CONTEXT CLASSIFICATION FOR IMPROVED SEMANTIC UNDERSTANDING OF MATHEMATICAL FORMULAE

by

RANDA ALMOMEN

A thesis submitted to
The University of Birmingham
for the degree of
DOCTOR OF PHILOSOPHY

School of Computer Science
College of Engineering and Physical Sciences
The University of Birmingham
March 2017

ABSTRACT

The correct semantic interpretation of mathematical formulae in electronic mathematical documents is an important prerequisite for advanced tasks such as search, accessibility or computational processing. Especially in advanced maths, the meaning of characters and symbols is highly domain dependent, and only limited information can be gained from considering individual formulae and their structures. Although many approaches have been proposed for semantic interpretation of mathematical formulae, most of them rely on the limited semantics from maths representation languages whereas very few use maths context as a source of information.

This thesis presents a novel approach for principal extraction of semantic information of mathematical formulae from their context in documents. We utilise different supervised machine learning (SML) techniques (i.e. Linear-Chain Conditional Random Fields (CRF), Maximum Entropy (MaxEnt) and Maximum Entropy Markov Models (MEMM) combined with Rprop$^-$ and Rprop$^+$ optimisation algorithms) to detect definitions of simple and compound mathematical expressions, thereby deriving their meaning. The learning algorithms demand annotated corpus which its development considered as one of this thesis contributions. The corpus has been developed utilising a novel approach to extract desired maths expressions and sub-formulae and manually annotated by two independent annotators employing a standard measure for inter-annotation agreement. The thesis further developed a new approach to feature representation depending on the definitions' templates that extracted from maths documents to defeat the restraint of conventional window-based features. All contributions were evaluated by various techniques including employing the common metrics recall, precision, and harmonic F-measure.

ACKNOWLEDGEMENTS

In the name of Allah, the Most Gracious and the Most Merciful

I thank Allah that he provided me with guidance and determination to complete this thesis.

After thanking Allah, I would like to express my special appreciation and thanks to my supervisor; Volker Sorge, for his support, encouragement and guidance. Volker has believed in my ability to accomplish this thesis and stood by me during my difficult times.

I would like to thank the members of my research monitoring group; Mark Lee and John Barnden, for their helpful suggestions and academic support.

I would like to express my gratitude and thanks to my parents; Abdulrahman and Badriya, who never spare an effort to facilitate my way to achieve the dream.

Special gratitude and thanks go to my dear husband, Rayid, and my lovely kids; Rafa, Omar and Maice, for their support and patience when I was busy with my research.

I extend my gratitude to my brothers and sisters; Hesham, Abdullah, Osama, Susan and Entessar, who never hesitate to encourage, advice and support me.

I would like to acknowledge all my friends; especially, Souad, Lulua, Doaa, Marwa, Nada, Fahda and Marwa, who always put a smile on my face even during hard times.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

Abbreviations	Full Form
A	Accuracy
CRF	Linear-Chain Conditional Random Fields
F	F-measure
GSC	Gold Standard Corpus
IE	Information Extraction
κ-statistic|κ	Kappa statistic
MaxEnt	Maximum Entropy
MEMM	Maximum Entropy Markov Models
ML	Machine Learning
NP	Noun Phrase
P	Precision
POS	Part-Of-Speech
R	Recall
Rprop	Resilient Propagation Algorithms
Rprop$^+$	Variation of Resilient Propagation Algorithm
Rprop$^-$	Variation of Resilient Propagation Algorithm
SML	Supervised Machine Learning
SMT	Statistical Machine Translation
SSL	Semi-Supervised Learning
VP	Verb Phrase

CHAPTER 1

INTRODUCTION

The correct semantic interpretation of mathematical formulae that are recognised in documents is significant in several areas such as improving the precision of systems that translate documents into speech or other formats. It is also important in improving the accessibility of maths documents and precision of existing maths search systems. For instance, the formula in Equation 1.1 could be understood as f, a and b are variables, $+$ is the traditional addition operation, and there is an invisible multiplication between f and the opening bracket.

$$f(a+b) \tag{1.1}$$

Therefore a possible interpretation of Equation 1.1 would be

$$f(a+b) = f.a + f.b \tag{1.2}$$

Another possibility is that f is a function to be applied to the variable $a+b$. Indeed both semantic meanings are perfectly legal, but it depends on the context or the document the maths formula has been extracted from. Also, if we have the formula

$$H \leqslant G \tag{1.3}$$

and we know that the context of this formula is in the domain of Group Theory, then we can interpret Equation 1.3 as H is a subgroup of a group G.

There is an unlimited number of maths symbols which have unlimited usage, and mathematicians use them in different ways. Therefore, it is crucial to understand how mathematicians use maths symbols and formulae and how they tend to define them. Moreover, it is noticeable and acceptable that a maths symbol is used before being precisely declared within its context [70]. In general, when mathematicians write a document, they tend to define some of the used maths symbols and formulae in the context and leave some others without definitions. This usage is summarised in three possibilities:

- **A maths expression is never defined within the document.** This is because the undefined maths expressions have well-known meanings either in general such as the symbol $=$ or in a particular maths field such as $\leqslant$ which means a subgroup of a group in the field of Group Theory.

- **A maths expression is defined once within the document.** This means that the expression has a unique definition throughout the document.

- **A maths expression is defined several times within the document.** This means that the definition of this expression is changing throughout the document such as starting with a particular definition and later in the document adding some restrictions on the initial definition. In this case, it is vital to determine the scope of each definition of the math expression.

In recent years many approaches have been proposed for semantic interpretation of mathematical formulae. Most of them have relied on the limited semantics from maths representation languages in order to be used for various tasks, for example, semantical enrichment of mathematical markup languages like producing Content MathML from Presentation MathML. However, there have been limited attempts made using maths context as a source of semantic information.

Our research aims to narrow the gap between what an expert mathematician can interpret and what a machine can interpret. Therefore, we developed an approach to determine the semantics of mathematical formulae by analysing both the mathematical formulae and their context. To ease the start of our research, we restricted our data to maths documents from a specific maths domain; in particular, Elementary Number Theory. Nevertheless, this restriction could be released afterwards to extend the research.

In this thesis, a novel approach is proposed for extracting semantic information from maths context by adapting supervised machine learning; in particular, statistical learning algorithms. In our approach, the maths context information is used to distinguish and extract the defined maths formulae within a document alongside their definitions. We demonstrate our approach for building MathExtractor, a tool to extract the desired maths expression depending on their properties such as type, position and font. Moreover, we present our approach to build a gold standard corpus (GSC) that required by the statistical algorithms. Alongside the research, all the contributions were evaluated both quantitatively and qualitatively employing different techniques.

1.1 The scientific questions

In this thesis I address the following scientific questions:

- How can the maths formulae be recognised and extracted from the XML format of documents depending on maths formulae properties?
- How can one extract semantic information for a particular mathematical formula from the context information?
- How can one adapt supervised machine learning techniques for text analyses in the presence of mathematical formulae?
- Which probabilistic model (i.e. classifier) is the most efficient for extracting the

defined maths formulae with their definitions from maths documents?

- What are the instructive features that can be obtained from mathematical documents to be utilised by the probabilistic model?

1.2 Contributions

A summary of the contributions of this thesis is as follows:

1. Describing a novel approach for developing MathExtractor, a tool that extracts mathematical formulae from the XML format of the documents depending on formulae properties such as type, position and font.

2. Demonstrating the possibility of adapting the supervised machine learning techniques for text analyses in the presence of mathematical formulae by abstracting mathematical documents from maths formulae and replacing them with unique IDs.

3. Demonstrating a novel approach for extracting the semantic information for mathematical formulae from the context information by adapting supervised machine learning techniques; in particular, statistical learning algorithms. I apply three classifiers; Maximum Entropy Markov Models (MEMM), Maximum Entropy (MaxEnt) and Linear-Chain Conditional Random Fields (CRF) combined with Rprop$^-$ and Rprop$^+$ optimisation algorithms, to extract semantic information from maths context. I evaluate and compare their performance to investigate the sufficient one among them for our task.

4. Developing a manually-created gold standard corpus (GSC), which its documents are mathematical that harvested from the ArXive.

5. Developed a new approach for feature representation relying on the definitions' templates that extracted from maths documents to defeat the restraint of conventional window-based features; and therefore, enhancing the performance of the classifier.

6. Describing the extraction of basic semantic information such as font, maths style and the syntactic and semantic roles from the representation of maths formulae; which is the internal representation of LaTeXML.

1.3 Publications

Part of this thesis is based on published work in conferences and workshops as follows:

- Almomen, R. and Sorge, V., Semantic Understanding of Mathematical Formulae in Documents. In *Automated Reasoning Workshop 2015 Bridging the Gap between Theory and Practice ARW 2015.*

In Writing Papers:

- Almomen, R. and Sorge, V., A Gold Standard Corpus for Mathematical Documents. In Writing.
- Almomen, R., Alotaibi, Fahd S. and Sorge, V., Towards Context-based Extraction of Mathematical Formulae. In Writing.

1.4 Thesis Overview

This thesis consists of four parts which include eight chapters (not including the introduction). The thesis is structured as follows:

Part I: Background

Chapter 2 provides an overview of different markup languages for representing mathematical formulae. Also, it provides an overview of some tools for different tasks: LaTeX to XML converter and annotation tool. It is presenting an overview of information extraction (IE) methods and different approaches for evaluating IE systems.

Chapter 3 provides an overview of some research that related to the work presented in this thesis.

Part II: Resource Creation

Chapter 4 provides an overview of our system architecture. It demonstrates the methodology for extracting maths expressions from the documents. Also, it demonstrates our approach to build a gold standard corpus using mathematical documents.

Chapter 5 provides an overview of our implementation methodology.

Part III: Mathematical Semantic Recognition

Chapter 6 presents our approach for extracting the semantic information of maths formulae from two different sources; the representation of maths formulae and maths context where we utilised supervised machine learning techniques.

Chapter 7 discuss our approach to improve the extraction of maths definitions from their context by enhancing the performance of the classifiers.

Part IV: Evaluation

Chapter 8 presents different quantitative and qualitative evaluation techniques for each stage of our approach.

Chapter 9 concludes the thesis and presents some future work to improve and extend our research.

Part I

Background

CHAPTER 2

RESEARCH BACKGROUND

In this chapter, we provide an overview of different markup languages concerning representing mathematical formulae and some tools for different tasks: LaTeX to XML converter and annotation tool. We also provide an overview of information extraction methods and different approaches for evaluating IE systems. Besides, we discuss building a gold standard corpus.

2.1 Representation of Mathematics

A mathematical markup language is computer documentation that represents mathematical expressions. There are different mathematical representation markup languages; as some of them are software dependent relying on particular semantic interpretation systems such as computer algebra systems. On the other hand, some general markup languages are primarily concerned with the syntax side such as LaTeX and Presentation MathML. Others are more concerned with the semantic side, such as Content MathML, OpenMath and the internal representation of LaTeXML for maths formulae.

2.1.1 Syntactic Representation

There are some markup languages such as LaTeX and Presentation MathML that are concerned with the layout structures of mathematical formulae. Therefore, they illustrate

Table 2.1: Examples of presentation MathML

$b+2$	$I^2 = \int x\,\mathrm{d}x$

```
<math>
  <mrow>
    <mi>b</mi>
    <mo>+</mo>
    <mn>2</mn>
  </mrow>
</math>
```

```
<math>
  <mrow>
    <mrow>
      <msup>
        <mi>I</mi>
        <mn>2</mn>
      </msup>
    </mrow>
    <mo>=</mo>
      <mrow>
        <msubsup>
          <mo>&int;</mo>
        </msubsup>
        <mrow>
          <mi>x</mi>
          <mo>&dd;</mo>
          <mi>x</mi>
        </mrow>
      </mrow>
  </mrow>
</math>
```

how mathematical formulae appear regarding characters used, size, colour and the precise positioning of each character [53, 9].

Such markup languages are useful in case the display of maths formulae is an important issue such as using maths on a web page for reading only. [69].

For instance, Equation 2.1 and Equation 2.2 would be written in Presentation MathML as shown in Table 2.1

$$b+2 \tag{2.1}$$

$$I^2 = \int x\,\mathrm{d}x \tag{2.2}$$

However, this representation of Equation 2.2 is read as: I, second power, equals, integral sign, x, d, x. So Presentation MathML illustrates only the way of presenting the formula rather than its actual meaning.

On the other hand, LaTeX is the typesetting system that is the most popular and powerful in the scientific world. It assists the user in rendering mathematical formulae to a high level of typographic [26, 2]. However, Equation 2.1 and Equation 2.2 are written in LaTeX respectively as:

\[$b + 2$ \]

and

\[I $\wedge$ 2 = \int x \, $mathrm{d}$x \]

2.1.2 Semantic Representation

Several markup languages represent mathematical formulae and concern themselves more with the semantic side. This sub-section presents a summary of the three primary examples, Content MathML, OpenMath and the internal representation of LaTeXML for mathematical expressions.

2.1.2.1 Content MathML

Content MathML is a general markup language that represents mathematical formulae according to their logical meaning [69]. Its main goal is as a bridge between the layout of formulae and their semantics [68]. Content MathML is useful when the mathematical meaning is an important issue such as displaying maths expressions on a web page where users can copy and past these expressions [69]. The markup language is composed of approximately 140 elements and 12 attributes. Table 2.2 shows the content MathML representation of the Equation 2.1 and Equation 2.2. Indeed Equation 2.2 is read as I to the second power is equal to the integral of x with respect to x. Content MathML successfully conserves the semantics of math formulae.

2.1.2.2 OpenMath

OpenMath is a general markup language that represents mathematical formulae with their semantics [12]. "OpenMath is aimed at encoding the semantics of mathematics and,

Table 2.2: Examples of content MathML

$b+2$	$I^2 = \int x \, dx$
<math> <apply> <plus/> <ci>b</ci> <cn>2</cn> </apply> </math>	<math> <apply> <eq/> <apply> <power/> <ci>I</ci> <cn>2</cn> </apply> <apply> <int/> <bvar> <ci>x</ci> </bvar> <ci>x</ci> </apply> </apply> </math>

via its extensible Content Dictionary mechanism, may be applied to arbitrary areas of mathematics without the need for any central agreement to change the language" [64]. Equation 2.1 is written in OpenMath as:

```
<OMOBJ>
  <OMA>
    <OMS cd = "arith1" name="plus"/>
    <OMV name="b"/>
    <OMI>2</OMI>
  </OMA>
</OMOBJ>
```

2.1.2.3 Internal Representation of LaTeXML

LaTeXML is the LaTeX to XML/HTML/MathML converter [43]. By using LaTeXML to convert the mathematical documents from LaTeX format into XML format, we can have the maths expressions represented in Presentation MathML or the internal representation

of LaTeXML format which is concern about the semantics of these expressions.
In the internal format of LaTeXML, all the representation of formula is stored in a Math element which serves as the primary repository for this representation. The Math element has an attribute 'mode' that determine whether the formula to be inline or on display. However, the element Math is looked at as an inline maths, and if the expression should be in a display mode, then Math is contained in another element such as 'equation' or 'equationgroup'. Figure 4.2 and Figure 2.1 show examples of using Math element as inline and as in display mode contains in an equation tag, respectively.

The representation of sub-expressions are given different tags. In the following list we show the main tags:

- **XMApp:** The tag to presents "the generalized application of some function or operator to arguments" [4]. The first child node presents the operator while the rest nodes present the arguments.
- **XMTok:** The tag to provide information about a mathematical symbol which possibly includes text [4].
- **XMDual:** Integrates the first child which is the content's representation, and the second child which is the presentation [4].
- **XMWrap:** Confirm the predictable subexpression's role or type which might be hard to determine its intended meaning [4].

Each of these tags may have extra attributes such as name, font and style for the tag XMTok.

Figure 2.1: $I_0 \geq x\pi^{-1}(n) - 2^k$[49] in LaTeXML representation format

```
<equation xml:id="S3.Ex7">
  <Math mode="display" xml:id="S3.Ex7.m1" tex="I_{0}\geq x{\
     pi}^{-1}(n)-2^{k}." text="I _ 0 &gt;= x * pi ^ (- 1) * n
      - 2 ^ k">
    <XMath>
      <XMApp punctuation=".">
        <XMTok meaning="greater-than-or-equals" name="geq"
           role="RELOP">?</XMTok>
        <XMApp>
          <XMTok role="SUBSCRIPTOP" scriptpos="post2"/>
          <XMTok role="UNKNOWN" font="italic">I</XMTok>
          <XMTok meaning="0" role="NUMBER">0</XMTok>
        </XMApp>
        <XMApp>
          <XMTok meaning="minus" role="ADDOP">-</XMTok>
          <XMApp>
            <XMTok meaning="times" role="MULOP">?</XMTok>
            <XMTok role="UNKNOWN" font="italic">x</XMTok>
            <XMApp>
              <XMTok role="SUPERSCRIPTOP" scriptpos="post2"/>
              <XMTok name="pi" possibleFunction="yes" role="
                 UNKNOWN" font="italic">?</XMTok>
              <XMApp>
                <XMTok meaning="minus" role="ADDOP">-</XMTok>
                <XMTok meaning="1" role="NUMBER">1</XMTok>
              </XMApp>
            </XMApp>
            <XMTok close=")" open="(" role="UNKNOWN" font="
               italic">n</XMTok>
          </XMApp>
          <XMApp>
          <XMTok role="SUPERSCRIPTOP" scriptpos="post2"/>
          <XMTok meaning="2" role="NUMBER">2</XMTok>
          <XMTok role="UNKNOWN" font="italic">k</XMTok>
        </XMApp>
       </XMApp>
      </XMApp>
    </XMath>
  </Math>
</equation>
```

2.2 Information Extraction

2.2.1 An Overview of IE

Information extraction is defined by Moens [44] as "the identification, and consequent or concurrent classification and structuring into semantic classes, of specific information found in unstructured data sources, such as natural language text, making the information more suitable for information processing tasks".

This definition means that the IE is utilised to obtain information from unstructured data [59]. Unstructured data refers to the information that presented in a format which is hard for a computer to decide its intended meaning instantly such as text, images, audio and video. Our research is concerned about IE from a text as unstructured data; therefore, the other type of unstructured data will be disregarded in this section.

An IE system is employing a group of obtained patterns that formulated manually or learned automatically to get information from text and express it in a structured format [44]. The most common IE tasks incorporate named entity recognition, i.e. identifying already defined types of named entities such as organisations, person names, date, time and locations [51]. Other tasks are event extraction; which detect events and their details and constructions, and relation extraction which recognises the relations between entities in the text [51]. Besides, some domain dependent tasks are popular such as extracting scientific information from publications, extracting the indications and treatments of disease from patient records [44].

2.2.2 Approaches to IE

The approaches to information extraction are categorised into two main streams: handcrafted rule-based IE and machine learning (ML) based IE [59]. The ML is categorised into supervised, semi-supervised and unsupervised ML [73]. The approaches that used in the course of this research are reviewed in this section.

2.2.2.1 Handcrafted Rule-Based IE

Handcrafted rule-based systems concern about formulating and implementing syntactic extraction patterns and employ available information to recognise the targeted entities [73]. For instance, a rule for finding emails could be "an email is three list of characters X, Y and Z that have an '@' between X and Y and a '.' between Y and Z. For example, X could be randa, Y could be gmail and Z could be com; therefore the email 'randa@gmail.com' can be identified by such a rule.

