From Textbooks to Knowledge: A Case Study in Harvesting Axiomatic Knowledge from Textbooks to Solve Geometry Problems

Mrinmaya Sachan **Avinava Dubey** **Eric P. Xing**
School of Computer Science
Carnegie Mellon University
{mrinmays, akdubey, epxing}@cs.cmu.edu

Abstract

Textbooks are rich sources of knowledge. Harvesting knowledge from textbooks is a key challenge in many educational applications. In this paper, we present an approach to obtain axiomatic knowledge of geometry in the form of horn-clause rules from math textbooks. The approach uses rich contextual and typographical features extracted from the textbooks. It also leverages the redundancy and shared ordering of axioms across multiple textbooks to accurately harvest axioms. These axioms are then parsed into horn-clause rules that are used to improve the state-of-the-art in solving geometry problems.

1 Introduction

Recently, researchers have proposed standardized tests as "drivers for progress in AI" (Clark and Etzioni, 2016). There is a growing body of work in solving standardized tests such as reading comprehensions (Richardson et al., 2013; Sachan et al., 2015, inter alia), science question answering (Schoenick et al., 2016; Sachan et al., 2016, inter alia), algebra word problems (Kushman et al., 2014, inter alia), geometry problems (Seo et al., 2015), pre-university entrance exams (Fujita et al., 2014), etc. A major challenge in building these solvers is the lack of subject knowledge. For example, geometry tests require knowledge of geometry axioms and pre-university exams require knowledge of laws of physics, chemistry, etc.

In this paper, we present an automatic approach that can (a) harvest such subject knowledge from textbooks, and (b) parse the extracted knowledge to structured programs that the solvers can use. Unlike information extraction systems trained on domains such as web documents (Chang et al., 2003; Etzioni et al., 2004, inter alia), learning an information extraction system that can extract axiomatic knowledge from textbooks is challenging because of the small amount of in-domain labeled data available for these tasks. We tackle this challenge by (a) leveraging the *redundancy* and *shared ordering* of axiom mentions across multiple textbooks[1], and (b) utilizing rich contextual and typographical features[2] from textbooks to effectively extract and parse axioms. Finally, we also provide an approach to parse the extracted axiom mentions from various textbooks and reconcile them to achieve the best program for each axiom.

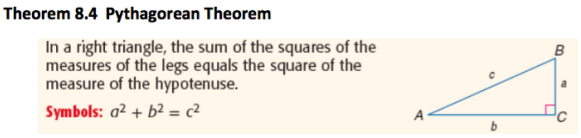


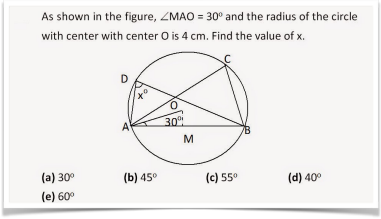
Figure 1: An excerpt of a textbook from our dataset that introduces the Pythagoras theorem. The textbook has a lot of typographical features that can be used to harvest this theorem: The textbook explicitly labels it as a "theorem"; there is a colored bounding box around it; an equation writes down the rule and there is a supporting figure. Our models leverages such rich contextual and typographical information (when available) to accurately harvest axioms and then parses them to horn-clause rules. The horn-clause rule derived by our approach for the Pythagoras theorem is: $isTriangle(ABC) \wedge perpendicular(AC, BC) \implies BC^2 + AC^2 = AB^2$.

As a case study, we use our approach to harvest axiomatic knowledge of geometry from math textbooks, and use this knowledge to improve the state-of-the-art system for solving SAT style geometry problems. Seo et al. (2015) recently presented *GEOS*, an automated end-to-end system that solves SAT style geometry questions such as the one shown in Figure 2. *GEOS* derives a logical expression that represents the meaning of the

[1]The same axiom can be potentially mentioned in a number of textbooks in different ways. All textbooks typically introduce axioms in roughly the same order – for example, pythagorous theorem would typically be introduced after introducing the notion of a right angled triangle.

[2]Textbooks contain rich context and typographical information (see Figure 1 for an illustrative example). We use this rich information as features in our model.

Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, pages 782–793
Copenhagen, Denmark, September 7–11, 2017.



