Department of Computer Science, Jacobs University Bremen
D-28759 Bremen, Germany

Department of Computational Linguistics, Saarland University
66041 Saarbrücken, Germany

Master Thesis
submitted in partial fulfillment of the requirements for the degree of Master of Science in Computer Science

Knowledge-poor Interpretation of Mathematical Expressions in Context

Mihai Grigore

Submitted: August 30, 2010

Supervisors:

Magdalena Wolska, Saarland University
Prof. Dr. Michael Kohlhase, Jacobs University Bremen
Prof. Dr. Manfred Pinkal, Saarland University

Contents

Chapter 1

Introduction

1.1 Motivation and research questions

In this thesis we investigate the... Our motivation for this proposal is twofold: on the one hand we are motivated by ..., and on the other hand, ...

Our second motivation is that... We argue that ...

1.2 The task of mathematical expression disambiguation

The central idea of this thesis is to combine the information provided by the context of symbolic expressions with the implicit knowledge used in mathematical notations, in order to develop a system that automatically understands the meaning of mathematical objects. More precisely, we will investigate to what extent we can combine the background mathematical knowledge with the information provided by both the surrounding text and the notational conventions in order to resolve ambiguities in mathematical discourse.

1.3 Related work

1.4 The structure of the thesis

We begin in Chapter 2 by introducing the reader to the basic notions of ... Chapter 2 presents the data to be used in this work.

a descriptive overview of mathematics from a linguistic perspective. Â§2.6 is of particular importance as it highlights

We then continue in Chapter 3 with ... Chapter 4 presents the ..., followed by Chapter 5 Chapter 4 discusses the phenomena in math ... :)

Chapter 5, the main contribution of this thesis, presents ... in detail ... We follow this in Chapter 8 with an example system and evaluation.

Chapter 9 summarizes the thesis and gives an outlook.

In this thesis we have investigated ... The result of our research is an incremental .... of processing mathematical documents :-)

Our motivations for this research came from two directions. The first was our exoerience ... The second motivation was the use of ...

There are two main contributions id this thesis. The first is the design and the implementation of the .... The second and the most important contribution of the thesis is the

The results of this thesis suggest two general directions for future work

Chapter 2

The Data

2.1 Introduction

In this chapter we describe the data we will use in this work. We first present the arXiv.org e-Print Archive of scientific publications and discuss the motivation to translate the TEX/LATEX-based mathematical papers into a format which is more suitable for automated processing. We then describe the architecture of the LaTeXML document processing system and present the main output formats for representing mathematical content it uses. We finally summarize the preparation of the corpus subset which will be used in the experiments throughout this thesis.

2.2 The arXMLiv corpus

The arXiv repository [5], administered by the Cornell University Library, is one of the largest collections of scientific mathematical papers, typeset in TEX and LATEX. The database contains over $600,000$ scientific e-print documents, representing research literature from the fields of *Physics, Mathematics, Computer Science, Quantitative Biology, Quantitative Finance and Statistics*. In order to investigate ambiguity phenomena in mathematical discourse, we will use a subset of arXMLiv documents from the area of *Mathematics*. As TEX/LATEX [17] is a presentation-oriented typesetter, in general it does not provide suitable information for semantic analysis of formal mathematical fragments. Instead of LATEX we therefore want to use a different format for representing mathematical fragments which facilitates our semantic exploration.

Below we describe the LaTeXML document processing system which automates the conversion of TEX/LATEX-based documents into a machine understandable XML format. The arXMLiv corpus is created by converting the TEX/LATEX sources of documents contained in the arXiv repository to an XML format, using the LaTeXML software.

2.2.1 The LaTeXML document processing system

The currently under development LaTeXML system [24] is intended to support the creation of the *Digital Library of Mathematical Functions* [20]. The system converts TEX/LATEX sources into a custom XML representation, by attempting to emulate the behavior of the TEX typesetting system. Below we present the architecture of the LaTeXML system in detail and describe its main output formats for representing mathematical expressions.

2.2.1.1 The architecture

The LaTeXML architecture [25] (Figure 2.1) comprises two components: 1) the TEX emulator, LaTeXML, which converts the TEX sources into an intermediate LATEX-like XML representation based on an implicit document model used by LATEX, and 2) the post-processor, LaTeXMLpost, which converts the intermediate results into XML-based output formats.

The TEX emulator LaTeXML realizes the XML conversion in the following stages: it first converts the input TEX/LATEX documents into boxes which mimic the structure of the parse trees of the LATEX sources (*TEX-like processing* phase), the boxes are then converted into an XML Document Object Model [2] (*Construction* phase), further modified (*Rewriting* phase), and finally output into an XML document (*Serialization* phase). The result aims at preserving the structure and possibly the semantics implied by the TEX markup.

The post-processor LaTeXMLpost uses a pipeline of filters to convert the resulting LATEX-like XML representation into various output formats (XHTML + MathML, XHTML + Images, etc.). The main capability of LaTeXMLpost is to transform the previously obtained intermediate XML-based format into a MathML representation [39], currently largely used for embedding mathematical content into the XHTML web format [1].

In our work we will use three different formats for representing mathematical expressions: a presentational representation and two specific LaTeXML

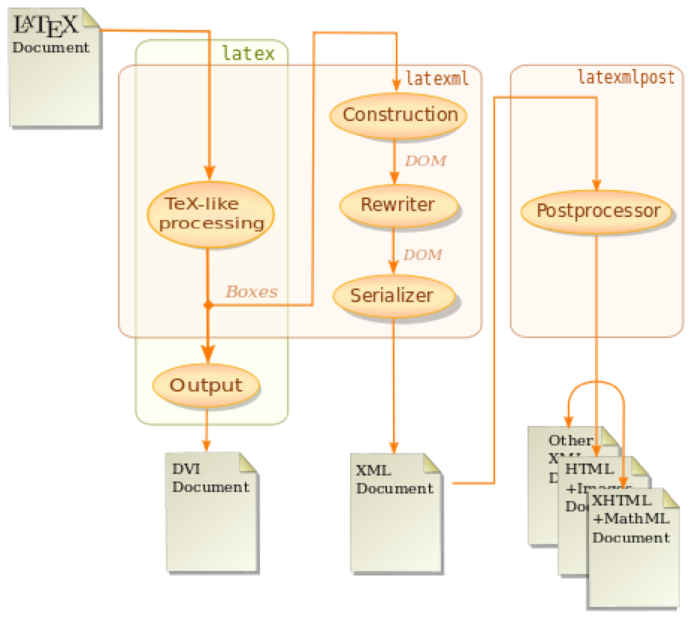

Figure 2.1: The LaTeXML system architecture (from [24])

representations, which we below describe in detail.

2.2.1.2 Mathematical expression representation formats

The following example will guide the presentation of the LaTeXML formats which are of interest for our semantic investigations:

> "*. . . Let $\mathcal{D}$ be the group of smooth diffeomorphisms of* Σ *, and* $\mathcal{D}_0$ *the group of diffeomorphisms homotopic to the identity map . . . The discrete group* $\mathcal{D}/\mathcal{D}_0$ *is called the mapping class group . . .* " from [41]

Below we present the main formats output by the LaTeXML program, complemented with examples of the LaTeXML translations corresponding to the LaTeX source `${\cal D}/{\cal D}_0$` of the expression $\mathcal{D}/\mathcal{D}_0$.

1. `Linear representation` (Figure 2.2), `noparse.xml:` Mathematical expressions are represented via a linear sequence of tokens, with the explicit requirement for LaTeXML not to generate any semantic parse tree beyond the token level (unless the semantics is explicitly encoded in the LaTeX source).

```
<Math mode="inline" tex="{\cal D}/{\cal D}_{0}" xml:id="S1.p3.m6">
        <XMath>
                <XMTok role="UNKNOWN" font="caligraphic">D</XMTok>
                <XMTok meaning="divide" role="MULOP" style="inline">/</XMTok>
                <XMTok role="UNKNOWN" font="caligraphic">D</XMTok>
                <XMApp role="POSTSUBSCRIPT" scriptpos="2">
                        <XMArg rule="Subscript">
                                <XMTok meaning="0" role="NUMBER">0</XMTok>
                        </XMArg>
                </XMApp>
        </XMath>
</Math>
```

Figure 2.2: LaTeXML-specific linear representation of the expression $\mathcal{D}/\mathcal{D}_0$

2. `Derivation tree representation` (Figure 2.3), `tex.xml`: This format is equivalent to `noparse.xml` with the exception that the mathematical expressions are parsed and derivation trees are generated using a grammar and simple heuristics. The main purpose of the `tex.xml` representation is to enable the LaTeXMLpost processing.

```
<Math mode="inline" tex="{\cal D}/{\cal D}_{0}" xml:id="S1.p3.m6" text="D / D _ 0">
        <XMath>
        <XMApp>
                <XMTok meaning="divide" role="MULOP" style="inline">/</XMTok>
                <XMTok role="UNKNOWN" font="caligraphic">D</XMTok>
                <XMApp>
                        <XMTok role="SUBSCRIPTOP" scriptpos="post2"/>
                        <XMTok role="UNKNOWN" font="caligraphic">D</XMTok>
                        <XMTok meaning="0" role="NUMBER">0</XMTok>
                </XMApp>
        </XMApp>
        </XMath>
</Math>
```

Figure 2.3: LaTeXML-specific derivation tree representation of the expression $\mathcal{D}/\mathcal{D}_0$

3. `Presentational representation` (Figure 2.4): In addition to the `linear` representation and the `derivation tree` representation LaTeXMLpost provides additional MathML and/or OpenMath representations of the mathematical expressions. Our point of interest lies in the Presentation MathML representation. Since the current version of the transformation of mathematical fragments to Content MathML is still prone to errors, we avoid using it in our semantic investigations. Despite the fact that the structure of mathematics in Presentation MathML is ambiguous, we prefer this format to Content MathML.

```
<m:math display="inline">
        <m:mrow>
                <m:mi mathvariant="script">D</m:mi>
                <m:mo>/</m:mo>
                <m:msub>
                        <m:mi mathvariant="script">D</m:mi>
                        <m:mn>0</m:mn>
                </m:msub>
        </m:mrow>
</m:math>
```

Figure 2.4: Presentational representation of the expression $\mathcal{D}/\mathcal{D}_0$

Figure 2.2 and Figure 2.3 show a specific LaTeXML representations of mathematical expressions explained above. The *<XMath>* element is a container for an internal representation, *<XMTok>* represents a general mathematical token, *<XMApp>* is a generalized application of a function or operator, and *<XMArg>* represents an argument of a superior element (*<XMApp>*, in this case). Figure 2.4 shows the MathML representation obtained after running the
LaTeXML postprocessor. The top-level element is represented using a *<m:math>* element, *<m:mrow>* indicates a horizontal row of fragments of a mathematical expression, *<m:mi>* represents an identifier element, *<m:mo>* corresponds to an operator element, *<m:msub>* to a subscript mathematical expression, and *<m:mn>* to a numerical element.

2.2.2 The corpus subset

This section presents the preparation of the data used in the experiments described in this thesis. In selecting a corpus subset, we performed a three-step procedure consisting of the following operations:

1. **Document conversion:** The documents contained in the *Mathematics* area of the ARXIV collection were translated into the corresponding `linear`, `derivation tree`, and `presentational` representations, following the procedure described in [33, 6].

2. **Document selection:** From the above-mentioned collection of documents, we randomly selected a corpus subset consisting of 10, 000 articles in which no errors were encountered during the conversion.

