Magdalena Wolska; Mihai Grigore; Michael Kohlhase

Using Discourse Context to Interpret Object-Denoting Mathematical Expressions

In: Petr Sojka and Thierry Bouche (eds.): Towards a Digital Mathematics Library. Bertinoro, Italy, July 20-21st, 2011. Masaryk University Press, Brno, Czech Republic, 2011. pp. 85--101.

Persistent URL: http://dml.cz/dmlcz/702605

Using Discourse Context to Interpret Object-Denoting Mathematical Expressions

Magdalena Wolska[1], Mihai Grigore[2⋆], and Michael Kohlhase[3]

[1] Computational Linguistics and Phonetics, Saarland University
D-660 41 Saarbrücken, Germany
magda@coli.uni-saarland.de
[2] Information Systems Engineering, Goethe University
D-603 23 Frankfurt am Main, Germany
grigore@wiwi.uni-frankfurt.de
[3] Computer Science, Jacobs University Bremen
D-287 59 Bremen, Germany
m.kohlhase@jacobs-university.de

Abstract. We present a method for determining the context-dependent denotation of simple object-denoting mathematical expressions in mathematical documents. Our approach relies on estimating the similarity between the linguistic context within which the given expression occurs and a set of terms from a flat domain taxonomy of mathematical concepts; one of 7 head concepts dominating a set of terms with highest similarity score to the symbol's context is assigned as the symbol's interpretation. The taxonomy we used was constructed semi-automatically by combining structural and lexical information from the Cambridge Mathematics Thesaurus and the Mathematics Subject Classification. The context information taken into account in the statistical similarity calculation includes lexical features of the discourse immediately adjacent to the given expression as well as global discourse. In particular, as part of the latter we include the lexical context of structurally similar expressions throughout the document and that of the symbol's declaration statement if one can be found in the document. Our approach has been evaluated on a gold standard manually annotated by experts, achieving 66% precision.

1 Introduction

Consider the following discourse fragment from [14]:

> ... Let $f : X \to B$ be a surjective morphism and let $\omega_{X/B}$ denote the relatively canonical sheaf of differentials. Let us assume that the generic fibre is smooth of genus g and let us denote by δ the number of singular points in the fibres. We write Λ_n for the determinant of $f_*\omega^n_{X/B}$ and λ_n for the degree of $\boxed{\Lambda_n}$. ...

⋆ This work was performed while the second author was visiting the Computational Linguistics and Phonetics Department of Saarland University as a student of Jacobs University Bremen.

Petr Sojka, Thierry Bouche (editors): DML 2011, Towards a Digital Mathematics Library, pp. 85–101.
 ISBN 978-80-210-5542-1

Even a layperson, without any knowledge whatsoever on the subject matter of the discourse from which the above fragment has been extracted, is capable of inferring the name of the object which the boxed expression, Λ_n, denotes: In the same sentence in which the expression in question occurs she finds a statement "We write Λ_n for the determinant of $f_*\omega^n_{X/B}$" from which she can infer that Λ_n must denote an object called "determinant". She may not know what a determinant is specifically and how to compute it, but she can at least identify the domain term that names the object for which the symbol stands in order to find its meaning, for instance, in a textbook.[4]

The above-quoted fragment exemplifies a fairly typical way in which mathematical discourse is written. While mathematical documents abound in symbols, a large proportion of the symbols used are explicitly introduced in the discourse or stated to denote specific objects. A corpus study on symbol declarations in mathematical writing revealed that around 70% of object-denoting symbolic expressions randomly selected from mathematical scientific papers were explicitly stated to denote objects of specific types [15].

In computational linguistics, the problem of identifying which sense of a polysemous word is meant in a given sentence is known as *word sense disambiguation* (WSD) and has been one of the active research areas since the beginning of interest in word sense disambiguation in the forties.[5] Clearly, an automated text understander, if it is to make inferences about a discourse, must be in a position to discriminate between the meanings of words and correctly recognize the meaning in context. With the increasing interest in automated processing of technical and scientific documents, in particular, with the view to building interactive digital libraries of scientific writing[6] the same holds of automated processing of scientific prose. In particular, in case of exact sciences which make use of symbolic notation, identifying the meaning of not only the linguistic expressions, but also formal expressions is an obvious task and challenge. Interpretation of symbolic mathematical expressions can be a useful source of information in a number of sub-tasks in a mathematical document processing pipeline for digitizing mathematics. For instance, in the task of parsing mathematical notation, i.e. identifying the structure and (compositional) semantics of symbolic expressions, the information about the expressions' interpretation can guide the selection (or weighing) of likely parse candidates. This could be useful in processing LaTeX documents as well as in mathematical OCR, in particular, in handwriting recognition; for instance, in examples such as

[4] Note that for the purposes of the knowledge-poor methods discussed in this paper, it is sufficient to determine that "determinant of $f_*\omega^n_{X/B}$" denotes an object via a domain term ("determinant"). In fact, as a reviewer pointed out, in this particular example, this determinant is a sheaf over a smooth fibration, not a number computed from a matrix, as a non-expert might suspect. This shows that for more knowledge-rich methods, a tight collaboration between authors and linguists is of essence.

