Sense disambiguation:
from natural language words to mathematical terms

Minh-Quoc Nghiem, [1] **Giovanni Yoko Kristianto,** [2] **Goran Topić,** [3] **Akiko Aizawa** [2,3]
[1] The Graduate University for Advanced Studies, Tokyo, Japan
[2] The University of Tokyo, Tokyo, Japan
[3] National Institute of Informatics, Tokyo, Japan
{nqminh, giovanni, goran_topic, aizawa}@nii.ac.jp

Abstract

This paper addresses the open problem of mathematical term sense disambiguation. We introduce a method that uses a MathML parallel markup corpus to generate relevant training and testing datasets. Based on the dataset generated, we use Support Vector Machine classifier to disambiguate the sense of mathematical terms. Experimental results indicate we can generate such data automatically and with reasonable accuracy.

1 Introduction

Word-sense disambiguation (WSD) refers to the process of identifying the correct sense or meaning of a word in a sentence when the word has multiple meanings. WSD remains a difficult open problem in natural language processing. Current WSD systems are based on supervised, unsupervised, and knowledge-based approaches (Navigli, 2009). This paper focuses on the problem of disambiguating the sense of mathematical terms occurring within normal text, an aspect little discussed to date.

The problem of achieving automated understanding of mathematical expressions can be illustrated quite clearly. For instance, depending on context, the mathematical term δ can be interpreted to refer to `Kronecker Delta`, `Dirac Delta`, `Discrete Delta`, or simply to a variable δ. Another example is `i`, which can be interpreted to mean `the imaginary constant`, `the index variable`, or `the bound variable` of an operation. Other examples include α, β, σ, ϕ, ω, Φ, B, H, x, y, sim. In many such cases, disambiguation can play a crucial role in the automated understanding, translation, and calculation of mathematical expressions.

One major issue in early research on machine understanding of mathematical terms found in text was the lack of evaluation datasets. A previous study (Wolska et al., 2011) was based on a small evaluation set of 200 mathematical expressions annotated by experts. Clearly, large samples of sense-tagged data would require significant human annotation and labor. Fortunately, then, Ide et al. (2002) showed that sense distinctions derived from cross-lingual information are at least as reliable as those made by human annotators. The novel research described in our paper presents a fully automated method for generating large samples of mathematical terms with sense-tagged data.

As part of the effort described here to address mathematical term sense disambiguation (MTSD), we first propose a method that uses a MathML parallel markup corpus to generate training and testing datasets. Second, we propose heuristics that improve alignment results for the parallel markup corpus. Third, we present a classification-based approach to the MTSD problem. To the best of our knowledge, this study is the first to make use of parallel corpora to address MTSD.

The rest of this paper is organized as follows: Sections 2 and 3 provide a brief overview of the background and related work; Section 4 presents our methods; Section 5 describes the experimental setup and results; Section 6 concludes the paper and points to directions for future research.

2 Background

Web pages and documents represent mathematical expressions in many formats: images, TeX, MathML (Ausbrooks et al., 2010), OpenMath (Buswell et al., 2004), OMDoc (Kohlhase, 2006), or the ISO/IEC standard Office Open XML (Miller et al., 2009). This paper uses MathML markup, a format recommended by the W3C Math Working Group, as a standard for rep-

International Joint Conference on Natural Language Processing, pages 809–814,
Nagoya, Japan, 14-18 October 2013.

resenting mathematical formulas. MathML uses presentation markup to capture notational structures and content markup to capture mathematical structures and mathematical meaning. MathML parallel markup provides both forms of markup for the same mathematical expression. Figure 1 shows the MathML presentation and content markup for the expression `arctan(0)=0` [1].

