Discourse in Multimedia: A Case Study in Extracting Geometry Knowledge from Textbooks

Mrinmaya Sachan
Carnegie Mellon University
School of Computer Science
Machine Learning Department
mrinmays@cs.cmu.edu

Avinava Dubey
Carnegie Mellon University
School of Computer Science
Machine Learning Department

Eduard H. Hovy
Carnegie Mellon University
School of Computer Science
Language Technologies Institute

Tom M. Mitchell
Carnegie Mellon University
School of Computer Science
Machine Learning Department

Dan Roth
University of Pennsylvania
Department of Computer and
Information Science

Eric P. Xing
Carnegie Mellon University
School of Computer Science
Machine Learning Department

Submission received: 25 August 2018; revised version received: 3 January 2019; accepted for publication: 10 February 2019.

https://doi.org/10.1162/COLI_a_00360

To ensure readability, text is often written and presented with due formatting. These text formatting devices help the writer to effectively convey the narrative. At the same time, these help the readers pick up the structure of the discourse and comprehend the conveyed information. There have been a number of linguistic theories on discourse structure of text. However, these theories only consider unformatted text. Multimedia text contains rich formatting features that can be leveraged for various NLP tasks. In this article, we study some of these discourse features in multimedia text and what communicative function they fulfill in the context. As a case study, we use these features to harvest structured subject knowledge of geometry from textbooks. We conclude that the discourse and text layout features provide information that is complementary to lexical semantic information. Finally, we show that the harvested structured knowledge can be used to improve an existing solver for geometry problems, making it more accurate as well as more explainable.

1. Introduction

The study of discourse focuses on the properties of text as a whole and how meaning is conveyed by making connections between component sentences. Writers often use certain linguistic devices to make a discourse structure that enables them to effectively communicate their narrative. The readers, too, comprehend text by picking up these linguistic devices and recognizing the discourse structure. There are a number of linguistic theories on discourse relations (Van Dijk 1972; Longacre 1983; Grosz and Sidner 1986; Cohen 1987; Mann and Thompson 1988; Polanyi 1988; Moser and Moore 1996) that specify relations between discourse units and how to represent the discourse structure of a piece of text (i.e., discourse parsing; Duverle and Prendinger 2009; Subba and Di Eugenio 2009; Feng and Hirst 2012; Gosh, Riccardi, and Johansson 2012; Feng and Hirst 2014; Ji and Eisenstein 2014; Li et al. 2014; Li, Ng, and Kan 2014; Wang and Lan 2015). These discourse features have been shown to be useful in a number of NLP applications such as summarization (Dijk 1979; Marcu 2000; Boguraev and Neff 2000; Louis, Joshi, and Nenkova 2010; Gerani et al. 2014), information retrieval (Wang et al. 2006; Lioma, Larsen, and Lu 2012), information extraction (Kitani, Eriguchi, and Hara 1994; Conrath et al. 2014), and question answering (Chai and Jin 2004; Sun and Chai 2007; Narasimhan and Barzilay 2015; Sachan et al. 2015).

Most linguistic theories of discourse consider written text without much formatting. However, in this multimedia age, text is often richly formatted. Be it newsprint, textbooks, brochures, or even scientific articles, text is usually appropriately formatted and stylized. For example, the text may have a heading. It may be divided into a number of sections with section subtitles. Parts of the text may be italicized or boldfaced to place appropriate emphasis wherever required. The text may contain itemized lists, footnotes, indentations, or quotations. It may refer to associated tables and figures. The tables and figures, too, usually have associated captions. All these text layout features ensure that the text is easy to read and understand. Even articles accepted for *Computational Linguistics* follow a due formatting scheme.

These text layout features are in addition to other linguistic devices such as syntactic arrangement or rhetorical forms. Relations between textual units that are not necessarily contiguous can thus be expressed thanks to typographical or dispositional markers. Such relations, which are out of reach of standard NLP tools, have only been studied within some specific layout contexts (Hovy 1998; Pascual 1996; Bateman et al. 2001a,

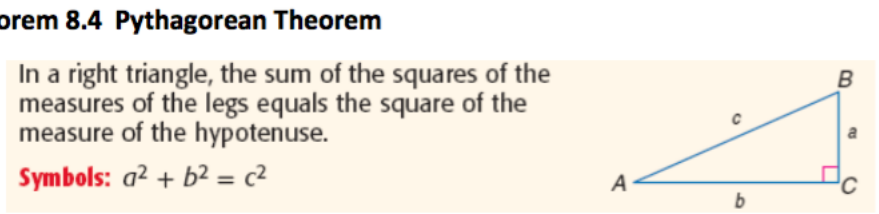


Figure 1
An excerpt of a textbook from our data set that introduces the Pythagorean theorem. The textbook has many typographical features that can be used to harvest this theorem: The textbook explicitly labels it as a "theorem"; there is a colored bounding box around it; an equation sets down the rule and there is a supporting figure. Our model leverages such rich contextual and typographical information (when available) to accurately harvest axioms and then parses them to horn-clause rules.

inter alia)[1] and there are not many comprehensive studies on the various kinds of discourse features and how they can be leveraged to improve NLP tasks.

In this article, we study some of these discourse features in multimedia text and what communicative function they fulfill in the context. As a case study, we study the problem of harvesting structured subject knowledge of geometry from textbooks and show that the formatting devices can indeed be used to improve a strong information extraction system in that domain. We show that the discourse and text layout features provide information that is complementary to lexical semantic information commonly used for information extraction.

With the intent of making the subject material easy to grasp and remember for students, textbooks often contain rich discourse and formatting features. Crucial material such as axioms or theorems are presented with stylistic highlighting or bounding boxes. Often, mathematical information such as equations are presented in a separate color and font size. Often, theorems are numbered or named (e.g., Theorem 8.4). For example, Figure 1 shows a snapshot of a math textbook that describes the Pythagorean theorem. The textbook explicitly labels it as a "theorem"; there is a colored bounding box around it; an equation sets down the rule and there is a supporting figure. In this article, we will try to answer the question: *Can this rich contextual and typographical information (whenever available) be used to harvest these axioms in the form of structured rules?* Our goal is to not only extract the axiom mentioned in Figure 1 but also map it to a rule corresponding to the Pythagorean theorem:

$$isTriangle(ABC) \wedge perpendicular(AC, BC) \implies BC^2 + AC^2 = AB^2$$

We present an automatic approach that can (a) harvest such subject knowledge from textbooks, and (b) parse the extracted knowledge to structured rules. We propose novel models that perform sequence labeling and alignment to extract redundant axiom mentions across various textbooks, and then parse the redundant axioms to structured rules. These redundant structured rules are then resolved to achieve the best correct structured rule for each axiom. We conduct a comprehensive feature analysis of the usefulness of various discourse features: shallow discourse features based on discourse markers, a deep one based on *Rhetorical Structure Theory* (Mann and Thompson 1988),

1 Please see related work (Section 2) for a complete list of references.

and various text layout features in a multimedia document (Hovy 1998) for the various stages of information extraction. Our experiments show the usefulness of all the various typographical features over and above the various lexical semantic and discourse level features considered for the task.

We use our model to extract and parse axiomatic knowledge from a novel data set of 20 publicly available math textbooks. We use this structured axiomatic knowledge to build a new axiomatic solver that performs logical inference to solve geometry problems. Our axiomatic solver outperforms *GEOS* on all existing test sets introduced in Seo et al. (2015) as well as a new test set of geometry questions collected from these textbooks. We also performed user studies on a number of school students studying geometry who found that our axiomatic solver is more interpretable and useful compared with GEOS.

2. Background and Related Work

Discourse Analysis: Discourse analysis is the analysis of semantics conveyed by a coherent sequence of sentences, propositions, or speech. Discourse analysis is taken up in a variety of disciplines in the humanities and social sciences and a number of discourse theories have been proposed (Mann and Thompson 1988; Kamp and Reyle 1993; Lascarides and Asher 2008, inter alia). Their starting point lies in the idea that text is not just a collection of sentences, but also includes relations between all these sentences that ensure its coherence. It is often assumed that discourse analysis is a three-step process:

1. splitting the text into discourse units (DUs),
2. ensuring the attachment between DUs, and then
3. labeling links between DUs with discourse relations.

Discourse relations may be divided into two categories: nucleus-satellite (or subordinate) relations, which link an important argument to an argument supporting background information, and multinuclear (or coordinate) relations, which link arguments of equal importance. Most discourse theories (DRT, RST, SDRT, etc.) acknowledge that a discourse is hierarchically structured thanks to discourse relations. A number of discourse relations have been proposed under various theories for discourse analysis.

Discourse analysis has been shown to be useful for many NLP tasks, such as question answering (Chai and Jin 2004; Lioma, Larsen, and Lu 2012; Jansen, Surdeanu, and Clark 2014), summarization (Louis, Joshi, and Nenkova 2010), and information extraction (Kitani, Eriguchi, and Hara 1994). However, to the best of our knowledge, we do not have a theory or a working model of discourse in a multimedia setting.

Formatting in Discourse: Psychologists and educationists have frequently studied multimedia issues such as the impact of illustrations (pictures, tables, etc.) in text, design principles of multimedia presentations, and so forth (Dwyer 1978; Fleming, Levie, and Levie 1978; Hartley 1985; Twyman 1985). However, these discussions are usually too general and hard to build on from a computational perspective. Thus, most studies of multimedia text have only been theoretical in nature. Larkin and Simon (1987), Mayer (1989), and Petre and Green (1990) attempt to answer questions: whether a graphical notation is superior to text notation, what makes a diagram (sometimes) worth ten

thousand words, how illustration effects thinking. Hovy (1998), Arens and Hovy (1990), Arens (1992), and Arens, Hovy, and Van Mulken (1993) provide a theory of the communicative function fulfilled by various formatting devices and use it in text planning. In a similar vein, Dale (1991b, a), White (1995), Pascual and Virbel (1996), Reed and Long (1997), and Bateman et al. (2001b) discuss the textual function of punctuation marks and use it in the text generation process. André et al. (1991) and André (2000) build a system *WIP* that generates multimedia presentations via layered architecture (composed of the control layer, content layer, design layer, realization layer, and the presentation layer) and with the help of various content, design, user, and application experts. Mackinlay (1986) discuss the automatic generation of tables and charts. Luc, Mojahid, and Virbel (1999) study enumerations. Feiner (1988), Arens et al. (1988), Neal et al. (1990), Feiner and McKeown (1991), Wahlster et al. (1992), Arens, Hovy, and Vossers (1992), and Maybury (1998) discuss various aspects of processing and knowledge required for automatically generating multimedia. Finally, Stock (1993) discusses using hypermedia features for the task of information exploration.

However, all the aforementioned studies were merely theoretical. All the models were hand-coded and not trained from multimedia corpora. In this paper, we provide a corpus analysis of multimedia text and use it to show that the formatting devices can indeed be used to improve a strong information extraction system in the geometry domain.

Solving Geometry Problems: Although the problem of using computers to solve geometry questions is old (Feigenbaum and Feldman 1963; Schattschneider and King 1997; Davis 2006), NLP and computer vision techniques were first used to solve geometry problems in Seo et al. (2015). Seo et al. (2014) only aligned geometric shapes with their textual mentions, but Seo et al. (2015) also extracted geometric relations and built *GEOS*, the first automated system to solve SAT style geometry questions. *GEOS* used a coordinate geometry based solution by translating each predicate into a set of manually written constraints. A Boolean satisfiability problem posed with these constraints was used to solve the multiple-choice question. *GEOS* had two key issues: (a) It needed access to answer choices that may not always be available for such problems, and (b) It lacked the deductive geometric reasoning used by students to solve these problems. In this article, we build an axiomatic solver that mitigates these issues by performing deductive reasoning using axiomatic knowledge extracted from textbooks. Furthermore, we use ideas from discourse to automatically extract these axiom rules from textbooks.

Automatic approaches that use logical inference for geometry theorem proving, such as the Wus method (Wen-Tsun 1986), Grobner basis method (Kapur 1986), and angle method (Chou, Gao, and Zhang 1994), have been used in tutoring systems such as *Geometry Expert* (Gao and Lin 2002) and *Geometry Explorer* (Wilson and Fleuriot 2005). There has also been research in synthesizing geometry constructions, given logical constraints (Gulwani, Korthikanti, and Tiwari 2011; Itzhaky et al. 2013) or generating geometric proof problems (Alvin et al. 2014) for applications in tutoring systems. Our approach can be used to provide the axiomatic information necessary for these works.

Other Related Tasks: Our work is also related to Textbook Question Answering (Kembhavi et al. 2017), which proposes the task of multimodal machine comprehension where the context needed to answer questions composes of both text and images. The TQA data set is built from middle school science textbooks and pairs a given question to a limited span of knowledge needed to answer it. Also related is the work on Diagram

QA (Kembhavi et al. 2016), which proposes the task of understanding and answering questions based on diagrams from textbooks, and FigureSeer (Siegel et al. 2016), which parses figures in research papers.

Information Extraction from Textbooks: Our model for extracting structured rules of geometry from textbooks builds upon ideas from information extraction (IE), which is the task of automatically extracting structured information from unstructured and/or semi-structured documents. Although there has been a lot of work in IE on domains such as Web documents (Chang, Hsu, and Lui 2003; Etzioni et al. 2004; Cafarella et al. 2005; Chang et al. 2006; Banko et al. 2007; Etzioni et al. 2008; Mitchell et al. 2015) and scientific publication data (Shah et al. 2003; Peng and McCallum 2006; Saleem and Latif 2012), work on IE from educational material is much more sparse. Most of the research in IE from educational material deals with extracting simple educational concepts (Shah et al. 2003; Canisius and Sporleder 2007; Yang et al. 2015; Wang et al. 2015; Liang et al. 2015; Wu et al. 2015; Liu et al. 2016b; Wang et al. 2016) or binary relational tuples (Balasubramanian et al. 2002; Clark et al. 2012; Dalvi et al. 2016) using existing IE techniques. On the other hand, our approach extracts axioms and parses them to horn-clause rules. This is much more challenging. Raw application of rule mining or sequence labeling techniques used to extract information from Web documents and scientific publications to educational material usually leads to poor results as the amount of redundancy in educational material is lower and the amount of labeled data is sparse. Our approach tackles these issues by making judicious use of typographical information, the redundancy of information, and ordering constraints to improve the harvesting and parsing of axioms. This has not been attempted in previous work.

