Diagram Understanding in Geometry Questions

Min Joon Seo[1], **Hannaneh Hajishirzi**[1], **Ali Farhadi**[1], **Oren Etzioni**[2]
[1]{minjoon, hannaneh, farhadi}@washington.edu, [2]orene@allenai.org
[1]University of Washington, [2]Allen Institute for AI

Abstract

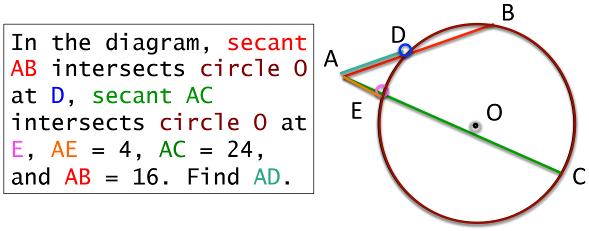
Automatically solving geometry questions is a long-standing AI problem. A geometry question typically includes a textual description accompanied by a diagram. The first step in solving geometry questions is diagram understanding, which consists of identifying visual elements in the diagram, their locations, their geometric properties, and aligning them to corresponding textual descriptions. In this paper, we present a method for diagram understanding that identifies visual elements in a diagram while maximizing agreement between textual and visual data. We show that the method's objective function is submodular; thus we are able to introduce an efficient method for diagram understanding that is close to optimal. To empirically evaluate our method, we compile a new dataset of geometry questions (textual descriptions and diagrams) and compare with baselines that utilize standard vision techniques. Our experimental evaluation shows an F1 boost of more than 17% in identifying visual elements and 25% in aligning visual elements with their textual descriptions.

1 Introduction

Designing algorithms that can automatically solve math and science questions is a long-standing problem in AI (Feigenbaum and Feldman 1963). In this paper, we focus on geometry questions where the question text is accompanied by a diagram. More specifically, we address the problem of diagram understanding in geometry questions (Figure 1), a prelude to more sophisticated diagram understanding in scientific textbooks.

Diagram understanding is the problem of discovering visual elements, their locations, their geometric properties in the diagram, and their alignment to text. For example, understanding the diagram in Figure 1 entails identifying the location and the area of the circle O, secant AB, their geometric relations, and aligning pixels in the diagram to the corresponding textual mentions (color coded in the figure).

By and large, previous work in diagram understanding has studied the problems of text analysis and diagram understanding separately. Several algorithms have identified individual shapes such as circles (Zhang and Sun 2011), lines (Climer and Bhatia 2003), triangles or rectangles (Li et al. 2013; Jung and Schramm 2004) from images, but ignore other shapes in the diagram, and do not attempt to *discover* them as we do. Furthermore, little attention has been paid to identifying shapes in a diagram by also utilizing the corresponding text.



```
In the diagram, secant
AB intersects circle O
at D, secant AC
intersects circle O at
E, AE = 4, AC = 24,
and AB = 16. Find AD.
```

Figure 1: Diagram understanding: identifying visual elements in the diagram and aligning them with their textual mentions. Visual elements and their corresponding textual mentions are color coded. This Figure is best viewed in color.

Inspired by the growing body of work that has coupled textual and visual signals (e.g., (Gupta and Mooney 2010)), we present a novel method G-ALIGNER for diagram understanding in geometry questions by discovering visual elements and aligning them with their corresponding textual mentions. Our G-ALIGNER method identifies visual elements by maximizing the coverage of explained pixels of the diagram, the agreement between visual elements and their textual mentions, and the coherence of the identified elements. G-ALIGNER can identify a wide range of shapes including lines, circles, polygons, and other shapes composed from visual primitives (see Section 5).

We show that G-ALIGNER's objective function is submodular. This observation allows us to devise a greedy but accurate approximation procedure to identify visual elements in diagrams and align them with text. G-ALIGNER has another key advantage in being much more robust than standard vision techniques like the Hough transform (Stockman and Shapiro 2001). Whereas standard vision techniques require parameter tuning when moving from one diagram to the next, based on factors like the number of shapes in the diagram and their size, G-ALIGNER does not.

To evaluate G-ALIGNER, we manually compiled a dataset of geometry questions (textual descriptions and diagrams) that includes ground truth labels for visual elements and their correct alignments to textual mentions. To our knowledge, no comparable dataset existed previously. We evaluate G-ALIGNER on two tasks of identifying visual elements and aligning them to mentions in text. Our experiments show that for both tasks G-ALIGNER significantly outperforms baselines that use standard vision techniques. Moreover, our experiments show the benefit of incorporating textual information.

Our contributions are three-fold: (a) We present G-ALIGNER, a method for diagram understanding that both discovers visual elements in diagrams and aligns them to textual mentions; (b) We introduce a submodular optimization formulation and a greedy but accurate approximation procedure for diagram understanding; (c) We introduce a new dataset for geometry questions that includes ground truth labels for visual elements and their alignment to textual mentions. Our experiments show improvement of at least 25% in F1 over baselines in identifying visual elements, and of 17% in aligning visual elements to textual mentions.[1]

2 Related Work

Diagram understanding has been explored since early days in AI (e.g., (Srihari 1994; Lin et al. 1985; Ferguson and Forbus 1998; Hegarty and Just 1989; Ferguson and Forbus 2000; Novak 1995)). Space does not allow comprehensive review of original attempts at the problem. We refer interested readers to (O'Gorman and Kasturi 1995). Most previous work differ from our method because they address two problems of geometry understanding and text understanding in isolation. Our paper is related to early work on coupling over textual and visual data (Bulko 1988; Novak and Bulko 1990; Srihari 1994), however these methods assume that the visual primitives of diagrams are manually identified. This paper aims at revisiting the problem of diagram understanding by coupling two tasks of visual understanding of diagrams and detecting alignments between text and diagrams.