3. **Document preprocessing:** the documents contained in the corpus subset were further preprocessed for the experiments in the following stages:

- We word- and sentence-tokenized the documents, stemmed the words, and normalized the mathematical expressions by replacing them with unique identifiers;
- We stored the mappings between the unique identifiers of all the mathematical expressions occuring in our subset of documents and their corresponding `linear`, `derivation tree`, and `presentational` representations (Figures 2.2, 2.3, and 2.4 respectively).

Our corpus subset consists of $10,000$ documents in which the mathematical fragments are represented in the above-mentioned three formats.

2.3 Summary

In this chapter we outlined the way we collected and preprocessed the data for our experiments. We presented the arXiv.org e-Print Archive and the LaTeXML system. We then detailed how we used both in order to obtain the data set which we will extensively use in our research.

In the next chapter we will discuss the key phenomena identified in the newly-created subset of the arXMLiv corpus of mathematical scientific texts. We will focus on discussing the interplay between the symblic and the textual mathematics.

Chapter 3

Discourse Phenomena Involving Mathematical Notation

3.1 Introduction

In this chapter we investigate how mathematical objects are specified, referenced and used in mathematical discourse. We first discuss the structural and semantic aspects of symbolic notation. We then present the types of ambiguity occurring in mathematical documents. We finally outline the main sources of knowledge that are used in interpreting symbolic expressions when reading a mathematical text. The corpus analysis we conducted lead us to the identification of a series of interesting phenomena which motivated the disambiguation approach preposed in this thesis.

3.2 Mathematical notation

Mathematical notation is a language that uses symbolic expressions to represent mathematical objects. The vocabulary of the symbolic language of mathematics consists of a large variety of symbols: numbers, letters from different alphabets, operators, punctuation marks, etc. The grammar of the symbolic mathematical language contains syntactic rules that specify how symbols from the vocabulary may be arranged in certain configurations to form meaningful mathematical expressions.

Below we briefly discuss two important aspects of mathematical notation: the (syntactic) structure and the compositional semantics of mathematical expressions.

Structure In mathematical documents, symbolic expressions are written or graphically rendered in a two-dimensional layout. However, mathematical expressions may be also analyzed by means of structured symbolic constructs. The process of parsing a mathematical expression aims at recursively decomposing it into mathematical subexpressions, according to the production rules specified by the grammar of the symbolic language. The result of the parsing process is a tree-based representation of the internal structure of the mathematical expression, in which the nodes represent subexpressions and the links correspond to production rules.

Semantics In symbolic expressions, the choice of notation (i.e. the symbols and the way they are combined) reflects semantics. The denotational semantics of a mathematical expression can be constructed using the production rules specified by the grammar of the symbolic language. Discovering the compositional semantics of a mathematical expression means identifying the meaning of each symbol (i.e. the type of object it denotes) and inferring the semantics of the larger expression using grammar rules.

As we will see in the next section, ambiguous rules in the grammar of the symbolic language lead to the generation of multiple parse trees corresponding to the same mathematical expression.

3.3 Ambiguities in mathematical notation

A mathematical expression is ambiguous if it can be interpreted in at least two distinct ways. Ambiguities in mathematical notations may lead to false interpretations of symbolic expressions, as mathematicians tend to leave in some of the ambiguous symbolism without explicit interpretation. Intricacies in understanding mathematics in scientific documents appear when authors make the assumption that readers are already familiar with the conventions in use and thus they are able to easily recover the intended meaning of the mathematical expressions.

We identified two types of ambiguity occurring in mathematical documents: 1) structural ambiguity and 2) semantic ambiguity. Below we present several examples that illustrate typical situations of ambiguity in mathematical notation.

Structural Ambiguity Structural ambiguity in mathematical notation arises when there is more than one possible way of parsing a mathematical expression. The elision of multiplication signs is a frequent source of structural ambiguity in mathematical notation. For example, it is not clear whether the expression $\sin x - \pi/4$ denotes the application of the sin function to the argument x and the subtraction of the result by $\pi/4$, or the application of the *sin* function to the argument $x - \pi/4$. The omission of parentheses in the expression $\tan\ x/y$ makes it difficult to identify the argument of the function tan and thus indicating the correct reading between $\tan(x/y)$ and $(\tan x)/y$.

Semantic Ambiguity A mathematical notation is semantically ambiguous if it may have multiple meanings, depending on the different uses (i.e. the internal tree structure is the same, while the subexpressions and hence the entire expression can be interpreted in different ways). For instance, the expression ω^{-1} is semantically ambiguous, in the sense that its meaning can be interpreted either as an inverse function, if ω is known to be a function, or as $1/\omega$, if ω is a scalar. The mathematical expression a^i may refer both to the i-th element of a sequence (i.e. an algebraic object) or to the element a raised to the power i (which could be a number object). The $u \cdot v$ notation may be understood both as a multiplication of the scalars u and v (i.e. a number), or as a dot product (a vector, i.e. an algebraic object).

In general, without domain knowledge, it is not possible to identify the internal structure of mathematical expressions by only making use of their presentational layout (i.e. their surface structure). Yet, in most cases, a reader can almost immediately decide which reading is intended. The following sections discuss the types of knowledge which is used in the disambiguation process.

3.4 Sources of knowledge in disambiguation

We identified two main sources of knowledge employed in the process of establishing the correct reading of symbolic expressions: the conventions in use and the context of mathematical expressions.

3.4.1 Conventions

In mathematics there exist a number of notational conventions which are frequently employed by the authors of scientific texts. Mathematical symbols that follow these general conventions are usually used in the discourse without being previously specified. Several examples of such general conventions are:

- capital letters usually denote sets;

- variables or unspecified quantities are usually denoted by letters at the end of the alphabet, i.e. x, y, u, v;
- constants or known values are usually denoted by letters at the beginning of the alphabet, i.e. a, b, c;
- the concept of *area* is denoted by S, while V specifies the *volume*;
- $\vec{i}$ and $\vec{j}$ refer to unit vectors;
- letters of the same alphabet specify the same category of mathematical objects, i.e. if α is used as a parameter, then β or γ are likely to also denote parameters.

Mathematical objects may also have different notations, depending on the domain in which they are used. For instance, $\sqrt{-1}$ is denoted by i in complex analysis, while engineers use the letter j.

Knowledge of operator precedence[1] is another type of convention which may help the disambiguation process. A mechanism of establishing operator precedence may be also used in the automated disambiguation of symbolic expressions, by decomposing the expression according to their operators' decreasing precedence. For instance, assuming that the operator $\rhd$ has a higher priority than the operator $\lhd$ would prevent us from considering $x \lhd y$ as a subobject of $x \lhd y \rhd z$. In general, it is not possible to automatically infer new operators precedence unless this information is explicitly stated in the discourse.

3.4.2 Context

As the context of a symbolic expression one can understand the set of information concerning that expression. This information can be divided into two types: a priori knowledge of the mathematical domain in which the given expression is used, and knowledge provided by the linguistic context.

Mathematical domain context Mathematics is organized into mathematical subfields, each of which defines its concepts of interest (i.e. mathematical objects relevant in the domain), their properties, and the relations between them. Mathematical language includes words that have terminological meaning in mathematics. This domain-specific knowledge has a key role in recovering the information that is not explicitly introduced in the discourse. Understanding mathematical texts usually involves both recognizing the explicit and

[1]defined in [9] as "...a ranking of mathematical operators, to indicate the priority of evaluation of operations"

the implicit mathematical knowledge of a specific mathematical sub-domain. For instance, let us assume that, in some document, $\oplus$ has been introduced as an operator that inherits the properties of addition in complex numbers. Mathematical domain knowledge allows us to deduce that $\oplus$ is a *binary operator* on *complex numbers*; thus, if further in that document we encounter an expression $z_1 \oplus z_2$, while neither z_1 nor z_2 have been explicitly introduced, we can reason as follows: the operator $\oplus$ is applied to complex numbers, thus both z_1 and z_2, as well as the result of the evaluation are complex numbers.

Apart from a priori notational conventions, mathematical discourse often contains statements that introduce new mathematical objects explicitly, e.g. *"Consider M a positive number*", and provide information about the internal tree-based structure of mathematical objects. Symbolic expressions may also be referenced by natural language expressions (i.e. *"this value"*, *"the constant"*), named using other mathematical objects (i.e. *"Let D denote the above expression"*), or renamed within a limited portion of text (i.e. *"For the rest of the proof of this lemma, let us denote p_1 by p_1^d in order to stress the dependence on the dimension d"*).

Linguistic context Mathematical documents are structured discourses subdivided into units such as definitions, propositions, lemmas, theorems or proofs which provide the linguistic context of symbolic expressions. As the mathematical discourse consists of both natural language and embedded symbolic constructions, the cotext[2] often provides information about the meaning of symbolic expressions.

In explicit declarations of mathematical expressions, mathematical objects may be introduced either in a main clause (e.g."*Let D be an open nonempty set*") or in a subordinate clause (e.g. "*. . . , where D is an open nonempty set.*"). Introducing a mathematical object with assumption (e.g. "*Suppose that $p = 2n$ is a prime number, where $n \geq 2$*") shows an uncertainty regarding the existence of the object. There are situations in which mathematical objects are introduced without a symbolic expression ("*Consider the joint density corresponding to the random variables X and Y*").

The following two examples show how the linguistic context of mathematical expressions can be used for inferring their meaning. Consider the following two ways of specifying the type of a mathematical expression in the expression's left linguistic context:

[2]The textual context which surrounds an expression, i.e. the words preceding and following it.

> “ *The transition density* $p_D(t, x, y)$ *is defined by . . .* ” [7]

and the right context:

> “ *. . . , where* $p_D(t, x, y)$ *is the transition density.*”

In the first example, the symbolic expression $p_D(t, x, y)$ is an appositional modifier of the noun phrase (NP) “*The transition density*”, while in the second example, it is the subject of a copular subordinate clause, identifying it with the concept “the transition density”. Using both the linguistic context and prior knowledge of mathematical domain, we can infer that the above mathematical expression denotes a function.

The following example:

> “ *. . . there exists an orthonormal basis of eigenfunctions* $\{\varphi_n\}_{n=1}^{\infty}$ ”

shows that the meaning of the appositional symbolic expression $\{\varphi_n\}_{n=1}^{\infty}$ can be identified using the structure of the noun phrase “*an orthonormal basis of eigenfunctions*”. More precisely, we observe that the syntactic structure of the above noun phrase[3] and the high level structure of the mathematical expression $\{\varphi_n\}_{n=1}^{\infty}$ are analogous. By high level structure we mean the structure modulo the content of the sub-/super-script elements. A syntactic analysis of the noun phrase “*an orthonormal basis of eigenfunctions*”, followed by a structural analysis of the symbolic expression[4] reveals that the symbol φ_n denotes an “*eigenfunction*”, while the symbolic expression as a whole, $\{\varphi_n\}_{n=1}^{\infty}$, denotes the object specified by the head noun of the NP, “*basis*”. Further analysis of the structure of appositional constructions would be needed in order to find other similar interpretation patterns.

In the example below:

> “*Let* $0 < \epsilon < 1/2$*,* $r > 0$ *and suppose that G is a* ***measurable*** (ϵ, r)***-good subset*** *of* ∂D*,* ***the boundary of a Lipschitz domain****.*”

in order to infer that the symbol ∂D denotes a *boundary* (of a Lipschitz domain), we need to analyse the right context of the symbol ∂D. We will address this issue in detail in Chapter 5.