Presentation MathML

```
<mrow>
   <mrow>
      <msup>
         <mi>tan</mi>
         <mrow>
            <mo>-</mo>
            <mn>1</mn>
         </mrow>
      </msup>
      <mo>(</mo>
      <mn>0</mn>
      <mo>)</mo>
   </mrow>
   <mo>=</mo>
   <mn>0</mn>
</mrow>
```

ContentMathML

```
<apply>
   <eq/>
   <apply>
      <arctan/>
      <cn>0</cn>
   </apply>
   <cn>0</cn>
</apply>
```

Figure 1: MathML presentation and content markup for the expression `arctan(0)=0`

Natural language sentences and presentation mathematical expressions have several key similarities and differences. A token element in a mathematical expression can be regarded as a word in a sentence. In presentation markup, token elements are divided into four main types: identifiers (`<mi>x</mi>`), operators (`<mo>+</mo>`), numbers (`mn>2</mn>`), and text (`<mtext>non zero</mtext>`). A sentence may contain certain layout elements, such as subscripts or superscripts, while a mathematical expression may contain numerous layout elements, such as `<mrow>`, `<msup>`, `<munderover>`, and `<mfrac>`. As noted by Ausbrooks et al. (2010), mathematical notation, while more rigorous than natural language, is ambiguous and context-dependent.

3 Related Work

Several studies have shown encouraging results for WSD based on parallel corpora (Diab and Resnik, 2002; Tufiş et al., 2004; Chan and Ng, 2005; Carpuat and Wu, 2007; Padó and Lapata, 2009; Lefever and Hoste, 2010; Lefever et al., 2011). Ide et al. (2002) used translation equivalents derived from parallel aligned corpora to determine sense distinctions applicable to automatic sense-tagging. They evaluated their work using a subset of 33 nouns covering a range of occurrence frequencies and degrees of ambiguity (Ide et al., 2001), with results indicating no significant difference in agreement rates for the algorithm and for human annotators. The main limitation of this study is its dependence on aligned corpora, which are not easily obtainable.

Wolska et al. (Wolska and Grigore, 2010; Wolska et al., 2011) presented a knowledge-poor method for identifying the denotation of simple symbolic expressions in mathematical discourse. Based on statistical co-occurrence measures, the system sorted a simple symbolic expression under one of seven predefined concepts. Here, the authors found that lexical information from the linguistic context immediately surrounding the expression improved results. This approach achieves 66% agreement with the gold standard of manual annotation by experts. From our perspective, the predefined concepts are closely related to syntactic function, not the semantics of the terms.

4 Our Approach

4.1 Generating the Datasets

We compiled our MTSD data using parallel MathML markup expressions gathered from the Web. First, using a set of heuristic rules, we preprocessed the parallel MathML markup expressions. We then used the GIZA++ toolkit to obtain node-to-node aligned data. Based on the node-to-node aligned data, we created subtree-to-subtree aligned data. Finally, we extracted ambiguous terms from the subtree-to-subtree aligned data to obtain data for MTSD. Figure 2 gives the steps taken to generate the data.

A crucial step in generating MTSD data is achieving alignment between the Presentation side and the Content side of the expressions. Given a set of several MathML parallel markup expressions, we used the automated word alignment

[1] http://functions.wolfram.com/01.14.03.0001.01

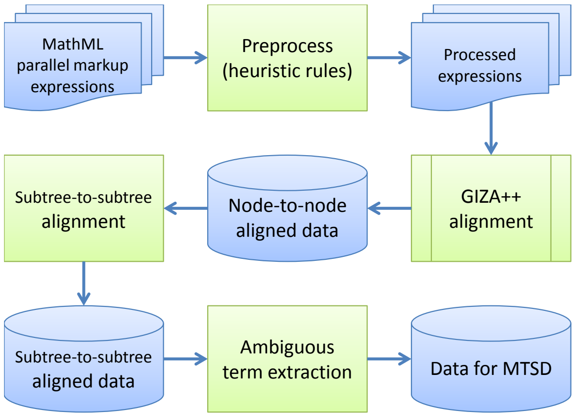
MathML parallel markup expressions
Preprocess (heuristic rules)
Processed expressions
GIZA++ alignment
Node-to-node aligned data
Subtree-to-subtree alignment
Subtree-to-subtree aligned data
Ambiguous term extraction
Data for MTSD

Figure 2: Steps for generating the data for MTSD.