Text Description:

measure(∠MAO, 30°)
isCircle(O)
radius(O, 4 cm)
?x

Diagram:

liesOn(A, circle O), liesOn(B, circle O),
liesOn(C, circle O), liesOn(D, circle O)
isLine(AB), isLine(BC), isLine(CA), isLine(BD), isLine(DA)
isTriangle(ABC), isTriangle(ABD), isTriangle(AOM)
measure(∠ADB, x), measure(∠MAO, 30°)
measure(∠AMO, 90°)
...

Figure 2: An example SAT style geometry problem with the question text, corresponding diagram and (optionally) answer candidates. Below: A logical expression that represents the meaning of the text description and the diagram in the problem. *GEOS* derives a weighted logical expression where each predicates also carries a weighted score but we do not show them here for clarity.

text description and the diagram (also shown in Figure 2), and then solves the geometry question by checking the satisfiablity of the derived logical expression. While this solver has its basis in coordinate geometry and indeed works, it has some key issues: *GEOS* requires an explicit mapping of each predicate into a set of constraints over point coordinates[3]. 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[4].

We use our model to extract and parse axiomatic knowledge from a novel dataset 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 to *GEOS*.

2 Background: GEOS

Our work reuses *GEOS* to parse the question text and diagram into its formal problem description as shown in Figure 2. *GEOS* parses the question text and the diagram to a formal problem description. *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). For text parsing, *GEOS* takes a multi-stage 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* use 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. Since this is a cumbersome process, *GEOS* has an incomplete mapping of literals to indicator functions.

3 Set up for the Axiomatic Solver

In this work, we replace the numerical solver of *GEOS* 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 3. The logical program has a set of declarations from the *GEOS* text and diagram parsers which describe the problem specification

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

[4]For example, the deductive reasoning required to solve the question in Figure 2 is: (1) Use the axiom that the sum of interior angles of a triangle is 180°and the fact that ∠AMO is 90°to conclude that ∠MOA is 60°. (2) △MOA ∼ △MOB (using a similar triangle axiom) and then, ∠MOB = ∠MOA = 60°(using the axiom that corresponding angles of similar triangles are equal). (3) Use angle sum rule to conclude that ∠AOB = ∠MOB + ∠MOA = 120°. (4) Use the axiom that the angle subtended by an arc of a circle at the centre is double the angle subtended by it at any point on the circle to conclude that ∠ADB = 0.5×∠AOB = 60°.

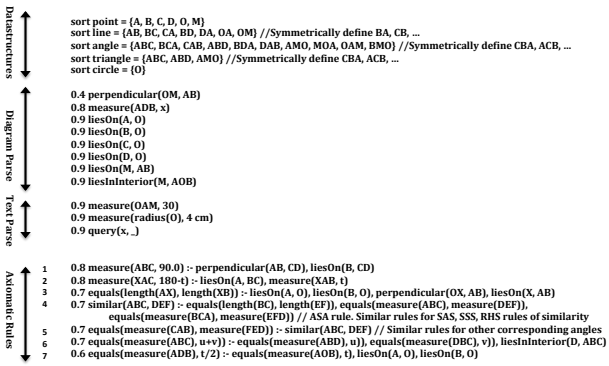


Figure 3: A sample logical program (in prolog style) that solves the problem in Figure 2. The program consists of a set of data structure declarations that correspond to types in the prolog program, a set of declarations from the diagram and text parse and a subset of the geometry axioms written as horn 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 readers 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.

and the parsed horn clause rules describe the underlying theory. Normalized confidence scores from question text, diagram and axiom parsing models are used as probabilities in the program. Next, we describe how we harvest structured axiomatic knowledge from textbooks.

4 Harvesting Axiomatic Knowledge

We present a structured prediction model that identifies axioms in textbooks and then parses them. Since harvesting axioms from a single textbook is a very hard problem, we use multiple textbooks and leverage the redundancy of information to accurately 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 (XML in our experiments), we extract chapters relevant for geometry in each of them to obtain a sequence of sentences (with associated typographical information) from each textbook. Let $\mathbf{S}_b = \{s_0^{(b)}, s_1^{(b)}, \ldots s_{|\mathbf{S}_b|}^{(b)}\}$ denote the sequence of sentences in textbook b. $|\mathbf{S}_b|$ denotes the number of sentences in textbook b.