Language to Programs: After harvesting axioms from textbooks, we also parse the axiom mentions to horn-clause rules. This work is related to a large body of work on semantic parsing (Zelle and Mooney 1993, 1996; Kate et al. 2005; Zettlemoyer and Collins 2012, inter alia). Semantic parsers typically map natural language to formal programs such as database queries (Liang, Jordan, and Klein 2011; Berant et al. 2013; Yaghmazadeh et al. 2017, inter alia), commands to robots (Shimizu and Haas 2009; Matuszek, Fox, and Koscher 2010; Chen and Mooney 2011, inter alia), or even general purpose programs (Lei et al. 2013; Ling et al. 2016; Yin and Neubig 2017; Ling et al. 2017). More specifically, Liu et al. (2016a) and Quirk, Mooney, and Galley (2015) learn "If-Then" and "If-This-Then-That" rules, respectively. In theory, these works can be adapted to parse axiom mentions to horn-clause rules. However, this would require a large amount of supervision, which would be expensive to obtain. We mitigated this issue by using redundant axiom mention extractions from multiple textbooks and then combining the parses obtained from various textbooks to achieve a better final parse for each axiom.

3. Data Format

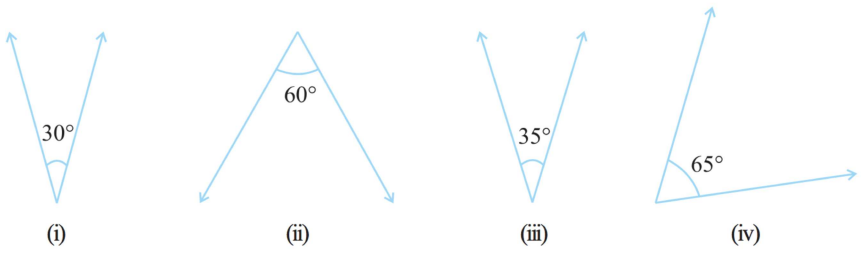
Large-scale corpus studies of multimedia text have been rare because of the difficulty in obtaining rich multimedia documents in analyzable data structures. A large proportion of text today is typeset using some typesetting software such as LaTeX, Word, HTML, and so on. These features can also serve as useful cues in downstream applications and a model for text formatting is required.

Table 1
Some more excerpts of textbooks from our data set that describe (a) complementary angles, (b) exterior angles, and (c) parallelogram diagonal bisection axioms. Each excerpt contains rich typographical features that can be used to harvest the axioms. (a) For the complementary angles mention, the textbook explicitly labels the section name "5.2.1 Complementary Angles" with boldface and color; the axiom name "complementary angles" is in bold font, and there is a supporting figure. (b) For the exterior angles mention, the axiom statement is boldfaced, the axiom rule is mentioned via an equation (which is emphasized with the boldfaced string "To show"), and there is a supporting figure. (c) For the parallelogram diagonal bisection mention, the axiom statement is emphasized with the boldfaced string "Property," the axiom statement itself is italicized, there is a supporting figure, and the axiom rule is written as an equation. Our model will leverage such rich contextual and typographical information (when available) to accurately harvest axioms and then parses them to horn-clause rules.

(a)

5.2.1 Complementary Angles

When the sum of the measures of two angles is 90°, the angles are called **complementary angles.**



Are these two angles complementary? Yes

Are these two angles complementary? No

Fig 5.4

Whenever two angles are complementary, each angle is said to be the **complement** of the other angle. In the above diagram (Fig 5.4), the '30° angle' is the complement of the '60° angle' and vice versa.

(b)

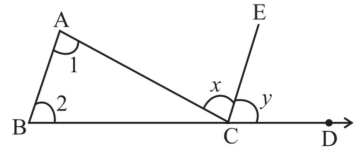
An exterior angle of a triangle is equal to the sum of its interior opposite angles.

Given: Consider ΔABC.

∠ACD is an exterior angle.

To Show: $m\angle ACD = m\angle A + m\angle B$

Through C draw $\overline{CE}$, parallel to $\overline{BA}$.



(c)

Property: *The diagonals of a parallelogram bisect each other (at the point of their intersection, of course!)*

To argue and justify this property is not very difficult. From Fig 3.30, applying ASA criterion, it is easy to see that

Δ AOB ≅ Δ COD (How is ASA used here?)

This gives AO = CO and BO = DO

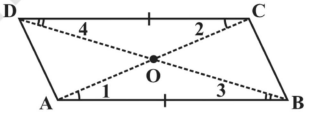


Fig 3.30

Table 2
Corresponding JSON file for the example textbook excerpts shown in Figure 1. We mark the various typographical features that can be used to harvest the axioms in red: features such as the heading, the bounding box, a supporting figure, and the equation.


```
{
"frame-type": "heading",
"text": "Theorem 8.4 Pythagorean Theorem",
"font-style"="bold"
}
{
"frame-type": "bounding box",
"box color": "yellow",
"box-elements":[
        {
        "frame-type": "text",
        "frame-style": {
        "font-size"=24px,
        "font-style"="normal",
        },
        "text": "In a right triangle, the sum of squares of the measures of the legs equals the square of the measure of the hypotenuse."
        },
        {
        "frame-type": "text",
        "frame-style": {
        "font-size"=24px,
        "font-style"="mixed",
        "bold-faced"="1-2"
        },
        "text": Symbols: pow(a, 2) + pow(b, 2) = pow(c, 2)"
        },
        {
        "frame-type": "figure",
        "figure-url": "14133.jpg"
        }
]
}
```

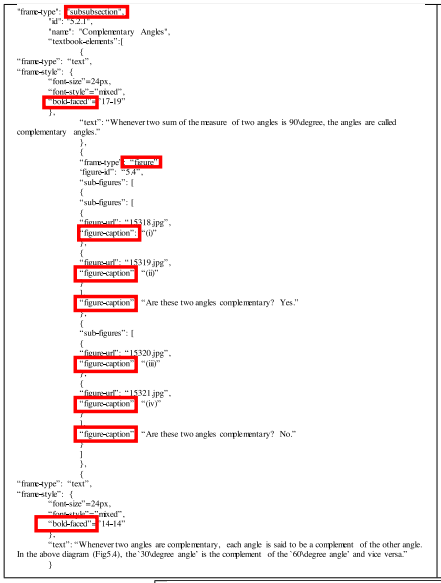

Table 1 shows some excerpts of textbooks from our data set that describe complementary angles, exterior angles, and parallelogram diagonal bisection axioms. As described, each excerpt contains rich typographical features, such as the section headings, italicization, boldface, coloring, explicit axiom name, supporting figures, and equations that can be used to harvest the axioms. We wish to leverage such rich contextual and typographical information to accurately harvest axioms and then parse them to horn-clause rules. The textbooks are provided to us in rich JSON format, which retains the rich typesetting of these textbooks as shown in Tables 2 and 3. For demonstration, we have manually marked the various typographical features that can be used to harvest the axioms. We will show how we can use these features to harvest axioms of geometry from textbooks and then parse them to structured rules.

4. Text Formatting Elements in Discourse

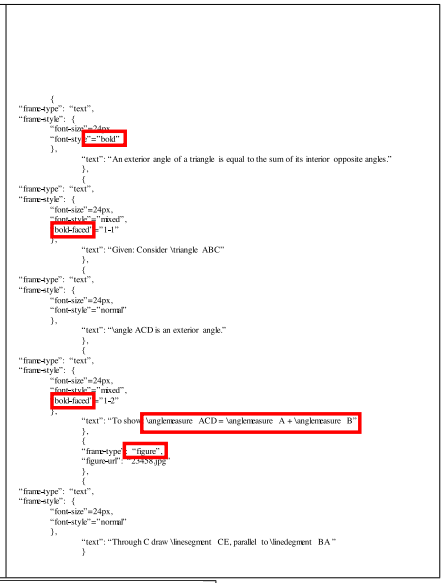
In this section, we review various text formatting devices used in a typical multimedia system and identify what communicative function they serve. This will help us come up with a theory for text formatting in discourse and also motivate how these features can be used in a typical NLP application like information extraction. This theory is inspired from various style suggestions for English writing (Strunk 2007). The goal of a text formatting device in a multimedia text is to delimit the portion of text for which certain exceptional conditions of interpretation hold. We categorize text formatting devices into

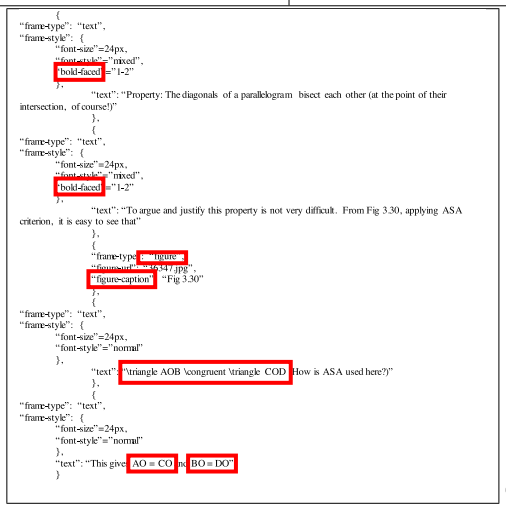
Table 3
Corresponding JSON files for the example textbook excerpts shown in Table 1. We mark the various typographical features that can be used to harvest the axioms in red: (a) For the complementary angles mention, we have features such as the subsubsection "5.2.1 Complementary Angles" with boldface and color; the axiom name "complementary angles" is in bold font, and there is a supporting figure. (b) For the exterior angles mention, the axiom statement is boldfaced, the axiom rule is mentioned via an equation (which is emphasized with the boldfaced string "To show"), and there is a supporting figure. (c) For the parallelogram diagonal bisection mention, the axiom statement is emphasized with the boldfaced string "Property," the axiom statement itself is italicized, there is a supporting figure, and the axiom rule is written as an equation.

(a)



(b)





(c)

four broad categories: *depiction*, *position*, *composition*, and *substantiation*, and describe the various text formatting devices here:

- **Depiction:** Depiction features concern with how a string of text is presented in the multimedia. These include features such as capitalization, font size/color, boldface, italicization, underline, strikethrough, parenthesis, quotation marks, use of bounding boxes, and so forth.
- **Position:** Position features concern with the positioning of a piece of text relative to the remaining material in the document. These features include in-lining, text offset, footnotes, headers and footers, text separation or isolation (a block of text separated from the rest to create a special effect).
- **Composition:** Composition features are concerned with the internal structuring of a piece of text. Examples include graphical markers such as paragraph breaks, sections (having sections, chapters, etc., in the document), lists (itemization, enumeration), concept definition using a parenthesis or colon, and so on.
- **Substantiation:** Substantiation features are used to further substantiate the discourse argument. Examples include associated figures or tables, references to tables, figures (e.g., Figure 1.2), or external links that are very important in understanding a complex multimedia document.

5. Text Formatting Features for Information Extraction?

A key question for research is: *Are these text formatting features useful for NLP tasks?* In particular, in this article, we will try to identify whether these text formatting features are useful for information extraction. In a typical multimedia document, authors use various text formatting devices to better communicate the content to their readers. This helps the readers digest the material quickly and much more easily. Thus, can these text formatting features be useful in an information extraction system too? We experimentally validate our hypothesis in the application of harvesting axioms of geometry from richly formatted textbooks.

Then, we show that these harvested axioms can improve an existing solver for answering SAT style geometry problems. SAT geometry tests the student's knowledge of Euclidean geometry in its classical sense, including the study of points, lines, planes, angles, triangles, congruence, similarity, solid figures, circles, and analytical geometry. A typical geometry problem is provided in Figure 2. Geometry questions include a textual description accompanied by a diagram. Various levels of understanding are required to solve geometry problems. An important challenge is understanding both the diagram (which consists of identifying visual elements in the diagram, their locations, their geometric properties, etc.) and the text simultaneously, and then reasoning about the geometrical concepts using well-known axioms of Euclidean geometry.

We first recap *GEOS*, a completely automatic solver for geometry problems. We will then use the rich contextual and typographical information in textbooks to extract structured knowledge of geometry. This structured knowledge of geometry will then be used to improve *GEOS*.

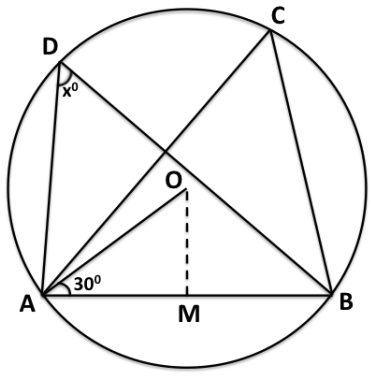


Figure 2
An example SAT style geometry problem. The problem consists of a diagram as well as the question text. In order to solve such a question, the system is required to understand both the diagram as well as the question text, and also reason about geometrical concepts using well-known axioms of Euclidean geometry.

6. Background: GEOS

Our work reuses *GEOS* (Seo et al. 2015) to parse the question text and diagram into its formal problem description as shown in Figure 3. *GEOS* uses a logical formula, a first-order logic expression that includes known numbers or geometrical entities (e.g., 4 cm) as constants, unknown numbers or geometrical entities (e.g., O) as variables, geometric or arithmetic relations (e.g., *isLine*, *isTriangle*) as predicates, and properties of geometrical entities (e.g., *measure*, *liesOn*) as functions.

This is done by learning a set of relations that potentially correspond to the question text (or the diagram) along with a confidence score. For diagram parsing, *GEOS* uses a publicly available diagram parser for geometry problems (Seo et al. 2014) to obtain the set of all visual elements, their coordinates, their relationships in the diagram, and their

Text Description:	**Diagram:**
measure(∠MAO, 30°)	liesOn(A, circle O), liesOn(B, circle O),
isCircle(O)	liesOn(C, circle O), liesOn(D, circle O)
radius(O, 4 cm)	isLine(AB), isLine(BC), isLine(CA), isLine(BD), isLine(DA)
?x	isTriangle(ABC), isTriangle(ABD), isTriangle(AOM)
	measure(∠ADB, x), measure(∠MAO, 30°)
	measure(∠AMO, 90°)
	…

Figure 3
A logical expression that represents the meaning of the text description and the diagram in the geometry problem in Figure 2. *GEOS* derives a weighted logical expression where each predicate also carries a weighted score, but we do not show them here for clarity.

alignment with entity references in the question text. The diagram parser also provides confidence scores for each literal to be true in the diagram. For text parsing, *GEOS* takes a multistage approach, which maps words or phrases in the text to their corresponding concepts, and then identifies relations between identified concepts.