The conclusion of our corpus investigation is that the linguistic context of mathematical expressions does show potential in the automated disambiguation task. To what extent this linguistic context suffices, we will verify in

[3](NP (NP an orthonormal basis) (PP of (NP eigenfunctions)))

[4]φ_n is the subexpression of $\{\varphi_n\}_{n=1}^{\infty}$ which contains the dominant symbol φ

experiments described in Chapter 6. We will distinguish mathematical expressions whose meaning we can infer using their left and right textual context.

3.5 Summary

In this chapter we presented the key phenomena we identified while analysing our sub-corpus. We discussed symbolic notation in mathematical discourse and outlined the two types of ambiguities in mathematical notation. We furthermore presented the sources of knowledge in the disambiguation of mathematical fragments, with an emphasis on the potential of linguistic context.

In the next chapter we will present a broad investigation of symbol declaration statements and an automated method to identify symbol declaration statements in mathematical writing.

Chapter 4

Symbol Declaration in Mathematical Discourse

In this chapter we present a method to automatically identify symbol declaration statements in mathematical writing. We first outline the purpose of our investigation and establish the connection with the task of mathematical expression disambiguation. We then introduce three quantitative studies on symbol declaration statements in mathematical texts and present our findings. As a result of these studies, we collect symbol declaration statements which we further use in a learning approach to automatically discover symbol declaration patterns. Finally, we perform an evaluation of the proposed method of finding declaration patterns and draw conclusions from our studies.

4.1 Motivation

The purpose of our investigations is three-fold. First, performing systematic quantitative analyses on explicit declarations of object denoting symbols could provide insights about the potential of using symbol declaration statements for inferring the meaning of larger mathematical expressions. Due to the richness in the syntax of symbolic language of mathematics and the sheer number of symbolic expressions used by mathematicians, we expect that a large number of notational conventions may be explicitly introduced in mathematical statements. Second, we want to investigate to what extent is it possible to automatically detect symbol declaration statements. Third, as an extension of our previous aim, we want to explore automated means to identify the concept denoted by a mathematical symbol based on its declaration statement.

4.2 Investigation of symbol declarations

The investigation of symbol declaration statements is prompted by the intuition that a significantly large number of mathematical symbols are explicitly introduced in the mathematical discourse. We are interested in studying to what extent automated understanding of mathematical symbols can be supported by an analysis of both global discourse context and locally co-occuring symbolic expressions of similar structure.

4.2.1 Experiments

We performed three corpus-based studies on simple object denoting symbols. By *"simple"* symbols, we refer to atomic identifiers and super- or subscripted atomic identifiers; we do not, however, analyse the expression(s) in the super-/sub-scripts. For instance, the expression α_i is a *"simple"* symbol, while the expression $1 - \alpha_i$ is not. In the following three studies, the term *simple mathematical expression* will be used to refer to this class of symbols. We extract simple mathematical expressions from documents based on their `derivation tree` (see Figure 2.2) and the `presentational` (see Figure 2.4) representations.

In the first and the second study, we distinguished between two types of symbol declarations. The first type is encountered when symbols are explicitly introduced in isolation, as in the fragment:

> *"Let F be a Hermitian vector bundle over W . . . ".*

We will refer to this type of declaration as *unqualified*. The other type occurs when the declared symbol is embedded in a larger symbolic expression which additionally elaborates the properties of the object denoted by the symbol, as in:

> *"Consider the cylinder $U = M \times [-\epsilon, 0)$ . . . ",*

where U is further qualified to have a certain property; we will further refer to this type of declaration as *elaborated*. The reason why we distinguished between the two declaration types is that the elaborated declarations may be more difficult to process by automated means.

One must, namely, first recognize that an expression with a surface structure of a formula is in face used in the linguistic context as a term. The right side of the formula $U = M \times [-\epsilon, 0)$ has a role of a post-modifier: "the cylider $U = M \times [-\epsilon, 0)$" is interpreted as "the cylider U" (which is) defined on $M \times [-\epsilon, 0)$.

Below we present the three studies in detail.

4.2.1.1 The first study

In the first study, we aimed at investigating to what extent mathematical symbols are properly declared in mathematical texts. For this study, we used a randomly collected set of $2,500$ sentences, each of them containing at least one simple mathematical expression. The sentences were extracted from a random set of 50 documents from our corpus subset in the following way: from each document we collected the sentences containing the first 5 occurences of 10 randomly extracted simple mathematical expressions. For each of the 500 simple expressions, we manually checked whether is explicitly declared among its first 4 occurences or later.

4.2.1.2 The second study

In the second study, we analysed the declaration statements of simple expressions which are embedded in complex expressions. In particular, we studied the case of those simple expressions 1) whose grammatical role has been left unresolved by the LaTeXML processing and 2) which occurred in isolation in the discourse.

When building the derivation tree of an expression, the LaTeXML processor attempts to assign a `role` attribute to each node of the expressions's XMATH representation it produces. The "grammatical role", specified in the `role` attribute of the XMTok tag of the `derivation tree`, captures the syntactic nature of a symbol, i.e. the syntactic role that the object takes within an expression. The `role` attribute is used in generating the Presentation MathML markup and it may also help drive the derivation of the semantics of an expression. Examples of `role` attributes which the LaTeXML parser does recognize are shown in Table 4.1 [23]. The unrecognized symbols are assigned the `UNKNOWN` attribute by default, as illustrated in Figure 4.1.

Role	Description of role
`ATOM`	a general atomic subexpression
`APPLYOP`	an explicit infix application operator
`RELOP`	a relational operator
`ADDOP`	an addition operator
`INTOP`	an integral operator

Table 4.1: Examples of `role` attributes recognized by LaTeXML

The setup of this study consisted of a subset of 100 mathematical documents

```
Let
<XMath>
    <XMApp>
        <XMTok role="SUBSCRIPTOP" scriptpos="post2"/>
        <XMTok role="UNKNOWN" font="italic">C</XMTok>
        <XMTok role="UNKNOWN" font="italic">i</XMTok>
    </XMApp>
</XMath>
be the closed convex hull in
<XMath>
    <XMTok role="UNKNOWN" font="italic">Y</XMTok>
</XMath>
of the tail end of the sequence.
```

Figure 4.1: Example of LaTeXML output

which were radomly extracted from the corpus subset. From each of the 100 mathematical documents, we randomly selected 3 mathematical expressions whose LaTeXML-specific representations contained at least 3 simple expressions. We then identified all the simple sub-expressions of the resulting 300 expressions, which were tagged as `UNKNOWN` in the `derivation tree` representation and which *also* occurred independently in the discourse. For instance, for the expression $\rho =< \omega_i, \lambda >$, we would have extracted the following simple expressions: ρ, ω_i, and λ. Then we would check which of the simple expressions were assigned an `UNKNOWN` tag and also occurred in isolation in the document, and select those for the analysis. Manual analysis of the extracted instances was performed in a similar way to the first study.

4.2.1.3 The third study

In the third study, we aimed at finding out whether structurally similar symbols occuring in the same local discourse context are also semantically related. As structurally similar symbols, we considered simple mathematical terms that obey the following two constraints: they share the same root/top-level node in the expression tree, and they are structurally similar modulo the structure of the subscript and superscipt terms. For instance, the following two symbols are structurally similar according to our criteria: ω, ω_i and ω_{n-1}. By contrast, P_c^2 and A_n^k are not similar because they differ in the top-node operator. We were interested in checking whether structurally similar symbols within local discourse context are semantically related, i.e. they denote the same type of object.
The motivation underlying this study is that the automated identification of semantically related pairs of terms could be used in an approach to constructing sets of mathematical expressions which denote the same mathematical

Category		Occurrence	n (%)	N (%)
Explicitly declared	*unqualified*	1st	290 (58%)	337 (67.4%)
		2nd	15 (3.0%)	
		3rd	11 (2.2%)	
		4th	6 (1.2%)	
		5th	2 (0.4%)	
	elaborated	–	13 (2.6%)	
Not explicitly declared				134 (26.7%)
Other				30 (5.9%)

Table 4.2: Results of the first study on symbol declaration

concept. More precisely, n symbolic expressions would form a set if each of the possible C_n^2 pair-wise combinations fulfilled the above-mentioned conditions. Consider, for instance, the (unordered) pairs of simple mathematical expressions: (c, c_1), (c_2, c_1), and (c_2, c) which fulfill the criteria. They form a set $\{c, c_1, c_2\}$. Assuming that the expression c has been previously interpreted, for instance, as a constant, the expressions c_1 and c_2 are likely to have the same mathematical interpretation.
The setup of this study consisted of 25 randomly selected mathematical documents. From each section of these documents we extracted all the pair-wise combinations of simple mathematical expressions which shared the same root symbol (same identifier) and either had the same surface structure or one expression was embedded in the other; 496 such pairs were extracted. We performed a manual analysis on the extracted pairs in order to decide which of them denote the same mathematical concept (or different instances of the same concept) in the context of the section scope.

4.2.2 Results

Tables 4.2 through 4.4 show the results corresponding to the three studies. We report absolute and percentage counts (columns labelled 'N(%)' and 'n(%)') for the two subcategories of explicit declarations (the first column of Table 4.2 and Table 4.3) and, respectively, for the location of the declaration (the occurrence at which the symbol is explicitly introduced in the discourse).
For the fist study, approximately 67% of simple mathematical expressions were explicitly introduced in the discourse, and, in most cases (58%), the first occurences in documents were within explicit declarations. There are though

Category		Occurrence	n (%)	N (%)
Explicitly declared	*unqualified*	1st	331 (53.5%)	449 (72.5%)
		2nd	22 (3.5%)	
		3rd	23 (3.7%)	
		4th	7 (1.1%)	
		5th	20 (3.2%)	
	elaborated	–	46 (7.4%)	
Not explicitly declared				170 (27.5%)

Table 4.3: Results of the second study on symbol declaration

Category	N (%)
Same concept	441 (88.9%)
Different concept	28 (5.6%)
Not classifed	27(5.4%)

Table 4.4: Results of the third study on symbol declaration

relatively infrequent cases when the declarations come after the second occurence or are elaborated (only about 3% of occurrences were elaborated by means of a symbolic expresson). In 6% cases we encountered processing errors or were not able to distinguish how an object was declared.

The results of the second study indicate that about 72% of simple sub-terms of complex expressions, which were left unexplained by the LaTeXML procesing, have been explicitly introduced in the discourse. We can observe a similar pattern regarding the declaration within the first occurence; though the cases in which we encountered elaborated declarations were more frequent in comparison to the first study.

In both the first and the second study, about 27% of the symbols were not declared in the documents. We expect that inferring their roles by automated means would require more sophisticated approach based on the context of the other occurrences.

The results of the third study indeed show that about 89% of locally occurring structurally similar expressions which share the root identifier are semantically related. We were unable to relate the expressions in 5% of the cases.

4.2.3 Conclusion

The results confirm the intuition about mathematicians' tendency to formulate explicit declarations of symbols denoting new mathematical concepts. The fact that approximately 67% of simple mathematical expressions we analysed in the first study were explicitly introduced in the discourse motivates us to pursue further investigations on the mathematical language used in symbolic expression declarations. We will use the symbol declaration statements extracted in the first and second study to discover further declaration patterns in the corpus subset. Below we describe the approach.

4.3 Identifying declaration statements in mathematical documents

In this section we present a method to bootstrap new symbol declaration statements from a corpus subset. Identification of symbol declaration statements is meant to provide a step towards inferring the meaning of mathematical symbols by automated means. For the disambiguation purpose, the importance of the automated acquisition of symbol declaration statements resides in facilitating the automated identification of the mathematical concepts denoted by those mathematical symbols which are explicitly introduced into the discourse. Below we describe the bootstrapping method in detail.