[5] For a recent comprehensive overview of the state of the art, see [8].

[6] (See, for instance, EuDML (`http://www.eudml.eu/`) or WDML (`http://www.mathunion.org/WDML/`) for recent efforts in this direction.)

above, in deciding between horizontal adjacency and super-/subscript relation when the super-/subscript is written partly across the centre horizontal line of the expression.

Our previous study on disambiguating symbolic expressions has shown that the local linguistic context, within which mathematical expressions are embedded, provides a good source of information for recognizing a class of objects to which a mathematical expression belongs [5]. However, the approach addressed only those mathematical expressions which are syntactically part of a nominal group and, in particular, are in an apposition relation with an immediately preceding noun phrase; i.e. the expressions addressed came from a linguistic pattern: "... noun_phrase symbolic_math_expression ...", as in the example: "... the inverse function ω_1 ...". Only the immediate left linguistic context of a symbolic was used in the disambiguation process, despite the fact that mathematical texts are known to introduce notations and concepts as they go along.

In this paper we propose a new approach to interpreting mathematical expressions. Our interpretation strategy is inspired by recent computational WSD approaches which use statistical co-occurrence measures to estimate semantic relatedness between lexical contexts. In our case, co-occurrence statistics are computed using on the one hand, both the local discourse within which the expression under analysis is embedded as well as the relevant segments of the entire document (global discourse) and, on the other hand, sets of terms from a lexical resource we built. As in [5], we use a lexical resource of mathematical terms which corresponds to a flat taxonomy of mathematical objects and which provides an association between sets of domain terms which name mathematical objects and names of broader semantic classes of mathematical objects. The taxonomy has been constructed semi-automatically by combining structural and lexical information from the Mathematics Subject Classification and the Cambridge Mathematics Thesaurus. The class names themselves serve as symbolic interpretations of mathematical expressions under analysis.

The class of expressions addressed In this work we attempt to interpret only *simple object-denoting mathematical expressions*. "Simple" refers to the expressions' high-level structure: The terms may be atomic identifiers and super-/sub-scripted atomic identifiers; the expression(s) in the super-/subscripts can be of arbitrary complexity. For instance, Λ_n and $\omega_{X/B}$ are simple expressions, while $f : X \to B$ is not. Throughout this paper, the term "simple mathematical expression(s)" refers to this class of symbols.

Problem statement We formulate the interpretation problem as follows: Given a mathematical document containing a target mathematical expression of the type described above, a simple object-denoting term, can we indicate one (or more) concepts from a predefined set of concepts which corresponds to a coarse-grained denotation of the given expression?

Outline The paper is organized as follows: In Section 2 we introduce the corpus from which the documents we analyze stem and outline the preprocessing steps. In Section 3 we introduce the taxonomy of mathematical objects constructed for this study. In Section 4 we describe the approach to interpreting simple object-denoting mathematical terms: We introduce the similarity measures, the two types of context based on which similarity is computed, and the algorithm itself. In Section 5 we summarize the creation of a gold standard for evaluation, the evaluation measures we used, and the results themselves.

2 The Data and Preprocessing

For the purposes of this work we used 10,000 mathematical documents from the arXMLiv collection, processed by the `LaTeXML` system [10,12]. arXMLiv is subset of arXiv, an archive of electronic preprints of scientific papers in the fields of, among others, mathematics, statistics, physics, and quantitative biology[7]. The documents we used stemmed from the mathematics subset of arXiv.

`LaTeXML` uses three formats for representing mathematical expressions of which two are relevant for this study: In the XMath format mathematical expressions are encoded as 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). The presentation format, MathML, is a widely used W3C standard for rendering mathematical content on the Web [1].[8] Figure 1 shows the XMath and MathML representations of the expression $\mathcal{D}/\mathcal{D}_0$. The two formats are used to retrieve simple mathematical terms as defined in the Introduction.

2.1 Tokenization and identification of target expressions

Each of the 10,000 documents in the corpus was word- and sentence-tokenized,[9] and the words were stemmed.[10] Then mathematical expressions were normalized by replacing them with unique identifiers and the mappings between the identifiers and the two relevant representations were stored for each mathematical expression. Simple mathematical expressions were identified

[7] `http://www.arxiv.org`

[8] `http://www.w3.org/Math/`

[9] A sentence is understood, in a standard sense, as a grammatical unit consisting of one or more clauses. Sentence-tokenization was performed using a rule-based tokenizer based on a standard set of end-of-sentence punctuation marks and a number domain-specific rules for sentences ending with mathematical expressions which may not end with end-of-sentence punctuation.