Given this formal problem description, *GEOS* uses a numerical method to check the satisfiablity of literals by defining a relaxed indicator function for each literal. These indicator functions are manually engineered for every predicate. Each predicate is mapped into a set of constraints over point coordinates.[2] These constraints can be non-trivial to write, requiring significant manual engineering. As a result, *GEOS*'s constraint set is incomplete and it cannot solve a number of SAT style geometry questions. Furthermore, this solver is not interpretable. As our user studies show, it is not natural for a student to understand the solution of these geometry questions in terms of satisfiability of constraints over coordinates. A more natural way for students to understand and reason about these questions is through deductive reasoning using axioms of geometry.

7. Set-up for the Axiomatic Solver

To tackle the aforementioned issues with the numerical solver in *GEOS*, we replace the numerical solver with an axiomatic solver. We extract axiomatic knowledge from textbooks and parse them into horn-clause rules. Then we build an axiomatic solver that performs logical inference with these horn-clause rules and the formal problem description. A sample logical program (in prolog notation) that solves the problem in Figure 2 is given in Figure 4. The logical program has a set of declarations from the *GEOS* text and diagram parsers that describe the problem specification; and the parsed horn-clause rules describe the underlying theory. Normalized confidence scores from question text and diagram and axiom parsing models are used as probabilities in the program. Figure 5 shows a block diagram of the overall system that solves geometry problems. Also, Figure 6 pictorially shows the two step procedure for obtaining structured axiomatic knowledge from textbooks:

1. **Axiom Identification and Alignment:** In this stage, we identify axiom mentions in all textbooks and align the mentions of the same axiom across different textbooks.
2. **Axiom Parsing:** In this stage, we parse each of these axiom mentions into implication rules and then resolve the implication rules for various axiom mentions referring to the same axiom mention.

Next, we describe how we harvest structured axiomatic knowledge from textbooks.

8. Harvesting Axiomatic Knowledge

We present a structured prediction model that identifies axioms in textbooks and then parses them. Because harvesting axioms from a single textbook is a very hard problem, we use multiple textbooks and leverage the redundancy of information to accurately

2 For example, the predicate *isPerpendicular*(AB, CD) is mapped to the constraint $\frac{y_B - y_A}{x_B - x_A} \times \frac{y_D - y_C}{x_D - x_C} = -1$.

```
Datastructures
    sort point = {A, B, C, D, O, M}
    sort line = {AB, BC, CA, BD, DA, OA, OM} //Symmetrically define BA, CB, ...
    sort angle = {ABC, BCA, CAB, ABD, BDA, DAB, AMO, MOA, OAM, BMO} //Symmetrically define CBA, ACB, ...
    sort triangle = {ABC, ABD, AMO} //Symmetrically define CBA, ACB, ...
    sort circle = {O}

Diagram Parse
    0.4 perpendicular(OM, AB)
    0.8 measure(ADB, x)
    0.9 liesOn(A, O)
    0.9 liesOn(B, O)
    0.9 liesOn(C, O)
    0.9 liesOn(D, O)
    0.9 liesOn(M, AB)
    0.9 liesInInterior(M, AOB)

Text Parse
    0.9 measure(OAM, 30)
    0.9 measure(radius(O), 4 cm)
    0.9 query(x, _)

Axiomatic Rules
1   0.8 measure(ABC, 90.0) :- perpendicular(AB, CD), liesOn(B, CD)
2   0.8 measure(XAC, 180-t) :- liesOn(A, BC), measure(XAB, t)
3   0.7 equals(length(AX), length(XB)) :- liesOn(A, O), liesOn(B, O), perpendicular(OX, AB), liesOn(X, AB)
4   0.7 similar(ABC, DEF) :- equals(length(BC), length(EF)), equals(measure(ABC), measure(DEF)),
        equals(measure(BCA), measure(EFD)) // ASA rule. Similar rules for SAS, SSS, RHS rules of similarity
5   0.7 equals(measure(CAB), measure(FED)) :- similar(ABC, DEF) // Similar rules for other corresponding angles
6   0.7 equals(measure(ABC), u+v)) :- equals(measure(ABD), u)), equals(measure(DBC), v)), liesInInterior(D, ABC)
7   0.6 equals(measure(ADB), t/2) :- equals(measure(AOB), t), liesOn(A, O), liesOn(B, O)
```

Figure 4
A sample logical program (in prolog style) that solves the problem in Figure 2. The program consists of a set of data structure declarations that correspond to types in the prolog program, a set of declarations from the diagram and text parse, and a subset of the geometry axioms written as horn-clause rules. The axioms are used as the underlying theory with the aforementioned declarations to yield the solution upon logical inference. Normalized confidence weights from the diagram, text, and axiom parses are used as probabilities. For reader understanding, we list the axioms in the order (1 to 7) they are used to solve the problem. However, this ordering is not required. Other (less probable) declarations and axiom rules are not shown here for clarity but they can be assumed to be present.

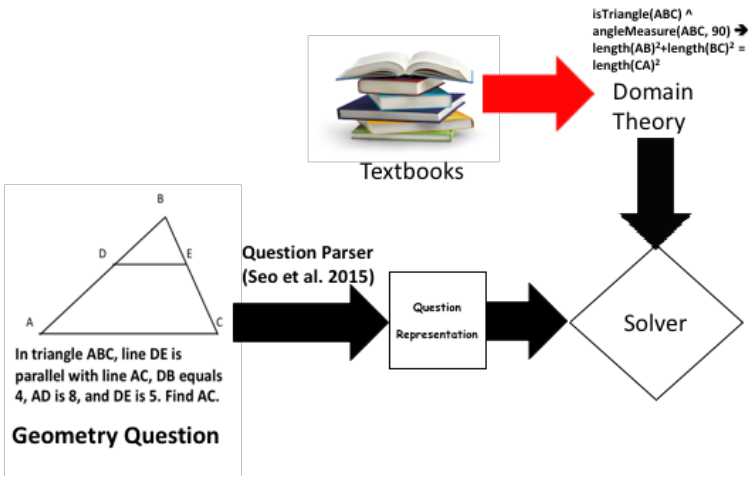


Figure 5
Block diagram of our overall system that solves geometry problems. We use *GEOS* (Seo et al. 2015) — previous work that parses geometry questions into a formal problem description. In this article, we describe an approach to harvest geometry axioms from textbooks and then parse them to rules. Then, we use an off-the-shelf prolog style probabilistic reasoner (solver) to perform logical inference with these horn-clause rules and the formal problem description to obtain the answer. Our focus in this article is on the task of harvesting knowledge of geometry from textbooks.

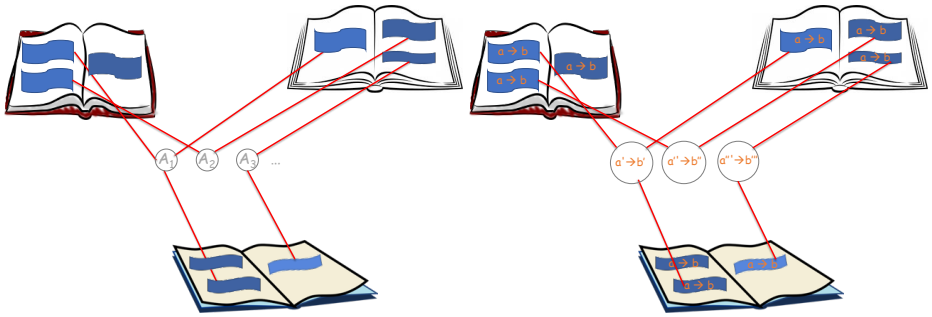


Figure 6
Pictorial representation of our two step procedure for obtaining structured axiomatic knowledge from textbooks. Left: In the first step, we identify axiom mentions in all the textbooks (shown in blue) and align the mentions of the same axiom across different textbooks (shown in red). Right: In the second step, we parse each of these identified axiom mentions into implication rules and then resolve the implication rules for various axiom mentions referring to the same axiom mention.

extract and parse axioms. We first define a joint model that identifies axiom mentions in each textbook and aligns repeated mentions of the same axiom across textbooks. Then, given a set of axioms (with possibly multiple mentions of each axiom), we define a parsing model that maps each axiom to a horn-clause rule by utilizing the various mentions of the axiom.

Given a set of textbooks $\mathcal{B}$ in machine readable form (JSON in our experiments), we extract chapters relevant for geometry in each of them to obtain a sequence of discourse elements (with associated typographical information) from each textbook. We assume that the textbook comprises an ordered set[3] of **discourse elements** where a discourse element could be a natural language sentence, heading, title, figure, table, or caption. The discourse element (e.g., a sentence) could have additional typographical features. For example, the sentence could be written in boldface, underline, and so forth. These properties of discourse elements will be useful features that can be leveraged for the task of harvesting axioms. Let $\mathbf{S}_b = \{s_0^{(b)}, s_1^{(b)}, \ldots s_{|\mathbf{S}_b|}^{(b)}\}$ denote the sequence of discourse elements in textbook b. $|\mathbf{S}_b|$ denotes the number of discourse elements in textbook b.

8.1 Axiom Identification and Alignment

We decompose the problem of extracting axioms from textbooks into two tractable sub-problems:

1. identification of axiom mentions in each textbook using sequence labeling
2. alignment of repeated mentions of the same axiom across textbooks

Then, we combine the learned models for these sub-problems into a joint optimization framework that simultaneously learns to identify and align axiom mentions.

3 Given a textbook in JSON format, we can construct this ordered set by preorder traversal of the JSON tree.

8.1.1 Axiom Identification. Linear-chain conditional random field formulation (Lafferty, McCallum, and Pereira 2001) can be used for the subproblem of axiom identification. Given $\{\mathbf{S}_b | b \in \mathcal{B}\}$, a sequence of discourse elements (with associated typographical information) from each textbook, the model labels each discourse element $s_i^{(b)}$ as **B**efore, **I**nside, or **O**utside an axiom. Hereon, a contiguous block of discourse elements labeled **B** or **I** will be considered as an axiom mention. Let $\mathcal{T} = \{\mathbf{B}, \mathbf{I}, \mathbf{O}\}$ denote the tag set. Let $y_i^{(b)}$ be the tag assigned to $s_i^{(b)}$ and $\mathbf{Y}_b$ be the tag sequence assigned to $\mathbf{S}_b$. The conditional random field defines:

$$p(\mathbf{Y}_b | \mathbf{S}_b; \boldsymbol{\theta}) \propto \prod_{k=1}^{|\mathbf{S}_b|} \exp\left(\sum_{i,j \in \mathcal{T}} \boldsymbol{\theta}_{ij}^T \mathbf{f}_{ij}(y_{k-1}^{(b)}, y_k^{(b)}, \mathcal{S}_b) \right)$$

We find the parameters $\boldsymbol{\theta}$ using maximum-likelihood estimation with L2 regularization:

$$\boldsymbol{\theta}^* = \arg\max_{\boldsymbol{\theta}} \sum_{b \in \mathcal{B}} \log p(\mathbf{Y}_b | \mathbf{S}_b; \boldsymbol{\theta}) - \lambda ||\boldsymbol{\theta}||_2^2$$

We use limited memory BFGS (L-BFGS) to optimize the objective and Viterbi decoding for inference. λ is tuned on the dev set.

Features: Features f look at a pair of adjacent tags $y_{k-1}^{(b)}$, $y_k^{(b)}$, the input sequence $\mathbf{S}_b$, and where we are in the sequence. The features (listed in Table 4) include various content-based features encoding various notions of similarity between pairs of discourse elements (in terms of semantic overlap, more refined match of geometry entities, and certain keywords) as well as various typographical features such as whether the discourse elements are annotated as an axiom (or theorem or corollary) in the textbook; contain equations, diagrams, or text that is bold or italicized; are in the same node of the JSON hierarchy; are contained in a bounding box, and so forth. We also use features directly from an existing RST parser (Feng and Hirst 2014); discourse structure can be useful to understand if two consecutive discourse elements are together part of an axiom (or not).

Some extracted axiom mentions contain pointers to a diagram (e.g., "Figure 2.1"). In all these cases, we consider the diagram to be a part of the axiom mention. We will discuss the impact of the various content- and typography-based features later in Section 11.

8.1.2 Axiom Alignment. Next, we leverage the redundancy of information and the relatively fixed ordering of axioms in various textbooks. Most textbooks typically present all axioms of geometry in approximately the same order, moving from easier concepts to more advanced concepts. For example, all textbooks will introduce the definition of a right-angled triangle before introducing the Pythagorean theorem. We leverage this structure by aligning various mentions of the same axiom across textbooks and introducing structural constraints on the alignment.

Table 4
Feature set for our axiom identification model. The features are based on content and typography.

Content	Sentence overlap	Semantic textual similarity between the current and next discourse element. We include features that compute the proportion of common unigrams and bigrams across the two discourse elements. This feature is conjoined with the tag assigned to the current and next sentence.
	Geometry entities	Number of geometry entities (constants, predicates, and functions)—normalized by the number of tokens in this discourse element. This feature is conjoined with the tag assigned to the current discourse element.
Discourse	Keywords	Indicator that the current discourse element contains any one of the following words: *hence*, *if*, *equal*, *twice*, *proportion*, *ratio*, *product*. This feature is conjoined with the tag assigned to the current discourse element.
	RST edge	Indicator for the RST relation between the current and next discourse element. This feature is conjoined with the tag assigned to the current and next sentence.
	Axiom, Theorem, Corollary Mention	(a) The current (or previous) discourse element is mentioned as an Axiom, Theorem, or Corollary (e.g., *Similar Triangle Theorem* or *Corollary 2.1*). (b) The section or subsection in the textbook containing the current (or previous) discourse element mentions an Axiom, Theorem, or Corollary. This feature is conjoined with the tag assigned to the current (and previous) discourse element.
	Equation	The current (or next) discourse element contains an equation (e.g., $PA \times PB = PT^2$). This feature is conjoined with the tag assigned to the current (and next) sentence.
	Associated diagram	The current discourse element contains a pointer to a figure (e.g., "Figure 2.1"). This feature is conjoined with the tag assigned to the current discourse element.
	Bold/ Underline	The discourse element (or previous discourse element) contains text that is in bold font or underlined. Conjoined with the tag assigned to the current (and previous) discourse element.
	Bounding box	Indicator that the current and previous discourse elements are bounded by a bounding box in the textbook. Conjoined with the tag assigned to the current (and previous) discourse element.
	JSON structure	Indicator that the current and previous discourse element are in the same node of the JSON hierarchy. Conjoined with the tag assigned to the current (and previous) discourse element.