4.1 Axiom Identification and Alignment

We decompose the problem of extracting axioms from textbooks into two tractable sub-problems: (a) identification of axiom mentions in each textbook using a sequence labeling approach, and (b) aligning 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. Joint modeling of the axiom identification and alignment is necessary as both sub-problems can help each other.

4.1.1 Axiom Identification

Linear-chain CRF formulation (Lafferty et al., 2001) can be used for the subproblem of axiom identification. Given $\{\mathbf{S}_b | b \in \mathcal{B}\}$, the model labels each sentence $s_i^{(b)}$ as **B**efore, **I**nside or **O**utside an axiom. Hereon, a contiguous block of sentences labeled $\mathbf{B}$ or $\mathbf{I}$ 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 CRF 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 L-BFGS to optimize the objective and Viterbi decoding for inference.

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 1) include various content based features encoding various notions of similarity between pairs of sentences as well as various typographical features such as whether the sentences are annotated as an axiom (or theorem or corollary) in the textbook, contain equations, diagrams, text that is bold or italicized, are in the same node of the xml hierarchy, are contained in a bounding box, etc.

Some extracted axiom mentions contain pointers to a diagram eg. "Figure 2.1". We consider the diagram to be a part of the axiom mention.

4.1.2 Axiom Alignment

Next, we leverage the redundancy of information and the relatively fixed ordering of axioms in various textbooks by aligning various mentions of the same axiom across textbooks and introducing structural constraints on the alignment.

Content	Sentence Overlap	Semantic Textual Similarity between the current and next sentence. We include features that compute the proportion of common unigrams and geometry entities (constants, predicates and functions) across the two sentences. This feature is conjoined with the tag assigned to the current and next sentence.
	Geometry entities	No. of geometry entities (normalized by the number of tokens) in this sentence. This feature is conjoined with the tag assigned to the current sentence.
	Intra-sentence semantics	Indicator that the current sentence 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 sentence.
Typography	Axiom, Theorem, Corollary Mention	(a) The current (or previous) sentence 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) sentence mentions an Axiom, Theorem or Corollary. This feature is conjoined with the tag assigned to the current (and previous) sentence.
	Eqn. Template	The current (or next) sentence contains an equation eg. $PA \times PB = PT^2$. This feature is conjoined with the tag assigned to the current (and next) sentence.
	Assoc. Diagram	The current sentence contains a pointer to a figure eg. "Figure 2.1". This feature is conjoined with the tag assigned to the current sentence.
	RST edge	Indicator for the RST relation between the current and next sentence. This feature is conjoined with the tag assigned to the current and next sentence.
	Bold/Underline	The sentence (or previous) sentence contains text that is in bold font or underlined. Conjoined with the tag assigned to the current (and previous) sentence.
	XML structure	Indicator that the current and previous sentence are in the same node of the XML hierarchy. Conjoined with the tag assigned to the current and previous sentence.
	Bounding box	Indicator that the current and previous sentence are bounded by a bounding box in the textbook. Conjoined with the tag assigned to the current and previous sentence.

Table 1: Feature set for our axiom identification model. The features are based on content and typography.

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 $\mathbf{A}^* = (A_1^*, A_2^*, \ldots, A_U^*)$ where U is some pre-defined upper bound on the total number of axioms in geometry. Then, we emphasize that the axiom mentions extracted from each textbooks (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:

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

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

C1: An axiom appears in one 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:

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

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

Note that the constraints $C1 \ldots 3$ define the feasible space of alignments. Our sampler always samples the next $Z_{ik}^{(b)}$ in this feasible space.