In the literature, some known rule-based systems that recognise names entities using cautiously handcrafted regular expressions such as FASTUS [7]. Also, some rule-based systems extract candidate entities using substantial lookup lists of names of entities and grammar rules such as LaSIE II [31].

The rule-based IE systems are functional for fields that have a particular formalism for the expressions' structures [73]. The biology is an example of such fields where there are some related works done as in [20, 56].

Nevertheless, the crucial shortcoming of handcrafted rule-based systems is their excessive cost as they require experts in the domain, in the language and in programming as well to recognise and extract patterns manually [44]. Consequently, the researchers' attention has switched to machine learning approaches.

2.2.2.2 Machine-learning Based IE

Machine Learning is defined as "the field of study that gives computers the ability to learn without being explicitly programmed" [58]. There are three types of approaches to machine learning (ML); supervised, semi-supervised (SSL) and unsupervised learning. The supervised machine learning (SML) uses training (i.e. labelled) data in order to build a trained model and use it to predict labels for unseen data. Whereas the unsupervised ML determines patterns and essential structures in new (i.e. unseen) data without a need for training data but using a descriptive model. Thus, the SSL is learning by using both labelled data and unlabeled data.

Because our focus is on using an SML in an IE task, we will discuss the concept of SML in more detailed in this section. Supervised machine learning has two types of techniques:

- **Classification:** which is aiming at the discrete data such as whether a word is a noun or not.
- **Regression:** which is aiming at the data that changing continually such as temperature.

The problem of classification in SML can be solved by using a variety of algorithms such as logic-based algorithms and statistical learning algorithms [34]. Researchers have handled IE employing SML as the sequence tagging has been approached in part of speech (POS) and text chunking. It is essential to choose the appropriate type of algorithm depending on the task and requirement such as speed, memory usage and indeed the domain specifications as one of the constituents that influence the execution of an SML system is the probabilistic model.

2.2.2.2.1 Probabilistic Models

In the literature, many various probabilistic models have been used to develop IE systems such as Support Vector Machine, Maximum Entropy Models (MaxEnt), Hidden Markov Models, Conditional Random Fields (CRF), Maximum Entropy Markov Models (MEMM) and Decision Rules and Trees.

- **Maximum Entropy (MaxEnt)**

  MaxEnt is a discriminative model that categorise a character or a chain of characters into a class by integrating an extensive range of evidence [10]. Suppose an input sequence of words $w = (w_1, w_2, ..., w_n)$ and a defined set of m tags $t = (t_1, t_2, ..., t_m)$. The mission is to explore the most functional sequence of tags with the highest conditional probability between the possible tag sequences. The highest conditional probability appointed to tag t_i is regarded as a required class that w_i belongs to statistically;

$$t_i = \arg\max p(t_1^m | w_i)$$

The constraints that imposed by the training set on the model are symbolised by the state feature functions that depend on the current state only and defined as:

$$f_j(w_i, t_i) = \begin{cases} 1, & \text{if } (w_i, t_i) \text{ satisfies a certain constraint} \\ 0, & \text{otherwise} \end{cases} \tag{2.3}$$

The MaxEnt sequence labelling formula is defined as:

$$p(t_i|w_i) = \frac{1}{Z} \exp\left(\sum_{j=1}^{k} \lambda_j f_j(w_i, t_i)\right), 0 < \lambda_j < \infty$$

Where λ_j are variables that customised to model the observed statistics and Z is a normalising constant such that:

$$Z = \sum_{y} \exp\left(\sum_{j=1}^{k} \lambda_j f_j(w_i, t_i)\right)$$

- **Maximum Entropy Markov Models (MEMM)**

  MEMM is a probabilistic model that used for different text-related tasks such as information extraction and determining semantic role tags [10], yet it is performing better than the MaxEnt on such tasks [41]. MEMM is permitting symbolising observations as arbitrary overlapping features and determining the restricted possibility of state sequences given observation sequences, i.e. it is "a conditional model that represents the probability of reaching a state given an observation and the previous state" [41].

  Suppose S is a set of states such that s and s' are current and previous state $\in S$; respectively, and O is a set of possible observations. Then for each pair $< a, s >$ where a is a binary feature of the observation, the transition feature functions that

depend on the previous and current states are defined as:

$$f_{<a,s>}(o_i, s_i) = \begin{cases} 1, & \text{if } a(o_i) \text{ is true and } s = s_i \\ 0, & \text{otherwise} \end{cases}$$

The MEMM sequence labelling formula is defined as:

$$p_{s'}(s|o) = \frac{1}{Z(o, s')} \exp\left(\sum_k \lambda_k f_k(o, s)\right),$$

where $k =< a, s >$, λ_k are parameterised features, and $Z(o, s')$ is a normalising constant which makes the distribution sum to one.

- **Linear-Chain Conditional Random Fields (CRF)**

  CRF is the state-of-the-art for sequence labelling tasks [36]. It is a discriminative model that offers a flexible and robust technique to employ arbitrary sets of features while dependent on the surrounding words' tags [59]. Given a sequence of observations, a CRF determines the probabilities of likely tag concatenation [36]. Besides the previously defined state feature function Equation 2.3, transition feature function; which rely on the previous and current states is defined as:

$$f(y_{i-1}, y_i, x, i) = \begin{cases} 1, & \text{if } y_{i-1} \text{ , } y_i \text{ and } x \text{ satisfies certain constraints} \\ 0, & \text{otherwise} \end{cases}$$

  Where x is an input sequence, i an input position and y_{i-1} and y_i are class tags. Thus, the CRF is defined as:

$$p(y|x) = \frac{1}{Z} \exp\left(\sum_{j=1}^{k} \sum_{i=1}^{n} \lambda_j f_j(y_{i-1}, y_i, x, i)\right),$$

where λ_j are parameterised features and Z is a normalising constant such that:

$$Z = \sum_{y} \exp\left(\sum_{j=1}^{k}\sum_{i=1}^{n} \lambda_j f_j(y_{i-1}, y_i, x, i)\right)$$

2.2.2.2.2 Optimisation Algorithms

The backpropagation algorithm that applies the principle of gradient descent is the learning rule of the widely prevalent supervised learning systems [57].

Adjusting gradient-based algorithms with an independent step-sizes attempt to defeat the complexity of choosing the proper learning rates. This achieved by constraining the weight update for all connections in the course of the learning progress to reduce the oscillations to the minimum and to increase the update step-size to its maximum [32].

Let E be an arbitrary error measure which is differentiable with respect to the weights and w_{ij} be the weight from neuron j to neuron i. During each learning iteration, the weights are specified by:

$$w_{ij}^{(t+1)} = w_{ij}^{(t)} + \Delta w_{ij}^{(t)}$$

The learning algorithm halts as particular ending specifications are met.

Resilient back**prop**agation (Rprop) is an efficient learning algorithm in which the orientation of weight update is established on the sign of the partial derivative $\partial E/\partial w_{ij}$; where a "step-size Δ_{ij}, i.e., the update amount of a weight w_{ij}, is adapted for each weight individually" [32].

Rprop algorithm proposed by Riedmiller and Braun is a robust, precise and rapid algorithm in contrast with different supervised learning approaches [32].

The principal distinction about Rprop systems is that the step-sizes are not influenced by the partial derivatives' absolute value does not influence the step-sizes. Therefore, the step-sizes are calculated as follow:

$$\Delta w_{ij}^{(t)} = -\text{sign}\left(\frac{\partial E}{\partial w_{ij}}^{(t)}\right) \Delta_{ij}^{(t)}$$

Thus, the step-size Δ_{ij} is modified for each weight w_{ij} as:

$$\Delta_{ij}^{(t)} = \begin{cases} \min(\eta^{+}\Delta_{ij}^{(t-1)}, \Delta_{\max}), & \text{if } \frac{\partial E}{\partial w_{ij}}^{(t-1)} \quad \frac{\partial E}{\partial w_{ij}}^{(t)} > 0 \\ \max(\eta^{-}\Delta_{ij}^{(t-1)}, \Delta_{\min}), & \text{if } \frac{\partial E}{\partial w_{ij}}^{(t-1)} \quad \frac{\partial E}{\partial w_{ij}}^{(t)} < 0 \\ \Delta_{ij}^{(t-1)}, & \text{otherwise} \end{cases} \tag{2.4}$$

There are some variations of the Rprop algorithm such as Rprop^{+} [55] and Rprop^{-} [54].

- **Rprop^{+}:**

  Its idea is enhancing network training (weight-backtracking); in other words, for some or the whole weights, retracting a prior update. Following modifying the step-sizes in line with Equation 2.4, the weight updates w_{ij} are specified. The two recognised possibilities are:

  - if there is no difference in the sign of the partial derivative, then a regular weight update is carried out as:

$$\textbf{if } \frac{\partial E}{\partial w_{ij}}^{(t-1)} \quad \frac{\partial E}{\partial w_{ij}}^{(t)} \geq 0 \textbf{ then } \Delta w_{ij}^{(t)} = -\text{sign}\left(\frac{\partial E}{\partial w_{ij}}^{(t)}\right) \Delta_{ij}^{(t)} \tag{2.5}$$

  - If the sign of the partial derivative has changed, the former weight update is reverted:

$$\textbf{if } \frac{\partial E}{\partial w_{ij}}^{(t-1)} \quad \frac{\partial E}{\partial w_{ij}}^{(t)} < 0 \textbf{ then } \left\{ \Delta w_{ij}^{(t)} = -\Delta w_{ij}^{(t-1)}; \frac{\partial E}{\partial w_{ij}}^{(t)} = 0 \right\}$$

- **Rprop^{-}:**

  It is a Rprop without weight-backtracking. In other words, The weight-backtracking is excluded and the right-hand side of Equation 2.5 is employed in all situations. Therefore, keeping the previous weight updates is no longer needed.

To sum up, Rprop^{+} and Rprop^{-} algorithms employ the gradient to detect a right search path, but not to select the action to make in that path. Moreover, they do not require

parameter tuning to accomplish great predictions [55, 52].

2.2.2.3 Hybrid Based IE

In hybrid approaches, different systems are integrated into a single model. The objective of such approaches is to take advantage of the integrated systems in addition to the possibility of vanishing disadvantages to some extent by the usefulness [28]. For instance, the expert knowledge deficiency in the rule-based may clear up to some extent by integrating a statistical approach. In general, the prosperous rule-based models mostly comprise a hybrid model of handcrafted rule-based and automated systems [59]. The systems in [13, 18, 19, 33] are examples of hybrid models.

The hybrid approach in this thesis refers to a model that employs handcrafted rules to extract patterns from a text, then convey these patterns into a statistical model. The two concerns that have significant consequences on this approach is that the rules extraction task demands linguistic knowledge in the targeted domain; in addition to generalising the rules to evade the issue of overfitting.

2.2.3 Gold Standard Corpora

Gold Standard Corpora is annotated data with a standard level of reliability [27]. Having Gold Standard Corpora (GSC) is an essential requirement for a classifier [23, 27, 73]. This annotated data is enriched text with the needed information which can not replace the original text [23] and stands like a model that the machine learning algorithms are following to be trained and tested [73]. The annotation should be carried out by field experts to assure the valuable standard that will be learnt from [73]. Different experts are producing different annotations, as research shows [30]. Therefore, it is crucial to assess the reliability of annotations that are made by two or more annotators using the inter-annotator agreement. There are many different methods to measure the inter-annotator agreement such as F-measure [29] and Kappa statistic (κ-statistic) [14]. κ-statistic takes into consideration the possibility of agreement occurring by chance between annotators

and can be calculated as:

$$\kappa = \frac{P(A) - P(E)}{1 - P(E)} \tag{2.6}$$

where $P(A)$ is the number of actual agreement and $P(E)$ is the number of chance agreements [14].

Although Hripcsak and Rothschild [29] claimed that F-measure could be used to measure the inter-annotator agreement, it is known in the literature to measure a test's accuracy as illustrated in Section 2.2.5.

2.2.4 Annotation tools

Annotation tools are needed to ease the annotation process. There are various tools which depend on the desire annotation [17, 23]. Mainly, the annotations can be divided into two styles [23]:

- Dynamic annotations: which is associated with the text.
- Static annotations: which is associated with a specific location in the text pages.

The dynamic annotations are functional for the tasks of IE; therefore, the static annotations will be neglected in this thesis. In this section, we will exhibit the dynamic annotation tool Brat [62] which is used in our research as it is advocating structured annotations that are necessary for tasks involving both syntactical and semantical text properties.

Brat is "a web based structured annotation tool for text documents" [23]. It has two simple predetermined classifications:

- **Tagset:** which contain the tags that can be assigned to words in the text to be annotated.
- **Relations:** which is to connect the tags that are given to the words (i.e. tokens).

For example, Figure 4.7 shows the sentence "Let $\mathbb{Z}_n$ denote the ring of integers modulo n." [60] annotated in Brat. In this example, expression and definition are the tags from tagset. The definition on the arrow is a relation from expression to a definition. Figure 4.8 is another example showing a segment of text annotated in Brat.

Brat has useful administration functionalities such as an individual address for each label and an excellent searching tool for labelling.

2.2.5 Evaluation of IE

As a system in information extraction is built, it is essential to evaluate it to perceive its acting in comparison with a gold standard and other available systems [44]. In the literature, several metrics commonly used to evaluate the performance of IE systems such as the standard metrics: accuracy, recall, precision and F-measure. These metrics are defined as follows:

- **Accuracy** is "the percentage of correct predictions divided by the total number of predictions" [34].

  i.e.

$$A = \frac{\text{correct and found items + not correct and not found items}}{\text{total number of predictions made}} \tag{2.7}$$

- **Recall** is the number of "relevant items that we identified" [10].

  i.e.

$$R = \frac{\text{correct and found items}}{\text{correct items}} \tag{2.8}$$

- **Precision** is the number of "items that we identified were relevant" [10].

  i.e.

$$P = \frac{\text{correct and found items}}{\text{found items}} \tag{2.9}$$

- **F-measure** is a harmonic metric that combining both recall and precision as follows:

$$F_\beta = \frac{(1+\beta^2)PR}{(\beta^2 P)+R} \tag{2.10}$$

  Where β is controlling the weight of precision in this equation; i.e. setting β to be one is allowing a balanced weight for both precision and recall.

The error in labelling is prevalently analysed using a confusion matrix mainly if the classification task contains more than two classes [40]. The confusion matrix express for each pair of classes <c1, c2> the number of tokens from class <c1> was incorrectly allocated to class <c2> and contrariwise. Therefore, it assists in discovering the opportunity to improve the performance of the system; by precisely determine the type of the most frequent error in tagging. For instance, Table 2.3 shows that the IE system accomplishes to differentiate the three classes: expression (exp), definition (def), and a first part of the definition (p1D) while producing errors within two classes. This number of errors could be reduced by providing different features that differentiate between def and p1D.

Table 2.3: An example of confusion matrix for a IE system that classify tokens into three classes (i.e. exp, def, and p1D)

	exp	def	p1D
exp	20	0	0
def	0	12	2
p1D	0	4	5

CHAPTER 3

RELATED WORK

In order to understand mathematical content, we need to study both the mathematical formulae and their context in addition to any background knowledge. For instance, consider the following fragment:

$$\ldots \text{ Let } f \text{ be a function such that } f(x) = x^2 + 1, x \in \mathbb{R} \ldots$$

The reader can intuit the interpretation of some symbols in this formula. As $+$ is the standard addition, $\in$ means "belong to", $=$ means "equals" and so on. Indeed there are some polysemous symbols and characters such as $\leqslant$ in Equation 1.3 which could be understood either by interpreting the context or from the domain of the document.

Nevertheless, recently there have been some efforts to analyse the maths formulae and their context in order to automate the understanding of the interpretation of maths expressions, and we will discuss some of these approaches in this chapter.

3.1 Extracting Semantic and Structural Information from Mathematical Formulae

It is possible to understand maths formulae up to a certain level by analysing the formulae themselves without taking account of their context. One approach of doing this is by analysing the surrounding whitespace of the sub-formula within a PDF file by using fonts information and character spacing [8]. Nonetheless, this approach provides very limited,

and in some cases, inaccurate semantics as it classifies maths entities into one of the following categories as stated in [8]:

- Ord: Ordinary symbol, such as Roman or Greek letters, digits.
- Op: Large operators, such as sum or integral signs.
- Bin: Binary operator, such as plus or minus signs.
- Rel: Relational operator, such as equality or greater than signs.
- Open: Opening punctuation, such as opening brackets.
- Close: Closing punctuation, such as closing brackets.
- Punct: Other punctuation, such as commas, exclamation marks.
- Inner: Fractional expression, such as an ordinary division.

Another approach to extract semantics of maths expressions is by analysing the expressions themselves in their LaTeX format to enrich their presentation MathML format semantically [63] using some grammar rules. This approach is combining several stages; initially, the document in LaTeX format is parsed with the SGLR parser [66]. This followed by a series of rewriting of the parse tree that maintains all the syntactical features of the mathematical formula and then unparsing the previously constructed XML parse tree to construct an XML document. Finally, the resulted XML document is passed into the Mozilla tool to present it in the representation MathML format. However, this approach is limited to a particular maths area and project, yet can be extended to other domains or different mathematician's writing by engineering the used grammar in a particular way that suits the targeted project. Besides, with this approach, there are some cases where the maths could not be treated automatically such as the case with the formula $\int \int \cdots \int f(t)(dt)^n$.

The method proposed by Nghiem et al. [47] aimed at enriching the presentation MathML of maths expressions semantically to produce their content MathML format. Their approach is incorporating automatic discovering of rules' fragment and statistical machine translation (SMT); where the used dataset is mathematical formulae from The Wolfram Functions Site [5] which provides the maths formulae in Presentation MathML

along with Content MathML format. For the evaluation, they contrast their results with a baseline model that built utilising the SMT on the original Wolfram maths expressions using the Translation Error Rate metric. Their utilisation of the fragment rules decreased the error rate by 10%. However, two critical issues with this approach; that the SMT does not fully meet the demand of the translation for long-distance reordering in addition to the severe difficulty with translating the long and intricate math formulae.

In general, the approach of interpreting maths by analysing the expressions in isolation from their context and domain information generates limited and sometimes inaccurate semantic information as it uses the maths expressions alone without considering their context and domain which carries crucial semantic information.

3.2 Extracting Semantic Information from Mathematical Context

In recent years, there have been limited attempts to use mathematical context information in order to interpret mathematical formulae. Yokoi et al. [72] presented an approach that interpreted maths formulae utilising semantic analysis of the maths context. The dataset is a 100 computer science papers that published by the Information Processing Society of Japan [6]. The initial step is to convert all maths formulae that included in the dataset into presentation MathML format. This followed by annotating the dataset manually so that each maths expression is connected to its name and definition, where for the easiness, the mathematical references nominees are restricted to the compound nouns only. Finally, the maths expressions with their interpretations are recognised in three different ways as follows:

- **Baseline Model:** In this experiment, the aim is to recognise phrases which refer to maths formula's interpretation to serve as a baseline model. Starting by parsing the sentences that comprise desired maths formulae, then obtaining the noun phrases

(NP) employing simple extraction rules. Subsequently, a binary classification is employed for each NP to conclude if it is correlated to the desired maths formula within the same sentence.