Let $\mathbf{A}_b = \left(A_1^{(b)}, A_2^{(b)}, \ldots, A_{|\mathbf{A}_b|}^{(b)}\right)$ be the axiom mentions extracted from textbook b. Let $\mathbf{A}$ denote the collection of axiom mentions extracted from all textbooks. We assume a global ordering of axioms $\mathbf{A}^* = \left(A_1^*, A_2^*, \ldots, A_U^*\right)$ where U is some predefined upper

bound on the total number of axioms in geometry. Then, we emphasize that the axiom mentions extracted from each textbook (roughly) follow this ordering. Let $Z_{ij}^{(b)}$ be a random variable that denotes if axiom $A_i^{(b)}$ extracted from book b refers to the global axiom A_j^*. We introduce a log-linear model that factorizes over alignment pairs:

$$P(\mathbf{Z}|\mathbf{A};\boldsymbol{\phi}) = \frac{1}{Z(\mathbf{A};\boldsymbol{\phi})} \times \exp\left(\sum_{\substack{b_1,b_2\in\mathcal{B}\\ b_1\neq b_2}} \sum_{1\leq k\leq U} \sum_{\substack{1\leq i\leq|\mathbf{A}_{b_1}|\\ 1\leq j\leq|\mathbf{A}_{b_2}|}} Z_{ik}^{(b_1)} Z_{jk}^{(b_2)} \boldsymbol{\phi}^T \mathbf{g}(A_i^{(b_1)}, A_j^{(b_2)})\right)$$

Here, $Z(\mathbf{A};\boldsymbol{\phi})$ is the partition function of the log-linear model. $\mathbf{g}$ denotes a feature function that measures the similarity of two axiom mentions (described in detail later). We introduce the following constraints on the alignment structure:

C1: An axiom appears in a book at most once.

C2: An axiom refers to exactly one theorem in the global ordering.

C3: Ordering Constraint: If i^{th} axiom in a book refers to the j^{th} axiom in the global ordering then no axiom succeeding the i^{th} axiom can refer to a global axiom preceding j.

Learning with Hard Constraints: We find the optimal parameters $\boldsymbol{\phi}$ using maximum-likelihood estimation with L2 regularization:

$$\boldsymbol{\phi}^* = \arg\max_{\boldsymbol{\phi}} \log P(\mathbf{Z}|\mathbf{A};\boldsymbol{\phi}) - \mu||\boldsymbol{\phi}||_2^2$$

We use L-BFGS to optimize the objective. To compute feature expectations appearing in the gradient of the objective, we use a Gibbs sampler. The sampling equations for Z_{ik}^b are:

$$P(Z_{ik}^{(b)}|rest) \propto \exp\left(T_b(i,k)\right) \tag{1}$$

$$T_b(i,k) = Z_{ik}^{(b)} \sum_{\substack{b'\in\mathcal{B}\\ b'\neq b}} \sum_{1\leq j\leq|A_{b'}|} Z_{jk}^{(b')} \boldsymbol{\phi}^T \mathbf{g}(A_i^{(b)}, A_j^{(b')})$$

Note that the constraints $C1\ldots3$ define the feasible space of alignments. Our sampler always samples the next $Z_{ik}^{(b)}$ in this feasible space. μ is tuned on the development set.

Learning with Soft Constraints: We might want to treat some constraints, in particular, the ordering constraints $C3$ as soft constraints. We can write down the constraint $C3$ using the alignment variables:

$$Z_{ij}^{(b)} \leq 1 - Z_{kl}^{(b)}$$
$$\forall\, 1 \leq i < k \leq |\mathbf{A}_b|, 1 \leq l < j \leq U$$
$$\forall\, b \in \mathcal{B}$$

To model these constraints as soft constraints, we penalize the model for violating these constraints. Let the penalty for violating this constraints be the $\exp\left(\nu \max\left(0, 1 - Z_{ij}^{(b)} - Z_{kl}^{(b)}\right)\right)$. Thus, we introduce a new regularization term:

$$\mathbf{R}(\mathbf{Z}) = \sum_{\substack{1 \le i < k \le |\mathbf{A}_b| \\ 1 \le l < j \le U \\ b \in \mathcal{B}}} \exp\left(\nu \max\left(0, 1 - Z_{ij}^{(b)} - Z_{kl}^{(b)}\right)\right)$$

Here ν is a hyper-parameter to tune the cost of violating a constraint. We write down the following regularized objective:

$$\boldsymbol{\phi}^* = \arg\max_{\boldsymbol{\phi}} \log P(\mathbf{Z}|\mathbf{A}; \boldsymbol{\phi}) - \mathbf{R}(\mathbf{Z}) - \mu||\boldsymbol{\phi}||_2^2$$

We use L-BFGS to find the optimal parameters $\boldsymbol{\phi}*$. We perform Gibbs sampling to compute feature expectations. The sampling equation for $Z_{ik}^{(b)}$ is similar (Equation (1)), but:

$$\begin{aligned} T_b(i,k) = & \sum_{\substack{b' \in \mathcal{B} \\ b' \neq b}} \sum_{1 \le j \le |A_{b'}|} Z_{ik}^{(b)} Z_{jk}^{(b')} \boldsymbol{\phi}^T \mathbf{g}(A_i^{(b)}, A_j^{(b')}) \\ & + \nu \sum_{\substack{b' \in \mathcal{B} \\ b' \neq b}} \sum_{i < j \le |A_{b'}|} \sum_{1 \le l < k} \left(1 - Z_{ik}^{(b)} - Z_{jl}^{(b')}\right) \\ & + \nu \sum_{\substack{b' \in \mathcal{B} \\ b' \neq b}} \sum_{1 \le j < i|} \sum_{k < l \le U} \left(1 - Z_{ik}^{(b)} - Z_{jl}^{(b')}\right) \end{aligned}$$

Features: Now, we describe the features g. These too include content-based features encoding various notions of similarity between pairs of axiom mentions (such as unigram, bigram, dependency and entity overlap, longest common subsequence [LCS], alignment, MT, and summarization scores) as well as various typographical features, such as matching of the current (and parent) node of axiom mentions in respective JSON hierarchies, equation template matching, and image caption matching. The features are listed in Table 5. We will further discuss the impact of the various content- and typography-based features later in Section 11.

8.1.3 Joint Identification and Alignment. Joint modeling of axiom identification and alignment components is useful as both problems potentially help each other. Correct axiom identification can help predict correct alignments and axiom alignments can help predict correct axiom mention boundaries. Hence, we combine the respective models for identification and alignment into a joint model. Let $Y_{ij}^{(b)}$ denote that the discourse element $s_i^{(b)}$ from book b has tag j. We reuse the definitions of the alignment variables $Z_{ij}^{(b)}$ as before. We further define $Z_{i0}^{(b)}$ such that it denotes that the i^{th} axiom in textbook b

Table 5
Feature set for our axiom alignment model. The features are based on content and typography.

Content	Unigram, Bigram, Dependency and Entity Overlap	Real valued features that compute the proportion of common unigrams, bigrams, dependencies, and geometry entities (constants, predicates, and functions) across the two axioms. When comparing geometric entities, we include geometric entities derived from the associated diagrams when available.
	Longest Common Subsequence	Real valued feature that computes the length of longest common subsequence of words between two axiom mentions normalized by the total number of words in the two mentions.
	Number of discourse elements	Real valued feature that computes the absolute difference in the number of discourse elements in the two mentions.
	Alignment Scores	We use an off-the-shelf monolingual word aligner—*JACANA* (Yao et al. 2013) pretrained on PPDB—and compute alignment score between axiom mentions as the feature.
	MT Metrics	We use two common MT evaluation metrics *METEOR* (Denkowski and Lavie 2010) and *MAXSIM* (Chan and Ng 2008), and use the evaluation scores as features. While *METEOR* computes n-gram overlaps controlling on precision and recall, *MAXSIM* performs bipartite graph matching and maps each word in one axiom to at most one word in the other.
	Summarization Metrics	We also use *Rouge-S* (Lin 2004), a text summarization metric, and use the evaluation score as a feature. *Rouge-S* is based on skip-grams.
Discourse (Typography)	JSON structure	Indicator matching the current (and parent) node of axiom mentions in respective JSON hierarchies; i.e., are both nodes mentioned as axioms, diagrams or bounding boxes?
	Equation Template	Indicator feature that matches templates of equations detected in the axiom mentions. The template matcher is designed such that it identifies various rewritings of the same axiom equation, e.g., $PA \times PB = PT^2$ and $PA \times PB = PC^2$ could refer to the same axiom with point T in one axiom mention being point C in another mention.
	Image Caption	Proportion of common unigrams in the image captions of the diagrams associated with the axiom mentions. If both mentions do not have associated diagrams, this feature does not fire.

is not aligned with any global axiom. We again define a log-linear model with factors that score axiom identification and axiom alignments.

$$p(\mathbf{Y}, \mathbf{Z} | \{\mathbf{S}_b\}; \boldsymbol{\theta}, \boldsymbol{\phi}) \propto f_{AI}(\mathbf{Y} | \{\mathbf{S}_b\}; \boldsymbol{\theta}) \times f_{AA}(\mathbf{Z} | \mathbf{Y}, \{\mathbf{S}_b\}; \boldsymbol{\phi})$$

Here, the factors:

$$f_{AI} = \exp\Big(\sum_{b \in \mathcal{B}} \sum_{k=1}^{|\mathbf{S}_b|} \sum_{i,j \in \mathcal{T}} Y^{(b)}_{k-1i} Y^{(b)}_{kj} \boldsymbol{\theta}_{ij}^T \mathbf{f}_{ij}(i, j, \mathcal{S}_b)\Big)$$

$$f_{AA} = \exp\Big(\sum_{\substack{b_1, b_2 \in \mathcal{B} \\ b_1 \neq b_2}} \sum_{1 \leq k \leq U} \sum_{\substack{1 \leq i \leq |\mathbf{A}_{b_1}| \\ 1 \leq j \leq |\mathbf{A}_{b_2}|}} Z^{(b_1)}_{ik} Z^{(b_2)}_{jk} \boldsymbol{\phi}^T \mathbf{g}(A^{(b_1)}_i, A^{(b_2)}_j)\Big)$$

Note that an error in axiom identification would result in a change in the axiom alignment feature function **g** and hence would worsen the quality of axiom alignments. This motivates our joint modeling of axiom identification and alignment.

We again have the following model constraints:

C1′: Every discourse element has a unique label

C2′ Tag O cannot be followed by tag I

C3′ Consistency between *Y*s and *Z*s, i.e., axiom boundaries defined by *Y*s and *Z*s must agree.

C4′ = C3.

We use L-BFGS for learning. To compute feature expectations, we use a Metropolis Hastings sampler that samples **Y***s* and **Z***s* alternatively. Sampling for **Z***s* reduces to Gibbs sampling and the sampling equations are the same as before (Section 8.1.2). For better mixing, we sample **Y** in blocks. Consider blocks of **Y***s* which denote axiom boundaries at time stamp *t*; we define three operations to sample axiom blocks at the next time stamp. The operations (shown in Figure 7) are:

Update axiom: The axiom boundary can be shrunk, expanded, or moved. The new axiom, however, cannot overlap with other axioms.

Delete axiom: The axiom can be deleted by labeling all its discourse elements as *O*.

Introduce axiom: Given a contiguous sequence of discourse elements labeled *O*, a new axiom can be introduced.

Note that these three operations define an ergodic Markov chain. We use the axiom identification part of the model as the proposal:

$$Q(\tilde{\mathbf{Y}}|\mathbf{Y}) \propto \exp\left(\sum_{b\in\mathcal{B}}\sum_{k=1}^{|\mathbf{s}_b|}\sum_{i,j\in\mathcal{T}} \tilde{Y}^{(b)}_{k-1i}\tilde{Y}^{(b)}_{kj}\boldsymbol{\theta}^{T}_{ij}\mathbf{f}_{ij}(i,j,\mathcal{S}_b)\right)$$

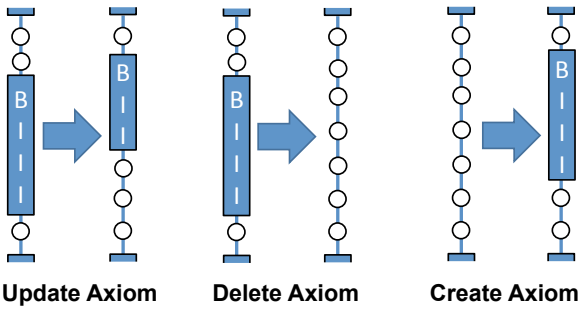


Figure 7
An illustration of the three operations to sample axiom blocks.

Hence, the acceptance ratio only depends on the alignment part of the model: $R(\bar{\mathbf{Y}}|\mathbf{Y}) = \min\left(1, \frac{U(\bar{\mathbf{Y}})}{U(\mathbf{Y})}\right)$ where $U(\mathbf{Y}) = f_{AA}$. We again have two variants, where we model the ordering constraints (C4′) as soft or hard constraints.

8.2 Axiom Parsing

After harvesting axioms, we build a parser for these axioms that maps raw axioms to horn-clause rules. The axiom harvesting step provides us a multiset of axiom extractions. Let $\mathcal{A} = \{\mathbf{A}_1, \mathbf{A}_2, \ldots, \mathbf{A}_{|\mathcal{A}|}\}$ represent the multiset where each axiom $\mathbf{A}_i$ is mentioned at least once. Each axiom mention, in turn, comprises a contiguous sequence of discourse elements and optionally an accompanying diagram.

Semantic parsers map natural language to formal programs such as database queries (Liang, Jordan, and Klein 2011, inter alia), commands to robots (Shimizu and Haas 2009, inter alia), or even general purpose programs (Yin and Neubig 2017). More specifically, Liu et al. (2016a) learn "If-Then" program statements and Quirk, Mooney, and Galley (2015) learn "If-This-Then-That" rules. In theory, these works can be used to parse axioms to horn-clause rules. However, semantic parsing is a hard task and would require a large amount of supervision. In our setting, we can only afford a modest amount of supervision. We mitigate this issue by using the redundant axiom mention extractions from multiple sources (textbooks) and combining the parses obtained from various textbooks to achieve a better final parse for each axiom.

First, we describe a base parser that parses axiom mentions to horn-clause rules. Then, we utilize the redundancy of axiom extractions from various sources (textbooks) to improve our parser.

8.2.1 Base Axiomatic Parser. Our base parser identifies the *premise* and *conclusion* portions of each axiom and then uses *GEOS*'s text parser to parse the two portions into a logical formula. Then, the two logical formulas are put together to form horn-clause rules.

