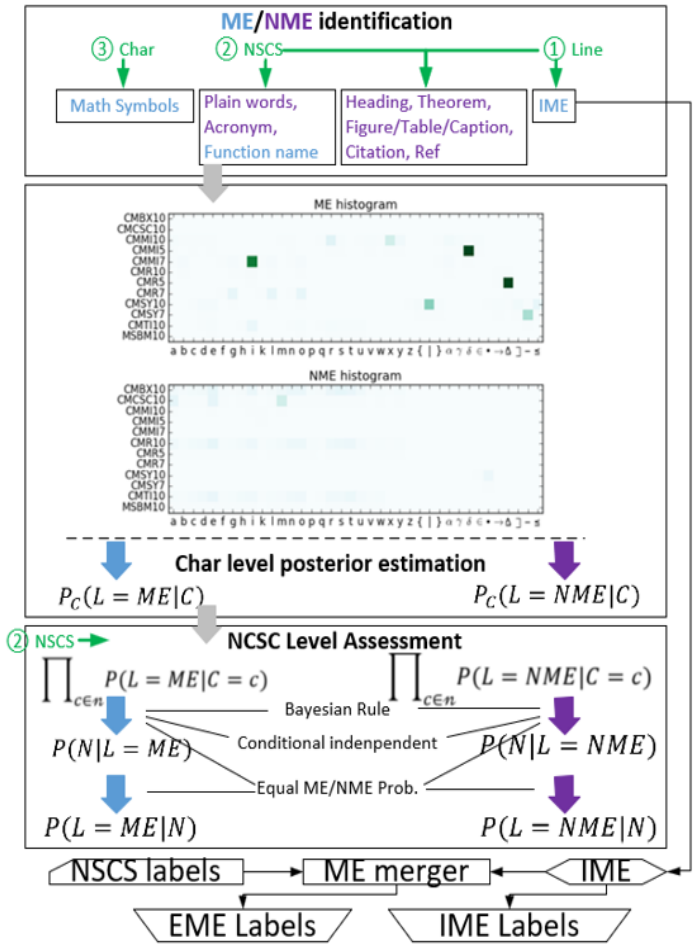



**Figure 10 The workflow for the Typesetting-based Bayesian model, reprinted with permission from [102]**

Given that the combinatorial space for NSCS $N$ is too large for probability estimation, the assumption of conditional independence is made here, where the $P(N|L)$ in (2) could be decomposed as follows:

$$P(N = n|L) = \prod_{c \in n} P_C(C = c|L)$$

The Bayesian rule to transform the char level likelihood $P_C(C|L)$ to posterior $P_C(L|C)$:

$$P(N = n|L) = \prod_{c \in n} P_C(L|C = c)\, P(C = c)/P(L)$$

It is further assumed the equal prior probability of ME vs. NME, i.e., $P(L = \text{ME}) = P(L = \text{NME})$. Then plug in the expansion based on conditional independency into the likelihood ratio test and cancel out $P(C = c)$, leading to:

$$L_R(n) = \prod_{c \in n} \frac{P_C(L = ME\ |C = c)}{P_C(L = NME\ |C = c)}$$

However, errors occur frequently for the punctuation and digits, leading to the split of one ME into multiple parts. This problem will be elaborated in the next sequential EME extraction section.

### II.4 An MRF-based sequential modeling for EME extraction

Labeling of EME is still a problem not fully solved due to the fuzzy boundary. For instance, many EMEs are incorrectly split due to misidentification of a few characters. As shown in Figure 11, the fuzzy boundary is mainly due to the discrepancy between the physical layout units separated by the red lines and the logical structures, causing errors in the EME prediction marked in the blue shaded area. Existing work and my TSB model only use information within NSCS without systematic incorporation of neighboring information.

By further exploring the log of the posterior probability of each NSCS as ME and NME in Figure 11, the observation to correct the NSCS label prediction by its neighbors is shown. The plaintext words ("for," "so," "that") have large log probability as NME compared with ME. For most of the ME parts, they have large log probability as ME compared with NME. However, there are less determinant zone such as punctuations and digits, causing the over split of ME. However, the label of their direct neighbors could play an important role in predicting the right label.

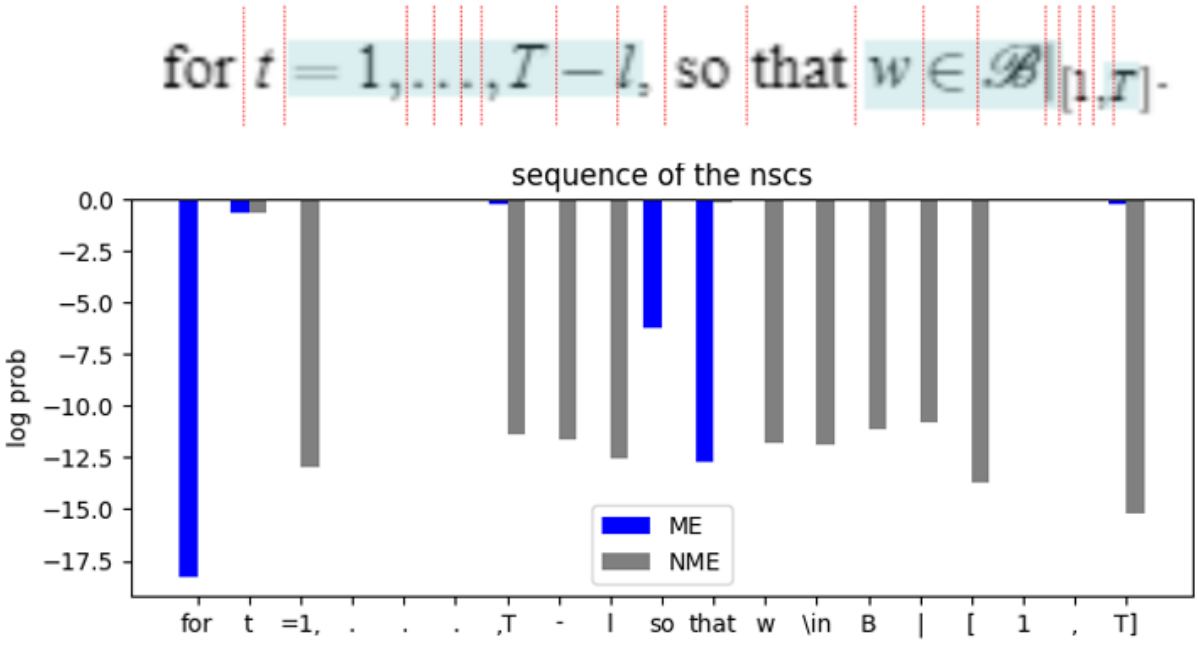



**Figure 11 The motivation for sequential tagging and the related posterior probability $log(P(ME|n^i))$ and $log(P(NME|n^i))$, parts of this figure are adapted from 10.1.1.6.2281_9 in Marmot dataset [24]**

Inspired by the pair-wise potential concept commonly used in the Markov Random Field algorithm, an MRF based extension to the existing TSB model for sequential prediction is proposed, which incorporates neighbor constraints in labeling of EME vs. plaintext. Experimental results show that this technique significantly reduces splitting of EMEs, with small gains in the false and miss rate.

The rest of this section is organized in the following order: In this section, the MRF-TSB pair-wise potential model for sequential EME prediction is first presented. Then, an example is used to illustrate how MRF-TSB works as well as a sensitivity analysis for the parameter settings is given. At last, the optimization solver design is presented.

*II.4.1 Problem formulation of MRF-TSB model*

The embedded mathematical expression identification will be on the lines not identified as IME. Given such a line $L$ composed as a NSCS sequence $\{n_1, \dots, n_{N_L}\}$, the goal is to predict EME label sequence $y = \{y_1, \dots, y_{N_L}\}$, where the superscript is omitted for convenience, $N_L$ is the number of elements in the line. $y_i \in \{0,1\}$, where 0 indicates plaintext and 1 indicates EME. From the view of the pointwise decision process, the existing TSB model could be modeled as posterior probability maximization. It is equivalent to minimizing the negation of summation of log probability $-\sum_{i\in[1,N_L]} log(P(y_i|n_i)))$, i.e.,

$$U(y) = -\sum_i y_i \log(P(L = ME|N = n_i)) + (1 - y_i)\log(P(L = NME|N = n_i))$$

, where $P(L = [N]ME|N = n_i)$ is cacluated from TSB model. For convenience, let $U(y)$ denote $-\sum_i y_i log(L_R(n_i))$, where $\mathrm{L_R}(n_i) = P(L = ME|N = n_i)/P(L = NME|N = n_i)$.

Given this observation, a heuristic is proposed, which prefers the label of $y_i$ to be similar with the label of its neighbors $y_{i-1}$ and $y_{i+1}$. Mathematically, a penalty is added for the

difference in the consecutive labels, i.e., $P(y) = \sum_{i \in [1, N_L)} |y_i - y_{i+1}|$. By merging the above two factors, we have the following minimization objective function $U(y) + \lambda P(y)$, where $\lambda > 0$ is a weight parameter.

*II.4.2 How MRF-TSB model works and the parameter setting*

We will study two scenarios based on the above example. For the latest quadruple sequential of [“[”, “1”, “,”, ”T]”]. The values of objective function under different predicted labels are enumerated in Table 3. From the table, we can see that, if we assign the less-determinant NSCS as NME (label 0) between highly determinant ME, penalty $2\lambda$ will be introduced, which is consistent with the requirement $\lambda > 0$ for our formulation to help alleviate the over split issue.

**Table 3 Objective value table for the case [“[”, “1”, “,”, ”T]”]**

| Label | Objective value | Reduced |
|---|---|---|
| [1,0,0,1] | 1*-14+0*0+0*0+1*-15+2$\lambda$ | -29+2$\lambda$ |
| [1,0,1,1] | 1*-14+0*0+1*0+1*-15+2$\lambda$ | -29+2$\lambda$ |
| [1,1,1,1] | 1*-14+1*0+1*0+1*-15 | -29 |

**Table 4 Objective value table for the case [“that”, “w”]**

| Label | Objective value | Reduced |
|---|---|---|
| [0,0] | 0*13+0*-12 | 0 |
| [0,1] | 0*13+1*-12+$\lambda$ | -12+$\lambda$ |
| [1,0] | 1*13+0*-12+$\lambda$ | 13+$\lambda$ |
| [1,1] | 1*13+1*-12 | 1 |

On the other hand, we should not set $\lambda$ too high. For example, ["that", "w"], we enumerate the objective function value under different labeling situation in Table 4. The objective value of the ground truth is -12+$\lambda$. However, if we set the $\lambda > 12$, then the best prediction will be [0,0]. More analysis will be presented in the experiment section on how the parameter setting for $\lambda$ will affect the final decision.

The parameter $\lambda$ should be larger than 0 to penalize the difference in consecutive labels. But, it should not be too large, so that it has more effect than the unary potential, leading to false negatives. From the statistics of negative log likelihood ratio $-\log(\mathrm{LR}(\mathrm{n}^{\mathrm{i}}))$ in Figure 12, we can see that most of the false-negative samples causing the over split are with 0 value. The false negative means that they should be ME but predicted as NME, like the case in Table 3. While to avoid over-correction that label ME as NME illustrated by the case in Table 4, $\lambda$ should be smaller than the absolute value of the true positive statistics in the first row. This parameter setting is in accordance with the general performance to be presented in the experiment section, where smaller $\lambda$ leads to better performance.

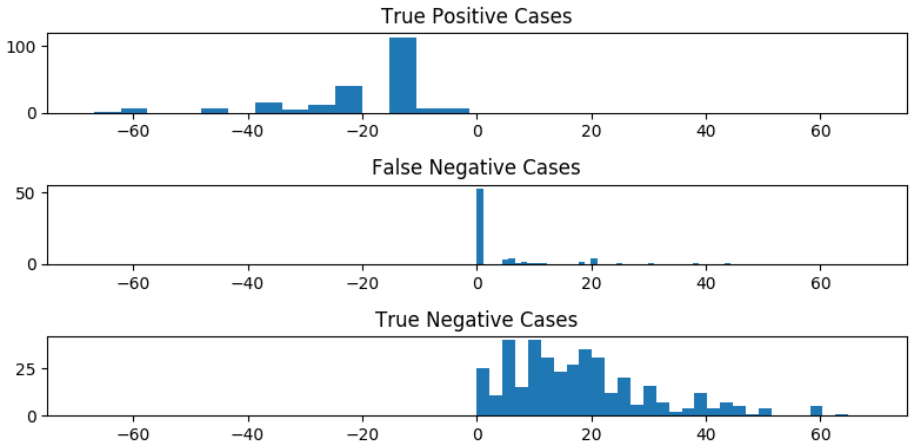



**Figure 12 Statistics of the negative log-likelihood ratio for "10.1.1.6.2281_9 [103]"**

### *II.4.3 Solver design*

The condition $y_i \in \{0,\ 1\}$ indicates the optimization as a mixed integer programming (MIP) problem. However, the absolute value lead to non-linearity. It is transformed in the following way: for each absolute value $|y_i - y_{i+1}|$, $i \in [1, N_L)$, two auxiliary variables $z_i^+$ and $z_i^-$ are introduced with the following constraint set $C$: $z_i^+ + z_i^- = |y_i - y_{i+1}|$, $z_i^+ - z_i^- = y_i - y_{i+1}$, $z_i^+, z_i^- \in \{0,1\}$. Then the optimization goal is transformed into the following MIP problem: minimize

$$f(y,z) = -\sum_{i\in[1,N_L]} \log\big(\mathrm{L_R}(n_i)\big)y_i + \lambda \sum_{i\in[1,N_L)} (z_i^+ + z_i^-)$$

, subject to the constraint set $C$ and $y^i \in \{0,\ 1\}$.

## II.5 Performance and analysis for ME extraction

The dataset and the evaluation criteria will be introduced first. Then, the experiment settings for compared methods are presented. At last, I will show the performance statistics for the TSB, MRF-TSB model, and other comparison models, followed by some case studies.

### *II.5.1 Dataset and evaluation criteria*

In this paper, the Marmot dataset and the criteria in [24] are used. The dataset contains 400 papers with additional 1888 ME labeled in [104]. MEs in figures were mostly not labeled in the previous work. Thus also do not consider them in the evaluation process. The ME in caption and footnote are kept as the original ground truth.

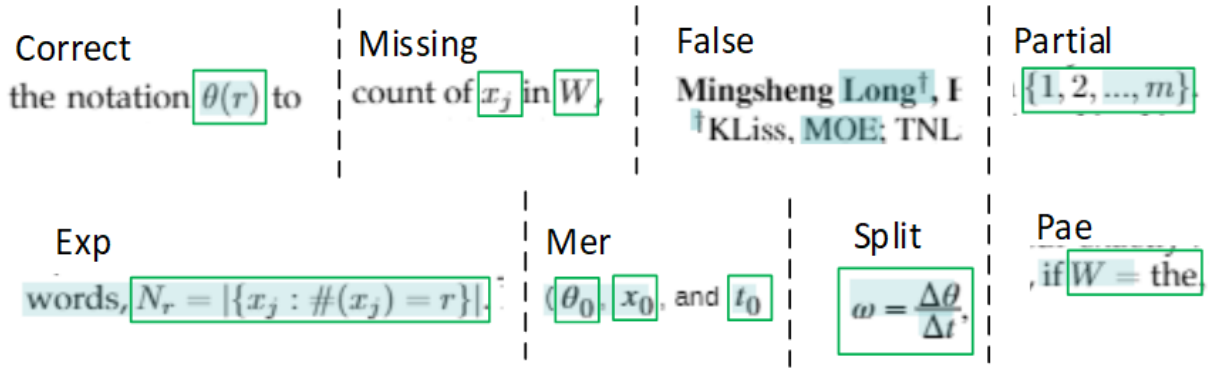



**Figure 13 The criteria for the performance evaluation of ME Extraction**

The evaluation is challenging given the possibility of only partial element extracted. All possible matching situation between the ground truth and the prediction is illustrated in Figure 13. Given a set of ground truth ME $M_{\mathrm{gt}}$ and a set of predicted ME $M_{\mathrm{pd}}$. First, an ME in $M_{\mathrm{gt}}$ is missing if it does not overlap with any predicted MEs. For a predicted ME $m_{\mathrm{pd}}$ could be one of the seven relations: Correct, Expanding, Merging, Partial, Split, Partial&Expanding(PAE), and False . Correct means fully are overlapping. Expanding (Exp) means that the $m_{\mathrm{pd}}$ contains only one ground truth ME $m_{\mathrm{gt}}$, and expanding and merging (Mer) mean that $m_{\mathrm{pd}}$ is equal the merge of multiple ground truth MEs. Partial and Split (Spl) mean the predicted ME is only partial of an ME $m_{\mathrm{gt}}$ in ground truth, where the partial (Par) indicates only the predicted ME is contained in the $m_{\mathrm{gt}}$. The remaining overlapping situation is marked as PAE. In addition to the detail number in each matching category, three coarse level measurements are adopted: the miss rate $r_m = \frac{\#(Mis)}{\#(Total)-\#(Fal)}$, the false rate $r_f = \frac{\#(Fal)}{\#(Total)-\#(Mis)}$, and $F1 = \frac{2*(1-r_m)*(1-r_f)}{(1-r_m)+(1-r_f)}$, where $\#(Total)$ is the number of processed MEs in prediction or ground truth.

### *II.5.2 Experiment settings for comparison CRF based method*

The CRF based EME sequential labeling system [40] is used for comparison. Given the line, $L = \{n_1, \dots, n_{N_L}\}$, the desired label sequence of {"B", "I", "O"} need to maximize:

$$P_\Theta(\mathbf{y}|\mathbf{x}) \propto \exp(\sum_i (\sum_j \lambda_j f_j(e_i) + \sum_k \mu_k g_k(n_i, y_i)))$$

The label "B" indicates the beginning of math, "I" for in math, and "O" for out of math. $f_j(e_i)$ is the j-th feature defined on the edge $e_i = (y_{i-1}, y_i)$ and $g_k(n_i, y_i)$ is the k-th feature define over the NSCS $n_i$ together with the label $y_i$. Features $\{g_k\}$ are the same with previous work except for the block feature covering the font, word, and character. Further, to avoid information inequality between the CRF based method and TSB/MRF-TSB, three features are added, including plaintext words, citations, and reference to figures and tables used in our TSB model. Since CRF is supervise training model, a 5-folder cross-validation is adopted. Python-CRFSuite toolkit [105] is used for training.

### *II.5.3 Performance*

The TSB and MRF-TSB are compared against two state-of-the-art systems. Lin [24] is the baseline. TSB is the font-setting based Bayesian Model. The CRF1 is the same with [40], and CRF2 is enhanced with the features used in the TSB model. The performance at the macro-level is shown in Table 5. First, our TSB model outperforms the state-of-art method Lin for both the missing rate and false rate, corresponding to a 0.1 increase in the F1 score. Further, when the 1888 ME samples that were identified in, the performance of earlier work may need to be adjusted.

**Table 5 Coarse performance statistics for EME detection**

| | $r_m$ | $r_f$ | F1 | | $r_m$ | $r_f$ | F1 |
|---|---|---|---|---|---|---|---|
| Lin | 0.159 | 0.23 | 0.804 | $SEQ_{.5}$ | 0.0782 | 0.079 | **0.921** |
| FSM | 0.083 | 0.089 | 0.916 | $SEQ_1$ | 0.0975 | 0.074 | 0.914 |
| CRF1 | 0.206 | 0.049 | 0.87 | $SEQ_2$ | 0.111 | 0.070 | 0.909 |
| CRF2 | 0.217 | 0.050 | 0.858 | | | | |

The TSB model along also outperform the CRF model on the F1 measurement. CRF is with a lower false rate at the cost of high miss rate, which might be due to its sensitivity to training data. The Marmot data is randomly selected from Citeseerx and has more noise than the ACL repository in [40]. Adding the information in FSB model into the CRF model is not helpful, which will be explained in the following CRF case study.

By extending the TSB model with sequential modeling, the MRF-TSB outperforms the baselines, TSB and CRF. The performance is high when the $\lambda$ is set to a smaller value with the reason discussed in pthe arameter selection section.

The most common false cases are the section numbers, reference to the equation and some plaintext words connected with bracket. A particular example is a file with square brackets surrounding the reference. As for the missing part, single char variables are the common cause. The capitalized variables are also confused with acronyms.

Besides the miss rate and false rate, the over split issue is another principal target of this paper with the result shown in Table 6. The MRF-TSB model alleviates the over split problem by over 1/3 and reduces the false cases. When the parameter $\lambda$ is set to a high value, it will have lower false and partial rate, at the cost of increased miss and expansion cases.

**Table 6 Detail Performance statistics for EME detection**

| | Cor | Mis | Fal | Par | Exp | Pae | Mer | Spl |
|---|---|---|---|---|---|---|---|---|
| **TSB** | 4906 | **762** | 921 | **3091** | 841 | 580 | 1 | 4 |
| **$SEQ_{.5}$** | 4418 | **872** | 773 | **1887** | 2000 | 717 | 3 | 0 |
| **$SEQ_1$** | 4393 | **964** | 714 | **1820** | 1999 | 701 | 3 | 0 |
| **$SEQ_2$** | 4336 | **1088** | 664 | **1728** | 1996 | 684 | 3 | 0 |
| **$CRF_1$** | 4029 | **1981** | 396 | **1461** | 1605 | 559 | 3 | 0 |
| **$CRF_2$** | 4071 | **1945** | 404 | **1447** | 1605 | 570 | 3 | 0 |

*II.5.4 Case study for CRF model to show its drawback*

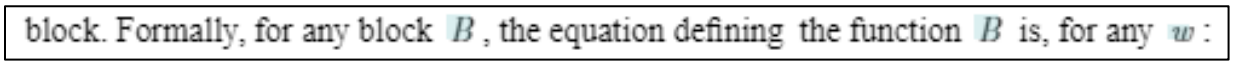



**Figure 14 Example to show the fallacy of the CRF model, parts of this figure are adapted from 10.1.1.6.2308_3 in Marmot dataset [24]**

The CRF method has a high miss rate. The reason is explained using one case study shown in Figure 14. The ground truth is that "B" and "w" marked in the light blue background are mathematical notations. But they are not predicted as ME. We study the unary probability for the token "w" in Figure 15. Given the conflicting situation, the linear summation of the coefficient given the math label "B" (0.632) is smaller than the plaintext label "O" (2.87). The main contributing factors for this wrong prediction are a few general features: only contain letters, no Greek symbols, no math symbols. These global features are mostly *reverse sufficient condition*. Here, *reverse sufficient* means when the value switched, they are good sufficient

indicator. For example, if "greek=T," i.e., there exist greek symbols, it is very likely that the NSCS is EME. But, no greek symbol "greek=F" in this case does not mean it will not be EME.

```
word=w, B:0.341769, O:-0.21002
font=DKFYPO+CMTI10, B:nan, O:nan
fontsuffix=CMTI10, B:0.081808, O:0.276641
length=1, B:-0.058035, O:0.022353
samew=T, B:0.201584, O:0.138891
samef=T, B:0.258952, O:0.059454
samewf=T, B:0.251179, O:-0.004849
alpha=T, B:-0.057297, O:0.906074
greek=F, B:-0.146672, O:0.397723
math=F, B:-0.687054, O:1.553613
single=T, B:-0.058035, O:0.022353
mainfont=F, B:0.503912, O:-0.28969
gd_sum:0.632111, pd_sum:2.872543
```

**Figure 15 The feature weight for CRF based EME identification**

Another issue is that the parameter is sensitive the training dataset. This case is in the fold 3 experiment. The weight of "fontsuffix=CMTI10" is with low weight for "B" in comparison with "O," which is not true for the parameter of the fold 1 experiment shown in top parameter weight.

### *II.5.5 Computational cost*

The average execution time (Python code based) for one PDF page is decomposed as follows: 1.89 seconds for layout analysis, 2.25 seconds for heuristic rule matching and font statistics, 0.22 seconds for IME identification, and 0.12 seconds for EME identification. In comparison, the supervised machine learning methods would take about 1 second to predict a line, 10 seconds to predict a word. It took 12 and 763 seconds to train line and word classifiers, respectively. The enhanced MRF-TSB is slower because it will call an external MIP solver.

## II.6 Conclusion

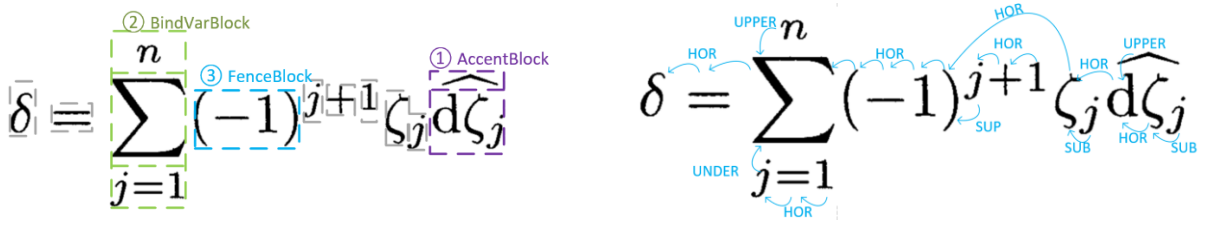
In this section, two open problems in the extraction of ME are attacked and partially solved: the customized font usage and the EME-splitting problem due to the discrepancy between the physical layout units and semantic logical structure. The ME extraction is a complex task involving many processing steps for PDF parsing, document layout analysis and construction of resources. A weak supervised typesetting-based Bayesian (TSB) model is proposed first by leveraging on knowledge about the natural language, technical publication practice, and probabilistic models. The TSB model could adapt to the input PDF about the font usage based on elements extracted from heuristic rules. Then a Bayesian inference is conducted for each NSCS. Second, a Sequential EME extraction model is developed to incorporate the neighbor information during the decision-making. Results show that the TSB outperformance state of the art by 10% regarding missing and false rate. The sequential modeling can significantly reduce the over split issue, which is very important in the later stage of ME parsing. Both TSB model and MRF-EME model are explainable and easy to be interpreted and intervene.

CHAPTER III

CONTENT CONSTRAINED SPATIAL MODEL FOR ME LAYOUT ANALYSIS*

## III.1 Overview of the chapter for ME layout analysis

Representing MEs at the semantic level is the basis for high-level task information retrieval [42], machine reading [106], and even auto-proofing [3]. ME could be treated as a type of visual language [107], and the semantics of MEs is manifested by both the particular values of the characters $\{c_i\}$ in an ME such as operators/alphabets and the ME layout as illustrated in Figure 16. The ME layout is a hierarchical grouping of the characters and the relative spatial relationships among blocks. It could be transformed into equivalent character-level dominance shown in Figure 16.b. This chapter focuses on recovering the ME layout from typesetting information in PDF, where the typesetting only contains the symbol value and their size/position.




(a) Hierarchy of ME Layout (b) Character-level dominance

**Figure 16 Example of ME layout**

---

*Reprinted with permission from "A content-constrained spatial (CCS) model for layout analysis of mathematical expressions" by Wang, Xing and Liu, Jyh-Charn, 2017. *Twelfth International Conference on Digital Information Management (ICDIM)*, Fukuoka, 2017, pp. 334-339. Copyright 2017 IEEE.

The composition of MEs covers the following two aspects: the atomic building units of characters and the hierarchical spatial arrangement. First, the characters are the atomic building units of an ME. The character values are indicators of their semantics. Alphabets and Greeks are commonly used as variables and operations/relations are expressed by values such as summation, less than, etc. Some special character values are indicators to look for particular layout structures. For example, the accents, binding operators, and fraction line are indicators to look for the upper/under associated elements. The challenge from the first aspect is that the same character might have different semantic meanings and layout convention. Take '*' sign in Figure 19.b for example. When used as a binary operator, it is in a horizontal relationship with the left operand and the right operand. When used as identifier decorator, it is commonly placed at the superscript of an identifier.

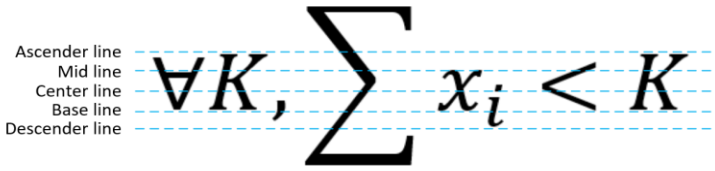
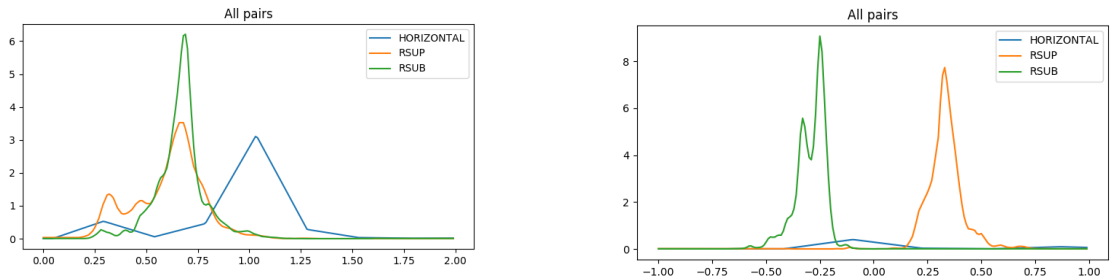



**Figure 17 The challenge from the glyph of the characters**

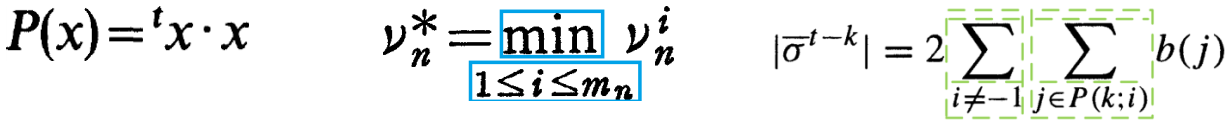
Besides the meaning ambiguity for the same character, the glyph design also need to be normalized carefully to assess the relative spatial relationship. At each layer of the hierarchy, the characters are commonly arranged from left to right on several baselines, which could also be placed at the super/subscript and upper/under position for different decorative purposes. Smaller glyph sizes and a shift in vertical direction indicate being dominated such as sub/superscript or

under/over parts, playing decorative roles. But it is non-trivial to resolve the relative spatial relationship due to the difference in the glyph design, visual appearance and placement for each character as shown in Figure 17. Even with the same font size, the glyphs of some characters are designed smaller than others for ease of reading. For example, the character 'i' is higher than the character 'n' in function name 'min' in Figure 19.b. Although most of the alphabets, digits and Greek letter are aligned with the typographic reference lines, there are half of the mathematical operators (43% of all ME characters in [47]) not aligned with the reference line. It is difficult to estimate the baseline for the non-aligned characters directly. Further, there are special symbols that are usually small such as the punctuation and accent characters, and there are many big operator and fence characters with varying size. The varying and small size leads to the challenge in recovering the normalized height from the ascender line to descender line to judge whether two characters on the same baseline level based on the size. If not normalized, there will be significant overlapping in the distribution, limiting the upper bound of the discrimination performance shown in Figure 18.




(a) Probability density function for HR (b) Probability density function for NVCD

**Figure 18 Degraded discrimination performance**




*(a) Left or right superscript* *(b) The scope of the under/upper* *(c) The scope of binding operations*

**Figure 19 Examples to illustrate the ME layout and challenges, parts of the figure are adapted from Infty-CDB [47]**

Second, the characters are grouped into a hierarchical structure as illustrated in Figure 16.a. The hierarchy origin from ME semantics from a top-down decomposition. Partial of the structure could be recovered based on symbol dominance of the binding operator/accent/fence or the matching of common practice such as the function "min." But the loss of the grouping information leads to the ambiguity that one character could be interpreted to be affiliated with many neighbors. In Figure 19.a, the superscript "t" should be attached to the left operator "=" or the right variable "x." In Figure 19.b, "1" could be interpreted as the under part of "=" or the left part of "≤." In Figure 19.c, there are two consecutive summation binding operators, and the algorithm needs to make sure the "j" is grouped to the under parts of the second binding operator rather than the first one. Another challenge brought by the hierarchical structure is the degradation of feature discrimination ability. A common way to calculate features between blocks [64] is to use the whole block, but the whole block might not reflect the real baseline such as the "min" structure in Figure 19.b and the bind operators in Figure 19.c.

Though there are only limited relationships types between blocks, the possible combination will explode when building the hierarchy bottom-up for many characters. Local greed approach [46], [45] face the challenge of error propagation especially when a

misprediction is inevitable based on the feature distribution as shown in Figure 18. On the other hand, the global inference faces the challenge high computational cost. The PCFG based method [64] has a complexity of $O(n^3 lgn|P|)$, where P is the set of derivation rules. Further, the method heavily depends on the grammar rules [64] or the symbol dominance rules [45] will fail when there is out of rule situation when the author develop their notation and layout system.

Given the natural of intertwining between the semantic and layout, a content-constrained spatial (CCS) model is proposed to solve the challenges of the ME layout prediction. The following issues will be explored:

- Formalize the typographic model and the recovery of the perceived normalized height and vertical center.
- Enumerate the ME Layout hierarchy systematically and partially recovery structure based on character dominance and high confidence spatial relationships.
- Design discriminative features capturing the long-distance dependency relationship and a parametric approximation for fast inference

This chapter is organized as follows. Before going into details of the proposed method, the background knowledge about the typographic design and a few critical reference lines are introduced. Next, we present ME layout taxonomy, which is the basis of our divide-and-conquer approach. In the first phase, the rule-based approach will identify the partial ME Blocks based on the symbol dominance and high confident spatial analysis. In the second phase, a global inference model is proposed to identify the super/subscripts among horizontally arranging ME blocks. For the efficient inference of the best ME layout using the CCS model, a parametric approximation of the probability density function is developed for the features to discriminate the relative spatial relationship by modeling the relative sizing and shifting of the

super/subscripts. Experiment evaluation and analysis are conducted on the public InftyCDB dataset by the end.

## III.2 Typographic System

The digital typographic system arranges the glyph of characters in 2D space. The perceived height and vertical center difference are very important for the baseline assessment. In this section, the reference lines to place characters will be introduced first. Then, categorization of characters based on the alignment to the reference lines is presented, together with the models to recover the perceived normalized height.

### *III.2.1 Typographic lines*

In the typographic system, the glyph of characters is placed based on the five reference lines (RL) used in typography systems are illustrated in Figure 20. Most characters use the baseline as their anchoring level, upon which letters may extend downward (upward) to reach the descender (ascender) line. The midline is meant to be the middle point between the baseline and the ascender line, which is the upper boundary for characters such as “o.” The centerline is the midpoint between the descender and ascender line.


**Figure 20 The typographic reference lines**

*III.2.2 Categorization of characters and recovery of the normalized height*

When only considering the tight bounding box (bbox) of each character, one might get the wrong conclusion. For example, in Figure 20, "p" might be misidentified as the subscript of "H." For this reason, it is necessary to recover the normalized height, or equivalently recover the ascender line and descender line from the tight bbox. Based on how the elements are aligned with the reference lines, the characters are organized based on their glyph as shown in Table 7.

**Table 7 Glyph types and categorized of the characters**

| Glyph Type | Characters |
|---|---|
| RL aligned | + ± ∓ × ⋊ ⋉ † ‡ ⊖ < ≤ ≪ ⊂ ⊆ ⊏ ⊑ ∈ ≺ ≤ ≤ ⊈ ⊀ ⊄ ⊈ ⊉ ⊊ > ≥ ≫ ⊃ ⊇ ⊐ ⊒ ∋ ≻ ≥ ≥ ⊉ ⊁ ⊅ ⊋ ⊉ ≠ ∥ ⊥ \| ∤ ⋕ ⊢ ⊣ ⊥ ⋱ ⋰ ⋮ ; : ! ↑ ⇑ ↓ ⇓ ↕ ⇕ ↗ ↘ ↙ ↖ ∞ ∅ ℵ □ ■ ⋈ *fl ff fi* ⫛ ∃ ∀ ∂ ∇ |
| | Greek, Alphabets, Digits |
| Narrow, width stable | ÷ − = ≐ ≅ ≡ ≢ ~ ≃ ≈ ⋌ ⋎ ⌣ ⌢ ¬ ≙ |
| centered, varying size | ⨿ ∑ ∫ ∮ ∏ ∐ ∩ ∪ ⋂ ⋃ ⊓ ⊔ ⨅ ⨆ ∨ ∧ ⋁ ⋀ \ ⧹ / ⧸ ∉ ≠ ⊀ ⊁ ≠ ⊈ ⊉ ⊬ ⊭ ∦ ∤ ∝ ⋈ ⊨ ⊙ ⊙ ⊗ ⊗ ⊕ ⊎ ⨁ ⨄ ⊘ ◯ ◁ ◁ ⊴ ▷ ▷ ⊵ △ ▽ ▽ ◇ □ ≀ \| ∤ ( ) [ ] { } ⟩ ⟨ ⌈ ⌉ ⌊ |
| Narrow centered | · * ⋯ ⋆ ★ ♮ ♯ ∘ ● ⋄ © ← ⇐ ← ↦ ↩ ⟵ ⟸ ⟼ ⇝ → ⇒ → ⟶ ⟹ ↪ ↔ ⇔ ⇌ ⟷ ⟺ ← ← ⇀ ⇁ |
| special characters | √ ′ ″ , . . . . . ^ ~ ¨ ‾ _ ⏟ |

Given the five types for the glyphs, only the characters that are RL aligned or width stable as illustrated in Figure 21 could have their normalized height recovered reliably. Given a character $c$, denote $y_c^t$, $y_c^b$, $x_c^l$, and $x_c^r$ as the top, bottom, left and right of the tight bounding box. For a character $c$ that is aligned with reference lines vertically, its ascender line $c.ascender$ is calculated as the $y_c^t + h_c^t \times ur^v$, where $h_c^t$ is the tight height of character $c$ and $ur^v$ is the ratio between the ascender line - glyph top gap $c.ascender - y_c^t$ and the tight height $h_c^t$. Similarly,

the descender line is recovered as $c.descender$ as $y_c^b - c.h^t \times dr^v$. The ascender/descender line derivation for the narrow and width-stable characters is illustrated on the right of Figure 21. Though there are a few special characters having neither the normalized height nor vertical center, their semantics is a strong indicator of the possible layouts. For example, the punctuation is in horizontal with its right neighbor, and the prime symbols is attached as the right superscript.