[10] We use stemming as a knowledge-poor substitute for lemmatization. This solution has obvious drawbacks, however, context-sensitive lemmatization is out of scope at the time this work is conducted because we do not have access to a large-scale dictionary for mathematical discourse, nor to any standard language processing tools for this domain.

```
<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>
```

```
<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>
```

Fig. 1. XMath (top) and MathML (bottom) representations of the expression $\mathcal{D}/\mathcal{D}_0$

by analyzing the MathML and XMath representations[11] and the results were manually verified for the expressions used for the gold standard.

2.2 Domain term identification

The purpose of identifying mathematical domain terms was two-fold: First, we identify domain terms in the Mathematics Subject Classification while building the lexical resource for interpretation and, second, we use domain terms in the course of identifying symbol declaration statements which are used in the interpretation process.

To identify domain terms, we implemented a modified version of the algorithm presented in [4]. In our implementation, only n-gram counts are used and no linguistic information; in particular, we do not have part of speech (POS) tag information which the authors use to identify noun phrases.[12] We

[11] We omit the algorithm here.

[12] Again, due to the notorious lack of linguistic processing tools for mathematical discourse we opt for a knowledge-poor approach here. We are presently working on building up an annotated corpus of mathematical discourse in order to train a

Table 1. An excerpt from MSC 2010

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

therefore employed a tailored stop-word list including items which are not closed-class words and not part of classical stop-word lists, but which are also not likely to be part of names of mathematical domain objects. These included, for the most part, common verbs.[13]

The threshold for discarding n-gram candidates was set at five or less occurrences in the corpus (low-frequency n-grams). As in the original algorithm, the remaining n-grams were scored by taking into account their length, frequency, and the number of their nested occurrences in longer n-grams. The score threshold for discarding candidate domain terms was set at 10.[14]

3 A Taxonomy of Mathematical Objects

3.1 The resources

Mathematics Subject Classification The Mathematics Subject Classification[15] (MSC) is a hierarchically organized classification of mathematical domains encompassing over 5,000 sub-areas of mathematics and has been developed with the view to helping retrieval of documents from the AMS Mathematical Reviews Database (MathSciNet)[16] and the Zentralblatt MATH (ZMATH)[17]. Table 1 shows an excerpt from the MSC 2010 representing the first level of the "SEQUENCES, SERIES, SUMMABILITY" class. Each MSC subject class consists of a class code and a high-level class name, and includes a list of mathematical sub-areas subsumed under the given class. The sub-ares, in turn, may also

POS tagger for the domain. Small-scale experiments with tagging using off-the-shelf POS-tagging models yielded, unsurprisingly, highly sub-standard results, therefore we aim at training a dedicated tagger for mathematical discourse.

[13] The extended stop-word list did not, however, include prefixes which do occur in mathematical object names, such as "semi-", "quasi-", "sub-", "pseudo-", etc., be it hyphenated or not.

[14] For the details of the algorithm, please refer to the cited article.

[15] `http://www.ams.org/mathscinet/msc/`

[16] `http://www.ams.org/mathscinet/`

[17] `http://www.zentralblatt-math.org`

```
function ExtractMinimalLengthPaths(MSC, CMT)

MathTerms := findMultiwordTerms(MSC)
removeModifiers(mathTerms)
TopNode := CMT node with no node along "broader"-relation
SetOfPaths := ∅
foreach Term in Mathterms
   if Term occurs in CMT
      MinLengthPath=Dijkstra(TopNode, Term)
      add MinLengthPath to SetOfPaths
return SetOfPaths
```

Fig. 2. Pseudo-code of the minimal length path extraction algorithm

include sub-classes which denote more fine-grained topical distinctions within the given sub-domain. Using the domain term identification algorithm outlined above, we automatically extract mathematical domain terms contained in the names of the MSC classes.

Cambridge Mathematics Thesaurus The University of Cambridge Mathematics Thesaurus[18] (CMT) is part of the Millennium Mathematics Project.[19] The CMT contains 4,583 concepts together with short explanations and thesaurus relations such as "broader/narrower" and "references/referenced by". We exploit the thesaurus' hierarchy by following the "broader/narrower" relations in order to find hypernyms of mathematical terms.