Axiom mentions (for example, the Pythagorean theorem mention in Figure 1) are often accompanied by equations or diagrams. When the mention has an equation, we simply treat the equation as the *conclusion* and the rest of the mention as the *premise*. When the axiom has an associated diagram, we always include the diagram in the *premise*. We learn a model to predict the split of the axiom text into two parts, forming the *premise* and the *conclusion* spans. Then, the *GEOS* parser maps the *premise* and *conclusion* spans to *premise* and *conclusion* logical formulas, respectively.

Let Z_s represent the split that demarcates the *premise* and *conclusion* spans. We score the axiom split as a log-linear model: $p(Z_s|a; \mathbf{w}) \propto \exp\left(\mathbf{w}^T \mathbf{h}(a, Z_s)\right)$. Here, $\mathbf{h}$ are feature functions described later. We found that in most cases (>95%), the premise and conclusion are contiguous spans in the axiom mention where the left span corresponds to the *premise* and the right span corresponds to the *conclusion*. Hence, we search over the space of contiguous spans to infer Z_s. Joint search over the latent variables Z_s, Z_p, and Z_c is exponential. Hence, we use a greedy procedure, beam search, with a fixed beam size (10) for inference. That is, in each step, we only expand the ten most promising candidates so far given by the current score. We first infer Z_s to decide the split of the axiom and then infer Z_p and Z_c to obtain the parse of the premise and the conclusion, using the two-part approach described before. We use L-BGFGS for learning.

Features: We list the features $\mathbf{h}$ defined over candidate spans forming the text split in Table 6. The features are similar to those used in previous work on discourse analysis,

Table 6
Feature set for our axiom parsing model.

Content	Span Similarity	Proportion of (a) words, (b) geometry relations, and (c) relation-arguments shared by the two spans.
	Number of Relations	Number of geometry relations represented in the two spans. We use the Lexicon Map from GEOS to compute the number of expressed geometry relations.
	Span Lengths	The distribution of the two text spans is typically dependent on their lengths. We use the ratio of the length of the two spans as an additional feature.
	Relative Position	Relative position of the two lexical heads and the text split in the discourse element sentence. We use the difference between the lexical head position and the text split position as the feature.
	Discourse Markers	Discourse markers (connectives, cue-words, or cue-phrases, etc.) have been shown to give good indications on discourse structure (Marcu 2000). We build a list of discourse markers using the training set, considering the first and last tokens of each span, culled to top 100 by frequency. We use these 100 discourse markers as features. We repeat the same procedure by using part-of-speech (POS) instead of words and use them as features.
Discourse (Typography)	Punctuation	Punctuation at the segment border is another excellent cue for the segmentation. We include indicator features to show whether there is punctuation at the segment border.
	Text Organization	Indicator that the two text spans are part of the same (a) sentence, (b) paragraph.
	RST Parse	We use an off-the-shelf RST parser (Feng and Hirst 2014) and include an indicator feature that shows that the segmentation matches the parse segmentation. We also include the RST label as a feature.
	Soricut and Marcu Segmenter	Soricut and Marcu (2003) (section 3.1) presented a statistical model for deciding elementary discourse unit boundaries. We use the probability given by this model retrained on our training set as a feature. This feature uses both lexical and syntactic information.
	Head/ Common Ancestor/ Attachment Node	Head node is defined as the word with the highest occurrence as a lexical head in the lexicalized tree among all the words in the text span. The attachment node is the parent of the head node. We use features for the head words of the left and right spans, the common ancestor (if any), the attachment node, and the conjunction of the two head node words. We repeat these features with part-of-speech (POS) instead of words.
	Syntax	Distance to (a) root, and (b) common ancestor for the nodes spanning the respective spans. We use these distances and the difference in the distances as features.
	Dominance	*Dominance* (Soricut and Marcu 2003) is a key idea in discourse that looks at syntax trees and studies sub-trees for each span to infer a logical nesting order between the two. We use the dominance relationship as a feature. See Soricut and Marcu (2003) for details.
	JSON structure	Indicator that the two spans are in the same node in the JSON hierarchy. Conjoined with the indicator feature that shows that the two spans are part of the same paragraph.

in particular on the automatic detection of elementary discourse units (EDUs) in rhetorical structure theory (Mann and Thompson 1988) and discourse parsing (Marcu 2000; Soricut and Marcu 2003). These include ideas such as the use of a list of discourse markers, punctuation, and natural text and JSON organization as an indicator of

discourse boundaries. We also use an off-the-shelf discourse parser and an *EDU* segmenter from Soricut and Marcu (2003). Then we also used syntax-based cues, such as span lengths, head node attachment, distance to common ancestor/root, relative position of the two lexical heads and the text split; and *dominance*, which have been found to be useful in discourse parsing (Marcu 2000; Soricut and Marcu 2003). Finally, we also used some semantic features, such as the similarity of the two spans (in terms of common words, geometry relations and relation-arguments), and number of geometry relations in the respective span parses. We will discuss the impact of the various features later in Section 11. Given a beam of *premise* and *conclusion* splits, we use the *GEOS* parser to obtain *premise* and *conclusion* logical formulas for each split in the beam and obtain a beam of axiom parses for each axiom in each textbook.

8.2.2 Multisource Axiomatic Parser. Now, we describe a multisource parser that utilizes the redundancy of axiom extractions from various sources (textbooks). Given a beam of 10-best parses for each axiom from each source, we use a number of heuristics to determine the best parse for the axiom:

1. **Majority Voting:** For each axiom, pick the parse that occurs most frequently across beams.
2. **Average Score:** Pick the parse that has the highest average parse score (only counting top 5 parses for each source) for each axiom.
3. **Learn Source Confidence:** Learn a set of weights $\{\mu_1, \mu_2, \ldots, \mu_S\}$, one for each source, and then pick the parse that has the highest average weighted parse score for each axiom.
4. **Predicate Score:** Instead of selecting from one of the top parses across various sources, treat each axiom parse as a bag of premise predicates and a bag of conclusion predicates. Then, pick a subset of premise and conclusion predicates for the final parse, using average scoring with thresholding.

9. Experiments

Data Sets and Baselines: We use a collection of grade 6–10 Indian high school math textbooks by four publishers/authors (*NCERT*, *R S Aggarwal*, *R D Sharma*, and *M L Aggarwal*)—a total of $5 \times 4 = 20$ textbooks to validate our model. Millions of students in India study geometry from these books every year and these books are readily available online. We manually marked chapters relevant for geometry in these books and then parsed them using Adobe Acrobat's *pdf2xml* parser and AllenAI's *Science Parse* project.[4] Then, we annotated geometry axioms, alignments, and parses for grade 6, 7, and 8 textbooks by the four publishers/authors. We use grade 6, 7, and 8 textbook annotations for development, training, and testing, respectively. Grade 9 and 10 data are used as unlabeled data. Thus our method is semi-supervised. During training our axiom identification, alignment, and joint axiom identification and alignment models, the latent variables **Z** are fixed for the training set and are not sampled. For the remaining data, these variables are sampled using our Gibbs sampler. All the hyper-parameters in all

4 https://github.com/allenai/science-parse

the models are tuned on the development set using grid search. Then, these hyper-parameter values are fixed and the entire training + development set is used for training (along with the unlabeled data) and all the models are evaluated on the test set.

GEOS used 13 types of entities and 94 functions and predicates. We add some more entities, functions, and predicates to cover other more complex concepts in geometry not covered in *GEOS*. Thus, we obtain a final set of 19 entity types and 115 functions and predicates for our parsing model. We use Stanford CoreNLP (Manning et al. 2014) for feature generation. We use two data sets for evaluating our system: (a) practice and official SAT style geometry questions used in *GEOS*, and (b) an additional data set of geometry questions collected from the aforementioned textbooks. This data set consists of a total of 1,406 SAT style questions across grades 6–10, and is approximately 7.5 times the size of the data set used in *GEOS*. We split the data set into training (350 questions), development (150 questions), and test (906 questions), with equal proportion of grade 6–10 questions. We annotated the 500 training and development questions with ground-truth logical forms. We use the training set to train another version of *GEOS* with the expanded set of entity types, functions, and predicates. We call this system *GEOS++*, which will be used as a baseline for our method.

Results: We first evaluate the axiom identification, alignment, and parsing models individually.

For axiom identification, we compare the results of automatic identification with gold axiom identifications and compute the precision, recall, and F-measure on the test set. We use strict as well as relaxed comparison. In strict comparison mode the automatically identified mentions and gold mentions must match exactly to get credit, whereas in the relaxed comparison mode only a majority (>50%) of sentences in the automatically identified mentions and gold mentions must match to get credit. Table 7 shows the results of axiom identification, where we clearly see improvements in performance when we jointly model axiom identification and alignment. This is due to the fact that both components reinforce each other. We also observe that modeling the ordering constraints as soft constraints leads to better performance than modeling them as hard constraints. This is because the ordering of presentation of axioms is generally (yet not always) consistent across textbooks.

To evaluate axiom alignment, we first view it as a series of decisions, one for each pair of axiom mentions, and compute precision, recall, and F-score by comparing automatic decisions with gold decisions. Then, we also use a standard clustering metric, Normalized Mutual Information (NMI) (Strehl and Ghosh 2002) to measure the quality

Table 7
Test set Precision, Recall, and F-measure scores for axiom identification when performed alone and when performed jointly with axiom alignment. We show results for both strict as well as relaxed comparison modes. For the joint model, we show results when we model ordering constraints as hard or soft constraints.

	Strict Comp.			Relaxed Comp.		
	P	**R**	**F**	**P**	**R**	**F**
Identification	64.3	69.3	66.7	84.3	87.9	86.1
Joint-Hard	68.0	68.1	68.0	85.4	87.1	86.2
Joint-Soft	69.7	71.1	**70.4**	86.9	88.4	**87.6**

Table 8
Test set Precision, Recall, F-measure, and NMI scores for axiom alignment when performed alone and when performed jointly with axiom identification. For the joint model, we show results when we model ordering constraints as hard or soft constraints.

	P	R	F	NMI
Alignment	71.8	74.8	73.3	0.60
Joint-Hard	75.0	76.4	75.7	0.65
Joint-Soft	79.3	81.4	**80.3**	**0.69**

of axiom mention clustering. Table 8 shows the results on the test set when gold axiom identifications are used. We observe improvements in axiom alignment performance too when we jointly model axiom identification and alignment jointly both in terms of F-score as well as NMI. Modeling ordering constraints as soft constraints again leads to better performance than modeling them as hard constraints in terms of both metrics.

To evaluate axiom parsing, we compute precision, recall, and F-score in (a) deriving literals in axiom parses, as well as for (b) the final axiom parses on our test set. Table 9 shows the results of axiom parsing for *GEOS* (trained on the training set) as well as various versions of our best performing system (*GEOS++* with our axiomatic solver) with various heuristics for multisource parsing. The results show that our system (single source) performs better than *GEOS*, as it is trained with the expanded set of entity types, functions, and predicates. The results also show that the choice of heuristic is important for the multisource parser—though all the heuristics lead to improvements over the single source parser. The average score heuristic that chooses the parse with the highest average score across sources performs better than majority voting, which chooses the best parse based on a voting heuristic. Learning the confidence of every source and using a weighted average is an even better heuristic. Finally, predicate scoring, which chooses the parse by scoring predicates on the premise and conclusion sides, performs the best leading to 87.5 F1 score (when computed over parse literals) and 73.2 F1 score (when computed on the full parse). The high F1 score for axiom parsing on the test set shows that our approach works well and we can accurately harvest axiomatic knowledge from textbooks.

Table 9
Test set Precision, Recall, and F-measure scores for axiom parsing. These scores are computed over literals derived in axiom parses or full axiom parses. We show results for the old *GEOS* system; for the improved *GEOS++* system with expanded entity types, functions, and predicates; and for the multisource parsers presented in this paper.

		Literals			Full Parse		
		P	R	F	P	R	F
	GEOS	86.7	70.9	78.0	64.2	56.6	60.2
GEOS++	**Single Src.**	91.6	75.3	82.6	68.8	60.4	64.3
	Maj. Voting	90.2	78.5	83.9	70.0	63.3	66.5
	Avg. Score	90.8	79.6	84.9	71.7	66.4	69.0
	Src. Confid.	91.0	79.9	85.1	73.3	68.1	70.6
	Pred. Score	92.8	82.8	**87.5**	76.6	70.1	**73.2**

Table 10
Scores for solving geometry questions on the SAT practice and official data sets and a data set of questions from the 20 textbooks. We use SAT's grading scheme that rewards a correct answer with a score of 1.0 and penalizes a wrong answer with a negative score of 0.25. *Oracle* uses gold axioms but automatic text and diagram interpretation in our logical solver. All differences between *GEOS* and our system are significant ($p < 0.05$ using the two-tailed paired t-test).

	Practice	**Official**	**Textbook**
GEOS	61	49	32
Our System	**64**	**55**	**51**
Oracle	80	78	72

Finally, we use the extracted horn-clause rules in our axiomatic solver for solving geometry problems. For this, we over-generate a set of horn-clause rules by generating three horn-clause parses for each axiom and use them as the underlying theory in prolog programs such as the one shown in Figure 4. We use weighted logical expressions for the question description and the diagram derived from *GEOS++* as declarations, and the (normalized) score of the parsing model multiplied by the score of the joint axiom identification and alignment model as weights for the rules. Table 10 shows the results for our best end-to-end system and compares it to *GEOS* on the practice and official SAT data set from Seo et al. (2015) as well as questions from the 20 textbooks. On all the three data sets, our system outperforms *GEOS*. Especially on the data set from the 20 textbooks (which is indeed a harder data set and includes more problems that require complex reasoning based on geometry), *GEOS* does not perform very well, whereas our system still achieves a good score. *Oracle* shows the performance of our system when gold axioms (written down by an expert) are used along with automatic text and diagram interpretations in *GEOS++*. This shows that there is scope for further improvement in our approach.

10. Explainability

Students around the world solve geometry problems through rigorous deduction, whereas the numerical solver in *GEOS* does not provide such explainability. One of the key benefits of our axiomatic solver is that it provides an easy-to-understand student-friendly deductive solution to geometry problems.

To test the explainability of our axiomatic solver, we asked 50 grade 6–10 students (10 students in each grade) to use *GEOS* and our system (*GEOS++* with our axiomatic solver) as a Web-based assistive tool while learning geometry. The tool uses the probabilistic prolog solver (Fierens et al. 2015) to derive the most probable explanation (MPE) for a solution. Then, it lists, one by one, the various axioms used and the conclusion drawn from the axiom application, as shown in Figure 8. The students were each asked to rate how 'explainable' and 'useful' the two systems were on a scale of 1–5. Table 11 shows the mean rating by students in each grade on the two facets. We can observe that students of each grade found our system to be more interpretable as well as more useful to them than *GEOS*. This study lends support to our claims about the need for an interpretable deductive solver for geometry problems.