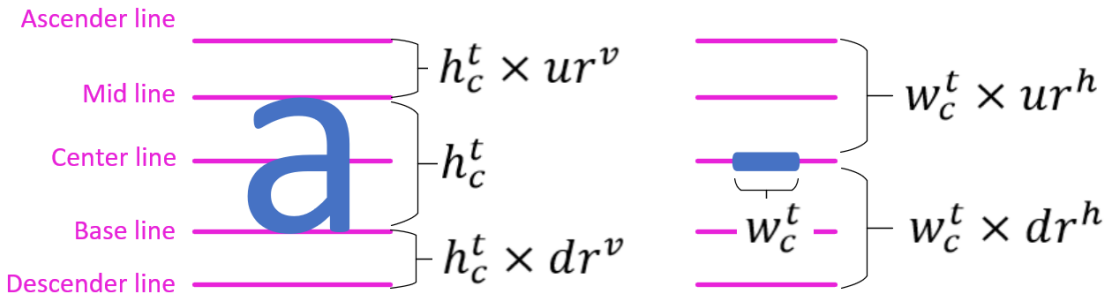



**Figure 21 Recover the ascender/descender line and the normalized height for height stable or width stable characters**

The statistics in Figure 22 show the necessity of categorizing the characters based on their glyphs. The first two columns show the histogram statistics of the vertical adjustment ratio $ur^v$ and $dr^v$ for height stable character "A" and varying size character "sum." The second two columns show the horizontal adjustment ratio $ur^h$ and $dr^h$ for width stable character "-" and speicial character ",." From the statistics, the adjustment ratio for height stable and width stable characters are mostly concentrated in a small region near the peak, showing a distribution like a normal distribution. On the other side, the adjustment ratios for the varying size character summation shows two peaks. The horizontal adjustment ratio for the special characters, comma, show a scattered distribution cover a large value range.

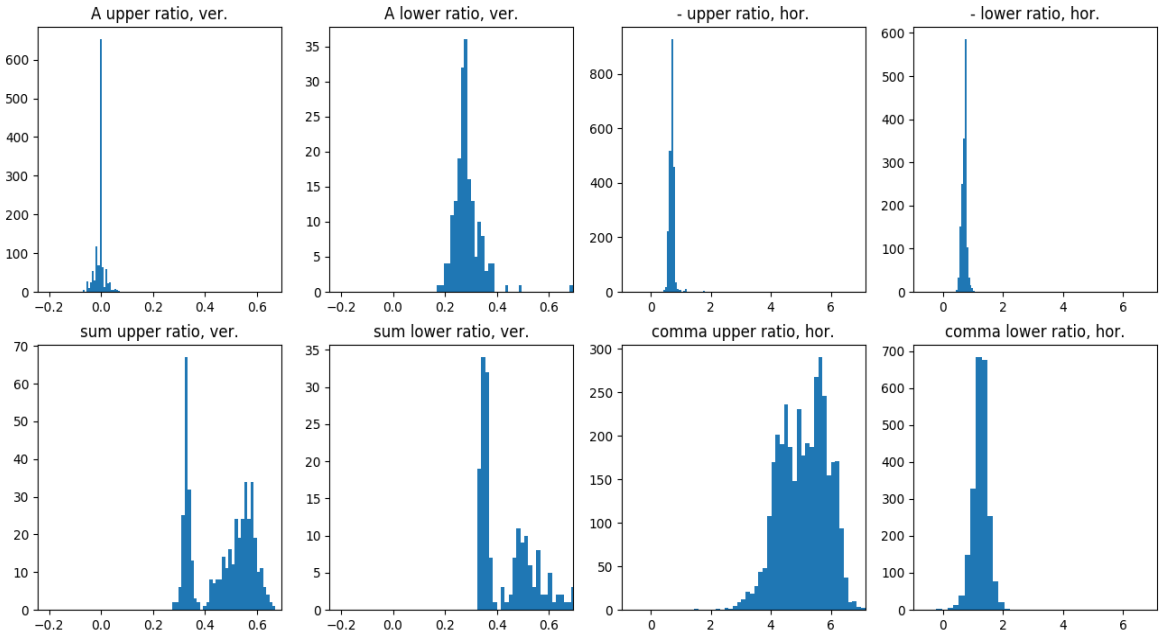



**Figure 22 Histogram of the vertical adjustment upper/under ratio for "A" and "sum" and horizontal adjustment upper/under ratio for "-" and ","**

In summary, after grouping the characters by their glyph type, the difference in the glyph design for different values are normalized. The characters, for which the normalized height and vertical center could be accurately recovered, are enumerated. The recovered normalized height and vertical center are the same as human readers perceive. These observations lay a solid foundation for the later stage of relative spatial relationship assessment.

### III.3 Hierarchical ME layout taxonomy

As MEs are organized hierarchically, a complete enumeration of the possible ME layout structures is the guideline for a systematic solution for the ME layout recovery. In this section, the taxonomy for ME Layout and the common properties for the building blocks are elaborated.

### *III.3.1 ME layout taxonomy*

Different types of ME Blocks as the taxonomy of ME layout are presented using Unified Model Language (UML) in Figure 23. Each rectangle could be an interface if there is a description "<<interface>>" at the top or a class otherwise. The class name or interface is placed on the top, and the member variables and functions are listed in the following rows. Each row is in the format of "name: type," and the parentheses in the name indicate that the line describes a function. The type after the colon of each row indicates the type of a member or the return type of a function.

The ME Blocks are composed of atomic building units such as MESymbol and MEPath. The MESymbol covers all characters, including alphabets, Greeks, operation, relations, and accent. The MEPath are horizontal lines that play as fraction line or top line of a radical structure. Each MEBlock has its members, which are the MEBlocks being dominated. Next, the ME blocks are elaborated based on the processing sequence to be elaborated later. Firstly, the MEAccentBlock, MERadicalBlock, MEFractionBlock, MEBindVarBlock, and MEFenceBlock are structures that could be identified by the particular characters. The second groups of MEBlock are related to the vertical relationship, including the MESupSubBlock and MEUnder/ Upperblocks related with vertical under/upper relation. At last, the MEHorBlock, MESupBlock, and MESubBlock describe with horizontally arranged blocks.

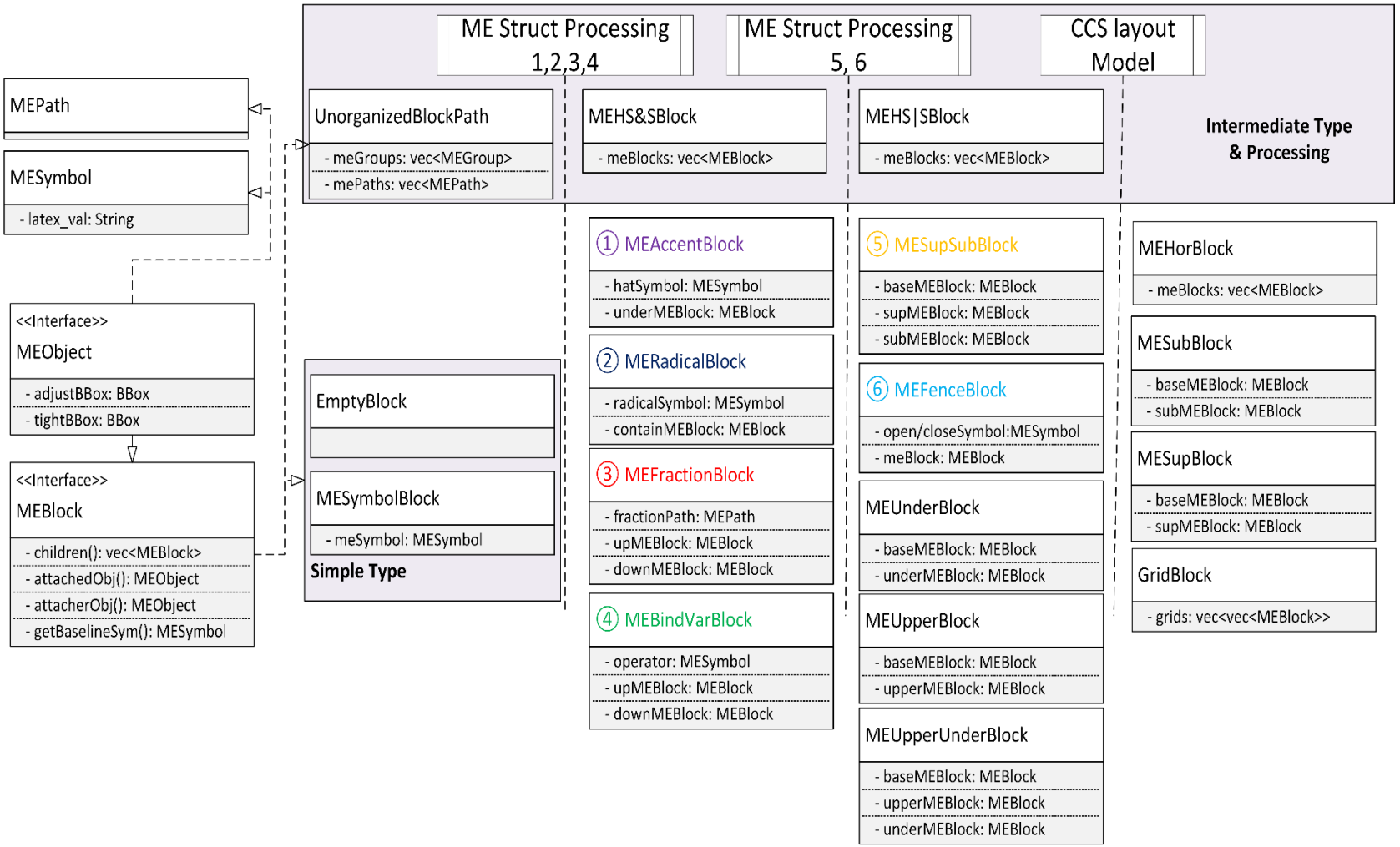
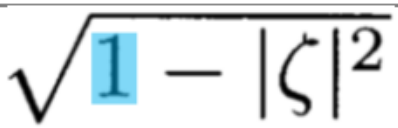
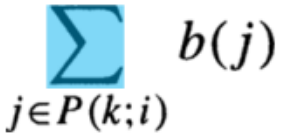
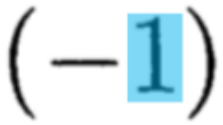
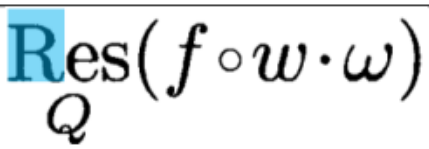
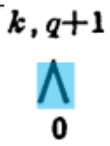



**Figure 23 The taxonomy of ME Layout**

**Table 8 Illustration of ME blocks and the baseline character, parts of this table are adapted from InftyCDB [47]**

| Types | Illustration | Type | Illustration |
|---|---|---|---|
| Accent |  | Radical |  |
| Fraction |  | Bind Op |  |
| SupSub |  | Fence |  |
| Under |  | Upper |  |
| UpperUnder |  | Hor |  |
| Sub |  | Sup |  |

Besides, there are three types of intermediate MEBlock type. The UnorganizedBlockPath is generated in the beginning without any information about the relationship among the children ME blocks. The HS&SBlock might contain an MEBlock with both superscript and subscript, while an MEBlock could only have superscript or subscript in HS|SBlock. Examples of each type of ME Blocks could be found in Table 8. Note that there is a special MEBlock called EmptyBlock. It is used when the accent symbol did not see the expected base part, or the fence did not see the contained part, they should be filled with EmptyBlock.

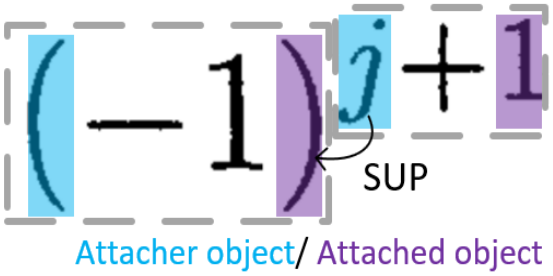



**Figure 24 Illustration of attacher and attached object, parts of this figure are adapted from InftyCDB [47]**

*III.3.2 Common interface for ME layout blocks*

Besides these ME block classes and their members, their common interface will also be presented. The ME Object interface describes common functions about the geometric measures of the tight bounding box and height-adjusted bounding box. The tight bounding box is the minimal rectangle that contains all the pixels of glyphs. When the normalized height of a glyph could be estimated, the adjusted bounding box is obtained with the top and bottom aligned with the ascender and descender line. Extending ME Object interface, the MEBlock is an abstract interface about the common operations/properties that an ME layout structure could have. The interface is illustrated in Figure 24.

- Children: The ability to access all the children is necessary as some transformations are recursively applied to all the children/descenders. For the base ME block "$(-1)$" in Figure 24, it has a child ME Block which is of type MEHorBlock containing two MESymbol, "-" and "1."
- Attacher and attached object. These two concepts are essential to recover the attachment tree structure so that the evaluation could be done against the InftyCDB dataset [47]. An

example is given in Figure 24. There are two MEBlocks where the HorBlock "$j+1$" is the superscript of the FenceBlock "$(-1)$". When recovering the attachment tree is defined at the character level, the attacher object of the superscript MEBlock (character "$j$") is attached to the attached object of the base MEBlock (character ")").

- Baseline character is a very important concept to determine the relative spatial relation among MEBlock according to the height, baseline and center line. The baseline symbol for different types of ME Block is illustrated in Table 8.
  - For the fraction block, a fake MESymbol is created with value "/," Its bounding box is the same size of the primary baseline character but shifted vertically to be centered at the fraction line.
  - The baseline symbol of an accent/radical/fence block is the same as the baseline symbol of the dominated block.
  - For binding variable blocks, the baseline symbol is the binding operator.
  - The baseline symbol of the UpperBlock, UnderBlock, and UpperUnderBlock is the baseline symbol of their baseMEBlock.

### III.4 Two-phase ME layout analysis architecture

In this work, a two-phase architecture is proposed as shown in Figure 25. In the first phase, heuristic rules are applied to identify vertical, enclosed and some horizontal structures, so that the characters are organized into a hierarchical of horizontally arranged blocks. Then, in the second phase, the super/sub-script relationship for the horizontally arrange blocks of each layer in the hierarchy are resolved using the proposed global spatial inference model. As both phases use the character content either as constraints or clues for spatial relation identification, this model is named as content-constrained spatial (CCS) model. These two phases will be elaborated

in detail in the next two sections. One example will be given next to illustrate the processing and the rationality of the execution order.

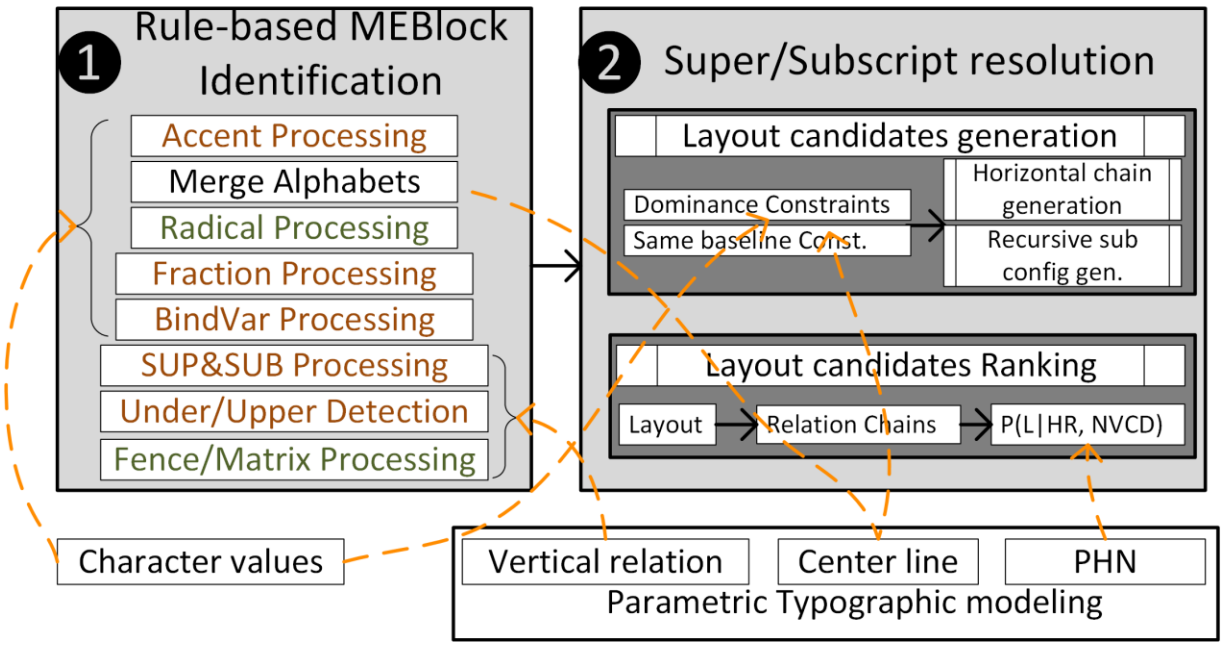



**Figure 25 Two-phase ME layout analysis architecture, parts of this figure are adapted with permission from [108]**

For the example in Figure 16.a, an MESymbolBlock is created for each symbol (marked in grey dashed rectangles) and an MEPath for each horizontal vector graphic line in the beginning. The MESymbolBlocks and MEPaths together form an UnOrganizedBlockPath (UBP). The elements in UnOrganizedBlockPath will be processed sequentially in seven steps to identify the accent, radical, fraction, binding operators, both superscript and subscript, fence, and upper/under structures. After the above mentioned six steps of processing, the UBP is transformed into a hierarchy of horizontally arranged blocks. With each group of horizontally arranged blocks, the only possible relationships between the blocks are same baseline (HOR),

superscript (SUP), and subscript (SUB), which will be resolved through our Content-constrained spatial model to be explained in next section.

The execution order does matter. Another execution sequence might lead to the wrong results. The accent structure processing is adopted as an example.

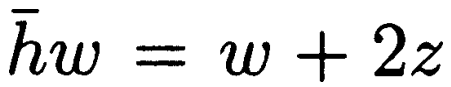


**Figure 26 Merging alphabetic MEHorBlocks after the accent analysis, parts of this figure are adapted from ME 28016825 in InftyCDB [47]**

First, identifying other structures first might hurt the accent structure. If the 'merging consecutive alphabets' procedure is executed before the identification of accent structure, the symbols dominated by accent and symbols not dominated by the accent might be merged such as 'h' and 'w' in Figure 26. But only 'h' belongs to the accent structure.

Second, identifying the accent structure first will not affect the identification of other structure. By the nature of the hierarchical structure, one character will be assigned to only one MEBlock in the hierarchy. And if by further assuming that the procedure to find the dominated blocks of MEAccentBlock is accurate. The way to prove that the accent identification does not hurt other ME structures is as follows: 1) the symbols not belonging to the accent block are not touched so that other structure will not have missing symbols. 2) the symbols belonging to the accent struct are all extracted so that they will not be assigned to other structures.

Note that similar elaborations could be found for accent processing, radical, fraction and consecutive alphabetic HorBlock. There are a few rare cases where the accent symbols are not

used conventionally, which will violate our assumption above. As for the binding variable processing, both superscript and subscript, and general upper/under structure, the decision boundary for their vertical decorative parts is not clear. The evaluation section also confirms with this observation.

## III.5 Rule-based MEBlock identification

The rule-based MEBlock identification targets at the recovery of MEBlocks with indicators, vertically stacked structures, and pre-merging of MEHorBlocks. It consists of sequential processing of seven steps shown to the left of Figure 25. The details of each processing will be given one by one.

### *III.5.1 Accent Structure*

The accent processing is an iterative process described in Figure 28 and illustrated by the example in Figure 27. In each iteration, the smallest accent symbol is identified first, which does not contain other accent symbols horizontally. The list of accent values is predefined as: "acute", "grave", "hat", "tilde", "check", "breve", "overline", "dot", "ddot", "vec", "dddot", "underline", "underbrace". In this example, it is the smaller hat character $c_3$ that is closer to $y$ in the first iteration. After the identification of the smallest accent symbol, $c_3$, the iterativeExpand procedure in Figure 29 is used to find the blocks dominated by the accent symbol based on the following assumptions: 1) The elements dominated by the accent symbol overlap vertically; 2) The dominated blocks should be horizontally overlapping with the accent block. For the hat character $c_3$, the dominated MEBlock is only the MESymbolBlock for $c_4$ of value ‘y’. In the second iteration, the hat character $c_1$ is identified and the dominated blocks includes the MESymbolBlock for open fence $c_2$, the MEAccentBlock $b_1$ constructed in previous iterathe tion, and the MESymbolBlock for the close fence $c_5$.

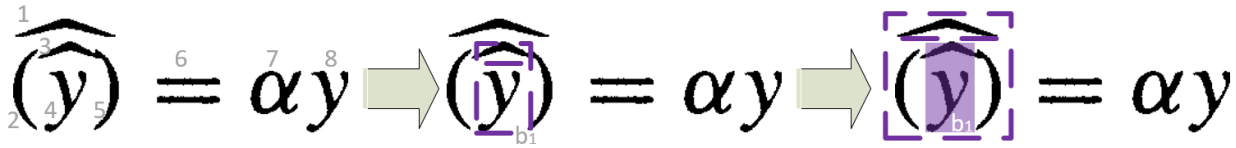


**Figure 27 Illustration of the iterative accent structure analysis, parts of this figure are adapted from ME 28008501 of InftyCDB [47]**

```
input: UnorganizedBlockPath m

Identify the smallest accent symbol block a
if not found:
  return m
if isOverAccent(a): % Group the dominated symbols/paths
  ubp = iterativeExpand(a, m, up)
else:
  ubp = iterativeExpand(a, m, down)
a' = createAccentBlock(a, ubp)
newMbs = (m.mbs \ ubp.mbs) ∪ {a'}
m.mbs = newMbs
re-iterate from line 3
```

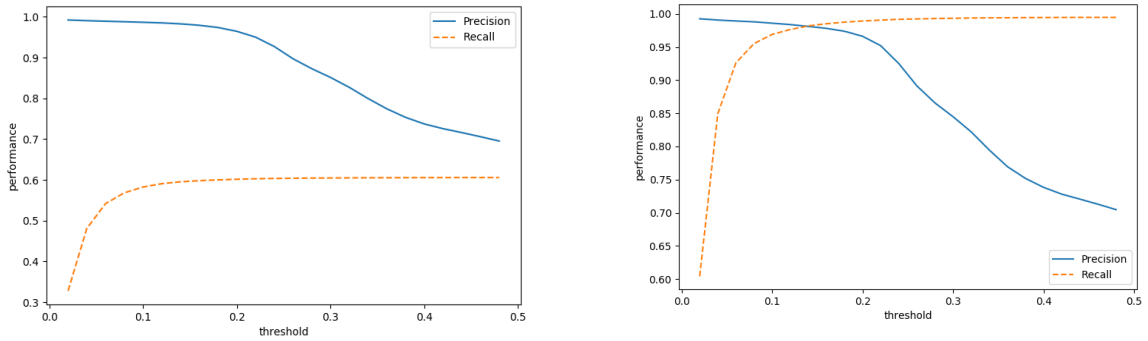
**Figure 28 Accent structure processing**

```
% This is used in the accent, fraction analysis

function iterativeExpand
input: MEObject a, UnorganizedBlockPath m, DirectiontoExpand d

if d == down:
  b* = argmax_{b∈m.mbs, a.over(b)} dist(a,b) % find the nearest under the object
else:
  b* = argmax_{b∈m.mbs, a.under(b)} dist(a,b) % find the nearest above the object
blist = {b*}
bbox = b.bbox
while true:
  b' = {b | a.hOverlap(b) ∧ b.vOverlap(bbox)} % find the block horizontally overlapping
    with symbol and vertically overlapping with existing dominated blocks
  bbox = mergeBbox(bbox, b') % update the bbox of dominated blocks
```

**Figure 29 Iterative expanding procedure**

*III.5.2 Pre-merging of consecutive alphabets on the same baseline*

Some consecutive alphabets placed in the horizontal line, such as the function names 'min', should be merged before the vertical structure analyses. The normalized vertical center difference measurement is used to detect the characters in HOR relationship. The center line, instead of the baseline, is used for such analysis because there are more characters with estimable vertical center compared with the characters aligned with the reference lines to recover the baseline, shown in the typography knowledge section. In this work, two characters $c_i$ and $c_j$ are asserted to have the same center line based on the following criteria: $g_i^c - \eta_i * \alpha < g_j^c < g_i^c + \eta_i * \alpha$, where $\eta_i$ and $g_i^c$ are the normalized height and vertical center of the character $c_i$. This rule is valid subject to the condition HorByCenter, which requires $c_i$ with estimable normalized height $\eta_i$ and $c_j$ with estimable vertical center $g_i^c$.




*(a) all pairs satisfying HorByCenter* *(b) alphabetic characters*

**Figure 30 The tradeoff between the precision and recall for centerline-based analysis**

For the InftyCDB-I dataset [47], all pairs of characters that should lie on the same baseline are gathered first. The precision for the identified pairs lying on the same baseline is

drawn in blue curve against the threshold $\alpha$ in Figure 30. The corresponding recall is in the dashed orange curve. When the threshold gets larger, more same baseline pair could be discovered, but the precision degrades very faster. The recall rate reaches a plateau of 0.6 after $\alpha > 0.2$. The plateau is reached because of our rule is applied when the condition HorByCenter is satisfied. Though only covering 0.6 of all pairs, it is much better than alphabets pairs only, which only occupy about 26% of all pairs. This is very important for our later stage analysis of content-constraint HOR/SUB/SUP discrimination. When considering alphabets only, this rule could achieve a high precision and recall 0.97 at the same time as shown in Figure 30.b.

### *III.5.3 Fraction*

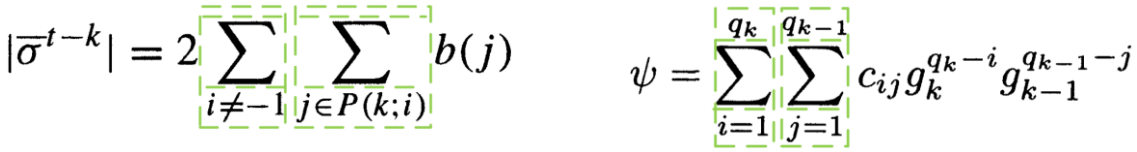
The fraction processing procedure in Figure 31 is similar to the accent processing. The only difference is that, given an identified fraction line with the smallest horizontal span, the iterative expansion in Figure 29 should be conducted for both the upper and under part.

```
input: UnorganizedBlockPath m

Identify the smallest horizontal path p
if not found:
    return m
upperUBP = iterativeExpand(p, m, up)
underUBP = iterativeExpand(p, m, down)
f = createFractionBlock(p, upperUBP, underUBP)
newMbs = (m.mgs ∖ (upperUBP.mbs ∪ underUBP.mbs)) ∪ {f}
m.mbs = newMbs
re-iterate from line 3
```

**Figure 31 Fraction structure processing**

*III.5.4 Big operator structure*

The binding operation processing here mainly refers to the binding operator with under and/or upper part as shown in Figure 32. If the scope of the binding operation is manifested as super/subscript, it will be processed in the later stage processing of both superscript and subscript.



(a) Binding operator with under parts, ME 28008168

(b) Binding operators with both upper and under parts, ME 28004533

**Figure 32 Example of big operator structures, parts of this figure are adapted from InftyCDB [47]**

The bind operation processing constructs a BindVarBlock for each big operator, together with the horizontally overlapping component as UBPs over and under it, such as the example in Figure 32.b. One particular situation is the consecutive binding operation with upper/under parts exceeding the horizontal range of the binding operator as the example in Figure 32.a shows. Currently, our solution is to treat the consecutive binding operator as a whole to discover the upper and under parts. Then the characters between the binding operators are segmented based on the largest gap, such as the gap between "1" and "j" in Figure 32.a.

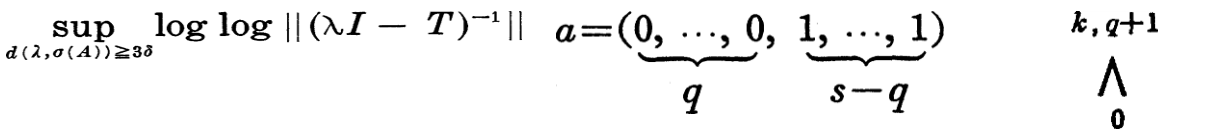
### *III.5.5 Fence, matrix, piecewise processing*

The paired fences are strong indicators both at the layout level and the semantic level. At the layout level, the left fence symbol is in a horizontal relationship with its right direct neighbors and the right fence symbol. More beneficially, it could divide a long structure into smaller units, thus reducing the computation complexity. The fence characters considered in this work include parenthesis "()", square bracket "[]", curve bracket "{}", and vertical bar "|." Note that there might be nothing between the paired fence.

After the identification of the paired fence, the content in the fence might be just one MEs or multiple MEs such as a matrix and a vertical vector. For the unmatched fence starting with curve bracket, it could be the piecewise ME with different values under different conditions. To detect the grid of elements in matrix or lines in the piecewise ME, projective-profile cutting technique is used to detect the vertical overlapping and horizontal overlapping region.

### *III.5.6 Element with both superscript and subscript*

After the previous processing, an MEBlock might still be associated with both the super- and subscript components. Structures with both super and subscript are identified first to reduce the complexity for the super/subscript resolution. Both sup/sub structure identification run in iterations. In each iteration, UBP in the existing MEBlock hierarchy is recursively traversed and processed. Within each UBP, the first MEBlock $s_b$ are located with two direct right up or down MEBlocks $s_u$, $s_d$ that do not overlap vertically. The superscript parts $s_u$ is expanded with vertically overlapping MEBlocks on its right that does not overlapping with $s_d$. Similar processing is applied to the subscript part. Each expansion step will create an UBP, and together with the base MEBlock $s_b$, they will construct an SSB. The process terminates when no SSB can be generated from an iteration.




*(a) function decorator* *(b) Under accent decorator* *(c) Operator decorator*

**Figure 33 The semantics related with upper/under structure, parts of this figure are adapted from InftyCDB [47]**

Example of general upper/under relationship is shown in Figure 33. The under and over parts might play as the decorator of the function, operators, or accent. The procedure to recover the upper/under structure is shown in Figure 34. It is an iterative procedure until there is no upper/under structure. First, among all sub MEblocks under $m$, an MEBlock $m.mbs[i]$ is identified to horizontally overlapping but not vertically overlapping with the next. Then, the upper and under part are expanded based on vertically overlapping. The next step will decide which part is the base and which is the decorator mainly based on two clues. The first clue is that some indicator such as function name or operator are the base part. The second clue is that the characters on the same baseline with the neighbors are the base part. If there are no special indicators, the default option is to choose the under part as the decorator.

```
Input: MEBlock m

sort m.mbs by left boundary
i = min{i'|m.mbs[i'].hOverlap(m.mbs[i'+1])∧!m.mbs[i'].vOverlap(m.mbs[i'+1])}
If i not exist:
    return m
upBlock, downBlock = gather_under_upper()  % gather the upper and under
if upUBP on the baseline:  % determine which is on the baseline
    m' = createUnderBlock(base=upBlock, down=downBlock)
else:
    m' = createUpperBlock(base=downBlock, up=upBlock)
m = HS|SB(m.mbs \ (upBlock.mbs ∪ downBlock.mbs) ∪ {m'}) % create new
Re-iterate the current procedure.
```

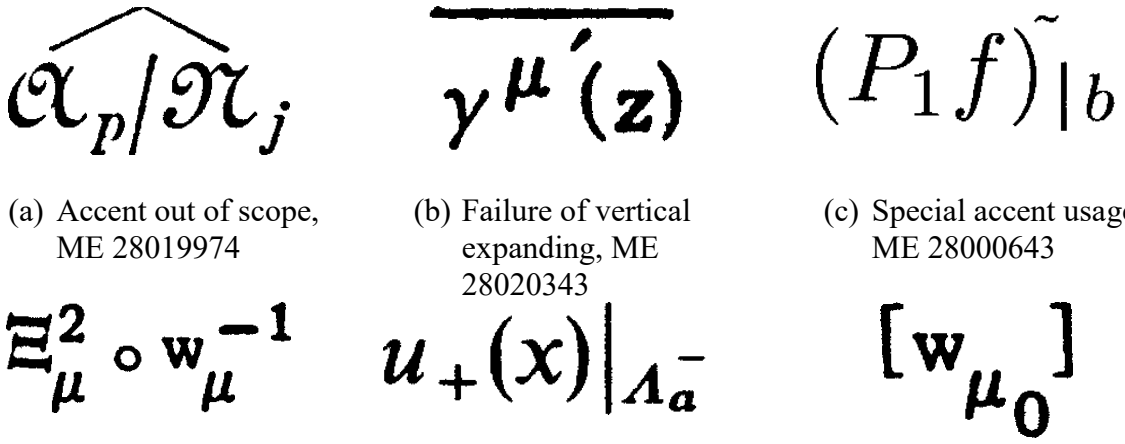
**Figure 34 General upper/under structure detection**

*III.5.8 Performance of non-horizontal structure identification*

**Table 9 Performance of Non-horizontal structure analysis**

| Structure Types | Accent | Fraction | Radical | Binding Op. | SupSub | Under/Upper |
|---|---|---|---|---|---|---|
| CaseCount | 2648 | 1110 | 37 | 1096 | 2226 | 202 |
| TruePositive | 2514 | 1100 | 35 | 1047 | 1980 | 156 |
| FalsePositive | 149 | 37 | 2 | 156 | 188 | 107 |
| FalseNegative | 134 | 10 | 2 | 49 | 246 | 46 |
| F1 | 0.947 | 0.979 | 0.946 | 0.911 | 0.901 | 0.671 |

The performance of non-horizontal structure identification is shown in Table 9. The performance for accent, fraction, and the binding operator is very accurate. There are still some errors in the identification of structure with both super- and sub-scripts. The recognition of general under/upper relation is still challenging due to the difficulty in identifying the right scoping of the multi-character variable/function names.



(a) Accent out of scope, ME 28019974

(b) Failure of vertical expanding, ME 28020343

(c) Special accent usage, ME 28000643

(d) Failure of the stop-expanding condition for both sup and sub, ME 28020495

(e) Minus sign on the superscript, ME 28018421

(f) Over shifting, ME 28020198

**Figure 35 Cases study for the rule-based MEBlocks identification, parts of this figure are adapted from InftyCDB [47]**

The error for accent structure analysis is mainly in two aspects: the mismatch of the horizontal scope of the accent characters and the special usage of the accent characters. For the first aspect, the first situation is that some characters might not fall under the scope of the accent character as shown in Figure 35.a. The second situation is that the accent characters might be too large to overlap with others. For the second aspect, some accent is used standalone in the superscript in Figure 35.c. For fraction errors, the errors mainly happened during the expanding process as shown in Figure 35.b. For the binding operator, both superscript and subscript, and the general under/upper structure, the primary challenge is the uncertainty of their horizontal scope. There are no reliable clues on when to stop expanding. Further, the noise of the over shifting and the special glyph characters also affect the assessment of the relative position based on the overlapping of the bounding box from the vertical or horizontal direction, as shown in Figure

35.d and Figure 35.f. The big integral operator which slant to its right also cause the failure of the detection of both super- and sub-script structure.