- **Pattern Matching Based Approach:** The objective of this approach is to illustrate in what way is getting the mathematical references; by some typical patterns that connect maths formulae and their references, useful. The most common eight patterns are obtained manually from five papers chosen from the same source of the dataset but different from them. Finally, the obtained patterns are utilised by a binary classifier to recognise the maths expressions with their connected NP that identical to any of these patterns.

- **Machine Learning Based Approach:** In this approach, a supervised machine learning (SVM) model is employed as a binary classifier to recognise the mathematical formulae's names and definitions following the same scheme as the pattern matching based method to discover utilising the fundamental patterns in addition to some linguistic information. They employed four strains of features are: the eight previously obtained patterns, some tokens that determine the sentences' construction to recognise the relation between NP and maths formula within the sentence, the surrounding tokens of both the NP and maths formula and finally the dependency that is connecting the NP and the expressions.

For the evaluation, they used the metrics recall, precision and F1-measure for each model. In general, the evaluation shows that the machine learning based approach achieved higher metrics' values than the pattern matching based approach. The best accomplishment resulted when the ML method with the dependency analysis features are employed. Moreover, they expected that their suggested approach could be usable in different languages as a result of the way that the mathematical expressions patterns follow.

Another approach presented by Stathopoulos and Teufel [61] to automate the recog-

nition of mathematical definitions in documents. They used the term 'type' to refer to a mathematical definition and described it as "any technical term that is (a) perceived by mathematicians to refer to mathematical objects, algebraic structures and mathematical notions and (b) can be instantiated in the discourse in the form of a variable" [61], where it is mostly confined as noun or prepositional phrases. Moreover, the recognised maths definitions were used to build a dictionary that plays an important rule in several aspects such as the mathematical information retrieval systems. In this approach, they started by extracting the candidate maths definitions utilising the C-Value algorithm [22] which integrates a statistical and a linguistic to recognise 'multi-word technical terms'. Then, the technical terms that are most probably be definitions are selected and comprised in a dictionary of maths definitions that consist of 10601 definitions. This approach was evaluated qualitatively employing a gold standard collection of definitions that created by five expert mathematicians. For the task of recognising the maths definitions, they reported 81.8% recall, 73.9% precision and 77.7% F-measure, as for the greater part judgment. Also, the inter-annotator agreement was measured applying Fleiss's Kappa [21], which had an intermediate range as $K = 0.65$.

Grigore et al. [25] have proposed an approach to exploratory investigate the disambiguation of a particular group of mathematical formulae in maths documents. They targeted the mathematical expressions that occur after a noun. They accumulated 'Term Clusters (TC)' from OpenMath Content Dictionaries and determined the nominee target maths expressions. Then, the 'corpus-based similarities' were calculated for every TC. Ultimately, each target formula was disambiguated relaying on their context and the TCs. For the performance assessment, the metrics recall, precision, $F_0.5$ and mean reciprocal rank (MRR) were used. The $F_0.5$, which is weighing precision twice as recall, was used as the correct disambiguation is preferable to coverage in the task of maths formula disambiguation. Moreover, two baselines were build to be used tor the evaluation task. The first baseline is trivial that solely allocated a random order of classes and has no

context information. Whereas, In the second baseline, restricted context information was employed as just the noun (NN) that occurs directly before the target maths expression was considered as a candidate for disambiguation. As there were restricted admittance on the baselines for the baselines, to context information, the performance of the two baselines was poor. Thus, despite the limited linguistic information employed in the stated method, the results are encouraging as the lexical context being beneficial to the task.

The approach presented in [25] was extended in [71] where they aim at simple maths expressions i.e. those with 'high- level structure'. Discovering the semantics of mathematical formulae have been performed over three main steps: Firstly, they preprocessed the documents and distinguished the simple math expressions. Next, the similarity between the linguistic context of identified maths expressions and all the collections of maths terms in the lexical resource were determined for each identified simple expression. Eventually, a scoring function was employed to specify the simple maths expressions' interpretation. To determine the semantic similarity among lexical contexts, the co-occurrence statistics were computed by applying the following: firstly, the bounded context of maths formula that being analysis ('local discourse') in addition to the appropriate parts of the document ('global discourse'). Secondly, collections of phrases of a built lexical source. A gold standard of maths expressions and their interpretations was produced by expert mathematicians to be utilised for the evaluation task of interpreting maths expressions. They employed two metrics for the evaluation measurement; precision and mean reciprocal rank (MRR). The estimated precision values promoted additional research, especially with more investigation regarding the linguistic information. Moreover, the results of this experiment confirmed that integrating the utilisation of both local and global context performed better than the utilisation of each of them separately.

In the literature, there have been some efforts to use mathematical context information in order to enrich mathematical markup languages such as converting Presentation

MathML into Content MathML. Nghiem et al. [46] have presented an approach that produces Content MathML format of maths formulae while having the input maths in the Presentation MathML format that was collected from The Wolfram Functions Site [5] which provides different levels of categorisation for each maths formulae. Therefore, in this approach, they do not have the actual maths context. They adapt Support Vector Machine classifier and use the categories of Wolfram to disambiguate the content of the identifier (mi) in the Presentation MathML. However, they claimed that if maths context is available, it will be used in a boolean way to judge whether an identifier has assigned a correct content or not.

Part II

Resource Creation

CHAPTER 4

GOLD STANDARD CORPUS GENERATION

In this chapter, we will demonstrate our approach to build a gold standard corpus which is an essential requirement for a probabilistic model. In Section 4.1 we will provide an overview of our approach to extract the semantic information of maths formulae from their context. The remaining of this chapter will demonstrate the methodology to build the gold standard corpus which includes collecting the data, extracting the maths formulae, preparing the data and finally annotating the data.

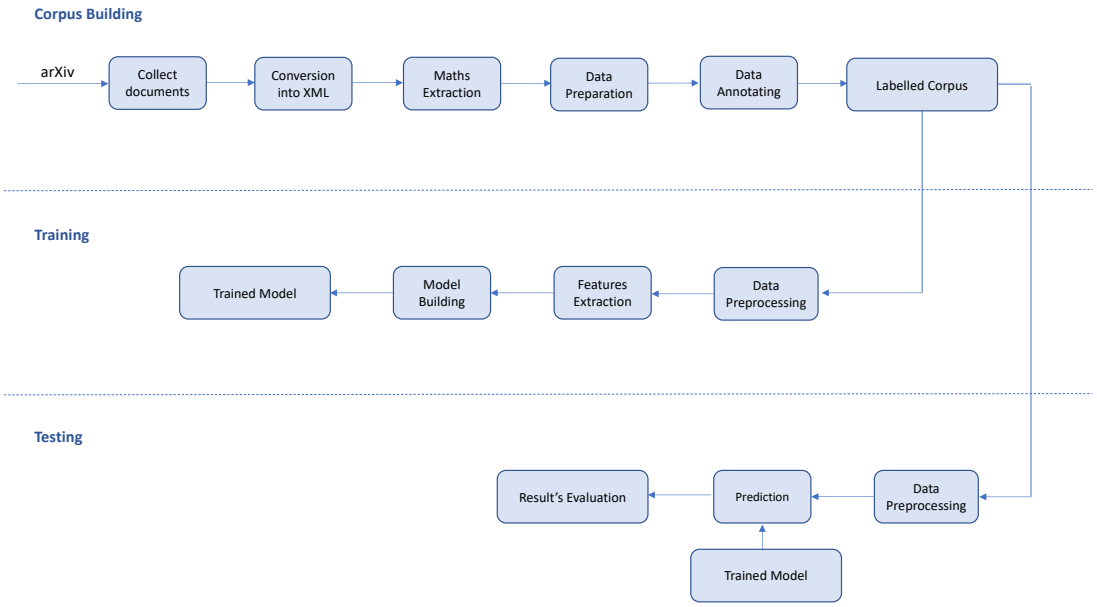
4.1 System Architecture

The architecture diagram in Figure 4.1 shows the main components of our system which are in three phases; building the corpus, the training phase and testing phase. In this chapter, we will demonstrate our approach to build the GSC. Whereas, the training and testing stages are the schemes for the experiments which will be discussed in Chapter 6. However, each of these phases consists of several steps as follows:

4.1.1 Building the Corpus

In this phase, we built a gold standard corpus (GSC). We started from the source code of the documents which are in LaTeX format and produced annotated data with a standard level of reliability in text format. This has been done in five stages as follows:

Figure 4.1: System architecture



1. **Collecting the Input Data**

   Maths documents in the field of Elementary Number Theory are collected randomly, and their source codes which are in LaTeX format are used as the initial input.

2. **Conversion into XML**

   The input as LaTeX format is converted into XML format in which each maths expression has a unique XML ID.

3. **Maths Extraction**

   All maths expressions are extracted and saved into XML files to facilitate dealing with the expressions such as accessing and editing.

4. **Data Preparation**

   In this stage we prepared the data for the labelling, i.e. annotating stage by the following steps:

   i) **Generating unique ID**

   A unique ID is generated for each unique maths expression. By unique expressions we mean the expressions without repetitions. For example, if we have in

a document the following expressions:

$F, G, x, x + y, x + y, F, y + x, F, G, G, y, y, y$

Then, the total number of maths expressions is 13 while the unique maths expressions are six which are $F, G, x, y, x + y$ and $y + x$.

ii) **Abstracting XML from maths**

At this step, we abstract the documents in XML format from the math expressions. This means each maths expression's node in the XML files is replaced by a node containing its XML ID together with its generated new unique ID.

iii) **Abstracting XML from the tags**

All the XML tags are stripped from XML files, leaving the text of the documents with maths IDs instead of the maths expressions themselves.

5. **Data Annotating**

The data is annotated by two expert mathematicians using a text annotation tool. During this stage, the annotators found and labelled the expressions which are defined in the context and determined their definitions. Also, to ensure the reliability of the annotations, the inter-annotator agreement is measured using κ-statistic.

4.1.2 Training Phase

In this phase, we used the labelled corpus, i.e. GSC which is built in the former stage to train a statistical classifier. This stage consists of three steps as follows:

1. Preprocessing the training data by cleaning and preparing it to be fed into the classifier.

2. Extracting some features from the data in the step that we believe has its influence on the prediction that is produced by the classifier.

3. Building the trained model, by learning the classifier from the training data.

4.1.3 Testing Phase

In the testing phase, we used the trained model resulting from the former stage to create a prediction for the testing data. This phase consists of three steps:

1. Preprocessing the testing data to make it ready to be used by the classifier.

2. Predicting the defined maths formulae and their definitions by the classifier on unseen data (testing data).

3. Evaluating the performance of the classifier by using different metrics.

A detailed explanation is shown in Section 6.2.

4.2 Documents Collection and Conversion

The data we used is a collection of maths documents which was randomly collected from the e-prints arXiv [3]. To ease the start of our research, we restricted the maths documents to be in a particular maths domain; Elementary Number Theory, where this restriction can be released afterwards. We collected ten maths documents from different authors (Table 4.1 shows the title, authors' names and the number of pages for each of these documents) with a total of 108 pages (100 without the references part) consisting of 2136 sentences. The data contains a total of 4001 maths formulae with 2569 of them being unique and 441 expressions are explicitly defined in the documents with 396 definitions. We started from the source code of the documents which are in LaTeX format to ease converting the documents into XML format which has the advantage of usability. Using the XML format makes it easy to process the documents as data, obtain information from it and edit it cleanly. The documents in LaTeX format are converted into XML format using the converter LaTeXML [43] (LaTeXML is the LaTeX to XML/HTML/MathML converter which is explained in Section 2.1.2.3).

Table 4.1: The maths documents which are used to build GSC

Document's title	Author	Number of pages	reference
An elemetary proof of an estimate for a number of primes less than the product of the first n primes	Romeo Meštrović	9	[42]
An infinite family of multiplicatively independent bases of number systems in cyclotomic number fields	Manfred Madritsch and Volker Ziegler	10	[38]
Construction of normal numbers via generalized prime power sequences	MG Madritsch and Robert F Tichy	13	[39]
Elementary results on the binary quadratic form $a^2 + ab + b^2$	Umesh P. Nair	11	[45]
Finding Almost Squares II	Tsz Ho Chan	4	[16]
Generalized Brouncker's continued fractions and their logarithmic derivatives	Olga Kushel	17	[35]
On Additive Combinatorics of Permutations of $\mathbb{Z}_n$	Nitin Singh, Deepak Rajendraprasad and L Sunil Chandran	9	[60]
On the Average of Triangular Numbers	Mario Catalani	7	[15]
The Period Length of Euler's Number e	Kurt Girstmair	11	[24]
Updating An Upper Bound Of Erik Westzynthius	Gerhard R. Paseman	17	[49]

4.3 Maths Formulae Extraction

Maths expressions in the XML files that are produced by LaTeXML [43] have a special format which is the presentation of LaTeXML for maths as explained in Section 2.1.2.3. The XML files have been studied carefully, which allow precise maths' extraction. The main issues noted are with the multiline equations. These issues are explained in detail in Section 5.2.

We have two types of extraction as follow:

1. **Extract all maths expressions**

   Using a recursive function based on All_math_extraction algorithm; Algorithm 5.2, which assures there is no duplicated maths. All_math_extraction algorithm is implemented in the Python language, and the extracted maths expressions are saved in XML format which facilitates handling the maths formulae such as accessing and editing. A detailed explanation of All_math_extraction algorithm, its implementation and the format of its result will be discussed in Section 5.2.1.

2. **Extract desired maths expressions**

   MathExtractor allows extraction of chosen maths expressions using a generic select function, Extraction algorithm, which uses XPath algorithms to extract the interesting expressions or subexpressions. Therefore, looking at maths expressions as trees, we can identify and extract maths expressions according to what we defined as an atomic structure which is either at leaf nodes or composite structures. In the Extraction algorithm, a predicate is determining which maths formula is to be extracted by using XPath functions implemented in the Python language, which will be explained in Section 5.2. By using the right predicate, we can obtain specific maths expressions such as accented characters like $\hat{H}$ or subscriptive expressions like P_i. Furthermore, we can extract maths expressions depending on the attributes of their nodes such as obtaining all maths symbols which have specific font or role. For example, using the predicate `//XMTok[text()]` we extracted all single maths symbols, i.e. atomic expressions in the form of leaf nodes; such as Z in the node

   ```
   <XMTok font="blackboard"role="UNKNOWN">Z</XMTok>
   ```

   Another example is the maths formula in Equation 4.1 from [38].

$$\tilde{p}_d \leq \tilde{p}_{d+1} \tag{4.1}$$

   This formula is represented in the XML format; which is produced by LaTeXML, as shown in Figure 4.2. Looking at its tree view in Figure 4.3 we can extract the

atomic $\tilde{p}$; which is in a composite structure format of atomic, i.e. a node that has two leaf children using the predicate:

```
.//*[not(*/*) and count(*) = 2]
```

Also, single symbols such as $\sim$ or p; which are leaf nodes, can be extracted by using the predicates:

```
.//*[not(child::*)]
```

The results of applying these predicates are shown in Figure 4.4 and Figure 4.5; respectively.

4.4 Data Preparation

At this stage, our data is documented in LaTeX, and XML format and as explained in previous sections we have all maths expressions extracted. To be able to annotate this data we need to prepare it in a particular way that serves our need as follows:

1. **Generating a unique ID for each unique maths expression**

   Firstly, the unique maths expressions (i.e. expressions without repetitions as explained in Section 4.1) are determined and then for each of them a unique ID is generated. The new IDs are in the form "math$i$" where i is a unique number for each unique expression and different from its XML ID. Note that the same expression occurring in many positions within a document will have the same generated ID but a different XML ID which allows us to link the similar expressions in later stages.

2. **Abstracting XML from Maths Expressions**

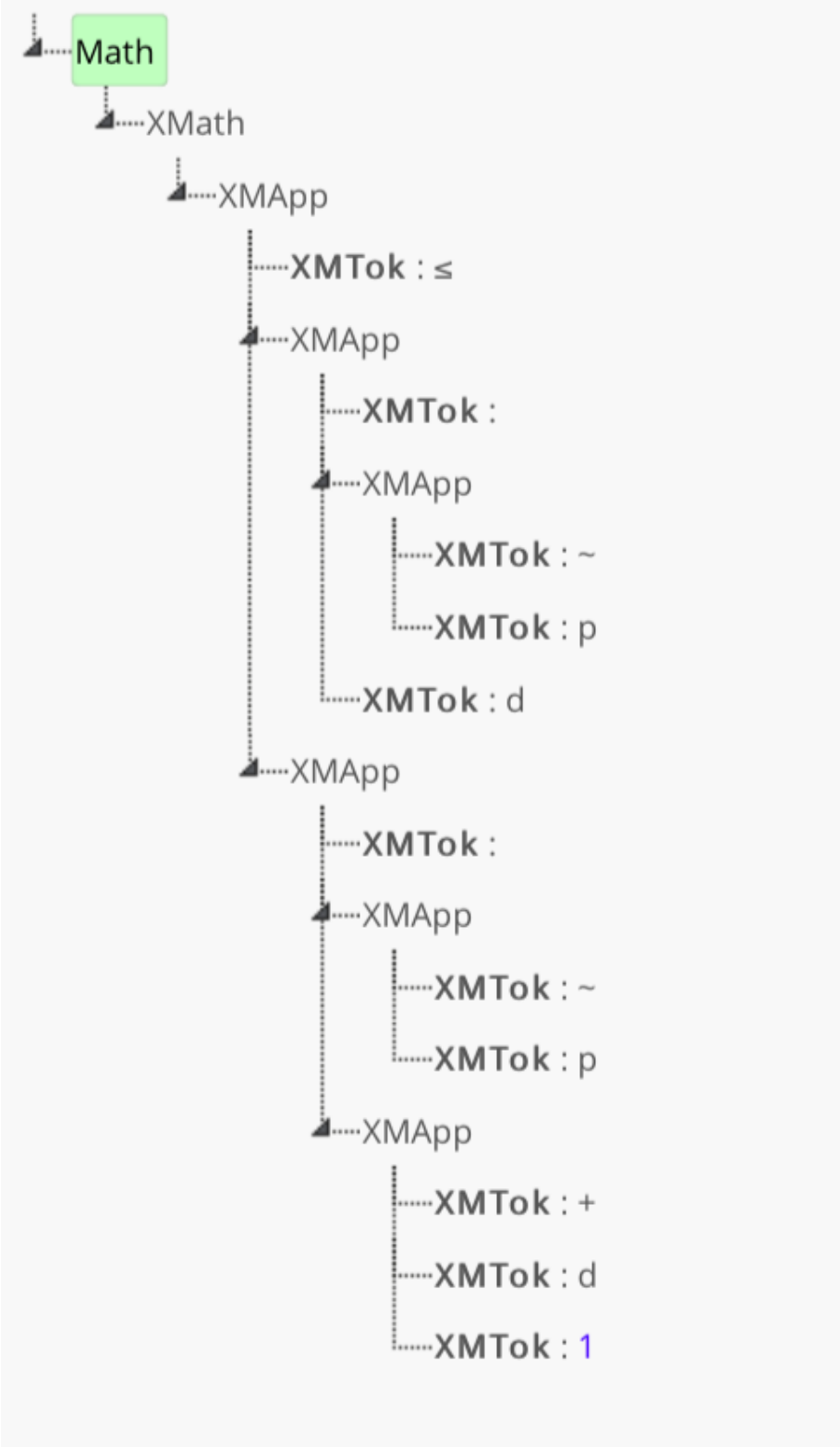
   The abstracting is done by replacing each maths expression with its ID which is a combination of its generated unique ID and its original XML ID in the form of _math$i$-XML ID_.