4.3.1 Method

The method aims at identifying linguistic patterns used to express explicit declarations of mathematical symbols. It consists of an iterative learning approach based on a bootstrapping procedure. More precisely, the method finds new declaration patterns by bootstrapping them from a set of already identified declaration statements. We start with an initial set of symbol declaration statements which we identified in the first two studies (see Section 4.2.1.1 and Section 4.2.1.2), and construct regular expressions which generalize their linguistic context. Next, we use the obtained regular expressions as an initial set of seeds to identify other declaration statements. We add new patterns to the seed based on the retrieved sentences. Once new patterns are created, are automatically included in the further iterations of the bootstrapping process. Below we present the bootstrapping method and describe the data preprocessing steps.

The Approach: The complete bootstrapping pipeline consists of three main steps: data preprocessing, the creation of the seed set and the bootstrapping iterations. We present a summary of each step as follows:

1. **Preprocessing**: We normalized all the documents contained in the corpus subset by replacing the mathematical domain term candidates and

Normalized symbol declaration statements
Let ME denote TERM (TERM) of ME and ...
TERM is denoted by ME , where ME is ...
If we denote by ME TERM from ME to ME , ...
Let ME be infinite TERM and ...
Suppose that ME and ME are TERM and ...
Consider ME as TERM ...

Table 4.5: A sample of the normalized declaration statements

the unique identifiers of mathematical expressions with generic tokens (*TERM* and *ME*, respectively). In order to automate the identification of mathematical term candidates, we implemented an adapted version of the algorithm presented in [8, 15]. We did not pursue an evaluation of the term identification approach, therefore we expect that some of the errors in finding declaration statements might be due to the errors resulting from the term identification step.

We exemplify the result of the preprocessing step by the following sample sentence:

"Let H be a separable Hilbert space . . . " from [11]

After applying the above presented steps, we obtain its normalized form:

Let ME be TERM ...

2. **Seed Set**: The initial set we used in the bootstrapping process was constructed using 250 declaration statements manually identified in the studies discussed in the previous section. We then created normalized versions of the declaration statements (see Table 4.5 for a sample of such normalized versions of declaration statements). In order to create the initial seed of patterns, we analysed the linguistic verbalizations of the extracted symbol declaration statements and created regular expressions which correspond to them. The seed set obtained from these

Seed Set
[[b|B]oth|where|[s|S]ince]*ME and ME are [a-zA-Z0-9-]{0,3}TERM
[[d|D]enote|[d|D]efine]+ [a-zA-Z0-9-]{0,3}TERM and TERM as ME and ME
[[d|D]efine|[w|W]rite]+ [a-zA-Z0-9-]{0,3}TERM ME and ME
TERM [of [a-zA-Z0-9-]{0,3}]*[are|is|,]+ [given|denoted]+ by ME
TERM [such as|for]+ ME
[[d|D]efine|[w|W]rite|[i|I]dentify|[i|I]dentifying|[u|U]se]+ ME [as|for]+ [a-zA-Z0-9-]{0,3}TERM
ME [which is|denotes|to denote|is|represents|stand for]+ [a-zA-Z0-9-] {0,3}TERM
[l|L]et [a-zA-Z0-9-]{0,3}*ME [and ME]*[denote|be]+ [a-zA-Z0-9-] {0,3}TERM
[l|L]et [us]* [denote|write]+ [a-zA-Z0-9-]{0,3}TERM by ME
[d|D]enote by ME [a-zA-Z0-9-]{0,3}TERM
[b|B]y ME we denote [a-zA-Z0-9-]{0,3}TERM
[[s|S]uppose|[c|C]onsider]+ [that]*ME [and ME]*[is|are|as]+ [a-zA-Z0-9-]{0,3}TERM
[c|C]onsider [a-zA-Z0-9-]{0,3}TERM ME as [a-zA-Z0-9-]{0,3}TERM

Table 4.6: Initial symbol declaration patterns

lexical patterns is shown in Table 4.6[1]. The bootstrapping process is initiated for the lexical patterns contained in the seed set.

3. **Bootstrapping new patterns**: We discover new patterns by using the *anchored patterns* approach similar to the one proposed in [18]. More precisely, the authors use doubly-anchored patterns of type:

```
<class name> such as <class member> and *
```

in order to identify instances of a specific class in text. This pattern is instantiated with both the name of the class to be learned (class name) and a member of the class (class member). As an example, the extraction of automobile names may be performed using the pattern: "CARS such as FORD and *".

[1]The patterns are written using the conventions of the Java Regular Expression language, available at: http://java.sun.com/docs/books/tutorial/essential/regex/index.htm

The anchored patterns in our study were obtained from regular expressions by fixing parts of the seed set of patterns. More precisely, we fixed the domain term tokens and the mathematical expressions (in both the normalized and the unnormalized versions), by instantiating them with occurences of mathematical expressions found in the sentences which matched a pattern from the seed set. The purpose of incorporating anchored patterns in the bootstrapping approach is to avoid overgeneration. Newly found patterns are first manually converted into regular expressions, added to the current seed set and then bootstrapping is repeated. As the bootstrapping progresses, each already identified lexical pattern is used for finding new patterns in the following iterations. The patterns we obtained after 85 bootstrapping iterations are shown in Table 4.8.

4.3.2 Evaluation and results

To evaluate the quality of the automatically identified declaration statement patterns, we used a subset of 250 documents out of the remaining documents. Note that, this set is disjunct from the document subset used for the identifying patterns. We then randomly extracted $1,000$ sentences as a test set (we discarded 41 cases of symbolic expressions in which we encountered different processing errors). Next, we manually identified those sentences which contained at least one symbol declaration statement. We obtained 374 such sentences. Then, we ran the algorithm on the test set and compared the results against the manual annotations. We report the obtained results in terms of precision and recall, two standard set-based performance measures. Precision is the number of correctly identified declaration statements (287) divided by the total number of declaration sentences identified by the approach (322), while recall is the proportion of correctly identified statements over the number of all declaration statements (374).Table 4.7 shows the results of the declaration statement bootstrapping study.

4.3.3 Discussion

We obtained a precision of 89.13% (287 from a total of 322 cases were correctly identified) and a recall of 76.73%. The results encourage us to continue the development of the bootstrapping approach. We plan to take into account a larger initial set of seeds and we expect to improve the current value of the recall by running more iterations of the boostrapping algorithm.

Error Analysis We analysed the approximately 11% cases in which the bootstrapping overgenerated declaration statements. In 63% of the cases (22), we noticed that the patterns are too permissive. More precisely, we identified two main sources of systematic errors in building the regular expressions:

- considering words from the left context of mathematical terms as modifiers; for instance, in the example “Since ME is in TERM ...”, the preposition “in” changes the meaning of the statement, while the regular expression is unable to filter it out;
- assuming that verbs matching regular expressions always represent a link between a mathematical concept and a mathematical expression; examples: “TERM are called ... if ME” and “TERM which is ergodic for ME”;
- assuming that a TERM denotes a mathematical expression, without checking the left context of the mathematical expression, as in: “Denote by TERM the TERM of ME”.

The remaining 13 cases consisted of errors of recognizing mathematical terms, also combined with one of the above three situations. For instance, in the example “Denote the minimal edge in the TERM by ME”, our approach fails to identify the term “minimal edge”. As an immediate improvement of our bootstrapping method, we plan to analyse the left contexts of both the mathematical expressions and mathematical concepts and to filter out those cases in which we encounter prepositions such as “in” and “of”, which may change the meaning of the sentence.

Implications for the disambiguation task The results of the bootstrapping approach encourage us to pursue further analysis of symbol declaration statements, in order to develop automated means to inferring symbols’ interpretation. According to [40], the *definiendum* of a mathematical definition represents "either an adjective that denotes a property that mathematical objects may have, or it may be a noun phrase that denotes a *type* of mathematical object with certain properties". Identifying the mathematical domain term which the mathematical symbol refers to (i.e. the term generically replaced by the token TERM in the normalized declaration statements presented in Table 4.5) is essential in inferring the *type* of mathematical object the symbol denotes. We present a study on this issue in the next chapter.

Performance Measure	$N(\%)$
Precision	287/322(89.13%)
Recall	287/374(76.73%)

Table 4.7: Results of the bootstrapping approach

4.4 Summary

In this chapter we presented an approach to automatic identification of symbol declaration statements in mathematical documents. We presented an experiment consisting of three quantitative studies on simple object denoting symbols occuring in the corpus subset and reported our findings. Building on the first two studies, we developpped an automated approach to identify linguistic declaration patterns. We finally presented the results and outlined the contribution our studies may bring to automated mathematical expression disambiguation, which will be the topic of the next chapter.

Bootstrapped declaration patterns
ME [a-zA-Z0-9-]{0,3}[[which|that]*[is|are]]+ [a-zA-Z0-9-]{0,3}TERM
ME [a-zA-Z0-9-]{0,3}[is|are]+ [a-zA-Z0-9-]{0,3}TERM
[L|l]et ME [and ME]?[be|stand for]+ [a-zA-Z0-9-]{0,3}TERM
[L|l]et [a-zA-Z0-9-]+ME [and ME]* [be|stand for]+ [a-zA-Z0-9-]{0,3}TERM
ME [and ME]?[defines|define]+ [a-zA-Z0-9-]{0,3}TERM
TERM [of [a-zA-Z0-9-]{1,3}]*[is|are|,]+ denoted [by]*ME
ME [and ME]?[[to]*denote|denotes]+ [a-zA-Z0-9-]{0,3}TERM
[b|B]y ME [we|is]+ [denote|denoted]+ [a-zA-Z0-9-]{0,3}TERM
[d|D]enote [a-zA-Z0-9-]?by ME[and ME]* [a-zA-Z0-9-]{0,3}TERM
[D|d]enote TERM [and TERM]?as ME [and ME]?
[D|d]enote [a-zA-Z0-9-]?TERM [a-zA-Z0-9-]{0,3}by ME
[l|L]et [a-zA-Z0-9-]?ME[and ME]* denote [a-zA-Z0-9-]{0,3}TERM
ME [is|are]+ defined [by|as]+ [a-zA-Z0-9-]{0,3}TERM
[D|d]efine ME [to be|as]+ [a-zA-Z0-9-]{0,3}TERM
[D|d]efine [a-zA-Z0-9-]{0,3}TERM [as|to be|by]+ ME
[n|N]otation ME [or ME]+ [for|means]+ [a-zA-Z0-9-]*TERM
TERM [signifies|signify] ME
ME stands for [a-zA-Z0-9-]{0,3}TERM
ME is [said|meant]+ to be [a-zA-Z0-9-]{0,3}TERM
[[S|s]ince|[H|h]ence]+ [a-zA-Z0-9-]{0,3}TERM is ME
TERM is [just]*[of form |[denoted|given|identified] [as|by|with]]*ME
ME as [a-zA-Z0-9-]{0,3}TERM
TERM [such]*as ME
TERM [, [say]*]*[ME |(ME)] [and ME]*
ME [must|can|to]+ be [a-zA-Z0-9-]{0,3}TERM
TERM [of [a-zA-Z0-9-]{0,3}]*must be ME
ME becomes [a-zA-Z0-9-]{0,3}TERM
TERM [a-zA-Z0-9-]{0,3}[identified with |given by]+ ME
[I|i]dentify[ing]* ME [as|with]+ [a-zA-Z0-9-]{0,3}TERM
ME [represents|represented by]+ [a-zA-Z0-9-]{0,3}TERM
use ME for [a-zA-Z0-9-]{0,3}TERM
TERM [a-zA-Z0-9-]0,4[represented by|coincides with]+ ME
ME , (called [a-zA-Z0-9-]{0,3}TERM)
ME is called [a-zA-Z0-9-]{0,3}TERM
[[A|a]ssociate[s]* to|[C|c]all[s]*]+ ME [a-zA-Z0-9-]{0,3}TERM
[W|w]rite ME for [a-zA-Z0-9-]{0,3}TERM
[U|u]se [a-zA-Z0-9-]{0,3}TERM for ME
[[s|S]uppose that|[C|c]onsider]+ [TERM|ME]+ [and ME]* [is|are|as]+ [a-zA-Z0-9-]{0,3}TERM

Table 4.8: Bootstrapped patterns

Chapter 5

Incremental Processing of Mathematical Documents

5.1 Introduction

In the previous chapter we investigated automated means to indentify symbol declaration statements in scientific mathematical documents. The results of the studies presented in the Sections 4.2 and 4.3 of Chapter 4 indicate the potential of using the global discourse context in an automated strategy for interpreting and disambiguating mathematical expressions. The two main types of ambiguity encountered in scientific mathematical documents as well as particular linguistic phenomena occuring in the context of introducing and using symbolic notations in mathematical texts were outlined in Chapter 3.