### III.6 Global inference for super/subscript resolution

After the rule-based ME layout structure identification, the characters of the ME are organized into a hierarchical structure of horizontally arranged ME blocks as shown in Figure 36. The only relationships between characters left are the horizontal (HOR), superscript (SUP), subscript (SUB) or the stacking of these relationships.

Existing works focused on the classification of HOR/SUP/SUB, but the error from local greedy decisions will propagate to neighbors. Further, the relationship between consecutive characters could be very complex beside the three relationships mentioned above, such as inverse superscript. To avoid the local error, a global inference is used for the analysis of the horizontally arranged MEblocks at each layer of the hierarchy. Then a probabilistic ranking that could cover the long-distance dependency is presented to find the candidate with the largest probability satisfying the constraints.

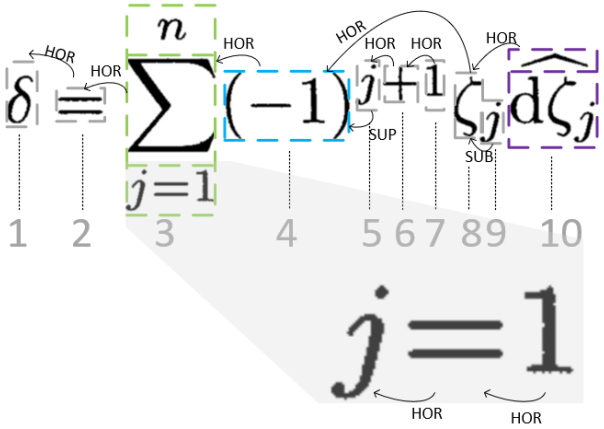



**Figure 36 Intermediate results after the rule-base ME layout structure analysis**

*III.6.1 Searching Space Enumeration*

Formally, for a horizontal chain arranged blocks $B = \{b_i\}$, each possible horizontal layout structure could be described as an $n-1$ triples, $L = \{\langle i, i_p, r_i \rangle\}$, where $i$ and $i_p$ are the index for ME blocks and $r_i \in \{H, SUP, SUB\}$ is the relative position between the $b_i$ and its sibling $b_{i_p}$. But not all such triples are valid horizontally layout, they should follow the four axioms and the content-based constraints introduced below. After the possible constraints are elaborated, the enumeration procedure to generate the ME layout candidates is presented.

It is assumed that there are no left super/subscript relationships. First, the left relationship is rare. A statistic over the InftyCDB dataset shows that there are only 12 MEs with left relationships out of 20K samples. Second, the left super/subscript could also be viewed as the right super/subscript if there is no global information, which could be handled by our existing procedure. Third, even if the left super/subscript is at the beginning, errors introduced is expected to be detected by our post checking modules, which is elaborate at the end of the experiment section.

**III.6.1.1 Four axioms and constraints**

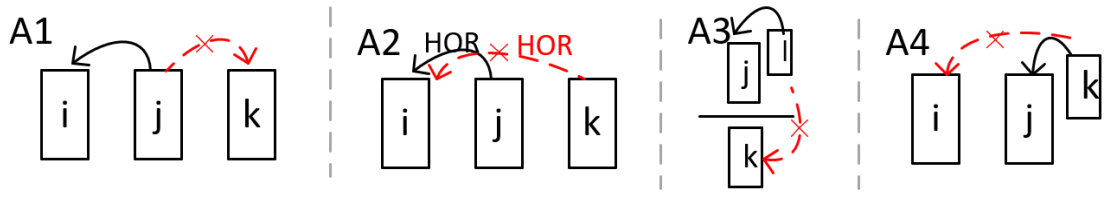



**Figure 37 Four axioms about the layout of horizontally arranged ME blocks, reprinted with permission from [108]**

The triples to describe horizontally arranged blocks satisfy four axioms based on writing convention as illustrated in Figure 37.

- Axiom A1 (OneSibling): Each ME block $b_j$ can only be attached to one $b_i$ on its left, i.e., $\forall\langle i, i_p r\rangle \in L, i_p < i$.
- Axiom A2 (OneSiblingRel): $\forall j \in [1, n], r \in \{H, SUP, SUB\}, !\exists i \neq i', \langle j, i, r\rangle \in L \wedge \langle j, i', r\rangle \in L$.
- Axiom A3 (VerOverlap): Each ME block $b_i$ should vertically overlap with $b_{i_p}$ based on the typesetting convention of superscript, subscript, or baseline.
- Axiom A4 (NoSkipScript): If $r \in \{SUP, SUB\}$, then there are no other MEBlocks between the $b_i$ and $b_{i_p}$ horizontally, i.e., $i_p = i - 1$.

Besides these axioms, the constraints are summarized below based on the symbol dominance and spatial relationship:

- A MESymbolBlock $b_i$containing a relation symbol or punctuation must be in a horizontal relationship with its right neighbor block $b_{i+1}$.
- A MESymbolBlock $b_i$containing a relation symbol or punctuation must be in a horizontal relationship with one of its left neighbor blocks $\{b_1, \dots, b_{i-1}\}$.
- Pairs of MEBlocks $\langle b_i, b_j\rangle$ with their baseline characters $c_i$ and $c_j$ satisfying the same center-line checking should be in HOR relationship.

The constraints mentioned above could all be represented as $\langle i, [j, k], t\rangle$, where $b_i$ is the ME block that triggers the constraint, the blocks $\{b_j, \dots, b_k\}$ are constrainted. The constraint type $t$ could be $HOR_{MUST}$ or $HOR_{EXIST}$.

### III.6.1.2 Enumeration Procedure

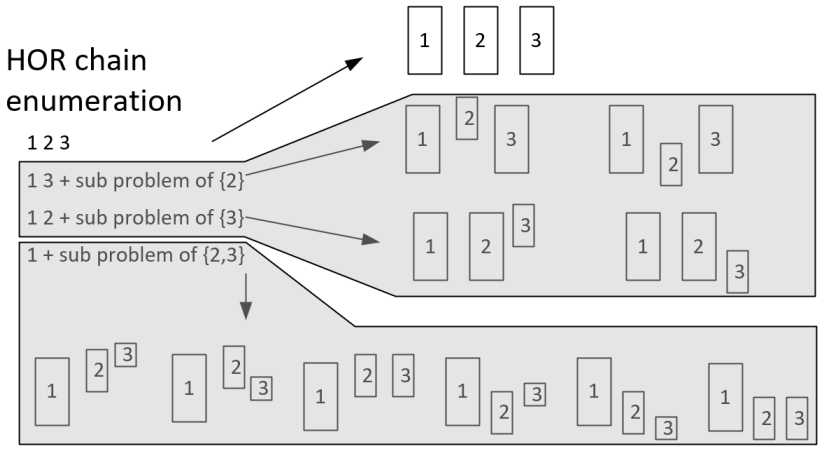



**Figure 38 Enumeration of the ME layout for horizontally arrange blocks**

```
EnumLayoutCand(B = {b_1, ..., b_n}, ℂ)

for hc = {i_1, ..., i_K} from combinatorial enumeration of [1, n]:
  skip if not ConstrainSat(hc, ℂ)
  SubLayoutCandsList = []
  for [i_k + 1, i_{k+1} − 1] with i_{k+1} − i_k > 1:
    ℂ′=CreateLocalConstraint(ℂ)
    SubLayoutCands = EnumLayoutCand({b_{i_k+1}, ..., b_{i_{k+1}−1}}, ℂ′)
    SubLayoutCandsList.push( {(k, L)|L ∈ SubLayoutCands} )
  SubLayoutCandProdSpace = SubLayoutCandsList_1 ×⋯× SubLayoutCandsList_K
  for SubLayoutCandProd in SubLayoutCandProdSpace:
    L = {⟨i_{k+1}, i_k, HOR⟩}
    for k, SubL in SubLayoutCandProd:
      shift SubL by index i_k and add to L
      add {⟨i_k + 1, i_k, SUP|SUB⟩} to L
      yield L
```

**Figure 39 Horizontal layout candidate enumeration**

The layout enumeration procedure in Figure 39 is design to find all possible layout among the blocks, where each layout could be formally represented as a triple set $\{\langle i, i_p, r_i\rangle\}$. The

input to the procedure is the MEBlocks $B = \{b_1, \dots, b_n\}$ to resovle the HOR/SUP/SUB relationship and the constraint set $\mathbb{C}$ is generated by the rules in the previous section. The procedure starts with a combinatorial enumeration of range $[1, n]$. The index of the selected blocks, $hc = \{i_1(= 1) < i_2 < \cdots < i_K \leq n\}$, are the blocks that lies on the main baseline, which will be tested through ConstrainSat($hc$, $\mathbb{C}$) before further processing. For the example in Figure 38, suppose that block 2 is a relation symbol, which must be in a horizontal relationship with its right neighbor and one of the elements in the left neighbor. This makes the $hc$ combinations $\{1\}$, $\{1,2\}$, and $\{1,3\}$ invalid. Given one $hc$, there are sub ranges between the elements in $hc$. For each sub range $[i_k + 1, i_{k+1} - 1]$ with at least 1 element, a local constraint set $\mathbb{C}'$ is created and enumeration is conducted on the local subrange. The layout enumeration for each sub range $\text{SubLayoutCandsList}_i$ will be merged together through set production to create the full enumeration space SubLayoutCandProdSpace. For each possibility in the product space, each sub range will be shifted by its starting index, and the subrange will attach to the original sequence as either subscript or superscript component. For the example in Figure 38, if $hc = \{1\}$, the enumeration will be applied on the range $[2,3]$. The layout candidates $23$, $2^3$, and $2_3$ could be attached to the base block 1 as either superscript or subscript, resulting in a total of 6 possibilities.

The search space of possible layout candidates $L(B)$ for a $B$ with $n$ blocks is $\Theta(2^n)$. To reduce the labeling search space, a simple heuristic partition technique is proposed to split blocks into two set of blocks, $\{b, \dots, b_{m-1}\}$ and $\{b_m, \dots, b_n\}$, where $b_m$ is the block with the largest height. Empirically it was found 98% of them were non-super/subscript and anchored on the same baseline as the first block.

*III.6.2 Global probabilistic inference and features*

For each $B = \{b_1, \dots, b_n\}$, the horizontal layout is found by optimizing:

$$arg\,max_{L \in \mathbb{L}(B)} \prod_{i<j} P\,(O_{ij}|RC_{ij}^{L})$$

where $\mathbb{L}(B)$ is generated by the layout candidate enumeration procedure, and $RC_{ij}^{L}$ is the relation chain between $b_i$ and $b_j$ under the horizontal layout $L$. The relation chain is constructured by finding the path between the blocks $b_i$ and $b_j$, $\{i = k_1, \dots, k_m = j\}$ so that $r_i$ is appended to $RC_{ij}^{L}$ if $\langle k_i, k_{i+1}, r_i \rangle \in L$ or $REV(r_i)$ is appended to $RC_{ij}^{L}$ if $\langle k_{i+1}, k_i, r_i \rangle \in L$, where the $REV$ denote the reverse of the spatial relation. For the example $a^{b_c}$, $c$ and $a$ are in relation chain of $[SUP, SUB]$.

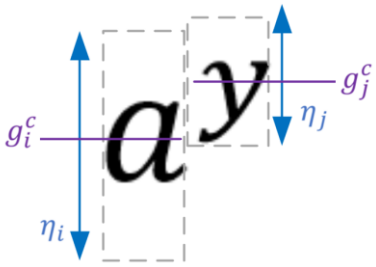


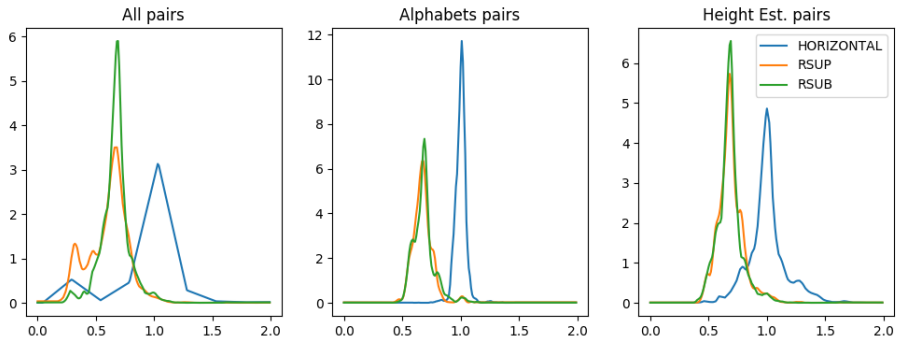

**Figure 40 Features for inference of super/subscript**

As the only possible relationship between ME blocks are HOR/SUB/SUP after the rule-based processing, two simple and powerful features are adopted here: the height ratio $\Phi_{ij}$ and the normalized vertical center different $\Psi_{ij}$ to capture the relative spatial relationship between MEBlock $b_i$ and $b_j$ as illustrated in Figure 40:$\Phi_{ij} = \frac{\eta_j}{\eta_i}, \Psi_{ij} = \frac{g_j^c - g_i^c}{\eta_i}$, $\eta_i$ and $g_i^c$ are the normalized height and vertical center of the baseline character $c_i$ of MEBlock $b_i$. However, not

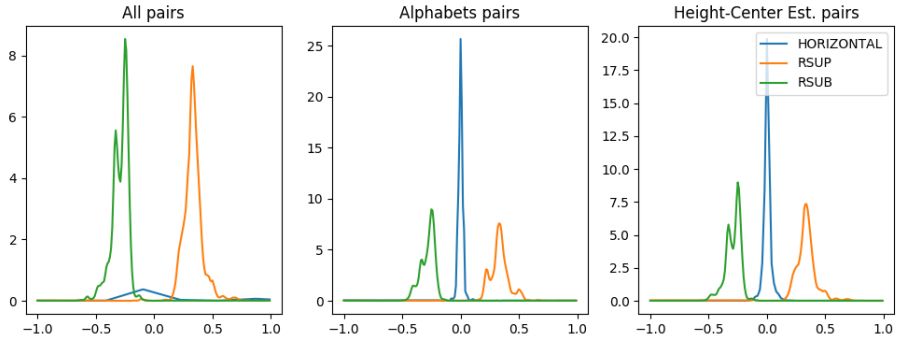
all character could reliably estimate the normalized height/vertical center. Based on the characteristics of the related characters:

$$P(O_{ij}|RC_{ij}^{L}) = \begin{cases} P(\Phi_{ij}|RC_{ij}^{L})P(\Psi_{ij}|RC_{ij}^{L}) & condition1 \\ P(\Psi_{ij}|RC_{ij}^{L}) & condition2 \\ 1 & otherwise \end{cases}$$

, where 1) condition 1 is satisfied if the normalized height of both $c_i$ and $c_j$ could be estimated; 2) condition 2 is satisfied if the normalized height of $c_i$ could be estimated and only the vertical center of $c_j$ could be estimated.




*(a) Height ratio comparison for all pairs, alphabets pairs, and height-estimable pairs*




*(b) Normalized vertical center difference distribution comparison for all pairs, alphabets pairs, and condition 2*

**Figure 41 Feature distribution for different relations with/out filtering**

The reasons to have different features based on the character values could be explained by Figure 41. For both HR and NVCD features, when only considering the alphabets pairs, the feature distributions for HOR/SUP/SUB show distinct patterns and rarely overlap. But when considering all pairs without differentiating the character values, there is quite a large area of overlapping, which will harm the classification or inference. When filter the character based on condition 1, the overlapping of the HR feature distribution for different spatial relationship below the value of 0.5 is gone. When filter the characters based on condition 2, the probability density function is even similar to that of the alphabets pairs. In summary, by applying the condition 1 and condition 2 to filter based on character values, the discriminating power of the feature is improved, and more characters are covered in comparison with alphabets only.

Note that the way $P(O_{ij}|RC_{ij}^{L})$ only depends on the character values $c_i$ and $c_j$. The number of multiplied items for $\prod_{i<j} P(O_{ij}|RC_{ij}^{L})$ will be the same, thus will not lead to the bias problem.

*III.6.3 Parametric modeling of Height ratio and Normalized vertical center difference (PHN)*

The global inference formation is powerful to capture the long-distance dependency. However, for practical purpose, it is necessary to efficiently calculate the likelihood of the HR feature $P(\Phi_{ij}|RC_{ij}^{L})$ and the likelihood of the NVCD feature $P(\Psi_{ij}|RC_{ij}^{L})$. The challenge for the likelihood estimation comes from the varieties of possible relation chains. One possible solution is based on simulation and non-parametric density estimation as shown in Figure 42. The probability density functions of the feature for the smaller sub relation chains $RC_{ik}^{L}$ and $RC_{kj}^{L}$ will be used to generate samples randomly to calculate the feature value $\phi_{ij}$ and $\psi_{ij}$ for larger relation chains $RC_{ij}^{L}$. Then all the generate samples $\{\phi_{ij}\}$ and $\{\psi_{ij}\}$ are feed into the non-

parametric kernel density estimation to get $\widehat{P}\left(\Phi_{ij}|RC_{ij}^{L}\right)$ and $\widehat{P}\left(\Psi_{ij}|RC_{ij}^{L}\right)$. But this way is computational inefficient.

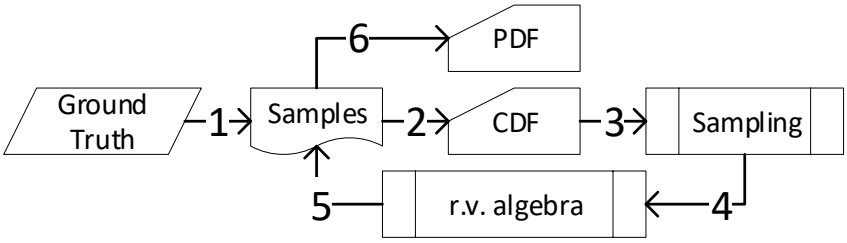
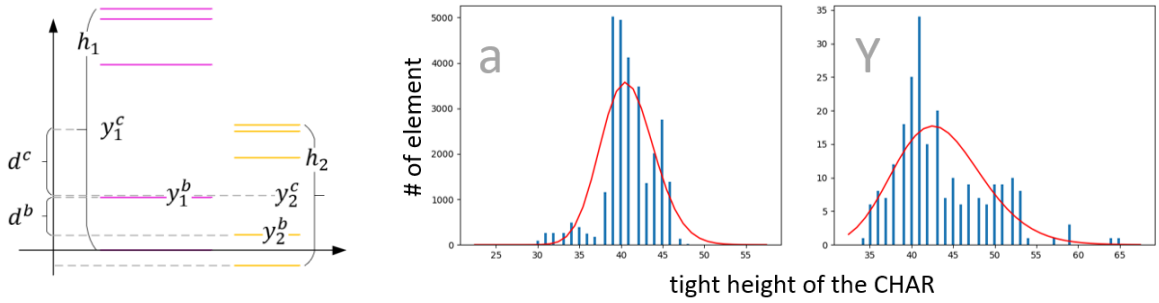



**Figure 42 Non-parametric estimation of the likelihood, reprinted with permission from [108]**




(a) Subscript example (b) lognormal distribution of the character height

**Figure 43 Relative sizing and baseline shifting of super/subscript**

An approximate Parametric model of Height ratio and Normalized vertical center difference (PHN) model is proposed to overcome the computational cost of the non-parametric density estimation,. The approximation model is based on two observation: the lognormality of the height of characters (in Figure 43.b) and the relative sizing and shifting of the super/subscripts (in Figure 43.a). Given these two assumptions, the HR and NVCD features are found to conform with lognormal distribution through linear transformations.

### III.6.3.1 Relative size and baseline shifting of superscript and subscript

Based on the description in [109], the size and baseline of the super/subscript could be modeled relatively concerning the base characters as illustrated in Figure 43.a. The first group of parameters is related to the characters $c_i$ in a PDF file:

- $y_i^c$, $y_i^b$, and $y_i^d$ denote the theoretical center position, the baseline, and the descender line, respectively of $c_i$. The random variable (r.v.) g denotes the corresponding observed value of $y$, e.g., $g_i^c$ denotes the observed center position.
- $h_i$ represents the normalized theoretical height for the difference between the ascender line and the descender line. The r.v. $\eta_i$ denotes the observed $h_i$. Histograms (in blue) as well as the fitted lognormal probability density curve (in red) on the height for glyphs of character values 'a' and 'Y' in InftyCDB-1 are plotted in Figure 43.b. The figures suggest that the normalized height could be approximated as a lognormal distribution skewed to the left, i.e., $\eta_i \sim \mathcal{L}(\mu_i, \sigma_i^2)$, where $\mu_i = log(h_i)$ and $\sigma_i$ are related to the glyph design of $c_i$.

  Other needed parameters are related to the document preparation system [109]:

- $\delta_{SUB}/\delta_{SUP}$: The drop-off and the raise-up ratio of the baseline of SUB/SUP. $\delta_{HOR} = 0$.
- $\theta_b$: The ratio of the baseline-descender difference $(y_i^b - y_i^d)$ with respect to $h_i$.
- $\gamma_{SUB/SUP}$: The ratio of the theoretical normalized height of SUB/SUP w.r.t the normalized height of the base character. $\gamma_{HOR} = 1$.
- When a character $c_j$ is the SUB/SUP of $c_i$, there baseline difference $d_{i,j}^b = y_j^b - y_i^b = \delta_{\#} h_i$, and $h_j = \gamma_{\#} h_i$, where the relation $\# \in \{SUB, SUP\}$.

**Table 10 Parameter estimation of the RSBS**

| | $\delta_{SUP}$ | $\delta_{SUB}$ | $\theta_b$ | $\gamma_{SUB}$ | $\gamma_{SUP}$ |
|---|---|---|---|---|---|
| Mean | 0.434 | -0.192 | 0.206 | 0.7 | 0.7 |
| Std. | 0.065 | 0.07 | 0.05 | 0.126 | 0.106 |

Different document processing systems use different parameters in the rendering of the super/subscript [109]. Assuming the ME are produced using the same document preparation system, the parameter estimation for InftyCDB-I dataset [47] is shown in Table 10. Note that $\gamma_{SUB}$ and $\gamma_{SUP}$ appeared to be consistent with the setting of the Latex system and $\delta_{SUP}$ and $\delta_{SUB}$ does not match any of the parameters reported in the literature.

The only left parameter is the standard derivation for the lognormal distribution of the observed normalized height for each font size. We build a toy document with five paragraphs of the font size 8, 9, 10, 11, and 12, to gain some insights on the relation between the estimated standard derivation and mean of the lognormal distribution. However, no apparent linear relationship could be observed based on the mean and std. $\sigma_i$ of the lognormal distributions. As such, the median value is taken the lognormal std. for each character value in InftyCDB-I dataset, which is 0.097, for $\sigma_i$ of all characters.

### III.6.3.2 Overview of the parametric derivation of $\boldsymbol{P(\Phi_{ij}|RC_{ij}^{L})}$ and $\boldsymbol{P(\Psi_{ij}|RC_{ij}^{L})}$

With the knowledge about the typography system and the relative sizing and baseline shift for the super/subscripts, the process to derive the approximate probability density function of the feature HR and NVCD for a pair of character $(c_i, c_j)$ with a relation chain $RC_{i,j} = \{RC_{i,j,k}\}_{k=1,\dots,K_{ij}}$ is illustrated in Figure 44.

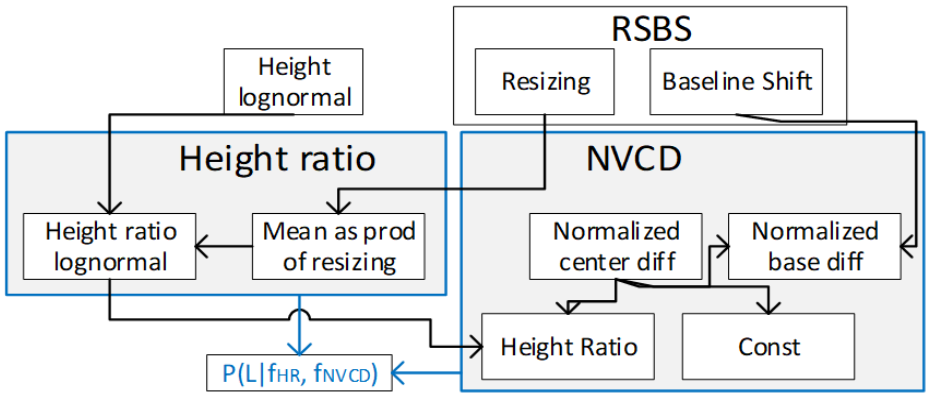



**Figure 44 Outline for the inference process for the PHN model**

Succinctly put, the height ratio $\Phi_{ij}$ conforms to lognormal distribution because the height of each character is assumed to follow lognormal distribution and the ratio of lognormal distribution is still a lognormal distribution. The NVCD $\Psi_{ij}$ also follows lognormal distribution because it could be expressed as a linear transformation to the height ratio feature approximately. The constant factor and the relative baseline shift are derived from the RSBS parameters given the relation chain $RC_{ij}$. As such, the likelihood of both features could be calculated efficiently.

### III.6.3.3 Derivation of the likelihood of height ratio $\boldsymbol{P(\Phi_{ij}|RC_{ij}^{L})}$

HR for the pair $(c_i, c_j)$ is defined as $\Phi_{i,j} = \eta_j/\eta_i$. Given the r.v. of the observed normalized height of $c_i$, $\eta_i \sim \mathcal{L}(\mu_i, \sigma_i^2)$, and the normalized height of the second character $\eta_j \sim \mathcal{L}(\mu_j, \sigma_j^2)$, we have $log(\eta_i) \sim \mathcal{N}(\mu_i, \sigma_i^2)$ and $log(\eta_j) \sim \mathcal{N}(\mu_j, \sigma_j^2)$. The scripted notation $\mathcal{L}$ and $\mathcal{N}$ indicates the lognormal distribution and the normal distribution respectively. By the definition of $\Phi_{i,j}$, $log(\Phi_{i,j}) = log(\eta_j) - log(\eta_i)$. Knowing that addition of 2 normal distributions produces another normal distribution with the mean and variance as the sum of the two original distributions, $log(\Phi_{i,j}) = \mathcal{N}(\mu_j - \mu_i, \sigma_i^2 + \sigma_j^2)$ or $\Phi_{i,j} \sim \mathcal{L}(\mu_j - \mu_i, \sigma_i^2 + \sigma_j^2)$.

As elaborated in the parameter estimation section, the same variance is chosen for all characters. For the mean difference $\mu_j - \mu_i$, the simple case of a character $c_i$ and its parent $c_{i_p}$ with parential relationship $r_i$ is considered first. Then, the result is generalized to the case where the characters $c_i$ and $c_j$ satisfy relation chain $RC_{ij}$.

Given a relation label $r_i$ between $c_i$ and its parent $c_{i_p}$, $h_i = \gamma_{r_i} h_{i_p}$ from the RSBS model, where $\gamma_{r_i}$ denotes the HR of $c_i$ and $c_{i_p}$ . By plugging in $h_i = exp(\mu_i)$, $e^{\mu_i} = e^{\mu_{i_p}} * \gamma_{r_i}$ , or $\mu_i - \mu_{i_p} = log(\gamma_{r_i})$ . This also implies that the mean of distribution for HR $\Phi_{i_p,i}$ should only be tied to the RSBS parameter $log\ \delta_{r_i}$.

Next, the distribution derivation for $c_i$ and $c_j$ satisfying relation chain $RC_{ij}$ is elaborated. For convenience, the index sequence of the characters on $RC_{ij}$ is $\{\mathbb{i}_{i,j,1} = i, \dots, \mathbb{i}_{i,j,K_{ij}+1} = j\}$, where $\langle \mathbb{i}_{i,j,k+1}, \mathbb{i}_{i,j,k}, RC_{i,j,k} \rangle \in HL$ or $\langle \mathbb{i}_{i,j,k}, \mathbb{i}_{i,j,k+1}, REV(RC_{i,j,k}) \rangle \in HL$. Then,

$$\mu_j - \mu_i = \mu_{\mathbb{i}_{i,j,K_{ij}+1}} - \mu_{\mathbb{i}_{i,j,1}} = \sum_{k=1}^{K_{ij}} \mu_{\mathbb{i}_{i,j,k+1}} - \mu_{\mathbb{i}_{i,j,k}}$$

. By the fact that $\mu_{\mathbb{i}_{i,j,k+1}} - \mu_{\mathbb{i}_{i,j,k}} = log(\gamma_{RC_{i,j,k}})$ for $c_{\mathbb{i}_{i,j,k}}$ as the parent of $c_{\mathbb{i}_{i,j,k+1}}$ with relation $RC_{i,j,k}$, we would have

$$\mu_j - \mu_i = \sum_{k=1}^{K_{ij}} log(\gamma_{RC_{i,j,k}})$$

, which implies that:

$$\Phi_{i,j} \sim \mathcal{L}(\sum_{k=1}^{K_{ij}} \log(\gamma_{RC_{i,j,k}}), \sigma_i^2 + \sigma_j^2)$$

### III.6.3.4 Derivation of the likelihood of normalized vertical center difference $P(\boldsymbol{\Phi_{ij}}|\boldsymbol{RC_{ij}^{L}})$

The NVCD feature for $(c_i, c_j)$ is defined as $\Psi_{i,j} = (g_j^c - g_i^c)/\eta_i$. From the RSBS model, we have $y_i^c = y_i^b - h_i\theta_b + h_i/2$. By replacing the theoretical '$h/y$' in to observation random variable '$\eta/g$', and assuming the observed baseline $g_i^b = y_i^b$, an approximation heuristic rule is obtained that $g_i^c \approx y_i^b - \eta_i\theta_b + \eta_i/2$ when $h_i$ is replaced by $\eta_i$. Then,

$$\Psi_{i,j} \approx \frac{\left(y_j^b - \eta_j\theta_b + \frac{\eta_j}{2}\right) - \left(y_i^b - \eta_i\theta_b + \frac{\eta_i}{2}\right)}{\eta_i}$$

$$= \frac{y_j^b - y_i^b}{\eta_i} + (\theta_b - 0.5) + (0.5 - \theta_b)\Phi_{i,j}$$

A further approximation is taken that $((y_j^b - y_i^b))\ /\ \eta_i \approx (y_j^b - y_i^b)/h_i$. Then, $\Psi_{i,j}$ is decomposed into three parts: the relative baseline shifting $(y_j^b - y_i^b)/h_i$, a constant related to the baseline-descender ratio $(\theta_b - 0.5)$, and the weighted HR $(0.5 - \theta_b)\Phi_{i,j}$. The only unknown is the relative baseline shifting $(y_j^b - y_i^b)/h_i$, which will be denoted as $\rho_{i,j}$ in the following discussion. After estimation of the first two factors, $\Phi_{i,j}$ is a linear function of r.v. HR by a constant factor, which needs to be derived based on the relation chain $RC_{i,j}$ and the RSBS parameters. The observed value of the NVCD feature $\Psi_{i,j}$ could be shifted and rescaled so that the following value conforms to the lognormal distribution of $\Phi_{i,j}$:

$$(\Psi_{i,j} - \rho_{i,j} + (0.5 - \theta_b))\ /\ ((0.5 - \theta_b))$$

Next, let us discuss the inference of $\rho_{i,j}$. Similar to the parameter estimation of HR, we also start with the pair of character $c_i$ and its parent $c_{i_p}$ in relation $r_i$. Then, the analysis is expanded to $RC_{ij}$. When $r_i \in \{SUB, SUP, HOR\}$, $y_i^b - y_{i_p}^b = \delta_{r_i} h_{i_p}$ from the RSBS model, i.e., $\rho_{i_p,i} = \rho(r_i) = \delta_{r_i}$, where the notation $\rho(\cdot)$ is valid only for pairs with a parental relationship.

Through a few algebra steps, we get $\rho(REV_{SUP}) = -1/(\gamma_{SUP}\delta_{SUP})$ and $\rho(REV_{SUB}) = 1/(\gamma_{SUB}\delta_{SUB})$. Then for $r_i$ we get

$$\Psi_{i_p,i} = \rho(r_i) + (\theta_b - 0.5) + (0.5 - \theta_b)\Phi_{i_p,i}$$

When extending to $RC_{ij}$,

$$(y_j^b - y_i^b)/h_i = \sum_{k=1}^{K_{ij}} \frac{y^b_{\mathbb{i}_{i,j,k+1}} - y^b_{\mathbb{i}_{i,j,k}}}{h_i} = \sum_{k=1}^{K_{ij}} \frac{h_{\mathbb{i}_{i,j,k}}\rho(RC_{i,j,k})}{h_i} = \sum_{k=1}^{K_{ij}} \Gamma_{ik}\,\rho(RC_{i,j,k}),$$

where

$$\Gamma_{ik} = \prod_{m=1}^{k-1} h_{\mathbb{i}_{i,j,m+1}} / h_{\mathbb{i}_{i,j,m}} = \prod_{m=1}^{k-1} \gamma_{RC_{i,j,m}}.$$

The second step and the inference of $\Gamma_{ik}$ is due to $c_{\mathbb{i}_{i,j,k}}$ is the parent of $c_{\mathbb{i}_{i,j,k+1}}$ with relation $RC_{i,j,k}$. Then,

$$\Psi_{i,j} = \sum_{k=1}^{K_{ij}} \Gamma_{ik}\,\rho(RC_{i,j,k}) + (\theta_b - 0.5) + (0.5 - \theta_b)\Phi_{i,j}$$

**III.6.3.5 Probability density function in the joint space of HR and NVCD**

The PHN has an analytical form for inference the PDF of any relation chain. As shown in Figure 45, 11 relations are enumerated for the toy example, "$a^{b^c_d}d_{f^g_h}$". Due to the space limit of the figure, H/U/D stands for HOR/SUP/SUB and R prefix indicate a reverse relation. The figure shows the distribution for one-hop relations [H, U, D, RU, RD], two-hop relations [U-U, U-D, D-U, D-D], tri-hop relation [RU-H-D] between "b" and "f", and quad-hop relation "RU-H-D-U" between "b" and "g". In the joint space of HR&NVCD, the simplest relation chain could be discriminated well. However, the "RU-H-D-U" in yellow overlaps a lot with "D-D" in blue-green showing the challenge to discriminate the stacking of super/subscript relations.

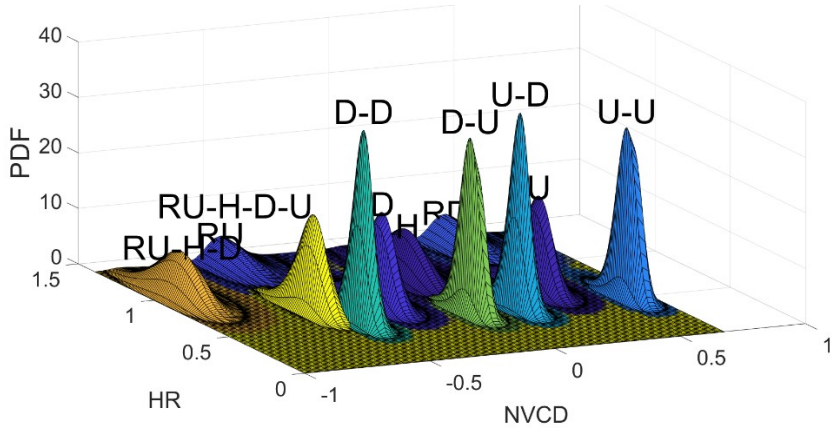



**Figure 45 The likelihood for each relation chain in HR&NVCD joint space**

## III.7 ME layout analysis experiment and results

### *III.7.1 Dataset and evaluation criteria*

InftyCDB-I [47] is a database of 20,767 mathematical expressions extracted from 476 pages of 30 English articles. The ground truth of each character in InftyCDB-I is described by id, name, bounding box, the parent id, and the relative relationship with its parent. The layout for one ME is represented as a set of triples $\{\langle i, i_p, r_i\rangle\}$, where each triple $\langle i, i_p, r_i\rangle$ indicates that $c_i$ is dominated by its parent $c_{i_p}$ by relationship $r_i$. The relative spatial relationship has 8 possibilities as illustrated in Figure 46.

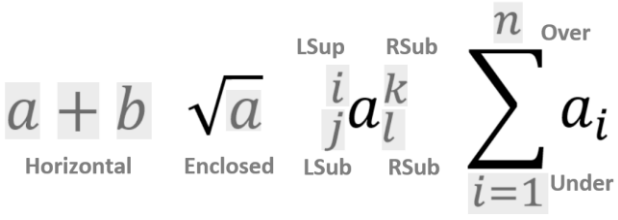



**Figure 46 The relationship among ME symbols in InftyCDB**

Both the ground truth ME layout $L_{ME}^{GD}$, and the predicted ME layout $L_{ME}^{pred}$, are represented by $L_{ME} = \{T_i\}$, where $T_i = \langle i, i_p, r_i \rangle$ denotes that the character $c_i$ is in relation $r_i$ with the parent $c_{i_p}$. The reconstructed ME layouts are used for the character height normalization, non-horizontal structure evaluation, and the global parameter estimation. Evaluation of the ME layout recognition was conducted at 1) the character-level about the parent and relative relation, 2) the whole ME, and 3) the tree structure edit distance of MathML.

