Anaphora without syntax - A Multi-lingual Approach for Geometry Constructions

Pankaj Prateek Kewalramani
Dept. of Comp. Sc. & Engi.
IIT, Kanpur, UP, India
pratikkr@iitk.ac.in

Jeetesh Mangwani
Dept. of Comp. Sc. & Engi.
IIT, Kanpur, UP, India
jeeteshm@iitk.ac.in

Amey Karkare
Dept. of Comp. Sc. & Engi.
IIT, Kanpur, UP, India
karkare@cse.iitk.ac.in

Sumit Gulwani
Microsoft Research, Redmond
sumitg@microsoft.com

Amitabha Mukerjee
Dept. of Comp. Sc. & Engi.
IIT, Kanpur, UP, India
amit@cse.iitk.ac.in

Abstract

In this work, we follow a text driven approach to anaphora, for interpreting text and drawing geometric diagrams. To make this approach suitable for resource poor languages, we propose creating a small training corpus by mapping raw text to a formal (postfix) *bridge* language. Now, an alignment approach can discover maps for nearly all the units and phrases. After each sentence, the bridge language mappings are used to draw the desired figure, given what has been drawn earlier. Anaphora are disambiguated in the task context. The system is language agnostic and we attempt it on textbooks in English and Hindi, and show that we obtain complete anaphora coverage, though several elliptic references are missed. Whole sentence interpretation is correct for 95.3% cases (English) or 92.7% (Hindi), on the NCERT class VI textbooks.

1 Introduction

"With center as Q, draw an arc cutting the arc drawn earlier at R". Here, *arc drawn earlier* is a phrasal anaphora which is harder to handle than pronominal anaphora such as *it, these, each other*.

Traditional approaches to handling anaphora have been either purely syntax-driven (Hobbs, 1978), (Lappin and Leass, 1994), or based on semantics that ultimately derives from the parse (Mitkov, 2014).

In this paper, we emphasize the semantic considerations and context awareness that pervades anaphora interpretation. A valid interpretation of a geometric text is that which permits the construction to be carried out. For example, the sentence *"Draw any two of its chord"* involves finding an existing object which can have two chords. The context is defined by a set of already existing geometric objects and their relations.

Instead of a truth-valued semantics (eg DRF), we use a set of procedures which can be applied to a small set of objects (points, lines, arcs etc.). More complex objects like triangles and quadrilaterals are not modelled explicitly.

1.1 Geometry Problems in a Multi-lingual Scenario

Students in India study in 22 languages. Worldwide, fewer than 10 of the 600+ languages in wide use are resource-rich in the sense of having Penn-tree banks and other large-sized standardised corpora. Therefore, language independent processes become more important. An important observation is that fixed grammars, POS tags etc. used in traditional parsers are not always the best way to break up a language. Taking a note of these shortcomings, we take a statistical approach to learn the mapping directly to the drawing task. Our long-term goal is to follow the instructions for geometric construction, and execute these step by step through a proof, thus guiding the student. In this paper, we focus on interpreting the NL input at various stages in understanding and solving geometry problems.

Since we wish to be language independent, we use no rich language models such as a parser or Wordnet. All language structures are learned from the small set of annotated NL input.

To establish whether a given interpretation can be executed as a drawing, we develop a plotter for mapping the bridge language to a set of plottable objects.

Problem:	
1. Construct a triangle ABC, given that AB = 5 cm, BC = 6 cm and AC = 7 cm.	1. एक त्रिभुज ABC की रचना कीजिए जबकि AB = 5 सेमी, BC = 6 सेमी और AC = 7 सेमी दिया है
Solution:	
1. Draw a line segment BC of length 6 cm.	1. 6 सेमी लंबाई का एक रेखाखंड BC खींचिए
2. With B as center, draw an arc of radius 5 cm.	2. B को केंद्र मानकर और 5 सेमी त्रिज्या लेकर एक चाप खींचिए
3. With C as center, draw an arc of radius 7 cm.	3. C को केंद्र मानकर और 7 सेमी त्रिज्या लेकर एक चाप खींचिए
4. Mark an intersection point of these arcs as A.	4. इन चापों के प्रतिच्छेद बिंदु को A से अंकित कीजिए
5. Join AB.	5. AB को जोड़िए
6. Join AC.	6. AC को जोड़िए

Table 1: The same problem, in English and Hindi. We consider this problem as our running example. Sentence 1 from both languages is analyzed in depth in section 3.1

1.2 Objective

To design and implement an interpreter for geometric construction sentences that is language-independent (demonstrated here for English and Hindi). The system receives steps for a geometric construction as input (e.g. *"With center B and radius 5cm, draw an arc cutting the previously drawn arc at C"*, केंद्र B और त्रिज्या 5 सेमी लेकर एक चाप खींचिए जो पहले खींची चाप को C पर काटता हो etc), handles anaphora and also some ellipsis in the task context (*"Construct a line segment AB such that the length of AB is equal to the difference of length of CD and EF"*) and outputs the geometric figure as instructed.