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:

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

To model these constraints as soft constraints, we penalize the model for violating these constraints. Let the penalty for violating the above constraint be $\exp\left(\nu \max\left(0, 1 - Z_{ij}^{(b)} - Z_{kl}^{(b)}\right)\right)$. We introduce a new regularization term: $\mathbf{R}(\mathbf{Z}) = \sum_{\substack{1\leq i<k\leq|\mathbf{A}_b|\\ 1\leq l<j\leq U\\ b\in\mathcal{B}}} \exp\left(\nu \max\left(0, 1 - Z_{ij}^{(b)} - Z_{kl}^{(b)}\right)\right)$. 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 (eq 1), but:

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

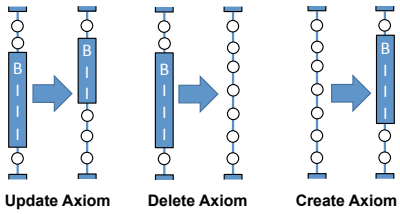


Figure 4: An illustration of the three operations to sample axiom blocks.

$$+\nu \sum_{\substack{b' \in \mathcal{B} \\ b' \neq b}} \sum_{i<j\leq |A_{b'}|} \sum_{1\leq l<k} \left(1 - Z_{ik}^{(b)} - Z_{jl}^{(b')}\right)$$
$$+\nu \sum_{\substack{b' \in \mathcal{B} \\ b' \neq b}} \sum_{1\leq j<i|} \sum_{k<l\leq U} \left(1 - Z_{ik}^{(b)} - Z_{jl}^{(b')}\right)$$

Features: Now, we describe the features g. These too include content based features encoding various notions of similarity between pairs of axiom mentions as well as various typographical features. The features are listed in Table 2.

4.1.3 Joint Identification and Alignment

Joint modeling of axiom identification and alignment components is useful as both problems potentially help each other. Let $Y_{ij}^{(b)}$ denote that the sentence $s_i^{(b)}$ 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 is not aligned to 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:

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

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

We write down the model constraints below:

C1': Every sentence 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 $\mathbf{Y}'s$ and $\mathbf{Z}'s$ alternatively. Sampling for $\mathbf{Z}'s$ reduces to Gibbs sampling and the sampling equations are as same as before (Section 4.1.2). For better mixing, we sample $\mathbf{Y}$ in blocks. Consider blocks of $\mathbf{Y}$'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 4) 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 sentences as O.

Introduce axiom: Given a contiguous sequence of sentences 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:

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

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.

4.2 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 multi-set of axiom extractions. Let $\mathcal{A} = \{\mathbf{A}_1, \mathbf{A}_2, \ldots, \mathbf{A}_{|\mathcal{A}|}\}$ represent the multi-set where each axiom $\mathbf{A}_i$ is mentioned at least once.

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.

4.2.1 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 Pythagoras 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 ax-

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 sub-sequence of words between two axiom mentions normalized by the total number of words in the two mentions.
Number of sentences	Real valued feature that computes the absolute difference in the number of sentences 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.
Equation Template	Indicator feature that matches templates of equations detected in the axiom mentions.
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 doesn't fire.
XML structure	Indicator matching the current (and parent) node of axiom mentions in respective XML hierarchies.

Table 2: Feature set for our axiom alignment model. The features are based on content, structure and typography.

iom 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. We use L-BGFGS for learning.

Features: We list the features $\mathbf{h}$ in Table 3. The features are defined over candidate spans forming the text split, are strongly inspired from rhetorical structure theory (Mann and Thompson, 1988) and previous works on discourse parsing (Marcu, 2000; Soricut and Marcu, 2003). Given a beam of *Premise* and *Conclusion* splits, we use the *GEOS* parser to get *Premise* and *Conclusion* logical formulas for each split in the beam and obtain a beam of axiom parses for each axiom in each textbook.

4.2.2 Multi-source Axiomatic Parser

Now, we describe a multi-source 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 picks 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.

5 Experiments

Datasets: 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. 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. All the hyper-parameters in all the models are tuned on the development set using grid search.

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 datasets for evaluating our system: (a) practice and official SAT style geometry questions used in *GEOS*, and (b) an additional dataset of geometry questions collected from the aforementioned textbooks. This dataset consists of a total of 1406 SAT style questions across grades 6-10, and is approximately 7.5 times the size of the dataset used in *GEOS*. We split the dataset into training (350 questions),

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.
Punctuation	Punctuation at the segment border is an excellent cue. We include indicator features whether there is a punctuation at the segment border.
Text Organization	Indicator that the two text spans are part of the same (a) sentence, (b) paragraph.
XML Structure	Indicator that the two spans are in the same node in the XML hierarchy. Conjoined with the indicator feature that the two spans are part of the same paragraph.
RST Parse	We use an off-the-shelf RST parser (Feng and Hirst, 2014) and include an indicator feature that the segmentation matches the parse segmentation. We also include the RST label as a feature.
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.
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 feature. This feature uses both lexical and syntactic information.
Head / Common Ancestor/ Attachment Node	Head node is 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 have 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 (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 which 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 is a feature. See Soricut and Marcu (2003) for details.
Span Similarity	Proportion of (a) words (b) geometry relations (c) relation-arguments shared by the two spans.
No. 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.
Relative Position	Relative position of the two lexical heads and the text split in sentence.