3. **Abstracting the XML format from its tags**

Figure 4.2: $\tilde{p}_d \leq \tilde{p}_{d+1}$ in LaTeXML representation format

```
<Math mode="inline" xml:id="S2.p2.m8" tex="\tilde{p}_{d}\leq\
   tilde{p}_{d+1}" text="(tilde@(p)) _ d less= (tilde@(p)) _
   (d + 1)">
  <XMath>
    <XMApp>
      <XMTok meaning="less-than-or-equals" name="leq" role="
         RELOP"> ≤</XMTok>
      <XMApp>
        <XMTok role="SUBSCRIPTOP" scriptpos="post2"/>
        <XMApp>
          <XMTok name="tilde" role="OVERACCENT" stretchy="
             false">˜</XMTok>
          <XMTok role="UNKNOWN" font="italic">p</XMTok>
        </XMApp>
        <XMTok role="UNKNOWN" font="italic">d</XMTok>
      </XMApp>
      <XMApp>
        <XMTok role="SUBSCRIPTOP" scriptpos="post2"/>
        <XMApp>
          <XMTok name="tilde" role="OVERACCENT" stretchy="
             false">˜</XMTok>
          <XMTok role="UNKNOWN" font="italic">p</XMTok>
        </XMApp>
        <XMApp>
          <XMTok meaning="plus" role="ADDOP">+</XMTok>
          <XMTok role="UNKNOWN" font="italic">d</XMTok>
          <XMTok meaning="1" role="NUMBER">1</XMTok>
        </XMApp>
      </XMApp>
    </XMApp>
  </XMath>
</Math>
```

Figure 4.3: $\tilde{p}_d \leq \tilde{p}_{d+1}$ in tree view mode



At this step, all the XML tags are stripped so that the data becomes the text of the maths documents with the maths expressions' IDs in place of the maths expressions

Figure 4.4: Results of using XPath predicates on the formula $\tilde{p}_d \leq \tilde{p}_{d+1}$ to select $\tilde{p}$

```
<XMApp>
  <XMTok name="tilde" role="OVERACCENT" stretchy="false">~</
     XMTok>
  <XMTok font="italic" role="UNKNOWN">p</XMTok>
</XMApp>
<XMApp>
  <XMTok name="tilde" role="OVERACCENT" stretchy="false">~</
     XMTok>
  <XMTok font="italic" role="UNKNOWN">p</XMTok>
</XMApp>
```

Figure 4.5: Results of using XPath predicates on the formula $\tilde{p}_d \leq \tilde{p}_{d+1}$ to select all single symbols

```
<XMTok meaning="less-than-or-equals" name="leq" role="RELOP">
    (*$\leq$*)</XMTok>
<XMTok role="SUBSCRIPTOP" scriptpos="post2"/>
<XMTok name="tilde" role="OVERACCENT" stretchy="false">~</
   XMTok>
<XMTok font="italic" role="UNKNOWN">p</XMTok>
<XMTok font="italic" role="UNKNOWN">d</XMTok>
<XMTok role="SUBSCRIPTOP" scriptpos="post2"/>
<XMTok name="tilde" role="OVERACCENT" stretchy="false">~</
   XMTok>
<XMTok font="italic" role="UNKNOWN">p</XMTok>
<XMTok meaning="plus" role="ADDOP">+</XMTok>
<XMTok font="italic" role="UNKNOWN">d</XMTok>
<XMTok meaning="1" role="NUMBER">1</XMTok>
```

themselves. Therefore, these resulting documents contain text only without any maths expressions which form the corpus to be annotated as will be explained in the following section.

For example, applying these three steps to the definition "Let $\mathbb{Z}_n$ denote the ring of integers modulo n." [60] transform it into "Let _math35-m2_ denote the ring of integers modulo _math24-m3_." as shown in Table 4.2. The implementation of the preparation steps is discussed in detail in Section 5.3.

4.5 Data Annotation

The annotation step is about tagging each word in the data with the appropriate category from a predetermined tagset to build a gold standard corpus. The aim is to determine both the maths expressions that are defined in the documents and their definitions. This task has been done by two expert mathematicians using the annotation tool brat [62] which is explained in Section 2.2.4.

There are two types of definitions:

- The standard way of defining an object and the most common one where either the expression is mentioned first followed by the definition or the other way around as in "Let $\mathbb{Z}_n$ denote the ring of integers modulo n." and "define an integer N ...".
- The second type is where the definition starts before the expression and finishes after it as in the definition "a subspace S of V" where the expression S is defined as a subspace of V. We call such definitions 'divided definitions'.

In our data, there are 396 explicit definitions of maths formulae; 368 of them are of the first type and 28 of them are of the second type.

This annotation task is a multi-class classification problem containing a one-level tagset as shown in Table 4.3. The tag "exp" is used to annotate each defined maths expression, and the tag "def" is used to annotate definitions. However, in the case of the second type

Table 4.2: Preparing documents for annotation, example from [60]

PDF	Let $\mathbb{Z}_n$ denote the ring of integers modulo n.
Original XML	Let <Math mode="inline" xml:id="m2" tex="\mathbb{Z}_{n}" text="Z _ n"> <XMath> <XMApp> <XMTok role="SUBSCRIPTOP" scriptpos="post3"/> <XMTok role="UNKNOWN" font="blackboard">Z</XMTok> <XMTok role="UNKNOWN" font="italic">n</XMTok> </XMApp> </XMath> </Math> denote the ring of integers modulo <Math mode="inline" xml:id="m3" tex="n" text="n"> <XMath> <XMTok role="UNKNOWN" font="italic">n</XMTok> </XMath> </Math>.
Generated unique ID	ID "math35" for the expression: <Math mode="inline" xml:id="m2" tex= ... </Math> ID "math24" for the expression: <Math mode="inline" xml:id="m3" tex= ... </Math>
Combination of unique ID with XML ID	_math35-m2_ _math24-m3_
Abstract XML from maths	Let <expression>_math35-m2_</expression> denote the ring of integers modulo <expression>_math24-m3_ </expression>.
Strip XML tags	Let _math35-m2_ denote the ring of integers modulo _math24-m3_.

of definitions, the tag "p1D" is used to annotate the first part of the definition that occurs before the defined maths expression while "def" is used for the rest of the definition that occurs after the expression.

There are three relations between the tags as shown in Table 4.4.

For example, the definition "a subspace S of V" [60] is annotated as shown in Figure 4.6, where each maths expression is replaced with its ID as explained in the former section (i.e. the expressions S and V are represented by their IDs; _math11-S2.Thmdefn2.p1.m13_ and _math58-S2.Thmdefn2.p1.m14_, respectively).

Another example is the definition in "Let $\mathbb{Z}_n$ denote the ring of integers modulo n." [60] which is illustrated in Table 4.2. This sentence is annotated in brat tool as shown in Figure 4.7.

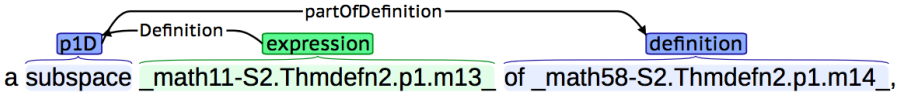
Table 4.3: The one-level tagset

Entities	Label
expression	exp
definition	def
part1Definition	p1D

Table 4.4: The relations between entities

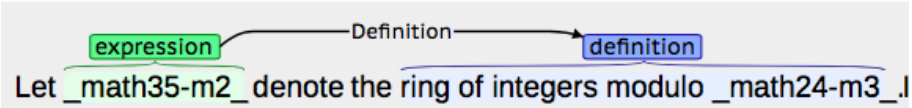
Relations	Arg1	Arg2
definition	exp	def
definition	exp	p1D
partOfDefinition	p1D	def

Figure 4.6: Example of brat annotation for the divided definitions "a subspace S of V" [60]



However, brat produces an annotation file for each annotated document which contains the entities and relations and their positions in the text. The annotation information for the example in Table 4.2 is as follows:

Figure 4.7: Example of annotation on brat of the first type of definition "Let $\mathbb{Z}_n$ denote the ring of integers modulo n." [60]



T1 expression 86 97 _math35-m2_

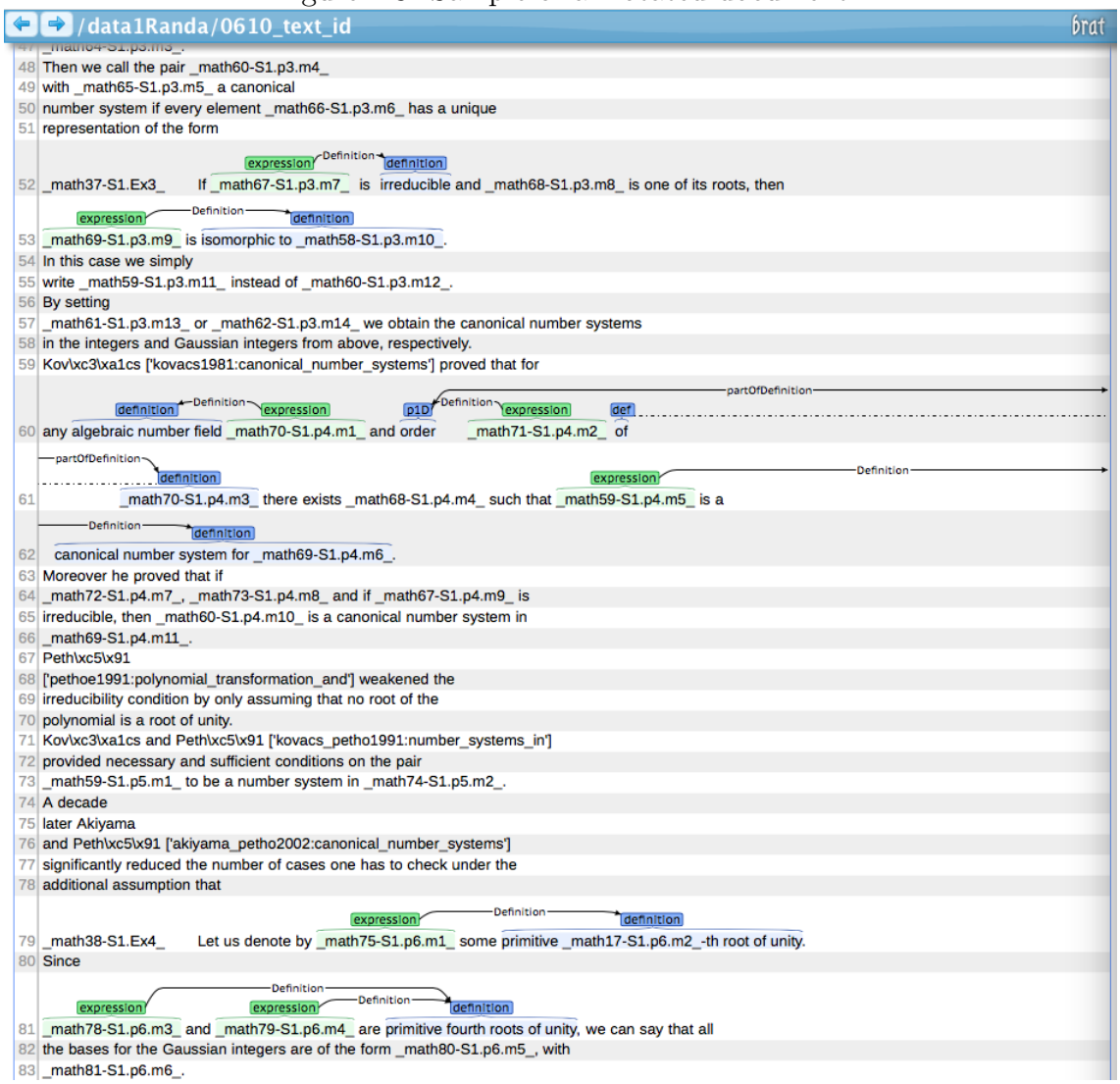
T2 definition 109 144 ring of integers modulo _math24-m3_

R1 Definition Arg1:T1 Arg2:T2

Initially, the annotators were provided with guidelines that stated precisely the attributes of different classes. They started with a training session where they annotated only 10% of the documents to be able to observe the restraint that could occur during the annotation process. Following that, they annotated the rest of the papers. Figure 4.8 shows part of the annotation of the article [38].

The inter-annotator agreement is evaluated using kappa statistic (κ-statistic) to assure the level of annotation reliability. The calculated metric is $\kappa = 0.9334$ which is high enough to proceed with the annotation.

Figure 4.8: Sample of annotated document



4.6 Chapter Summary

In this chapter, we provided an overview of our system architecture. Also, we demonstrated our approach to build our gold standard corpus starting from the source code of mathematical documents in LaTeX format then converted them into XML format. We extracted all maths expressions, determined the unique ones among them and generated a unique ID for each one. Then the XML format of documents was abstracted from maths expressions followed by stripping all XML tags which produced the text files that were ready to be annotated. Subsequently, two expert mathematicians annotated the data by finding the maths expressions that are defined in documents and determined their definitions. Moreover, κ-statistic is used to evaluate the inter-annotator agreement which showed that the annotation is reliable. Therefore, our GSC was built and ready to be used by a probabilistic model.

CHAPTER 5

IMPLEMENTATION

The architecture diagram in Figure 4.1 shows the principal segments of our system which are three phases; building a gold standard corpus, the training phase and the testing phase. In this chapter, the implementation of three steps of the first phase is discussed; converting the LaTeX format of maths documents into XML, extracting maths expressions and preparing the data to be annotated.

We used the pipeline architecture where we have some consecutive stages. One step's output is the input for the following step. All the implementation has been done by using the functional programming language Python.

However, the evaluation of the implemented algorithms will be discussed in Chapter 8.

5.1 Converting LaTeX into XML using LaTeXML

There are two ways to convert LaTeX format into XML format using LaTeXML:

- Using maths expression:

  The first option is transforming a TeX/LaTeX maths expression into various formats such as presentation MathML, content MathML, openMath and LaTeXML's internal format. However, this option is not useful for our task as it does not read files but only accepts maths expressions in LaTeX format.

- Using file in LaTeX format:

This option transforms a TeX/LaTeX file into XML file which has the maths formulae in the LaTeXML's internal format and it is the one that is used for converting our text files. We use it as

latexml –output=outputFile.xml inputFile.tex

A sample of the XML resulting file is shown in Appendix B

5.2 Maths Formulae Extraction

The XML files contain maths formulae in the LaTeXML's internal format. To be able to extract maths precisely we studied these XML files especially the maths format. Therefore, we noticed that the multi lines equations are written using align environment in the LaTeX files and as equationgroup node in the XML files. The equationgroup node format is shown in Figure 5.1 and we have two issues with it.

- **The children nodes of equationgroup node:**

  An equationgroup node contains an equation node which contains a Math node for each line of the original equation. This means that we do not want to extract the Math node only or the equation node only as this will result in having each line of the equation as a standalone maths formula among the extracted formulae and not having the whole equation together. Instead, we extracted the whole equationgroup as one maths expression without diving into it and extracting sub expressions separately to prevent repetition. This issue is considered when designing the extraction algorithm.

- **MathBranch nodes:**

  An equationgroup node contains an equation node for each line of the original equation which includes a MathBranch node. The MathBranch node contains two children nodes; one for each side of the current equation. We chose to remove the MathBranch nodes from the children of equationgroup nodes as they are not needed and have no different information but duplication. To delete the mathBranch nodes

Figure 5.1: Format of equationgroup node in LaTeXML representation

```
<equationgroup>
  <equation refnum="2.1" xml:id="S2.E1">
    <MathFork>
      <Math ...>
        ⋮
      </Math>
      <MathBranch>
        <td align="right">
          <default:Math ...>
            ⋮
          </default:Math>
        </td>
        <td align="left">
          <default:Math ...>
            ⋮
          </default:Math>
        </td>
      </MathBranch>
    </MathFork>
  </equation>
  <equation xml:id="S2.Ex1">
    ⋮
  </equation>
  <equation xml:id="S2.Ex2">
    ⋮
  </equation>
</equationgroup>
```

from equationgroup nodes, we used our MathBranch_elimination algorithm Algorithm 5.1.

Algorithm 5.1: MathBranch_elimination

Input:

XML file inF

Output:

XML file F

Method:

```
1    let S = set of all nodes with tag = 'equationgroup'
2      foreach N ∈ S do
3        foreach C ∈ childNodes(N) do
4          if tag(C) == 'MathBranch' then
5            delete C
6          end
7        done
8      done
```

As explained in Section 4.3, we have two type of extraction: extraction of all maths expressions and extraction of desired maths expressions which will be discussed in the following two subsections.

5.2.1 Extract all Maths Expressions

The extraction of all maths expressions has been done by using a recursive algorithm; All_math_extraction algorithm, which is shown in Algorithm 5.2. Using this algorithm assures that there are no duplicated maths extracted. Our implementation saved maths expressions into XML files such that each file contains all maths extracted from one document of our data. Each created XML file has a root node with the document's name and an xmlID node for each maths expression of this document. By saving maths

Figure 5.2: Example of the format of the XML file resulting from implementing All_math_extraction algorithm

```
<file_name fileName="0109">
  <xmlID idName="A0.EGx1">
    <equationgroup ...>
      ⋮
    </equationgroup>
  </xmlID>
  <xmlID idName="I1.i3.p1.m12">
     <Math ...>
       ⋮
     </Math>
  </xmlID>
    ⋮
</file_name>
```

expressions in XML format we make it easy to deal with them such as accessing or editing them. An example of this format is shown in Figure 5.2.

Algorithm 5.2: All_math_extraction

Input:

XML file inF

Output:

XML file F

Method:

```
1     let inR = inF.root
2     let R = F.root
3     let S = set of all inR.childNodes
4       foreach C ∈ S do
5         if tag(C) == 'equationgroup' or 'equation' or 'Math' then
6           append C to R
7         elif childNodes(C) ≠ null then
8           let S2 = set of all C.childNodes
```

```
9              foreach L ∈ S2 do
10                 go to step 4
11             done
12         end
13     done
```

5.2.2 Extract Desired Maths Expressions

MathExtractor is a tool we built using a generic select function to allow extraction of chosen maths expressions. In this function a predicate is determining which maths formula is to be extracted taking advantage of the XPath functions; see Algorithm 5.3. XPath functions travel through the XML nodes using the predicate as a path to the targeted nodes. Using MathExtractor gives us a choice to determine the desired maths expression such as all single symbols, atomic expressions or even expressions with a particular attribute like font or role. Thus, using the right predicates allow us to extract specific maths expressions. For instance, for the math expression "$\mathcal{S}(\mathbb{Z}_n)$" [60] which has an XML representation as shown in Figure 5.3 we can extract the atomic expression $\mathbb{Z}_n$ using the predicate:

```
.//*[not(*/*) and count(*) = 3]
```

which selects nodes shown in Figure 5.4.