GIZA++ (Och and Ney, 2003) to obtain alignment between the Presentation terms and Content terms. Developed to train word-based translation models, the GIZA++ toolkit is not directly applicable to a tree-based corpus. One common solution is to convert the tree into a sentence by extracting the leaf nodes of the tree and to form a sequence (Sun et al., 2010). While this approach works well for natural language text, it is less effective with mathematical expressions, since the intermediate nodes of these expressions contain layout information.

Before using GIZA++, to enhance alignment precision, we apply two heuristic rules to the presentation tree based on information on its structure. The first heuristic rule converts the intermediate layout nodes (except `mrow`) to leaves on the tree by moving them to the position of their first child. When moving an intermediate layout node, we create a temporary ('temp') node to replace the moved node and to keep the other child nodes intact. Unnecessary parentheses, which indicate that the expressions in the parentheses belong together, are also removed. Figure 3 illustrates an example of this heuristic. In this example, we moved the `msup` node to a leaf of the tree and removed a pair of parentheses, `<mo>(</mo>` and `<mo>)</mo>`, near `<mn>0</mn>` node.

The second heuristic rule moves operator (`mo`) nodes to the beginning of the subtree if that subtree contains operator nodes. This rule reduces cross alignments, since most notations in content MathML are prefix notations and placed in leaf nodes. In Figure 3, the `<mo>=</mo>` node is moved to the first position of the tree. The `<mo>-</mo>` node is not moved because it is already the first child of its parent node. This figure also shows alignment results for GIZA++ before and after applying heuristic rules for the expression `arctan(0)=0`.

To extract more complex mathematical terms, we expand the node-to-node alignments to subtree-to-subtree alignments. In this study, we expanded the subtree alignment only to the parent of the `mi` nodes. The criteria used here to achieve subtree aligned pair are similar to that used by Tinsley et al. (2007). First, a node can be linked only once. Second, descendants of a presentation node can link only to descendants of its content counterpart. Third, ancestors of a presentation node can link only to ancestors of its content counterpart (a node counts as its own ancestor).

If one presentation node links to more than one content node, we keep only the link with the highest alignment score, as given by Equation 1. The number of alignments between the presentation tree $tree_P$ and the content tree $tree_C$ is the sum of (1) the number of alignments from the leaf children of $tree_P$ to the leaf children of $tree_C$ and (2) the number of alignments from the leaf children of tree to the leaf children of $tree_P$. For more accurate results, we removed node-to-node alignments if alignment probabilities fell below a certain threshold (0.2). In Equation 1, P_{child} and C_{child}, respectively, refer to the child nodes of $tree_P$ and $tree_C$. The blue lines in Figure 3 represent the expanded alignments between subtrees.

$$score(\mathrm{tree_P}, \mathrm{tree_C}) = \frac{\#\ \mathrm{alignments}}{\#\ \mathrm{P_{child}} + \#\ \mathrm{C_{child}}} \quad (1)$$

Based on the alignment results, we extracted pairs of presentation mathematical terms and their associated content terms. A mutually aligned presentation subtree and content subtree form a pair. This paper will consider only mathematical terms containing `mi` (e.g. tan^{-1}, `Ai`, `Ai(0)`, Γ, $\Gamma(\frac{2}{3})$). Only terms associated with ambiguous mapping are retained to generate training and testing data.