- The character-level F1 score is defined as F1=2*Precision*Recall/(Precision+Recall), where Precision = |TP|/ (|TP|+|FP|), and Recall= |TP|/(|TP| +|FN|). $T_i \in L_{ME}^{pred}$ is a true positive (TP) if $T_i \in L_{ME}^{GD}$ also, otherwise a false positive (FP). $T_i$ is a false negative (FN) if it is in $L_{ME}^{GD}$ but not in $L_{ME}^{pred}$.
- For exact ME-level matching, an ML is correctly labeled if $L_{ME}^{GD} = L_{ME}^{pred}$.
- Besides the exact ME-level matching, the EMERS [110] is adopted to compare the performance of CCS-PHN against that of the two-dimensional stochastic parsing [64]. EMERS is defined over the presentational MathML [9] tree to capture the edit distance between two trees. The edit distance offers more detailed information about false predictions than other performance measures.

There is another public dataset [111] which only contains LaTeX and the automated generated PDF files. However, it is hard to evaluate the recognition performance based on the Latex or the rendered images. LaTeX is a representation language, but the same ME layout could be written in different LaTeX code. Figure 47 shows the ground truth and the predicted Latex code by our system as well as the rendered ME. Though both LaTeX codes render the same, they have a considerable edit distance of 182. On the other side, any minor error in the LaTeX code

might cause significant changes in the whole 2D layout structure. Given the reason above, only the InftyCDB-I dataset is used for the evaluation of ME layout prediction.

$$ds^2 = (1 - \frac{qcos\theta}{r})^{\frac{2}{1+\alpha^2}} \{dr^2 + r^2 d\theta^2 + r^2 sin^2\theta d\varphi^2\} - \frac{dt^2}{(1-\frac{qcos\theta}{r})^{\frac{2}{1+\alpha^2}}} .$$

(a) Rendered ME

```
ds^{2} = (1 - {qcos\theta\over r})^{2\over 1 + \alpha^{2}}
\lbrace dr^2+r^2d\theta^2+r^2sin^2\theta d\varphi^2\rbrace
-{dt^2\over  (1 - {qcos\theta\over r})^{2\over 1 + \alpha^{2}}}\, .\label{eq:sps1}
```

(b) The ground truth LaTeX value

```
{d {s}^{{2}} = {( {1 - \frac{{q {c o s} \theta}}{{r}}} )}^{{\frac{{2}}{{1 + {\alpha}^{{2}}}}}}
\{ {d {r}^{{2}} + {r}^{{2}} d {\theta}^{{2}} + {r}^{{2}} {{s i n}}^{{2}} \theta d {\phi}^{{2}}} \}
- \frac{{d {t}^{{2}}}}{{{( {1 - \frac{{q {c o s} \theta}}{{r}}} )}^{{\frac{{2}}{{1 + {\alpha}^{{2}}}}}}}} .}
```

(c) The predicted LaTeX value

**Figure 47 Example to show fallacy to evaluate using edit distance of LaTeX**

*III.7.2 ME layout prediction evaluation at the character level*

For character-level evaluation, the overall F1 score is 0.975. By taking a further look into the details as shown in Figure 48, it appears that the F1 score is ME length dependent, as expected. When the ME length is shorter than 60, the F1 score is mostly higher than 0.95. The score fluctuates significantly once the length becomes longer than 60. The sub-optimal heuristics in the handling of a large number of the non-horizontal structure might cause the fluctuating performance.

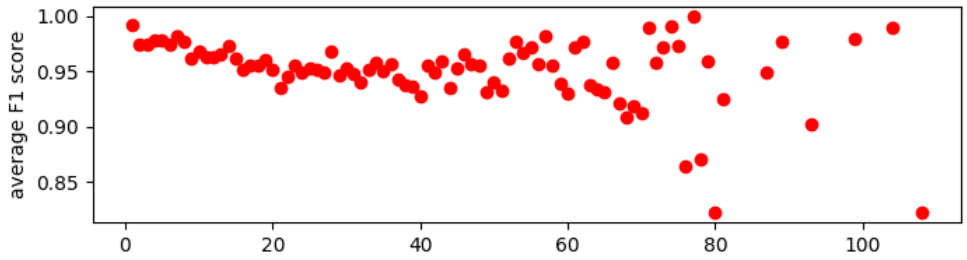



**Figure 48 The performance of F1 score vs. the number of characters in MEs**

*III.7.3 ME layout prediction evaluation by exact matching*

The ME level evaluation tests the correctness of labeling for a full ME. Some earlier work reported in [46] achieved the accuracy of 38% for the hand-written dataset UW-III. As for printed ME recognition, Okamoto [112] reported the correct ratio as 96.83% based on results tested for 3000 MEs. However, their dataset is limited only containing papers from one journal where the fonts are constrained. On the other hand, our algorithm achieved the accuracy of 89.6% on the InftyCDB-I dataset, which consists of over 20,000 MEs extracted from 13 journals.

*III.7.4 ME layout prediction evaluation by EMERS tree edit distance*

Presentational MathML (PML) is another common way to describe the layout structure of MEs in a tree structure. However, the same ME could be represented by different MathML structures. EMERS, which stands for Evaluation of Mathematical Expression Recognition System, is the first attempt to normalize the difference among MathML trees using tree edit distance as a way to measure the quality of the recognition. Though EMERS tries to normalize the ME, there are still many situations not covered. As a result, cross-comparison between different methods may not be as consistent as based on other measures. For example in Figure 49, the ME 28000767 in InftyCDB is $B_{\alpha}$ with the character $B$ as the parent of $\alpha$. But there might

be different PMLs for this simple ME. The ground truth MathML for $B_\alpha$ is shown on the left, while the MathML generated by our procedure is shown on the right. Though they are expressed the same ME from the view of the layout, their MathML representations are not exact the same. The converted MathML is with extra root 'mathml' tag and without the extra 'mrow' tag.

```
<msub>
    <mi>B</mi>
    <mrow>
        <mi>&alpha;</mi>
    </mrow>
</msub>
```

A) InftyCDB Ground Truth

```
<mathml>
    <msub>
        <mi>B</mi>
        <mi>&alpha;</mi>
    </msub>
</mathml>
```

B) MathML generated by our procedure

**Figure 49 Example of the flexibility of the presentational MathML**

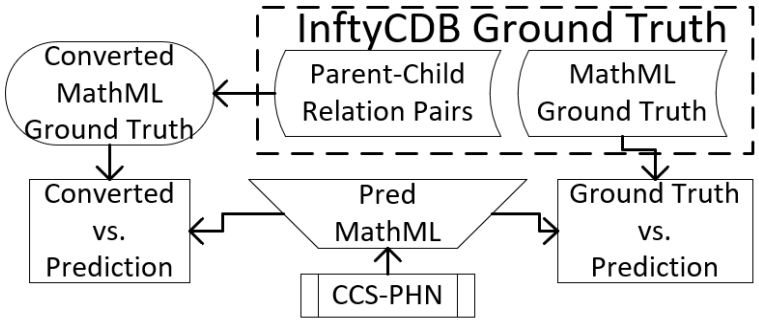



**Figure 50 ME-level evaluation using MathML representation**

**Table 11 ME-level evaluation using the EMERS edit distance on MathML**

| HC length | 2-7 | 8-14 | 15-21 | >22 | overall |
|---|---|---|---|---|---|
| Case Ratio | 69.5% | 21.0% | 7.3% | 2.1% | 100% |
| 2DPCFG [64] | 0.8±1.5 | 2.6±3.1 | 4.1±4.2 | 8.4±9.3 | 2.3±3.8 |
| GT-PD | 1.0±1.1 | 2.5±1.9 | 4.0±2.8 | 5.8±3.4 | 1.6± 1.9 |
| CT-PD | 0.14±0.6 | 0.6±1.5 | 1.1±2.1 | 1.4±2.3 | 0.3±1.1 |

Regardless, to gain some insights on PHN's performance at this level, a pipeline designed is shown in Figure 50 to evaluate the conformance of our prediction versus the ground truth as well as studying the effect of our MathML generation utility on the EMERS score. The parent-child relationship from our CCS-PHN model is converted to "Pred MathML."

The result is shown in Table 11. The first evaluation is between "Pred MathML" and the ground truth MathML provided by the InftyCDB. The result is shown in the row "GT-PD." The 'GT-PD' is using the same evaluation protocol as the Alvero's 2D PCFG method [64]. Our Method achieves smaller edit distance and smaller standard derivation, especially for the longer MEs, showing the advantage of the global modeling. We further create the "Converted MathML" ground truth directly from the parent-child relationship ground truth to reduce the discrepancy between our MathML generation procedure with that of the InftyCDB. The evaluation between the Prediction "Pred MathML" and the "Converted MathML" ground truth is show in the row "CT-PD" of the performance table. In comparison with "GT-PD," the average edit distance is further reduced by more than 1, which is due to that our procedure consistently have one root node of 'mathml' while the ground truth does not have it.

### *III.7.5 Post Checking*

The results of the rule-based processing show that the vertical structure identification performance still needs improvement. The error in the rule-based stage will propagate to the next stage, causing cascading errors. It would be helpful to have a self-checking mechanism so that human intervention could step in when necessary. There is one way for post-checking to identify whether the predicted ME layout is correct. Based on the analysis in the per-merging of consecutive alphabets, the center line analysis is a very reliable feature to decide whether two

characters satisfying the condition HorByCenter are on the same baseline. In Figure 30, almost all pairs of alphabets on the same baseline are covered, when the threshold $\alpha > 0.3$. This is in accordance with the statistics on the NVCD features. For pairs of alphabetic characters on the same baseline, the mean value and std value of NVCD feature are 0.0004 and 0.09, respectively. The $3\sigma$ is rougly corresponding to 0.3 as elaborated above. This means for a pair of characters predicted to be on the same baseline, but their NVCD feature is larger than $3\sigma$, it is identified as a wrong prediction. After the checking, there are about 730 MEs identified that could not pass the test. And after the filtering, the F1 score is increase from 0.971 to 0.983.

### *III.7.6 Execution Speed*

Being a parametric model with an analytical form of the density function, PHN is much faster than its non-parametric counterpart. Figure 51 shows the average execution time in log scale concerning the number of characters in an ME. The splitting threshold was set to be 10, which clamps down the growth of the computing cost. On the other hand, the running time for the non-parametric method (shown in the orange curve) is about 12 seconds for an ME with eight characters. The F1 score loss by switching from the non-parametric model to the PHN model is negligible.

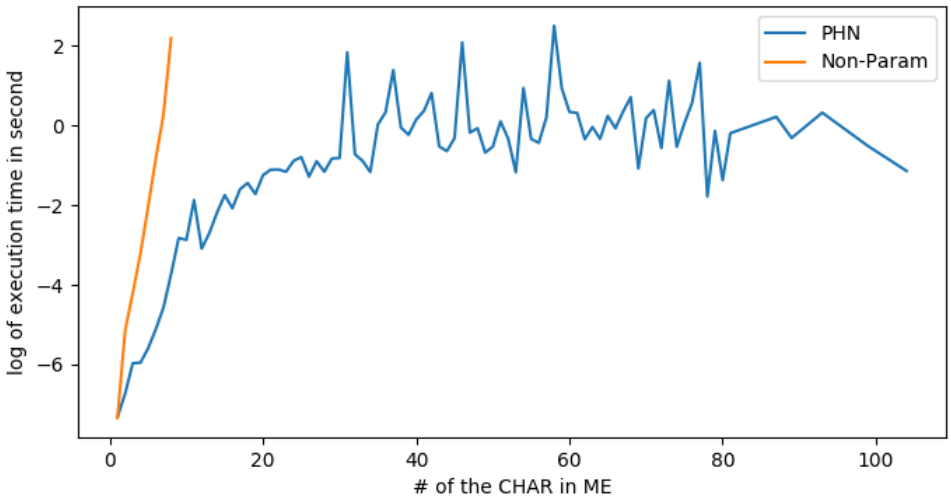



**Figure 51 Speed performance comparison**

### III.8 Conclusion of CCS-PHN model

This chapter presents a systematical analysis of the ME layout analysis covering the typography design, common writing practice, and a global inference model with parametric approximation PHN. The refined analysis of the typography design improved the discriminating ability of the features and acting as the basis for pre-mering and post checking. PHN can efficiently estimate the probability distribution of the HR and NVCD features for pairs of characters satisfying any chain of relations, thus enable global inference. The proposed content-constrained spatial model outperforms the state-of-art system under multiple evaluation criteria with low computational cost.

# CHAPTER IV

# ME SEMANTICS ANALYSIS

## IV.1 Overview of the chapter

The ME layout analysis only specifies the attachment and relative position among characters. However, the hierarchical grouping/segmentation and the meaning of the hierarchical structure is not understood, which is the scope of the semantic analysis. The recovery of ME semantics is vital because this is the level that humans perceive the MEs. Experiments have shown that the semantic level ME could improve the performance of mathematical information retrieval [42]. Besides, the symbolic computing, auto proving [3], code generation all require the ME to be at the semantic level. However, there are three levels of challenges:

- One character could have multiple meanings due to the mathematical dialect. The character "|" could mean absolute value or carnality of a set. "$\Delta$" could be used as the difference operator or a variable. "d" could be a variable or the prefix of the integral target variable.
- There exists ambiguity in the grouping of characters for the correct order of execution. For example, $\sum_i x_i + \sum_j y_j$ could be interpreted as $(\sum_i x_i) + (\sum_j y_j)$ or $\sum_i( x_i + \sum_j y_j)$.
- The same spatial relationship could have different meanings:
  - The superscript could mean indexing, notation, exponent, derivation, or inverse function.
  - Consecutive alphabets could mean multiple-character variable, notation, or multiplication without an operator.
  - There might be multiple ways to group the elements.

In this work, the semantic taxonomy of ME is presented, which provides a standard guideline for the parsing. Second, the probabilistic context-free grammar framework is adopted to resolve the scoping ambiguity of the hierarchical structure. Further, two of the challenges mentioned above will be attacked, i.e., consecutive alphabets disambiguation and superscript semantics. For consecutive alphabets, the normalized pointwise mutual information (NPMI) is used to identify multi-character identifiers based on the frequency of their occurrence. Heuristic rules are proposed to resolve the superscript semantics ambiguity. By the end, the ME semantic parsing system is evaluated at multiple datasets.

### IV.2 ME semantic taxonomy

In the ME layout section, some introduced MEBlocks already have semantic meanings. In this section, we go beyond the ME layout and present the semantic taxonomy of ME. ME semantics are at the human-level concepts, such as triangle functions, exponent, integral, etc. Using the content MathML [9] and OpenMath [55] as the references, a semantic taxonomy of ME is summarized. A portion of the ME taxonomy for the atomic expression is illustrated in Figure 52. For every ME, they have the interface as specified in the abstract class ‘Expression’ to access the equivalence, containing relationship, retrieval all the children, and convert to Content MathML, Latex, or operator tree. The ME objects are organized into atomic ME Expression and compounded ME Expression. The simple ME could be an identifier, modified identifier, constant number, each including different members. The common compounded ME includes relation expression, function application expression, and the bind var expression. Besides the atomic expression package, and the compounded expression package, in the appendix, there are other domain-specific packages for domains such as calculus, set theory, functional analysis, logic, number theory, probability.

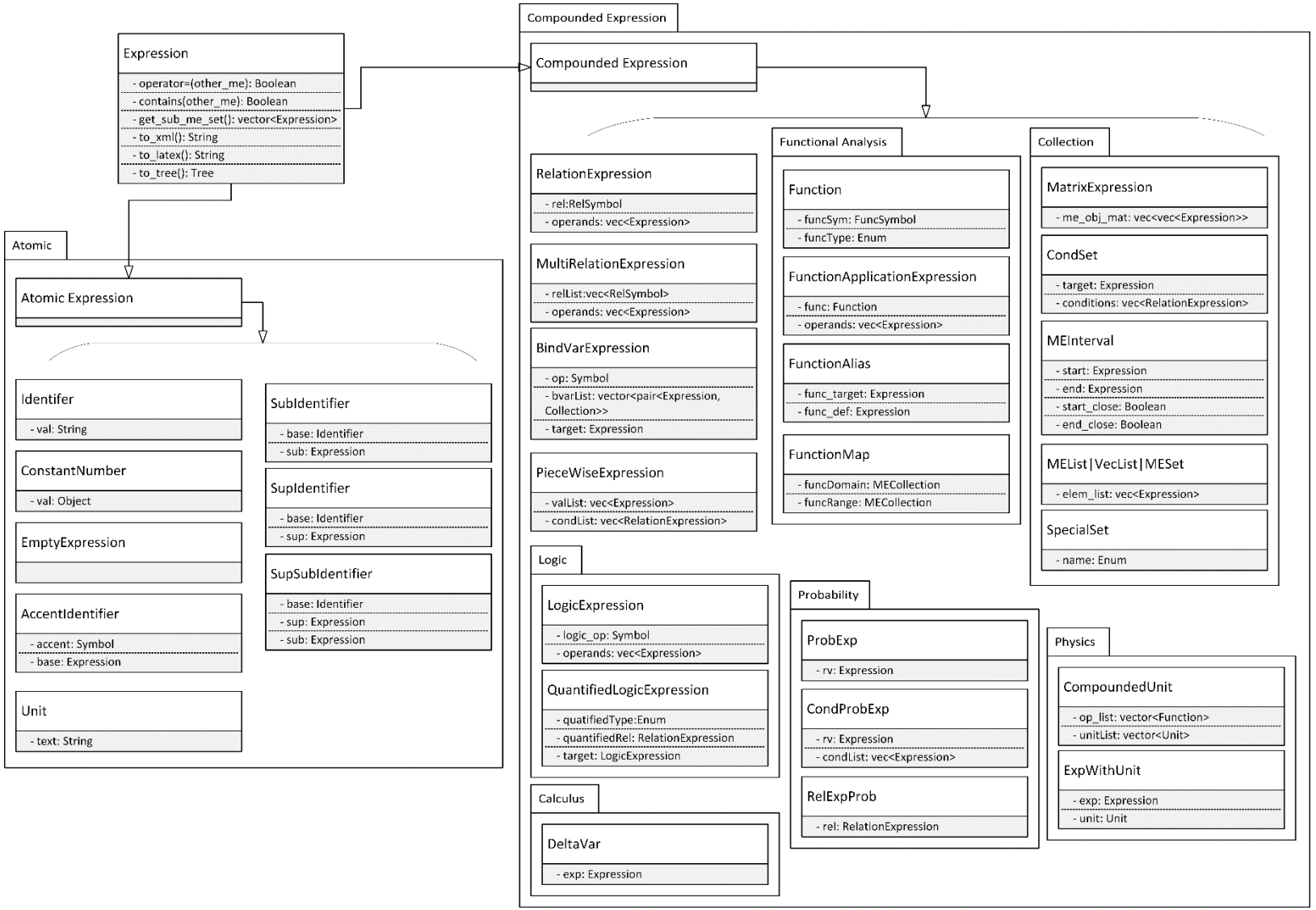



**Figure 52 The ME semantics taxonomy**

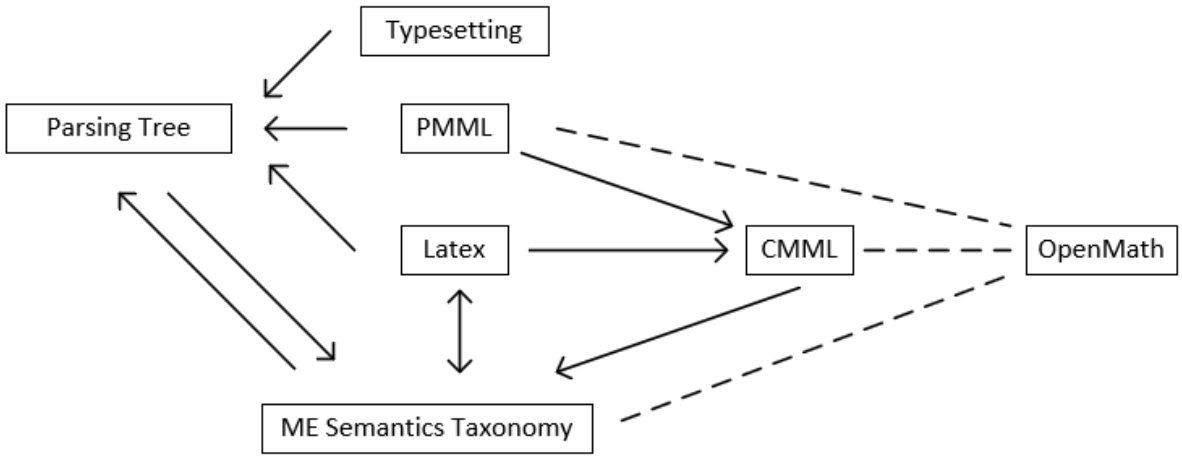



**Figure 53 Conversion among different standard**

Our ME semantic taxonomy can cover all the concepts mentioned in the official content dictionary of OpenMath Standard [55] as of Aug. 2018. One advantage of our ME semantic taxonomy data structure is that common operations could be defined as the function shown in the Expression class. The conversion between different standards is illustrated in Figure 53.

- The conversion from typesetting, Latex, and Presentational MathML (PMML) to parsing tree is the target of this chapter.
- There is a bi-direction conversion between the parsing tree and the target ME semantics taxonomy.
- The conversion from the parsing tree to the ME object in the semantic taxonomy is the parsing process while the creation of the parsing tree from the ME object is for the training of PCFG parser.

- The conversion from PMML to Content MathML (CMML) is realized by the LateXML tool. An eXtensible Stylesheet Language Transfomer (XSLT) is adapted to convert PMML to Latex during the evaluation.
- The OpenMath is connected to some other standard as a cross-checking.

### IV.3 The parsing algorithm and ambiguities resolution

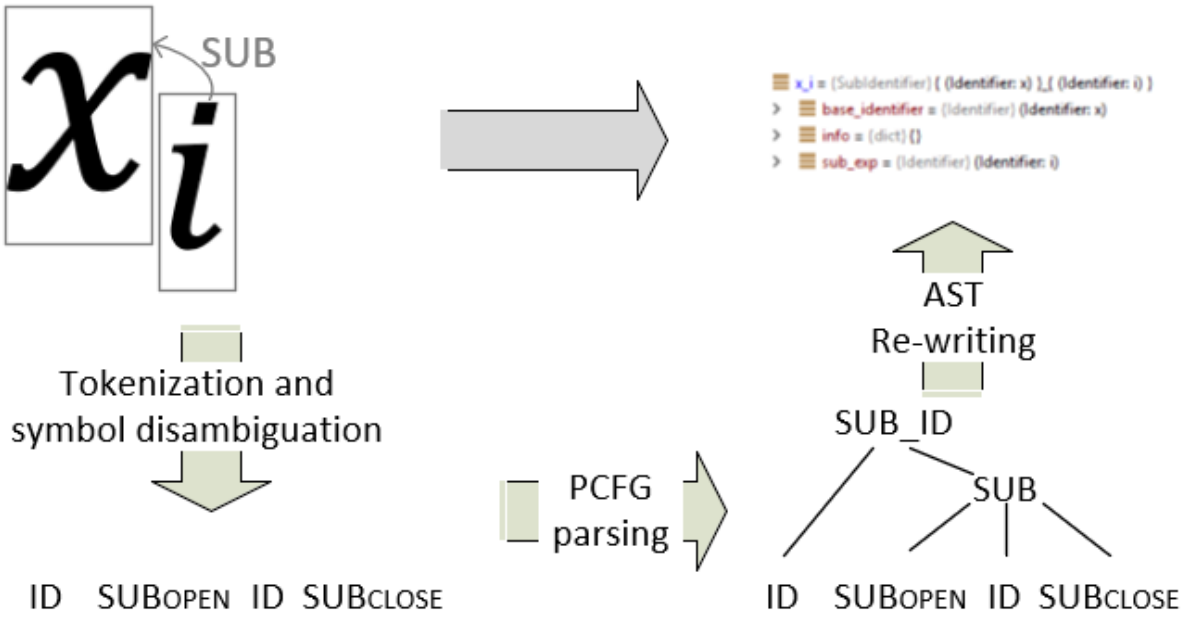



**Figure 54 ME semantics parsing pipeline**

The framework to parse the ME semantics is shown in Figure 54. In this work, the Probabilistic Context-Free Grammar framework is adopted. For the situation with ambiguity, two mechanisms are proposed to resolve them. First, for the multi-character element, the NPMI is used measurement to merge them first. Second, the operator hierarchy is resolved during the PCFG parsing process. At last, the layout structure with multiple meanings is resolved during the

abstract syntax tree transformation process. Context-sensitive rules will be triggered based on the type of element generated.

### *IV.3.1 Consecutive alphabets ambiguity resolution*

The consecutive characters could be multi-character variable/notation/function or multiplication omitting the operators. If it is a structure omitting the multiplication operators, the subcomponents will also occur in combination with other symbols. A co-occurrence measurement is likely to differentiate between the two situations. In this work, the normalized pointwise mutual information (NPMI) [113] is used:

$$NPMI(x,y) = \frac{PMI(x,y)}{h(x,y)}, PMI(x,y) = \log\frac{p(x,y)}{p(x)p(y)}$$

, where $h(x,y) = -\log p(x,y)$. The NPMI has a value of -1 if $x$ and $y$ never co-occur together, 0 for independent, and 1 for complete co-occurrence. In this work, the $NPMI^*$ is calculated as the maximum NPMI score by separating the string $s$ of size $n$ into two parts.

$$NPMI^* = max_{i\in[1,n-1]}NPMI(s[1:i], s[i+1,n])$$

By applying it to one of the test files, the multi-character elements, as well as their substring, are found with high $NPMI^*$ score for the aggieSTEM test case in Table 12. There are only three noise situations for “mgh,” “rnet,” and “MR,” when the $NPMI^*$ is larger than 0.3. But their frequency is low. In this work, after manually setting the $NPMI^*$ threshold as 0.3 and the frequency threshold as 3. Most of the usage for multiple-character identifier are recovered, including “net,” “KE,” “PE,” and “rot.”

**Table 12 NPMI score & frequency for the AggieSTEM test case**

| Token | $NPMI^*$ | Frequency | Token | $NPMI^*$ | Frequency |
|---|---|---|---|---|---|
| mgh | 1.000 | 2 | **tran** | 1.000 | 2 |
| **grav** | 1.000 | 2 | **trans** | 1.000 | 2 |
| **ne** | 0.975 | 29 | **net** | 0.975 | 29 |
| **KE** | 0.915 | 21 | **rans** | 0.869 | 2 |
| gh | 0.869 | 2 | **ans** | 0.763 | 2 |
| **gra** | 0.738 | 2 | **PE** | 0.615 | 5 |
| **et** | 0.585 | 29 | MR | 0.556 | 2 |
| rne | 0.495 | 2 | rnet | 0.495 | 2 |
| **OE** | 0.449 | 1 | **rot** | 0.427 | 8 |
| **ro** | 0.427 | 8 | **tra** | 0.425 | 2 |
| **ot** | 0.392 | 8 | **rav** | 0.377 | 2 |
| mg | 0.377 | 2 | **ran** | 0.351 | 2 |
| **ns** | 0.246 | 2 | mr | 0.245 | 12 |
| **gr** | 0.184 | 2 | ML | 0.167 | 2 |
| ma | 0.128 | 5 | mv | 0.017 | 2 |
| **av** | -0.067 | 2 | **an** | -0.093 | 2 |
| rF | -0.130 | 3 | **ra** | -0.148 | 4 |
| rn | -0.202 | 2 | at | -0.286 | 2 |
| Fr | -0.289 | 1 | **tr** | -0.395 | 2 |

### *IV.3.2 Probabilistic Context Free Grammar*

After the layout of an ME is identified, the next task is to create a hierarchical grouping of the symbols or multi-character tokens corresponding to semantic ME object. The way human creates sentences or MEs could be described as grammars, which consist of terminal tokens, internal states, the target states, and production rules. Though the context is essential, it will lead to computational infeasibility. When the context-free assumption is introduced, a dynamic programming CYK algorithm [114] is available to generate the syntactic structure in polynomial

time. Given the probability for each production rules, each possible syntactic structure for a sentence/ME would be associated with a probability to rank the likelihood. In this section, the PCFG for ME parsing is introduced in a bottom-up fashion.

### IV.3.2.1 Terminal tokens

The characters, multi-character tokens, and the spatial relationships are mapped to terminal tokens in the grammar as shown in Table 13. The alphanumeric and Greek characters are mapped to a corresponding token. They are not mapped to identifier directly as some of them might be physical unit. They could also be merged as a multi-character function or variable. The digits are not pre-merged because there are production rules for integer/float number. The 'Operation', 'Big Operator', and 'Named function' group are operators. The relation symbols are group into general relation, number relation and distributed into the token set for each domain. The punctuation and fence symbols also carry important meanings. The spatial token corresponds to the relative spatial relationship among blocks. Special notations for the set theory, logic and calculus are also covered. A complete list of terminal tokens and the explanation could be found in the appendix.

**Table 13 Terminal tokens of the PCFG for ME semantic parsing**

| Group | description | Group | Description |
|---|---|---|---|
| Symbol Groups | Greeks, Alphanumeric | Fence | (), [], {}, ⌈ ⌉, etc. |
| Operations | +, -, ×, ⊕, etc. | Spatial | SUB_OPEN, SUB_CLOSE, etc. |
| Big Operators | ∑, ∏, ∐, etc. | Set | ∈, ∪, etc. |
| General relations | =, ≡, ≠, etc. | Logic | ∀, ∧, etc. |
| Numerical relation | <, ≤ | Calculus | ∂, ∫ |
| Punctuation | ·, ∗, ′, !, :, etc. | Misc. | Other assistant tokens |

#### IV.3.2.2 Internal states

The internal states are grouped as shown in Table 14 and Table 15. First, the characters are mapped to the general categories such as DIGIT, GREEK, and ALPHABET. They could also be merged as NUMBERs, UNITS, or NAMED_FUNC. For the operator and relations, they are grouped based on the priority of execution based on the production rules. Each spatial structure has corresponding elements. Fence also play important roles in the ME scoping and semantics. At last, these basic units are built into a hierarchical structure, ranging from FACTOR, TERM, EXP, REL_EXP, and ME. The state token ME is the target root state of the PCFG representing a mathematical expression.

#### IV.3.2.3 Production rule and probability

The production rules for different types of ME are manually constructed. Only parts of the rules and the associated probability are shown in Table 16. An expression (EXP) is an ME object with numerical values. It could be a TERM which denotes the result of multiplication, or the plus/minus of multiple TERMs. As shown in the probability, a large portion (>97%) of the EXP is composed of merely one TERM. Addition fence could be applied to enforce the evaluation order. But they are rare in our training data gathered from Arxiv. The EXP could be chained into a list of expression, i.e., EXP_LIST, which is used in vector, set, etc. A complete grammar of the production rules could be found in the appendix.

**Table 14 Internal states of the PCFG for ME semantic parsing**

| | | |
|---|---|---|
| Symbols | SCI_NUM_FACTOR | Scientific format float |
| | GREEK | Greek characters |
| | DIGIT | Digits |
| | ALPHABET | Alphabets |
| | ALPHABET_SEQ | Alphabetic string |
| | NUM_FACTOR | Number |
| | INT_NUM_FACTOR | Integer |
| | FLOAT_NUM_FACTOR | Float |
| Op/Rel | ARITHM_OP_LEVEL1 | Null |
| | ARITHM_OP_LEVEL2 | TODO: multiply and division |
| | ARITHM_OP_LEVEL3 | TODO: plus |
| | SET_OP_LEVEL1 | set operation |
| | DEF | := |
| | SINGLE_OP | operator expect one argument |
| | QUANTIFIER | logic quantifier |
| | REL_OP | relation |
| | REL_OP_SET_LEVEL1 | relation of set |
| | REL_OP_ARITHM_LEVEL1 | relation of numbers |
| Function | NAMED_FUNC | common function such as lim, min |
| | USER_FUNC | user-defined function composed of alphabets |
| | BIG_OP | big operator such as sum, prod |
| | FUNC | a function |
| | FUNC_DELC | declaration of the function |
| | DENOTATION | denotation of one symbol as others |
| Physics | CUNIT | compounded physical unit |
| | UNIT | physical unit |
| Spatial | SUB_SUP_FACTOR | factor with both sub/superscript |
| | SUB_FACTOR | factor with subscript |
| | SUP_FACTOR | factor with superscript |
| | OVER_EXP | over parts |
| | SUB_EXP | subscript |
| | SUP_EXP | superscript |
| | UNDER_EXP | under parts |

**Table 15 Internal states of the PCFG for ME semantic parsing (Cont.)**

| ME Object | VARSYM | a variable denotation by a character |
|---|---|---|
| | REL_OP_EXP | to form relation |
| | EXT_REL_EXP_LIST | to form a list of relation expression |
| | PUNCT_COMMA_REL_EXP | to form a list of relation expression |
| | EXT_TERM | to form exp |
| | EXT_FACTOR | to form term |
| | MUL_OP_FACTOR | to form term |
| | BIG_OP_SUB_EXP_1 | big operator with subscript |
| | VAR | a variable |
| | EXP_LIST | to form expression list |
| | EXT_EXP_LIST | to form expression list |
| | PUNCT_COMMA_EXP | to form expression list |
| | ME | the root node |
| | EXP | such as a+b |
| | REL_EXP | relation expression |
| | REL_EXP_LIST | a list of relation expression |
| | TERM | such as a*b |
| | FACTOR | factor |
| | NORM_FACTOR | Norm Fence |
| | SET_FACTOR | Set |
| | ETC_FACTOR | etc ... in set |
| | BIG_OP_FACTOR | big operator factor |
| Fence | VEC_FENCE_OPEN \| CLOSE | Open, close fence for vector |
| | FENCE_ARG_OPEN \| CLOSE | open (, close ) for the function arguments |
| | FENCE_OPEN_RANGE_OPEN | begin of open interval |
| | FENCE_OPEN_RANGE_CLOSE | end of open interval |
| | FENCE_CLOSE_RANGE_OPEN | begin of closed interval |
| | FENCE_CLOSE_RANGE_CLOSE | end of close interval |
| | FENCE_MATRIX_OPEN \| CLOSE | begin of a matrix structure |
| | FENCE_GROUP_OPEN \|CLOSE | open (, close ) to enforce execution order |
| | FENCE_SET_OPEN\|CLOSE | begin of set mark \{ , \} |
| | FENCE_CASE_OPEN | indicator of piecewise ME |
| | FENCE_ABS_OR_CARD_OPEN | vertical bar as abs or carnality |
| | FENCE_ABS_OR_CARD_CLOSE | vertical bar as abs or carnality |
| | FENCE_NORM_OPEN \|CLOSE | double vertical bar as norm |
| | PR_OPEN_FENCE | open ( for the probability |
| | PR_CLOSE_FENCE | close ) for the probability |
| | COND_SET_FENCE | vertical bar in the conditional probability |

**Table 16 Production rules and the probability**

| Rule | Probability |
|---|---|
| EXP -> TERM | 0.974 |
| EXP -> TERM, EXT_TERM | 0.0218 |
| EXP -> BRACKET_OPEN, EXP, BRACKET_CLOSE | 0.0003 |
| EXP -> SQ_BRACKET_OPEN, EXP, SQ_BRACKET_CLOSE | 0.0003 |
| EXP -> GROUP_OPEN, EXP, GROUP_CLOSE | 0.0003 |
| EXP -> TERM, UNIT | 0.0003 |
| EXP_LIST -> EXP, EXT_EXP_LIST | 0.909 |
| EXT_EXP_LIST -> PUNCT_COMMA_EXP | 0.612 |
| EXT_EXP_LIST -> PUNCT_COMMA_EXP, EXT_EXP_LIST | 0.388 |
| PUNCT_COMMA_EXP -> PUNCT_COMMA, EXP | 1.0 |

### IV.3.2.4 Training and inference

As the production rules, terminals, and states are manually enumerated, the training phase for the PCFG is only to estimate the probability of each rule. Given the property that the sum of probability of all productions rules with the same left-hand side (LHS) equals 1, the probability of a rule $r$ is estimated as:

$$P(r) = \frac{\widehat{f}\,(r)}{\sum_{r'.lhs=r.lhs} \widehat{f}\,(r')}$$

The notation $\widehat{f}\,(r)$ indicates the normalized frequency of the production rule $r$. Let $f(r)$ denote the frequency of the production rule $r$. Then,

$$\widehat{f}\,(r) = \begin{cases} f(r) & iff(r) > 0 \\ 0.1\dfrac{min\{f(r'): f(r') > 0, r'.lhs = r.lhs\}}{|\{r': f(r') = 0, r'.lhs = r.lhs\}|} & otherwise \end{cases}$$

The second rule means the 0-frequency rules are normalized based on the minimal non-zero frequency rule with the same *lhs*. The 'frequency' is first reduced by a 0.1 factor and then further

reduced by the number of the zero-frequency rule with the same *lhs*. In this way, the normalization will not give too much weight for the zero-frequency rules.