Algorithm 5.3: Extracting desired maths expressions

Input:

L : List of Math nodes

X : XPath expression

Output:

E: Node List of Math expressions

Method:

```
1   let E = list []
```

Figure 5.3: $\mathcal{S}(\mathbb{Z}_n)$ in LaTeXML representation format

```
<Math mode="inline" xml:id="S1.p1.m7" tex="{\cal S}(\mathbb{Z
   }_{n})" text="S * Z _ n">
  <XMath>
    <XMApp>
      <XMTok meaning="times" role="MULOP"></XMTok>
      <XMTok possibleFunction="yes" role="UNKNOWN" font="
         caligraphic">S</XMTok>
      <XMApp close=")" open="(">
        <XMTok role="SUBSCRIPTOP" scriptpos="post2"/>
        <XMTok role="UNKNOWN" font="blackboard">Z</XMTok>
        <XMTok role="UNKNOWN" font="italic">n</XMTok>
      </XMApp>
    </XMApp>
  </XMath>
</Math>
```

Figure 5.4: Results of using XPath predicates on the formula $\mathcal{S}(\mathbb{Z}_n)$ to select $\mathbb{Z}_n$ in XML format

```
<XMApp close=")" open="(">
  <XMTok role="SUBSCRIPTOP" scriptpos="post2"/>
  <XMTok font="blackboard" role="UNKNOWN">Z</XMTok>
  <XMTok font="italic" role="UNKNOWN">n</XMTok>
</XMApp>
```

```
2      foreach C ∈ XPath.eval(X) do
3        append C to E
4      end
5    return E
```

5.3 Data Preparation

There are several steps involved in preparing our data to be annotated. For each document we have the LaTeX format, the XML format and all maths expressions extracted and stored in XML format. Preparation of the data to be annotated using a text annotation tool

involves three steps as follows:

1. Generating a unique ID for each unique maths expression. By unique expressions, we refer to the expressions with no repetitions as explained in Section 4.1. Note that some expressions are similar; nonetheless, they have different unique IDs due to the different fonts specifications. For example:

   `<XMTok role="UNKNOWN">n</XMTok>`

   and

   `<XMTok font="italic"role="UNKNOWN">n</XMTok>`

   are similar expressions, but the first node is embedded in text that is all in italic font.

   The implementation of this step requires determining the unique expressions first then generating a new unique ID for each one of them. For this task, we implemented our Uniqueness_new_ID algorithm; Algorithm 5.4, in Python.

Algorithm 5.4: Uniqueness_new_ID

Input:

XML file inF

Output:

XML file F

Method:

```
1     let dic = dictionary {}, i = 1, frequency = 0
2     let list L = all expressions in document D
3     let s = length of L
4     foreach e1 ∈ L do
5       let p = index(e1)
6       let id1 = xmlID(e1)
7       foreach j in range (p+1, s) do
8         if e2 ∈ L and index(e2) = j then
9           let id2 = xmlID(e2)
```

```
10          compare e1 and e2
11          if e1 == e2 then
12            if e1 ∈ dic then
13              dic[e1](frequency) = frequency + 2
14              append id1 and id2 to dic[e1]
15            else
16              i = i + 1
17              frequency = 2
18              append (e1, 'math-i', id1, id2, frequency) to dic
19            end
20          elif e1 ≠ e2 then
21            if e1 ∈ dic then
22              dic[e1](frequency) = frequency + 1
23              append id1 to dic[e1]
24            else
25              i = i + 1
26              frequency = 1
27              append (e1, 'math-i', id1, frequency) to dic
28            end
29            if e2 ∈ dic then
30              dic[e2](frequency) = frequency + 1
31              add id2 to value dic[e2]
32            else
33              i = i + 1
34              frequency = 1
35              append (e2, 'math-i', id2, frequency) to dic
36            end
37          end
```

```
38            end
39          done
40        done
```

However, step number 10 in the Uniqueness_new_ID algorithm (Algorithm 5.4) is a function that compares two expressions, i.e. two XML nodes. Obviously, this comparison does not include the attributes xmlID as this is unique for each maths expression. Yet the comparison consists of comparing nodes' type, tag, depth, attributes except xmlID, text and nodes' children.

2. Abstracting XML from Maths Expressions. In this step, we replace maths expressions with their ID in the XML files. Maths ID here refers to a combination of the generated unique ID and the original XML ID. This is a straightforward task done by implementing our algorithm XML_Doc_no_Math; Algorithm 5.5, in Python.

Algorithm 5.5: Abstracting XML from Maths Expressions

Input:

XML file inF

Output:

XML file F

Method:

```
1     let EG = all nodes with tag = 'equationgroup'
2     let E = all nodes with tag = 'equation'
3     let M = all nodes with tag = 'Math'
4     foreach eg ∈ EG do
5       if childNodes(eg) ≠ null
6         delete childNodes(eg)
7         if 'xmlID' ∈ eg[attribute]  then
8           let att = eg[attribute = 'xmlID']
9         end
```

```
10          delete eg[attribute]
11          let tag(eg) = 'expression'
12          append att to eg[text]
13        end
14      done
15      foreach e ∈ E do
16        repeat steps 5 to 13
17      done
18      foreach m ∈ M do
19        repeat steps 5 to 13
20      done
```

3. Abstracting XML from the tags. This step is done by stripping all the XML tags which leads to the text of the documents with the maths ID being in place of the maths expressions. However, we notice that the reference nodes in our XML data look like: <ref labelref="LABEL:C"/> and by stripping the XML tags we will lose such references. Therefore; to keep such reference in our data we edited the reference nodes before stripping the tags by appending the value of attributes labelref and bibrefs to the node's text which makes them look like: <ref labelref="LABEL:C">"LABEL:C"</ref>.

5.4 Chapter Summary

In this chapter, we discussed our methodology for implementing part of our system and some important science aspects that related to the implementation. These discussions included the stages of converting LaTeX into XML using LaTeXML, the stage of extracting maths formulae which consist of extracting all maths formulae and the robust extraction of desired maths expressions in addition to the stage of preparing the data for the annotating step. We adopted the pipeline architecture for implementation; where we have some consecutive stages and the output of one stage is the input for the following one.

Part III

Mathematical Semantics Recognition

CHAPTER 6

EXTRACTING SEMANTICS OF FORMULAE

In this chapter, we will present our approach for extracting the semantic information of maths formulae from two different sources; the representation of maths formulae and maths context. In Section 6.1 we will discuss our approach for obtaining the basic semantic information for maths formulae from its representation; which is the internal representation of LaTeXML. Section 6.2 will demonstrate extracting maths semantic information from their context using supervised machine learning (SML) techniques. This includes addressing the preprocessing step, the dataset and the feature selection approach. In addition, in Section 6.2.4 we will develop a baseline model based on the MaxEnt classifier. In Section 6.2.5, we will learn different classifiers such as MEMM and CRF. The effect of using different features with the CRF will be discussed in Section 6.2.6. Finally, Section 6.2.7 will explore the influence of using a hybrid approach by injecting rule-based features into the statistical model. The evaluation of all the experiments will be discussed in Chapter 8.

6.1 Basic Semantic Information in the Representation of Maths Formulae

One of our aims is to extract the primary semantic information for each maths expression from its representation. As we used LaTeXML [43] to convert our source code of the

documents into XML format, we have the maths expressions in the representation of LaTeXML which is explained in Section 2.1.2.3. By studying the XML format of our documents, we found that the required semantic information for each maths expression is stored in the attributes of that expression's node. The attributes may specify different aspects of maths expressions such as font, maths style and the syntactic and semantic roles of them. Our targeted basic semantic information included in the attributes 'meaning', 'possibleFunction' and 'role'; although 'role' could be 'UNKNOWN' if LaTeXML could not categorise the symbol's role, Appendix C shows a sample of these attributes. Indeed these attributes' values are not always reliable, but it is of interest to compare and combine them with the definitions extracted from the context. Therefore, we extracted such semantic information from the nodes' attributes of maths expressions using the MathExtractor tool which is addressed in Section 4.3.

6.2 Semantic Information in the Maths Context

In this section, we will demonstrate our methodology to extract the semantic information of maths formulae from their context using SML techniques. In Chapter 4 we presented our approach for building a GSC which will serve as the data for the statistical learning algorithms. We will rely on the system architecture that presented in Figure 4.1 to develop the probabilistic model.

For sequence labelling task, we depend on a rapid discriminative toolkit called Wapiti [37] which has MaxEnt, MEMM and CRF statistical learning algorithms implemented and enclosed with variance optimisation algorithms. Wapiti is advocated over other toolkits for its fastness, and it has been used in several sequence labelling tasks such as in [11] and [48].

6.2.1 Preprocessing

The gold standard corpus (GSC) that built as explained in Chapter 4 is used in this step. As the annotated documents resulted from stripping XML tags from the XML format of the documents, our data was not correctly formatted. Therefore, we ensured that the labelled data is in a format that is easy to deal with in the later stages by preprocessing it as follows:

1. Data Cleansing: This is a crucial step in which we cleaned and prepared our GSC for the processing step. For instance, some sentences were split by needless empty lines and some other sentences were concatenated without a space between them. Therefore, we looked into the data and edited it to assure that it is correctly formatted.

2. Conversion of the text data into CoNLL format: The CoNLL structure so it can be processed by the classifier; which is a single word (i.e. token) per line followed by a space separation then the value of the annotation tag and sentences are separated by empty lines. For instance, for the sentence "Let $\mathbb{Z}_n$ denote the ring of integers modulo n." [60]; the LaTeX, XML, the abstracted text format, the annotation result and the CoNLL format are shown in Table 6.1. The first four formats resulted from previous steps that are discussed in Chapter 4 and Chapter 5. In the CoNLL format, B-expression and B-definition refer to the beginning of the expression and definition, respectively. Where I-definition refers to the rest of the definition.

6.2.2 Dataset

It is a prevalent method in the literature to split the data equally into training and testing. However, one of the problems with this approach is overfitting which results in poor performance of the classifier, i.e. the trained model is poorly generalised. Overfitting occurs whenever excessive features are used in particular if the dataset is relatively small [10]. To

Table 6.1: Example from the data demonstrating the steps from the input until CoNLL format

PDF	Let $\mathbb{Z}_n$ denote the ring of integers modulo n.
LaTeX	Let $\mathbb{Z}_n$ denote the ring of integers modulo n.

XML	`Let`
	`<Math mode="inline" xml:id="m2" tex="\mathbb{Z}_{n}"`
	`text="Z _ n">`
	`<XMath>`
	`<XMApp>`
	`<XMTok role="SUBSCRIPTOP" scriptpos="post3"/>`
	`<XMTok role="UNKNOWN" font="blackboard">Z</XMTok>`
	`<XMTok role="UNKNOWN" font="italic">n</XMTok>`
	`</XMApp>`
	`</XMath>`
	`</Math>`
	`denote the ring of integers modulo`
	`<Math mode="inline" xml:id="m3" tex="n" text="n">`
	`<XMath>`
	`<XMTok role="UNKNOWN" font="italic">n</XMTok>`
	`</XMath>`
	`</Math>.`

Abstracted text	Let _math35-m2_ denote the ring of integers modulo _math24-m3_.
Annotation results	T1 expression 86 97 _math35-m2_ T2 definition 109 144 ring of integers modulo _math24-m3_ R1 Definition Arg1:T1 Arg2:T2

CoNLL	`Let O`
	`_math35-m2_ B-expression`
	`denote O`
	`the O`
	`ring B-definition`
	`of I-definition`
	`integers I-definition`
	`modulo I-definition`
	`_math24-m3_ I-definition`
	`. O`

overcome the problem of overfitting, the K-fold cross-validation technique is used, where the dataset is divided into k subsets; k = 10. Every time, one of the k folds is used as testing data and the remaining nine are used as training data. Therefore we trained and tested the classifier k times (i.e. ten times). The conventional way to do this is by randomly dividing the dataset into ten equivalent folds. However, this task is slightly tricky as the division could be expected to occur in the middle of a sentence. In this case, the division took place at the end of that sentence. This dataset is used in all subsequent experiments that discussed in this chapter.

6.2.3 Features Selection

The used features are a significant factor that influences the outcome of sequence labelling algorithms regarding accuracy and reliability. Using all the extracted features may result in noisy data that causing the statistical algorithms to perform deficiently. Therefore, selected features are employed in this task. For feature selection, we used the Stepwise Regression technique where features are added or removed consecutively up to the point where the resulting prediction is not improved any more.

6.2.4 Baseline Model Based on Maximum Entropy

Since there is no equivalent work done using the same dataset as the one used in this research and to ensure the conducting sound comparison experiments, a baseline model is developed using MaxEnt in combination with the Rprop$^+$ optimisation algorithm to function as a bottom line performance to evaluate our approach. The MaxEnt model is chosen as it is commonly employed in IE [44]. It also does not require a long time or a large memory to run.

6.2.4.1 Features Extraction

The used features influence the functionality of the machine learning algorithms. Superb functionality of the learning algorithms resulted when they are integrated with features

that provide useful information.

Contextual features, i.e. the features associated with the context are used for the baseline model. The contextual features refer to the window size of the neighbouring tokens (i.e. words) and the position of the token in a sentence. We used the window size features; where we consider the current token, before and after tokens in a particular range. For instance, considering the window size -/+1 means if we let the current word W_i, then the window of one word in two directions (before and after) would be $W_i - 1$, W_i, $W_i + 1$. For the baseline model, the basic unigram features with the window of two words in two directions (-/+ 2) are selected according to the Stepwise Regression technique as discussed in Section 6.2.3.

6.2.4.2 Results of the Baseline Model

The results of training the MaxEnt while using the optimisation algorithm Rprop$^+$ and the selected basic features predict maths definitions with average metrics as presented in Table 6.2. The results show that the average precision is about 41% whereas the average recall is about half of it which results in a low average F-measure, 27.16%.

Table 6.2: Average metrics (%) for the results of the baseline model based on MaxEnt combined with the Rprop$^+$

A	P	R	F
94.6	41.27	20.76	27.16

6.2.5 Using MEMM and CRF as Different Classifiers

One of the important aspects that influence IE systems is the used probabilistic model. As the efficiency of the MEMM model is better than the MaxEnt's on text-related IE applications [41] and the fact that the CRF is the state-of-the-art for sequence labelling tasks [36], we experimented to learn the best practice which can be employed into our task. In this experiment, the MEMM and CRF discriminative sequence labelling models

in combination with the Rprop$^+$ and Rprop$^-$ optimisation algorithms are used with the identical features and strategy that used in the baseline model. Moreover, the MaxEnt combined with the Rprop$^-$ is used under the same conditions.

Table 6.3 presents the results of these experiments and show the baseline in green colure. The MEMM performed better than the MaxEnt, yet the CRF performance is the best. However, using the optimisation algorithm Rprop$^-$ improved the performance over the Rprop$^+$ as the results of F-measure shows. It is noticeable that generally, the recall scores are very low in comparison with the precisions' scores which exhibit that the used classifiers are very selective and missing lots of definitions; nevertheless, most of the recognised definitions are correct definitions. The best record for the F-measure is achieved by CRF when combined with Rprop$^-$ as F = 33.8% which is better than the result of the baseline model by 6.64%.

Table 6.3: The results of the CRF, MEMM and MaxEnt in combination with the Rprop$^+$ and Rprop$^-$ using the same features as used in the baseline model

Classifier	Metric (%)	Optimisation Algorithm	
		Rprop$^+$	Rprop$^-$
MaxEnt	A	94.6	94.62
	R	20.76	20.55
	P	41.27	42.96
	F	27.16	27.33
MEMM	A	94.65	94.81
	R	19.76	20.74
	P	67.94	65.96
	F	29.95	31.02
CRF	A	94.66	94.85
	R	22.19	23.06
	P	66.72	71.46
	F	32.62	33.8

6.2.6 Using Different Features with the CRF

The Conditional Random Fields (CRF) when combined with the Rprop$^-$ performed the best in the previous experiment. Therefore; we conduct a set of experiments and play

with the features involved which are the contextual window size.

Table 6.4 presents the results of these experiments which shows that the best performance with F = 38.36% when using a -/+ 1 window size. Moreover, increasing the window size is resulting in a more mediocre performance of the classifier.

Table 6.4: The results of the CRF in combination with the Rprop$^-$ using different features

window size	F-measure (%)
Unigram (-/+ 1)	38.36
Unigram (-/+ 2)	35.84
Unigram (-/+ 3)	23.67
Unigram (-/+ 1), Bigram (-/+ 2)	36.32
Unigram (-/+ 1), Bigram (-/+ 3)	34.21
Unigram (-/+ 1), Bigram (-/+ 2), Trigram (-/+ 1)	33.72
Unigram (-/+ 1), Bigram (-/+ 2), Trigram (-/+ 3)	31.86
Unigram (-/+ 2), Bigram (-/+ 5), Trigram (-/+ 3)	29.16

6.2.7 Exploiting Hybrid Approach

A hybrid approach is used in this experiment by injecting rule-based features into the statistical model that used in the previous experiment; which used the CRF with the Rprop$^-$ and the contextual window size features, aiming to gain the usefulness of both approaches.

The rule-based features refer to definitions' templates which are extracted by expert mathematicians (the annotators) as they studied and analysed the context of maths formulae within our data.

6.2.7.1 Templates' Morphological Features

We detected different definitions' templates in the data and used them as features that enrich the gold standard corpus. The total of 33 templates are detected in the data, some of them are used more frequently than the others. Table 6.6 shows these templates and their frequency in our data. The most popular template is "def exp" as in "... vectors

$u, v \in V \ldots$" which is used 108 times among the total of 396 definitions. The second most frequent template is "let exp be def" as in "Let V be an n-dimensional vector space over the field F" which is used 77 times within the data. Nevertheless, there are 12 templates that each is used only once in the whole corpus.

The extracted features are fed into the corpus as an extra column per feature inserted between the words' column and the value of the annotation tag's column. However, not all the detected templates were used as features. In the step of features selection (explained in Section 6.2.3), we recognised the desirable features which led to the best prediction. The recognised definitions' templates and some examples of them from the data are shown in Appendix A. For example, the sentence "Let $\mathbb{Z}_n$ denote the ring of integers modulo n." [60] is used in Table 6.5 to show a fragment of the annotated corpus after it is enriched with features. However, in this table Temp1 (tl1) refers to the first template in the features' set; which is "Let exp denote def", and Exp refers to the expressions which are defined in the context. One of the substantial contributions of this thesis is advocating some definitions' templates as novel features to improve context classification and therefore maths understanding.