The most common approach to diagram understanding is a bottom up method where primitives can be linked together (Lin and Nevatia 1998) to form larger elements such as rectangles (Li et al. 2013) or general shapes (Moon, Chellappa, and Rosenfeld 2002). Using Hough transform is another popular alternative in detecting visual elements (Zhang and Sun 2011; Jung and Schramm 2004). What is common among almost all conventional methods of visual element identification is thresholding of a scoring function that determines the existence of visual elements. Although being considered as a well studied subject, our experiments reveal that the thresholding step hinders applications of such techniques on real-world geometry questions. Our data suggests that there is no single threshold that results in a reliable discovery of visual elements across different diagrams. In this paper, we propose a method that initially overestimates the visual elements, but then benefits from submodular optimization coupled with textual information to home in on the correct elements.

Coupling visual and textual information has recently attracted attention in both vision and NLP (Farhadi et al. 2010; Kulkarni et al. 2011; Gupta and Mooney 2010). We build on this powerful paradigm, but utilize it for the more manageable task of understanding diagrams in geometry questions. Understanding these diagrams is more manageable because diagrams are less ambiguous, expose less visual variance, have smaller vocabulary of elements than images typically studied in machine vision. This easier task allows us to have more reliable estimates of visual elements and focus on interactions between textual mentions and visual elements.

3 Problem Definition

This paper addresses the problem of understanding diagrams (Figure 1) by coupling discovering visual elements in the diagram with aligning them with textual mentions. Before giving the formal description of the problem, we first define keywords that we use throughout the paper.

Definition 1. A *primitive* is a line segment or a circle segment (arc) extracted from the diagram. The set of all primitives extracted from a diagram image is $L = \{L_1, L_2, ..., L_n\}$.

Definition 2. A *visual element* is a combination of primitives that has specific properties. For instance, a *triangle* is a visual element that consists of three connected *lines* in a specific way. The vocabulary of all visual elements and their corresponding geometric properties is represented with V. The primitives in V includes: line, segment, chord, diameter, secant, tangent, radius, circle, arc, point, intersection, triangle, rectangle, trapezoid, square, altitude, base. For their geometric properties, please refer to our project web page.

Definition 3. A *textual mention* is a word or phrase that corresponds to a visual element. For instance, the word `circle` is the textual mention of the visual element *circle*. The set of all textual mentions extracted from the question is $T = \{T_1, T_2, ..., T_m\}$.

The input to our method is an image of a diagram with non-white pixels D accompanied with the text of the question that includes textual mentions T. The output is a subset of primitives along with their alignments to textual mentions. Figure 1 shows examples of detections and alignments established by our method.

4 Optimization for Primitive Identification and Alignment

Our key insight is to benefit from coupling textual and visual information available in geometry questions. This problem is a search for the best subset $\hat{L}$ of all initial primitives L extracted from the diagram. An ideal subset $\hat{L}$ should contain primitives that: (1) explain all important pixels in the diagram, (2) are visually coherent, and (3) form visual elements that align well with textual mentions in the question.

[1] Our dataset and a demo of G-ALIGNER are publicly available at:http://cs.washington.edu/research/ai/geometry

4.1 Formulation

We first intuitively define a set function $\mathcal{F}$ that measures the quality of a subset $\hat{L}$ based on the above properties (Equation 1). We then introduce the formal definition (Equation 2).

First, $\mathcal{F}$ has a component $\mathcal{P}$ to ensure that the ideal subset $\hat{L}$ has good coverage of the diagram image. That is, most of the non-white pixels in the diagram D should be explained by the subset of primitives $\hat{L}$.

Second, $\mathcal{F}$ has a component $\mathcal{C}$ to encourage the selection of primitives $\hat{L}$ that can form a bigger and coherent visual element. This can be encoded by the visual agreement between primitives in terms of distances between identified primitives in $\hat{L}$ and the actual corners C (corners are extracted from the diagram image and explained in Section 5.1).

Third, $\mathcal{F}$ has a component $\mathcal{S}$ to model the alignment between textual mentions T in the question and visual elements discovered from the diagram.

For any given subset $\hat{L} \subseteq L$, we define:

$$\mathcal{F}(\hat{L}, D, T) = \mathcal{P}(D, \hat{L}) + \mathcal{C}(C, \hat{L}) + \mathcal{S}(T, \hat{L}) \qquad (1)$$

where D is the diagram image, T is the set of all textual mentions in the question. The best subset is the one that maximizes the set function $\mathcal{F}$.

Here, we present an optimization for identifying primitives and aligning them with textual mentions. We introduce a binary matrix $W \in \{0,1\}^{|L|\times|T|}$ where $W_{i,j}$ identifies whether the i^{th} primitive l_i is aligned with the j^{th} textual mention T_j, or not. In particular, each row i in the identifier matrix W represents textual mentions that include the primitive l_i, and each column j in the identifier matrix represents primitives that are aligned with the textual mention T_j. Therefore, a matrix W can represent both the set of primitives as well as alignment between textual mentions and visual elements.