Parsing tree data at the ME semantic level is precious. However, there is not such a data available. First, the MEs are mostly represented in Latex, Presentation MathML, Content MathML. A conversion is needed from these formats into a parsing tree representation. Second, different parsing system might use different rules. The converter must be customized based on target parsing system. This is also common for Natural language processing, where the Brown PoS tagging dataset use the coarse level, while the Penn Tree dataset uses fine level tagging [101]. In this work, 43245 MEs from 100 Arxiv papers are processed using LateXML [32]. The output XMath internal format is converted into our semantic taxonomy. Then, based on a customized converter, the ME objects are converted into all possible parsing trees by our manually constructed rules. Production rules are collected from each parsing tree for the raw frequency statistics.

For the inference, a dynamic programming approach in NLTK is adopted. Given a sequence of terminal tokens, it will return a list of parsing trees together and the probability in descending order w.r.t the probability.

### *IV.3.3 ME objects generation*

Given the abstract syntax tree (AST) is built from the PCFG parsing system, a reclusive procedure to construct the ME Object. Most of the rules simply return one of the elements corresponding to the right states. For each terminal token, a ‘Symbol’ object is created. Most rules are designed to have unique interpretation. For example, the rule ‘UNIT-> M_LOW_TEXT’ will create a unit object uniquely. But there are few cases that ambiguity might be introduced. Special attentions are paid to the superscript function classification as exponent,

index, or function operator. The following heuristics are proposed to resolve the actual intent of a superscript based on the context, neighbor and object type:

- function operator: if the superscript is a constant number with value -1
- exponential function: if the super component is a ConstantNumber, if the base expression is a complex expression such as the Fence Expression or FunctionApplicationExpression
- a super identifier: if the base expression is a Symbol, Identifier, SubIdentifier

## IV.4 Experiment and Result Analysis

The dataset/ data sources used for evaluated ME semantics understanding are first introduced. Then the evaluation criteria and the performance of recognition are presented.

### *IV.4.1 Dataset collection*

For our data collection purposed, similar with [115], [116], an extension to the LaTeX is proposed so that the semantic and the scoping are enforced without any ambiguity. The special tags for the STeX is illustrated in Table 17. The annotation interface is illustrated in Figure 55. The image for the ME and the current semantic representation in the indented string format is shown in the first two row. A user could input the correction in STeX format and then click check to show the parsed the semantic representation in the last text area.

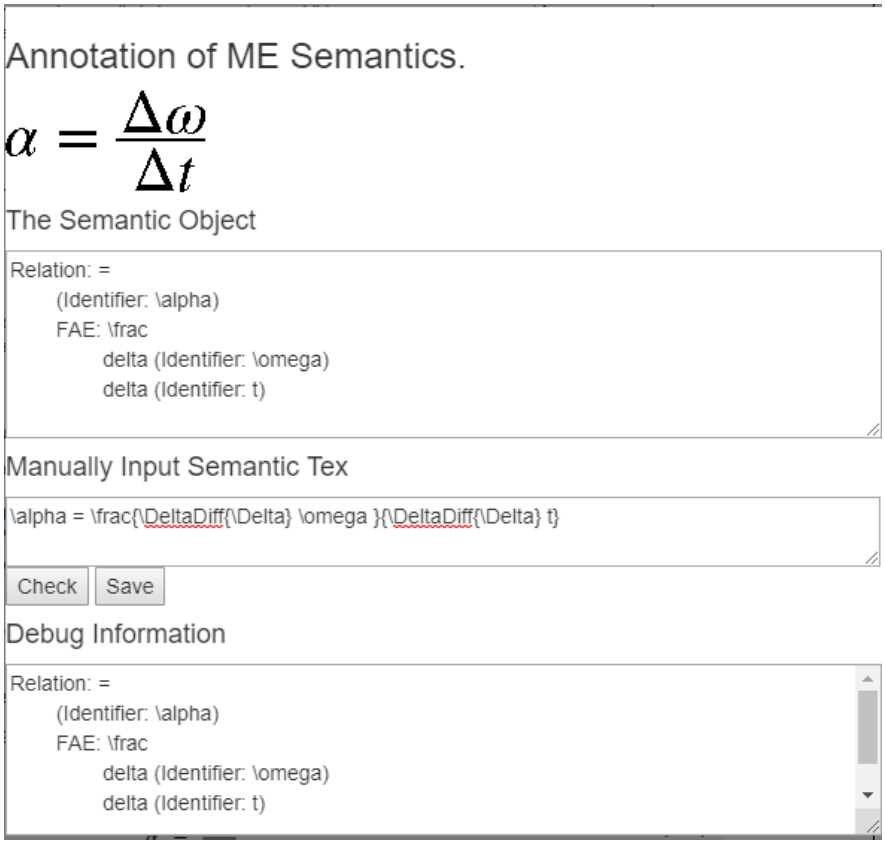



**Figure 55 The interface for the annotation of the ME semantics in STeX**

**Table 17 Extended LaTeX tags to annotate the ME semantics**

| | |
|---|---|
| Alphanumeric | Var, Func, Const, Unit, Token,<br>ConsecutiveMulitiply, ConsecuiveNotation,<br>DeltaDiff, DeltaVar,<br>MinusSym, NegSym |
| Superscript | SupIndex, SupNotation, SupExp, SupTranspose, SupFuncInv, SupFuncDiff |
| Fence | Open/CloseParArg, Open/CloseParGroup,<br>Open/CloseSqVec, Open/CloseSqIndex,<br>BarAbs, BarNorm |

*IV.4.2 Evaluation criteria and ME semantics recognition performance*

**Table 18 The performance for ME semantic analysis**

| Method | Total Number | Exact Match # | Operator Tree Edit Distance |
|---|---|---|---|
| | | AggieSTEM | |
| LateXML | 283 | 169 | $3.76 \pm 5.85$ |
| MECA-PCFG | | 175 | $2.597 \pm 4$ |
| MECA-PCFG+MER | | 185 | $2.13 \pm 3.62$ |
| | | Geo Simulation | |
| LateXML | 38 | 4 | $19.39 \pm 22.13$ |
| MECA-PCFG | | 8 | $7.08 \pm 9.39$ |
| MECA-PCFG+MER | | 8 | $7.08 \pm 9.39$ |

The evaluation will be at the exact matching level and edit distance on the operator tree level based on the tree edit distance [117]. The performance for ME semantics analysis for the AggieSTEM physics chapter and a Geographical simulation model is shown in Table 18.

There are 283 MEs collected for the AggieSTEM chapter. For the simple MEs such as identifier or identifier with super/subscript or accent, both LateXML and MECA-PCFG parser handles them well. But the MECA-PCFG is a bit better in the exact matching and with significantly smaller operator tree edit distance. The smaller operator tree edit-distance shows that the MECA-PCFG is better at capturing partial structure. But for the identifier with multiple characters, both systems could not correctly merge them. With pre-merging, ten more MEs are correctly identified, and the average operator tree edit distance is reduced by 0.4.

As for the other geographical simulation modeling paper, only the large IMEs are collected to test the performance of the system on complex MEs. The average number of characters within an ME in the LaTeX format is 89, with a standard derivation of 55. Six MEs that LateXML failed to process are removed to avoid bias in the comparison. For these large MEs, neither system is good at fully recover the ME semantics. But our MECA-PCFG is a bit better. The pre-merger does not help a lot because the consecutive notations or variables are rare in the study sample. From the view of the operator tree edit distance, our MECA-PCFG gives much smaller edit distance in comparison with the LateXML.

## IV.5 Conclusion for ME-semantics parsing

In this chapter, a three-phase PCGF based parser is proposed for ME semantics understanding to systematically resolve the ambiguities at three levels: symbol tokenization, hierarchical structure recovery, and AST interpretation. With pre-merging according to the NPMI score, multiple-character identifiers are per-merged. A heuristic rule-based superscript interpreter is adopted to create different ME Object accordingly. On a preliminary dataset with semantics annotation, the experiment shows that our ME semantics outperform the state-of-art system LateXML. The performance improvement is significant for large MEs.

Note that we only provide a framework for the ME semantics parsing. More modules could be easily added as extensions to make the parser more powerful. For example, in the tokenization phase, classifiers could be built to decide: whether the vertical bar as absolute value or carnality; whether $\Delta$ as a variable or a function; whether d/D mean differentiation or a variable. At the last stage of ME object construction, the type of each Identifier or Notation could be inferenced according to the context and some global statistics. Further, the OpenMath

provides paired the representational MathML and the Content MathML. It is worthwhile to evaluate the ME semantics parsing performance on it.

CHAPTER V

DECLARATION EXTRACTION AND MIXED WORD-ME PROCESSING

## V.1 Overview of the chapter

The mathematical expressions (ME) alone are not enough for readers to understand the complex system, as the mathematical notations need to be mapped back to the abstract/physical concepts through declarations. It is a typical writing practice that a notation must be introduced or declared before being used. This writing practice makes it possible for the automated extraction of the declaration. Besides acting as the notation table to help the reader navigate between MEs and concepts, the automatically extracted declarations are also very helpful for cross-paper analysis. It has been shown that the declaration words/phrase could help enhance the semantics of MEs for better mathematical information retrieval [8].

For automated declaration extraction, existing systems follow a two-phase framework [67], [81], [68]. First, the noun phrases (NP) are extracted as the candidates of the declaration based on traditional constituent parsing. Then, a prediction is made for each pair of ME-NP about whether the NP is the declaration of the ME using a binary classifier. The classifier is trained using the features concerning the common declaration patterns, the words/part-of-speech (PoS) of neighbor tokens, and structure features.

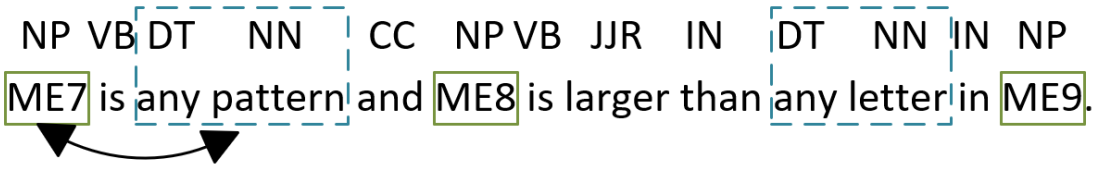



**Figure 56 Three-phase framework**

However, a degradation of the part-of-speech (PoS) tagging and constituent parsing was observed when applying the traditional NLP toolkit to the mixed word-ME (MWM) sentences. ME could be very complicated, corresponding to sentences or subordinate clauses. The sentence role of ME does not exist in the traditional PoS annotation schema. It leads to cascading error for later constituent parsing. We overcome the limitation with a new three-phase framework in Figure 56. First, a customized PoS tagger is trained for the (MWM) sentences using the tri-gram HMM model. Then, the NPs (marked in dashed rectangles) are extracted as the declaration candidates by a shallow parser, i.e., noun phrase chunker. At last, a decision procedure decides whether an NP candidate is the declaration for an ME in the same sentence. Experiment results show that our MWM PoS tagger could improve the tagging quality. As a consequence, the declaration extraction is significantly improved for both pattern-matching and SVM-based classification methods.

The above example shows that the ways people declare MEs seem to follow limited patterns. However, it is a non-trivial to enumerate all the patterns manually. A semi-automated weakly-supervised (SAWS) approach is proposed to mine the patterns from a large quantality of unlabeled data. The SAWS approach is based on the observation that 58% of the first-time occurrence of simple mathematical expression is with declaration [62]. Using TFIDF to rank and some heuristic to filter the patterns from a collection of 14K Arxiv papers, many meaningful patterns are identified.

In this chapter, the MWM processing for PoS tagging and NP extraction will be introduced first. Then, the declaration extraction methods and SAWS pattern mining is presented. Experiments and result analysis are given by the end.

## V.2 Mixed word-ME processing

The goal of mixed word-ME (MWM) processing is to accurately extract noun phrases (NP) as the candidates for the declaration. Part-of-speech (PoS) tagging is the most important low-level task, which is the foundation of high-level tasks such as NP extraction and information extraction. However, the MWM sentences introduce new usage patterns compared with the everyday language, leading to the degradation of the PoS tagging, noun phrase extraction, and syntactic structure parsing. A customized PoS tagger for MWM sentences is proposed to attck the challenge. After the PoS are accurately identified, the NPs are identified using the Linear SVM-based consecutive NP chunker [118], [101].

### *V.2.1 New ME-PoS tag and In-sufficiency of existing NLP toolkit*

The mathematical notation system itself could be treated as a language. This fact implies that one ME could be very complex and even correspond to a sentence or subordinate clause of everyday language. Follow the convention in the Elsevier dataset [66]; there are three syntactic roles for ME as shown in Table 19.

**Table 19 PoS for ME and examples**

| ME-PoS | Example |
|---|---|
| NP | Let $G = limG_n$ be the projective limit of this system. |
| NML | This happens $lgn$ times by repeating squaring. |
| S | Note that $[f]p = [f_0]p$ and $[f']p = [f_0{}' - f_1]p$. |

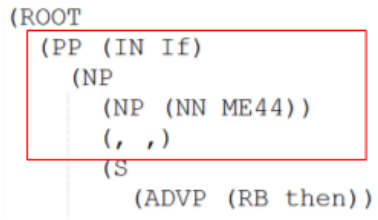



**Figure 57 The error propagation from PoS tagging to constituent parsing**

An ME could be very complex corresponding to sentences. Failure to identify such roles also leads to degradation for the PoS tagging of other words from the F1 score of 0.96 to 0.93. Further, the error will propagate to constituent parsing phrase, affecting the NP candidate generation for declaration [67]. For example, the PoS error for 'ME44' messes up the structure analysis of the sentence in Figure 57.

*V.2.2 MWM PoS tagger*

The task of PoS tagging is to predict the PoS label $l_i$ for each token $w_i$ in a sentence $s = \{w_1, \dots, w_n\}$. When the token is a plaintext word, the label candidates are the PennTreeBank PoS tags, such as NN, JJ. When the token is an ME, there are three possible labels: $\{S, NP, NML\}$. Based on the Tri-gram Hidden Markov Model based PoS tagging framework [71], the PoS tagging is formulated as the optimization goal $l^* = argmax_l P(l|s)$, where

$$P(l|s) = \prod_{i \in [1,n]} P\left(l_i | l_{i-2}, l_{i-1}\right) \prod_{i \in [1,n]} P\left(w_i | l_i\right)$$

. Two additional labels $l_{-1} = l_0 = *$ are added to the front of each sentence. The conditional probability $p(l_i|l_{i-2}, l_{i-1})$ is smoothed as

$$p(l_i|l_{i-2}, l_{i-1}) = \lambda_1 \hat{p}\left(l_i\right) + \lambda_2 \hat{p}\left(l_i|l_{i-1}\right) + \lambda_3 \hat{p}\left(l_i|l_{i-2}, l_{i-1}\right)$$

The $\hat{p}$ indicate the unsmoothed probability and $\{\lambda_*\}$ are estimated using a global context-independent smoothing [71]. For the rare words with a frequency less than 5, their suffixes are used to estimate the probability $p(w_i|l_i)$. Viterbi algorithm is used for the efficient prediction of tagging based on the tokens.

The MWM PoS tagging is evaluated on the Elsevier open access dataset [66]. It consists of 10 papers from different domains. There are 346 MWM sentences are containing 545 MEs. A 10-fold cross-validation experiment is designed to test generalization ability. In each fold, one file is picked as the test data set. The other nine files and the CoNLL2000 & Penn Treebank from NLTK [101] are used for training. A micro performance of over 0.97 for precision/recall/F1 is reached as shown in Table 20. Besides, the PoS prediction F1 score for the normal words also reaches 0.97.

**Table 20 TnT-based PoS tagging prediction performance**

| Label | Size | Prec. | Recall | F1 |
|---|---|---|---|---|
| NML | 72 | 0.96 | 0.97 | 0.97 |
| NP | 399 | 0.99 | 0.97 | 0.98 |
| S | 74 | 0.90 | 0.99 | 0.94 |
| Avg. | 545 | 0.98 | 0.97 | 0.97 |

### V.3 Declaration extraction system description

The system diagram for the declaration extraction is shown in Figure 58. Following the existing paradigm, NP is extracted as the declaration candidates using the customized PoS tagger and NP chunker. For each NP candidate, a decision will be made whether the NP is the declaration of any ME in the same sentence. Two methods are adopted to make the decision:

pattern matching and classification. Besides, a sequential tagging framework is explored, and more declaration patterns are collected using a weakly-supervised method.

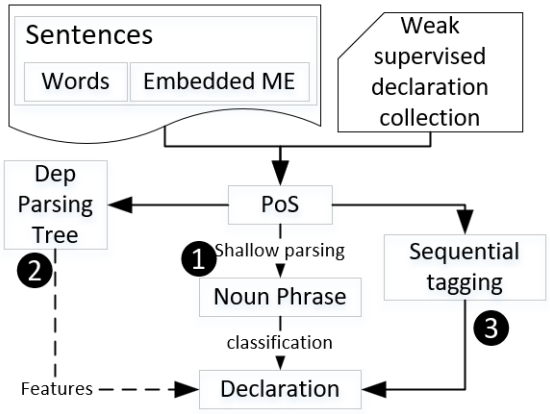



**Figure 58 The system architecture for declaration extraction**

**Table 21 Features for declaration extraction**

| Category | Features |
|---|---|
| Patterns (10) | 9 basic and the 10th is any of them is satisfied. |
| Surface value | ':', ',', other MEs between ME and DEC (3) |
| | ME/DEC in parenthesis (2) |
| | ME before DEC (1) |
| | ME-DEC token dist (1) |
| Surface/PoS Enum | bi-gram of token/pos of previous/following of DEC (4) |
| | token/pos of the first/last of the DEC (4) |
| | tri-gram of token/PoS of previous/following of ME (4) |
| | uni/bi/tri-gram of token of previous/following × first/last token of DEC (4*) |
| | verbs between ME and DEC (1*) |
| Structure | distance on dependency parsing tree |
| | whether ME (DEC) is head of DEC (ME) |
| | uni/bi/tri-gram of token seq. on the path from DEC to ME |

**Table 22 Patterns of declaration**

| | | |
|---|---|---|
| DEC ME | ME denote / mean / stand for DEC | ME is denoted / defined / given by DEC |
| ME DEC | ME is / are DEC | Let ME be denoted by DEC |
| DEC (ME) * ME | Let / Set ME denote / be DEC | denote (as / by) ME DEC |

From the view of patterns, there are two common groups: the appositions such as 'a hidden vector $h_t$' and the neighbor clues words such as 'denote' and 'as' in 'we denote $h_t$ as the hidden vector'. However, the ways are also flexible. Similar clues words or phrase could be adopted, and the order could also be reordered using passive tense. Our pattern matching based system baseline adopt 9 patterns shown in

Table 22 based on previous works [67], [81], [68]. For the classification machine learning approaches, the patterns, surface text, PoS tags, and structure features from the dependency parsing are used as features as summarized in Table 21, where 'DEC' indicates the declaration candidates.

Further, another paradigm that adopts a sequential tagging approach is tested without the generation of NP candidates. There are two implications without the NP candidates: First, all the features mentioned above are not applicable since there is no NP extracted as the candidates. Second, the features for the CRF training should cover the knowledge for NP extraction. Given these two requirements, the following features are proposed: the lower case of the token and its suffix of length 2 and 3; PoS tag and its prefix of length 2; whether the token is upper case, digit; the distance from the token to a target ME.

### V.4 Weakly-supervised learning

The weak-supervised learning process is based on the observation that 58% of the first time occurs variables are with a declaration [62]. The workflow designed to remove the left 40% noise and collect the patterns for declaration is shown in Figure 59.

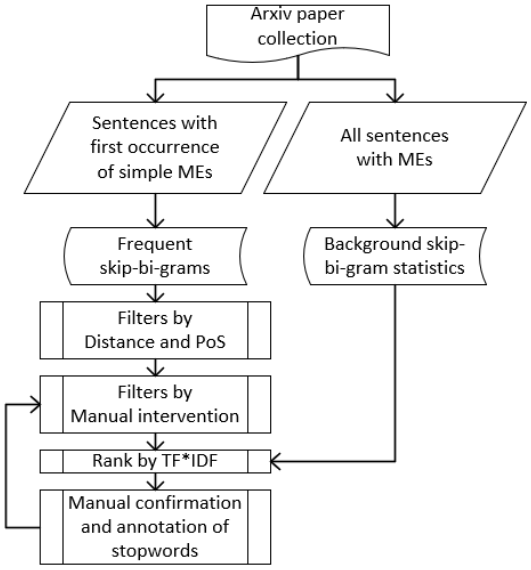



**Figure 59 Semi-automated weakly-supervised process for declaration pattern extraction**

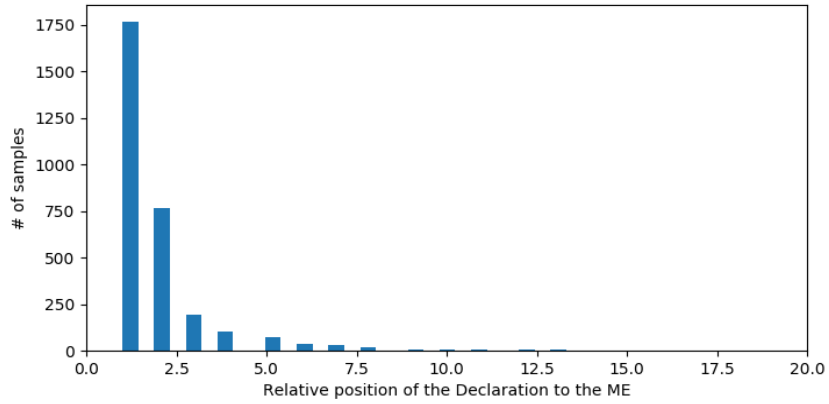



**Figure 60 Statistics of the position of the declaration relative to the ME**

The first step is to gather the possible pairs of ME and associated declaration. The MEs of concern are simple variables such as identifiers or identifiers with superscript, subscript or accent. And if there is an NP nearby, a pair of ME-DEC is created. Based on the statistics from an annotated dataset, the distance between ME and the corresponding declaration are mostly within a range of 5 as shown in Figure 60. When preparing the ME-DEC pairs for the unsupervised Arxiv dataset, an even stricter threshold is set so that only the pair with a distance of less or equal than three are considered.

The second question is what the pattern templates to generate patterns candidates are. The patterns without clues words are simply the case where ME is the apposition of DEC. The patterns with one clue word are also limited to use the word "is," "are," etc. More patterns are with more than one words such as "Let ME be DEC" or "denote DEC by ME." Given the above observation, the skip-bi-gram patterns are mined around the ME-DEC candidates, where the *skip* is to ensure the flexibility. The clues words should also be not far from the ME-DEC pair, and we use the same threshold 3.

Third, given the two tokens as the clue words, some obvious patterns will not lead to a pattern for declaration. The contributing words are mostly verb, prepositions, and parenthesis. The token pair containing words of other PoS are filtered out. A complete table of the related PoS could be found in the Appendix. The following patterns are removed from consideration:

- the PoS of both tokens are preposition "IN"
- one of the tokens is a preposition, and another word is "be" word
- there are clues tokens between ME and DEC
- there is an unmatched parenthesis.

**Table 23 Trivial patterns collected simply by frequency**

| | | | |
|---|---|---|---|
| ...of... DEC ... ME ...is... | 391 | ...is... DEC ... ME ...is... | 286 |
| ...Let... ME ...be... DEC ... | 204 | ...of... ME ... DEC ...of... | 161 |
| ...Let... ME ... DEC ...of... | 126 | ...in... DEC ... ME ...is... | 120 |

**Table 24 Manual intervention for declaration pattern extraction, round 1**

| | | |
|---|---|---|
| good patterns | ...use... ME ...denote... DEC ... | ...Let... ME ...be... DEC ... |
| | ...with... ME ...being... DEC ... | ... DEC ...denote...by... ME ... |
| | ...(... ME ...)... DEC ... | ... ME ...to...denote... DEC ... |
| | ...define... ME ...as... DEC ... | ...let... ME ...be... DEC ... |
| | ...with... ME ...denoting... DEC ... | ...(... DEC ...)... ME ... |
| | ...denote... ME ...as... DEC ... | ...denote... DEC ...by... ME ... |
| | ...Define... ME ...as... DEC ... | ...use... ME ...represent... DEC ... |
| | ...denote... DEC ...as... ME ... | ...Denote... ME ...as... DEC ... |
| ignore patterns | ...use... ME ...to... DEC ... | ...as... DEC ...where... ME ... |
| | ...to... ME ...if... DEC ... | ...exists... ME ...that... DEC ... |
| | ...is... ME ...and... DEC ... | ...in... DEC ...let... ME ... |
| | ...with... ME ...if... DEC ... | ...of... ME ...if... DEC ... |
| | ...the... DEC ...of... ME ... | ...of... ME ...th... DEC ... |
| | ...denote... DEC ...of... ME ... | ...denotes... DEC ...of... ME ... |
| | ...of... DEC ...let... ME ... | ...consider... DEC ...of... ME ... |
| | ...of... ME ...and... DEC ... | ...given... DEC ...of... ME ... |
| | …… | |
| stop words | th | the |
| | where | if |
| | that | and |
| | of | in |
| | over | consider |
| | exits | say |

**Table 25 Manually constructed patterns from mined skip-bi-gram**

| | | |
|---|---|---|
| let ME be/define DEC | use ME as DEC | use ME [to]* denote/represent DEC |
| with ME being/denoting DEC | ME refer[s]* to DEC | refer ME as DEC |
| refer DEC as ME | write ME for DEC | ME is called DEC |
| DEC represented by ME | ME corresponding to DEC | ME (DEC) |
| DEC denote[d]* by/as ME | define/denote ME as DEC | define/denote DEC as ME |

The last question is how to rank the patterns to get the possible patterns for declaration. By simply collecting the frequent skip-gram pattern around the ME-DEC pairs, there are lots of trivial stop-words patterns extracted as shown in Table 23. To avoid such a situation, the TF*IDF is used to filter out common patterns that are not specific to declarations. The TF refers to the frequency of the skip-bigram pattern. The IDF is the inverse document frequency of the skip-bigram pattern on a collection of more than 1K Arxiv paper.

Given the above procedures, there are still some skip-bi-gram that does not contribute to the declaration extraction. A human in the loop procedure is developed to manually confirm and deny the patterns, as well as building stop words for declaration extraction. The result for the first round is given in Table 24. In the table, there are some prepositional stop words to show the relationship between elements, such as “of,” “in,” and “over.” The word “th” is also very common, where people write *i*th as the indexing.

In summary, given the clues from the mined skip-bi-gram patterns. 12 patterns are constructed as shown in Table 25. They will be used as additional patterns for declaration extraction in the experiment section.

## V.5 Experiment Result and Analysis

### *V.5.1 Dataset and evaluation criteria*

The NTCIR10 math understanding dataset [4] is used for evaluation. There are 35 papers with a total 9172 MEs. There are two types of annotation: short and full. For the sentence 'Let $ME143$ be a graph with $ME144$ vertices', 'a graph with $ME144$ vertices' is called a full declaration, while the core, 'a graph', is called a short declaration. There are 3076 short declarations and 3053 full declarations. There are two evaluation modes: strict and soft. The strict matching requires exact matching, while the soft mode only requires partial overlapping. If our prediction is 'a graph' 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 are precision, recall, and F1 score.

### *V.5.3 Experiment design*

The experiments are designed to answer the following questions:

1. Is the customized MWM toolkit improve the performance of the declaration extraction?
2. Which features are the major contributing factors?
3. Will sequential tagging approach improve the performance by omitting the candidate enumeration step?
4. Do the mined declaration patterns help improve the declaration extraction?

To answer the first question, a comparison is made between the NP candidates extracted from the Stanford CoreNLP toolkit [119] and our customized MWM toolkit. Then a pattern matching module will make the final decision of the declaration identification. The experiments are named "Pattern Matching [Stanford]" and "Pattern Matching [MWM]."

For the second question, the MWM toolkit is used to extract the candidates, and an SVM-based classification approach is adapted to make the final decision. The following feature groups are added sequentially: patterns, token/Pos, dependency parsing tree. As the enumeration feature of token or tree structure is very large, a Chi-square feature selection is adopted to keep the first 1,000 features.

For the third question, the Conditional Random Field is adapted for directly sequential tagging. For the last question, the automatically mined skip-bi-gram patterns and manually constructed patterns are added for both the pattern matching and the SVM-based ME-declaration pair classification.

### *V.5.3 Result and Analysis*

**Table 26 Short declaration extraction performance**

| Method | Soft Matching | | | Strict Matching | | |
|---|---|---|---|---|---|---|
| | Prec. | Recall | F1 | Prec. | Recall | F1 |
| MCAT | 0.817 | 0.483 | 0.562 | 0.682 | 0.404 | 0.508 |
| Pattern Matching [Standford] | 0.415 | 0.261 | 0.321 | 0.311 | 0.196 | 0.241 |
| Pattern Matching [MWM] | 0.722 | 0.610 | 0.661 | 0.558 | 0.471 | 0.511 |
| SVM (Pattern) | 0.699 | 0.660 | 0.679 | 0.526 | 0.496 | 0.511 |
| SVM (Pattern+Token/PoS) | 0.729 | 0.594 | 0.655 | 0.566 | 0.461 | 0.508 |
| SVM (Pattern+Token/PoS+Dep) | 0.752 | 0.590 | 0.661 | 0.583 | 0.457 | 0.512 |
| CRF (Token/PoS) | 0.164 | 0.500 | 0.247 | 0.149 | 0.454 | 0.225 |
| Pattern Matching (Existing+Mined) | 0.721 | 0.610 | 0.661 | 0.557 | 0.471 | 0.510 |
| SVM (Existing+Mined) | 0.698 | 0.659 | 0.678 | 0.526 | 0.496 | 0.510 |
| SVM (Ex.+50 Mined Skip-Gram) | 0.696 | 0.649 | 0.672 | 0.526 | 0.490 | 0.508 |
| SVM (Ex.+100 Mined Skip-Gram) | 0.696 | 0.649 | 0.672 | 0.526 | 0.490 | 0.508 |

The comparison is made in two aspects: the candidate generation and the methodologies. For declaration candidate generation, the comparison is made between our new MWM pipeline with existing Stanford constituent parsing for NP candidate generation. From the view of the

methodologies, the comparison is made between different methods, including pattern matching, classification, and sequential tagging.

**Table 27 Long declaration extraction performance**

| Method | Soft Matching | | | Strict Matching | | |
|---|---|---|---|---|---|---|
| | Prec. | Recall | F1 | Prec. | Recall | F1 |
| MCAT | 0.873 | 0.483 | 0.622 | 0.620 | 0.373 | 0.466 |
| Pattern Matching [Standford] | 0.430 | 0.273 | 0.334 | 0.106 | 0.067 | 0.083 |
| Pattern Matching [MWM] | 0.729 | 0.620 | 0.670 | 0.447 | 0.380 | 0.411 |
| SVM (Pattern) | 0.713 | 0.674 | 0.693 | 0.422 | 0.399 | 0.410 |
| SVM (Pattern+Token/PoS) | 0.739 | 0.602 | 0.664 | 0.474 | 0.386 | 0.425 |
| SVM (Pattern+Token/PoS+Dep) | 0.762 | 0.598 | 0.670 | 0.486 | 0.381 | 0.427 |
| CRF (Token/PoS) | 0.165 | 0.507 | 0.249 | 0.130 | 0.397 | 0.196 |
| Pattern Matching (Existing+Mined) | 0.728 | 0.620 | 0.670 | 0.446 | 0.380 | 0.410 |
| SVM (Existing+Mined) | 0.713 | 0.672 | 0.692 | 0.422 | 0.398 | 0.410 |
| SVM (Ex.+50 Mined Skip-Gram) | 0.712 | 0.663 | 0.686 | 0.427 | 0.398 | 0.412 |
| SVM (Ex.+100 Mined Skip-Gram) | 0.712 | 0.663 | 0.686 | 0.427 | 0.398 | 0.412 |

The performance comparison of different method for the short/long declaration under both the strict and soft evaluation criteria are shown in Table 26 and Table 27. We have the following observations:

- First, the MWM processing is significantly improving the performance in general, which could be verified by that fact “Pattern [MWM]” is better than “Pattern [Stanford].”
- Using the SVM algorithm for candidate classification, our system outperforms the best performance of the MCAT system [81]. The pattern features carry the most weight during the SVM prediction. As more features are added to the system, the precision gets higher. But the recall goes lower.

- The CRF sequential tagging is worse than the pattern matching and SVM classification, possibly due to the insufficient training data.
- Soft matching is usually better than the strict matching. Under strict matching, the short version is better, due to that the NP candidates are generaly short.
- When adding the new patterns mined through SAWS, the performance does not gain. But the patterns are confirmed manually, and they might help in a larger scale evaluation.

## V.6 Conclusion

In this work, we identified one bottleneck for ME declaration extraction, i.e., the processing mixed Word-ME (MWM) sentences. The customized PoS tagger and NP chunker are proposed to enhance the preprocessing. Evaluation on Elsevier dataset shows that the customized PoS tagger could greatly enhance the PoS tagging performance for MWM sentences. The declaration extraction performance is also greatly enhanced using the NP candidates generated from the customized processing toolkit. Comparisons show that the declaration pattern features play the most important role both for pattern matching or SVM-base classification-based declaration classification. A semi-automatic weakly-supervised approach is proposed to mine more patterns from a large collection on unlabeled data. Manual inspection shows that many new patterns are identified. However, there is no performance gain when applying these new features to the NTCIR dataset. Quantitative verification is expected for the future large-scale experiment.

# CHAPTER VI

# QUALITATIVE-QUANTITATIVE MAPPING OF SCIENTIFIC PUBLICATIONS*

## VI.1 Overview of the chapter

It is a rewarding experience to understand the technical materials, but there are three aspects of challenges in the existing practice:

- Technical writing transforms complex interrelated scientific abstractions into a linear representation based on the mixed use of words and mathematical language. To digest the original idea, one must walk forward and backward through a paper to reestablish the complex relations from the linearized writing, as well as look up the external materials. Missing a subtle point may impede a reader from capturing the essence of a paper. Also, given the different background of the reader, she/he might want to read it in a different way rather than the order presented by the author.
- Papers contain lots of redundant information. Our experimental outcomes showed that the amount of MEs and their relevant words could be very dense. Sometimes, too many MEs used for the formalism of a presentation may even interfere with the understanding of the key logic flows which are carried by less frequently used MEs.
- MEs and words are carefully bonded in technical writing to characterize physical concepts and their interactions *quantitatively*, and *qualitatively.* The mapping between the physical world and the abstract math world should be done through the declaration.

---

*Reprinted with permission from "QuQn Map: Qualitative-Quantitative Mapping of Scientific Papers" by Wang, Xing, Lin, Jason, Vrecenar, Ryan, and Liu, Jyh-Charn, 2018. Proceedings of the 2017 ACM Symposium on Document Engineering, Halifax, Nova Scotia, Canada, 2018. Copyright 2018 ACM. https://doi.org/10.1145/3209280.3229116

Given the rich analytical products from the ME analysis and ME-word bonding mining, in this chapter, the Qualitative-Quantitative (QuQn) map (also known as QuQn graph) is proposed as an abstraction of scientific papers to depict the dependency among MEs and their most related adjacent words. QuQn map aims to offer a succinct representation of the reasoning logic flow in a paper.

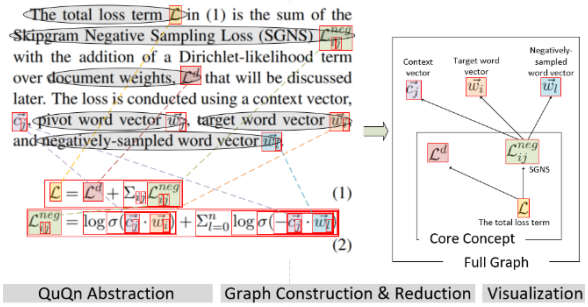



**Figure 61 QuQn map architecture, reprinted with permission from [120]**

The QuQn map supports interactive rendering of words and MEs based on their qualitative and quantitative dependencies. QuQn map supports the selective pruning of nodes and links based on different filtering rules. It uses spatial layout and color style to highlight the dependency relationship among automatically discovered MEs and words. The first processing step of the QuQn map is QuQn ME extraction and transformation. Given an ME expressed in LaTeX, Presentation MathML (PML), Content MathML (CML) or PDF, we extract, parse, and convert the MEs into a semantic taxonomic structure, which can be further decomposed into sub-expressions for high-level semantic analysis. Examples of the parsed MEs marked in the nested red rectangle boxes are shown in Figure 61. Each ME paired with its declaration words is compared with other MEs to generate their dependency graph. Based on different filtering rules,

the user can prune the dependency graph for QuQn map to render only the needed information on the limited 2D display space. The QuQn visualizer is integrated with an online pdf viewer as a reading assistant.