4.2 Disambiguating Mathematical Terms

We created a labeled training set, then used Support Vector Machines (SVM) to learn a classifier from this labeled data. Assume that a presentation term e has n ways of translating to content MathML term. Then, for each mathematical expression, we create one positive instance by combining e and its correct translation. We also create $n1$ negative instances by combining e and its incorrect translations. We will assign each instance

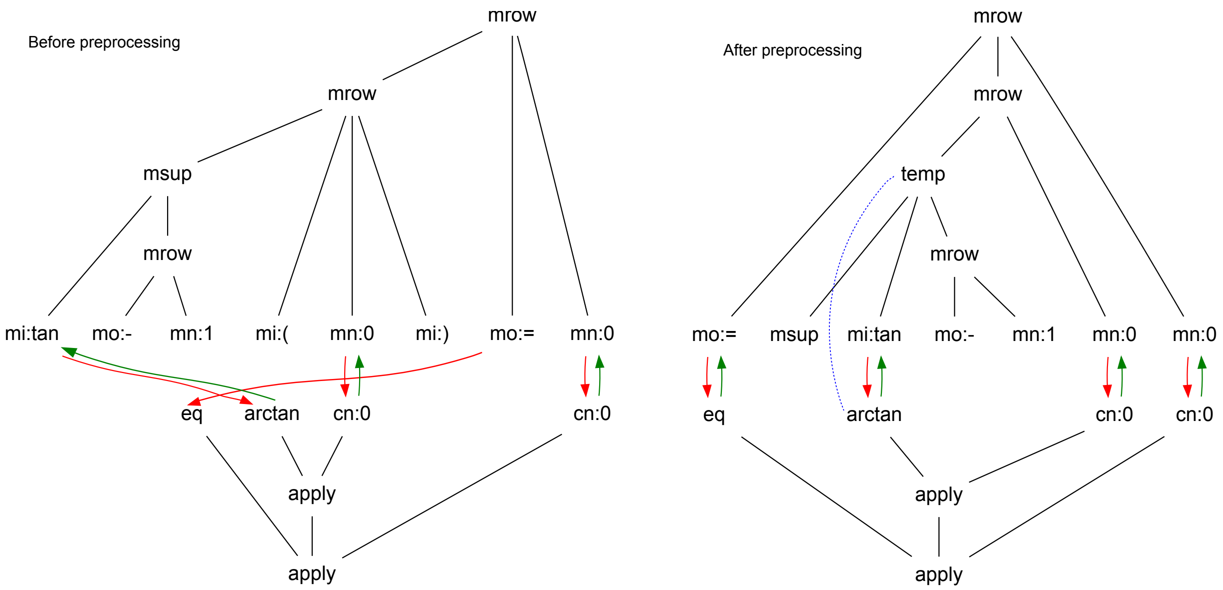

Figure 3: Example of alignment results for GIZA++ before and after applying the heuristic rules for the expression `arctan(0)=0`. Red lines represent alignments from presentation nodes to content nodes; green lines represent alignments from content nodes to presentation nodes; blue lines represent expanded alignments between subtrees.

to one of two classes, depending on the candidate translation: The class is ‘true’ if the content term is the correct translation of the presentation term; otherwise, the class is ‘false.’

We can divide the features used in SVM disambiguation into two main groups: presentation MathML and text features. Presentation MathML features are extracted from the presentation MathML markup of the mathematical expressions. Mathematical compendium websites often group expressions into several categories. The only text feature we use here is the name of the category to which a mathematical expression belongs. Table 1 shows the features we used for classification.

Table 1: Features used for classification

Feature	Description
Only child	Is it the only child of its parent
Preceded by mo	Is it preceded by an `mo` node
Followed by mo	Is it followed by an `mo` node
mo’s name	The name of the followed `mo`
Parent’s name	The name of its parent node
Node name	The name of the node
Identifier’s name	The name of the first `mi` child
Category	Relation between category name & candidate translation

Our experiment involved seven presentation MathML features. The first determines whether the term is the only child of its parent. The next three features encode the relationship between the term and the surrounding *mo* elements. The last three features represent the parent’s name, the term’s own name, and the first `mi` child’s name. Since mathematical terms differ from natural language words, the features differ as well.

5 Evaluation

5.1 Evaluation Setup

For these experiments, we collected parallel MathML markup expressions from the Wolfram Functions Site[2] (WFS), the world’s largest collection of formulas and graphics related to mathematical functions. All mathematical expressions on WFS are available in MathML parallel markup. For simplicity, we excluded long expressions containing more than 30 leaf nodes. We collected a total of 20,314 mathematical expressions.

