Mining coreference relations between formulas and text using Wikipedia

Minh Nghiem Quoc [1], **Keisuke Yokoi** [2], **Yuichiroh Matsubayashi** [3] **Akiko Aizawa** [1 2 3]
[1] Department of Informatics, The Graduate University for Advanced Studies
[2] Department of Computer Science, University of Tokyo
[3] National Institute of Informatics
{nqminh, kei-yoko, y-matsu, aizawa}@nii.ac.jp

Abstract

In this paper, we address the problem of discovering coreference relations between formulas and the surrounding text. The task is different from traditional coreference resolution because of the unique structure of the formulas. In this paper, we present an approach, which we call *'CDF (Concept Description Formula)'*, for mining coreference relations between formulas and the concepts that refer to them. Using Wikipedia articles as a target corpus, our approach is based on surface level text matching between formulas and text, as well as patterns that represent relationships between them. The results showed the potential of our approach for formulas and text coreference mining.

1 Introduction

1.1 Motivation

Mathematical content is a valuable information source for many users: teachers, students, researchers need access to mathematical resources for teaching, studying, or obtaining updated information for research and development. Although more and more mathematical content is becoming available on the Web nowadays, conventional search engines do not provide direct search of mathematical formulas. As such, retrieving mathematical content remains an open issue.

Some recent studies proposed mathematical retrieval systems that were based on structural similarity of equations (Adeel and Khiyal, 2008; Yokoi and Aizawa, 2009; Nghiem et al., 2009). However, in these studies, the semantics of the equations is still not taken into account. As mathematical equations follow highly abstract and also rewritable representations, structural similarity alone is insufficient as a metric for semantic similarity.

Based on this observation, the primary goal of this paper is to establish a method for extracting implicit connections between mathematical formulas and their names together with the descriptions written in natural language text. This enables keywords to be associated with the formulas and makes mathematical search more powerful. For example, it is easier for people searching and retrieving mathematical concepts if they know the name of the equation *"$a^2 + b^2 = c^2$"* is the *"Pythagorean Theorem"*. It could also make mathematics more understandable and usable for users.

While many studies have presented coreference relations among texts (Ponzetto and Poesio, 2009), no work has ever considered the coreference relations between formulas and texts. In this paper, we use Wikipedia articles as a target corpus. We chose Wikipedia for these reasons: (1) Wikipedia uses a subset of TEX markup for mathematical formulas. That way, we can analyze the content of these formulas using TEX expressions rather than analyzing the images. (2) Wikipedia provides a wealth of knowledge and the content of Wikipedia is much cleaner than typical Web pages, as explained in Giles (2005).

Proceedings of the Second Workshop on NLP Challenges in the Information Explosion Era (NLPIX 2010), pages 69–74,
Beijing, August 2010

1.2 Related Work

Ponzetto and Poesio (2006) attempted to include semantic information extracted from WordNet and Wikipedia into their coreference resolution model. Shnarch et al. (2009) presented the extraction of a large-scale rule base from Wikipedia designed to cover a wide scope of the lexical reference relations. Their rule base has comparable performance with WordNet while providing largely complementary information. Yan et al. (2009) proposed an unsupervised relation extraction method for discovering and enhancing relations associated with a specified concept in Wikipedia. Their work combined deep linguistic patterns extracted from Wikipedia with surface patterns obtained from the Web to generate various relations. The results of these studies showed that Wikipedia is a knowledge-rich and promising resource for extracting relations between representative terms in text. However, these techniques are not directly applicable to the coreference resolution between formulas and texts as we mention in the next section.

1.3 Challenges

There are two key challenges in solving the coreference relations between formulas and texts using Wikipedia articles.

- First, formulas have unique structures such as prior operators and nested functions. In addition, features such as gender, plural, part of speech, and proper name, are unavailable with formulas for coreference resolution. Therefore, we cannot apply standard natural language processing methods to formulas.

- Second, no labeled data are available for the coreference relations between formulas and texts. This means we cannot apply commonly used machine learning-based techniques without expensive human annotations.