1. Sum of interior angles of triangle is 180°
=> ∠OAM+∠AMO +∠MOA = 180°
=> ∠MOA = 60°

2. Similar triangle theorem
=>△MOB ~ △MOA
=> ∠MOB = ∠MOA = 60°

3. ∠AOB = ∠ MOB +∠ MOA
=>∠ AOB = 120°

4. Angle subtended by a chord at the center is twice the angle subtended at the circumference
=>∠ ADB = 0.5 x ∠ ADB
= 60°

Figure 8
An example demonstration on how to solve the problem in Figure 1: (1) Use the theorem that the sum of interior angles of a triangle is 180° and additionally the fact that ∠AMO is 90° to conclude that ∠MOA is 60°. (2) Conclude that △MOA ∼ △MOB (using a similar triangle theorem) and then conclude that ∠MOB = ∠MOA = 60° (using the theorem that corresponding angles of similar triangles are equal). (3) Use angle sum rule to conclude that ∠AOB = ∠MOB + ∠MOA = 120°. (4) Use the theorem that the angle subtended by an arc of a circle at the center is double the angle subtended by it at any point on the circle to conclude that ∠ADB = 0.5 × ∠AOB = 60°.

11. Feature Ablation

In this section, we will measure the value of the various features in our axiom harvesting and parsing pipeline. Note that we have described three sets of features **f**, **g**, and **h**—corresponding to the various steps in our pipeline: axiom identification, axiom alignment, and axiom parsing in Tables 4, 5, and 6. We will ablate each of the three features one by one via **backward selection** (i.e., we will remove features and observe how that affects performance).

Table 11
User study ratings for *GEOS* and our system (O.S.) by students in grades 6–10. Ten students in each grade were asked to rate the two systems on a scale of 1–5 on two facets: 'explainability' and 'usefulness'. Each cell shows the mean rating computed over ten students in that grade for that facet.

	Explainability		Usefulness	
	GEOS	*O.S.*	*GEOS*	*O.S.*
Grade 6	2.7	**2.9**	2.9	**3.2**
Grade 7	3.0	**3.7**	3.3	**3.6**
Grade 8	2.7	**3.5**	3.1	**3.5**
Grade 9	2.4	**3.3**	3.0	**3.7**
Grade 10	2.8	**3.1**	3.2	**3.8**
Overall	2.7	**3.3**	3.1	**3.6**

Table 12
Ablation study results for the axiom identification component. We remove features of the axiom identification component one by one as listed in Table 4 and observe the fall in performance in terms of the axiom identification performance as well as the overall performance to gauge the value of the various features.

		Axiom Identification F1		SAT Scores		
		Strict Comp.	Relaxed Comp.	Practice	Official	Textbook
Content	Sentence Overlap	56.2	73.8	56	43	42
	Geometry entities	64.0	80.4	61	49	46
	Keywords	67.5	81.0	62	54	48
Discourse (Typography)	RST edge	66.6	78.9	58	46	44
	Axm, Thm, Corr.	62.6	77.8	57	47	43
	Equation	66.2	78.6	57	46	42
	Associated Diagram	68.5	84.4	61	52	49
	Bold / Underline	68.2	82.0	62	52	48
	Bounding box	59.7	75.5	55	47	40
	XML structure	67.4	80.6	60	51	46
	Unablated	70.4	87.6	64	55	51

11.1 Ablating Axiom Identification Features

Table 12 shows the fall in performance in terms of the axiom identification performance, as well as the overall performance as we ablate various axiom identification features listed in Table 4. We can observe that removal of any of the features results in a loss of performance. Thus, all the content as well as typographical features are important for performance. We observe that the content features such as sentence overlap, geometry entity sharing, and keyword usage are clearly important. At the same time, the various discourse features such as the RST relation, axiom, theorem, corollary annotation, use of equations and diagrams, bold/underline, bounding box, and XML structure are all important. Most of these features depend on typographical information that is vital in performance of the axiom identification component as well as the overall model. In particular, we can observe that the axiom, theorem, corollary annotation, and bounding box features contribute most to the performance of the model as they are direct indicators of the presence of an axiom mention.

11.2 Ablating Axiom Alignment Features

Table 13 shows the fall in performance in terms of the axiom alignment performance as well as the overall performance as we ablate various axiom alignment features listed in Table 5. We again observe that removal of any of the features results in a loss of performance. Thus, the various content as well as typographical features are important for performance. We observe that the content features such as unigram, bigram and entity overlap, length of the longest common subsequence, number of sentences and various aligner, MT, and summarization scores are clearly important. At the same time, the various discourse features such as the XML structure, equation template, and image

Table 13
Ablation study results for the axiom alignment component. We remove features of the axiom alignment component one by one as listed in Table 5 and observe the fall in performance in terms of the axiom alignment performance, as well as the overall performance to gauge the value of the various features.

				SAT Scores		
		F1	NMI	Practice	Official	Textbook
Content	Overlap	70.7	0.54	57	45	45
	LCS	78.7	0.64	61	53	49
	Number of Sentences	78.5	0.65	62	54	48
	Alignment Scores	72.6	0.57	59	49	48
	MT Metrics	74.8	0.60	62	52	49
	Summarization Metrics	75.9	0.63	62	54	50
Typography	XML Structure	71.5	0.57	58	47	46
	Equation Template	76.6	0.61	57	47	43
	Image Caption	77.9	0.65	62	53	47
	Unablated	80.3	0.69	64	55	51

caption match are all important. Note that these features depend on typographical information that is again vital in performance. In particular, we can observe that the overlap and the XML structure features contribute most to the performance of the model.

11.3 Ablating Axiom Parsing Features

Table 14 shows the fall in performance in terms of the axiom parsing performance as well as the overall performance as we ablate various axiom parsing features listed in Table 6. We again observe that removal of any of the features results in a loss of performance. The axiom parsing component uses a few content-based features, such as span similarity and number of relations, span lengths, and relative position; and various discourse features, such as discourse markers, punctuations, text organization, RST parse, an existing discourse segmentor from Soricut and Marcu (Soricut and Marcu 2003), node attachment, syntax, dominance, and XML structure; and all are clearly important. In particular, we can observe that span similarity and punctuation features contribute most to the performance of the model.

12. Axioms Harvested

We qualitatively analyze the structured axioms harvested by our method. We show the few most probable horn-clause rules for some popular named theorems in geometry in Figure 9, along with the confidence of our method on the rules being correct. Note that some horn-clause parsed rules can be incorrect. For example, the second most probable horn-clause rule for the Pythagorean theorem is partially incorrect (does not state which angle is 90°). Similarly, the second and third most probable horn-clause for the circle secant tangent theorem are also incorrect. Our problog solver can use these redundant but weighted horn-clause rules for solving geometry problems.

Table 14
Ablation study results for the axiom parsing component. We remove features of the axiom parsing component one by one as listed in Table 6 and observe the fall in performance in terms of the axiom parsing performance as well as the overall performance to gauge the value of the various features.

	F1		SAT Scores		
	Literals	Full Parse	Practice	Official	Textbook
Span Similarity	71.8	64.6	51	40	42
No. of Relations	82.3	70.5	60	51	49
Span Lengths	86.0	72.0	63	54	50
Relative Position	83.9	69.2	60	52	47
Discourse Markers	77.4	68.4	55	48	47
Punctuations	73.5	65.0	52	45	45
Text Organization	74.4	66.2	52	47	46
RST Parse	84.6	70.8	62	52	49
Soricut & Marcu	83.2	69.8	61	52	50
Head Node, etc.	85.3	71.6	62	54	49
Syntax	75.5	66.6	54	47	46
Dominance	73.9	66.1	53	47	44
XML Structure	77.6	68.0	59	51	46
Unablated	87.5	73.2	64	55	51

Pythagorous Theorem:
0.84 isTriangle(ABC) ^ angleMeasure(ABC, 90) ➔ length(AB)^2 + length(BC)^2 = length(CA)^2
0.53 isTriangle(PQR) ^ isRightTriangle(PQR) ^ angleMeasure(PQR, 90) ➔ length(PQ)^2 + length(QR)^2 = length(RP)^2

Sum of angles of a Triangle
0.94 isTriangle(ABC) ➔ angleMeasure(ABC) + angleMeasure(BCA) + angleMeasure(CAB) = 180

Circle Secant Tangent Theorem
0.77 isCircle(O) ^ isTangentAt(PT, O, T) ^ isSecantAt(PA, O, A) ^ isSecantAt(PB, O, B) ^ liesOn(A, PB) ➔ length(PA) * length(PB) = length(PT)^2
0.46 isCircle(O) ^ isTangent(PT, O) ^ isSecant(PA, O) ^ isSecant(PB, O) ➔ length(PA) * length(PB) = length(PT)^2
0.41 isCircle(O) ^ isTangentAt(PT, O, T) ^ isSecantAt(PA, O, A) ^ isSecantAt(PB, O, B) ➔ length(PA) * length(PB) = length(PT)^2

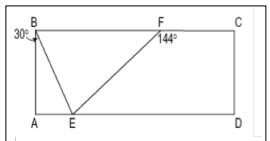
Congruent Triangles
0.82 isTriangle(ABC) ^ isTriangle(DEF) ^ length(AB) = length(DE) ^ length(BC) = length(EF) ^ length(CA) = length(FD) ➔ congruentTriangles(ABC, DEF)
0.73 isTriangle(ABC) ^ isTriangle(DEF) ^ length(AB) = length(DE) ^ angleMeasure(ABC) = angleMeasure(DEF) ^ length(BC) = length(EF) ➔ congruentTriangles(ABC, DEF)
0.76 isTriangle(ABC) ^ isTriangle(DEF) ^ angleMeasure(ABC) = length(DEF) ^ length(BC) = length(EF) ^ angleMeasure(BCA) = angleMeasure(EFD) ➔ congruentTriangles(ABC, DEF)
0.78 isTriangle(ABC) ^ isTriangle(DEF) ^ angleMeasure(ABC, 90) ^ angleMeasure(DEF, 90) ^ length(AB) = length(DE) ^ length(CA) = length(FD) ➔ congruentTriangles(ABC, DEF)

Figure 9
Horn-clause rules for some popular named theorems in geometry harvested by our approach. We also show the confidence our method has on the rule being correct (which is used in reasoning via the problog solver).

13. Example Solutions and Error Analysis

Next, we qualitatively describe some example solutions of geometry problems as well as perform a qualitative error analysis. We first show some sample questions that our solver can answer correctly in Table 15. We also show the explanations generated by our

Table 15
Some correctly answered questions along with explanations generated by our deductive solver for these problems.



In the accompanying diagram of rectangle ABCD, <ABE = 30 and <CFE = 144. Find <BEF.

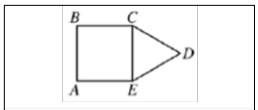
1. Each angle of a rectangle is 90^0
=> $\angle BAE = 90^0$

2. Each angle of a rectangle is 90^0
=> $\angle CDE = 90^0$

3. Sum of internal angles of a triangle is 180^0
=> $\angle BAE + \angle AEB + \angle EBA = 180^0$
=> $90^0 + \angle AEB + 30^0 = 180^0$
=> $\angle AEB = 60^0$

4. Sum of internal angles of a quadrilateral is 360^0
=> $CDE + \angle DEF + \angle EFC + \angle FCD = 180^0$
=> $90^0 + \angle DEF + 144^0 + 90^0 = 360^0$
⇒ $\angle DEF = 36^0$

5. Angles on one side of a straight line is 180^0
=> $\angle AEB + \angle BEF + \angle DEF = 180^0$
=> $60^0 + \angle BEF + 36^0 = 180^0$
=> $\angle BEF = 84^0$

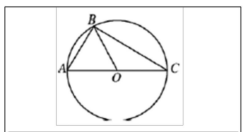


In the figure above, CDE is an equilateral triangle and ABCE is a square with an area of 1. What is the perimeter of polygon ABCDE?

1. Side of a square is square root of the area.
=> AB = BC = CE = EA = 1

2. All sides of an equilateral triangle are equal
=> CD = DE = EA = 1

3. Perimeter of a polygon is a the sum of its sides
=> perimeter(ABCDE) = AB + BC + CD + DE + EA
=> perimeter(ABCDE) = 1 + 1 + 1 + 1 + 1
= 5

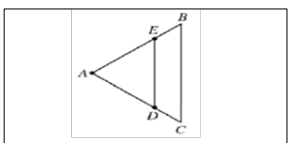


In the figure above, triangle ABC is inscribed in the circle with center O and diameter AC. If AB = AO, what is the degree measure of angle ABO?

1. Circle Definition: The length of all line segments joining the center of a circle with any point that lies on the circle are equal to the radius of the circle.
=> AO = BO

2. A triangle with all equal sides is equilateral
=> △ABO is equilateral.

3. Every internal angle of an equilateral triangle is 60^0
=>$\angle ABO = 60^0$

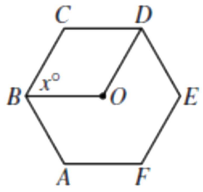
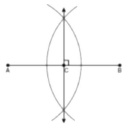
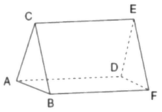
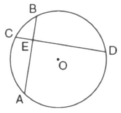
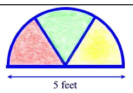


In triangle ABC above, AB = AC, E is the midpoint of line AB, and D is the midpoint of line AC. If AE = x and ED = 4, what is length BC?

1. Similar Triangles
Note that $\frac{AD}{AC} = \frac{AE}{AB}$ and EAD = BAC
=> EAD ~ BAC (SAS similarity)

2. Ratio of the lengths of corresponding sides of similar triangles are equal
=> $\frac{DE}{CB} = \frac{AD}{AC}$
=> $\frac{DE}{CB} = \frac{1}{2}$
=> $\frac{4}{CB} = \frac{1}{2}$
=> CB = 8
=> BC = 8

Table 16
Some example failure cases of our approach for solving SAT style geometry problems. In (i) the axiom set contains an axiom that the internal angle of a regular hexagon is 120° and that each side of a regular polygon is equal. But there is no way to deduce that the angle CBO is half of the internal angle ABC (by symmetry). On the other hand, the coordinate geometry solver can exploit these three facts as maximizing the satisfiability of the various constraints to answer the question. (ii) The solver does not contain any knowledge about construction. The question cannot be correctly interpreted and the coordinate geometry solver also gets it wrong. (iii) The solver does not contain any knowledge about construction or prisms. The question cannot be correctly interpreted and the coordinate geometry solver also gets it wrong. (iv) The question as well as the answer candidates cannot be correctly interpreted (as the concept of perpendicular to plane is not in the vocabulary). Both solvers get it wrong. (v) The parser cannot interpret that angle AC is indeed angle AEC. This needs to be understood by context as it defies the standard type definition of an angle. Both solvers get it wrong. (vi) Both diagram and text parsers fail here. Both solvers answer incorrectly.