In this chapter, we present an architecture supporting automated disambiguation of symbolic mathematical expressions. For this purpose, we construct a hierarchical terminological resource consisting of sets of mathematical lexical terms denoting concepts taxonomically subordinate to 9 high level mathematical object types. We first describe the process of constructing the resource. Then we present a method of using the global discourse context in the process of disambiguating simple mathematical expressions, by mapping the terms contained in the expression's context to each object type from the lexical resource. We use an incremental method of analysing the discourse context, i.e. we collect all the symbolic terms that are related to the target expression and that appear up to the specific occurence of that expression in the discourse. We finally discuss the implications of using type information in parsing and disambiguating symbolic mathematical content.

5.2 The disambiguation strategy

In this section we outline the main components of an automated approach to mathematical expression disambiguation. We begin with a short presentation of our previous study on inferring the semantics of symbolic expressions. We then discuss how the encountered challenges and limitations may be overcome in the current disambiguation approach. We finally provide a high level description of the disambiguation scenario.

Previous work and limitations In a previous study on disambiguating symbolic expressions, we have shown that the local linguistic context, within which mathematical expressions are embedded, provides a good source of information for recognizing the class to which a mathematical expression [16] belongs. The approach treated only the case of mathematical expressions which are syntactically part of a *nominal group* and, in particular, are in an *apposition relation* with an immediately preceding noun phrase. More precisely, the target expressions came from a linguistic pattern: "… *noun_phrase symbolic_math_expression* …", as in the example: "… *the inverse function* ω^{-1} …".
In the disambiguation process we used a lexical resource in which some unrelated terms were grouped in one cluster of mathematical terms. Figure 5.1 shows an extract of the mathematical terms grouped in the cluster called "property". One may notice that both "diffusion" and "concavity" appear in the same cluster, despite the fact that the first refers to a *physical property*, while the second refers to a *property of a mathematical object* (e.g. "set", "function", etc.).

`Term Cluster Name`	Mathematical Terms
`property`	associativity, commutativity, concavity, continuity, convexity, differentiability, diffusion, distributivity, goodness, linearity, noise, property, violation, …

Table 5.1: Extract from the lexical resource used in a previous disambiguation approach

Another important limitation resided in the fact that only the *left* context of a symbolic expression was used in the disambiguation process, despite the fact that mathematical texts are well-known to introduce notations and concepts as they go along. As a conclusion we pointed out the need for broader discourse analysis that would detect notation introductions to be used in the disambiguation process.

The new approach The current disambiguation approach is based on the use of lexical information provided by both the global discourse context and a semi-automatically built lexical resource of mathematical concepts. In order to disambiguate a target symbolic expression, we extract all the sentences containing either the symbol or its related symbols and identify the declaration statements, following the approach described in Section 4.3. As lexical context of the target symbolic expression, we consider the mathematical concepts denoted by the target expression in the identified declaration statements, as well as the mathematical terms contained in the current and the previous sentence corresponding to the expression. For disambiguation, we consider only term-denoting mathematical expressions, i.e. we exclude those symbolic expressions which represent mathematical statements.
Our current disambiguation method computes similarity scores between the terms from the lexical context of the target expression and the terms associated with each class of a new hierarchical lexical resource we built. As the interpretation of a target mathematical expression we indicate one (or more) mathematical object types from the lexical resource whose corresponding mathematical terms are highly similar to the concepts from the lexical context of the expression. We start by introducing the lexical resource below.

5.3 Lexical resource

In this section we describe the steps we employed in order to semi-automatically generate a hierarchical lexical resource which is the core knowledge source supporting the disambiguation process. We first describe the contents of two online mathematical resources and outline the features that make them relevant to our research. We then present the way in which we extracted and processed information from both resources. We finally describe the way we organized mathematical concepts in a lexical resource suitable for disambiguation purposes.

Mathematics Subject Classification The Mathematics Subject Classification (MSC) [3] is a hierarchically organized mathematical resource containing a set of over $5,000$ classes corresponding to disciplines of mathematics. The purpose of the resource is to provide a classification of the abstracts covered by both the Zentralblatt MATH (Zbl) [42] and the Mathematical Reviews

Database (MRDB) [4]. Table 5.2 presents an extract from MSC 2010 corresponding to the area of *Sequences, series, summability*.

40-xx	Sequences, series, summability
40 Axx	Convergence and divergence of infinite limiting processes
40 Bxx	Multiple sequences and series
40 Cxx	General summability methods
40 Dxx	Direct theorems on summability
40 Exx	Inversion theorems
40 Fxx	Absolute and strong summability
40 Gxx	Special methods of summability
40 Hxx	Functional analytic methods in summability
40 Jxx	Summability in abstract structures

Table 5.2: Extract from MSC 2010

Each publication indexed by either Zbl or MRDB is associated with an MSC-specific code, *primary classification*, indicating its principal contribution to a mathematical area of use. The entry may also be assigned to several *secondary classification* numbers, that are meant to indicate contributions of the publication to other scientific fields.

Our interest in the Mathematics Subject Classification lies in the automated extraction and manipulation of the mathematical terms contained in the titles of the classifications (see Section 5.3 - Automated processing). These terms will be used in the construction of the lexical resource for mathematical expression disambiguation.

Cambridge Mathematics Thesaurus The University of Cambridge "Mathematics Thesaurus" (CMT) [38] is part of the Millennium Mathematics Project (MMP) [37]. The resource contains a number of 4583 concept names, and for each of them provides short explanations and thesaurus-like relations of types "broader/narrower" and "references/referenced". We will analyse the graph structure induced by the "broader/narrower" relations in order to generate a hierarchy of mathematical concepts. More specifically, we will combine the information from the MSC and the CMT by introducing a taxonomical structure to the concept names from the MSC based on retrieving their hypernyms from the CMT. The processing stages are outlined below.

Automated processing We first extracted the mathematical domain terms contained in the Mathematics Subject Classification using the algorithm proposed by [15]; see Section 5.4.1 for a summary of the method. We normalized the extracted terms by removing their modifiers, and obtained a set of 341 unique higher mathematical concepts, out of which 170 were also contained in the Cambridge Mathematics Thesaurus.

```
MINIMAL LENGTH PATH EXTRACTION
INPUT : Math Subj Classification (MSC) & Cambridge Math Thesaurus (CMT)
OUTPUT : Minimal Length Paths extracted from CMT
1  MathTerms := termExtraction(MSC)
2  normalize(MathTerms)
3  topNode := node in CMS s.t. it has no predecessor
4  setOfPaths := ∅
5  for each term in MathTerms
6      if term occurs in CMT
7          compute minLengthPath from topNode to term
8          add minLengthPath to setOfPaths
9  OUTPUT: setOfPaths
```

Figure 5.1: Minimal length path extraction algorithm

We further used the “broader/narrower concept” relations provided by the Cambridge Mathematics Thesaurus in order to construct its directed concept graph structure and find hypernym terms. We restricted the obtained graph to its minimal subgraph induced by the set of common higher mathematical concepts. After analysing the structure of the subgraph, we identified a node which did not have any predecessor, the *top node*. The automated processing of the two resources resulted in the extraction of the minimal length paths from the *top node* to each of the 170 concepts. The summary of the steps we employed in the extraction of the minimal length paths is presented in Figure 5.1.

The purpose of generating the minimal length paths was to provide a starting point in the process of clustering higher mathematical concepts. For instance, the extracted minimal length paths corresponding to the concepts of “monoid”, “position”, and “divisor” are shown in Table 5.3.

Sample of mathematical object type classification
Algebraic object-Set-Semigroup-Monoid
Attribute-Quality-Property-Physical property-Position
Number-Real-Rational-Integer-Divisor

Table 5.3: Examples of minimal length paths extracted from the Cambridge Mathematics Thesaurus

These paths allowed us to further manually classify the concepts into specific object types, e.g. “monoid” as an *algebraic object*, “position” as a *qualitative attribute*, “divisor” as a *numer object*. Below we provide the details of the classification of mathematical concepts according to their object type.

Manual processing We manually transformed the obtained paths into reduced versions of depth at most 2, i.e. each term has at most one intermediate hypernym. We obtained the following mathematical object types: *Algebraic object: General algebraic object, Algebraic object: Mapping and function, Algebraic object: Number object, Notational and logical object, Geometric object, Qualitative attribute, Quantitative attribute, Method or Process*. We then manually classified the remaining 171 mathematical concepts that were not contained in the Cambridge Mathematics Thesaurus into the obtained classes. An extract of the obtained classification is presented in Table 5.4. The complete lexical resource is presented in Table 8.1 of the Appendix.

Mathematical Object Type	Corresponding Mathematical Concepts
`Algebraic object: General algebraic object`	array, element, field, intersection, group, module, matroid, matrix, ring, category, groupoid, set, domain, neighborhood, pair, range, region, semigroup, monoid, ...
`Algebraic object: Mapping or function`	code, correspondence, function, functor, intersection, metric, morphism, order, transformation, bundle, functional, mapping, norm, operator, kernel, homomorphism, ...
`Number object`	number, quaternion, harmonic, dimension, prime, limit, index, exponent, real, error, rational, fraction, integer, divisor, factor, quotient, residue, constant, difference, ...

Notational or logical object	equation, formula, notation, symbol, variable, unknown, index, form, representation, scheme, condition, conjecture, constraint, convention, criterion, hypothesis, lemma, ...
Geometric object	curve, path, trajectory, diagram, figure, polygon, square, graph, network, lattice, tessellation, tiling, polyhedron, torus, space, ...
Method or Process	algorithm, inference, calculation, computation, inverse, method, transformation, dilation, reduction, glide, differentiation, integration, measurement, operation, ...
Qualitative attribute	concentration, position, property, invariant, symmetry, singularity, convexity, complexity, additivity, adjunction, coherence, compactness, computability, connectedness, ...

Table 5.4: Lexical resource for mathematical expression disambiguation

The newly created lexical resource which we will use in disambiguation is superior with resprect to the resource we used in the previous study in several resprects:

1. a clear relation between the higher object types and the members of the term cluster sets: in all cases the relation is of *is-a* type;
2. coherent sets of objects clustered under a more general common type which they denote;
3. a smaller number of classes.