We reformulate the problem of searching for the best subset $\hat{L}$ that maximizes $\mathcal{F}$ in Equation 1 as the problem of finding an identifier matrix $W \in \{0,1\}^{|L|\times|T|}$. Optimizing for $\hat{W}$ results in a unified solution for both problems of primitive identification and alignment. In this setting, $\hat{L} = L \times (\hat{W} \times \mathbf{1}_{|T|\times 1})$ where the binary vector $\hat{W} \times \mathbf{1}_{|T|\times 1}$ represents what primitives in L are included in $\hat{W}$. Therefore, $\mathcal{P}(D, \hat{L})$ in Equation 1 is represented as $\mathcal{P}(D, L \times (\hat{W} \times \mathbf{1}))$ in the new setting. Finally, the optimization in equation 1 is reformulated as follows:

$$\begin{aligned}\mathcal{F}(W, L, D, T) = & \qquad (2)\\ \mathcal{P}(D, L \times (W \times \mathbf{1})) + \mathcal{C}(C, L \times (W \times \mathbf{1})) + \mathcal{S}(T, W) &\end{aligned}$$

The best subset of primitives and alignments are derived by maximizing for the identifier matrix $\hat{W} = \arg\max_W \mathcal{F}$. Here we formally define each component in the equation.

Definition 4. Let D be the set of pixels in the diagram, L be the set of all the primitives initially identified in the diagram, W be the identifier matrix, and $\hat{L} = L \times (\hat{W} \times \mathbf{1})$ be the set of identified primitives.

- **Coverage function $\mathcal{P}$:** If $D_{\hat{L}}$ represents the set of pixels covered by the identified primitives $\hat{L}$ then $\mathcal{P} : D \times L \to \mathbb{R}$ is $\mathcal{P}(D, \hat{L}) = \frac{|D_{\hat{L}}|}{|D|}$.
- **Visual coherence function $\mathcal{C}$:** Let C be the set of corners initially detected in the diagram. We consider a corner $c \in C$ to be matching, if there exists a point $e \in D_{\hat{L}}$ that is close enough to the detected corner c (i.e., $|c - e| < \epsilon$ for a fixed ϵ). If $C_{\hat{L}}$ is the set of matched corners, then $\mathcal{C} : C \times L \to \mathbb{R}$ is $\mathcal{C}(C, \hat{L}) = \frac{|C_{\hat{L}}|}{|C|}$.
- **Alignment constraint function $\mathcal{S}$:** Let T be the set of textual mentions in the text of the question. The vocabulary V consists of geometric descriptions of each visual element. For example, *Circle* corresponds to the set of points that have the same distance to the center, etc. A textual mention in the text is aligned if our method can find a corresponding model from the primitives. For example, to align a textual mention like `Triangle ABC`, our method needs to find three *lines* that mutually intersect at corners close to labels *A*, *B*, and *C* in the diagram.[2] A visual element like a *triangle* can be textually described in multiple different ways. For example, `Triangle ABC` or three lines `AB, BC, AC`. To avoid redundancy between the visual elements, we need to penalize our model for predicting overlapping visual elements. We define redundancy between two lines l_1, l_2 as a function of the intersection of the projection of l_2 to l_1 over their union. For arcs we do the same with the convex area of the arcs.
If $T_{\hat{W}}$ is the set of textual mentions covered in $\hat{W}$, and $r_{\hat{W}}$ is the redundancy among the primitives in $\hat{W}$ that are not mentioned in $T_{\hat{W}}$ then $\mathcal{S} : T \times W \to \mathbb{R}$ is $\mathcal{S}(T, \hat{W}) = \frac{|T_{\hat{W}}|}{|T|} - r_{\hat{W}}$.

Optimizing Equation 2 is a combinatorial optimization that requires $2^{|L|}$ evaluations of $\mathcal{F}$. In the next section we show how to optimize Equation 2.

4.2 Optimization

Optimizing for Equation 2 is NP-hard by reduction from weighted set cover problem. However, the objective function is submodular. This means that there exists a greedy method that can accurately approximate the optimal solution.

Lemma 1. The objective function $\mathcal{F}$ in Equation 2 is submodular.

Proof sketch. To show that the objective $\mathcal{F}$ in Equation 2 is submodular we need to show that for $L'' \subseteq L' \subseteq L$, and for $l \in L \setminus L'$

$$\mathcal{F}(L'' \cup l) - \mathcal{F}(L'') \geq \mathcal{F}(L' \cup l) - \mathcal{F}(L') \qquad (3)$$

We compare components of $\mathcal{F}$ in two sides of inequality 3:

$$\begin{aligned}(|D_{L''\cup l}| - |D_{L''}|)/|D| &\geq (|D_{L'\cup l}| - |D_{L'}|)/|D|\\ (|T_{L''\cup l}| - |T_{L''}|)/|T| &\geq (|T_{L'\cup l}| - |T_{L'}|)/|T|\\ (|C_{L''\cup l}| - |C_{L''}|)/|C| &\geq (|C_{L'\cup l}| - |C_{L'}|)/|C|\\ -(|r_{L''\cup l} - r_{L''}) &\geq -(|r_{L'\cup l} - r_{L'})\end{aligned}$$