Table 6.5: An example of annotated text after enriched with features

Token	Temp1	Exp	Tag
Let	tl1	O	O
math35-m2	tl1	exp	B-expression
denote	tl1	O	O
the	tl1	O	O
ring	tl1	O	B-definition
of	tl1	O	I-definition
integers	tl1	O	I-definition
modulo	tl1	O	I-definition
math24-m3	tl1	O	I-definition
.	O	O	O

Table 6.6: Template's frequency

Template	abbreviation	Frequency
def exp	td1	108
let exp be def	tl3	77
exp is/are/be def	te1	46
where exp is/are def	te72	38
for def exp	td33	24
exp def	te2	22
let exp and/with exp be def	tl4	12
let exp denote def	tl1	8
def exp for/of def	td2	8
permutation exp of/in exp	t2	6
exp denotes def	te8	6
subspace exp of exp	t3	4
denoting by exp def	te12	4
when exp is/are def	te71	4
with def exp	td32	3
collection exp of . . . exp	t4	3
assume (that) exp is/are def	te73	3
def is denoted by exp	td4	2
consider def exp	td35	2
exp stands for def	te9	2
subset exp of . . . exp	t1	2
set exp to be def	te11	1
exp as def	te3	1
with exp and exp as def and def respectively	te6	1
let us call exp as def	tl2	1
relation exp on def	t5	1
use exp for def	te13	1
factors exp of def	t6	1
define exp to be def	te5	1
define exp as def	te4	1
suppose (that) exp is/are def	te74	1
let exp stand for def	tl6	1
let def be denoted by exp	tl5	1
	Total	396

6.2.7.2 Results of the Hybrid Approach

In this experiment, we investigated combining the rule-based features (i.e. templates' morphological features) and the contextual window size features in one approach to gain

the greatest advantages from both models. The best results obtained when involving the following window size: unigram -/+ 2, bigram -/+ 5 and trigram -/+ 3 in combination with some of the extracted definitions' templates. It is noticeable that some of the templates are not beneficial to the classifier but creating more noise. The best results accomplish when employing some of the most frequent definitions' templates presented in Table 6.7, which shows a significant improvement in the averages of both recall and precession which enhance the F-measure score.

Table 6.7: The results of the hybrid approach using window-based and templates' morphological features

A	P	R	F
97.27	94.21	71.55	81.10

6.3 Chapter Summary

In this chapter, we illustrated the methodology for extracting the semantic information from the representation of maths formulae. On the other hand, we demonstrated different approaches to extract mathematical semantic information from their context using SML techniques. We discussed learning the MaxEnt probabilistic model in combination with the Rprop^{+} optimisation algorithm to develop a baseline model using the contextual window size features. Subsequently, we learnt different probabilistic models; CRF and MEMM in combination with the Rprop^{+} and Rprop^{-} optimisation algorithms. However, as the CRF combined with the Rprop^{-} performed the best over the other classifiers, a further experiment was conducted to investigate the impact of different window size features on its performance. Ultimately, in a hybrid approach, we explore the influence of injecting rule-based features; i.e. definitions' templates, into the statistical model. Chapter 8 will discuss the evaluation of all the experiments that demonstrated in this chapter.

CHAPTER 7

VARIATIONS OF THE APPROACH TO EXTRACT SEMANTICS OF FORMULAE

In this chapter, we will demonstrate our approaches to improve the extraction of maths definitions from their context by enhancing the classifier's performance. Thus, we will present the conducting a series of experiments using the same scheme as the used in the hybrid approach (see Section 6.2.7); i.e. CRF statistical model based on the Rprop$^-$ optimisation algorithm, the contextual window size features and the rule-based features (definitions' templates). Section 7.1 will present the experiment of removing the stop words from our data. Section 7.2 will describe the experiment of extending the used tagset. Lastly, the effect of adding part of speech tags as features will be discussed in Section 7.3. The detailed evaluation of the three experiments will be discussed in Chapter 8.

7.1 The Experiment of Removing the Stop Words

In this section, we investigate the performance of our hybrid approach when removing the words that are very frequently used (stop words) such as: 'a', 'an' and 'the' from the data. In natural language processing research, there is no comprehensive record of stop words; hence, we relied on the Natural Language Toolkit (NLTK) list of stop words [10]. Therefore, the stop words are eliminated from both testing and training data. Then the exact scheme and features that used in Section 6.2.7 are used.

The results of this experiment as shown in Table 7.1 is boosted over the results when

having the stop words, with 1.7% and 0.61% increase in the average of both precision and recall, respectively. Thus, 1.01% improvement in the F-measure average is reported.

Table 7.1: The results of the CRF classifier when removing the stop words from the data

A	P	R	F
96.92	95.91	72.16	82.11

7.2 Extending the Annotation's tagset

In Section 4.5, we presented the tagset that utilised in our approach and explained the use of 'p1D' tag in tagging the divided definitions, which are the definitions that starting before the defined maths formulae and ending after them, i.e. when the defined maths expression is embedded in its definition as the example in Figure 4.6 shows. In this case, the definition is divided by the formula into two parts; the first part which is before the expression and tagged with 'p1D' (which means part one of the definition) and the second part that comes after the expression and tagged with 'definition'.

In this section, we examine the influence of extending the tagset to include a tag named 'p2D', for the second part of such definitions, on the performance of the hybrid approach that discussed in Section 6.2.7. Therefore, the targeted type of definitions is re-annotated using the extended tagset (that include the tag 'p2D').

Table 7.2 presents the average metrics resulted from conducting this experiment which shows a slight enhancement that varies between 0.14% and 0.35%.

Table 7.2: The results of the CRF classifier when including 'p2D' in the tagset

A	P	R	F
97.41	94.56	71.81	81.39

7.3 Future Experiment, Injecting Part of Speech as Features into the Hybrid Approach

Part-of-Speech (POS) tag is a class that is allocated to a word according to its syntactic functions in the context. In the English language, common classes incorporate noun, pronoun, preposition, verb, adjective, adverb and others. In the literature, there are various sets of POS tags; such as the Universal POS tagset [50] and The Brown Corpus tagset [67], to be employed by different tools (taggers) such as the Stanford Part-of-Speech Tagger [65], Tree Tagger [1] and the Natural Language Toolkit (NLTK) [10].

Utilising POS as features may have a significant impact; either positive or negative, on the performance of the classifier. Therefore, injecting POS as features into our hybrid approach is an experiment that intended to be conducted as future work to inspect the effect of utilising such features on predicting mathematical definitions.

7.4 Chapter Summary

In this chapter, we presented our investigation to enhance the extraction of maths definitions from their context employing the CRF statistical algorithm in combination with the Rprop$^-$ optimisation algorithm and utilising the same features that used in our hybrid approach (see Section 6.2.7). In Section 7.1, we explored the effect of eliminating the stop words from our data. In Section 7.2, we investigated extending the tagset to include an extra tag for the divided definition (i.e. when the defined maths expression is embedded in its definition). Subsequently, in Section 7.3 we presented our approach to injecting the POS tags into the features that employed in our hybrid approach as future work. The evaluation of all the experiments that demonstrated in this chapter will be discussed in Chapter 8.

Part IV

Evaluation

CHAPTER 8

EVALUATION

The evaluation of IE is a significant stage to investigate the performance sufficiency of the IE process. We evaluated our approach progressively throughout the research as each step is evaluated at a time. We used different evaluation techniques for different tasks as follows:

- In the corpus building phase:
  - The extraction of maths formulae was evaluated by building a ground truth and comparing the extracted results with it.
  - The reliability of the annotation was quantitatively evaluated using κ-statistic to measure the inter-annotator agreement.
- In the testing phase:
  - Performance of the statistical learning models was quantitatively evaluated using conventional metrics; accuracy, recall, precision and F-measure.
  - The error in labelling is analysed using a confusion matrix which exhibits the variance between the correct and predicted labels.
  - The resulted predictions were qualitatively evaluated by inspecting and comparing them with the GSC.

These evaluation techniques will be discussed in this chapter.

8.1 Maths Formulae Extraction

To be able to evaluate the performance of our implementation of the algorithms which are used to extract maths formulae (maths formulae extraction was discussed in Section 4.3 and Section 5.2), we built a ground truth of a number of representative documents which eventually we compared to it the extracted maths expressions. The ground truth was built by manually annotating a randomly selected 10% of the collected data (collecting data was discussed in Section 4.2). The annotation in this step involved determining maths expressions within the selected context. Therefore, the maths expressions extracted by our algorithms were visually compared with this annotation. Our approach to automated maths expressions extraction achieved 100% of all the metrics; accuracy, recall and precision.

8.2 Data Annotation

A reliable annotation is essential to building a GSC. Therefore, we started the annotation step (discussed in Section 4.5) by providing the annotators with guidelines that stated precisely the attributes of different classes. They began with a training session where they annotated only 10% of the documents to be able to observe the restraint that could occur during the annotation process. Following that, they annotated the rest of the documents. Moreover, we evaluated the reliability of our annotation by measuring the inter-annotator agreement. For this measurement, we used the robust qualitative metrics κ-statistic. The result achieved is $\kappa = 0.9334\%$ which is high enough to proceed with our annotation.

8.3 Extract the Semantic Information of Maths Formulae from their Context

To extract the semantic information from the maths context we used SML; in particular, we used statistical algorithms. To evaluate the performance of the classifiers, we applied the standard quantitative metrics; accuracy, recall, precision and F-measure. In addition, the K-fold cross-validation (where k = 10) model was adopted to avoid the overfitting problem. Therefore, we applied these metrics ten times, once at each round of the 10-fold then their average was calculated.

The averages of accuracy for all our experiments (see Section 6.2) are floating between 94.6% and 97.3%. These numbers are generally high which is expected as the set of defined maths and their definitions is relatively small in comparison with the entire text. Moreover, to evaluate the performance of the classifiers and express the differentiation between the correct annotations and the predictions, we used an error analysis method which is a confusion matrix.

8.3.1 Evaluation of the Baseline Model Based on MaxEnt

In this experiment, we trained the MaxEnt model in combination with the Rprop$^+$ optimal algorithm; using basic unigram contextual window size features. Table 6.2 present the resulted metrics for this experiment which shows a low recall average (about 20.8%) and about twice it is the precision average. Consequently, a low average of F-measure has resulted; F = 27.2%.

8.3.2 Evaluation of Using MEMM and CRF as Different Classifiers

We conducted a set of experiments where the same features and scheme that used in the baseline model is used, yet with different classifiers and optimisation algorithms. In

these experiments, the MEMM and CRF statistical models combined with the Rprop$^+$ and Rprop$^-$ optimisation algorithms are used; in addition to the MaxEnt which is used for the baseline model but in this experiment, it is combined with the Rprop$^-$. The results presented in Table 6.3 show that using Rprop$^-$ optimisation algorithm is slightly improving the performance of the classifiers over the Rprop$^+$. Indeed, CRF functioning better than MEMM and also the MaxEnt models. In general, all models achieved low recall in contrast with their high precision; which indicates that the classifiers are very selective and omit most of the definitions even though the recognised ones are correct. Among the used classifiers, CRF combined with the Rprop$^-$ achieved the best F-measure result as F = 33.797%.

8.3.3 Evaluation of Using Different Features with the CRF

In the previous experiments, we inspected the performance of different classifiers when combined with different optimisation algorithms. The CRF model when integrated with the Rprop$^-$ decision function achieved the best results among the other used classifiers. Consequently, we contrived a series of studies using the CRF with the Rprop$^-$, yet playing with the used features which are the contextual window size. The results of these experiments are presented in Table 6.4 which shows that as the window size is extended, the performance of the classifier is getting poorer. The highest average of F-measure is achieved when using a -/+ 1 window size as F = 38.36%.

8.3.3.1 Error Analysis (Confusion Matrix)

A confusion matrix is a method to assess and show the diversity among the predicted and targeted tags. For each class, it presents the number of times it was predicted rightly or wrongly. Table 8.1 presents the confusion matrix of using the CRF\Rprop$^-$ statistical model employing a -/+ 1 window size features. The numbers in magenta show the accurate labelling. For instance, the class 'exp' has been accurately prophesied 160 times, while it has been inaccurately prophesied three times as 'def' and 272 as 'O' which means it

was not be able to be prophesied 272 times. Also, the class 'def' has been correctly predicted 211 times, only four times predicted as 'exp' and 1092 times was not be able to be predicted. Interestingly, the class 'p1D' failed to be predicted at all, and thus it has been predicted 40 times as 'O'.

Table 8.1: Confusion Matrix of using CRF with Rprop$^-$ and the contextual window size features

		Prediction			
		exp	def	p1D	O
Target	exp	160	3	0	272
	def	4	211	0	1092
	p1D	0	0	0	40
	O	32	116	0	27824

8.3.4 Evaluation of the Hybrid Approach

In this experiment, we injected rule-based features; which are the templates' morphological features, into the CRF statistical model to benefit the most from both models. It is observed that the templates that have a low frequency (see Table 6.6) rose the noise in the data and not adding any benefit to it. Table 6.7 presents the best results achieved in this experiment where not all of the extracted templates' features used and the window sizes are between -/+ 2 and -/+ 5. Injecting the rule-based features into the statistical model has certainly boosted both recall and precision; from about 28% to 71.5% and from 62% to 94%, respectively, which enhance the F-measure score to be F = 81.1%.

8.3.4.1 Error Analysis (Confusion Matrix)

The confusion matrix for the hybrid-based approach is presented in Table 8.2. It shows a significant improvement in tagging the class 'exp' as 423 were correctly predicted and only 12 failed to do so. For the class 'def', there is a considerable refinement over the previous experiment (evaluated in Section 8.3.3), as 592 were rightly predicted, nine were

wrongly predicted as 'exp' and 706 were failed to be predicted. However, none of the 'p1D' tags was prophesied which means it was predicted as 'O'.

Table 8.2: Confusion Matrix of the hybrid-based experiments

		Prediction			
		exp	def	p1D	O
Target	exp	423	5	0	7
	def	9	592	0	706
	p1D	0	0	0	40
	O	0	14	0	27958

8.3.5 Evaluation of the Experiment of Removing the Stop Words

In this study, the stop words were removed from our data, and the same scheme as the one used in the hybrid approach is used to predict maths definitions. The results of this experiment are presented in Table 7.1 and show an advanced performance of the model. Although withdrawing the stop words from the text may result in losing information when extract information from the text, this is not the case in our task. Comparing with the results of the hybrid approach (see Section 8.3.4), the results show 1.7% and 0.61% increment in the average of precision and recall, respectively, which led to 1.01% improvement in the F-measure average over the hybrid results.

8.3.5.1 Error Analysis (Confusion Matrix)

The confusion matrix for the experiment of removing the stop words from the data while employing the hybrid approach is presented in Table 8.3. It suggests that the classifier shows a distinguish difficulty in tagging the 'def' class as approximately half of the correct tags were failed to be predicted; despite this, it still shows improved over the hybrid model performance. There are nine of the 'def' tag that predicted as 'exp' which are originally maths expressions that are part of definitions, i.e. they are not targeted as expressions but as definitions. For the class 'exp', the classifier is slightly impair as 13 expressions

were mistakenly predicted as definitions, and 21 expressions were unsuccessfully predicted (i.e. tagged as 'O'); Nevertheless, 400 of 'exp' were accurately prophesied. Moreover, as in all the previous experiments, none of the 'p1D' tags was predicted.

Table 8.3: Confusion Matrix of the hybrid-based experiments when removing the Stop words

		Prediction			
		exp	def	p1D	O
Target	exp	400	13	0	21
	def	8	488	0	467
	p1D	0	0	0	36
	O	0	12	0	17404

8.3.6 Evaluation of Extending the Annotation's tagset

We inspected the performance of our hybrid approach (Section 6.2.7) while extending the used tagset to include the tag 'p2D' as explained in Section 7.2. Table 7.2 presents the average metrics of the performance of this experiment. It shows an insignificant improvement; comparing with the hybrid approach, overall the used metrics with 0.35%, 0.26% and 0.29% increased in the average of the precision, recall and F-measure, respectively.

8.3.6.1 Error Analysis (Confusion Matrix)

Table 8.4 presents the confusion matrix of the experiment of extending the used tagset. It shows that 21 of the 'exp' tags and 577 of the 'def' tags have been mispredicted. It also shows that the classifier still has the same confusion of the second type of the mathematical definitions that tags with 'p1D' and 'p2D' as none of them were predicted. This result suggests more investigation for different approaches to extract such maths definitions.

Table 8.4: Confusion Matrix of the hybrid-based experiments when extending the tagset

		Prediction				
		exp	def	p1D	p2D	O
Target	exp	414	12	0	0	9
	def	9	644	0	0	568
	p1D	0	0	0	0	40
	p2D	0	0	0	0	85
	O	0	24	0	0	27949

8.4 Qualitative Evaluation

To qualitatively evaluate our approach, we inspected the predictions that resulted from our different experiments and compared them with our GSC. This evaluation led to various observations:

- We explained in Section 4.5 the use of 'p1D' tag where we have a divided definition, i.e. when the maths expression is embedded in its definition as the example in Figure 4.6 shows. In this case, the definition is divided by the expression into two parts; the first part which is before the expression and tagged with 'p1D' and the second part that comes after the expression and tagged with 'def'.

  We have 28 definitions of this kind; however, none of them was predicted. I believed that this could be improved by extending the tagset. Therefore, in Section 7.2, we conducted an experiment trying to improve this results by extending the tagset to include a tag called 'p2D' that used for the second part of such definitions. Nevertheless, none of such definitions was predicted.

- The classifier never predicts a definition without its expression; which is the right behaviour. However, 31% of the predicted expressions have no predicted definitions which means that the classifier successfully predicted some of the defined maths expressions but failed to predict their definitions.

- If a sentence contains only one definition, it has a higher chance of being predicted;

56.6% of such definitions were predicted.

- Nested definitions, where a definition is embedded in another definition, are a problem and were not predicted.

- The confusion matrices presented in Section 8.3 showed that a large number of the 'def' tag was mispredicted and labelled with the 'O' tag. Part of this could be a result of the restricted boundary of the predicted definitions that we applied throughout our experiments.

In general, the resulted predictions for the definitions of maths expressions are reasonably good and promising and could be improved significantly. Improving the performance of our approach will be discussed in Section 9.2.1.

8.5 Chapter Summary

This chapter presented different evaluation techniques for different stages in our approach; both quantitative and qualitative evaluation, which we used alongside the progress of our research. A ground truth of a number of representative documents was built to evaluate the maths extraction algorithms and its implementation. To evaluate the reliability of our annotation, we measured the inter-annotator agreement using the qualitative metric κ-statistic which validated our annotation. Moreover, K-fold cross-validation was used to improve the performance of the classifier and to overcome the problem of overfitting. To evaluate the performance of the different classifiers used, we applied the standard quantitative metrics; accuracy, recall, precision and F-measure; in addition to the confusion matrix as an error analysis of tagging. Finally, the resulted predictions were qualitatively evaluated by inspecting them and comparing them with the GSC. Therefore, we distinguished some shortcomings of our approach, that could be overcome by applying some suggestions which will be discussed in Section 9.2.1. The overall evaluation process demonstrates that our approach for extracting semantic information from maths documents is acceptable and promising.

CHAPTER 9

CONCLUSIONS AND FUTURE WORK

9.1 Conclusions

In this thesis, we have presented an approach for determining and extracting the semantics of mathematical formulae in mathematical documents by analysing both the representation of mathematical formulae and their context. To ease the challenge, we restricted our research on the math documents in a specific domain; which is Elementary Number Theory. Nonetheless, this restriction could be released afterwards to extend the research. This thesis answered the following research questions:

- How can the maths formulae be recognised and extracted from the XML format of documents depending on maths formulae properties?
- How can one extract semantic information for a particular mathematical formula from the context information?
- How can one adapt supervised machine learning techniques for text analyses in the presence of mathematical formulae?
- Which probabilistic model (i.e. classifier) is the most efficient for extracting the defined maths formulae with their definitions from maths documents?
- What are the instructive features that can be obtained from mathematical documents to be utilised by the probabilistic model?

We developed a novel approach for developing MathExtractor, which is a tool that extracts mathematical formulae from the XML format of the documents depending on the properties of the formulae; such as type, position and font. MathExtractor has the advantage of the powerful XPath functions which are based on travelling through the XML nodes using a predicate as a path to the targeted nodes; i.e. maths expression. The MathExtractor tool was evaluated by visually compared its extractions with a manually built ground truth of a number of representative documents, which achieved 100% of all the metrics; accuracy, recall and precision.