Following the visualization principles by Tufte [121], the linkage among MEs are manifested by their spatial affinity. Information on demand is achieved by highlighting the dependent MEs and synchronization between the QuQn map with the original PDF file. Searching ability provide a unified interface for the user to quickly locate the related information by typing in word description or ME Latex code. In this way, readers could quickly grasp the main essence of an idea and follow the inference process.

Various filters can be applied to a QuQn map to reduce redundant/indirect links, control the display of problem settings (simple ME variables with declaration), and prune nodes with specific topological properties such as the largest connected subgraph. A visualization tool prototype is developed to support interactive browsing of the technical contents at different granularities of detail.

## VI.2 QuQn abstraction and essence graph construction

### *VI.2.1 QuQn abstraction construction*

The QuQn map is represented by a set triple $\langle M, E, D \rangle$. From the quantitative aspect, $M = \{m_i\}$ denotes the set of MEs and their sub expressions in a document $U$, $E$ the set of links indicating the dependency among MEs. $D$ is the set of ME denotations which can be a word description from the qualitative aspect or other equivalent MEs. Note that an ME may have multiple denotations, yet some MEs have no denotation. Extraction of equivalent or related MEs from $U$ requires an understanding of the semantics of MEs. As such, we first introduce our semantic taxonomy of ME and the notion of "equal," "sub-component," and "left hand side" of

MEs. Then, the formulation of ME denotations for linking of MEs and association between MEs and words is presented.

#### VI.2.1.1 ME Processing

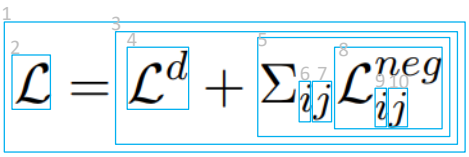


**Figure 62 Examples to illustrate ME decomposition, parts of this figure are adapted from LDA2Vec [122]**

As described in the ME semantics analysis section, MEs from different sources are converted into our own ME semantic taxonomy. Matching at the semantic level, rather than the character/layout level, will significantly reduce the false positive. After transforming the ME into the object according to the ME semantic taxonomy, the ME could be viewed as a hierarchical composition structure. For example, the ME $m_1$ ($L = L^d + \sum_{ij} L_{ij}^{neg}$) is decomposed into 10 (sub)expressions $\{m_k\}$ with the subexpression number $k$ marked on the left top corner in Figure 62 from the paper *lda2vec* [122]. The superscript *d* and neg are not marked as ME as they only play as notations rather than variables. On the other hand, the subscript $i$ and $j$ are marked as ME as they are actively used for indexing. The direct subcomponents of $m_k$ is denoted as $\Phi(m_k)$. For example, the ME $m_2$ ($L$) and the ME $m_3$ ($L^d + \sum_{ij} L_{ij}^{neg}$) are the direct subcomponents of $m_1$. $\Psi(m_k)$ is denoted as all subcomponents of $m_k$. In this example, $\Psi(m_1) = \{m_i\}_{i \neq 1}$.

### VI.2.1.2 Denotation and Link Identification

Denotation refers to the semantic equivalent information for an ME. A denotation of ME can be a qualitative description expressed in words or a quantitative description expressed by another ME. And they are called word denotation and ME denotation, respectively. Denotation is critical for the detection of relations between MEs and linking an ME to their related words. There are three related concepts for ME denotation and ME relation extraction:

- $m_i = m_j$ if the two expressions are the same, such as the case of $m_6$ and $m_9$.
- $m_i$ is a subcomponent of $m_j$, denoted as $m_i \in m_j$, iff $\exists\ m_{i'} \in \Psi(m_j),\ m_i = m_{i'}$.
- The left(right)-hand side function $L(R)HS$ is used to represent ME types such as relation expression and function declaration expression. For example, $\text{LHS}(m_1) = m_2$. If there is no valid LHS, $\text{LHS}(m) = null$.

The set of ME denotations $D_M$ is constructed so that each element represents an ME $m$ that has the elements of LHS, RHS, and the relation "=", where the denotation is expressed as $\langle LHS(m), RHS(m)\rangle$. In addition to ME denotations, one ME $m_i$ may optionally associate with a word denotation consisting of a sequence of words $W_i = \{w_i^j\}$. The declarations are extracted based on the pattern matching described in the previous chapter. By the end, the $\langle m, W\rangle$ are obtained to form the set of word denotations $D_W$ for a document.

Given the ME-based denotation $D_M$, the linkages among ME $E$ are identified as $\{\langle m_i, m_j\rangle\colon\ \langle m_i, m'\rangle \in D_M,\ m_j \in \Psi(m'),\ m_i, m_j \in M_d^r\}$. The condition $\langle m_i, m'\rangle \in D_M,\ m_j \in \Psi(m')$ states that $m_j$ is the subexpression of $m'$, which is equivalent to $m_i$.

*VI.2.2 essence graph construction*

The QuQn map triple $\langle M, E, D \rangle$ represents a significant reduction of information from its original document. When all elements in $\langle M, E, D \rangle$ are included, the graph can become overcrowded with low-level details and repetitive occurrences of certain MEs/words. To improve its readability, the essence graph is proposed and its progressive visualization of publications based on the following pruning rules.

An essence graph can be reduced from the raw QuQn map based on node pruning and link pruning. Node pruning is based on three rules:

- The first criterion is to keep the MEs with denotation only for users to understand the semantics of every ME node. After the first filtering, we get $M_d = \{m: \langle m, * \rangle \in D_M \cup D_W\}$.
- The second criterion is to remove duplicate occurrences of an ME. Formally, among the MEs with denotation $M_d = \{m_{d_i}\}$, the MEs with multiple denotations are removed to get $M_d^r$. That is, if two equal MEs $m_{d_i} = m_{d_j} \in M_d$ are in the reduced set, then only the first $m_{d_i}$ in the ME with denotation is kept.
- The numerous MEs for the problem settings often clutter the essence graph even after the pruning steps above. The third heuristic detects and removes MEs primarily for problem settings, including 1) identifiers (with optional sub/superscripts, accent), which do not interact with others, and 2) relation expression with a constant number on the right-hand side, which are usually the detail of the implementation.

These conditions eliminate a significant number of nodes, but the dependency graph can be still too crowded for visualization. A further step of link pruning can be based on the following two rules:

- Remove indirect edges. If there exists an edge $\langle m_i, m_j \rangle$, $\langle m_j, m_k \rangle$, then edge $\langle m_i, m_k \rangle$ also exists, which is indirect because of the intermediate node $m_j$. The indirect edges clutter the graph without adding new information.
- Keep only the largest connected subgraph, which in its effect likely removes local discussions.
- The graph is reduced into acyclic *dependency tree* by removing edges to increase the readability of the essence graphs. This is based on the observation that the vast majority of quality works actively avoid circular reasoning.

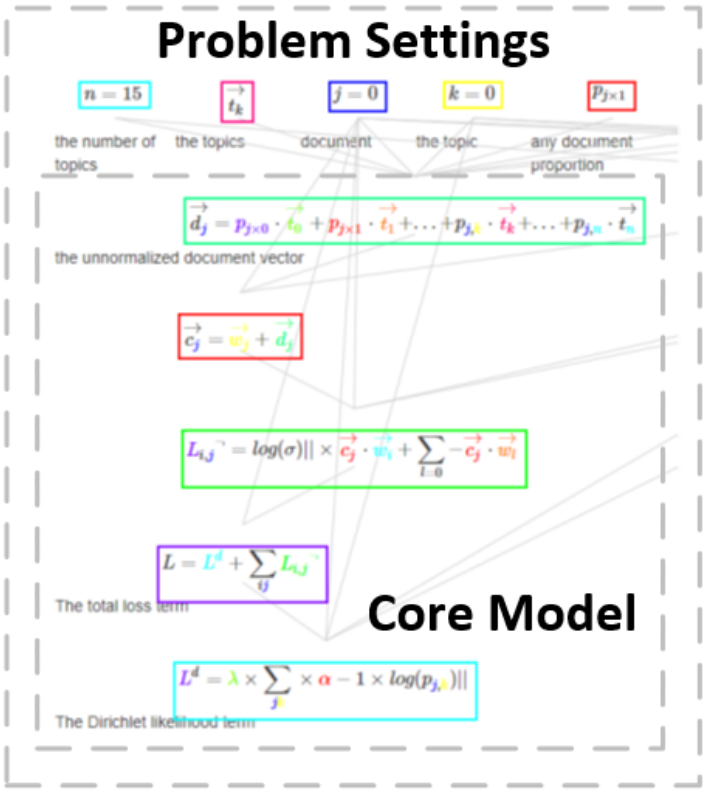



**Figure 63 Colored visualization of a (cropped) essence graph pruned from its raw QuQn map, parts of this figure are adapted from paper LDA2Vec [122]**

The final essence graph generated for the paper lda2vec [122], after all the pruning steps, is plotted in Figure 63 as a tree hierarchy. The atomic building units such as variables are located on the top, and the high-level compounded expressions such as the problem formulation are placed at the bottom. Note that this tree visualizer is feasible only for small papers with not too much MEs. For a big paper with a large amount of MEs, a dynamic force based interactive graphical visualizer will be introduced in the next section. The number of MEs is reduced by 75%, or only 25% of MEs were retained in the resulting essence graph. A very similar level of compression rate was achieved for three other papers with Arxiv identifier 1412.5567, 1508.04395, 1806.07495. The entire process to generate the essence graph from the four files was manually validated. Note that the pruning rules are not generalized. For other papers, different pruning rules may be more effective to produce optimally minimal essence graph(s) to capture the core model(s) and their problem settings.

### VI.3 Essence graph visualization

Given the graph constructed and pruned, the next task is to design the user interface to visualize the dependency graph using the location combined with color to meet the following needs:

- Visualize the essence graphs of different sizes and different topological structure
- Highlight the linkage among MEs through the spatial affinity, interaction animation, and customized location.
- Navigate between the source file and the essence graph visualization easily
- Locate the desired information quickly

The system interface that meets the above criteria is shown in Figure 64. First, to accommodate the visualization of graphs of any size, an infinity drawing space is created in which the user could easily surf around through pan and zoom in/out operations. Second, to highlight the linkage among MEs, mass is added to each node to attract each other through gravity, and charge assigned to nodes as the repulsive force.

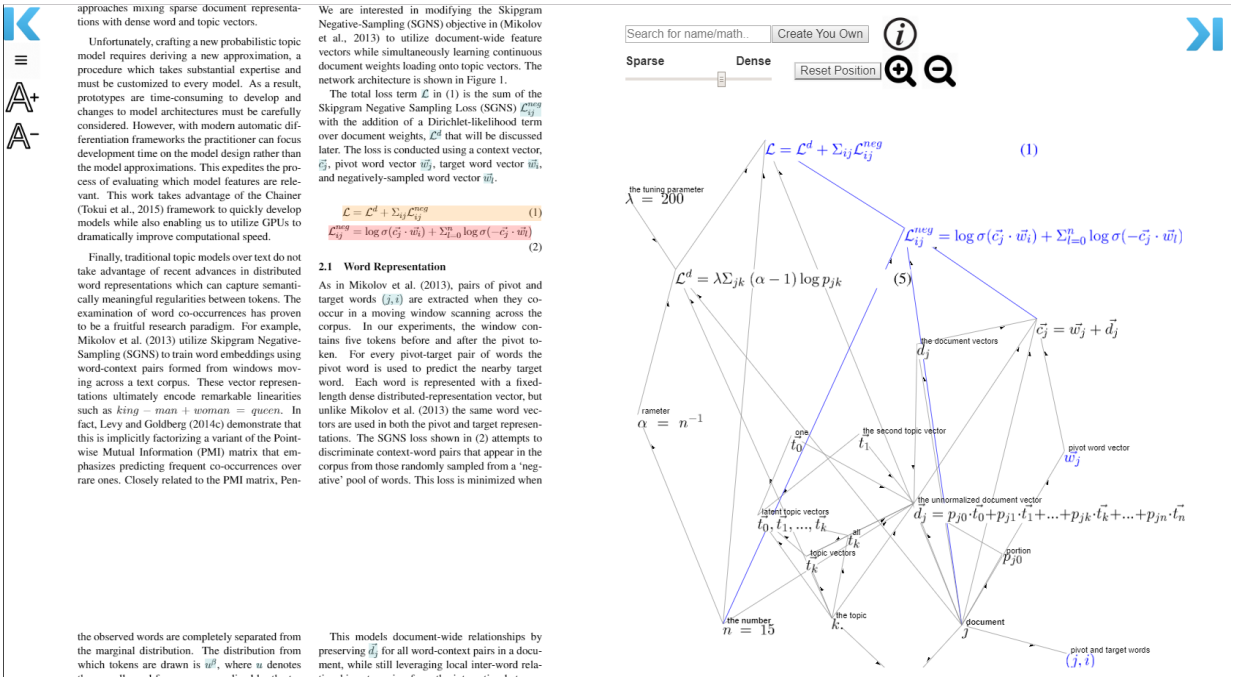



**Figure 64 Graphical user interface (GUI) for the QuQn visualization and interaction, parts of this figure are adapted from the paper LDA2Vec [122]**

To further highlight the linkage among related MEs, unrelated MEs will hide when the mouse holds on a specific ME. The user could also customize the location of certain MEs by dragging them to the desired locations. One could also hide specific MEs if they do not contribute to the understanding of the system. Third, for the easiness of navigation between the

source file and the essence graph, color encoding is used. When an ME is clicked, the source file will scroll to the page where the ME is located, and the ME on the PDF will be highlighted as red. When the source file is scrolling, the MEs on the page will show as blue on the dependency graph visualization. At last, to better serve the need for ease of locating information, a search bar is provided where the user could type in text description or an ME. Related MEs will be highlighted by the orange color and larger font size.

The system is implemented using JavaScript libraries, d3, and pdf.js. The image for MEs is cropped from PDF files and colored. The information for ME, declaration, essence graphs are pre-calculated and might be manually corrected.

### VI.4 Summary

This section presents a novel abstraction of technical papers called the Qualitative-Quantitative (QuQn) map to represent MEs and their ME-ME and ME-word dependency relationships. The sequential elaboration from the original technical material is segmented and re-arranged in the graph format to show the dependency among different factors. Information overload is a critical problem for content analysis. Node/link pruning is a crucial process to control the amount of information display that captures the relationships among MEs in an essence graph, especially for the analysis of complex papers. The proposed progressive pruning heuristics appear to be highly promising for the design of automated pruning solutions. All the ME related analysis is conducted at the semantic level for better accuracy, include the dependency recovery and the pruning.

# CHAPTER VII

# APPLICATIONS OF MECA

In this chapter, we will show three use cases of the MECA system. The first use case applies MECA system in an educational environment to help high school students to understand the dependency relationship among factors in the rotational physical system. The second use case creates a mapping and analyzes the evolution of knowledge through the analysis of the conference paper in NIPS. The last use case is to illustrate the possible effect when comparing the technical essence of publications using QuQn.

## VII.1 User study in AggieSTEM summer camp

The learning process is an iterative process that first locates the relevant material, then consume the knowledge, and apply the learned knowledge. The search engine has greatly enhanced the experience in finding relevant materials, but the tools to boost the learning process are mostly under development. One of major bottleneck is that the low-level digit file could not be recovered efficiently as structured information to serve high-level learning applications. However, our MECA system provides a foundational solution to overcome the gap, and the next question is how to use the structured information.

Learning is a process that internalizes the concepts and knowledge expressed in a medium such as languages and mathematical notations. “Cognitive learning theory suggests that the brain learns most effectively by relating new experiences and knowledge with previous knowledge” [123]. Being able to automatically discover highly related system concepts, which are mostly represented in MEs and words from technical papers, will significantly improve the learning experience and the research productivity. In this work, we propose to fill the gap by creating a

graphical mapping of scientific publications to highlight the connection between different factors as well as the mapping between the abstract math notation with the physical world.

A pilot user study during a high school summer camp is conducted to evaluate the effectiveness of QuQn map. The experiment settings are first presented, followed by the pre/post questionnaires and the evaluation metrics. At last, the quantitative results are given.

### *VII.1.1 Experiment settings*

The study happened during a high school physics summer camp. The control group has 15 students and the experiment group has 16 students. During the summer camp, they will learn rotational physics concepts as well as apply the knowledge to design a spinner. The related concepts include "rotational velocity," "rotational inertia," "angular momentum," and "rotational kinetic energy." Their linkage relationship is shown Figure 65.

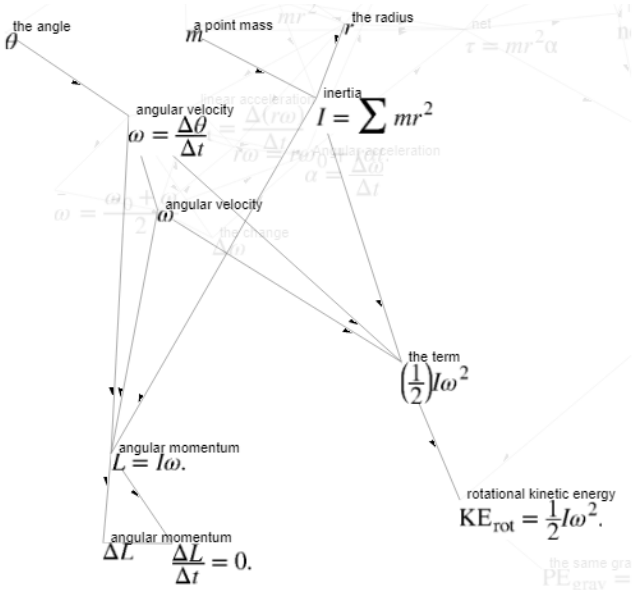



**Figure 65 The final QuQn map created by the teacher, parts of this figure are adapted from [124]**

During the physics knowledge learning phase, both groups are allowed to use whatever resources they can access, while only the experiment group is exposed to our system. In the experiment group, when explaining the concepts, the teacher frequently use the QuQn system to explain the linkage among different factors.

### *VII.1.2 Questionnaire and evaluation metrics*

The same questionnaire is taken both before and after the summer camp. The questionnaire covers both the attitude, knowledge, and connection among concepts. There is a significant increase in the attitude and connection questions for the experiment group in comparison with the control group and a modest increase n the knowledge aspect. In this dissertation, the attitude questions are emphasized. The following attitude questions are asked in both the pre and post questionnaire.

1. I feel that complex physics concepts are approachable.
2. I am comfortable exploring new topics in physics.
3. I understand the ways in which physics concepts are related to each other.
4. I enjoyed learning physics.
5. I am able to learn difficult physics concepts.
6. The Mathematics Equations Map helped me to understand connections between concepts (Exp Posttest only)

The questions are answered by a value from 1 – 5, where the larger value indicates better opinion. The goal of education is to increase the mean value and reduce the difference among students. The pre-post effect size (PPES) measurement [125] is commonly used to measure the difference before and after an experiment:

$$PPES = \frac{\mu_{POST} - \mu_{PRE}}{(\sigma_{POST} + \sigma_{PRE})/2}$$

The larger PPES is, the better effect the education process is. Both the large difference in the gain or smaller standard derivation will contribute to the increase of PPES score. Further, to evaluate the effectiveness of our QuQn map, the percentage gain (PG) of the experiment group versus the control group is calculated as:

$$PG = \frac{PPES_{EXP} - PPES_{CONT}}{PPES_{CONT}}$$

*VII.1.3 Quantitative results*

The statistics of the mean and standard derivation for the pre/post testing of the experiment and control group are shown in Table 28. For the last question, the high school students mostly agreed that the QuQn map helps them understand the connection between concepts. For the other questions, it would be more meaningful to look at the comparison between the experimental group and the control group as shown in Table 29.

**Table 28 Mean and Std. statistics for the pre/post**

| | | Pre-test | | Post-test | |
|---|---|---|---|---|---|
| | | Exp | Control | Exp | Control |
| Q1 | Mean | 3.75 | 3.80 | 4.00 | 4.13 |
| | Std. | 0.68 | 1.08 | 0.73 | 0.83 |
| Q2 | Mean | 4.44 | 4.00 | 4.44 | 4.13 |
| | Std. | 0.73 | 1.07 | 0.51 | 0.83 |
| Q3 | Mean | 3.31 | 3.13 | 4.31 | 3.87 |
| | Std. | 0.95 | 1.19 | 0.70 | 0.99 |
| Q4 | Mean | 3.94 | 3.53 | 4.13 | 3.67 |
| | Std. | 1.00 | 0.83 | 0.72 | 0.98 |
| Q5 | Mean | 3.50 | 3.53 | 4.00 | 3.67 |
| | Std. | 0.97 | 0.92 | 0.73 | 0.98 |
| Q6 | Mean | - | - | 4.38 | - |
| | Std | - | - | 0.96 | - |

**Table 29 Comparison of pre-post**

| | Q1 | Q2 | Q3 | Q4 | Q5 |
|---|---|---|---|---|---|
| Exp Pre-Post Effect Size | 0.35 | 0.00 | 1.20 | 0.22 | 0.58 |
| Cont Pre-Post Effect Size | 0.35 | 0.14 | 0.67 | 0.15 | 0.14 |
| Percentage Diff Exp vs. Cont | 2% | N/A | 79% | 47% | 314% |

There is a significant gain for the experiment group vs. the control group for question 3 and 5. For the question 3, the experiment group shows that they could understand more about how the physics concepts are related to each other. This result indicates that the goal of the QuQn map is met. Based on question 5, the QuQn map boosts the confidence of the student to learn complex concepts. In summary, this pilot study shows that the QuQn map could help the students understand the linkage among concept and boost their confidence in learning complex systems. The user study shows that the QuQn map helps the students understand the dependency among different factors and increased their confidence in learning complex systems.

### VII.2 Knowledge mapping and evolution analysis

Understanding the evolution trends of topics is valuable for both researcher and funding agents to locate valuable research topics and methodologies. The similarity assessment between papers is the key for clustering and evolution analysis. The existing work for evolution analysis is mostly based on co-citation clustering [126] and co-word clustering and topic modeling [127]. On the one hand, the citations might have different purposes as studied by Teufel [128]; being cited together might not reflect the similarity of their technical essence. On the other hand, using all the words in the document is at a coarse grain and might introduce noise for the content

analysis. Secondly, it is also a challenging issue to visualize the topics, links among papers together with their evolution.

In this work, the similarity among papers is measured based on the declaration words. The declarations are related with the technical essence such as 1) the mathematical methods used, function, matrix, vector, bandit, policy, etc. 2) the problem settings that link the variable with the real-world concept such as the webpage, image, etc. Meaningful clusters are detected at the level of research methodology and compare against the cluster based on the full document. Further, the similarity also provides a secondary checking for the strength of the citations for their technical relevance.

*VII.2.1 Declaration-based document clustering*

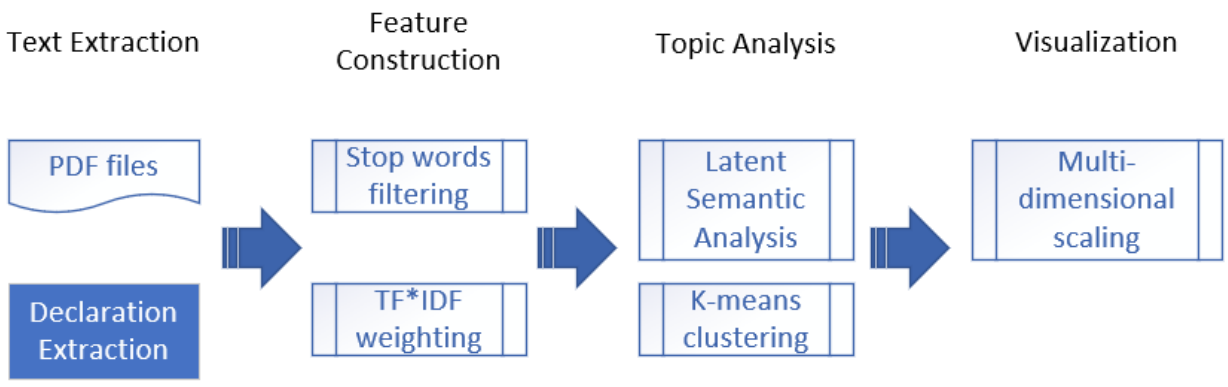



**Figure 66 Pipeline for declaration-based topic mining**

Feeding all the document to the topic modeling procedure might introduce too much noise that leads to degradation of clustering performance. In this work, we proposed to feed the text of the declarations, which are highly related to the problem settings and the mathematical methodology. The other processing steps in the pipeline shown in Figure 66 are all common

techniques used in NLP and data mining. The pipeline starts with declaration extraction as described in Chapter V. All the declaration within a PDF file are merged as a new document representation. The new document will first be filtered by stop words [101] and then construct the vector space model. Each dimension is re-weighted by the IDF value. Then the re-weighted TF*IDF model is passed to the latent semantic analysis [129] (LSA) module to reduce to a lower dimension representation, in which spectrum clustering [130] will be applied. Spectrum clustering is preferred over the k-means as it could capture the manifold structure. At last, a multi-dimensional scaling (MDS) techniques is used to map the low-dimension representation from LSA (still larger than 3 dimensions) into 2D for visual inspection of the clusters.

The generated clusters are described by the top words with the largest weight based on the LSA. Further, the declaration-based clusters are compared against the full document-based analysis as shown in Figure 67. In general, the declaration-based clusters capture the essence of the methodology used while the full document-based analysis captures the topic domain.

- For the cluster concerning clustering algorithms F1 based on the full document, it consists of two major clusters from the declaration based analysis: (D2) kernel-based method such as spectral clustering and (D6) node/edge graph theory-based method.
- For the convex optimization cluster F2, the corresponding papers mostly lie in the (D2) kernel based convex optimization cluster by declaration analysis.
- For the kernel classification cluster F3, their technical essences are related to kernel (D2) and the loss formulation (D3).
- The latent topic modeling (F4) has three parts: (D1) document topic modeling, (D4) neural network, and (D5) latent probabilistic model
- The object detection topic (F6) is mostly related to the (D4) neural network.
- The reward action, re-enforcement learning (F7) is also detected by the declaration-based clustering (F7) as this is a more theoretical work.

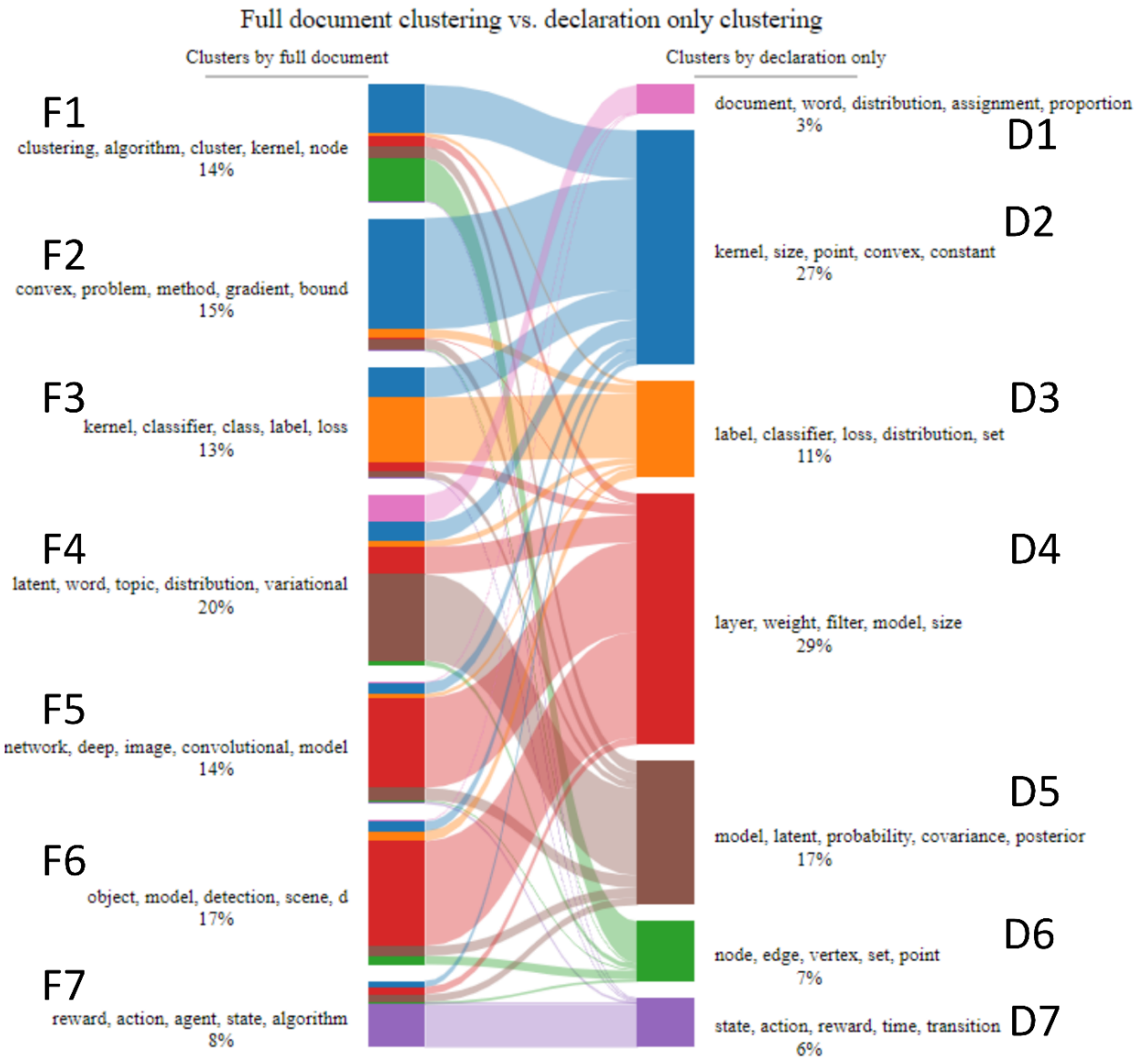



**Figure 67 Full document-based clustering vs. Declaration-based clustering**

*VII.2.2 Evolution visualization and analysis*

Given the clusters obtained from the view of their research methodology, visualizing the trends of the research methodology would be very helpful in the curriculum design for education purposes and the understanding the fundamental research methodology for the funding agencies.

From the view of the visualization techniques, it is desirable to show the evolution in the limited 2D space that could accommodate any many years as possible. It is also desired that the global trends could be easily identified and while preserving the ability to inspect the local detail. For such purpose, the evolution wheel is designed as shown in Figure 68.

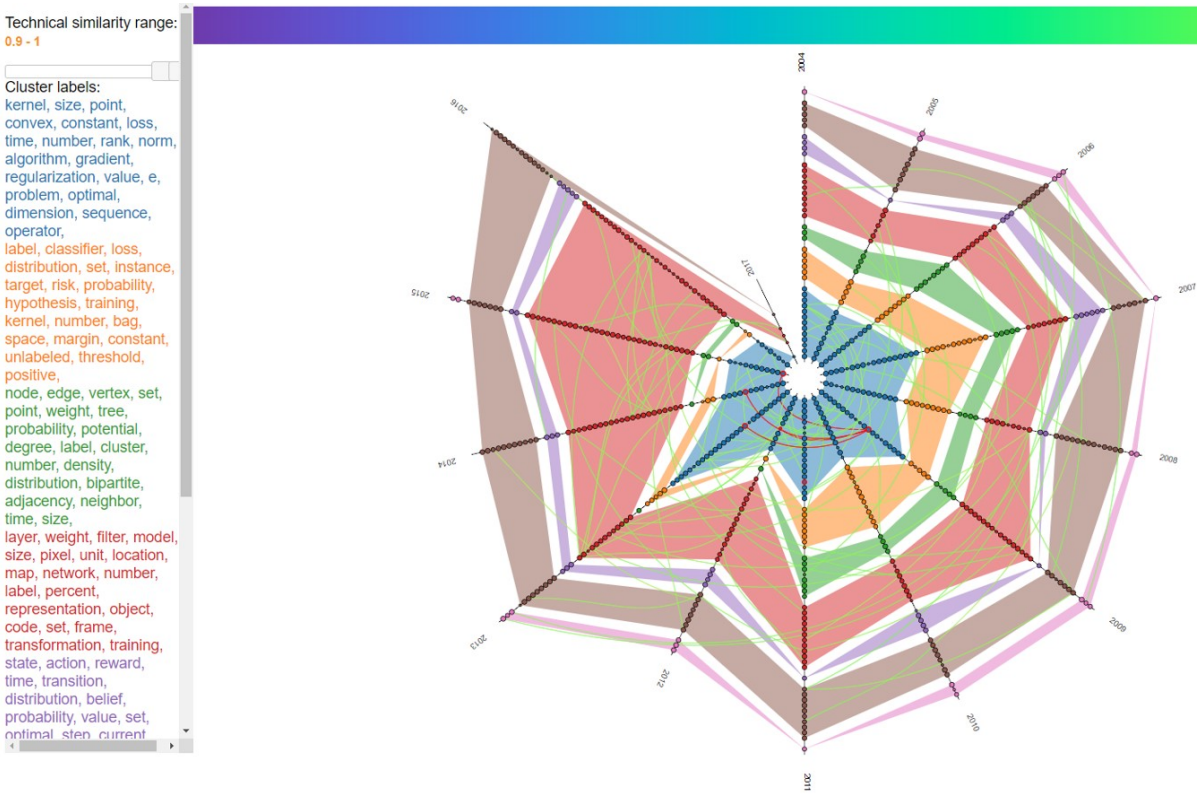



**Figure 68 Evolution visualization for the Neural Information Processing System (NIPS) conference papers (2013-2017)**

In the evolution wheel, the papers in each year are manifested as dots on a line. The dot size is positively related to the number of citations. The color of the dot for papers is related to the topic from the declaration clustering. Papers belonging to the same topic are grouped for ease of identifying the topics, its scale, and comparison across years. The links among papers are

shown as curved lines. The color tone shows the strength of the similarity from the declaration topic analysis. The dashed line indicates that the similarity strength is smaller than a certain threshold.

To easily identify the citation influence among papers, interaction features are provided for both the nodes and links. When hovering over one paper, the paper, the cited papers, the citing papers, and the links are highlighted with red boundary. After clicking on the paper, the information mentioned above will be shown on the top right information panel. When clicking over the citation link, two related papers are shown. One could click the hyperlink to view the paper and the associated QuQn map. Besides, a similarity range filter is provided to remove the citation links that are not "technical relevant."

From the topic evolution graph, we could find three major topics: convex optimization, neural networks, and latent probabilistic modeling.

- The convex formulation is popular until 2013 and decreases since then.
- The Neural Network (NN) based formulation is getting more attention. From the correspondence between full doc and declaration-based analysis, there are three driving forces: latent topic modeling, Convolutional NN (CNN) for image classification, and object detection.
- The last one is the probabilistic latent topic modeling.
- Some small topics are dying such as the cluster, graph structure modeling, manifested as green.

### *VII.2.3 Weak citation linked analysis based on declaration similarity*

Besides capturing the trends of the techniques used, the declaration similarity could also be applied to detect the weak citations. The five citations links in Figure 69 are manually

verified. The citations with the smallest similarity are not meaningful. For the citation link 1, the syntactic topic model in the year 2008 [131] is only one application of the Poisson Dirichlet process for topic mining [132]. For the citation link 2, they are both related to Latent Dirichlet Analysis. But one paper focus on the distributed learning [133] and another focus on online learning [134]. This online learning paper is cited in two later papers through link 3 and 4, which are general multi-task Bayesian optimization [135], [136] as a generalization of the previous paper. For link 5, these two papers [137], [138] only worked on the same topic but using different methodology. For the citation link 6, the later paper [139] concern policy design where the convex loss function optimization is based on the previous paper [140].