[2] For finding positions of labels we use an off-the-shelf OCR package of Tesseract.

```
Inputs:
• $V$: the set of known visual elements and their geometric properties.
• $D$: the set of non-white pixels in the diagram.
• $\hat{L}$: the set of identified primitives
1. **Initialization** (section 5.1)
  (a) Initialize primitives $L$ in the diagram
    i. Run Hough transform to initialize lines and circles segments
    ii. set $L \leftarrow$ top $n$ picks from the output of the line and circle detection where $n$ is generously high
  (b) Initialize corners $C$ in the diagram
  (c) Initialize mentions $T$ in the text
2. **Optimize Equation 2 to identify primitives and alignments given the diagram and text** (section 5.2)
  (a) Let $\hat{L} \leftarrow \emptyset$
  (b) Repeat
    i. For every primitive $l \in L$:
      A. Compute $\mathcal{G}(l) \leftarrow \mathcal{F}(\hat{L} \cup l) - \mathcal{F}(\hat{L})$ using $\mathcal{P}, \mathcal{C}, \mathcal{S}$ from Equation 2
    ii. select $l \leftarrow \arg\max_{l \in L} \mathcal{G}(l)$
    iii. add $l$ to the set of primitives $\hat{L}$
  (c) until $\nexists l \in L$ such that $\mathcal{G}(l) > 0$.
```

Figure 2: G-ALIGNER: Method for coupling primitive identification and alignment.

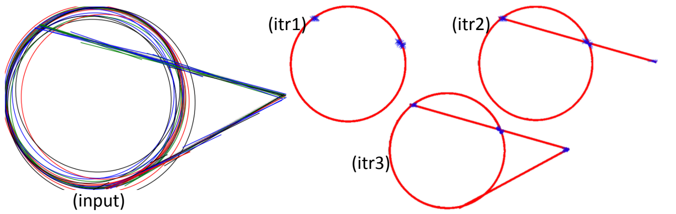


Figure 3: This figure shows steps of the method. It starts with an over-generation of primitives and at each iteration adds a primitive that provides the biggest gain based on Equation 4. Red line segments correspond to primitives that are added at each iteration. Blue crosses correspond to detected corners.

Adding a primitive to the smaller set does not decrease the coverage of pixels, corners, and alignments. The intuition behind the last inequality is that adding a primitive to a larger set of primitives will result in more redundancy. Summing over these inequalities proves that the inequality 3 holds. □

The objective function $\mathcal{F}$ in Equation 2 is also monotone until all the textual mentions in the text are covered (adding new primitives does not decrease the value of the objective function).

The objective function (Equation 2) is monotone and submodular. This means that there exist a greedy method that finds a $(1 - 1/e)$-approximation of the global optimum (Nemhauser, Wolsey, and Fisher 1978; Sviridenko 2004). In the next section we explain the greedy method to identify primitives and alignments.

5 Method

Figure 2 explains the steps in our method G-ALIGNER for diagram understanding. Submodularity of the objective function implies that we can introduce the following iterative greedy method with proven bounds. We first initialize the set of possible primitives (Section 5.1, Step 1 in Figure 2) and then iteratively add the primitives that maximize gain (Section 5.2, Step 2 in Figure 2). Figure 3 schematically depicts steps of G-ALIGNER.

5.1 Initialization

The left image in Figure 3 shows an example of initial sets of primitives from which our method starts.

Initialize primitives: For noise removal, we apply a weak Gaussian blur on the raw image and then binarize it using Otsu's method (Otsu 1975). We then use Hough transform (Stockman and Shapiro 2001) to extract primitives (line and circle segments) for a given diagram. The result of Hough transform has no information about the start and end points of the lines or arcs. Only the parametric representation of the primitive is known. Therefore, post-processing is required in order to determine endpoints. We detect and connect continuous binary points that lie on the same line or arc. For each primitive of interest, the result will be a few independent segments where the start and end of the segments are stored.

Standard application of Hough transform is not applicable to our problem. This is mainly due to (1) inaccuracies at the intersections (2) confusions between circles and polygons composed of several small lines; and (3) sensitivity to parameters of Hough transform and all post processing techniques. Our experimental evaluations show that there is no single set of parameters that work well on large number of samples. To overcome these issues, we set the threshold to a low number to over-generate a large number of primitives. This way, the right set of primitives are most likely among the over-generated set of primitives. We typically obtain 20 to 60 primitives in each diagram. Figure 3 shows an example of overproduced primitives L. We then use the optimization in Equation 2 to select the right set of primitives $\hat{L}$ from a big pool of noisy estimates of primitives L.

Initialize Corners: To enforce coherent visual elements, we need to encourage the set of selected primitives to be visually coherent. We use corners in diagrams as a gluing function. Two primitives that share endpoints very close to a corner in an image are preferred to primitives that are far away from corners. We use Harris Corner detectors (Harris and Stephens 1988) to identify possible locations of corners. Corners are scored based on how close they are to primitives.

Initialize mentions: We extract mentions from textual descriptions by keyword search using the list of visual elements in V.

5.2 Iterative Optimization

We initialize the optimal set of primitives as an empty set $\hat{L} = \emptyset$. Then we repeat the following step. At every iteration $k+1$, we select the primitive l that maximizes the following equation given that $\hat{L}^k$ is the best subset at iteration k.

$$\hat{L}^{k+1} = \arg \max_{l \in L \setminus \hat{L}^k} \mathcal{F}(\hat{L}^k \cup l) - \mathcal{F}(\hat{L}^k) \qquad (4)$$

Figure 3 shows three steps of this greedy method on a sample diagram.

6 Experiments

To experimentally evaluate our method we build a dataset of geometry questions along with annotations about visual elements and alignments. We test our method on how well it can identify visual elements and how accurate are the alignments established by our method. We compare our method G-ALIGNER with baselines that use standard techniques of diagram understanding. To also better understand our model we perform ablation studies for our model.