3.2 Building the taxonomy

Automated processing First, in order to obtain a set of concepts, multi-word mathematical terms were extracted from the MSC using a variant of the domain term identification algorithm from [4] (function `findMultiwordTerms` in the pseudo-code in Figure 2). The extracted multi-word terms were simplified to single-word terms by removing their adjectival or noun modifiers using lexical rules (`removeModifiers`). The obtained set consisted of 341 unique mathematical concept names. 170 of these were also found in the CMT and were used in further automated processing.

Next, we used the "broader/narrower" relations from the CMT to traverse the CMT graph in order to retrieve the hypernyms of the extracted MSC terms. For each of the 170 terms, the algorithm first finds the root of the CMT graph (a node without any parent nodes along the "broader" relation) and then looks for the shortest path down to the given term, i.e. we find the minimal sub-graph induced by the set of common higher mathematical concepts. The algorithm is summarized in Figure 2.

[18] `http://thesaurus.maths.org`
[19] `http://mmp.maths.org/`

```
Algebraic object : Set : Semigroup : Monoid
Attribute : Quality : Property : Physical property : Position
Number : Real : Rational : Integer : Divisor
```

Fig. 3. Examples of minimal length paths extracted from the CMT

The obtained minimal length paths serve as a starting point to clustering mathematical concepts under higher-level concepts. Consider, for instance, the extracted minimal length paths corresponding to the concepts **Monoid**, **Position**, and **Divisor** shown in Figure 3. These paths allowed us to further manually classify the concepts as more general object types, e.g. **Monoid** as an **Algebraic object**, **Position** as a **Qualitative attribute**, and **Divisor** as a **Number object**. The manual classification process is summarized below.

Manual processing We manually transformed the obtained minimal length paths into paths of length at most two (i.e. each term/concept has at most one intermediate hypernym/super-concepts) obtaining the following top-level classes of mathematical objects:

1. Algebraic object : General algebraic object,
2. Algebraic object : Mapping or function,
3. Number object,
4. Notational and logical object,
5. Geometric object,
6. Qualitative attribute,
7. Method or Process.

The top-level classes were selected in such way that they are the least ambiguous in terms of classifying a mathematical concept into one of them. We therefore merged two closely related classes: **Algebraic object : Number object** and **Quantitative attribute** because the distinction between them was too fine-grained, obtaining a common class for number concepts, **Number object**. The class **Algebraic object : Mapping or function** is the result of merging **Algebraic object : Mapping** and **Algebraic object : Function**; In the CMT **Function** is subsumed both under **Algebraic object** directly and under **Map** which is also subsumed under **Algebraic object**, resulting in a cycle. In order to avoid ambiguity in interpretation, these two classes were merged.

170 MSC concepts were already subsumed under the above-mentioned classes. We then manually classified the remaining 171 MSC concepts which were not found in the CMT, obtaining a flat taxonomy of mathematical objects. An excerpt of the taxonomy is shown in Table 2.

While the flat structure captures the complexity of the relations between mathematical object types only at a coarse-grained level, this is sufficient for our current purposes for two reasons: First, given the knowledge-poor approach we pursue, we aim at a high-level classification at present, and second, the

Table 2. An excerpt from the taxonomy/the lexical resource

Mathematical object class	Set of subsumed 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, ...
Qualitative attribute	concentration, position, property, invariant, symmetry, singularity, convexity, complexity, additivity, adjunction, coherence, compactness, computability, connectedness, ...
Method or process	algorithm, inference, calculation, computation, inverse, method, transformation, dilation, reduction, glide, differentiation, integration, measurement, operation, ...

purpose of the taxonomy is to serve as a *lexical resource* with all the 341 MSC terms subsumed under one of the above-mentioned names of mathematical object classes which correspond to high-level common denotations of the sets of terms.[20] Section 4.2 shows how the sets of terms are matched with linguistic contexts in which a symbolic mathematical expression, whose interpretation is to be disambiguated, occurs. The present approach to disambiguation is at its core similar to the one introduced previously in [5], however, the new lexical resource for interpretation is superior by comparison with the one we used earlier in several respects: First, there is a clear relation between the top-level classes and the subsumed concepts: in all the cases the relation is of *is-a* type. Secondly, as mentioned above, the sets of terms themselves are coherent: they cluster terms which are hyponyms of a more general term which all of the member terms denote (at a coarse level of detail). Finally, the lexical resource comprises a smaller number of classes which should remove some spurious ambiguity in selecting a type as an interpretation of a mathematical expression.

[20] Note that EngMath [6], an existing ontology of mathematics, cannot be directly used for this purpose; EngMath is a formal ontology developed with the goal of serving as a machine-readable formal specification. Also the scope of EngMath is focused on mathematics in the engineering domain; it encompasses the following concepts: scalar, vector, and tensor quantities, physical dimensions, units of measure, functions of quantities, and dimensionless quantities (ibid.)