Moreover, we described the methodology of extracting the basic semantic information; such as font, maths style and the syntactic and semantic roles, from the representation of maths formulae; which is the internal representation of LaTeXML.

Also, we demonstrated the possibility of adapting the supervised machine learning techniques for text analyses in the presence of mathematical formulae. Such an obstacle has been removed by abstracting mathematical documents from maths formulae and replacing them with unique IDs.

A gold standard corpus (GSC) is an essential requirement of the machine learning algorithms. Therefore, we have developed a manually-created GSC, which its mathematical documents are harvested from the ArXive. The annotation of the documents was carried out by two expert mathematicians. The reliability of the annotation was evaluated by measuring the inter-annotator agreement using the κ-statistic, which computed as $\kappa = 0.9334\%$.

We have demonstrated a novel approach for extracting the semantic information of mathematical formulae from the context information by adapting supervised machine learning techniques; in particular, statistical learning algorithms. A series of experiments were conducted as follows:

- A baseline model is developed using Maximum Entropy (MaxEnt) in combination with the Rprop$^+$ optimisation algorithm to function as a bottom line performance to evaluate our approach. The MaxEnt model is chosen as it is commonly employed

in IE and it does not require a long time or a large memory to run.

- We learnt different probabilistic models; MaxEnt in combination with the Rprop$^-$, CRF and MEMM in conjunction with both Rprop$^+$ and Rprop$^-$ optimisation algorithms. The evaluation of these classifiers using the four metrics accuracy, recall, precision and F-measure showed that the CRF classifier combined with Rprop$^-$ is the most efficient for extracting the defined maths formulae with their definitions from maths documents.

- Since the CRF classifier combined with the Rprop$^-$ performed the best over the other classifiers, a further experiment was conducted to investigate the impact of different window size features on its performance. The results of this experiment showed that as the window size is extended, the performance of the classifier is getting poorer.

- We investigated the influence of injecting rule-based features; i.e. definitions' templates, into the CRF statistical model to benefit the most from both models. In this approach, the performance of the classifier was boosted from 38.36% (before injecting the rule-based features) to 81.10% F-measure.

- Employing the hybrid mathod, we investigated the impact of removing the stop words from our corpus. This has shown an improvement in the prediction of the defined maths formulae and their definitions.

- The divided definitions are the type of definitions that failed to be predicted by the employed classifiers. We investigated the influence of extending the used tagset to include a new tag that assigned to the second part of such definitions. Extending the tagset have no effect on the predictions of this type of definitions as none of them was predicted. However, this experiment showed a minor improvement in the prediction of the standard kind of definitions.

We developed a new approach for feature representation relying on the definitions' templates that extracted by expert mathematicians from maths documents to defeat the

restraint of conventional window-based features; and therefore, enhancing the performance of the classifier as shown in the hybrid model.

9.2 Future Work

The substantial research presented in this thesis is currently the basis of ongoing research work to integrate recognised definitions into a semantic enrichment procedure that aims to improve the display and accessibility of mathematics in web documents. In particular, the results of the machine learning process can be exploited to inform better the presentation of formulas when rendered aurally using screen reading software, by drawing reference links to definitions of components in mathematical expressions.

Nevertheless, we have also identified some shortcomings of our approach for extracting semantic mathematical formulae in documents and composed a list of suggestions to improve its performance. Furthermore, it highlights the need for a plethora of work in other related areas.

9.2.1 Improving our Approach

This section suggests some ideas to improve the performance of our approach, and therefore improve the semantics understanding of mathematical formulae in documents.

- The dataset which we used in our system is relatively small. Increasing the size of the dataset will influence the predicted performance. Therefore, this is an aspect to be investigate.

- Two identified types of definitions never been predicted by the classifiers; that are the divided definitions and the nested definitions where a definition is embedded in another one. We inspected the impact of extending the tagset on predicting the divided definitions, and it does not solve the difficulty. Therefore, we suggest a new approach that is dividing the experiment into several classifier experiments to predict

each type of definitions in a separate phase; i.e. have the standard definitions, the divided and the embedded form of definitions predicted each in one classifier goes.

- In the step of features extraction, we extracted two types of features; the contextual window size features and the templates' morphological features. Extracting additional types of features such as word stemming, part of speech and representation of the base phrase chunk such as noun phrase (NP) and a verb phrase (VP) could be beneficial. Thus, there is a need to conduct a series of experiments to inspect the effect of employing such features on the classifier.

- We restricted the boundary of the predicted definitions throughout our experiments. We suggest experiments with a soft boundary, which could improve the predictions of maths definitions.

9.2.2 Future Research Areas

The extensive research presented in this thesis is inspiring a number of areas for future work. Indeed, these areas have their influences on our approach and accomplishing any of them would lead to better performance of our system.

- Developing a framework to represent the interpretation of maths formulae which resulting from our system in a way that can be used by other existing systems such as maths searching systems and maths display engines.

- In our approach, we restricted our data to mathematical documents from the domain of Elementary Number Theory to ease the start of our research. However, this restriction could be released to extend the study and generalise the work.

- Developing an efficient annotation tool that facilitates annotating mathematical documents semantically. Such a tool would enhance building gold standard corpora.

- In maths documents, not all the used maths expressions are defined within the context. The usage of maths symbols and expressions can be viewed as one of the

following:

- A maths expression is never defined within the document as it has well-known meanings either in general such as the symbol = or in particular maths field such as $\leqslant$ in the field of Group Theory.
- A maths expression is defined once within the document, i.e. it has a unique definition throughout the document.
- A maths expression is defined several times within the document. This means that the definition of this expression is changing throughout the document such as starting with a particular definition and later in the document adding some restrictions on the initial definition. In this case, it is essential to determine the scope of each definition of the math expression.

Therefore, it is an interesting area to explore and find the limits to distinguish different levels of knowledge that actually given in the documents.

Part V

Appendices

APPENDIX A

SAMPLE OF SOME DEFINITIONS' TEMPLATES

Table A.1: Let templates

Shortcut	Templates	Example
tl1	Let exp denote def	Let $\mathbb{Z}_n$ denote the ring of integers modulo n.
tl2	Let us call exp as def	let us call $2i$ as the *little* end.
tl3	Let exp be def	Let V be an n-dimensional vector space over the field F.
tl4	Let exp and/with exp be def	Let $a = (_0, \ldots, a_{n-1})$ and $b = (b_0, \ldots, b_{n-1})$ be distinct permutations of the set $\{0, \ldots, n-1\}$ such that the component-wise sums $c_i = a_i + b_i$ are all distinct.
tl5	Let def be denoted by exp	Let the sizes be denoted by $s(n)$ and $t(n)$ respectively.
tl6	Let exp stand for def	Let $\pi(x)$ stand for the number of primes less than or equal to x.

Table A.2: Other templates

Shortcut	Templates	Example
te1	... exp is/are/be def ...	... k is a power of 2,3 or 5 ...
te2	... exp def	For n odd, we prove ...

te3	exp as def.	. . . thus we can regard the permutations A^i as vectors in n-dimensional vector space $(\mathbb{Z}_n)^n$.
te4	define exp as def.	define $L(n)$ as the largest integer l so that . . .
te5	define exp to be def.	define $g(n)$ to be the smallest positive integer m such that . . .
te6	with exp and exp as def and def respectively.	. . . with $+$ and $.$ as addition and multiplication modulo n respectively.
te7-	te71 when / te72 where / te73 assume (that) / te74 Suppose (that) exp is/are def.	• . . . when n is even . . . • . . . where k is the number of distinct prime divisors of n. • assume that N is odd prime. • Suppose that s is odd.
te8	exp denotes def	• . . . recall that $\mathbb{Z}_n^\times$ denotes the set of invertible elements of $\mathbb{Z}_n$ • . . . where the c_i denote all integers coprime to (**totatives** of) n in increasing order. • The letter p will always denote a prime.
te9	exp stands for def	y stands for the fractional part of y.
te11	set exp to be def	set J to be greatest length of . . .

te12	denoting by exp def	Denoting by $\mathbb{Z}_n$ the sequence of the convergents ...
te13	use exp for def	We use $\omega(n)$ for the number of distinct (positive) prime factors of the positive integer n.
t-	• t1 subset/ t2 permutation/ t3 subspace/ t4 collection exp of ... exp. • t5 relation exp on def ... • t6 factors exp of def ...	• We are interested in obtaining bounds on the maximum size of a subset $\mathcal{P}$ of $\mathcal{S}(\mathbb{Z}_n)$ in the case when ... • if for any two distinct permutations σ, τ in $\mathcal{P}$, ... • For a subspace S of V ... • A collection $\mathcal{P}$ of permutations of $\mathbb{Z}_n$ ... of $\mathbb{Z}_n$, ... • define a relation $\sim$ on the set of permutations of $\mathbb{Z}_n$ • We list the prime factors q_i of n ...
td1	... def exp ...	• ... vectors $u, v \in V$ ... • ... the sets A_i and B_i ...

td2	... def exp for/of def ...	• ... linear functions $f_i : (\mathbb{Z}_n)^r \to \mathbb{Z}_n$ for $i = 1, \ldots, n - r$ such that ... • ... an interval $[y, x]$ of length L ... • The symbol $\|\|$ separates the first half of the period from the second.
td3-	td31 define/ td32 with/ td33 for/ td34 suppose/ td35 consider def exp ...	• define an integer N ... • ... with the standard inner product $\langle , \rangle$. • For any/some odd number k ... • Suppose the edges of the complete graph K_n, $n \geq 2$ are ... • consider the nonhomogeneous recurrence $w_n = w(k, r, s)$
td4	def is denoted by exp.	The set of all invertible elements of $\mathbb{Z}_n$ is called the *unit group* of $\mathbb{Z}_n$ and is denoted by $\mathbb{Z}_n^\times$.
td5	def will be called exp.	... $u = (u_1, \ldots, u_n)$ and $v = (v_1, \ldots, v_n)$ will be called the *standard* inner product on V.

APPENDIX B

SAMPLE OF XML FILES

Figure B.1: Sample of XML files

```
  <abstract name="Abstract">
    <p>Let <Math mode="inline" xml:id="m2" tex="\mathbb{Z}_{n}" text="Z _ n"><XMath><XMApp><XMTok role="SUBSCRIPTOP"
        scriptpos="post3"/><XMTok role="UNKNOWN" font="blackboard">Z</XMTok><XMTok role="UNKNOWN" font="italic">n</
        XMTok></XMApp></XMath></Math> denote the ring of integers modulo <Math mode="inline" xml:id="m3" tex="n"
        text="n"><XMath><XMTok role="UNKNOWN" font="italic">n</XMTok></XMath></Math>. In this paper
we consider two extremal problems on permutations of <Math mode="inline" xml:id="m4" tex="\mathbb{Z}_{n}" text="Z _ n"><
    XMath><XMApp><XMTok role="SUBSCRIPTOP" scriptpos="post3"/><XMTok role="UNKNOWN" font="blackboard">Z</XMTok><XMTok
    role="UNKNOWN" font="italic">n</XMTok></XMApp></XMath></Math>, namely,
the maximum size of a collection of permutations such that the sum of any two
<!-- %**** 0109.tex Line 50 **** -->distinct permutations in the collection is again a permutation, and the
maximum size of a collection of permutations such that the sum of any two
distinct permutations in the collection
is not a permutation. Let the sizes be denoted by <Math mode="inline" xml:id="m5" tex="s(n)" text="s * n"><XMath><XMApp><
    XMTok meaning="times" role="MULOP"></XMTok><XMTok possibleFunction="yes" role="UNKNOWN" font="italic">s</XMTok><XMTok
    close=")" open="(" role="UNKNOWN" font="italic">n</XMTok></XMApp></XMath></Math> and <Math mode="inline" xml:id="m6"
    tex="t(n)" text="t * n"><XMath><XMApp><XMTok meaning="times" role="MULOP"></XMTok><XMTok possibleFunction="yes"
    role="UNKNOWN" font="italic">t</XMTok><XMTok close=")" open="(" role="UNKNOWN" font="italic">n</XMTok></XMApp></
    XMath></Math>
respectively. The case when <Math mode="inline" xml:id="m7" tex="n" text="n"><XMath><XMTok role="UNKNOWN"
    font="italic">n</XMTok></XMath></Math> is even is trivial in both the cases, with
<Math mode="inline" xml:id="m8" tex="s(n)=1" text="s * n = 1"><XMath><XMApp><XMTok meaning="equals" role="RELOP">=</
    XMTok><XMApp><XMTok meaning="times" role="MULOP"></XMTok><XMTok possibleFunction="yes" role="UNKNOWN"
    font="italic">s</XMTok><XMTok close=")" open="(" role="UNKNOWN" font="italic">n</XMTok></XMApp><XMTok meaning="1"
    role="NUMBER">1</XMTok></XMApp></XMath></Math> and <Math mode="inline" xml:id="m9" tex="t(n)=n!" text="t * n =
    nfactorial"><XMath><XMApp><XMTok meaning="equals" role="RELOP">=</XMTok><XMApp><XMTok meaning="times" role="MULOP"></
    XMTok><XMTok possibleFunction="yes" role="UNKNOWN" font="italic">t</XMTok><XMTok close=")" open="(" role="UNKNOWN"
    font="italic">n</XMTok></XMApp><XMApp><XMTok meaning="factorial" role="POSTFIX">!</XMTok><XMTok role="UNKNOWN" font=
    "italic">n</XMTok></XMApp></XMApp></XMath></Math>. For <Math mode="inline" xml:id="m10" tex="n" text="n"><XMath><
    XMTok role="UNKNOWN" font="italic">n</XMTok></XMath></Math> odd, we prove <Math mode="inline" xml:id="m11" tex="s(n)\
    geq(n\phi(n))/2^{k}" text="s * n &gt;= (n * phi * n) / 2 ^ k"><XMath><XMApp><XMTok meaning="greater-than-or-equals"
    name="geq" role="RELOP">≥</XMTok><XMApp><XMTok meaning="times" role="MULOP"></XMTok><XMTok possibleFunction="yes"
    role="UNKNOWN" font="italic">s</XMTok><XMTok close=")" open="(" role="UNKNOWN" font="italic">n</XMTok></
    XMApp><XMApp><XMTok mathstyle="inline" meaning="divide" role="MULOP">/</XMTok><XMApp close=")" open="("><XMTok
    meaning="times" role="MULOP"></XMTok><XMTok role="UNKNOWN" font="italic">n</XMTok><XMTok name="phi"
    possibleFunction="yes" role="UNKNOWN" font="italic">ϕ</XMTok><XMTok close=")" open="(" role="UNKNOWN"
    font="italic">n</XMTok></XMApp><XMApp><XMTok role="SUPERSCRIPTOP" scriptpos="post3"/><XMTok meaning="2"
    role="NUMBER">2</XMTok><XMTok role="UNKNOWN" font="italic">k</XMTok></XMApp></XMApp></XMApp></XMath></Math>
where <Math mode="inline" xml:id="m12" tex="k" text="k"><XMath><XMTok role="UNKNOWN" font="italic">k</XMTok></XMath></
    Math> is the number of distinct prime divisors of <Math mode="inline" xml:id="m13" tex="n" text="n"><XMath><XMTok
    role="UNKNOWN" font="italic">n</XMTok></XMath></Math>. When <Math mode="inline" xml:id="m14" tex="n"
    text="n"><XMath><XMTok role="UNKNOWN" font="italic">n</XMTok></XMath></Math> is an
odd
```

APPENDIX C

SOME ATTRIBUTES OF ELEMENTS IN THE REPRESENTATION OF LaTeXML FOR MATHS EXPRESSIONS

(‘argclose’, ‘)’)
(‘argclose’, ‘]’)
(‘argclose’, ’|’)
(‘argclose’, ’}’)
(‘argclose’, ’⌉’)
(‘argclose’, ’⌋’)
(‘argclose’, ’⟩’)
(’argopen’, ’(’)
(’argopen’, ’[’)
(’argopen’, ’{’)
(’argopen’, ’|’)
(’argopen’, ’⌈’)
(’argopen’, ’⌊’)
(’argopen’, ’⟨’)
(’close’, ’)’)
(’close’, ’]’)
(’open’, ’(’)
(’open’, ’[’)
(’font’, ’blackboard upright’)
(’font’, ’bold italic’)
(’font’, ’caligraphic upright’)
(’font’, ’medium’)
(’mathstyle’, ’display’)
(’mathstyle’, ’inline’)
(’mathstyle’, ’script’)
(’mathstyle’, ’text’)
(’meaning’, ’1’)
(’meaning’, ’absent’)
(’meaning’, ’absolute-value’)
(’meaning’, ’annotated’)
(’meaning’, ’approximately-equals’)
(’meaning’, ’assign’)
(’meaning’, ’asymptotically-equals’)
(’meaning’, ’binomial’)

('meaning', 'ceiling')
('meaning', 'closed-interval')
('meaning', 'closed-open-interval')
('meaning', 'conditional-set')
('meaning', 'cosine')
('meaning', 'cotangent')
('meaning', 'divide')
('meaning', 'element-of')
('meaning', 'equals')
('meaning', 'equivalent-to')
('meaning', 'exists')
('meaning', 'exponential')
('meaning', 'factorial')
('meaning', 'floor')
('meaning', 'for-all')
('meaning', 'formulae')
('meaning', 'gcd')
('meaning', 'greater-than')
('meaning', 'greater-than-or-equals')
('meaning', 'hyperbolic-cosine')
('meaning', 'infinity')
('meaning', 'integral')
('meaning', 'intersection')
('meaning', 'less-than')
('meaning', 'less-than-or-equals')
('meaning', 'limit')
('meaning', 'list')
('meaning', 'logarithm')
('meaning', 'maps-to')
('meaning', 'maximum')
('meaning', 'minimum')
('meaning', 'minus')
('meaning', 'modulo')
('meaning', 'much-greater-than')
('meaning', 'much-less-than')
('meaning', 'multirelation')
('meaning', 'natural-logarithm')
('meaning', 'not-divides')
('meaning', 'not-element-of')
('meaning', 'not-equals')
('meaning', 'not-equivalent-to')
('meaning', 'nth-root')
('meaning', 'open-closed-interval')
('meaning', 'open-interval')
('meaning', 'parallel-to')
('meaning', 'partial-differential')
('meaning', 'perpendicular-to')
('meaning', 'plus')
('meaning', 'plus-or-minus')
('meaning', 'product')
('meaning', 'set')
('meaning', 'set-minus')
('meaning', 'similar-to')
('meaning', 'sine')
('meaning', 'square-root')
('meaning', 'subset-of')

('meaning', 'subset-of-or-equals')
('meaning', 'sum')
('meaning', 'supremum')
('meaning', 'times')
('meaning', 'vector')
('name', 'Delta')
('name', 'Gamma')
('name', 'Lambda')
('name', 'Longleftrightarrow')
('name', 'Phi')
('name', 'alpha')
('name', 'approx')
('name', 'asymp')
('name', 'beta')
('name', 'blacksquare')
('name', 'bmod')
('name', 'cap')
('name', 'cdot')
('name', 'cdots')
('name', 'colon')
('name', 'delta')
('name', 'dots')
('name', 'ell')
('name', 'epsilon')
('name', 'equiv')
('name', 'eta')
('name', 'forall')
('name', 'gamma')
('name', 'geq')
('name', 'gg')
('name', 'in')
('name', 'infty')
('name', 'int')
('name', 'lambda')
('name', 'langle')
('name', 'ldots')
('name', 'leq')
('name', 'list')
('name', 'll')
('name', 'longmapsto')
('name', 'longrightarrow')
('name', 'mapsto')
('name', 'mid')
('name', 'mu')
('name', 'neq')
('name', 'nmid')
('name', 'not-equiv')
('name', 'not-in')
('name', 'nu')
('name', 'omega')
('name', 'overline')
('name', 'partial')
('name', 'perp')
('name', 'phi')
('name', 'pi')
('name', 'pm')

('name', 'pmod')
('name', 'prime')
('name', 'prime2')
('name', 'prod')
('name', 'psi')
('name', 'qquad')
('name', 'quad')
('name', 'rangle')
('name', 'rho')
('name', 'rightarrow')
('name', 'sigma')
('name', 'sim')
('name', 'smallsetminus')
('name', 'square')
('name', 'subset')
('name', 'subseteq')
('name', 'tau')
('name', 'theta')
('name', 'tilde')
('name', 'to')
('name', 'varepsilon')
('name', 'varphi')
('name', 'xi')
('name', 'zeta')
('name', '‖')
('possibleFunction', 'yes')
('punctuation', ',')
('punctuation', '.')
('role', 'ADDOP')
('role', 'ARROW')
('role', 'BIGOP')
('role', 'CLOSE')
('role', 'OPEN')
('role', 'FENCED')
('role', 'ID')
('role', 'INTOP')
('role', 'LIMITOP')
('role', 'METARELOP')
('role', 'MODIFIEROP')
('role', 'MULOP')
('role', 'NUMBER')
('role', 'OPERATOR')
('role', 'OPFUNCTION')
('role', 'OVERACCENT')
('role', 'PERIOD')
('role', 'POSTFIX')
('role', 'PUNCT')
('role', 'RELOP')
('role', 'STACKED')
('role', 'SUBSCRIPTOP')
('role', 'SUMOP')
('role', 'SUPERSCRIPTOP')
('role', 'SUPOP')
('role', 'TRIGFUNCTION')
('role', 'UNKNOWN')
('role', 'VERTBAR')

('rpadding', '1.7pt')

('scriptpos', 'mid')

('scriptpos', 'post')

('separators', ', ,')

('separators', ',')

('separators', '. .')