6.1 Experimental Setup

Dataset We build a dataset of high school plane geometry questions where every question has a textual description in English accompanied by a diagram. For evaluation purposes, we annotate diagrams with correct visual elements as well as alignments between textual mentions and visual elements.

Questions are compiled from four test preparation websites (RegentsPrepCenter; EdHelper; SATMath; SATPractice) for high school geometry. Ground truth labels are collected by manually annotating all the primitives in the diagram. In addition, we annotate all the alignments between textual mentions and visual elements in the diagram. In total, our dataset consists of 100 questions and 482 ground truth alignments. The dataset is publicly available in our project web page.

Tasks and metrics We evaluate our method G-ALIGNER in two tasks of (1) identifying primitives in diagrams and (2) aligning visual elements with textual mentions.

For task 1, we compare detected primitives by G-ALIGNER against the ground truth dataset. For every identified primitive l' and the corresponding ground truth primitive l, we measure the amount of overlap. If the primitive is a line segment, we project l' onto l and measure the ratio of area of intersection over the union. For arc primitives we use the convex area and measure the amount of area of intersection over union. A detection l' is considered as a correct prediction if the amount of intersection over union is bigger than a threshold α (called overlap-to-ground-truth ratio). For evaluation purposes, we vary $\alpha \in [0.7 : 1]$ and report F1 scores. Precision is the number of correctly identified primitives divided by total number of identified primitives. Recall is the number of correctly identified primitives divided by the number of primitives in ground truth.

For task 2, we evaluate if G-ALIGNER correctly aligns a textual mention with a visual element. For that, we establish alignments and report the accuracy of G-ALIGNER compared to ground truth annotations.

Baselines We compare our method G-ALIGNER with a baseline that uses Hough transform for identifying primitives. Similar to almost all of the Hough-based methods, this baseline requires a set of sensitive parameters: two thresholds for picking elements in Hough line and circle space, respectively, and three non-maximum suppression neighborhood parameters (two for line and one for circle). To find the best possible baseline, we perform a sampled grid search. This way, we can find the best set of parameters as if we have tuned the parameters on the test set. This baseline works as an upper bound on how well one can detect visual primitives using standard Hough-based methods. The overall distribution of F1 for several samples of parameters is shown in 5.

For task 1, the baseline identifies primitives that score higher than a threshold. This threshold is manually set to produce the best possible detection in our dataset. For task 2, we use identified primitives from task 1 and align them with the mention corresponding to the closest visual element.

Parameters Our method G-ALIGNER also uses Hough transform to extract initial primitives out of diagrams. However, G-ALIGNER method is not sensitive to the choice of parameters in Hough transform. We set the parameters so that we always overly generate primitives. Our optimization method reasons about what primitives to select.

6.2 Results

Identifying primitives We report the performance of G-ALIGNER in identifying primitives and compare it with the best baseline explained above. Figure 4 shows the $F1$ score at different overlap-to-the-ground-truth ratios. G-ALIGNER significantly outperforms the baseline because (1) G-ALIGNER couples visual and textual elements (2) G-ALIGNER enforces diagram coherency (3) G-ALIGNER does not require parameter tuning. The baseline typically maintains relatively high recall but low precision. For example, at $\alpha = 0.8$ the baseline achieves precision of 69% at the recall of 86% compared to our precision of 95% at the recall of 93%.

Figure 5 reports the distribution of the F1 scores for the baseline for 500 samples of parameters. This figure also shows where the "best baseline" (whose parameters we manually tuned for the entire dataset) and

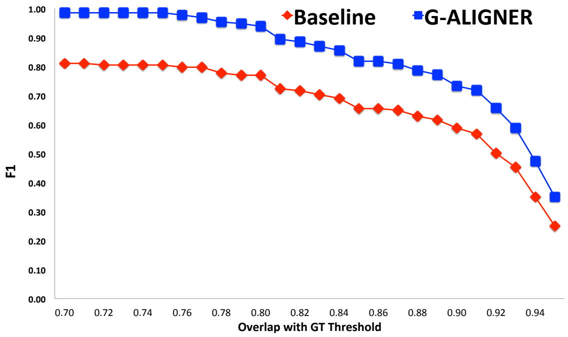


Figure 4: Comparison between G-ALIGNER and the baseline in task 1 in terms of F1 by varying overlap to the ground truth ratio α. This threshold is used to evaluate correct predictions of primitives.

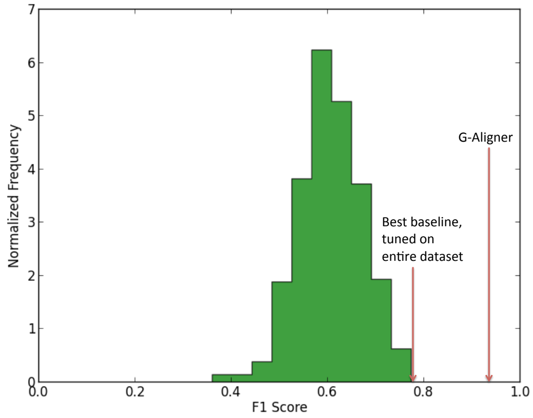
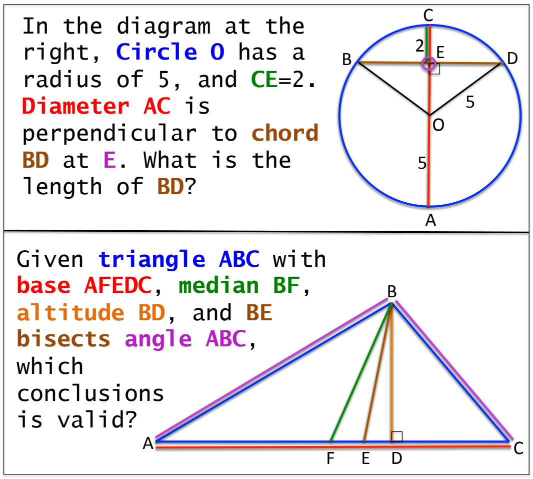