(i)	 In the figure above, ABCDEF is a regular hexagon, and its center is point O. What is the value of x?	 The diagram at the right shows the construction of the perpendicular bisector of AB. Which statement is not true? AC = CB AC = 2AB CB = 1/2 AB AC + CB = AB	(ii)
(iii)	 The figure in the diagram at the right is a triangular prism. Which statement must be true? line DE = line AB line AD = line CE line AD = line BC line De = line BC	 Lines k_1 and k_2 intersect at point E. Line m is perpendicular to lines k_1 and k_2 at point E. Which statement is always true? Lines k1 and k2 are perpendicular. Line m is parallel to the plane determined by lines k_1 and k_2. Line m is perpendicular to the plane determined by lines k_1 and k_2. Line m is coplanar with lines k_1 and k_2.	(iv)
(v)	 In the accompanying diagram of circle O, chords AB and CD intersect at E and angle AC : CB : BD : DA = 4 : 2 : 6 : 8. What is the angle DEB? 36 degrees 90 degrees 100 degrees 126 degrees	 A cathedral window is built in the shape of a semicircle. If the window is to contain three stained glass sections of equal size, what is the area of each stained glass section? Express answer to the nearest square foot. 1 sq. ft. 3 sq. ft. 13 sq. ft. 26 sq. ft.	(vi)

deductive solver for these problems (constructed in the same way as described earlier). Note that these problems are diverse in terms of question types, as well as the reasoning required to answer them, and our solver can handle them.

We also show some failure cases of our approach in Table 16. There are a number of reasons that could lead to a failure of our approach to correctly answer a question. These include an error in parsing the diagram, the text, or an incorrect or incomplete knowledge in the form of geometry rules. As can be observed in the failure examples, and also evaluated by us in a small error analysis of 100 textbook questions, our approach answered 52 questions correctly. Among the 48 incorrectly answered questions, our diagram parse was incorrect for 12 questions, and the text parse was incorrect for 15 questions. Our formal language was insufficiently defined to handle 6 questions (i.e., the semantics of the question could not be adequately captured by the formal language). Twenty-one questions were incorrectly answered due to missing knowledge of geometry in the form of rules. Note that several questions were incorrectly answered due to a failure of multiple system components (for example, failure of both the text and the diagram parser).

14. Conclusion

We presented an approach to harvest structured axiomatic knowledge from math textbooks. Our approach uses rich features based on context and typography, the redundancy of axiomatic knowledge, and shared ordering constraints across multiple textbooks to accurately extract and parse axiomatic knowledge to horn-clause rules. We used the parsed axiomatic knowledge to improve the best previously published automatic approach to solve geometry problems. A user-study conducted on a number of school students studying geometry found our approach to be more interpretable and useful than its predecessor. While this article focused on harvesting geometry axioms from textbooks as a case study, we would like to extend it to obtain valuable structured knowledge from textbooks in areas such as science, engineering, and finance.

References

Alvin, Chris, Sumit Gulwani, Rupak Majumdar, and Supratik Mukhopadhyay. 2014. Synthesis of geometry proof problems. In *Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence*, pages 245–252, Quebec.

André, Elisabeth. 2000. The generation of multimedia presentations. *Handbook of Natural Language Processing*. pages 305–327, Marcel Dekker Inc.

André, Elisabeth, Wolfgang Finkler, Winfried Graf, Thomas Rist, Anne Schauder, and Wolfgang Wahlster. 1991. WIP: The automatic synthesis of multimodal presentations. Technical report, University of Saarland.

Arens, Yigal. 1992. Multimedia presentation planning as an extension of text planning. In Dale, R., E. Hovy, D. Rösner, and O. Stock, editors. *Aspects of Automated Natural Language Generation*, pages 277–280, Springer.

Arens, Yigal and Eduard Hovy. 1990. How to describe what? Towards a theory of modality utilization. In *Proceedings of the 12th Annual Conference of the Cognitive Science Society*, volume 487, Cambridge, MA.

Arens, Yigal, Eduard Hovy, and Susanne Van Mulken. 1993. Structure and rules in automated multimedia presentation planning. In *IJCAI*, pages 1253–1259, Chambery.

Arens, Yigal, Eduard H. Hovy, and Mira Vossers. 1992. On the knowledge underlying multimedia presentations. Technical report, University of Southern California Marina Del Rey Information Sciences Inst.

Arens, Yigal, Lawrence Miller, Stuart C. Shapiro, and Norman K. Sondheimer. 1988. Automatic construction of user-interface displays. In *AAAI*, pages 808–813, St. Paul.

Balasubramanian, Niranjan, Stephen Soderland, Oren Etzioni Mausam, and Robert Bart. 2002. Out of the box information extraction: A case study using bio-medical texts. Technical report, University of Washington.

Banko, Michele, Michael J. Cafarella, Stephen Soderland, Matthew Broadhead, and Oren Etzioni. 2007. Open information extraction from the web. In *IJCAI 2007, Proceedings of the 20th International Joint Conference on Artificial Intelligence*, pages 2670–2676, Hyderabad.

Bateman, John, Jörg Kleinz, Thomas Kamps, and Klaus Reichenberger. 2001a. Towards constructive text, diagram, and layout generation for information presentation. *Computational Linguistics*, 27(3):409–449.

Bateman, John, Jörg Kleinz, Thomas Kamps, and Klaus Reichenberger. 2001b. Towards constructive text, diagram, and layout generation for information presentation. *Computational Linguistics*, 27(3):409–449.

Berant, Jonathan, Andrew Chou, Roy Frostig, and Percy Liang. 2013. Semantic parsing on freebase from question-answer pairs. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing, EMNLP 2013*, pages 1533–1544, Seattle, WA.

Boguraev, Branimir K. and Mary S. Neff. 2000. Discourse segmentation in aid of document summarization. In *System Sciences, 2000. Proceedings of the 33rd Annual International Conference*, pages 1–10, Washington, DC.

Cafarella, Michael J., Doug Downey, Stephen Soderland, and Oren Etzioni. 2005. Knowitnow: Fast, scalable information extraction from the web. In *HLT/EMNLP 2005, Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing, Proceedings of the Conference*, pages 563–570, Vancouver.

Canisius, Sander and Caroline Sporleder. 2007. Bootstrapping information extraction from field books. In *EMNLP-CoNLL*, pages 827–836, Prague.

Chai, Joyce Y. and Rong Jin. 2004. Discourse structure for context question answering. In *Proceedings of the Workshop on Pragmatics of Question Answering at HLT-NAACL 2004*, pages 23–30, Boston, MA.

Chan, Yee Seng and Hwee Tou Ng. 2008. Maxsim: A maximum similarity metric for machine translation evaluation. In *2008 Annual Conference of the Association for Computational Linguistics (ACL)*, pages 55–62, Columbus, OH.

Chang, Chia Hui, Chun-Nan Hsu, and Shao-Cheng Lui. 2003. Automatic information extraction from semi-structured web pages by pattern discovery. *Decision Support Systems*, 35(1):129–147.

Chang, Chia Hui, Mohammed Kayed, Moheb R. Girgis, and Khaled F. Shaalan. 2006. A survey of web information extraction systems. *IEEE Transactions on Knowledge and Data Engineering*, 18(10):1411–1428.

Chen, David L. and Raymond J. Mooney. 2011. Learning to interpret natural language navigation instructions from observations. In *Proceedings of the 25th AAAI Conference on Artificial Intelligence (AAAI-2011)*, pages 859–865. San Francisco, CA.

Chou, Shang Ching, Xiao-Shan Gao, and Jing-Zhong Zhang. 1994. *Machine Proofs in Geometry: Automated Production of Readable Proofs for Geometry Theorems*, volume 6. World Scientific.

Clark, Peter, Phil Harrison, Niranjan Balasubramanian, and Oren Etzioni. 2012. Constructing a textual KB from a biology textbook. In *Proceedings of the Joint Workshop on Automatic Knowledge Base Construction and Web-scale Knowledge Extraction*, pages 74–78, Montreal.

Cohen, Robin. 1987. Analyzing the structure of argumentative discourse. *Computational Linguistics*, 13(1–2):11–24.

Conrath, Juliette, Stergos Afantenos, Nicholas Asher, and Philippe Muller. 2014. Unsupervised extraction of semantic relations using discourse cues. In *Association for Computational Linguistics (ACL)*, pages 2184–2194, Dublin.

Dale, Robert. 1991a. Exploring the role of punctuation in the signalling of discourse structure. In *Proceedings of a Workshop on Text Representation and Domain Modelling: Ideas from Linguistics and AI*, pages 110–120, Berlin.

Dale, Robert. 1991b. The role of punctuation in discourse structure. In *Working Notes for the AAAI Fall Symposium on Discourse Structure in Natural Language Understanding and Generation*, pages 13–14, Asilomar, CA.

Dalvi, Bhavana, Sumithra Bhakthavatsalam, Chris Clark, Peter Clark, Oren Etzioni, Anthony Fader, and Dirk Groeneveld. 2016. IKE—an interactive tool for knowledge extraction. In *Proceedings of the 5th Workshop on Automated Knowledge Base*

Construction, AKBC@NAACL-HLT 2016, pages 12–17, San Diego, CA.
Davis, Tom. 2006. *Geometry with computers*. Technical report.
Denkowski, Michael and Alon Lavie. 2010. Extending the meteor machine translation evaluation metric to the phrase level. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 250–253, Los Angeles, CA.
Dijk, Teun A. Van. 1979. Recalling and summarizing complex discourse. *Text Processing*, pages 49–93.
Duverle, David A., and Helmut Prendinger. 2009. A novel discourse parser based on support vector machine classification. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP: Volume 2-Volume 2*, pages 665–673, Suntec.
Dwyer, F. M. 1978. Strategies for improving visual learning. *Learning Services*.
Etzioni, Oren, Michele Banko, Stephen Soderland, and Daniel S. Weld. 2008. Open information extraction from the web. *Communications of the ACM*, 51(12):68–74.
Etzioni, Oren, Michael J. Cafarella, Doug Downey, Ana-Maria Popescu, Tal Shaked, Stephen Soderland, Daniel S. Weld, and Alexander Yates. 2004. Methods for domain-independent information extraction from the web: An experimental comparison. In *Proceedings of the Nineteenth National Conference on Artificial Intelligence, Sixteenth Conference on Innovative Applications of Artificial Intelligence*, pages 391–398, San Jose, CA.
Feigenbaum, Edward A. and Julian Feldman. 1963. *Computers and Thought*. The AAAI Press.
Feiner, Steven. 1988. An architecture for knowledge-based graphical interfaces. *ACM SIGCHI Bulletin*, 20(1):76.
Feiner, Steven K. and Kathleen R. McKeown. 1991. Automating the generation of coordinated multimedia explanations. *Computer*, 24(10):33–41.
Feng, Vanessa Wei and Graeme Hirst. 2012. Text-level discourse parsing with rich linguistic features. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Long Papers-Volume 1*, pages 60–68, Jeju.
Feng, Vanessa Wei and Graeme Hirst. 2014. A linear-time bottom–up discourse parser with constraints and post-editing. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 511–521, Baltimore, MD.
Fierens, Daan, Guy Van den Broeck, Joris Renkens, Dimitar Shterionov, Bernd Gutmann, Ingo Thon, Gerda Janssens, and Luc De Raedt. 2015. Inference and learning in probabilistic logic programs using weighted Boolean formulas. *Theory and Practice of Logic Programming*, 15(3):358–401.
Fleming, M. L., W. H. Levie, and W. H. Levie. 1978. *Instructional Message Design: Principles from the Behavioral Sciences*. Educational Technology Publications.
Gao, Xiao-Shan and Qiang Lin. 2002. MMP/geometer — A software package for automated geometric reasoning. In *International Workshop on Automated Deduction in Geometry*, pages 44–66, Hagenberg Castle.
Gerani, Shima, Yashar Mehdad, Giuseppe Carenini, Raymond T. Ng, and Bita Nejat. 2014. Abstractive summarization of product reviews using discourse structure. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1602–1613, Doha.
Ghosh, Sucheta, Giuseppe Riccardi, and Richard Johansson. 2012. Global features for shallow discourse parsing. In *Proceedings of the 13th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 150–159, Seoul.
Grosz, Barbara J. and Candace L. Sidner. 1986. Attention, intentions, and the structure of discourse. *Computational Linguistics*, 12(3):175–204.
Gulwani, Sumit, Vijay Anand Korthikanti, and Ashish Tiwari. 2011. Synthesizing geometry constructions. In *ACM SIGPLAN Notices*, 46, pages 50–61.
Hartley, J. 1985. *Designing Instructional Text*. Kogan Page.
Hovy, Eduard H. 1998. Automatic generation of formatted text. *Readings in Intelligent User Interfaces*. page 262, Morgan Kaufmann Publishers Inc.
Itzhaky, Shachar, Sumit Gulwani, Neil Immerman, and Mooly Sagiv. 2013. Solving geometry problems using a combination of symbolic and numerical reasoning. In *International Conference on Logic for Programming Artificial Intelligence and Reasoning*, pages 457–472, Stellenbosch.
Jansen, Peter, Mihai Surdeanu, and Peter Clark. 2014. Discourse complements lexical semantics for non-factoid answer reranking. In *Proceedings of the 52nd Annual*

Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 977–986, Baltimore, MD.

Ji, Yangfeng and Jacob Eisenstein. 2014. Representation learning for text-level discourse parsing. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 13–24, Baltimore, MD.

Kamp, Hans and Uwe Reyle. 1993. From discourse to logic: Introduction to model theoretic semantics of natural language, formal logic and discourse representation. *Studies in Linguistics and Philosophy*.

Kapur, Deepak. 1986. Using gröbner bases to reason about geometry problems. *Journal of Symbolic Computation*, 2(4):399–408.