5.2 Evaluation Results

We began by investigating the quality of the generated MTSD data. Using WFS data, we generated 2,925 different mathematical terms. There are 390 distinct ambiguous terms and 2,535 distinct

[2] http://functions.wolfram.com/

unambiguous terms. Of the ambiguous terms, 90 distinct terms are single mi elements. There are 67,987 instances contain all the ambiguous terms in our data. Table 2 shows the generated data.

Table 2: Generated data

Type	Distinct term
Ambiguous mi terms	90
Other ambiguous terms	300
Unambiguous terms	2,535

The table shows that only 14% of the extracted mathematical terms are ambiguous. One possible explanation: In WFS data, people tend to use one meaning for a fixed notation. Another: The system depends on the quality of the alignment output. The aligner may ignore an alignment if the probability of the alignment is low. This also causes errors in sense extraction if a sub-tree is aligned with a single term but the links are not fully connected: for example, tan^{-1} (Presentation) and $arctan$ (Content).

Within the scope of this paper, we focused on the single mi element terms. (The same method can be expanded to encompass additional ambiguous terms.) We manually verified these single mi element terms to assess the quality of the generated MTSD data. Of 247 extracted senses, 197 were correct, an accuracy rate of 79.76% for the generated data. Each mi element term has an average of 2.74 senses. The term with the most senses was <mi>C</mi>, which had six senses: Catalan, CatalanNumber, C, GegenbauerC, Cyclotomic, and FresnelC.

Next, we set up an experiment using libSVM[3] in the Weka toolkit (Hall et al., 2009) to examine sense disambiguation results for each presentation MathML term. The data we used contained the 90 distinct ambiguous mi terms. In this evaluation, we compared the results for systems using different training data: automatically extracted data and manually verified data. We also compared the results of our approach to the 'most frequent' method, which chooses the interpretation of highest probability. Since in the real world not every mathematical expression is associated with its category name, we also set up another experiment to assess the performance of our approach with and without the 'category' feature.

We built two models using nine-tenths of the automatically extracted data and nine-tenths of the manually verified data. Both systems set aside one-tenth of the verified data for testing. Classification accuracies were computed over the set of binary decisions. We used the default libSVM parameters. Table 3 gives the disambiguation accuracy for ambiguous mi terms.

Table 3: Sense disambiguation accuracy for ambiguous mi terms

Method	Extracted data	Verified data
All feature	**91.40**	**93.94**
Without 'category' feature	91.22	92.41
Most frequent	85.01	89.76

The results in Table 3 indicate reasonable results for the automatically extracted data. We gained improvements ranging from 1.2 to 2.5 percent by building a model using manually verified data. The classifier with 'category' feature slightly outperformed the classifier without the 'category' feature. Overall, the results here were approximately 4 to 7 percent more accurate than for the 'most frequent' method. The explanation for the relatively high scores for the 'most frequent' method is that mathematical elements often have a preferred meaning.

The results suggest we can make direct use of automatically generated data when working on the MTSD problem. For mathematical expressions in MathML parallel markup, the generated data is good enough without manual checking. The results also show that the text feature-i.e., the category of the mathematical term-contributes to system performance. While this improvement is modest, it suggests that features aside from the mathematical term itself can be helpful. However, the system works well even without this feature.

6 Conclusion

This paper presents an approach to creating training data for the mathematical term sense disambiguation problem. Combining word-to-word alignment models and heuristic alignments, this approach shows that we can generate reasonably accurate MTSD data using parallel corpora. The data generated can then be used to train a classifier that allows automatic sense-tagging of mathematical expressions.

[3] http://www.csie.ntu.edu.tw/∼cjlin/libsvm/

In contrast to natural language text, mathematical expressions require specific processing methods. More work needs to be done to establish the features best-suited to mathematical terms in a larger dataset. An extension of the model with more text and context features, in addition to the category feature, should prove interesting. Since the alignments between presentation and the content tree affect the generated data, improving alignment accuracy may boost system performance.