Figure 5: Normalized histogram of 500 F1 scores for the baseline obtained by randomly chosen parameters. We observe a normal distribution centered at 0.5 with standard deviation 0.04. The F1 scores of best baseline parameters and G-ALIGNER also drawn.

model		F1	Precision	Recall
G-ALIGNER	$(\mathcal{P}+\mathcal{S}+\mathcal{C})$	**0.94**	**0.95**	0.93
No $\mathcal{C}$	$(\mathcal{P}+\mathcal{S})$	0.93	0.93	0.93
No $\mathcal{S}$	$(\mathcal{P}+\mathcal{C})$	0.89	0.85	0.93
No $\mathcal{S},\mathcal{C}$	$(\mathcal{P})$	0.83	0.76	0.93
baseline		0.77	0.69	0.86

Table 1: Ablation study on Task 1 (identifying primitives).



In the diagram at the right, Circle O has a radius of 5, and CE=2. Diameter AC is perpendicular to chord BD at E. What is the length of BD?

Given triangle ABC with base AFEDC, median BF, altitude BD, and BE bisects angle ABC, which conclusions is valid?

Figure 6: Examples of alignments produced by G-ALIGNER. Textual mentions and visual elements are color coded. This figure is best viewed in color.

G-ALIGNER stand. The comparison clearly demonstrates how G-ALIGNER outperforms the baseline with any combination of parameters.

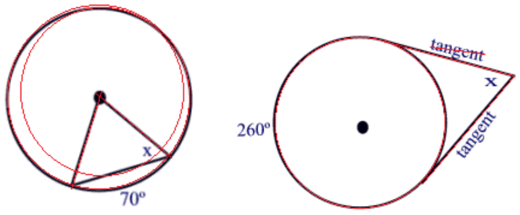
Ablation study We study the importance of each component in G-ALIGNER (optimization Equation 2). To study the effect of enforcing agreement between visual elements and textual mentions, we remove the term $\mathcal{S}$ from the optimization in Equation 2. In addition, to study the effect of enforcing diagram coherency we remove the term $\mathcal{C}$ from the optimization in Equation 2. We also show the effects of removing both $\mathcal{S}, \mathcal{C}$ from the equation. Table 1 shows the precision, recall, and F1 scores of identifying primitives for $\alpha = 0.8$. Removing both components decreases both precision and recall. The effect of removing the $\mathcal{S}$ is higher than that of $\mathcal{C}$ implying the importance of coupling textual and visual data.

Aligning textual mentions and visual elements To study the performance of G-ALIGNER in aligning textual mentions and visual elements, we compare our alignment results with baseline alignments. G-ALIGNER achieves an accuracy of 90% and baseline obtains the accuracy of 64% for the overlap ratio of $\alpha = 0.8$. This approves that coupling textual and visual information provides a better alignment.

Qualitative results Figure 6 shows examples of alignments between visual elements and textual mentions produced by G-ALIGNER . Mentions and their corresponding visual elements are color coded. For example, G-ALIGNER establishes an alignment between the textual mention `base AFEDC` and the corresponding line (*red line*) in Figure 6.

To show the effects of $\mathcal{S}, \mathcal{C}$ in G-ALIGNER, Figure 7 shows examples of mistakes in primitive identification that happens when either $\mathcal{S}$ or $\mathcal{C}$ are removed from G-ALIGNER. In Figure 7, black pixels correspond to the actual diagram and red lines and circles correspond to the detected elements. Removing $\mathcal{S}$ in (a) results in wrong detection of an extra circle. Without $\mathcal{S}$, G-ALIGNER does not know what to expect and therefore picks a circle whose coverage is larger than the rest of the lines in the diagram. Removing $\mathcal{C}$ in (b) results in estimates of an incorrect line on the pixels on the word *tangent* in the diagram. By considering agreements between corners, G-ALIGNER correctly discards this false detection. Note that, one might come up with heuristics to avoid any of these specific cases. However, this paper provides a unified method that reasons about diagrams and can handle these cases without any specific heuristic.

Limitations Our method shows promising results in our experiments. However, there are cases in which our model fails to identify diagram elements. The benefits of our model over the standard baselines is marginal if the text of the ques-



(a) removing $\mathcal{S}$ (b) removing $\mathcal{C}$

Figure 7: Examples of mistakes when $\mathcal{S}$ or $\mathcal{C}$ are removed from G-Aligner. In this figure, black pixels correspond to the actual pixels in the diagram and red lines or circles are detected elements. In (a), removing $\mathcal{S}$ results in adding a wrong detection of an extra circle whose coverage is actually bigger than some of the correct lines in the figure. In (b), removing $\mathcal{C}$ results in an incorrect detection of an incorrect line on the word *tangent*. G-Aligner correctly understands both of the diagrams above.