1.4 Our Approach and Key Contributions

In this paper, we present an approach, which we call *CDF (Concept Description Formula)*, for mining coreference relations between mathematical Formulas and Concepts using Wikipedia articles. In order to address the previously mentioned challenges, the proposed *CDF* approach is featured as follows:

- First, we consider not only the concept-formula pairs but extend the relation with descriptions of the concept. Note that a "concept" in our study corresponds to a "name" or a "title" of a formula, which is usually quite short. By additionally considering words extracted from the descriptions, we have a better chance of detecting keywords, such as mathematical symbols, and function or variable names, used in the equations.

- Second, we apply an unsupervised framework in our approach. Initially, we extract highly confident coreference pairs using surface level text matching. Next, we collect promising syntactic patterns from the descriptions and then use the patterns to extract coreference pairs. The process enables us to deal with cases where there exist no common words between the concepts and the formulas.

The remainder of this paper is organized as follows: In section 2, we present our method. We then describe the experiments and results in section 3. Section 4 concludes the paper and gives avenues for future work.

2 Method

2.1 Overview of the Method

In this section, we first explain the terms used in our approach. We then provide a framework of our method and the functions of the main modules.

Given a set of Wikipedia articles as input, our system outputs a list of formulas along with their names and descriptions. Herein

- **Concept**: A concept C is a phrase that represents a name of a mathematical formula. In Wikipedia, we extract candidate concepts as noun phrases (NPs) that are either the titles of

Wikipedia articles, section headings, or written in bold or italic. Additional NPs that contain at least one content word are also considered.

- **Description**: A description D is a phrase that describes the concept. In Wikipedia, descriptions often follow a concept after the verb "*be*".

- **Formula**: A formula F is a mathematical formula. In Wikipedia extracted XML files, formulas occur between the $< math >$ and $< /math >$ tags. They are encoded in TEX format.

- **Candidate**: A candidate is a triple of concept, description and formula. Our system will judge if the candidate is qualified, which means the concept is related to the formula.

Figure 1 shows a section of a Wikipedia article and the concepts, descriptions and formulas in this section. Table 1 shows the extracted candidates. Details of how to extract the concepts, descriptions and formulas and how to form candidates are described in the next sections.

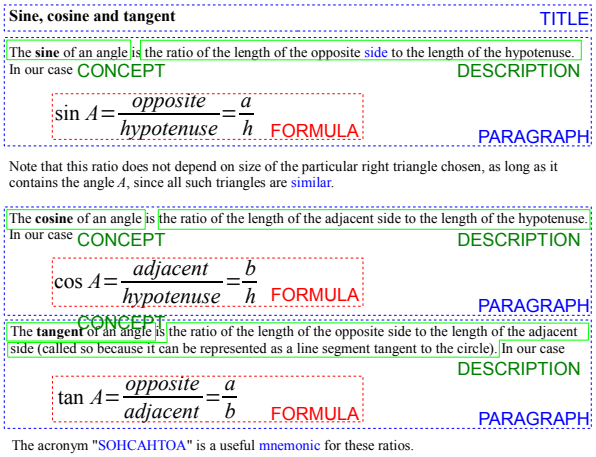

Figure 1: Examples of extracted paragraphs

The framework of the system is shown in Figure 2. The system has four main modules.

- **Text Preprocessor**: processes Wikipedia articles to extract *CDF (Concept Description Formula)* candidates.

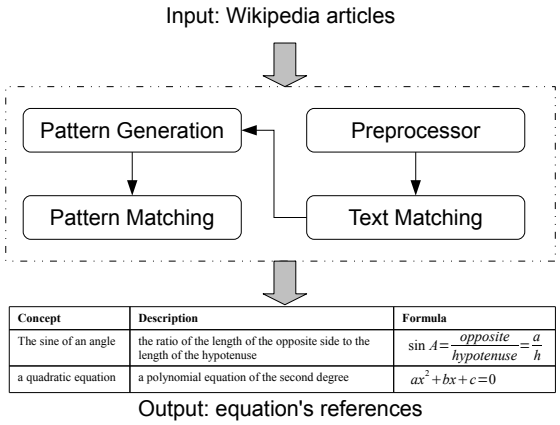

Figure 2: Framework of the proposed approach

- **Text Matching**: extracts reliable and qualified candidates using surface level text matching.