4 Interpreting Simple Object-Denoting Expressions

The process of interpreting mathematical expressions consists of three stages: First, the documents are preprocessed and mathematical expressions which are targets for interpretation, i.e. simple mathematical expressions, are identified (see Section 2). Then for each target mathematical expression, we calculate the similarity between the linguistic context in which it occurs and each set of mathematical terms in the lexical resource. The final interpretation of target expression is assigned using a scoring function.

4.1 Word-to-word similarity

In the disambiguation process we use similarity measures in order to decide which sets of terms from the lexical resource is closest to the lexical context of a target expression. The similarity between two words is calculated as follows:

$$\mathtt{sim}(w_1, w_2) = \begin{cases} \mathtt{Dice}(w_1, w_2) & \text{when } \mathtt{Dice}(w_1, w_2) > \lambda \\ \textit{Co-occurrence-based measure} & \text{otherwise} \end{cases} \quad (1)$$

where $\mathtt{Dice}(w_1, w_2)$ is the Dice's character-based word-to-word similarity:[21]

$$\mathtt{Dice}(w_1, w_2) = \frac{2 * n_{\text{common\_bigrams}}}{n_{\text{bigrams\_w}_1} + n_{\text{bigrams\_w}_2}} \quad (2)$$

and *Co-occurrence-based measure* is one of the following measures of lexical co-occurrence: Pointwise Mutual Information (PMI), Mutual dependency (MD), Pearson's χ^2, and Log-likelihood ratio (LL). All of these are standard corpus-based lexical association measures and have been previously successfully used in various computational linguistics tasks to estimate the relative probability with which words occur in proximity [13,3,2]. Based on experimentation we used $\lambda = 0.7$ as the threshold for using string-based similarity.

4.2 The interpretation algorithm

Before presenting the core interpretation algorithm we make precise what constitutes the components of the linguistic context of a target expression which we take into account in the course of interpretation.

The context of a mathematical expression For each target mathematical expression which we attempt to interpret, we take into account the global and local lexical context $C_C = C_L \cup C_G$ consisting of two sets of domain terms:[22]

C_L is the set of domain terms which occur in the *local context* of a mathematical expression, more specifically, within a window of textual content

[21] Dice accounts for different inflectional variants of words in the lexical resource and in the linguistic context of a mathematical expression.

[22] *Lexical* mathematical domain terms are meant here.

preceding and following the given mathematical expression, i.e. within the immediately preceding and following linguistic context,[23]

C_G is the set of domain terms which occur in the global context of the entire document. More specifically, we consider terms which occur in the *declaration statements* of the given target expression or of other expressions which are structurally similar to the target expression, according to the notion of structural similarity defined in [15].[24]

Each extracted mathematical term from C_L and C_G contributes to the final similarity score proportionally to its importance in the disambiguation process.

Disambiguation To infer the meaning a mathematical expression we use an approach inspired by methods of word sense disambiguation from computational linguistics which use inventories of word senses and measures of semantic similarity to map a word in context to its possible sense(s) from an inventory; see, for instance, [11,9].

Our approach to interpreting mathematical expressions uses the mathematical object classes shown in Table 2 on the left as the the inventory of possible "senses" of symbolic mathematical expressions. In order to identify the class which corresponds to the given use of a target mathematical expression, we map the mathematical terms (w) from the expression's context, C_C (defined above), to the mathematical terms ($\mathtt{term}$) subsumed under each class of mathematical objects from the taxonomy ($\mathtt{Class}$). To accomplish this, we adapt the approach to estimating the semantic similarity of two text segments T_1 and T_2 proposed in [9]. As estimates of semantic similarity between sets of words, we use the measures presented in Section 4.1. The Context-to-Class similarity is calculated using the following scoring function:

$$Sim(C, Class) = \sum_{w \in C} maxSim(w, Class) \times cw(w), \text{ where} \tag{3}$$

$$maxSim(w, Class) = \max_{\mathtt{term} \in Class} [sim(w, term)] \tag{4}$$

[23] In the current implementation we used the window of ± 2 sentences with respect to the sentence within which the target mathematical expression occurs. Paragraph and section boundaries were not considered at this time.

[24] Identification of declaration statements was performed automatically by applying a set of regular expressions to preprocessed documents in which domain terms have been identified (see Section 2). The set was bootstrapped from a small set of seed patterns using the simple variant of the *anchored patterns* approach proposed in [7]. Using the final set of bootstrapped patterns, the algorithm achieved retrieval precision of 89% and recall of 77% on a test set. We do not include the details of the approach here. For a general description of the method, see [7].