5.4 An architecture for disambiguating symbolic expressions

The disambiguation process consists of three main components. We first preprocess the documents and identify those mathematical expressions that are candidates for disambiguation. We then compute corpus-based similarities for each class of mathematical terms contained in the lexical resource. We finally infer the semantics of each target symbolic expression based on our disambiguation strategy. Below we present the disambiguation components.

5.4.1 Preprocessing

In order to prepare the documents for the disambiguation process, we identified the sentences, the symbolic expressions and the mathematical domain terms contained in the documents. Below we present these steps in detail.

Sentence and symbolic expression identification In Section 2.2.2 we described the preparation of the corpus subset to be used in the disambiguation task. For each of the $10,000$ documents we performed word- and sentence-tokenizations, and stemmed the words. We then normalized the mathematical expressions by replacing them with unique identifiers and stored the mappings between the obtained unique identifiers and the three representation types[1] corresponding to each mathematical expression.

Domain term identification The purpose of identifying mathematical terms in the corpus subset is two-fold: mathematical domain terms play a crucial role in building our lexical resource and in identifying symbol declaration statements. In order to automate the identification of mathematical term candidates, we implemented an adapted version of the algorithm presented in [8, 15]. We computed n-gram frequency statistics over the corpus subset and discarded those n-grams that either were very infrequent, i.e. occured less than 5 times in the corpus subset, or contained a word from a predefined list of stop words[2]. The candidate terms were selected from the set of the remaining n-grams according to a score that took into account the length of the n-gram, its frequency and the number of its nested occurences in longer n-grams. More precisely, for each n-gram, we computed the following information:

- the frequency of an n-gram candidate *Ng* in the analysed text (*f(Ng)*);
- the length of the candidate in words (*l(Ng)*);
- the frequency of the candidate when it appears nested in longer candidates and the number of those longer candidates.

[1] `linear` - see Figure 2.2, `derivation tree` - see Figure 2.3 and `presentational` - see Figure 2.4

[2] The stop word list we used was non-standard and contained certain word classes which are not likely to form part of domain terms. The reason why we explicitly enumerated these was because our term identification approach does not make use of any linguistic information other than the cooccuring lexemes themselves. We therefore wanted to minimize the number of errors due to incorrect noun phrase recognition

Given a list of canditate multi-word terms (n-grams) with their associated frequencies of occurence, the algoritm ranks the n-grams according to their termhood, which is given by the formula:

$$score(Ng) = \begin{cases} f(Ng) log_2 l(Ng), \text{ if } Ng \text{ is not nested} \\ f(Ng) log_2 l(Ng) - \frac{1}{|C|} \sum_{t \in C} f(t) log_2 l(Ng), \text{ otherwise} \end{cases}$$

where Ng is the candidate n-gram, C is the set of extracted candidates containing Ng, and $|C|$ is the cardinality of the set C. On the one hand, the above formula assumes that a high frequency of occurence and a large length of a candidate n-gram positively affect its termhood. On the other hand, if a candidate frequently appears nested in other n-grams, its termhood value is penalized accordingly. We filtered out those n-grams whose computed score was below 10.

5.4.2 Statistical association measures

In the disambiguation task, we use semantic similarity measures in order to decide which object types contained in the lexical resource may be associated with the lexical context of the target symbolic expression. The word-to-word similarity metrics existing in the current literature can be divided into two main classes: knowledge-based and corpus-based measures. Knowledge-based measures of word semantic similarity compute the path length between words contained in semantic networks, dictionaries or thesauri. Corpus-based similarity methods compute statistical association measures using large amounts of text.

For the task of inferring the meaning of symbolic expressions in scientific texts, we will experiment only with statistical association measures because of the following two reasons: 1) knowledge-based methods are, in general, difficult to apply in processing specific domain texts (i.e. mathematical domain) for which there are yet no broad-coverage lexical resources, and 2) our corpus subset of $10,000$ scientific documents is sufficiently large to provide significant information to be used in computing statistical association measures between words. Our disambiguation strategy is inspired by recent computational approaches which use statistical association measures to estimate semantic relatedness between words by analysing their distributional properties. In these approaches distributional similarity is computed based on, for instance, the Wikipedia, large corpora, or even the Web (see, for instance, [10, 22, 26, 29, 35]). Below we present several corpus-based word-to-word similarity measures which we will use in disambiguation experiments.

5.4.2.1 Similarity measures

Cooccurrence-based similarity In our experiments, we use the following lexical association measures that have been previously successfully used in computational linguistics tasks [36, 13, 12] in order to estimate the relative probability with which words occur in proximity:

- Pointwise Mutual Information (*PMI*) is a standard measure of the strength of word co-occurrence (see, for example, [36]). For two words w_1 and w_2, *PMI* is defined as the *log* of the ratio of the probability that the words co-occur to the product of the individual probabilities:

  $$PMI(w_1, w_2) = \log_2 \frac{P(w_1, w_2)}{P(w_1) \times P(w_2)}$$

  If w_1 and w_2 are statistically independent, then $P(w_1, w_2) = P(w_1) \times P(w_2)$, which implies that the corresponding *PMI* score is 0. If w_1 and w_2 are not independent and tend to co-occur, then $P(w_1, w_2)$ will be greater than $P(w_1) \times P(w_2)$, which is reflected by a positive value of the *PMI* score. Because *log* is monotonically increasing, relative ordinal rankings of *PMI* estimates are preserved if the *log* is dropped.

  Concerning the effectiveness of using *PMI* relative to the size of the data, previous studies have shown that *PMI* achieves better results in comparison with Latent Semantic Analysis (*LSA*) [19], when large amounts of data is used ([12, 13]). At the same time, [21] consider *PMI* as a measure of independence rather than of dependence, being less efficient on low-frequency events. For instance, the *PMI* value corresponding to perfectly dependent bigrams:

  $$P(w_1) = P(w_2) = P(w_1, w_2)$$

  is $PMI(w_1, w_2) = -\log_2 P(w_1)$, which shows that Pointwise Mutual Information exhibits preference to rare events (i.e. $P(w_1)$ is very small) because they a priori contain higher amount of information, in comparison to frequent events. This aspect motivated us to also experiment with other measures that perform better on low-frequency events (see the mutual dependency measure presented below).

- Mutual dependency (*MD*) is obtained by subtracting from *PMI* the amount of information carried by the whole event, i.e. $-\log_2 P(w_1, w_2)$:

  $$MD(w_1, w_2) = \log_2 \frac{P(w_1, w_2)^2}{P(w_1) \times P(w_2)}$$

Adding a slight bias towards frequency was reported to produce better results in collocation extraction [27]. The obtained measure provides a more intuitive relation between *PMI*'s upper bound and occurrence frequency:

$$MD_{biased}(w_1, w_2) = \log_2 \frac{P(w_1, w_2)^2}{P(w_1) \times P(w_2)} + \log_2 P(w_1, w_2)$$

- Pearson's χ^2 test sums the squared differences between the expected and the observed frequency values, scaled by the expected frequencies in all combinations of cooccurrences or occurences with other tokens of the words w_1 and w_2:

$$\chi^2(w_1, w_2) = \frac{(f(w_1, w_2) - \hat{f}(w_1, w_2))^2}{\hat{f}(w_1, w_2)} + \frac{(f(w_1, \bar{w}_2) - \hat{f}(w_1, \bar{w}_2))^2}{\hat{f}(w_1, \bar{w}_2)} +$$

$$\frac{(f(\bar{w}_1, w_2) - \hat{f}(\bar{w}_1, w_2))^2}{\hat{f}(\bar{w}_1, w_2)} + \frac{(f(\bar{w}_1, \bar{w}_2) - \hat{f}(\bar{w}_1, \bar{w}_2))^2}{\hat{f}(\bar{w}_1, \bar{w}_2)}$$

  Pearson's χ^2 test is usually preferred to the t-test as it does not require that the assumption of normality in the probability distributions is fulfilled [21].

- Log likelihood ratio tests the similarity between two words under the assumption that the words in text have a binomial distribution [14]:

$$LLR(w_1, w_2) = -2f(w_1, w_2) \log_2 \frac{f(w_1, w_2)}{\hat{f}(w_1, w_2)} - 2f(w_1, \bar{w}_2) \log_2 \frac{f(w_1, \bar{w}_2)}{\hat{f}(w_1, \bar{w}_2)}$$

$$-2f(\bar{w}_1, w_2) \log_2 \frac{f(\bar{w}_1, w_2)}{\hat{f}(\bar{w}_1, w_2)} - 2f(\bar{w}_1, \bar{w}_2) \log_2 \frac{f(\bar{w}_1, \bar{w}_2)}{\hat{f}(\bar{w}_1, \bar{w}_2)})$$

String similarity In addition to corpus-based similarity measures, we perform a character n-gram matching for computing similarities between words seen as strings. Given two words w_1 and w_2, their corresponding *Dice* coefficient is defined as twice the number of common character bigrams divided by the total number of bigrams in the two words:

$$Dice(w_1, w_2) = \frac{2 \times |\text{Common Bigrams}|}{|\text{Bigrams of } w_1| + |\text{Bigrams of } w_2|}$$

The value 1 of the *Dice* coefficient indicates that the two words are identical. We use this measure to account for different inflectional variants in the lexical resource and the words in the target window.

5.4.2.2 Context to concept mapping

To compute word-to-word similarity scores, we combine the *Dice* coefficient with normalized values of the above presented corpus-based measures. Based on experimentation we used the *Dice* coefficient for those words with *Dice* scores higher than a predefined threshold, $\lambda \approx 0.7$ (i.e. a word from context occurs in the lexical resource), otherwise we use the corpus-based measures:

$$sim(w_1, w_2) = \begin{cases} Dice(w_1, w_2) & \text{if } Dice(w_1, w_2) > \lambda \\ \text{Cooccurence-based Measure} & \text{otherwise} \end{cases}$$

The word-to-word similarity measures will be used in the disambiguation process for computing the similarity between the terms from the context of target symbolic expression and the terms subordinate to each class in the lexical resource. We explain the details of our disambiguation approach in Sections 5.4.3.

5.4.3 Simple expression disambiguation

In this section we present a method to disambiguate simple symbolic expressions. We first define the lexical context of the simple expression to be disambiguated. We then explain the mechanism of assigning an object type to the target expression using a measure of semantic similarity. We conclude this section with a short discussion on how to use this approach in the disambiguation of complex mathematical expressions.

Context For each target mathematical expression, we consider a lexical context C consisting of two sets of mathematical domain terms:

1. terms which appear in declaration statements and are denoted by either the target expression or its structurally similar symbols (we will call this set C_1);

2. terms which appear in the current and the previous sentence corresponding to the target mathematical expression (further called C_2).

Each extracted mathematical term from the context C contributes to the similarity score proportionally to its importance in the disambiguation process. The details are provided below.

Disambiguation To infer the meaning of mathematical symbols, we use an approach inspired by methods of word sense disambiguation (WSD) [31, 34] which use inventories of word senses (i.e. known usages of words) to discriminate which sense of a word is used in a given sentence. There is a large number of WSD algorithms using various measures of semantic similarity to map the given word in context to its possible senses contained in an inventory [29, 30, 22].