- **Pattern Generation**: generates patterns from qualified candidates.

- **Pattern Matching**: extends the candidate list using the generated patterns.

2.2 Text Preprocessor

This module preprocesses the text of the Wikipedia article to extract *CDF* candidates. Based on the assumption that concepts, their descriptions and formulas are in the same paragraph, we split the text into paragraphs and select paragraphs that contain at least one formula.

On these selected paragraphs, we run Sentence Boundary Detector, Tokenizer and Parser from OpenNLP tools. [1] Based on the parse trees, we extract the noun phrases (NPs) and identify NPs representing concepts or descriptions using the definitions in Section 2.1.

Following the general idea in Shnarch et al. (2009), we use the *"Be-Comp"* rule to identify the description of a concept in the definition sentence. In a sentence, we extract nominal complements of the verb 'to be', assign the NP that occurs after the verb 'to be' as the description of the NP that occurs before the verb. Note that some concepts have descriptions while others do not.

[1] http://opennlp.sourceforge.net/

Table 1: Examples of candidates

Concept	Description	Formula
the sine of an angle	the ratio of the length of the opposite side to the length of the hypotenuse	$\sin A = \frac{\text{opposite}}{\text{hypotenuse}} = \frac{\text{a}}{\text{h}}$
the cosine of an angle	the ratio of the length of the adjacent side to the length of the hypotenuse	$\cos A = \frac{\text{adjacent}}{\text{hypotenuse}} = \frac{b}{h}$
a quadratic equation	a polynomial equation of the second degree	$ax^2 + bx + c = 0$
the quadratic formula		$x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$
the complex number i		$i^2 = -1$
the Cahen–Mellin integral		$e^{-y} = \frac{1}{2\pi i} \int_{c-i\infty}^{c+i\infty} \Gamma(s) y^{-s}\, ds$

The *"Be-Comp"* rule can also identify if a formula is related to the concept.

After that, we group each formula F in the same paragraph with concept C and its description D to form a candidate (C, D, F). Table 1 presents candidate examples. Because we only choose paragraphs that contain at least one formula, every concept has a formula attached to it. In order to judge the correctness of candidates, we use the text-matching module, described in the next section.

2.3 Text Matching

In this step, we classify candidates using surface text. Given a list of candidates of the form (C, D, F), this module judges if a candidate is qualified by using the surface text in concept, description and formula. Because many formulas share the same variable names or function names (or part of these names) with their concepts (e.g. the first two candidates in Table 1), we filter these candidates using surface text matching.

We define the similarity between concept C, description D and formula F by the number of overlapped words, as in Eq. 1.

$$sim(\mathrm{F}, \mathrm{CD}) = \frac{|\mathrm{T_F} \cap \mathrm{T_C}|}{\min\{|\mathrm{T_C}|, |\mathrm{T_F}|\}} + \frac{|\mathrm{T_F} \cap \mathrm{T_D}|}{\min\{|\mathrm{T_D}|, |\mathrm{T_F}|\}} \quad (1)$$

T_F, T_C and T_D are sets of words extracted from F, C and D, respectively.

Candidates with $sim(\mathrm{F}, \mathrm{CD})$ no larger than a threshold θ_1 (1/3 in this study) are grouped into the group C_{true}. The rest are filtered and stored in C_0. In this step, function words such as articles, pronouns, conjunctions and so on in concepts and descriptions are ignored. Common operators in formulas are also converted to text, such as *'+' 'plus', '−' 'minus', '\frac' 'divide'*.

Using only concepts for text matching with formulas might leave out various important relations. For example, from the description of the first and second formula in Table 1, we could extract the variable names *"opposite"*, *"adjacent"* and *"hypotenuse"*.

By adding the description, we could get a more accurate judgment of whether the concept and the formula are coreferent. In this case, we can consider the concept, description and the formula form a coreference chain.