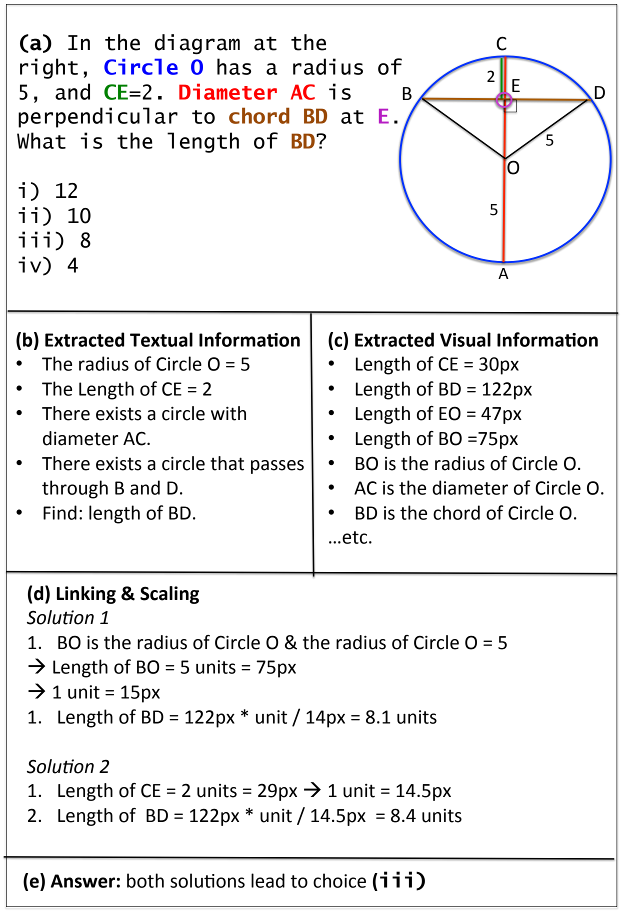
tion does not mention any of diagram elements. Our method doesn't recognize out of the vocabulary visual elements and fails if the scale of one visual elements is out of the range.

7 Conclusion and Future Directions

Our ultimate goal is to build an automated system that can solve geometry questions. The very first step toward this goal is to build a method that can understand diagrams. To our surprise, despite a large body of work in diagram understanding, the literature lacks a unified framework that does not require parameter tuning and problem specific heuristics. This paper is one step toward building such systems. We introduce G-Aligner that understands diagrams by coupling primitive detection and their alignments to text. The output of G-Aligner is a set of detected visual elements, their location, their geometric properties, and their corresponding textual mentions. Further, G-Aligner can exhaustively enumerate all possible visual elements (even the ones that are not explicitly mentioned in the text).

A direct extension of G-Aligner allows us to solve geometry problems with drawn-to-scale diagrams, through a *scaling* method. For example, G-Aligner identifies several visual primitives (i.e., lines BD, BO, DO and CA and circle O) in the problem in Figure 8. Additionally, G-Aligner can enumerate all possible visual elements, their geometric properties, and their geometric relations (Figure 8 (c)). For instance, G-Aligner identifies line CE and its length in pixels. Moreover, G-Aligner can capture geometric relations. For example, G-Aligner identifies line BO as the radius of the circle O using the visual information that one end of the line BO is near the center of the circle O and the other end of the line is on the circumference of the circle.

With simple textual processing, we also extract numerical relations from the question text and what the question is looking for (Figure 8 (b)). Textual information for the example question includes (1) the radius of circle O = 5, (2) line CE = 2, and (3) the question is looking for the length of line BD. By combining textual information (1) with the extracted visual information that states "BO is the radius of Circle O", we infer that the length of line BO = 5.



(a) In the diagram at the right, Circle O has a radius of 5, and CE=2. Diameter AC is perpendicular to chord BD at E. What is the length of BD?

i) 12
ii) 10
iii) 8
iv) 4

(b) Extracted Textual Information
- The radius of Circle O = 5
- The Length of CE = 2
- There exists a circle with diameter AC.
- There exists a circle that passes through B and D.
- Find: length of BD.

(c) Extracted Visual Information
- Length of CE = 30px
- Length of BD = 122px
- Length of EO = 47px
- Length of BO =75px
- BO is the radius of Circle O.
- AC is the diameter of Circle O.
- BD is the chord of Circle O.

...etc.

(d) Linking & Scaling

Solution 1

1. BO is the radius of Circle O & the radius of Circle O = 5
→ Length of BO = 5 units = 75px
→ 1 unit = 15px
1. Length of BD = 122px * unit / 14px = 8.1 units

Solution 2

1. Length of CE = 2 units = 29px → 1 unit = 14.5px
2. Length of BD = 122px * unit / 14.5px = 8.4 units

(e) Answer: both solutions lead to choice **(iii)**

Figure 8: Using G-Aligner to solve a geometry problem.

This enables us to compute the scale between the units in the question (BO=5) and pixels in the diagram (Length of line BO is 75 pixels). Using textual information (2), we can solve the example problem in a slightly different way (See Figure 8 (d) for step-by-step demonstration of how G-Aligner can solve this problem).