Table 3: Feature set for our axiom parsing model.

	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 4: 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.

	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**

Table 5: 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.

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 expanded set of entity types, functions and predicates. We call this system *GEOS++*.

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 4 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 the 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 of axiom mention clustering. Table 5 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 6 shows the re-

		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 6: 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 multi-source parsers presented in this paper.

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

Table 7: Scores for solving geometry questions on the SAT practice and official datasets and a dataset of questions from the 20 textbooks. We use SATs 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).

sults 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 multi-source 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 multi-source 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.

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 3 horn clause parses for each axiom and use them as the underlying theory in prolog programs such as the one shown in Figure 3. 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 7 shows the results for our best end-to-end system and compares it to *GEOS* on the practice and official SAT dataset from Seo et al. (2015) as well as questions from the 20 textbooks. On all the three datasets, our system outperforms *GEOS*. Especially on the dataset from the 20 textbooks (which is indeed a harder dataset and includes more problems which require complex reasoning based on geometry), *GEOS* doesn't 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.

	Interpretability		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 8: User study ratings for *GEOS* and our system (O.S.) by students in grade 6-10. Ten students in each grade were asked to rate the two systems on a scale of 1-5 on two facets: 'interpretability' and 'usefulness'. Each cell shows the mean rating computed over ten students in that grade for that facet.

Interpretability: Students around the world solve geometry problems through rigorous deduction whereas the numerical solver in *GEOS* does not provide such interpretability. 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 interpretability 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. They were each asked to rate how 'interpretable' and 'useful' the two systems were on a scale of 1-5. Table 8 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 of an interpretable deductive solver for geometry problems.

6 Related Work

Solving Geometry Problems: While 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). While Seo et al. (2014) only aligned geometric shapes with their textual mentions, 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 which may not always be available for such problems, and (b) it lacked the deductive geometric reasoning used by students to solve these problems. Our axiomatic solver mitigates these issues by performing deductive reasoning using axiomatic knowledge extracted from textbooks.

Information Extraction from Textbooks: Our model builds upon ideas from Information extraction (IE), which is the task of automatically extracting structured information from unstructured and/or semi-structured documents. While there has been a lot of work in IE on domains such as web documents (Chang et al., 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 present an approach to 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 et al., 2011; Berant et al., 2013; Yaghmazadeh et al., 2017, inter alia), commands to robots (Shimizu and Haas, 2009; Matuszek et al., 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 et al. (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.

7 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 paper focused on harvesting geometry axioms from textbooks as a case study, it can be extended to obtain valuable structured knowledge from textbooks in areas such as science, engineering and finance.

References

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

Michele Banko, 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, Hyderabad, India, January 6-12, 2007*, pages 2670–2676.

Jonathan Berant, 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, 18-21 October 2013, Grand Hyatt Seattle, Seattle, Washington, USA, A meeting of SIGDAT, a Special Interest Group of the ACL*, pages 1533–1544.

Michael J. Cafarella, 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, 6-8 October 2005, Vancouver, British Columbia, Canada*, pages 563–570.

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

Yee Seng Chan and Hwee Tou Ng. 2008. Maxsim: A maximum similarity metric for machine translation evaluation. In *The 2008 Annual Conference of the Association for Computational Linguistics (ACL)*.

Chia-Hui Chang, 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.

Chia-Hui Chang, 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.

David L. Chen 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.

Peter Clark and Oren Etzioni. 2016. My computer is an honor student - but how intelligent is it? standardized tests as a measure of ai. In *Proceedings of AI Magazine*.

Peter Clark, 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. Association for Computational Linguistics.

Bhavana Dalvi, 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, San Diego, CA, USA, June 17, 2016*, pages 12–17.