Kate, Rohit J., Yuk Wah, Wong Raymond, and J. Mooney. 2005. Learning to transform natural to formal languages. In *Proceedings of AAAI-05*, pages 1062–1068, Pittsburgh, PA.

Kembhavi, Aniruddha, Mike Salvato, Eric Kolve, Minjoon Seo, Hannaneh Hajishirzi, and Ali Farhadi. 2016. A diagram is worth a dozen images. In *European Conference on Computer Vision*, pages 235–251, Amsterdam.

Kembhavi, Aniruddha, Minjoon Seo, Dustin Schwenk, Jonghyun Choi, Ali Farhadi, and Hannaneh Hajishirzi. 2017. Are you smarter than a sixth grader? Textbook question answering for multimodal machine comprehension. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5376–5384, Honolulu, HI.

Kitani, Tsuyoshi, Yoshio Eriguchi, and Masami Hara. 1994. Pattern matching and discourse processing in information extraction from Japanese text. *Journal of Artificial Intelligence Research*, 2:89–110.

Lafferty, John, Andrew McCallum, and Fernando Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the Eighteenth International Conference on Machine Learning, ICML*, volume 1, pages 282–289, Williamstown, MA.

Larkin, Jill H. and Herbert A. Simon. 1987. Why a diagram is (sometimes) worth ten thousand words. *Cognitive Science*, 11(1):65–100.

Lascarides, Alex and Nicholas Asher. 2008. Segmented discourse representation theory: Dynamic semantics with discourse structure. In H. Bunt and R. Muskens, editors, *Computing Meaning*. Springer, pages 87–124.

Lei, Tao, Fan Long, Regina Barzilay, and Martin C. Rinard. 2013. From natural language specifications to program input parsers. In *Association for Computational Linguistics (ACL)*, pages 1294–1303, Sofia.

Li, Sujian, Liang Wang, Ziqiang Cao, and Wenjie Li. 2014. Text-level discourse dependency parsing. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 25–35, Baltimore, MD.

Liang, Chen, Zhaohui Wu, Wenyi Huang, and C. Lee Giles. 2015. Measuring prerequisite relations among concepts. In *EMNLP*, pages 1668–1674, Lisbon.

Liang, Percy, Michael I. Jordan, and Dan Klein. 2011. Learning dependency-based compositional semantics. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1*, pages 590–599, Portland.

Lin, Chin-Yew. 2004. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out: Proceedings of the ACL-04 Workshop*, volume 8, pages 74–81, Barcelona.

Lin, Ziheng, Hwee Tou Ng, and Min-Yen Kan. 2014. A PDTB-styled end-to-end discourse parser. *Natural Language Engineering*, 20(2):151–184.

Ling, Wang, Edward Grefenstette, Karl Moritz Hermann, Tomáš Kočiskỳ, Andrew Senior, Fumin Wang, and Phil Blunsom. 2016. Latent predictor networks for code generation. *arXiv preprint arXiv:1603.06744*.

Ling, Wang, Dani Yogatama, Chris Dyer, and Phil Blunsom. 2017. Program induction for rationale generation: Learning to solve and explain algebraic word problems. In *Association for Computational Linguistics (ACL)*, pages 158–167, Vancouver.

Lioma, Christina, Birger Larsen, and Wei Lu. 2012. Rhetorical relations for information retrieval. In *Proceedings of the 35th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 931–940, Portland.

Liu, Chang, Xinyun Chen, Eui Chul Shin, Mingcheng Chen, and Dawn Song. 2016a, Latent attention for if-then program synthesis. In D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett, editors, *Advances in Neural Information Processing Systems 29*. Curran Associates, Inc., pages 4574–4582.

Liu, Hanxiao, Wanli Ma, Yiming Yang, and Jaime Carbonell. 2016b. Learning concept graphs from online educational data. *Journal of Artificial Intelligence Research*, 55:1059–1090.

Longacre, Robert E. 1983. *Some Aspects of Text Grammars*. Springer.

Louis, Annie, Aravind Joshi, and Ani Nenkova. 2010. Discourse indicators for content selection in summarization. In *Proceedings of the 11th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 147–156, Uppsala.

Luc, Christophe, Mustapha Mojahid, and Jacques Virbel. 1999. A linguistic approach to some parameters of layout: A study of enumerations. In *Understanding or Retrieval of Documents, AAAI Fall Symposium.*, pages 35–44, Orlando.

Mackinlay, Jock. 1986. Automating the design of graphical presentations of relational information. *ACM Transactions On Graphics (Tog)*, 5(2):110–141.

Mann, William C. and Sandra A. Thompson. 1988. Rhetorical Structure Theory: Toward a functional theory of text organization. *Text*, 3(8):234–281.

Manning, Christopher D., Mihai Surdeanu, John Bauer, Jenny Finkel, Steven J. Bethard, and David McClosky. 2014. The Stanford CoreNLP natural language processing toolkit. In *Association for Computational Linguistics (ACL) System Demonstrations*, pages 55–60, Baltimore, MD.

Marcu, Daniel. 2000. *The Theory and Practice of Discourse Parsing and Summarization*. MIT Press.

Matuszek, Cynthia, Dieter Fox, and Karl Koscher. 2010. Following directions using statistical machine translation. In *2010 5th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, pages 251–258, Osaka.

Maybury, M. 1998. *Planning Multimedia Explanations Using Communicative Acts*. San Francisco: Morgan Kaufman.

Mayer, Richard E. 1989. Systematic thinking fostered by illustrations in scientific text. *Journal of Educational Psychology*, 81(2):240.

Mitchell, T., W. Cohen, E. Hruschka, P. Talukdar, J. Betteridge, A. Carlson, B. Dalvi, M. Gardner, B. Kisiel, J. Krishnamurthy, N. Lao, K. Mazaitis, T. Mohamed, N. Nakashole, E. Platanios, A. Ritter, M. Samadi, B. Settles, R. Wang, D. Wijaya, A. Gupta, X. Chen, A. Saparov, M. Greaves, and J. Welling. 2015. Never-ending learning. In *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence (AAAI-15)*, pages 2302–2310, Austin, TX.

Moser, Megan and Johanna D. Moore. 1996. Toward a synthesis of two accounts of discourse structure. *Computational Linguistics*, 22(3):409–419.

Narasimhan, Karthik and Regina Barzilay. 2015. Machine comprehension with discourse relations. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing of the Asian Federation of Natural Language Processing*, Volume 1: Long papers, pages 1253–1262, Beijing.

Neal, J. G., S. C. Shapiro, C. Y. Thielman, J. R. Gucwa, and J. M. Lammens. 1990. Intelligent multi-media integrated interface project. Technical report RADC-TR-90-128, Calspan UB Research Center, Buffalo, NY.

Pascual, Elsa. 1996. Integrating text formatting and text generation. In *Trends in Natural Language Generation An Artificial Intelligence Perspective*, Springer, pages 205–221.

Pascual, Elsa and Jacques Virbel. 1996. Semantic and layout properties of text punctuation. In *Proceedings of the Association for Computational Linguistics Workshop on Punctuation*, pages 41–48, Santa Cruz, CA.

Peng, Fuchun and Andrew McCallum. 2006. Information extraction from research papers using conditional random fields. *Information Processing & Management*, 42(4):963–979.

Petre, M. and T. R. G. Green. 1990. Is graphical notation really superior to text, or just different? Some claims by logic designers about graphics in notation. In *Proceedings of ECCE-5*, Urbino.

Polanyi, Livia. 1988. A formal model of the structure of discourse. *Journal of Pragmatics*, 12(5–6):601–638.

Quirk, Chris, Raymond J. Mooney, and Michel Galley. 2015. Language to code: Learning semantic parsers for if-this-then-that recipes. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing of the Asian Federation of Natural Language Processing, Volume 1: Long Papers*, pages 878–888, Beijing.

Reed, Chris and Derek Long. 1997. Generating punctuation in written arguments. Technical report 2694743, Department of Computer Science, University College, London.

Sachan, Mrinmaya, Avinava Dubey, Eric P. Xing, and Matthew Richardson. 2015. Learning answer-entailing structures for machine comprehension. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, pages 239–249, Beijing.

Saleem, Ozair and Seemab Latif. 2012. Information extraction from research papers by data integration and data validation from multiple header extraction sources. In *Proceedings of the World Congress on Engineering and Computer Science*, volume 1, pages 177–180, San Francisco, CA.

Schattschneider, Doris and James King. 1997. *Geometry Turned On: Dynamic Software in Learning, Teaching, and Research*. Mathematical Association of America Notes.

Seo, Min Joon, Hannaneh Hajishirzi, Ali Farhadi, and Oren Etzioni. 2014. Diagram understanding in geometry questions. In *Proceedings of AAAI*, pages 2831–2838, Quebec.

Seo, Min Joon, Hannaneh Hajishirzi, Ali Farhadi, Oren Etzioni, and Clint Malcolm. 2015. Solving geometry problems: Combining text and diagram interpretation. In *Proceedings of EMNLP*, pages 1466–1476, Lisbon.

Shah, Parantu K., Carolina Perez-Iratxeta, Peer Bork, and Miguel A. Andrade. 2003. Information extraction from full text scientific articles: Where are the keywords? *BMC Bioinformatics*, 4(1):20.

Shimizu, Nobuyuki and Andrew R. Haas. 2009. Learning to follow navigational route instructions. In *IJCAI 2009, Proceedings of the 21st International Joint Conference on Artificial Intelligence*, pages 1488–1493, Pasadena, CA.

Siegel, Noah, Zachary Horvitz, Roie Levin, Santosh Divvala, and Ali Farhadi. 2016. Figureseer: Parsing result-figures in research papers. In *European Conference on Computer Vision*, pages 664–680, Amsterdam.

Soricut, Radu and Daniel Marcu. 2003. Sentence level discourse parsing using syntactic and lexical information. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology-Volume 1*, pages 149–156, Edmonton.

Stock, Oliviero. 1993. Alfresco: Enjoying the combination of NLP and hypermedia for information exploration. In *AAAI Workshop on Intelligent Multimedia Interfaces*, pages 197–224, Anaheim.

Strehl, Alexander and Joydeep Ghosh. 2002. Cluster ensembles — A knowledge reuse framework for combining multiple partitions. *Journal of Machine Learning Research*, 3(Dec):583–617.

Strunk, William. 2007. *The Elements of Style*. Penguin.

Subba, Rajen and Barbara Di Eugenio. 2009. An effective discourse parser that uses rich linguistic information. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 566–574, Boulder, CO.

Sun, Mingyu and Joyce Y. Chai. 2007. Discourse processing for context question answering based on linguistic knowledge. *Knowledge-Based Systems*, 20(6):511–526.

Twyman, Michael. 1985. Using pictorial language: A discussion of the dimensions of the problem. In *Designing Usable Texts*, Elsevier, pages 245–312.

Van Dijk, Teun A. 1972. *Some Aspects of Text Grammars*. Mouton & Co. N.V.

Wahlster, Wolfgang, Elisabeth André, Som Bandyopadhyay, Winfried Graf, and Thomas Rist. 1992. Wip: The coordinated generation of multimodal presentations from a common representation. In A. Ortony, editor, *Communication from an Artificial Intelligence Perspective*, Springer, pages 121–143.

Wang, D. Y., Robert Wing Pong Luk, Kam-Fai Wong, and K. L. Kwok. 2006. An information retrieval approach based on discourse type. In *International Conference on Application of Natural Language to Information Systems*, pages 197–202, Klagenfurt.

Wang, Jianxiang and Man Lan. 2015. A refined end-to-end discourse parser. In *CoNLL Shared Task*, pages 17–24, Beijing.

Wang, Shuting, Chen Liang, Zhaohui Wu, Kyle Williams, Bart Pursel, Benjamin Brautigam, Sherwyn Saul, Hannah Williams, Kyle Bowen, and C. Lee Giles. 2015. Concept hierarchy extraction from textbooks. In *Proceedings of the 2015 ACM Symposium on Document Engineering*, pages 147–156.

Wang, Shuting, Alexander Ororbia, Zhaohui Wu, Kyle Williams, Chen Liang, Bart Pursel, and C. Lee Giles. 2016. Using prerequisites to extract concept maps from textbooks. In *Proceedings of the 25th ACM International Conference on Information and Knowledge Management*, pages 317–326, Indianapolis.

Wen-Tsun, Wu. 1986. Basic principles of mechanical theorem proving in elementary geometries. *Journal of Automated Reasoning*, 2(3):221–252.

White, Michael. 1995. Presenting punctuation. *CoRR*, abs/cmp-lg/9506012.

Wilson, Sean and Jacques D. Fleuriot. 2005. Combining dynamic geometry, automated geometry theorem proving and diagrammatic proofs. In *Workshop on User Interfaces for Theorem Proving (UITP)*.

Wu, Jian, Jason Killian, Huaiyu Yang, Kyle Williams, Sagnik Ray Choudhury, Suppawong Tuarob, Cornelia Caragea, and C. Lee Giles. 2015. PDFMEF: A multi-entity knowledge extraction framework for scholarly documents and semantic search. In *Proceedings of the 8th International Conference on Knowledge Capture*, Palisades.

Yaghmazadeh, Navid, Yuepeng Wang, Isil Dillig, and Thomas Dillig. 2017. Type- and content-driven synthesis of SQL queries from natural language. *CoRR*, abs/1702.01168.

Yang, Yiming, Hanxiao Liu, Jaime G. Carbonell, and Wanli Ma. 2015. Concept graph learning from educational data. In *Proceedings of the Eighth ACM International Conference on Web Search and Data Mining*, pages 159–168, Shanghai.

Yao, Xuchen, Benjamin Van Durme, Chris Callison-Burch, and Peter Clark. 2013. A lightweight and high performance monolingual word aligner. In *Proceeding of ACL, Volume 2*, pages 702–707, Sofia.

Yin, Pengcheng and Graham Neubig. 2017. A syntactic neural model for general-purpose code generation. In the *55th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 440–450 Vancouver.

Zelle, John M. and Raymond J. Mooney. 1993. Learning semantic grammars with constructive inductive logic programming. In *Proceedings of the 11th National Conference on Artificial Intelligence*, pages 817–822, Washington, DC.

Zelle, John M. and Raymond J. Mooney. 1996. Learning to parse database queries using inductive logic programming. In *Proceedings of the Thirteenth National Conference on Artificial Intelligence*, pages 1050–1055, Portland.

Zettlemoyer, Luke S. and Michael Collins. 2012. Learning to map sentences to logical form: Structured classification with probabilistic categorial grammars. *arXiv preprint arXiv:1207.1420*.