Following [15] we consider two simple expressions to be structurally similar if they share the same top-level node in the expression tree and their expression trees have the same structure modulo the structure of the super-/subscript terms. For instance, ω_i and ω_{n-1} are structurally similar according to these criteria. By contrast, P_c^2 and A_n^k are not similar because they differ in the top-node identifier.

```
function findCandidateInterpretations(targetME)

C_G := ∅, C_L := ∅
foreach occurrence of targetME
if occurrence is explicitly declared
   add definiendum to C_G
foreach ME structurally similar to targetME
   foreach occurrence of ME
      if occurrence is explicitly declared
         add definiendum to C_G
select ±2-sentence context window W of targetME
foreach word w in W
   add w to C_L

foreach C in {C_L, C_G}
   foreach Class in Taxonomy
      compute Sim(C, Class)
      update maxSim(C, Class)
   return Class corresponding to maxSim(C, Class)
```

Fig. 4. Pseudo-code of the interpretation algorithm; targetME is a simple mathematical expressions as defined in the Introduction

$Sim(\mathrm{C}, Class)$ is computed for each class of concepts from the taxonomy and represents the similarity score between the context of the given mathematical expression and the domain terms which name concepts from the given class, C is $\mathrm{C_L}$ or $\mathrm{C_G}$ (see above), $sim(w, term)$ is the word-to-word similarity defined by Equation 1, and $\mathrm{cw}(w)$ is a weight (see below). The pseudo-code of the interpretation algorithm is shown in Figure 4.

The weight $\mathrm{cw}(w)$ is computed according to the following criteria:

- For term in $\mathrm{C_L}$ (local context; here: window of ±2-sentences) we consider the distance to the target mathematical expression with the weights decreasing with the distance in words between the term and the target expression,
- For terms in $\mathrm{C_G}$ (global context; declarations) the weights decrease from the first to the last occurrence of the expression in the document. This reflects the fact that in most cases symbols are declared with their first occurrence [15].

The final score for a *Class* as an interpretation of the given mathematical expression is computed as a combination of the local and global context scores:

$$Score(\mathrm{C_C}, Class) = \alpha Sim(\mathrm{C_G}, Class) + (1 - \alpha) Sim(\mathrm{C_L}, Class) \tag{5}$$

where $Sim(C, Class)$ was defined by Equation 3. computed for all classes from the taxonomy. The class with the maximum score is assigned as the interpretation of the given mathematical expression.[25]

5 Evaluation

In order to evaluate the interpretation procedure we created a *gold standard* set of mathematical expressions with interpretations provided by experts. The interpretation algorithm was run on the gold standard and two evaluation measures were computed for different values of the α parameter.

5.1 The gold standard

The evaluation set Mathematical expressions for the gold standard set were selected as follows: A set of 200 mathematical documents was randomly selected from the preprocessed corpus described in Section 2. Then one random simple mathematical expression was picked from each of the selected documents yielding a set of 200 occurrences of different mathematical expressions. The selected mathematical expressions were annotated by experts as described below.

Procedure The data for the gold standard was randomly split into 7 disjoint annotation sets each of which contained from 28 to 30 mathematical expressions. The annotation was performed using a web-interface we created. Mathematical expressions were presented to the annotators together with the entire document. The annotators were asked to assign a type to a symbolic expression highlighted in the document. An excerpt from the annotation instructions is shown in Figure 5. The 7 object types listed in the instructions directly corresponded to the classes from the taxonomy we used as a lexical resource.

Annotators The annotators were recruited on voluntary basis from colleagues with strong mathematical background. We contacted 18 candidate annotators, out of whom 7 responded: five were computer scientists (three post-graduates and two with doctorates), and two were working mathematicians with doctorates in mathematics. Two sets were moreover annotated by the second author of the paper. Four sets were annotated by 2 annotators in order to verify agreement. 7 identified disagreement cases were adjudicated by the authors of the paper.

[25] Note that the lexical similarity-based approach as such is language-independent; it is likely, though, that for a heavily inflected language a different threshold for word-to-word similarity would have to be used. However, because the lexical resource we use is English and because the rules for identifying declaration statements are language-specific, the evaluation is currently limited to English discourse.

Your task is to annotate symbolic mathematical expressions in mathematical documents. For each indicated expression we ask you to provide the information on the type of object the expression denotes in the given context.

For this task we distinguish seven general classes of mathematical objects or concept types [...]. These are:

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.
...
7. Objects denoting methods or processes, such as "algorithm", "inference", "calculation", "computation", "inverse", "method", etc.

Many mathematical objects could be classified as more than one of the above types. For instance, many algebraic objects could be also classified as geometric objects. A manifold is such an example: it 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. In cases of such ambiguities, please annotate the type corresponding to the sense in which the object is used in the given context.