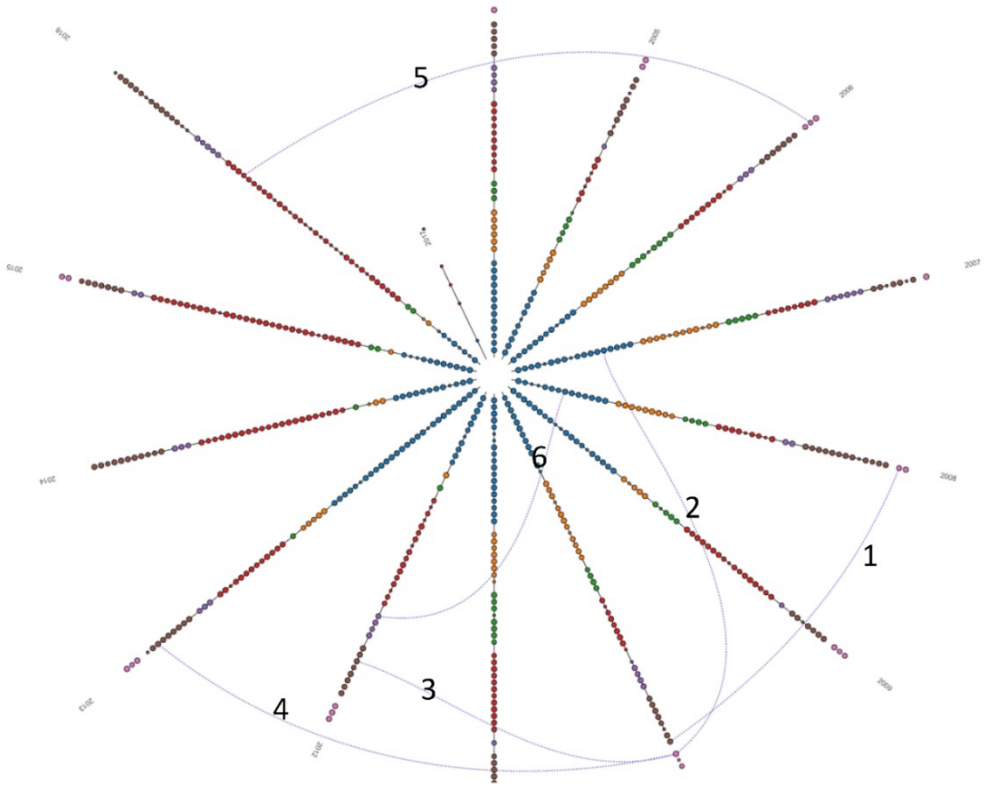



**Figure 69 Weak citations**

### VII.3 Differential publication analysis by QuQn map

Comparing the technical essence between publications is non-trivial due to differences in the notations and problem formulation. Given that the QuQn map is an abstraction of the publication, the notation differences could be potentially overcome by matching the declaration, and the problem formulation could be represented as the dependency graph in QuQn. It must be admitted that the matching of MEs by their declarations is still an open problem due to the flexible way to describe the concept in natural language. The matching of the logic flow between papers is also very hard as it involves the testing the semantical equivalence. These challenges are worth exploring for future work.

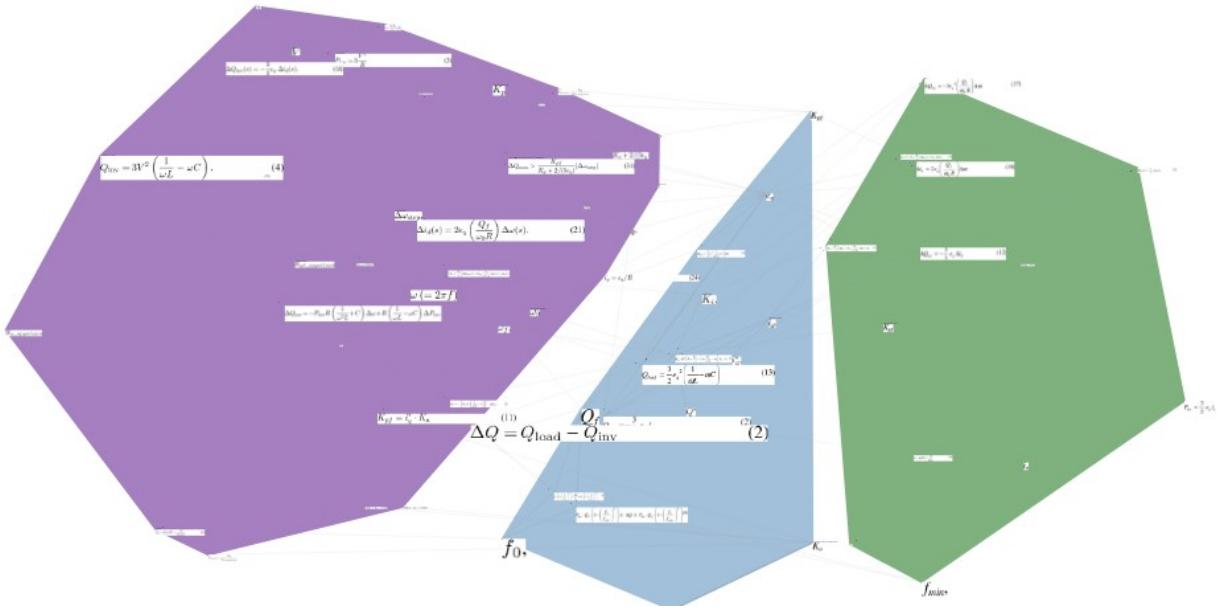



**Figure 70 Illustration of differential publication analysis, parts of this figure are adopted from [141] and [142]**

In this section, a simple example is given to illustrate the potential of applying MECA for differential publication analysis. The notation systems of the two papers are the same, given that the two papers selected [141] [142] are by the same author on the same topic. The QuQn maps of the two papers are constructed manually with necessary correction for the ME Extraction and ME semantic analysis. The MEs between two papers are matched based and the tree edit distance [117] on the corresponding operator tree as described in the ME semantic evaluation section. The distance is further normalized by the maximum number of elements in the two operator trees of concern. If the normalized distance for two MEs is less than 0.1, the corresponding nodes in the QuQn graph are merge into one. The merged QuQn graph is illustrated in Figure 70, where the common part of the two papers is placed in the middle and the different parts on the two sides. From this example, we could see that the QuQn based differential analysis could easily help the reader understand the overlapping of the concept between papers. It could also help reviewers to understand the new contribution of a submission easily.

### VII.4 Summary of MECA applications

In summary, the rich analytical products of MECA could support a broad spectrum of applicability ranging from the end-user (students/researchers) to the stage-holder (funding agency). The rich metadata could help users easily navigate through the technical material and understand the dependency relationships. Given the captured technical essence, the MECA system provides new insights into the evolution of the research methodologies.

# CHAPTER VIII

# CONCLUSION

MECA is designed to help the management and consumption of technical knowledge tied to mathematical abstractions. To meet such needs, a sequence of information extraction and transformation are designed to overcome the gap between the high-level semantics and the low-level digital representations. MECA system has significantly progressed regarding automated ME extraction, ME analysis, and their semantic bonding with words. It lays a solid foundation for the development of the next generation of deep content analysis solutions. Important lessons learned from the research and the potential future directions are discussed next.

## VIII.1 Summary of research findings

Our work starts with a weakly-supervised typesetting-based Bayesian (TSB) model for the identification of mathematical expressions (ME) from PDF files. To capture the customized font usage, a weakly-supervised methodology is applied to identify characters for ME and non-ME to estimate the posterior probability of a character as ME or NME. Then, the Bayesian inference is applied to the atomic physic units, non-separable character sequence (NSCS) to identify EME segments. Due to the discrepancy between the physical layout and the logical units, one EME might be split into multiple NSCS. At last, a Markov Random Field based sequential model (MRF-TSB) is applied to merge the over split EME segments by adding a pairwise potential. Experiment results show that the TSB model outperforms the state-of-art SVM based method by 10% in the miss rate and false rate. The MRF-TSB model significantly reduces the number of partial matching (by 1/3), which is crucial for the later stage of ME layout and semantic analysis.

After the identification of ME from PDF, the next step is the recovery of the ME layout hierarchy from the typesetting representation. Given the ME layout as a hierarchical structure, it is necessary to correctly group the characters into MEBlocks and make an assessment of the relative spatial relationship between MEBlocks. However, ambiguities might happen during the grouping process. The upper bound of the performance for discriminate analysis is limited by the overlapping of the distribution directly calculated from the character glyph and the bounding box of the MEBlocks. To overcome these challenges, a content-constrained spatial (CCS) model is proposed. The character dominance is first applied to identify MEBlocks such as accent and fraction structure. Then, by recovering the normalized height and vertical center, partial of the characters lying on the same baseline could be accurately recovered for the pre-merging of consecutive alphabets or playing as constraints in the later stage of super/subscript resolution. The normalized height and vertical center are also important in the design of high discriminative features, height ratio (HR) and normalized vertical center difference (NVCD), for the super/subscript resolution. Further, to avoid the local errors, a global inference model is proposed where the character values and confidence same baseline assessment play as constraints for the modeling. For the efficiency of the inference, a parametric approximation is proposed that fit the HR and NVCD features into lognormal distribution. Experiment results show that our CCS model outperforms the state-of-art algorithms in multiple evaluation criteria, with the target-ground truth edit distance decreased by more than 1. The analytics of the centerline analysis also provides a basis for the post check to identify the miss-predictions.

Though the ME layout already contains rich structure information and heavily used in mathematical information retrieval system, there is still a gap with the ME semantics which is crucial to the accuracy of high-level task such as dependency analysis. The first challenge for

ME semantics analysis is the lack of a standard and the common evaluation dataset. Though the MathML and OpenMath cover many core concepts, they still do not cover many mathematical dialects. This leads to the difficulty in the annotation of the ME semantics. From the view of the ME semantics recovery process, ambiguities at different levels are hindering the correct understanding, including the symbol level, structure level, and interpretation level. To resolve the ambiguities, a three-phase ME semantics understand framework is proposed. The first phase tokenizes the characters and assigns the terminal tokens to them. It will merge the multiple character identifier based on the normalized pointwise mutual information score. The spatial relationship is converted to special spatial token. The second phase is a probabilistic context-free grammar to build the abstract syntax tree, where the probability reflexes the likelihood of the syntax tree to be observed in a larger collection of the training dataset. The third phase is designed to resolve the different possible semantics under the same syntactic structure. The ground truth data collection is from the later user study experiments. Evaluate based on the exact matching and structure similarity both show that our ME semantics parser outperforms the state-of-art LateXML parser.

After the analysis of MEs themselves, one type of very important word/phrases that bonds with the MEs, the declaration, are extracted. The declaration manifests the physical meanings of the mathematical notations for readers to easily switch between the mathematical abstraction and the physical world. The challenges are at two folds: the processing of MWM sentences and the enumeration of declaration patterns. The ME is embedded in the sentences, but it could express very complex concepts such as a subordinate clause. In this work, the customized PoS tagger and noun phrase chunker are built to accommodate the MWM situation. This will provide a more accurate set of declaration candidates. The declarations are written in

limited patterns, and the previous experiment also shows that these pattern features played the most important role. But it is not trivial to enumerate these patterns. To help with the enumeration of the declaration patterns, frequency declaration patterns are identified by TFIDF ranking from the sentences where simple variables first occur. After a few rounds of human intervention, many patterns are identified. Experiments on the public evaluation testbed NTCIR math understanding shows that our customized PoS tagger and NP chunker could significantly improve the declaration extraction performance. Though the mined patterns did not improve the performance, they are expected to give an improvement on a larger test dataset.

At last, the rich analytical products above are consolidated into a QuQn map, which is a qualitative-quantitative mapping of the scientific publications from the knowledge understanding aspect. The QuQn map is expected to give a technical abstraction of the technical material. The sequential contents are decomposed and reconstructed as a graph-based abstraction. The dependencies are reconstructed from the ME object at the semantic level. Redundant information is pruned with a reduction ratio of 1:4.

User study in collaboration with AggieSTEM shows that this QuQn abstraction could help the high school students better understand the relationship among factors and boost their confidence in learning complex systems. The QuQn framework provides a solid foundation for the further large-scale user study at the post-graduate level for paper reading. In another knowledge evolution analysis case study, topic analysis based on the declaration is shown to capture the methodologies behind different application topics. The similarity metric derived from LSA could also play as a strong indicator for the citation strength.

### VIII.2 Lessons learned and implication for future work

The summary of the research findings could be consolidated into three aspects concerning the implication for future work: data, model interpretability, and user requirement.

The annotated dataset is crucial for the exploration of properties and the validation of models. However, the higher level the task is, the rarer the public datasets are. In this work, the datasets cover hundreds of PDF files for ME Extraction and ME Layout analysis. But there is no dataset for the ME semantics analysis and few datasets available for the declaration extraction. On the other sides, lots of unlabeled datasets are available, and the weakly-supervised techniques are applied twice in this work for ME Extraction and the declaration patterns collection. Due to the limited ways of presentations to express in technical materials, the weakly-supervised mechanism and unsupervised techniques might be useful to model and predict other types of bonding between the ME and words, e.g., attribution, derivation, constraints. Besides the availability, the lack of standard annotation schema and evaluation criteria also hinder the comparison between different systems. For the ME layout evaluation, the character pair, ME exact match, and MathML edit distance criteria are adopted. There are more works using the edit distance of the LaTeX representation or even image pixel level matching. At the ME semantic level, the OpenMath is supported by the community for a long time, but the officially supported names space is still limited to less than 30. The effort could either be placed on the unification of standards or provide tools to transform between different formats easily, and the latter one is adopted in this work.

The model interpretability is an important factor in the development of the advance model and gives a confidence level for the prediction. This is especially important when errors are inevitable due to the noise or the out of dictionary situations. Research is a creative process,

new notations, layout, and special meaning are defined constantly. They are not covered during the model construction process and could easily lead to a failure of the existing systems. However, if the model is a generative model and the statistical properties are well understood. Additional post-check could be conducted such as the post-checking for ME layout analysis. It will pinpoint errors and save the effort for manual validation.

The last aspect is the user study. The technical products are designed to better serve human in organizing the knowledge. The usefulness of the product could only be validated through a user study. As a point in [7], though there have been ten years of research on mathematical information retrieval (searching), formulae search is not "perceived as useful yet?" To understand the user needs and better improve the system, direct observation is required. Our reading assistant system provides a platform targeting at the knowledge exploration and consumption process. It will play an important role in gather user input in the future.

REFERENCES


[1] S. B. Roy, M. D. Cock, V. Mandava, S. Savanna, B. Dalessandro, P. Claudia, W. Cukierski and B. Hamner, "The microsoft academic search dataset and kdd cup 2013," in *Proceedings of the 2013 KDD cup 2013 workshop*, 2013.

[2] A. Naumowicz and A. Korniłowicz, "A Brief Overview of Mizar," in *TPHOLs '09 Proceedings of the 22nd International Conference on Theorem Proving in Higher Order Logics*, 2009.

[3] Y. Bertot and P. Casteran, Interactive theorem proving and program development: Coq'Art: the calculus of inductive constructions, Springer Science & Business Media, 2013.

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

[5] A. Aizawa, M. Kohlhase and I. Ounis, "NTCIR-11 Math-2 Task Overview," in *NTCIR*, Japan, 2014.

[6] M. Adeel, H. S. Cheung and S. H. Khiyal, "Math GO! prototype of a content based mathematical formula search engine," *Journal of Theoretical and Applied Information Technology ,* vol. 4, no. 10, pp. 1002-1012, 2008.

[7] F. Guidi and C. S. Coen, "A Survey on Retrieval of Mathematical Knowledge," *Mathematics in Computer Science,* vol. 4, no. 10, pp. 409-427, 2016.

[8] G. Y. Kristianto, G. Topic and A. Aizawa, "Utilizing dependency relationships between math expressions in math IR," *Information Retrieval Journal,* vol. 20, pp. 132-167, 2017.

[9] D. Carlisle, P. Ion and R. Miner, "Mathematical Markup Language (MathML) Version 3.0 2nd Edition," [Online]. Available: https://www.w3.org/TR/MathML3/. [Accessed 1 1 2018].

[10] "Arxiv e-print archive," [Online]. Available: https://arxiv.org/. [Accessed 26 8 2018].

[11] P. K. Choubey and R. Huang, "Improving Event Coreference Resolution by Modeling Correlations between Event Coreference Chains and Document Topic Structures," in *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics*, 2018.

[12] D. Shotton, "Semantic publishing: the coming revolution in scientific journal publishing," *Learned Publishing,* vol. 22, no. 2, pp. 85-94, 2009.

[13] "Information technology -- Document description and processing languages -- Office Open XML File Formats -- Part 1: Fundamentals and Markup Language Reference," [Online]. Available: https://www.iso.org/standard/71691.html. [Accessed 26 8 2018].

[14] A. S. Incorporated, PDF Reference, 2006.

[15] S. Mao, A. Rosenfeld and T. Kanungo, "Document structure analysis algorithms: a literature survey," in *Document Recognition and Retrieval X*, Santa Clara, CA, United States, 2003.

[16] R. Cattoni, "Geometric layout analysis techniques for document image understanding: a review," ITC-irst Technical Report 9703.09, 1998.

[17] M. Okamoto and B. Miao, "Recognition of mathematical expressions by using the layout structures of symbols," in *the First International Conference on Document Analysis and Recognition*, 1991.

[18] "PDFMiner," [Online]. Available: https://github.com/euske/pdfminer. [Accessed 1 3 2017].

[19] "PDFBox: A Java PDF Library," [Online]. Available: https://pdfbox.apache.org/. [Accessed 1 12 2017].

[20] "Digital documents research and development.," [Online]. Available: http://multivalent.sourceforge.net/. [Accessed 26 8 2018].

[21] J. Baker, A. Sexton and V. Sorge, "A linear grammar approach to mathematical formula recognition from PDF," in *International Conference on Intelligent Computer Mathematics*, 2009.

[22] "DrawPrintTextLocations Util of PDFBox," [Online]. Available: https://pdfbox.apache.org/docs/2.0.5/javadocs/org/apache/pdfbox/examples/util/DrawPrintTextLocations.html. [Accessed 26 8 2018].

[23] X. Lin, L. Gao, Z. Tang, J. Baker, M. Alkalai and V. Sorge, "A text line detection method for mathematical formula recognition," in *2013 12th International Conference on Document Analysis and Recognition*, 2013.

[24] 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.

[25] J. Lafferty, A. McCallum and F. Pereira, "Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data," in *Proceedings of the 18th International Conference on Machine Learning*, 2001.

[26] I. Councill, L. Giles and M.-Y. Kan, "ParsCit: an Open-source CRF Reference String Parsing Package," in *LREC*, 2008.

[27] H. Li, I. Councill, W.-C. Lee and C. L. Giles, "CiteSeerx: an architecture and web service design for an academic document search engine," in *Proceedings of the 15th international conference on World Wide Web*, 2006.

[28] C. A. Clark and S. K. Divvala, "Looking Beyond Text: Extracting Figures, Tables and Captions from Computer Science Papers," in *AAAI Workshop: Scholarly Big Data*, 2015.

[29] S. R. Choudhury, "Figure metadata extraction from digital documents," in *2013 12th International Conference on Document Analysis and Recognition*, 2013.

[30] J. Baker, A. Sexton and V. Sorger, "MaxTract: Converting PDF to LaTeX, MathML and Text," in *International Conference on Intelligent Computer Mathematics*, 2012.

[31] J. Gehrke, P. Ginsparg and J. Kleinberg, "Overview of the 2003 KDD Cup," *ACM SIGKDD Explorations Newsletter,* vol. 5, no. 2, pp. 149-151, 2003.

[32] B. Miller, "LaTeXML: A Latex to XML converter," [Online]. Available: https://dlmf.nist.gov/LaTeXML/. [Accessed 8 5 2018].

[33] "Pandoc," [Online]. Available: https://pandoc.org/. [Accessed 17 1 2018].

[34] X. Lin, L. Gao, Z. Tang, X. Lin and X. Hu, "Mathematical formula identification in PDF documents," in *2011 International Conference on Document Analysis and Recognition* , 2011.

[35] 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*, 2012, 2012.

[36] J. Jin, X. Han and Q. Wang, "Mathematical Formulas Extraction," in *Proceedings of the Seventh International Conference on Document Analysis and Recognition*, 2003.

[37] X. Lin, L. Gao, Z. Tang, X. Lin and X. Hu, "Performance evaluation of mathematical formula identification.," in *10th IAPR International Workshop on Document Analysis Systems*, 2012.

[38] A. Kacem, A. Belaid and M. Ahmed, "Automatic extraction of printed mathematical formulas using fuzzy logic and propagation of context," *International Journal on Document Analysis and Recognition,* vol. 4, no. 2, pp. 97-108, 2001.

[39] B. Yu, X. Tian and W. Luo, "Extracting Mathematical Components Directly from PDF Documents for Mathematical Expression Recognition and Retrieval," in *International Conference in Swarm Intelligence*, 2014.

[40] K. Iwatsuki, T. Sagara, T. Hara and A. Aizawa, "Detecting In-line Mathematical Expressions in Scientific Documents," in *Proceedings of the 2017 ACM Symposium on Document Engineering*, 2017.

[41] K.-F. Chan and D.-Y. Yeung, "Mathematical expression recognition: a survey," *International Journal on Document Analysis and Recognition,* vol. 3, no. 1, pp. 3-15, 2000.

[42] R. Zanibbi and D. Blostein, "Recognition and retrieval of mathematical expressions," *International Journal on Document Analysis and Recognition,* vol. 15, no. 4, pp. 331-357, 2012.

[43] "Processing mathematical notation," in *Handbook of Document Image Processing and Recognition*, Springer London, 2014, pp. 679-702.

[44] H.-J. Lee and J.-S. Wang, "Design of a mathematical expression understanding system," *Pattern Recognition Letters,* vol. 18, pp. 289-298, 1997.

[45] X. Zhang, L. Gao, K. Yuan, R. Liu, Z. Jiang and Z. Tang, "A Symbol Dominance Based Formulae Recognition Approach for PDF Documents," in *2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR)*, 2017.

[46] R. Zanibbi, D. Blostein and J. Cordy, "Recognizing mathematical expressions using tree transformation," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. 24, no. 11, pp. 1455-1467, 2002.

[47] M. Suzuki, F. Tamari, R. Fukuda, S. Uchida and T. Kanahori, "INFTY: an integrated OCR system for mathematical documents," in *Proceedings of the 2003 ACM symposium on Document engineering*, 2003.

[48] W. Aly, S. Uchida, A. Fujiyoshi and M. Suzuki, "Statistical classification of spatial relationships among mathematical symbols," in *10th International Conference on Document Analysis and Recognition*, 2009.

[49] M. Okamoto, H. Imai and K. Takagi, "Performance evaluation of a robust method for mathematical expression recognition," in *Proceedings. Sixth International Conference on Document Analysis and Recognition*, 2001.

[50] O. Ling, A symbol layout classification for mathematical formula using layout context, Rochester Institute of Technology, 2009.

[51] F. Alvaro and R. Zanibbi, "A shape-based layout descriptor for classifying spatial relationships in handwritten math," in *Proceedings of the 2013 ACM symposium on Document engineering*, 2013.

[52] F. Simistira, V. Katsouros and G. Carayannis, "Recognition of online handwritten mathematical formulas using probabilistic SVMs and stochastic context free grammars," *Pattern Recognition Letters,* vol. 53, pp. 85-92, 2015.

[53] Z. F. C. Wang, "Structural analysis of handwritten mathematical expressions," in *Proceedings of 9th International Conference on Pattern Recognition*, 1988.

[54] A. Raja, M. Rayner, A. Sexton and V. Sorge, "Towards a parser for mathematical formula recognition," *mathematical knowledge management,* vol. 6, no. 26, pp. 139-151, 2006.

[55] "OpenMath Home," [Online]. Available: https://www.openmath.org/. [Accessed 22 6 2018].

[56] C. So and S. Watt, "On the conversion between content MathML and OpenMath," in *Proc. of the Conference on the Communicating Mathematics in the Digital Era*, 2006.

[57] R. Fateman and T. Tokuyasu, "Progress in recognizing typeset mathematics," in *Document Recognition III*, 1996.

[58] E. Miller and P. Viola, "Ambiguity and Constraint in Mathematical Expression Recognition," in *Proceedings of the Fifteenth National/Tenth Conference on Artificial Intelligence/Innovative Applications of Artificial Intelligence*, Madison, Wisconsin, USA, 1998.

[59] S. Lavirotte and L. Pottier, "Optical formula recognition," in *Proceedings of the Fourth International Conference on Document Analysis and Recognition*, 1997.

[60] R. Anderson, "Syntax-directed recognition of two-dimensional mathematics," Ph. D. Thesis, Div. Engg. and App. Phy., Harvard Univ., 1965.

[61] A. Grbavec and D. Blostein, "Mathematics recognition using graph rewriting," in *Proceedings of the Third International Conference on Document Analysis and Recognition*, 1995.

[62] M. Wolska and M. Grigore, "Symbol Declarations in Mathematical Writing," in *Proceeding of the 3rd workshop on Towards a Digital Mathematics Library*, Paris, France, 2010.

[63] P. Chou, "Recognition of equations using a two-dimensional stochastic context-free grammar," *Visual Communications and Image Processing IV,* vol. 1199, pp. 852-866, 1199.

[64] F. Alvaro and J.-M. Benedi, "Recognition of printed mathematical expressions using two-dimensional stochastic context-free grammars," in *Proceeding of 2011 International Conference on Document Analysis and Recognition*, 2011.

[65] A. Youssef, "Part-of-math tagging and applications," in *International Conference on Intelligent Computer Mathematics*, 2017.

[66] "Elsevier Open Access Corpus," [Online]. Available: https://github.com/elsevierlabs/OA-STM-Corpus. [Accessed 1 5 2017].

[67] G. Y. Kristianto, M.-Q. Nghiem, Y. Matsubayashi and A. Aizawa, "Extracting Definitions of Mathematical Expressions in Scientific Papers," in *Proc. of the 26th Annual Conference of the Japanese Society for Artificial Intelligence*, 2012.

[68] R. Pagel and M. Schubotz, "Mathematical Language Processing Project," in *CICM Workshops*, 2014.

[69] U. Schöneberg and W. Sperber, "POS Tagging and Its Applications for Mathematics," in *Intelligent Computer Mathematics*, 2014.

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

[71] T. Brants, "TnT: a statistical part-of-speech tagger," in *Proceedings of the sixth conference on Applied natural language processing*, 2000.

[72] X. Wang, J. Lin, R. Vrecenar and J.-C. Liu, "Syntactic role identification of mathematical expressions," in *Proceeding of the Twelfth International Conference on Digital Information Management*, 2017.

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

[74] M. Ganesalingam, "The Language of Mathematics," PhD diss., University of Cambridge, 2010.

[75] K. Fundel, R. Küffner and R. Zimmer, "RelEx - Relation extraction using dependency parse trees," *Bioinformatics,* vol. 23, no. 3, pp. 365-371, 2007.

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

[77] 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.

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

[79] M. Goldberg, "An Unsupervised Model for Statistically Determining Coordinate Phrase Attachment," in *Proceedings of the 37th annual meeting of the Association for Computational Linguistics on Computational Linguistics*, 1999.

[80] 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, no. 1, pp. 95-130, 1999.

[81] G. Y. Kristianto, "MCAT Math Retrieval System for NTCIR-12 MathIR Task," in *NTCIR*, 2016.

[82] R. M. Oliveira, F. B. Gonzaga, V. C. Barbosa and G. B. Xexéo, "A distributed system for SearchOnMath based on the Microsoft BizSpark program," *arXiv preprint arXiv:1711.04189,* 2017.

[83] C.-C. Prodescu, "MathWebSearch," [Online]. Available: https://github.com/KWARC/mws. [Accessed 8 5 2018].

[84] S. Kamali and F. Wm Tompa, "Retrieving documents with mathematical content," in *Proceedings of the 36th international ACM SIGIR conference on Research and development in information retrieval*, 2013.

[85] R. Munavalli and R. Miner, "Mathfind: a math-aware search engine," in *Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*, 2006.

[86] A. Youssef, "Methods of Relevance Ranking and Hit-content Generation in Math Search," in *Towards Mechanized Mathematical Assistants*, 2007.

[87] R. Zanibbi and B. Yuan, "Keyword and image-based retrieval of mathematical expressions," in *Proceedings of SPIE, the International Society for Optical Engineering*, 2011.

[88] M. Kohlhase and I. Sucan, "A search engine for mathematical formulae," in *8th International Conference on Artificial Intelligence and Symbolic Computation*, 2006.

[89] T. Nguyen, K. Chang and S. Cheung Hui, "A math-aware search engine for math question answering system," in *Proceedings of the 21st ACM international conference on Information and knowledge management*, 2012.

[90] K. Yokoi and A. Aizawa, "An Approach to Similarity Search for Mathematical Expressions using MathML," in *Towards a Digital Mathematics Library*, 2009.

[91] P. Graf, "Substitution tree indexing," in *International Conference on Rewriting Techniques and Applications*, 1995.

[92] H. Schütze, C. Manning and P. Raghavan, Introduction to information retrieval, Cambridge University Press, 2008.

[93] "Mathematica," [Online]. Available: http://www.wolfram.com/. [Accessed 27 8 2018].

[94] "MML Query system," [Online]. Available: http://mmlquery.mizar.org/. [Accessed 12 8 2018].

[95] K. Nakagawa, A. Nomura and M. Suzuki, "Extraction of logical structure from articles in mathematics," in *International Conference on Mathematical Knowledge Management*, 2004.

[96] 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*, 2011.

[97] T. K. Attwood, D. B. Kell, P. McDermott, J. Marsh, S. Pettifer and D. Thorne, "Utopia documents: linking scholarly literature with research data," *Bioinformatics,* vol. 26, no. 18, pp. 568-574, 2010.

[98] E. Weng and A. Owens, "Lecture 11: The Good-Turing Estimate," 2010.

[99] S. Oh, S. Russell and S. Sastry, "Markov chain Monte Carlo data association for multi-target tracking," *IEEE Transactions on Automatic Control,* vol. 54, no. 3, pp. 481-497, 2009.

[100] M. Long, H. Zhu, J. Wang and M. I. Jordan, "Unsupervised domain adaptation with residual transfer networks," in *Advances in Neural Information Processing Systems*, 2016.

[101] S. Bird and E. Loper, "NLTK: the natural language toolkit," in *Proceedings of the ACL 2004 on Interactive poster and demonstration sessions*, 2004.

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

[103] I. Markovsky, J. Willems, S. V. Huffel, B. D. Moor and R. Pintelon, "Application of structured total least squares for system identification and model reduction," *IEEE Transactions on Automatic Control,* vol. 50, no. 10, pp. 1490-1500, 2005.

[104] X. Wang, "Missing MEs in Marmot dataset," [Online]. Available: http://rtds.cse.tamu.edu/resources/. [Accessed 1 7 2017].

[105] N. Okazaki, "CRFsuite: a fast implementation of Conditional Random Fields (CRFs)," 2007. [Online]. Available: http://www.chokkan.org/software/crfsuite/. [Accessed 1 1 2018].

[106] P. R. Cohen, "DARPA's Big Mechanism program," *Physical biology,* vol. 12, no. 4, 2015.

[107] K. Marriott, B. Meyer and K. B. Wittenburg, "A Survey of Visual Language Specification and Recognition," in *Visual Language Theory*, New York, NY, Springer New York, 1998, pp. 5-85.

[108] X. Wang and J.-C. Liu, "A content-constrained spatial (CCS) model for layout analysis of mathematical expressions," in *Twelfth International Conference on Digital Information Management*, Fukuoka, Japan, 2017.

[109] "Subscript and Superscript," [Online]. Available: https://en.wikipedia.org/wiki/Subscript_and_superscript. [Accessed 17 1 2018].

[110] K. Sain, A. Dasgupta and U. Garain, "EMERS: a tree matching--based performance evaluation of mathematical expression recognition systems," *International Journal on Document Analysis and Recognition ,* vol. 14, no. 1, pp. 75-85, 2011.

[111] Y. Deng, A. Kanervisto, J. Ling and A. M. Rush, "Image-to-Markup Generation with Coarse-to-Fine Attention," in *International Conference on Machine Learning*, 2017.

[112] H. M. Twaakyondo and M. Okamoto, "Structure analysis and recognition of mathematical expressions," in *Proceedings of 3rd International Conference on Document Analysis and Recognition*, 1995.

[113] G. Bouma, "Normalized (pointwise) mutual information in collocation extraction," in *Proceedings of GSCL*, 2009.

[114] J. Cocke, "Programming languages and their compilers: Preliminary notes," Courant Institute Math. Sci., 1970.

[115] M. Kohlhase, "Using LaTeX as a semantic markup format," *Mathematics in Computer Science,* vol. 2, no. 2, pp. 279-304, 2008.

[116] C. Lange and M. Kohlhase, "SWiM: A semantic wiki for mathematical knowledge management," in *Emerging Technologies for Semantic Work Environments: Techniques, Methods, and Applications*, 2008.

[117] K. Zhang and D. Shasha, "Simple fast algorithms for the editing distance between trees and related problems," *SIAM Journal on Computing,* vol. 18, no. 6, pp. 1245-1262, 1989.

[118] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss and V. Dubourg, "Scikit-learn: Machine learning in Python," *Journal of machine learning research,* vol. 12, pp. 2825-2830, 2011.

[119] 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*, 2014.

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

[121] E. Tufte and P. Graves-Morris, The visual display of quantitative information, 1983.

[122] C. E. Moody, "Mixing dirichlet topic models and word embeddings to make lda2vec," arXiv preprint arXiv:1605.02019, 2016.

[123] L. H. Hill, "Concept mapping to encourage meaningful student learning," *Adult Learning,* vol. 16, pp. 7-13, 2005.

[124] I. Lyublinska, G. Wolfe, D. Ingram, L. Pujji, S. Oberoi and N. Czuba, College Physics for AP Courses, OpenStax, 2015.

[125] M. W. Lipsey and D. B. Wilson, Applied social research methods series; Vol. 49. Practical meta-analysis, Thousand Oaks, CA, US: Sage Publications, Inc, 2001.

[126] C. Chen, J. Zhang and M. Vogeley, "Making sense of the evolution of a scientific domain: a visual analytic study of the Sloan Digital Sky Survey research," *Scientometrics,* vol. 83, no. 3, pp. 669-688, 2009.

[127] J. Tang, R. Jin and J. Zhang, "A Topic Modeling Approach and Its Integration into the Random Walk Framework for Academic Search," in *Proceeding of the Eighth IEEE International Conference on Data Mining*, 2008.

[128] 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*, 2006.

[129] T. K. Landauer, P. W. Foltz and D. Laham, "An introduction to latent semantic analysis," *Discourse processes,* vol. 25, pp. 259-284, 1998.

[130] J. Shi and J. Malik, "Normalized cuts and image segmentation," *IEEE Transactions on pattern analysis and machine intelligence,* vol. 22, no. 8, pp. 888-905, 2000.

[131] J. L. Boyd-Graber and D. M. Blei, "Syntactic topic models," in *Advances in neural information processing systems*, 2009.

[132] D. Lin, E. Grimson and J. W. Fisher, "Construction of dependent Dirichlet processes based on Poisson processes," in *Advances in neural information processing systems*, 2010.

[133] D. Newman, P. Smyth, M. Welling and A. U. Asuncion, "Distributed inference for latent dirichlet allocation," in *Advances in neural information processing systems*, 2008.

[134] M. Hoffman, F. R. Bach and D. M. Blei, "Online learning for latent dirichlet allocation," in *advances in neural information processing systems*, 2010.

[135] P. K. Gopalan, S. Gerrish, M. Freedman, D. M. Blei and D. M. Mimno, "Scalable inference of overlapping communities," in *Advances in Neural Information Processing Systems*, 2012.

[136] K. Swersky, J. Snoek and R. P. Adams, "Multi-task bayesian optimization," in *Advances in neural information processing systems*, 2013.

[137] A. Bissacco, M.-H. Yang and S. Soatto, "Detecting humans via their pose," in *Advances in Neural Information Processing Systems*, 2007.

[138] G. Rogez and C. Schmid, "Mocap-guided data augmentation for 3d pose estimation in the wild," in *Advances in Neural Information Processing Systems*, 2016.

[139] H. He, J. Eisner and H. Daume, "Imitation learning by coaching," in *Advances in Neural Information Processing Systems*, 2012.

[140] S. Shalev-Shwartz and S. M. Kakade, "Mind the duality gap: Logarithmic regret algorithms for online optimization," in *Advances in Neural Information Processing Systems*, 2009.

[141] S.-K. Kim, J.-H. Jeon, H.-K. Choi and J.-B. Ahn, "Design of frequency shift acceleration control for anti-islanding of an inverter-based DG," in *13th International Power Electronics and Motion Control Conference*, Poznan, 2008.

[142] S.-K. Kim, J.-H. Jeon, J.-B. Ahn, B. Lee and S.-H. Kwon, "Frequency-shift acceleration control for anti-islanding of a distributed-generation inverter," *IEEE Transactions on Industrial Electronics,* vol. 57, no. 2, pp. 494-504, 2010.