G-Aligner, with the *scaling*, finds correct answers for problems with well-drawn diagrams. In future, we plan to solve the problems using more complex mathematical and logical reasoning for geometry theorem proving. We also plan to link several uncertain visual and textual relations to formulate a collective probabilistic model that will output the most probable answer to the question. In addition, we intend to use the results of diagram understanding to help understand the semantics of sentences. This is feasible because diagrams are easier to understand compared to real images. Diagram understanding should also help recognition of out-of-vocabulary visual elements and disambiguation of textual mentions. The current formulation can handle extensions to diagrams that are composed of well-defined primitives. We also plan to extend G-Aligner to understand more complex diagrams in other science areas. The demo and the dataset are available on our project page.

Acknowledgements

The research was supported by the Allen Institute for AI, and grants from the NSF (IIS-1352249) and UW-RRF (65-2775). We thank Santosh Divvala and the anonymous reviewers for helpful comments and the feedback on the work.

References

Bulko, W. C. 1988. Understanding text with an accompanying diagram. In *Proceedings of the 1st International Conference on Industrial and Engineering Applications of Artificial Intelligence and Expert Systems - Volume 2*, IEA/AIE '88, 894–898. New York, NY, USA: ACM.

Climer, S., and Bhatia, S. K. 2003. Local lines: A linear time line detector. *Pattern Recogn. Lett.* 24(14):2291–2300.

EdHelper. http://edhelper.com.

Farhadi, A.; Hejrati, M.; Sadeghi, M. A.; Young, P.; Rashtchian, C.; Hockenmaier, J.; and Forsyth, D. 2010. Every picture tells a story: Generating sentences from images. In *Proceedings of the 11th European Conference on Computer Vision*, 15–29.

Feigenbaum, E., and Feldman, J., eds. 1963. *Computers and Thought*. New York: McGraw Hill.

Ferguson, R. W., and Forbus, K. D. 1998. Telling juxtapositions: Using repetition and alignable difference in diagram understanding. *Advances in analogy research* 109–117.

Ferguson, R. W., and Forbus, K. D. 2000. Georep: A flexible tool for spatial representation of line drawings. In *Association for the Advancement of Artificial Intelligence (AAAI)*, 510–516.

Gupta, S., and Mooney, R. J. 2010. Using closed captions as supervision for video activity recognition. In *Proceedings of the Twenty-Fourth Conference on Artificial Intelligence (AAAI-2010)*, 1083–1088.

Harris, C., and Stephens, M. 1988. A combined corner and edge detector. In *In Proc. of Fourth Alvey Vision Conference*, 147–151.

Hegarty, M., and Just, M. A. 1989. 10 understanding machines from text and diagrams. *Knowledge acquisition from text and pictures* 171.

Jung, C. R., and Schramm, R. 2004. Rectangle detection based on a windowed hough transform. In *Proceedings of the Computer Graphics and Image Processing, XVII Brazilian Symposium*, SIBGRAPI '04, 113–120. Washington, DC, USA: IEEE Computer Society.

Kulkarni, G.; Premraj, V.; Dhar, S.; Li, S.; Choi, Y.; Berg, A. C.; and Berg, T. L. 2011. Baby talk: Understanding and generating image descriptions. In *Proceedings of the 24th CVPR*.

Li, K.; Lu, X.; Ling, H.; Liu, L.; Feng, T.; and Tang, Z. 2013. Detection of overlapped quadrangles in plane geometric figures. In *ICDAR*, 260–264. IEEE.

Lin, C., and Nevatia, R. 1998. Building detection and description from a single intensity image.

Lin, X.; Shimotsuji, S.; Minoh, M.; and Sakai, T. 1985. Efficient diagram understanding with characteristic pattern detection. *Computer vision, graphics, and image processing* 30(1):84–106.

Moon, H.; Chellappa, R.; and Rosenfeld, A. 2002. Optimal edge-based shape detection. *IEEE Transactions on Image Processing* 11(11):1209–1227.

Nemhauser, G. L.; Wolsey, L. A.; and Fisher, M. L. 1978. An analysis of approximations for maximizing submodular set functionsI. *Mathematical Programming* 14:265–294.

Novak, G. S., and Bulko, W. C. 1990. In Shrobe, H. E.; Dietterich, T. G.; and Swartout, W. R., eds., *Association for the Advancement of Artificial Intelligence (AAAI)*, 465–470. AAAI Press / The MIT Press.

Novak, G. 1995. Diagrams for solving physical problems. *Diagrammatic reasoning: Cognitive and computational perspectives* 753–774.

O'Gorman, L., and Kasturi, R. 1995. *Document image analysis*, volume 39. Citeseer.

Otsu, N. 1975. A threshold selection method from gray-level histograms. *Automatica* 11(285-296):23–27.

RegentsPrepCenter. High school geometry: Regents prep center geometry
http://www.regentsprep.org/regents/math/geometry/math-geometry.htm.

SATMath. Sat math: Sat math problem solving: Practice tests and explanations
http://www.majortests.com/sat/.

SATPractice. Sat practice questions: Geometry
http://www.onlinemathlearning.com.

Srihari, R. K. 1994. Computational models for integrating linguistic and visual information: A survey. *Artificial Intelligence Review* 8(5-6):349–369.

Stockman, G., and Shapiro, L. G. 2001. *Computer Vision*. Upper Saddle River, NJ, USA: Prentice Hall PTR, 1st edition.

Sviridenko, M. 2004. A note on maximizing a submodular set function subject to a knapsack constraint. *Operations Research Letters* 32(1):41 – 43.

Zhang, X., and Sun, F. 2011. Circle text expansion as low-rank textures. In *ICDAR*, 202–206. IEEE.