Fig. 5. Excerpt from the annotation instructions

5.2 Evaluation measures

We use precision (P) and mean reciprocal rank (MRR) as evaluation measures. In classification, precision is the proportion of correctly labeled examples out of all labeled 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. More specifically,

$$P = \frac{tp}{tp + fp} \times 100 \quad \text{and} \quad MRR = \frac{1}{\mathrm{N}} \sum_{\mathrm{i}=1}^{\mathrm{N}} \frac{1}{rank_{\mathrm{i}}}$$

where tp is the number of true positive classifications, fp is the number of false positives, N is the number of evaluated instances, and $rank$ is the position of the correct classification in the list of results.

5.3 Results

Figure 6 shows the results plots for both evaluation measures at different values of α (the parameter which weighs the contribution of local vs. global context similarity scores to the overall score). The lines correspond to the different word-to-word similarity measures: Pointwise Mutual Information, Mutual Dependency, χ^2, and Log-Likelihood (see Section 4.1). Solid lines denote precision, dashed lines mean reciprocal rank.

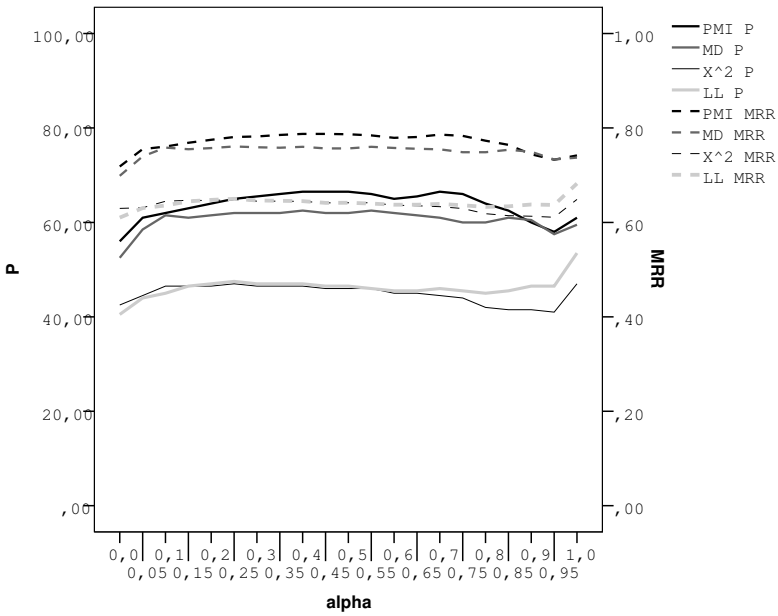

Fig. 6. Precision and mean reciprocal rank plots.

In general PMI and MD yield better results than χ^2 and LL on our data set. The maximum precision and mean reciprocal rank scores are obtained for $\alpha = \{0.40, 0.45, 0.50, 0.70\}$ using PMI as the similarity measure ($P = 66.50\%$, $MRR = 0.79$). The general pattern for PMI and MD appears the same: the combination of local and global context (α mid-range) gives better results than local or global context alone ($\alpha = 0$ and $\alpha = 1$, respectively). While yielding somewhat lower results, the MD plot line appears flatter than the PMI plot on both measures, that is, MD is more stable across the different values of α than PMI. Both χ^2 and LL perform best when relying on the global context alone, that is, when the interpretation is based solely on the explicit symbol declaration.

6 Conclusion

We presented a knowledge-poor method of finding a denotation of simple object-denoting symbolic expressions in mathematical discourse. We have shown that the lexical information from the linguistic context immediately surrounding the expression under analysis as well as the lexical information from the larger document context both contribute to achieving the best interpretation results.

Considering that the presented method relies on only limited linguistic knowledge (co-occurrence statistics over documents preprocessed using stemming and stop-word filtering), the precision results we have obtained encourage further exploration of the approach, in particular, extending it

with more linguistically-informed analysis. We are presently annotating a subset of the corpus used in the experiments described here with parts of speech tags in order to train domain-specific POS-tagging models. We expect several improvements due to POS-tagging, among others, better domain term identification and, consequently, better identification of declaration statements, as well as access to shallow syntactic analysis of the immediate context of mathematical expressions.