('separators', '.')

('separators', ':')

('separators', '; ;')

('separators', ';')

('stretchy', 'false')

('thickness', '0.0pt')

APPENDIX

BIBLIOGRAPHY

[1] TreeTagger.

[2] An introduction to latex, Feb 2008. `http://www.latex-project.org/intro.html, Accessed 26 September, 2012`.

[3] arxiv.org, e-print archive. http://www.arxiv.org, June 17 2013.

[4] Latexml the manual, chapter 5 mathematics. http://dlmf.nist.gov/LaTeXML/manual/math/math.details.html, March 17 2016.

[5] The wolfram functions site. http://functions.wolfram.com, 2016.

[6] Information processing society of japan. http://www.ipsj.or.jp, 2017.

[7] Douglas E. Appelt, Jerry R. Hobbs, John Bear, David Israel, Megumi Kameyama, David Martin, Karen Myers, and Mabry Tyson. Sri international fastus system: Muc-6 test results and analysis. In *Proceedings of the 6th Conference on Message Understanding*, MUC6 '95, pages 237–248, Stroudsburg, PA, USA, 1995. Association for Computational Linguistics. ISBN 1-55860-402-2. doi: 10.3115/1072399.1072420. URL https://doi.org/10.3115/1072399.1072420.

[8] J.B. Baker, A.P. Sexton, and V. Sorge. Faithful mathematical formula recognition from pdf documents. In *Proceedings of the 9th IAPR International Workshop on Document Analysis Systems*, pages 485–492. ACM, 2010.

[9] Kyle Barnhart. Presentation mathml versus content mathml, Aug 2009. http://cnx.org/content/m31620/latest, Accessed 20 September, 2012.

[10] Steven Bird, Ewan Klein, and Edward Loper. *Natural language processing with Python*. O'Reilly Media, Inc., 2009.

[11] Andreea Bodnari and Thomas Lavergne. P.:a supervised named-entity extraction system for medical text. In *In: Proceedings of ShARe/CLEF eHealth Evaluation Labs*, 2013.

[12] S. Buswell, O. Caprotti, D.P. Carlisle, M.C. Dewar, M. Gaetano, and M. Kohlhase. The open math standard. Technical report, version 2.0. Technical report, The Open Math Society, 2004. http://www. openmath. org/standard/om20, 2004.

[13] Mary Elaine Califf and Raymond J. Mooney. Bottom-up relational learning of pattern matching rules for information extraction. *J. Mach. Learn. Res.*, 4:177–210, December 2003. ISSN 1532-4435. doi: 10.1162/153244304322972685. URL https://doi.org/10.1162/153244304322972685.

[14] Jean Carletta. Assessing agreement on classification tasks: The kappa statistic. *Comput. Linguist.*, 22(2):249–254, June 1996. ISSN 0891-2017. URL http://dl.acm.org/citation.cfm?id=230386.230390.

[15] Mario Catalani. On the average of triangular numbers. *arXiv preprint math/0304160*, 2003.

[16] Tsz Ho Chan. Finding almost squares II. *arXiv preprint math/0503438*, 2005.

[17] Christian Chiarcos, Stefanie Dipper, Michael Götze, Ulf Leser, Anke Lüdeling, Julia Ritz, and Manfred Stede. A flexible framework for integrating annotations from different tools and tagsets. *Traitement Automatique des Langues*, 49(2):271–293, 2008.

[18] Yejin Choi, Claire Cardie, Ellen Riloff, and Siddharth Patwardhan. Identifying sources of opinions with conditional random fields and extraction patterns. In *Proceedings of the Conference on Human Language Technology and Empirical Methods in Natural Language Processing*, HLT '05, pages 355–362, Stroudsburg, PA, USA, 2005. Association for Computational Linguistics. doi: 10.3115/1220575.1220620. URL https://doi.org/10.3115/1220575.1220620.

[19] Ronen Feldman, Benjamin Rosenfeld, and Moshe Fresko. Teg—a hybrid approach to information extraction. *Knowledge and Information Systems*, 9(1):1–18, Jan 2006. ISSN 0219-3116. doi: 10.1007/s10115-005-0204-y. URL https://doi.org/10.1007/s10115-005-0204-y.

[20] Yi feng Lin, Tzong han Tsai, Wen chi Chou, Kuen pin Wu, Ting yi Sung, and Wen lian Hsu. A maximum entropy approach to biomedical named entity recognition. In *Proceedings of KDD Workshop on Data Mining and Bioinformatics*, 2004.

[21] J.L. Fleiss et al. Measuring nominal scale agreement among many raters. *Psychological Bulletin*, 76(5):378–382, 1971.

[22] Katerina T. Frantzi, Sophia Ananiadou, and Junichi Tsujii. The c-value/nc-value method of automatic recognition for multi-word terms. In Christos Nikolaou and Constantine Stephanidis, editors, *Research and Advanced Technology for Digital Libraries*, pages 585–604, Berlin, Heidelberg, 1998. Springer Berlin Heidelberg. ISBN 978-3-540-49653-3.

[23] Deyan Ginev, Sourabh Lal, Michael Kohlhase, and Tom Wiesing. Kat: an annotation tool for stem documents.

[24] Kurt Girstmair. The period length of euler's number e. *arXiv preprint arXiv:1303.2887*, 2013.

[25] M. Grigore, M. Wolska, and M. Kohlhase. Towards context-based disambiguation of mathematical expressions. In *The Joint Conference of ASCM*, pages 262–271, 2009.

[26] Jim Hefferon. What are tex, latex, and friends?, 2012. http://www.ctan.org/what_is_tex.html, Accessed 20 September, 2012.

[27] Annika Hinze, Ralf Heese, Markus Luczak-Rösch, and Adrian Paschke. *Semantic Enrichment by Non-experts: Usability of Manual Annotation Tools*, pages 165–181. Springer Berlin Heidelberg, Berlin, Heidelberg, 2012. ISBN 978-3-642-35176-1. doi: 10.1007/978-3-642-35176-1_11. URL http://dx.doi.org/10.1007/978-3-642-35176-1_11.

[28] Frederik Hogenboom, Flavius Frasincar, and Uzay Kaymak. An overview of approaches to extract information from natural language corpora. *Information Foraging Lab*, pages 69–70, 1010.

[29] George Hripcsak and Adam S Rothschild. Agreement, the f-measure, and reliability in information retrieval. *Journal of the American Medical Informatics Association*, 12(3):296–298, 2005.

[30] George Hripcsak and Adam Wilcox. Reference standards, judges, and comparison subjects. *Journal of the American Medical Informatics Association*, 9(1):1–15, 2002.

[31] Kevin Humphreys, Robert J. Gaizauskas, Saliha Azzam, Charles Huyck, Brian Mitchell, Hamish Cunningham, and Yorick Wilks. University of sheffield: Description of the lasie-ii system as used for MUC-7. In *Seventh Message Understanding Conference: Proceedings of a Conference Held in Fairfax, Virginia, USA, MUC 1998, April 29 - May 1, 1998*, 1998. URL https://aclanthology.info/papers/M98-1007/m98-1007.

[32] Christian Igel and Michael Hsken. Empirical evaluation of the improved rprop learning algorithms. *Neurocomputing*, 50:105 – 123, 2003. ISSN 0925-2312. doi: https://doi.org/10.1016/S0925-2312(01)00700-7. URL http://www.sciencedirect.com/science/article/pii/S0925231201007007.

[33] Sanjeet Khaitan, Ganesh Ramakrishnan, Sachindra Joshi, and Anup Chalamalla. Rad: A scalable framework for annotator development. *2008 IEEE 24th International Conference on Data Engineering*, pages 1624–1627, 2008.

[34] Sotiris B Kotsiantis, I Zaharakis, and P Pintelas. Supervised machine learning: A review of classification techniques, 2007.

[35] Olga Kushel. Generalized brouncker's continued fractions and their logarithmic derivatives. *The Ramanujan Journal*, 32(1):109–124, 2013.

[36] John D. Lafferty, Andrew McCallum, and Fernando C. N. Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the Eighteenth International Conference on Machine Learning*, ICML '01, pages 282–289, San Francisco, CA, USA, 2001. Morgan Kaufmann Publishers Inc. ISBN 1-55860-778-1. URL http://dl.acm.org/citation.cfm?id=645530.655813.

[37] Thomas Lavergne, Olivier Cappé, and François Yvon. Practical very large scale CRFs. In *Proceedings the 48th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 504–513. Association for Computational Linguistics, July 2010. URL http://www.aclweb.org/anthology/P10-1052.

[38] Manfred Madritsch and Volker Ziegler. An infinite family of multiplicatively independent bases of number systems in cyclotomic number fields. *arXiv preprint arXiv:1403.1673*, 2014.

[39] MG Madritsch and Robert F Tichy. Construction of normal numbers via generalized prime power sequences. *Journal of Integer Sequences*, 16(2):3, 2013.

[40] Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schtze. *Introduction to Information Retrieval*. Cambridge University Press, Cambridge, UK, 2008. ISBN 978-0-521-86571-5. URL http://nlp.stanford.edu/IR-book/information-retrieval-book.html.

[41] Andrew McCallum, Dayne Freitag, and Fernando C. N. Pereira. Maximum entropy markov models for information extraction and segmentation. In *Proceedings of the Seventeenth International Conference on Machine Learning*, ICML '00, pages 591–598, San Francisco, CA, USA, 2000. Morgan Kaufmann Publishers Inc. ISBN 1-55860-707-2. URL http://dl.acm.org/citation.cfm?id=645529.658277.

[42] Romeo Meštrović. An elemetary proof of an estimate for a number of primes less than the product of the first n primes. *arXiv preprint arXiv:1211.4571*, 2012.

[43] Bruce Miller. Latexml: A latex to xml converter. *Web Manual at http://dlmf.nist.gov/LaTeXML/, Accessed September2007*, 2010.

[44] Marie-Francine Moens. *Information Extraction: Algorithms and Prospects in a Retrieval Context*. Springer Netherlands, 2006. ISBN 978-90-481-7246-7.

[45] Umesh P Nair. Elementary results on the binary quadratic form aˆ 2+ ab+ bˆ 2. *arXiv preprint math/0408107*, 2004.

[46] Minh-Quoc Nghiem, Giovanni Yoko Kristianto, Goran Topić, and Akiko Aizawa. A hybrid approach for semantic enrichment of mathml mathematical expressions. In *Intelligent Computer Mathematics*, pages 278–287. Springer, 2013.

[47] Minh-Quoc Nghiem, Giovanni Yoko, Yuichiroh Matsubayashi, and Akiko Aizawa. Towards mathematical expression understanding. 2014.

[48] Damien Nouvel, Jean-Yves Antoine, Nathalie Friburger, and Arnaud Soulet. Coupling knowledge-based and data-driven systems for named entity recognition. In *Proceedings of the Workshop on Innovative Hybrid Approaches to the Processing of Textual Data*, HYBRID '12, pages 69–77, Stroudsburg, PA, USA, 2012. Association for Computational Linguistics. URL http://dl.acm.org/citation.cfm?id=2388632.2388642.

[49] Gerhard R Paseman. Updating an upper bound of erik westzynthius. *arXiv preprint arXiv:1311.5944*, 2013.

[50] Slav Petrov, Dipanjan Das, and Ryan McDonald. A universal part-of-speech tagset. In Nicoletta Calzolari (Conference Chair), Khalid Choukri, Thierry Declerck, Mehmet Uur Doan, Bente Maegaard, Joseph Mariani, Asuncion Moreno, Jan Odijk, and Stelios Piperidis, editors, *Proceedings of the Eight International Conference on Language Resources and Evaluation (LREC'12)*, Istanbul, Turkey, may 2012. European Language Resources Association (ELRA). ISBN 978-2-9517408-7-7.

[51] Jakub Piskorski and Roman Yangarber. *Information Extraction: Past, Present and Future*, pages 23–49. Springer Berlin Heidelberg, Berlin, Heidelberg, 2013. ISBN 978-3-642-28569-1. doi: 10.1007/978-3-642-28569-1_2. URL https://doi.org/10.1007/978-3-642-28569-1_2.

[52] Lutz Prechelt. Automatic early stopping using cross validation: quantifying the criteria. *Neural Networks*, 11(4):761–767, 1998.

[53] W3C Recommendation. Mathematical markup language (mathml) version 3.0: Presentation markup, October 2016. URL https://www.w3.org/TR/MathML3/chapter3.html#presm.intro.

[54] Martin Riedmiller. Advanced supervised learning in multi-layer perceptrons - from backpropagation to adaptive learning algorithms, 1994.

[55] Martin Riedmiller and Heinrich Braun. A direct adaptive method for faster backpropagation learning: The rprop algorithm. In *Neural Networks, 1993., IEEE International Conference on*, pages 586–591. IEEE, 1993.

[56] Angus Roberts, Robert Gaizauskas, Mark Hepple, and Yikun Guo. Combining terminology resources and statistical methods for entity recognition: an evaluation. In *Proceedings of the Conference on Language Resources and Evaluation (LRE08*, 2008.

[57] D. E. Rumelhart, G. E. Hinton, and R. J. Williams. Parallel distributed processing: Explorations in the microstructure of cognition, vol. 1. chapter Learning Internal Representations by Error Propagation, pages 318–362. MIT Press, Cambridge, MA, USA, 1986. ISBN 0-262-68053-X. URL http://dl.acm.org/citation.cfm?id=104279.104293.

[58] Arthur L Samuel. Some studies in machine learning using the game of checkers. *IBM Journal of research and development*, 3(3):210–229, 1959.

[59] Sunita Sarawagi. Information extraction. *FnT Databases*, 1(3), 2008.

[60] Nitin Singh, Deepak Rajendraprasad, and L Sunil Chandran. On additive combinatorics of permutations of n. *Discrete Mathematics & Theoretical Computer Science*, 16, 2014.

[61] Yiannos Stathopoulos and Simone Teufel. Mathematical information retrieval based on type embeddings and query expansion. In *COLING*, 2016.

[62] Pontus Stenetorp, Sampo Pyysalo, Goran Topić, Tomoko Ohta, Sophia Ananiadou, and Jun'ichi Tsujii. Brat: a web-based tool for nlp-assisted text annotation. In *Proceedings of the Demonstrations at the 13th Conference of the European Chapter of the Association for Computational Linguistics*, pages 102–107. Association for Computational Linguistics, 2012.

[63] Jürgen Stuber and Mark Van den Brand. Extracting mathematical semantics from latex documents. In *International Workshop on Principles and Practice of Semantic Web Reasoning*, pages 160–173. Springer, 2003.

[64] `the infrastructure group`. Openmath and mathml, 2012. `http://www.openmath.org/overview/om-mml.html, Accessed 18 April, 2012`.

[65] Kristina Toutanova, Dan Klein, Christopher D. Manning, and Yoram Singer. Feature-rich part-of-speech tagging with a cyclic dependency network. In *Proceedings of*

the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology - Volume 1, NAACL '03, pages 173–180, Stroudsburg, PA, USA, 2003. Association for Computational Linguistics. doi: 10.3115/1073445.1073478. URL https://doi.org/10.3115/1073445.1073478.

[66] Mark G. J. van den Brand, Jeroen Scheerder, Jurgen J. Vinju, and Eelco Visser. Disambiguation filters for scannerless generalized lr parsers. In R. Nigel Horspool, editor, *Compiler Construction*, pages 143–158, Berlin, Heidelberg, 2002. Springer Berlin Heidelberg. ISBN 978-3-540-45937-8.

[67] Hans van Halteren, editor. *Syntactic Wordclass Tagging*. Text, Speech and Language Technology. Kluwer Academic Publishers, 1999.

[68] Recommendation W3C. Mathematical markup language (mathml) version 3.0: Content markup, October 2010. `http://www.w3.org/TR/REC-MathML/chap4_1.html, Accessed 26 September, 2012.`

[69] Wolfram. Working with mathml, 2012. `http://reference.wolfram.com/mathematica/XML/tutorial/MathML.html, Accessed 18 April, 2012.`

[70] M. Wolska and M. Grigore. Symbol declarations in mathematical writing. *Towards a Digital Mathematics Library. Paris, France, July 7-8th, 2010*, pages 119–127, 2010.

[71] M. Wolska, M. Grigore, and M. Kohlhase. Using discourse context to interpret object-denoting mathematical expressions. In *Towards Digital Mathematics Library, DML workshop. Masaryk University, Brno*, 2011.

[72] Keisuke Yokoi, Minh-Quoc Nghiem, Yuichiroh Matsubayashi, and Akiko Aizawa. Contextual Analysis of Mathematical Expressions for Advanced Mathematical Search. *Polibits*, pages 81 – 86, 06 2011. ISSN 1870-

9044. URL http://www.scielo.org.mx/scielo.php?script=sci_arttext&pid=S1870-90442011000100011&nrm=iso.

[73] Ziqi Zhang. Named entity recognition: challenges in document annotation, gazetteer construction and disambiguation. 2013.