Our approach to disambiguate mathematical expressions uses the mathematical object types presented in Table 5.4 as the set of possible interpretations of a given symbolic expression. In order to identify the mathematical object type which best matches the lexical context of the target expression, we map the mathematical terms from its context to the mathematical terms corresponding to each class of mathematical objects.
To accomplish this, we follow the approach presented in [22] to estimate the semantic similarity of two short text segments T_1 and T_2. In brief, for each word w in a text segment, the approach identifies the word in the other segment with which it shares the highest word-to-word semantic similarity, $maxSim(w, T_i)$, $i \in \{1, 2\}$. Each similarity score is then weighted by inverse document frequency[3] ($idf(w)$) of the corresponding word [32, 28]. The resulting scoring function is obtained by averaging the semantic similarities of each text segment in turn relative to the other:

$$sim(T_1, T_2) = \frac{1}{2}\left(\frac{\sum_{w\in T_1} maxSim(w, T_2) \times idf(w)}{\sum_{w\in T_1} idf(w)} + \frac{\sum_{w\in T_2} maxSim(w, T_1) \times idf(w)}{\sum_{w\in T_2} idf(w)}\right)$$

Our approach uses a variant of the above scoring function to estimate the semantic similarity between the mathematical terms contained in the symbol's context (C) and the mathematical terms corresponding to each object type ($Class$) of the lexical resource presented in Section 5.3. As measures of semantic similarity, we have experimented with the metrics presented in Section 5.4.2. The obtained similarity scores are weighted, summed up, and normalized by the length of the considered context. The similarity is calculated using the following scoring function:

$$Sim(C, Class) = \sum_{w\in C} maxSim(w, Class) \times cw(w), \text{ where} \tag{5.1}$$

[3]IDF is a measure of word specificity that computes the ratio between the total number of documents in the corpus divided by the total number of documents including a given word

$$maxSim(w, Class) = \max_{term \in Class} \{sim(w, term)\} \qquad (5.2)$$

where $sim(w, term)$ is our context to concept mapping measure computed according to the formula presented in Section 5.4.2.2. The resulting assigned interpretation is the object type with the highest similarity score between the lexical context and the terms from each of the sets of mathematical terms contained in the lexical resource. A pseudocode corresponding to this approach is presented in Figure 5.2.

```
ALGORITHM: MATHEMATICAL EXPRESSION DISAMBIGUATION
INPUT : targetME in a scientific document
OUTPUT: math object type(s) corresponding to the targetME
Context := ∅
if targetME is SimpleME
for each occurence of targetME
    if occurence is explicitly declared
        add referred mathConcept to the Context
select a 20-word context window W of targetME
for each mathTerm in W
    add mathTerm to Context
for each ME structurally similar to targetME
    for each occurence of ME
        if occurence is explicitly declared
            add referred mathConcept to the Context
for each mathObjectType in LexicalResource
    select mathTerms corresponding to mathObjectType
    compute sim(Context, mathTerms)
    update maxSim(Context, mathTerms)
OUTPUT: mathObjType corresponding to maxSim(Context, mathTerms)
```

Figure 5.2: Algorithm for disambiguating simple mathematical expressions

In weighting the concepts from the context, the value of the function *cw* is computed according to the following two criteria:

1. In the case of mathematical terms contained in the current and the previous sentence corresponding to the target symbolic expression, we considered the distance to the expression with the weights decreasing according to the distance in words between the term and the target expression.

2. For the mathematical terms identified in the explicit introductions of mathematical symbols, the weights decreased starting from the first to the last occurence of the expression, reflecting the fact that in most cases symbols are declared with their first occurence (recall the two studies presented in the Sections 4.2.1.1 and 4.2.1.2 of the Chapter 4).

The total score corresponding to a mathematical object type ($Class$) is computed as follows:

$$Score(Class) = \alpha Sim(C_1, Class) + (1 - \alpha) Sim(C_2, Class) \tag{5.3}$$

where Sim was defined by Equation 5.1, and $C = C_1 \cup C_2$ represents the lexical context presented in Section 5.4.3. In order to disambiguate a symbolic expression, we compute the function $Score$ corresponding to all the mathematical object types contained in the lexical resource. The object type corresponding to the maximum computed value of function $Score$ will give the interpretation of the target expression.

5.5 Summary

In this chapter, we presented an architecture to support automated disambiguation of symbolic mathematical expressions. We described the process of constructing a hierarchical terminological resource consisting of classes of mathematical terms subordinate to high level mathematical object types. We presented a method of using the global discourse context in the process of disambiguating simple mathematical expressions, by mapping the terms contained in the expression's context to each object type from the lexical resource.

In the next chapter we will present an evaluation of the disambiguation approach and will discuss the results.

Chapter 6

Evaluation and Results

In this chapter we describe the evaluation of our approach to disambiguating symbolic expressions. We begin by presenting a walk-through example illustrating the steps we employed in assigning one (or more) object types to a given symbolic expression. We then describe how we selected our evaluation data set and how the evaluation gold standard was obtained. We continue by defining the performance measures we employed in evaluating the disambiguation approach. We finally present the results of the evaluation and draw the conclusions of the experiments.

6.1 Walk-through example

6.2 Gold standard

As any natural language processing application, our approach requires testing and evaluation. To establish the evaluation gold standard, we randomly selected a set of symbolic expressions which were classified by preselected judges into the corresponding mathematical object types. The gold standard is used for the automatic evaluation of our disambiguation approach. Below we present the data selection, describe the annotation procedure, and the way we adjudicated annotation conflicts.

Data set We conducted the evaluation of the disambiguation approach on a set of simple mathematical expressions which were selected as follows. We randomly selected a set of 140 mathematical documents from the corpus subset presented in Chapter 2. We then randomly picked one simple mathematical expression from each of the selected documents. We obtained a set of

140 occurences of different mathematical symbols which were prepared for the task of evaluating the disambiguation approach.

Procedure To establish the evaluation gold standard, we divided the data set into 5 disjunct annotation sets, each of them containing 28 symbolic expressions. We asked annotators to provide information on the type of mathematical object a symbolic expression denotes in the document.

The annotators were asked to distinguish between seven general classes of mathematical objects or concept types. The mathematical object types correspond to the ones from the lexical resource presented in Figure 5.4, i.e.:

1. *General algebraic objects*, such as "array", "element", "field", "intersection", "group", etc.
2. *Algebraic objects which denote correspondences*, i.e. mappings or functions, such as "correspondence", "function", "functor", "intersection", "metric", "morphism", etc.
3. *Number objects*, such as "dimension", "prime", "limit", "index", "exponent", "real", "error", etc.
4. *Objects which denote elements of notation or logical objects*, such as "formula", "notation", "symbol", "variable", "unknown", "form", "representation", etc.
5. *Objects which have a geometric interpretation*, such as "curve", "path", "trajectory", "diagram", "figure", "polygon", etc.
6. *Property or qualitative attribute denoting concepts*, such as "concentration", "position", "property", "symmetry", "singularity", "convexity", "complexity", "additivity", etc.
7. *Objects denoting methods or processes*, such as "algorithm", "inference", "calculation", "computation", "inverse", "method", etc.

Some mathematical objects could be classified as more than one of the above types, such is the case of *algebraic objects* that may also be classified as *geometric objects*. For instance, *"manifold"* can be viewed as a set on the one hand, i.e. a *general algebraic object*, or, in geometry, as a mathematical space with a dimension, a geometric property, i.e. a *geometric object*. This is the reason why the annotators were also asked to provide more options on the type corresponding to the sense in which the object is used in the given context.

Annotators The annotators were recruited on voluntary basis from colleagues with mathematical background. We contacted 12 annotators, out of whom 6 replied positively. We thus obtained 4 annotated sets, 2 of them being annotated by 2 annotators. We identified 7 disagreement cases which were adjudicated by the author. The remaining set was also annotated by the author.

The evaluation ground truth consisted of the resulting object types corresponding to each symbolic expression from our data set.

6.3 Performance measures

As evaluation metrics we use precision (*P*) and mean reciprocal rank (*MRR*). In classification, precision is the proportion of correctly labelled examples. *MRR* is one of the standard measures used in Information Retrieval for evaluating performance of systems which produce ranked lists of results, for example, ordered lists of documents retrieved in response to a query. It is the inverse of the rank of the expected (best) result item. More specifically,

$$P = \frac{tp}{tp + fp} \text{ and } MRR = \frac{1}{N} \sum_{i=1}^{N} \frac{1}{rank_i}$$

where *tp* are true positive classifications, *fp* are false positives, *fn* are false negatives, and *N* is the number of evaluated instances.

6.4 Results

Table 6.1 summarises the results of the evaluation we performed on the gold presented in Section 6.2. The columns of the Table 6.1 contain the scores computed by our disambiguation approach using the similarity measures presented in Section 5.4.2.1, i.e. *pointwise mutual information (PMI), mutual dependency (MD), Pearson's χ^2 test (PEA), and log likelihood (LL)*. For each of these 4 measures, we computed the precision and the mean reciprocal rank on the evaluation set.

Results are reported for different values of the parameter α which we used in weighting the individual disambiguation scores to obtain the overall disambiguation score which gives the type of mathematical object. For each similarity measure, we highlighted the line corresponding to the maximum precision score relative to all the values of the parameter α. The overall maximum precision and mean reciprocal rank scores were obtained for $\alpha \in \{0.35, 0.40\}$,

	PMI		MD		PEA		LL	
α	Prec	MRR	Prec	MRR	Prec	MRR	Prec	MRR
0.00	47.85%	66.71%	45.71%	65.21%	42.14%	58.94%	40.71%	60.11%
0.05	55.71%	72.0%	55.71%	71.9%	45.71%	61.78%	42.85%	61.45%
0.10	57.14%	72.81%	59.28%	74.07%	47.85%	63.34%	48.57%	65.2%
0.15	58.57%	74.04%	59.28%	73.97%	49.28%	65.0%	47.85%	65.07%
0.20	60.0%	74.74%	60.0%	74.31%	49.28%	65.57%	49.28%	66.34%
0.25	62.14%	75.93%	60.71%	74.78%	51.42%	67.16%	50.71%	67.52%
0.30	62.85%	76.11%	60.0%	74.31%	50.0%	66.85%	52.14%	68.39%
0.35	63.57%	76.47%	60.0%	74.19%	50.71%	67.72%	52.85%	68.79%
0.40	63.57%	76.47%	60.71%	74.54%	50.71%	68.24%	52.85%	69.67%
0.45	62.85%	76.11%	60.0%	74.07%	48.57%	67.31%	52.85%	69.95%
0.50	62.14%	75.75%	60.0%	74.03%	49.28%	67.13%	56.42%	72.39%
0.55	62.14%	75.75%	60.71%	74.51%	50.0%	67.35%	57.14%	72.59%
0.60	60.71%	75.04%	60.0%	74.15%	48.57%	66.39%	57.85%	72.62%
0.65	61.42%	75.39%	59.28%	73.79%	47.85%	66.36%	55.71%	71.61%
0.70	63.57%	76.35%	58.57%	73.44%	45.71%	65.23%	56.42%	71.66%
0.75	62.85%	75.99%	57.14%	72.56%	46.42%	64.89%	55.0%	70.63%
0.80	60.0%	74.5%	56.42%	72.2%	49.28%	65.78%	52.85%	69.12%
0.85	58.57%	73.64%	57.85%	72.92%	47.14%	63.98%	51.42%	67.89%
0.90	55.0%	71.0%	57.85%	72.8%	47.14%	62.97%	50.71%	67.45%
0.95	52.85%	70.01%	53.57%	70.51%	44.28%	61.54%	48.57%	65.94%
1.00	55.71%	70.73%	54.28%	69.78%	43.57%	60.45%	46.42%	64.31%

Table 6.1: Evaluation results

using *PMI* as similarity measure. We also observed that, for each of the values of the parameter α, both *PMI* and *MD* outperform *LEA* and *LL* on the considered data set. The extremal cases of values of parameter α represent the individual contributions of the two considered contexts. More precisely, $\alpha = 0.0$ indicates the values corresponding to the local lexical context, i.e. the terms contained in C_2, while $\alpha = 1.0$ indicates corresponds to the values obtained by taking into account the previous and the related symbol occurences, i.e. the terms contained in C_1.