We have also shown a method of constructing a flat taxonomy of mathematical objects which can serve as a lexical resource for corpus similarity-based approaches. Multi-annotator tagging of a subset of a gold standard by two annotators, using the classes from the taxonomy as annotation labels, resulted in only 7 disagreements on 112 instances. In spite of the low disagreement count, there are at least two obvious problems with the evaluation presented here: First, admittedly, the annotation with the taxonomy classes and the evaluation was conducted on a small-scale. We are planning further annotation experiments in order to further validate the suitability of the taxonomy for the mathematical expression interpretation task. Second, from a mathematical perspective, the taxonomy we constructed is disappointingly high-level. However, this is about all we can hope for with *knowledge-poor* methods. As we already remarked in footnote 4, *knowledge-rich* methods will need a tight collaboration between experts and linguists. The former need to supply machine-understandable mathematical domain ontologies (classifications of mathematical objects and relations between them) while the latter need to adapt parsing and semantic analysis algorithms to take advantage of these and also to accomodate the fact that these ontologies are dynamic, i.e., change over the course of a document (or document collection). We conjecture that ontologies needed for document processing tasks are best created by semantically annotating (and thus partially formalizing) the mathematical documents that introduce them — a process that will have to involve linguistic analysis to scale. The knowledge-poor methods presented in this paper can be viewed as a small step in this direction.

Acknowledgments. We are indebted to Deyan Ginev of Jacobs University Bremen for compiling and preparing the corpus used in this work and for the many preprocessing scripts without which it would not have been possible to conduct this study at ease. We would like to thank the annotators who were so kind as to dedicate their time and knowledge to constructing the gold standard we used in the evaluation. We would also like to thank the three anonymous reviewers for their insightful comments and suggestions.

References

1. Ausbrooks, R., Carlisle, S.B.D., Chavchanidze, G., Dalmas, S., Devitt, S., Diaz, A., Dooley, S., Hunter, R., Ion, P., Kohlhase, M., Lazrek, A., Libbrecht, P., Miller, B., Miner, R., Sargent, M., Smith, B., Soiffer, N., Sutor, R., Watt, S.: Mathematical Markup Language (MathML) version 3.0. W3C Working Draft of 24. September 2009, World Wide Web Consortium (2009), `http://www.w3.org/TR/MathML3`.

2. Budiu, R., Royer, C., Pirolli, P.: Modeling information scent: a comparison of LSA, PMI-IR and GLSA similarity measures on common tests and corpora. In: Proceedings of the 8th Conference on Large Scale Semantic Access to Content (RIAO-07). pp. 314–332 (2007).
3. Bullinaria, J., Levy, J.: Extracting semantic representations from word co-occurrence statistics: A computational study. Behavior Research Methods 39(3), 510–526 (2007).
4. Frantzi, K., Ananiadou, S., Mima, H.: Automatic recognition of multi-word terms: the C-value/NC-value method. International Journal on Digital Libraries 3(2), 115–130 (2000).
5. Grigore, M., Wolska, M., Kohlhase, M.: Towards context-based disambiguation of mathematical expressions. In: Selected Papers from the joint conference of ASCM 2009 and MACIS 2009: the 9th Asian Symposium on Computer Mathematics and the 3rd International Conference on Mathematical Aspects of Computer and Information Sciences. pp. 262–271 (2009).
6. Gruber, T., Olsen, G.: An ontology for engineering mathematics. In: Proceedings 4th International Conference on Principles of Knowledge Representation and Reasoning. pp. 258–269 (1994).
7. Kozareva, Z., Riloff, E., Hovy, E.: Semantic class learning from the Web with Hyponym Pattern Linkage Graphs. In: Proceedings of the ACL/HLT-08 Conference. pp. 1048–1056 (2008).
8. McCarthy, D.: Word sense disambiguation: An overview. Language and Linguistics Compass 3(2), 537–558 (2009).
9. Mihalcea, R., Corley, C., Strapparava, C.: Corpus-based and knowledge-based measures of text semantic similarity. In: Proceedings of the 21st National Conference on Artificial Intelligence. pp. 775–780 (2006).
10. Miller, B.: `LaTeXML`: A LaTeX to XML Converter. Web Manual at `http://dlmf.nist.gov/LaTeXML/` (September 2007).
11. Pedersen, T., Banerjee, S., Patwardhan, S.: Maximizing semantic relatedness to perform word sense disambiguation. Research Report 25, University of Minnesota Supercomputing Institute (2005).
12. Stamerjohanns, H., Kohlhase, M., Ginev, D., David, C., Miller, B.: Transforming Large Collections of Scientific Publications to XML. Mathematics in Computer Science 3, 299–307 (2010).
13. Turney, P.D.: Mining the Web for Synonyms: PMI-IR versus LSA on TOEFL. In: Proceedings of the 12th European Conference on Machine Learning. pp. 491–502 (2001), `http://cogprints.org/1796/`.
14. Wessler, M.: An algebraic proof of Iitaka's conjecture. Archiv der Mathematik 79, 268–273 (2002), `http://dx.doi.org/10.1007/s00013-002-8313-2`.
15. Wolska, M., Grigore, M.: Symbol declarations in mathematical writing. In: Sojka, P. (ed.) Proceedings of the 3rd Workshop on Digital Mathematics Libraries. pp. 119–127 (2010).