Tom Davis. 2006. Geometry with computers. Technical report.

Michael Denkowski 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. Association for Computational Linguistics.

Oren Etzioni, Michele Banko, Stephen Soderland, and Daniel S Weld. 2008. Open information extraction from the web. *Communications of the ACM*, 51(12):68–74.

Oren Etzioni, 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, July 25-29, 2004, San Jose, California, USA*, pages 391–398.

Edward A Feigenbaum and Julian Feldman. 1963. *Computers and thought*. The AAAI Press.

Vanessa Wei Feng 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.

Akira Fujita, Akihiro Kameda, Ai Kawazoe, and Yusuke Miyao. 2014. Overview of todai robot project and evaluation framework of its nlp-based problem solving. *World History*, 36:36.

Rohit J Kate, Yuk Wah, Wong Raymond, and J Mooney. 2005. Learning to transform natural to formal languages. In *Proceedings of AAAI-05*. Citeseer.

Nate Kushman, Yoav Artzi, Luke Zettlemoyer, and Regina Barzilay. 2014. Learning to automatically solve algebra word problems. In *Proceedings of ACL*.

John Lafferty, 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.

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

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

Percy Liang, 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. Association for Computational Linguistics.

Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out: Proceedings of the ACL-04 workshop*, volume 8. Barcelona, Spain.

Wang Ling, 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*.

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

Chang Liu, 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*, pages 4574–4582. Curran Associates, Inc.

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

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

Christopher D. Manning, 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.

Daniel Marcu. 2000. *The theory and practice of discourse parsing and summarization*.

Cynthia Matuszek, 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. IEEE.

T. Mitchell, 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)*.

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

Chris Quirk, 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, ACL 2015, July 26-31, 2015, Beijing, China, Volume 1: Long Papers*, pages 878–888.

Matthew Richardson, Christopher JC Burges, and Erin Renshaw. 2013. Mctest: A challenge dataset for the open-domain machine comprehension of text. In *Proceedings of Empirical Methods in Natural Language Processing (EMNLP)*.

Mrinmaya Sachan, 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*.

Mrinmaya Sachan, Kumar Avinava Dubey, and Eric P. Xing. 2016. Science question answering using instructional materials. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, ACL 2016, August 7-12, 2016, Berlin, Germany, Volume 2: Short Papers*.

Mrinmaya Sachan and Eric P. Xing. 2016. Easy questions first? A case study on curriculum learning for question answering. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, ACL 2016, August 7-12, 2016, Berlin, Germany, Volume 1: Long Papers*.

Ozair Saleem 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.

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

Carissa Schoenick, Peter Clark, Oyvind Tafjord, Peter D. Turney, and Oren Etzioni. 2016. Moving beyond the turing test with the allen AI science challenge. *CoRR*, abs/1604.04315.

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

Min Joon Seo, Hannaneh Hajishirzi, Ali Farhadi, Oren Etzioni, and Clint Malcolm. 2015. Solving geometry problems: combining text and diagram interpretation. In *Proceedings of EMNLP*.

Parantu K Shah, 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.

Nobuyuki Shimizu and Andrew R. Haas. 2009. Learning to follow navigational route instructions. In *IJCAI 2009, Proceedings of the 21st International Joint Conference on Artificial Intelligence, Pasadena, California, USA, July 11-17, 2009*, pages 1488–1493.

Radu Soricut 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. Association for Computational Linguistics.

Alexander Strehl and Joydeep Ghosh. 2002. Cluster ensembles—a knowledge reuse framework for combining multiple partitions. *Journal of machine learning research*, 3(Dec):583–617.

Shuting Wang, 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. ACM.

Shuting Wang, 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 on Conference on Information and Knowledge Management*, pages 317–326. ACM.

Jian Wu, 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*, K-CAP 2015.

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

Yiming Yang, 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, WSDM 2015, Shanghai, China, February 2-6, 2015*, pages 159–168.

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

Pengcheng Yin 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)*, Vancouver, Canada.

John M. Zelle and Raymond J. Mooney. 1993. Learning semantic grammars with constructive inductive logic programming. In *Proceedings of the 11th National Conference on Artificial Intelligence. Washington, DC, USA, July 11-15, 1993.*, pages 817–822.

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

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