6.5 Discussion

Considering lack of lexical knowledge for mathematical domain and the limited linguistic preprocessing we employ (the disambiguation method is based on lexical patterns and co-occurrence statistics with only stemming and stopword filtering) both the precision results and the mean reciprocal rank results on the evaluation set are encouraging.

Chapter 7

Conclusion and Further Work

Bibliography

[1] XHTML 1.0 The Extensible HyperText Markup Language (Second Edition). `http://www.w3.org/TR/xhtml1/`, Seen July 2009.

[2] XML Document Object Model Tutorial. `http://www.w3schools.com/dom/default.asp`, Seen July 2009.

[3] American Mathematical Society. 2010 Mathematics Subject Classification. `http://www.ams.org/mathscinet/msc/`, 2009.

[4] American Mathematical Society. Mathematical Reviews. `http://www.ams.org/mathscinet/index.html`, 2010.

[5] `arXiv.org e-Print archive`, 2010. `http://www.arxiv.org`, Seen January 2010.

[6] arXMLiv. Project home page at `http://arxmliv.kwarc.info/`, Seen July 2009.

[7] Rodrigo Bañuelos, Tadeusz Kulczycki, and Bartłomiej Siudeja. On the trace of symmetric stable processes on Lipschitz domains. *J. Funct. Anal.*, 257(10):3329–3352, 2009.

[8] Alberto Barrón-Cedeno, Gerardo Sierra, Patrick Drouin, and Sophia Ananiadou. An improved automatic term recognition method for spanish. In *CICLing '09: Proceedings of the 10th International Conference on Computational Linguistics and Intelligent Text Processing*, pages 125–136, Berlin, Heidelberg, 2009. Springer-Verlag.

[9] D. Blostein and A. Grbavec. Recognition of mathematical notation. *Handbook of Character Recognition and Document Image Analysis*, pages 557–582, 1997.

[10] D. Bollegala, Y. Matsuo, and M. Ishizuka. Measuring semantic similarity between words using web search engines. In *Proceedings of the 16th International Conference on World Wide Web*, pages 757–766, 2007.

[11] Nathanial P. Brown. Herrero's approximation problem for quasidiagonal operators. *Journal of Functional Analysis*, 186(2):360 – 365, 2001.

[12] R. Budiu, C. Royer, and P. Pirolli. Modeling information scent: a comparison of LSA, PMI-IR and GLSA similarity measures on common tests and corpora. In *Proceedings of the 8th RIAO Conference*, 2007.

[13] J. Bullinaria and J. Levy. Extracting semantic representations from word co-occurrence statistics: A computational study. *Behavior Research Methods*, 39(3):510–526, 2007.

[14] Ted Dunning. Accurate methods for the statistics of surprise and coincidence. *Computational Linguistics*, 19(1):61–74, 1993.

[15] Katerina Frantzi, Sophia Ananiadou, and Hideki Mima. Automatic recognition of multi-word terms: the C-value/NC-value method. *International Journal on Digital Libraries*, 3(2):115–130, 2000.

[16] Mihai Grigore, Magdalena Wolska, and Michael Kohlhase. Towards context-based disambiguation of mathematical expressions. In *The Joint Conference of ASCM 2009 and MACIS 2009: Asian Symposium on Computer Mathematics and Mathematical Aspects of Computer and Information Sciences*, volume 22 of *COE Lecture Notes*, pages 262–271, 2009.

[17] Donald E. Knuth. *Fundamental Algorithms*, volume 1 of *The Art of Computer Programming*, section 1.2, pages 10–119. Second edition, 1973.

[18] Zornitsa Kozareva, Ellen Riloff, and Eduard Hovy. Semantic class learning from the web with hyponym pattern linkage graphs. In *Proceedings of ACL-08: HLT*, pages 1048–1056, 2008.

[19] Thomas K. Landauer, Peter W. Foltz, and Darrell Laham. An Introduction to Latent Semantic Analysis. *Discourse Processes*, 25:259–284, 1998.

[20] D. W. Lozier. NIST Digital Library of Mathematical Function. *Annals of Mathematics and Artificial Intelligence — Special Issue on Mathematical Knowledge Management*, (38):105–119, 2003.

[21] C. D. Manning and H. Schütze. *Foundations of Statistical Natural Language Processing*. Massachusetts Institute of Technology, Revised version May 1999, 2003.

[22] R. Mihalcea, C. Corley, and C. Strapparava. Corpus-based and knowledge-based measures of text semantic similarity. In *Proceedings of the 21st National Conference on Artificial Intelligence*, pages 775–780, 2006.

[23] B. Miller. `LaTeXML`: A LaTeX to XML Converter. Web Manual at `http://dlmf.nist.gov/LaTeXML/`, September 2007.

[24] Bruce Miller. `LaTeXML`: A LaTeX to XML converter. Web Manual at `http://dlmf.nist.gov/LaTeXML/`, seen May 2010.

[25] Bruce Miller. `LaTeXML` architecture. `http://dlmf.nist.gov/LaTeXML/manual/architecture/latexmlarchitecture.xhtml`, seen June 2010.

[26] S. Mohammad and G. Hirst. Determining word sense dominance using a thesaurus. In *Proceedings of the 11st Conference of the European Chapter of the Association for Computational Linguistics*, pages 121–128, 2006.

[27] Aristomenis Thanopoulos Nikos, Nikos Fakotakis, and George Kokkinakis. Comparative evaluation of collocation extraction metrics. In *In Proceedings of the 3rd Language Resources Evaluation Conference*, pages 620–625, 2002.

[28] Kishore Papineni. Why inverse document frequency? In *NAACL '01: Second meeting of the North American Chapter of the Association for Computational Linguistics on Language technologies 2001*, pages 1–8, 2001.

[29] S. Patwardhan, S. Banerjee, and T. Pedersen. Using measures of semantic relatedness for word sense disambiguation. In *Proceedings of the 4th International Conference on Computational Linguistics and Intellignet Text Processing*, pages 241–257, 2003.

[30] T. Pedersen, S. Banerjee, and S. Patwardhan. Maximizing Semantic Relatedness to Perform Word Sense Disambiguation. Research Report UMSI 2005/25, University of Minnesota Supercomputing Institute, March 2005.

[31] Philip Resnik and David Yarowsky. A perspective on word sense disambiguation methods and their evaluation. In *Proceedings of the ACL SIGLEX Workshop on Tagging Text with Lexical Semantics: Why, What and How?*, pages 79–86, 1997.

[32] Stephen Robertson. Understanding inverse document frequency: On theoretical arguments for idf. *Journal of Documentation*, 60:2004, 2004.

[33] Heinrich Stamerjohanns, Michael Kohlhase, Deyan Ginev, Catalin David, and Bruce Miller. Transforming large collections of scientific publications to XML. *Mathematics in Computer Science*, 2010. in press.

[34] Jiri Stetina, Sadao Kurohashi, and Makoto Nagao. General word sense disambiguation method based on a full sentential context. In *Usage of WordNet in Natural Language Processing, Proceedings of COLING-ACL Workshop*, 1998.

[35] M. Strube and S. P. Ponzetto. WikiRelate! computing semantic relatedness using Wikipedia. In *Proceedings of the 21st National Conference on Artificial Intelligence*, pages 1419–1424, 2006.

[36] P. D. Turney. Mining the web for synonyms: PMI-IR versus LSA on TOEFL. *Lecture Notes in Computer Science*, pages 491–502, 2001.

[37] University of Cambridge. Millennium Mathematics Project. `http://mmp.maths.org/`, 1999.

[38] University of Cambridge. Mathematics Thesaurus. `http://thesaurus.maths.org`, 2004.

[39] W3C. Mathematical Markup Language (MathML) Version 3.0 (Third Edition).

[40] Charles Wells. A handbook of mathematical discourse, 2009.

[41] Sumio Yamada. On the Weil-Petersson Geometry of Teichmüller Spaces. *Math. Res. Let.* , pages 327–344, 2004.

[42] Zentralblatt MATH. `http://www.zentralblatt-math.org`, October 2009.

Chapter 8

Appendix

8.1 Lexical Resource

Mathematical Object Type	Corresponding Mathematical Concepts
`Algebraic object: General algebraic object`	array, element, field, intersection, group, module, matroid, matrix, ring, category, groupoid, set, domain, neighborhood, pair, range, region, semigroup, monoid, subset, chain, space, manifold, topology, vector, component, basis, kernel, ideal, subspace, minor, lattice, algebra, inverse, base, product, cycle, index, polynomial, coefficient, radical, value, integral, root, exponent, solution, result, invariant, algebroid, class, coalgebra, cobordism, equivalence, family, exponential, form, hierarchy, integrator, map, phase, quadrature, semilattice, subcategory, submanifold, system, variety, ufd, domain, retract, sequence, series, eigenvector

`Algebraic object: Mapping or function`	code, correspondence, function, functor, intersection, metric, morphism, order, transformation, bundle, functional, mapping, norm, operator, kernel, homomorphism, harmonic, change, derivative, integral, measure, operation, transform, product, wave, exponential, estimator, multiplication, ordering, quadrature, regulator, restriction, retraction, transition, shift, relation, distribution, eigenfunction, distribution
`Number object`	number, quaternion, harmonic, dimension, prime, limit, index, exponent, real, error, rational, fraction, integer, divisor, factor, quotient, residue, constant, difference, estimate, product, solution, result, norm, cardinal, mean, ordinal, radius, value, rank, spread, curvature, fraction, density, energy, gravity, force, tension, weight, mass, unit, volume, pint, complex, part, measure, ratio, proportion, percentage, statistic, correlation, deviation, mean
`Notational or logical object`	equation, formula, notation, symbol, variable, unknown, index, form, representation, scheme, condition, conjecture, constraint, convention, criterion, hypothesis, lemma, model, problem, rule, law, theorem, structure, proof, argument, inference, formalism, induction, semantics, type, logic, inequality
`Geometric object`	curve, path, trajectory, diagram, figure, polygon, square, graph, network, lattice, tessellation, tiling, polyhedron, torus, space, surface, polytope, rotundum, sphere, simplex, point, bifurcation, integral, cycle, ecliptic, orbit, coordinate, covering, hypersurface, quadrangle, quiver, radius, submanifold, supergeometry, torsion, area, volume, plane

`Method or Process`	algorithm, inference, calculation, computation, inverse, method, transformation, dilation, reduction, glide, differentiation, integration, measurement, operation, simulation, test, transform, aggregation, approach, coding, contraction, decomposition, descent, expansion, fitting, formation, generalization, generation, identification, induction, inclusion, labelling, learning, mechanization, minimization, numbering, optimization, ordering, percolation, procedure, refinement, synthesis, technique, treatment, extension, control, chaos, bifurcation, convection, process, turbulence, change, interaction, effect, control, reasoning, proving, modeling, rounding, estimation, analysis, interpolation, approximation, simplification, smoothing, splitting, stabilization, transfer, parametrization, progression, decomposition, elimination, fibration, fitting, inversion, action
`Qualitative attribute`	concentration, position, property, invariant, symmetry, singularity, convexity, complexity, additivity, adjunction, coherence, compactness, computability, connectedness, consistency, definability, dependence, determinacy, differentiability, distributivity, elasticity, holomorphy, independence, modularity, optimality, periodicity, randomness, semicontinuity, similarity, stability, uniformity

Table 8.1: Lexical resource for mathematical expression disambiguation