After this step, we have two categories, C_{true} and C_0. C_{true} contains qualified candidates while C_0 contains candidates that cannot be determined by text matching. The formulas in C_0 have little or no text relation with their concepts and descriptions. Thus, we can only judge the correctness of these candidates by using the text around the concepts, descriptions and formulas. The surrounding text can be formed into patterns and are generated in the next step.

2.4 Pattern Generation

One difficulty in judging the correctness of a candidate is that the formula does not share any relation with its concept and description. The third candidate in Fig. 1 is an example. It should be classified as a qualified instance but is left behind in C_0 after the "text matching" step.

In this step, we use the qualified instances in C_{true} to generate patterns. These patterns are used in the next step to judge the candidates in C_0. Patterns are generated as follows. First, the concept, description and formula are replaced by CONC, DESC and FORM, respectively. We then simply take the entire string between the first and the last appearance of CONC, DESC and FORM.

Table 2 presents examples of patterns extracted from group C_{true}.

Table 2: Examples of extracted patterns

Pattern
CONC is DESC: FORM
CONC is DESC. In our case FORM
CONC is DESC. So, ..., FORM
CONC FORM
CONC is denoted by FORM
CONC is given by ... FORM
CONC can be written as ... : FORM
FORM where CONC is DESC
FORM satisfies CONC

Using a window surrounding the concepts and formulas often leads to exponential growth in patterns, so we limit our patterns to those between any concept C, description D or formula F.

The patterns we obtained above are exactly the shortest paths from the C nodes to their F node in the parse tree. Figure 3 presents examples of these patterns in parse trees.

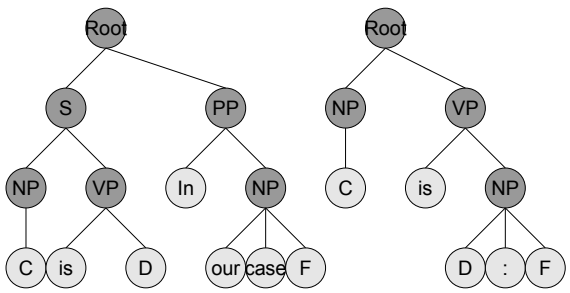

Figure 3: Examples of extracted patterns

2.5 Pattern Matching

In this step, we use patterns obtained from the previous step to classify more candidates in C_0. We use the string distance between the patterns, where candidates' patterns having a string distance to any of the patterns extracted in the previous step no larger than the threshold θ_2 are added into C_{true}.

3 Experiments

3.1 Data

We collected a total of 16,406 mathematical documents from the Wikipedia Mathematics Portal. After the preprocessing step, we selected 72,084 paragraphs that contain at least one formula. From these paragraphs, we extracted 931,716 candidates.

Because no labeled data are available for use in this task, we randomly chose 100 candidates: 60 candidates from C_{true} after the text matching step, 20 candidates added to C_{true} after pattern matching with $\theta_2 = 0$, and 20 candidates added to C_{true} after pattern matching with $\theta_2 = 0.25$ for our evaluation. These candidates were annotated manually. The sizes of the sample sets for human judgment (60, 20 and 20) were selected approximately proportional to the sizes of the obtained candidate sets.

3.2 Results

After the text matching step, we obtained 138,285 qualified candidates in the C_{true} group and 793,431 candidates in C_0. In C_{true}, we had 6,129 different patterns. Applying these patterns to C_0 by exact pattern matching ($\theta_2 = 0$), we obtained a further 34,148 qualified candidates. We obtained an additional 30,337 qualified candidates when we increased the threshold θ_2 to 0.25.

For comparison, we built a baseline system. The baseline automatically groups nearest formula and concept. It had 51 correctly qualified candidates. The results—displayed in Table 3 and depicted in Figure 4—show that our proposed method is significantly better than the baseline in terms of accuracy.