[143] H. White, "Pathfinder networks and author cocitation analysis: A remapping of paradigmatic information scientists," *Journal of the American Society for Information Science and Technology,* pp. 423-434, 2003.

[144] "Comparing approaches to mathematical document analysis from PDF," in *2011 International Conference on Document Analysis and Recognition*, 2011.

[145] S. Greengard, "Are We Losing our ability to think critically?," *Communications of the ACM,* vol. 52, no. 7, pp. 18-19, 2009.

[146] J. Hirsch, "An index to quantify an individual's scientific research output," in *Proceedings of the National academy of Sciences of the United States of America*, 2005.

[147] L. Page, S. Brin, R. Motwani and T. Winograd, "The PageRank Citation Ranking: Bringing Order to the Web," Stanford InfoLab, 1999.

[148] M. Singh, V. Patidar, S. Kumar, T. Chakraborty, A. Mukherjee and P. Goyal, "The Role Of Citation Context In Predicting Long-Term Citation Profiles: An Experimental Study Based On A Massive Bibliographic Text Dataset," in *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management2015*.

[149] F. Janssens, W. Glanzel and B. D. Moor, "Towards mapping library and information science," *Information Processing \& Management,* vol. 42, no. 6, pp. 1614-1642, 2006.

[150] R. Kostoff, A. Rio, J. Humenik, E. Garcia and A. Ramirez, "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.

[151] P. Glenisson, W. Glanzel and O. Persson, "Combining full-text analysis and bibliometric indicators. A pilot study," *Scientometrics,* vol. 63, no. 1, pp. 163-180, 2005.

[152] D. Blei, A. Ng and M. Jordan, "Latent dirichlet allocation," *Journal of machine Learning research,* vol. 3, pp. 993-1022, 2003.

[153] B. Cronin, The citation process: The role and significance of citations in scientific communication, London: Taylor Graham , 1984.

[154] 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*, 2006.

APPENDIX A PCFG PRODUCTOIN RULES FOR ME SEMANTICS PARSING

This appendix chapter for the ME semantics parsing will give a complete token list and the grammar for the parsing of ME semantics.

## A.1 A complete table of the terminal tokens

| Symbol Groups | Greek | greek characters |
|---|---|---|
| | Accent | indicator for Accent |
| | A-Za-Z0-9 | alphabets and digits |
| Operations | PM | corresponding to $\pm$, could mean two number or mean/std |
| | CIRC | TODO |
| | CDOT | vector product |
| | TIMES | corresponding to $\times$ |
| | OTIMES | corresponding to $\otimes$, for vector outer product |
| | DIV | corresponding to / |
| | FRAC | corresponding to frac in Latex, expect to have following numerator and denominator |
| | MINUS | corresponding to -, could mean negation or minus |
| | PLUS | corresponding to + |
| | OPLUS | corresponding to $\oplus$ |
| Big operator | SUM | summation |
| | PROD | production |
| | INT | integral |
| | CO_PROD | co-production |
| Named functions | LOG | log |
| | MIN | min |
| | FUNC_NAME | named function matching a predefined table |
| General Relation | EQUIV | two element are equivalent |
| | CONG | two element are essentially the same but not identical |
| | EQUAL | two elements are equal |
| | NOTEQUAL | two elements are not equal |
| | SIMEQUAL | two elements are similar |
| Number relation | LESS | one number is less than the other |
| | GREATER | one number is greater than the other |
| | LEQ | one number is less than or equal to the other |
| | GEQ | one number is greater than or equal to the other |

| | | |
|---|---|---|
| Punctuation | BULLET | Might be used as multiplication or divergent |
| | AST | Could mean matching arbitary or a special value |
| | PRIME | Could mean a special variable or the differentiation |
| | EXCLAM | factorial |
| | CDOTS | Etc. |
| | LDOTS | Etc. |
| | PUNCT_COMMA | separating elements |
| | PUNCT_PERIOD | accessing members |
| | PUNCT_COLON | definition $:=$, function mapping $f: R \rightarrow R$ |
| | PUNCT_SEMICOLON | separating parameters from variables |
| Paired Fences | BRACKET_OPEN | ( |
| | BRACKET_CLOSE | ) |
| | SQ_BRACKET_OPEN | [ |
| | SQ_BRACKET_CLOSE | ] |
| | CURVE_BRACKET_OPEN | \{ |
| | CURVE_BRACKET_CLOSE | \} |
| | ANGLE_BRACKET_OPEN | $\langle$ |
| | ANGLE_BRACKET_CLOSE | $\rangle$ |
| | FLOOR_OPEN | $\lfloor$ |
| | FLOOR_CLOSE | $\rfloor$ |
| | CEIL_OPEN | $\lceil$ |
| | CEIL_CLOSE | $\rceil$ |
| | VERT_BAR | \| for absolute or cardality |
| | DOUBLE_VERT_BAR | \|\| for norm |
| Spatial Tokens | GROUP_OPEN | special tag to mark the beginning of a semantic unit |
| | GROUP_CLOSE | special tag to mark the end of a semantic unit |
| | SUP_OPEN | start of an superscript |
| | SUP_CLOSE | end of an superscript |
| | SUB_OPEN | start of an subscript |
| | SUB_CLOSE | end of an subscript |
| | OVER_OPEN | start of an over structure |
| | OVER_CLOSE | end of an over structure |
| | UNDER_OPEN | start of an under structure |
| | UNDER_CLOSE | end of an under structure |
| | ACCENT_OPEN | start of an accent structure |
| | ACCENT_CLOSE | end of an accent structure |
| | SQRT_OPEN | start of the enclosed part for a radical structure |
| | SQRT_CLOSE | end of the enclosed part for a radical structure |

| | | |
|---|---|---|
| Set Theory | IN | element in set |
| | NOTIN | element not in set |
| | NSUBSETEQ | set 1 not as the subset of or equal to set 2 |
| | SUBSETEQ | set 1 as the subset of or equal to set 2 |
| | SUBSET | set 1 as the subset of set 2 |
| | CUP | merging of two set |
| | SETDIFF | corresponding to \\ |
| | CAP | intersection of two set |
| Logic | FORALL | $\forall$ |
| | WEDGE | and logic |
| | EXISTS | $\exists$ |
| Calculus | D_LOW_TEXT | intergal factor |
| | PARTIAL | partial derivation |
| | NABLA | gradient or divergent |
| SemanticsKnown | IDVAR | identifier for variables or functions |
| | MUL | the virtual concept of multiply |
| | FUNC_ARG_OPEN | ( as the open of function arguments |
| | FUNC_ARG_CLOSE | ) as the end of function arguments |
| | CN | constant number |
| Misc. | UNKNOWN | out of dictionary symbol |
| | EMPTY | empty set |
| | INFTY | infinity |
| | WRT | with respect to |
| | ST | such that |
| | SPACE | For an empty space |
| | PUNCT_AND | Not sure of the usage |

## A.2 A complete list of the production rules

### *A.2.1 Digits*

FACTOR -> NUM_FACTOR

NUM_FACTOR -> MINUS, NUM_FACTOR

NUM_FACTOR -> INT_NUM_FACTOR

DIGIT -> DIGIT_0 | DIGIT_1| DIGIT_2| DIGIT_3| DIGIT_4| DIGIT_5| DIGIT_6 | DIGIT_7| DIGIT_8 | DIGIT_9

INT_NUM_FACTOR -> DIGIT

INT_NUM_FACTOR -> DIGIT, INT_NUM_FACTOR

NUM_FACTOR -> FLOAT_NUM_FACTOR

FLOAT_NUM_FACTOR -> PUNCT_PERIOD, INT_NUM_FACTOR

FLOAT_NUM_FACTOR -> INT_NUM_FACTOR, PUNCT_PERIOD, INT_NUM_FACTOR

NUM_FACTOR -> SCI_NUM_FACTOR

SCI_NUM_FACTOR -> INT_NUM_FACTOR, E_LOW_TEXT, PLUS, INT_NUM_FACTOR

SCI_NUM_FACTOR -> INT_NUM_FACTOR, E_LOW_TEXT, MINUS, INT_NUM_FACTOR

### *A.2.2 Algebra*

FACTOR -> FRAC, EXP, EXP

FACTOR -> GROUP_OPEN, EXP, GROUP_CLOSE

FACTOR -> SQRT_OPEN, EXP, SQRT_CLOSE

FACTOR -> SQ_BRACKET_OPEN, EXP, SQ_BRACKET_CLOSE

FACTOR -> BRACKET_OPEN, EXP, BRACKET_CLOSE

ARITHM_OP_LEVEL2 -> WEDGE

FACTOR -> IDVAR

FACTOR -> CN

FACTOR -> INFTY

FACTOR -> CN, PERCENT

FACTOR -> BRACKET_OPEN, EXP, BRACKET_CLOSE

EXT_TERM -> SUM_OP_TERM

EXT_TERM -> SUM_OP_TERM, EXT_TERM

SUM_OP_TERM -> ARITHM_OP_LEVEL3, TERM

ARITHM_OP_LEVEL3 -> ARITHM_OP_LEVEL3, SUB_EXP

FACTOR -> BRACKET_OPEN, REL_EXP, BRACKET_CLOSE

FACTOR -> AST

ARITHM_OP_LEVEL3 -> PLUS | MINUS | PM

ARITHM_OP_LEVEL2 -> MUL | BULLET | AST | CIRC | CDOT | DIV | WEDGE

ARITHM_OP_LEVEL2 -> SPACE

VEC_FENCE_OPEN -> ANGLE_BRACKET_OPEN

VEC_FENCE_OPEN -> SQ_BRACKET_OPEN

VEC_FENCE_CLOSE -> ANGLE_BRACKET_CLOSE

VEC_FENCE_CLOSE -> SQ_BRACKET_CLOSE

FACTOR -> ANGLE_BRACKET_OPEN, EXP, ANGLE_BRACKET_CLOSE

FACTOR -> ANGLE_BRACKET_OPEN, EXP_LIST, ANGLE_BRACKET_CLOSE

FACTOR -> SQ_BRACKET_OPEN, EXP_LIST, SQ_BRACKET_CLOSE

FACTOR -> ANGLE_BRACKET_OPEN, EXP, PUNCT_COMMA, EXP, ANGLE_BRACKET_CLOSE

FACTOR -> BRACKET_OPEN, EXP, PUNCT_SEMICOLON, EXP, BRACKET_CLOSE

FACTOR -> BRACKET_OPEN, EXP_LIST, PUNCT_COMMA, ETC_FACTOR, PUNCT_COMMA, EXP, BRACKET_CLOSE

FACTOR -> IDVAR, VERT_BAR, IDVAR, VERT_BAR, ETC_FACTOR, VERT_BAR, IDVAR

FACTOR -> SQ_BRACKET_OPEN, EXP_LIST, PUNCT_COMMA, ETC_FACTOR, PUNCT_COMMA, EXP, SQ_BRACKET_CLOSE

FACTOR -> SQ_BRACKET_OPEN, EXP, PUNCT_COMMA, ETC_FACTOR, PUNCT_COMMA, EXP, SQ_BRACKET_CLOSE

FACTOR -> SQ_BRACKET_OPEN, EXP, ETC_FACTOR, EXP, SQ_BRACKET_CLOSE

FACTOR -> LESS, EXP, GREATER

FENCE_ABS_OR_CARD_OPEN -> VERT_BAR

FENCE_ABS_OR_CARD_CLOSE -> VERT_BAR

FACTOR -> FENCE_ABS_OR_CARD_OPEN, EXP, FENCE_ABS_OR_CARD_CLOSE

FACTOR -> NORM_FACTOR

NORM_FACTOR -> FENCE_NORM_OPEN, EXP, FENCE_NORM_CLOSE

NORM_FACTOR -> NORM_FACTOR, SUB_OPEN, EXP, SUB_CLOSE

NORM_FACTOR -> NORM_FACTOR, SUB_OPEN, REL_EXP_LIST, SUB_CLOSE

FENCE_NORM_OPEN -> DOUBLE_VERT_BAR

FENCE_NORM_CLOSE -> DOUBLE_VERT_BAR

FENCE_GROUP_OPEN -> BRACKET_OPEN

FENCE_GROUP_CLOSE -> BRACKET_CLOSE

FENCE_GROUP_OPEN -> SQ_BRACKET_OPEN

FENCE_GROUP_CLOSE -> SQ_BRACKET_CLOSE

FACTOR -> FENCE_GROUP_OPEN, EXP, FENCE_GROUP_CLOSE

TERM -> FACTOR

TERM -> FACTOR, EXT_FACTOR

EXT_FACTOR -> MUL_OP_FACTOR

EXT_FACTOR -> MUL_OP_FACTOR, EXT_FACTOR

MUL_OP_FACTOR -> ARITHM_OP_LEVEL2, FACTOR

MUL_OP_FACTOR -> FACTOR

ARITHM_OP_LEVEL2 -> ARITHM_OP_LEVEL2, SUB_EXP

MUL_OP_FACTOR -> SET_OP_LEVEL1, FACTOR

SET_OP_LEVEL1 -> CAP

SET_OP_LEVEL1 -> SETDIFF

SET_OP_LEVEL1 -> CUP

SET_OP_LEVEL1 -> SET_OP_LEVEL1, SUB_EXP

EXP -> BRACKET_OPEN, EXP, BRACKET_CLOSE

EXP -> SQ_BRACKET_OPEN, EXP, SQ_BRACKET_CLOSE

EXP -> GROUP_OPEN, EXP, GROUP_CLOSE

EXP -> TERM

EXP -> TERM, EXT_TERM

EXP -> TERM, UNIT

EXP_LIST -> EXP, EXT_EXP_LIST

EXT_EXP_LIST -> PUNCT_COMMA_EXP

EXT_EXP_LIST -> PUNCT_COMMA_EXP, EXT_EXP_LIST

PUNCT_COMMA_EXP -> PUNCT_COMMA, EXP

### *A.2.3 Binding operators*

FACTOR -> BIG_OP_FACTOR

BIG_OP_FACTOR -> BIG_OP, EXP

BIG_OP_FACTOR -> BIG_OP, UNDER_EXP, EXP

BIG_OP_FACTOR -> BIG_OP, UNDER_EXP, OVER_EXP, EXP

BIG_OP_FACTOR -> BIG_OP, SUB_EXP, EXP

BIG_OP_FACTOR -> BIG_OP, BIG_OP_SUB_EXP_1, EXP

BIG_OP_FACTOR -> BIG_OP, SUB_EXP, SUP_EXP, EXP

BIG_OP_SUB_EXP_1 -> SUB_OPEN, REL_EXP_LIST, EQUAL, CN, SUB_CLOSE

BIG_OP -> SUM | MIN | INT | PROD | CUP | OPLUS | OTIMES | WEDGE

### *A.2.4 Relation*

REL_EXP -> EXP, REL_OP_EXP

REL_EXP -> EXP_LIST, REL_OP_EXP

REL_EXP -> REL_EXP, REL_OP_EXP

REL_OP_EXP -> REL_OP, EXP

REL_OP_EXP -> REL_OP, EXP_LIST

REL_EXP -> EXISTS, REL_EXP

REL_EXP -> FORALL, REL_EXP

REL_EXP -> REL_EXP, FORALL, FACTOR

REL_EXP -> REL_EXP, FORALL, EXP_LIST

REL_EXP -> REL_EXP, FORALL, REL_EXP

REL_OP -> EQUAL | SIMEQUAL | NOTEQUAL | EQUIV | CONG | LEQ | GEQ | GREATER | LESS

REL_OP -> LEFTARROW | RIGHTARROW

REL_OP -> IN | NOTIN | NSUBSETEQ | SUBSETEQ | SUBSET

REL_EXP_LIST -> REL_EXP, EXT_REL_EXP_LIST

EXT_REL_EXP_LIST -> PUNCT_COMMA_REL_EXP

EXT_REL_EXP_LIST -> PUNCT_COMMA_REL_EXP, EXT_REL_EXP_LIST

PUNCT_COMMA_REL_EXP -> PUNCT_COMMA, REL_EXP

### *A.2.5 Spatial layout*

FACTOR -> SUB_FACTOR

SUB_FACTOR -> FACTOR, SUB_EXP

FACTOR -> SUP_FACTOR

SUP_FACTOR -> FACTOR, SUP_EXP

SUP_FACTOR -> EXP, SUP_OPEN, EXP, SUP_CLOSE

UNDER_EXP -> UNDER_OPEN, ME, UNDER_CLOSE

OVER_EXP -> OVER_OPEN, ME, OVER_CLOSE

UNDER_EXP -> UNDER_OPEN, ALPHABET_SEQ, UNDER_CLOSE

OVER_EXP -> OVER_OPEN, ALPHABET_SEQ, OVER_CLOSE

SUB_EXP -> SUB_OPEN, ME, SUB_CLOSE

SUB_EXP -> SUB_OPEN, DEC_SYMBOL, SUB_CLOSE

SUB_EXP -> SUB_OPEN, GROUP_OPEN, DEC_SYMBOL, GROUP_CLOSE, SUB_CLOSE

SUP_EXP -> PRIME

SUP_EXP -> PRIME, PRIME

SUP_EXP -> SUP_OPEN, ME, SUP_CLOSE

SUP_EXP -> SUP_OPEN, DEC_SYMBOL, SUP_CLOSE

SUP_EXP -> SUP_OPEN, GROUP_OPEN, DEC_SYMBOL, GROUP_CLOSE, SUP_CLOSE

DEC_SYMBOL -> DOWN_ARROW | UP_ARROW | AST | PRIME | T_CAP_TEXT | DAGGER

DEC_SYMBOL -> PRIME, PRIME

FACTOR -> ACCENT_SYM, ACCENT_OPEN, EXP, ACCENT_CLOSE

### *A.2.6 Calculus*

FACTOR -> DELTA_CAP, IDVAR

FACTOR -> DELTA_CAP, EXP

FACTOR -> DELTA_CAP, BRACKET_OPEN, EXP, BRACKET_CLOSE

FACTOR -> PARTIAL, DIV, PARTIAL, IDVAR

FACTOR -> NABLA, EXP

FACTOR -> NABLA, CDOT, EXP

FACTOR -> NABLA, BULLET, EXP

FACTOR -> NABLA, TIMES, EXP

### *A.2.7 Functions*

ME -> DENOTATION

ME -> FUNC_DELC

DENOTATION -> EXP, PUNCT_COLON, ME

FUNC_PARAM_HOLDER -> CDOT

FUNC_DELC -> EXP_LIST, PUNCT_COLON, EXP, MAPSTO, EXP

FUNC_DELC -> EXP_LIST, PUNCT_COLON, EXP, RIGHTARROW, EXP

FUNC_DELC -> EXP, PUNCT_COLON, EXP, MAPSTO, EXP

FUNC_DELC -> EXP, PUNCT_COLON, EXP, RIGHTARROW, EXP

FUNC_DELC -> EXP, PUNCT_COLON, EQUAL, ME

FACTOR -> IDVAR, MAPSTO, IDVAR

FACTOR -> IDVAR, MAPSTO, EXP

FUNC -> NAMED_FUNC

FUNC -> USER_FUNC

FUNC -> FUNC_NAME

USER_FUNC -> ALPHABET_SEQ

USER_FUNC -> VARSYM

USER_FUNC -> VARSYM, SUB_EXP

NAMED_FUNC -> LOG | MIN

SINGLE_OP -> LOG | O_CAP_TEXT | MIN | MINUS | PLUS | PM | PARTIAL | WEDGE

SINGLE_OP -> FUNC_NAME

SINGLE_OP -> L_LOW_TEXT, N_LOW_TEXT

FACTOR -> SINGLE_OP, TERM

FACTOR -> SINGLE_OP, FENCE_ARG_OPEN, EXP, FENCE_ARG_CLOSE

FACTOR -> FUNC, FENCE_ARG_OPEN, EXP, FENCE_ARG_CLOSE

FACTOR -> FUNC, FENCE_ARG_OPEN, REL_EXP, FENCE_ARG_CLOSE

FACTOR -> FUNC, FENCE_ARG_OPEN, EXP_LIST, FENCE_ARG_CLOSE

FENCE_ARG_OPEN -> BRACKET_OPEN

FENCE_ARG_CLOSE -> BRACKET_CLOSE

FENCE_ARG_OPEN -> FUNC_ARG_OPEN

FENCE_ARG_CLOSE -> FUNC_ARG_CLOSE

### *A.2.8 Probability*

PR -> P_CAP_TEXT

PR_OPEN_FENCE -> BRACKET_OPEN

PR_OPEN_FENCE -> SQ_BRACKET_OPEN

PR_OPEN_FENCE -> FUNC_ARG_OPEN

PR_OPEN_FENCE -> FENCE_ARG_OPEN

PR_CLOSE_FENCE -> BRACKET_CLOSE

PR_CLOSE_FENCE -> SQ_BRACKET_CLOSE

PR_CLOSE_FENCE -> FUNC_ARG_CLOSE

PR_CLOSE_FENCE -> FENCE_ARG_CLOSE

FACTOR -> PR, PR_OPEN_FENCE, FACTOR, VERT_BAR, REL_EXP, PR_CLOSE_FENCE

FACTOR -> PR, PR_OPEN_FENCE, REL_EXP, PR_CLOSE_FENCE

FACTOR -> PR, PR_OPEN_FENCE, EXP, PR_CLOSE_FENCE

FACTOR -> PR, PR_OPEN_FENCE, EXP, VERT_BAR, EXP_LIST, PR_CLOSE_FENCE

FACTOR -> PR, PR_OPEN_FENCE, EXP, VERT_BAR, EXP, PR_CLOSE_FENCE

### *A.2.9 Set*

FACTOR -> SET_FACTOR

SET_FACTOR -> SET_RANGE_OPEN, EXP, PUNCT_COMMA, EXP, SET_RANGE_CLOSE

SET_RANGE_OPEN -> BRACKET_OPEN

SET_RANGE_OPEN -> SQ_BRACKET_OPEN

SET_RANGE_CLOSE -> BRACKET_CLOSE

SET_RANGE_CLOSE -> SQ_BRACKET_CLOSE

FENCE_SET_OPEN -> CURVE_BRACKET_OPEN

FENCE_SET_CLOSE -> CURVE_BRACKET_CLOSE

SET_FACTOR -> FENCE_SET_OPEN, REL_EXP_LIST, FENCE_SET_CLOSE

SET_FACTOR -> FENCE_SET_OPEN, EXP_LIST, FENCE_SET_CLOSE

SET_FACTOR -> FENCE_SET_OPEN, EXP, FENCE_SET_CLOSE

SET_FACTOR -> FENCE_SET_OPEN, REL_EXP, FENCE_SET_CLOSE

SET_FACTOR -> REALDOMAIN

REALDOMAIN -> R_CAP_TEXT

SET_FACTOR -> EMPTY

SET_FACTOR -> SET_FACTOR, SUP_EXP

SET_FACTOR -> SET_FACTOR, SUB_EXP, SUP_EXP

COND_SET_FENCE -> VERT_BAR

COND_SET_FENCE -> PUNCT_COLON

SET_FACTOR -> FENCE_SET_OPEN, ME, COND_SET_FENCE, REL_EXP, FENCE_SET_CLOSE

SET_FACTOR -> FENCE_SET_OPEN, ME, COND_SET_FENCE, REL_EXP_LIST, FENCE_SET_CLOSE

SET_FACTOR -> FENCE_SET_OPEN, EXP_LIST, PUNCT_COMMA, ETC_FACTOR, PUNCT_COMMA, EXP_LIST, FENCE_SET_CLOSE

SET_FACTOR -> FENCE_SET_OPEN, EXP_LIST, PUNCT_COMMA, ETC_FACTOR, PUNCT_COMMA, EXP, FENCE_SET_CLOSE

SET_FACTOR -> FENCE_SET_OPEN, EXP, PUNCT_COMMA, ETC_FACTOR, PUNCT_COMMA, EXP_LIST, FENCE_SET_CLOSE

SET_FACTOR -> FENCE_SET_OPEN, EXP, PUNCT_COMMA, ETC_FACTOR, PUNCT_COMMA, EXP, FENCE_SET_CLOSE

ETC_FACTOR -> LDOTS

ETC_FACTOR -> CDOTS

ETC_FACTOR -> PUNCT_PERIOD, PUNCT_PERIOD

ETC_FACTOR -> PUNCT_PERIOD, PUNCT_PERIOD, PUNCT_PERIOD

### *A.2.10 Units*

UNIT -> R_LOW_TEXT, A_LOW_TEXT, D_LOW_TEXT

UNIT -> S_LOW_TEXT

UNIT -> K_LOW_TEXT, G_LOW_TEXT

UNIT -> N_CAP_TEXT

UNIT -> M_LOW_TEXT

UNIT -> P_CAP_TEXT, A_LOW_TEXT

UNIT -> BRACKET_OPEN, UNIT, BRACKET_CLOSE

UNIT -> SQ_BRACKET_OPEN, UNIT, SQ_BRACKET_CLOSE

UNIT -> CUNIT

CUNIT -> UNIT

CUNIT -> UNIT, CUNIT

CUNIT -> UNIT, ARITHM_OP_LEVEL2, CUNIT

CUNIT -> UNIT, SUP_OPEN, NUM_FACTOR, SUP_CLOSE

### *A.2.11 Incomplete*

INCOMPLETE_EXP -> ME, VERT_BAR

INCOMPLETE_EXP -> EXP, VERT_BAR

INCOMPLETE_EXP -> ME, BRACKET_CLOSE

INCOMPLETE_EXP -> BRACKET_OPEN, ME

INCOMPLETE_EXP -> ME, PUNCT_COMMA

INCOMPLETE_EXP -> ME, PUNCT_COLON

INCOMPLETE_EXP -> ME, PUNCT_PERIOD

INCOMPLETE_EXP -> ME, PUNCT_SEMICOLON

INCOMPLETE_EXP -> ME, PUNCT_COLON, EQUAL

INCOMPLETE_EXP -> REL_OP, ME

INCOMPLETE_EXP -> ME, REL_OP

INCOMPLETE_EXP -> ME, ARITHM_OP_LEVEL1

INCOMPLETE_EXP -> ME, ARITHM_OP_LEVEL2

INCOMPLETE_EXP -> ME, ARITHM_OP_LEVEL3

INCOMPLETE_EXP -> ARITHM_OP_LEVEL1, ME

INCOMPLETE_EXP -> ARITHM_OP_LEVEL2, ME

INCOMPLETE_EXP -> ARITHM_OP_LEVEL3, ME

INCOMPLETE_EXP -> UNIT

INCOMPLETE_EXP -> REL_OP

INCOMPLETE_EXP -> ARITHM_OP_LEVEL1

INCOMPLETE_EXP -> ARITHM_OP_LEVEL2

INCOMPLETE_EXP -> ARITHM_OP_LEVEL3

INCOMPLETE_EXP -> PUNCT_AND

INCOMPLETE_EXP -> PRIME

INCOMPLETE_EXP -> SPACE

INCOMPLETE_EXP -> ETC_FACTOR

FACTOR -> IDVAR, VERT_BAR, SUB_OPEN, IDVAR, SUB_CLOSE

EXP_LIST -> EXP, LDOTS, EXP

# APPENDIX B. MANUAL FILTERING FOR DECLARATION PATTERNS COLLECTION

This appendix chapter show the supplementary materials for the declaration pattern collection, including the PoS table for pattern filtering, and the history of the manual filtering process.

## B.1 PoS Table

**Table 30 PoS filtering for declaration pattern collection**

| category | tags |
|---|---|
| parenthesis | -LRB-, -RRB- |
| verb | VB, VBG, VBD, VBN, VBP, VBZ |
| preposition | IN, (might include TO) |
| noun | NN, NNS, NNP, NNPS |
| pronoun | PRP, PRP$ |
| adjective | JJ, NML, JJR, JJS |
| adverb | RB, RBR, RBS, WRB |
| ME | NP-ME, S-ME, NML-ME |
| punctuation | "$", ",", ".", ":", "#", "”", |
| wh-* | WDT, WP, WP$ |
| determinate | DT, PDT |
| particle | RP |
| digits/symbol | CD, SYM |
| misc. | MD, UH, -NONE-, FW, LS, EX, CC POS, |
| unknown | STOP |

## B.2 Manual filter process

**Table 31 Manual pattern filtering, round 2**

| | | |
|---|---|---|
| good patterns | ... ME ...refers...to... DEC ... | ...Let... DEC ...denote... ME ... |
| | ...with... ME ...,... DEC ... | ...write... ME ...for... DEC ... |
| | ...Let... DEC ...be... ME ... | ... ME ...is...called... DEC ... |
| | ... DEC ...denoted...by... ME ... | ...refer... ME ...as... DEC ... |
| | ...denoted... ME ...for... DEC ... | ...refer... DEC ...as... ME ... |
| | ... DEC ...denote...with... ME ... | ...use... ME ...as... DEC ... |
| | ...denote... DEC ...with... ME ... | ...by... ME ...represents... DEC ... |
| | ...denotes... DEC ...with... ME ... | ...Let... ME ...denote... DEC ... |
| | ...Let... DEC ...define... ME ... | ... ME ...corresponding...to... DEC ... |
| | ...mapping... ME ...to... DEC ... | ...with... ME ...defined... DEC ... |
| | ... ME ...to...represent... DEC ... | ...Let... ME ...as... DEC ... |
| | ...Denote... DEC ...by... ME ... | |
| ignore patterns | ...be... DEC ...let... ME ... | ...)... DEC ...(... ME ... |
| | ...for... DEC ...let... ME ... | ...divide... DEC ...into... ME ... |
| | ...is... DEC ...than... ME ... | ... ME ...are...respectively... DEC ... |
| | ...with... DEC ...than... ME ... | ...with... ME ...representing... DEC ... |
| | ...into... ME ...,... DEC ... | ...to... ME ...through... DEC ... |
| | ...For... DEC ...use... ME ... | ...we... ME ...to... DEC ... |
| | ...To... DEC ...let... ME ... | ...for... DEC ...given... ME ... |
| | ...use... DEC ...from... ME ... | ...hidden... ME ...at... DEC ... |
| | ...be... DEC ...than... ME ... | ...given... DEC ...with... ME ... |
| | ...to... DEC ...while... ME ... | ... DEC ...parameterized...by... ME ... |
| | ...to... DEC ...than... ME ... | ...is... DEC ...let... ME ... |
| | ...has... ME ...,... DEC ... | ...by... DEC ...denoted... ME ... |
| | ...given... ME ...with... DEC ... | ...replacing... ME ...with... DEC ... |
| | ... DEC ...indexed...by... ME ... | |
| stop words | divide | than |
| | respectively | we |
| | at | hidden |
| | while | parameterized |
| | replacing | indexed |

**Table 32 Manual pattern filtering, round 3**

| | | |
|---|---|---|
| good patterns | ... DEC ...denoted...as... ME ... | ...let... DEC ...denote... ME ... |
| | ...let... ME ...denote... DEC ... | ...Denoting... DEC ...by... ME ... |
| | ...Denote... DEC ...as... ME ... | |
| ignore patterns | ...assume... DEC ...has... ME ... | ...corresponding... ME ...for... DEC ... |
| | ...denote... DEC ...from... ME ... | ...define... ME ...on... DEC ... |
| | ... DEC ...denote...as... ME ... | ... ME ...belongs...to... DEC ... |
| | ...denote... DEC ...set... ME ... | ...containing... ME ...,... DEC ... |
| | ...add... ME ...to... DEC ... | ...denote... ME ...to... DEC ... |
| | ...set... ME ...be... DEC ... | ... ME ...according...to... DEC ... |
| | ... ME ...depends...on... DEC ... | ...Let... DEC ...set... ME ... |
| | ...define... ME ...for... DEC ... | ...use... ME ...for... DEC ... |
| | ...Let... DEC ...for... ME ... | ...Let... ME ...for... DEC ... |
| | ...using... ME ...to... DEC ... | ...write... ME ...to... DEC ... |
| | ...for... ME ...then... DEC ... | ...sends... ME ...to... DEC ... |
| | ...Let... ME ...indicate... DEC ... | ...by... DEC ...let... ME ... |
| | ...using... DEC ...with... ME ... | ...for... ME ...given... DEC ... |
| | ... DEC ...induced...by... ME ... | ...for... ME ...,... DEC ... |
| | ...think... ME ...as... DEC ... | ...with... ME ...nodes... DEC ... |
| | ...with... ME ...then... DEC ... | ...is... DEC ...containing... ME ... |
| | ...to... ME ...given... DEC ... | ... ME ...belonging...to... DEC ... |
| | ...to... ME ...has... DEC ... | ...node... ME ...node... DEC ... |
| | ...consisting... ME ...,... DEC ... | ...to... DEC ...Let... ME ... |
| | ...centered... DEC ...with... ME ... | ...mapping... ME ...from... DEC ... |
| | ...have... DEC ...with... ME ... | ... DEC ...update...gate... ME ... |
| stop words | assume | has |
| | belongs | according |
| | depends | then |
| | sends | indicate |
| | induced | nodes |
| | belonging | node |
| | consisting | centered |
| | mapping | embedding |
| | conditioned | update |

# APPENDIX C. MECA SOFTWARE SYSTEM

As shown in Figure 6, the MECA software prototype has a modularized system architecture to implement five content analysis models, which are integrated through a Graphical User Interface (GUI). Organized into packages, the analytical models include ME extraction, ME layout analysis, ME semantic analysis, declaration extraction, and QuQn dependency analysis. These components use a shared data pipeline to exchange data, and each of them has its training and testing components. In the *Document layout analysis* package in Figure 6, the ME Extraction module starts with the PDF parsing [19] [18], which extracts raw rendering units and organizes them into a hierarchy of column-line-NSCS by layout analysis. Then the TSM and MRF-TSM modules will identify MEs from NSCSes. The identified MEs are represented as a set of characters with values and positions, which are then processed by the *ME layout analysis* package. The CCS module organizes the set of characters into a hierarchy of ME layout structures on top of the typographic system and the parametric lognormal modeling of the relative spatial relationship, i.e., the PHN model. The ME layout structure is further parsed into ME objects according to the ME semantic taxonomy through tokenization, parsing, and tree re-writing as shown in the *ME semantic analysis* package. The PCFG parsing framework from NLTK [101] is adopted for prototyping the ME semantical parser. Besides the analysis of MEs alone, we also extract the bonding declaration phrase for MEs in the *Declaration extraction* package. The mixed Word-ME sentences obtained from the PDF parser are fed into our customized PoS tagger, NP chunker, and declaration extractor sequentially. At last, we consolidate the *qualitative* bonding declarations and the *quantitative* MEs into a unified graphical representation in the *QuQn abstraction* package. The Object-Oriented representation of

the ME semantic objects greatly simplifies the dependency analysis and pruning of the QuQn map.

The analytical products from the MECA modules are stored in MongoDB[2], which is a non-SQL database due to its flexibility in the table schema for fast iteration. The objects for the complex structure are serialized into a string to store into databased and deserialized when reading from the database. Besides saving the content extracted by the automated system, user corrections and annotations could also be stored into the database only with the extra field about the annotator and timestamps.

The graphical user interface GUI is based on the Model-View-Control (MVC) architecture pattern. Given that most of the models are written in Python, we choose a lightweight web framework *web.py*[3] to develop the MVC based GUI. The front-end interaction is based on Hypertext Markup Language (HTML) and JavaScript. The communication between the front end and backend subsystems is based on asynchronous JavaScript and XML (AJAX) for smooth interaction experience without refreshing the page. The Bootstrap CCS library[4] is used for the webpage layout design. Pdf.js library[5] is used for rendering PDF on the webpage and the annotations on the PDF are implement as colored transparent rectangles. The d3.js library[6] is used for the interactive visualization of the *QuQn map* and the *knowledge evolution*. The *reading*

---

[2] https://www.mongodb.com/

[3] http://webpy.org/

[4] https://getbootstrap.com/

[5] https://mozilla.github.io/pdf.js/

[6] https://d3js.org/

*assistant* integrates the *PDF viewer* and the *QuQn map* and synchronizes the focused information. The MEs visible in the current PDF viewer is highlighted in the QuQn map; a click on an ME in the QuQn map leads to the automated scrolling of PDF viewer to the corresponding page containing the clicked ME.

Parallel processing and caching are heavily adopted across the system for processing speed up and avoidance of duplicated computations. For parallelization, each PDF page is passed to one computing process for PDF parsing and ME Extraction. Each user at the frontend will be dispatched to a different process by the Apache web server. On the other side, Memcached[7] is used to store the short-term, intermediate results such as the request by a user, such as getting all the MEs for a PDF page. MongoDB is used to store long-term intermediate results that might be used by many later stages of processing such as the result of PDF parsing. Special care is taken to ensure the consistency among copies of the same information at different caching layers and the front-end GUI. For example, when a user corrects the extracted MEs, the correction is stored into the MongoDB. Correspondingly, we will also update Memcached for the corresponding item. Further, the highlight in the front-end is also updated accordingly, such as removing a deleted ME or create a new annotation rectangle for a new added ME.

[7] https://memcached.org/