Acknowledgments

This research was supported in part by the Japan Society for the Promotion of Science (JSPS) via a Grant-in-Aid for Scientific Research (Grant 24300062).

References

Ron Ausbrooks, Stephen Buswell, David Carlisle, Giorgi Chavchanidze, Stéphane Dalmas, Stan Devitt, Angel Diaz, Sam Dooley, Roger Hunter, Patrick Ion, et al. 2010. Mathematical markup language (MathML) version 3.0. W3C recommendation. *World Wide Web Consortium.*

Stephen Buswell, Olga Caprotti, David P Carlisle, Michael C Dewar, Marc Gaetano, and Michael Kohlhase. 2004. The open math standard version 2.0. Technical report.

Marine Carpuat and Dekai Wu. 2007. Improving statistical machine translation using word sense disambiguation. In *In The 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL 2007*, pages 61–72.

Yee Seng Chan and Hwee Tou Ng. 2005. Scaling up word sense disambiguation via parallel texts. In *Proceedings of the 20th national conference on Artificial intelligence - Volume 3*, pages 1037–1042.

Mona Diab and Philip Resnik. 2002. An unsupervised method for word sense tagging using parallel corpora. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, pages 255–262.

Mark Hall, Eibe Frank, Geoffrey Holmes, Bernhard Pfahringer, Peter Reutemann, and Ian H. Witten. 2009. The weka data mining software: an update. *SIGKDD Explor. Newsl.*, 11(1):10–18.

Nancy Ide, Toma Erjavec, and Dan Tufis. 2001. Automatic sense tagging using parallel corpora. In *In Proceedings of the 6 th Natural Language Processing Pacific Rim Symposium*, pages 212–219.

Nancy Ide, Tomaz Erjavec, and Dan Tufis. 2002. Sense discrimination with parallel corpora. In *Proceedings of the ACL-02 workshop on Word sense disambiguation: recent successes and future directions-Volume 8*, pages 61–66.

Michael Kohlhase. 2006. *An Open Markup Format for Mathematical Documents (Version 1.2)*. Lecture Notes in Artificial Intelligence, no. 4180. Springer Verlag, Heidelberg.

Els Lefever and Véronique Hoste. 2010. Semeval-2010 task 3: Cross-lingual word sense disambiguation. In *Proceedings of the 5th International Workshop on Semantic Evaluation*, SemEval '10, pages 15–20.

Els Lefever, Véronique Hoste, and Martine De Cock. 2011. Parasense or how to use parallel corpora for word sense disambiguation. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 317–322.

Frederic P. Miller, Agnes F. Vandome, and John McBrewster. 2009. *Office Open XML*. Alpha Press.

Roberto Navigli. 2009. Word sense disambiguation: A survey. *ACM Computing Surveys*, 41:1–69.

Franz Josef Och and Hermann Ney. 2003. A systematic comparison of various statistical alignment models. *Computational linguistics*, 29(1):19–51.

Sebastian Padó and Mirella Lapata. 2009. Cross-lingual annotation projection for semantic roles. *Journal of Artificial Intelligence Research*, 36:307–340.

Jun Sun, Min Zhang, and Chew Lim Tan. 2010. Exploring syntactic structural features for sub-tree alignment using bilingual tree kernels. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 306–315.

John Tinsley, Ventsislav Zhechev, Mary Hearne, and Andy Way. 2007. Robust language pair-independent sub-tree alignment. In *In Proceedings of MT Summit XI -07.*

Dan Tufiş, Radu Ion, and Nancy Ide. 2004. Fine-grained word sense disambiguation based on parallel corpora, word alignment, word clustering and aligned wordnets. In *Proceedings of the 20th international conference on Computational Linguistics*, COLING '04.

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

Magdalena Wolska, Mihai Grigore, and Michael Kohlhase. 2011. Using discourse context to interpret object-denoting mathematical expressions. In *Towards a Digital Mathematics Library. Bertinoro, Italy, July 20-21st, 2011*, pages 85–101.