As we can see from the results, when we lower the threshold, more candidates are added to C_{true}, which means we get more formulas and formula names; but it also lowers the accuracy. Although the performance is not as high as other existing coreference resolution techniques, the proposed

Table 3: Results of the system

Module	No. correct/ total	No. of CDF found
Text Matching	41 / 60	138,285
Pattern Matching $\theta_2 = 0$	52 / 80	172,433
Pattern Matching $\theta_2 = 0.25$	56 / 100	202,270

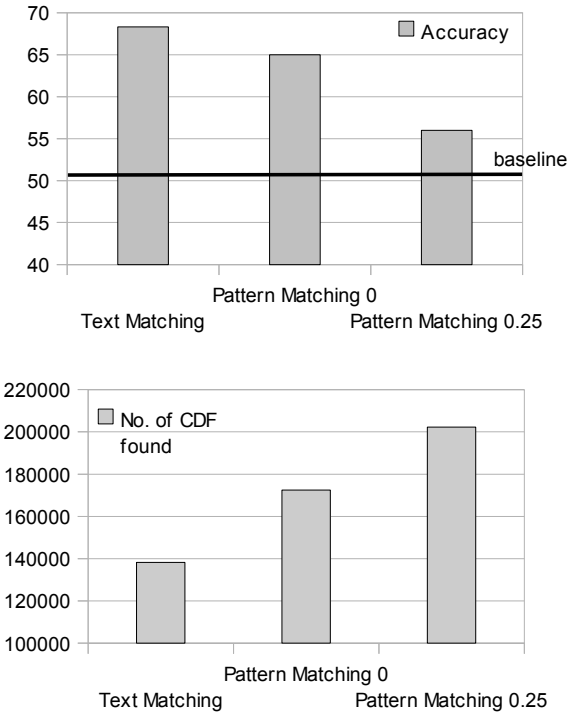

Figure 4: Results of the system

method is a promising starting point for solving coreference relations between formulas and surrounding text.

4 Conclusions

In this paper, we discuss the problem of discovering coreference relations between formulas and the surrounding texts. Although we could only use a small number of annotated data for the evaluation in this paper, our preliminary experimental results showed that our approach based on surface text-based matching between formulas and text, as well as patterns representing relationships between them showed promise for mining mathematical knowledge from Wikipedia. Since this is the first attempt to extract coreference relations between formulas and texts, there is room for further improvement. Possible improvements include: (1) using advanced technology for pattern matching to improve the coverage of the result and (2) expanding the work by mining knowledge from the Web.

References

Eyal Shnarch, Libby Barak and Ido Dagan. 2009. *Extracting Lexical Reference Rules from Wikipedia* Proceedings of the 47th Annual Meeting of the ACL and the 4th IJCNLP of the AFNLP, pages 450–458

Yulan Yan, Naoaki Okazaki, Yutaka Matsuo, Zhenglu Yang and Mitsuru Ishizuka. 2009. *Unsupervised Relation Extraction by Mining Wikipedia Texts Using Information from the Web* Proceedings of the 47th Annual Meeting of the ACL and the 4th IJCNLP of the AFNLP, pages 1021–1029

Simone Paolo Ponzetto and Massimo Poesio. 2009. *State-of-the-art NLP Approaches to Coreference Resolution: Theory and Practical Recipes* Tutorial Abstracts of ACL-IJCNLP 2009, page 6

Minh Nghiem, Keisuke Yokoi and Akiko Aizawa. 2009. *Enhancing Mathematical Search with Names of Formulas* The Workshop on E-Inclusion in Mathematics and Science 2009, pages 22–25

Keisuke Yokoi and Akiko Aizawa. 2009. *An Approach to Similarity Search for Mathematical Expressions using MathML* 2nd workshop Towards a Digital Mathematics Library, pages 27–35

Hui Siu Cheung Muhammad Adeel and Sikandar Hayat Khiyal. 2008. *Math Go! Prototype of a Content Based Mathematical Formula Search Engine* Journal of Theoretical and Applied Information Technology, Vol. 4, No. 10, pages 1002–1012

Simone Paolo Ponzetto and Michael Strube. 2006. *Exploiting Semantic Role Labeling, WordNet and Wikipedia for Coreference Resolution* In Proceedings of HLT-NAACL-06, pages 192–199

Jim Giles. 2005. *Internet Encyclopaedias Go Head to Head* Nature Volume: 438, Issue: 7070, pages 900–901

World Wide Web Consortium. *Mathematical Markup Language (MathML) version 2.0 (second edition)* http://www.w3.org/TR/MathML2/