An example and its solution from CBSE NCERT Mathematics for VII standard is shown in Table 1. The figure expected out of the construction step is shown in Figure 1.

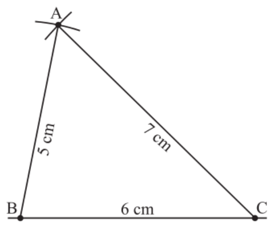

Figure 1: Sample Problem Figure

2 Related work

2.1 Anaphora

Traditional approaches to anaphora are satisfied with identifying the co-reference - i.e. the other part of the text that also refers to the same object. Much work in this area uses syntactic models and/or formal semantics (Rahman and Ng, 2011), (Nguyen and Poesio, 2012), (Stoyanov and Eisneree, 2012), (Raghunathan et al., 2010). All these approaches map anaphora to other parts of discourse and do not ground it or attempt it in a task context. In the areas of geometric problem solving, many approaches working on hand-crafted formalisations focussing on the logical/planning aspects have been suggested (Gulwani et al., 2011) (Schreck et al., 2012) (Itzhaky et al., 2012).

There has been some work on solving geometry construction problems from language. (Ahmed et al., 2012) look more into using domain-specific measures to minimize parser errors and augment the geometry problem solver, *GeoSynth*. (Mukherjee and Garain, 2008) attempts a similar problem, but is focused on problems in English, and use parsing and other resources along with the extensive knowledge-base, *GeometryNet*, to generate diagrams from text. Unlike any of this work, we use handle natural language input, but without any language-specific resources.

3 System Design and Bridge Language

The key steps in the system design are outlined in figure 2.

The bridge language was designed to capture the structure of the imperative geometric instructions. The bridge grammar and plottability to-

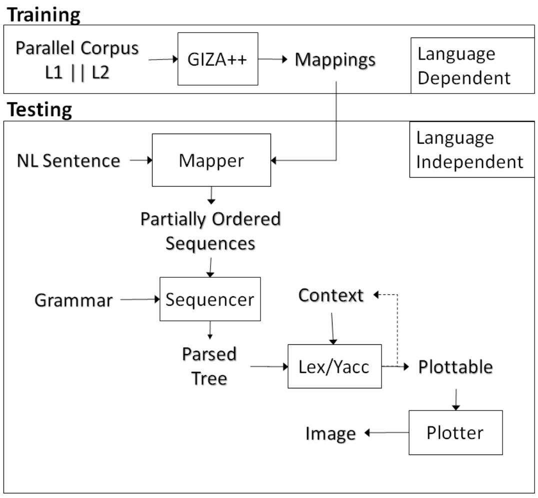

Figure 2: In learning, we learn a set of language-specific Mappings via bridge language mapping corpus. These mappings are used to interpret test input, producing partially ordered strings in the bridge language. These are then reordered in a sequencer, subject to type matching and parameter proximity constraints. This re-orders the string according to the grammar of the bridge language. This Bridge parse is then analyzed in the given context in order to a) identify parameters for the operations, b) disambiguate anaphora and ellipsis, c) generate the plotter commands and d) update the context.

English	Hindi	Bridge Language
Construct a line AB of length 4 cm	4 सेमी लम्बाई का एक रेखाखण्ड AB खींचिए	`construct lineSegment AB length 4 cm`
With A as center and radius 3 cm, draw an arc	केंद्र A और त्रिज्या 3 सेमी लेकर एक चाप खींचिए	`construct arc center A radius 3 cm`
With B as center and radius 5 cm, draw an arc cutting the previously drawn arc at C	केंद्र B और त्रिज्या 5 सेमी लेकर एक चाप खींचिए जो पहले खींची चाप को C काटता हो	`construct intersectingArc center C radius 5 cm cuts arc previous at C`

Table 2: Sample corpus with the same problem in English, Hindi and correct bridge language representation

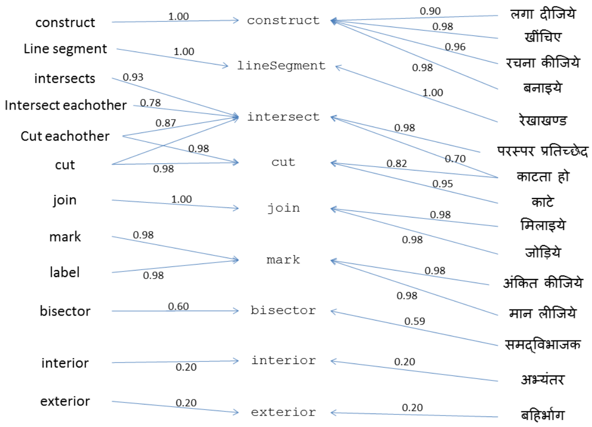

Figure 3: Sample alignment. English (left) and Hindi (right) are aligned with units from the bridge-language (middle).

gether impose a type constraint on the interpretation. Designing a bridge language is quite a challenging task since it captures the variability of all instructions appearing in geometric data, and follows a formal horn clause grammar. Our present bridge syntax is close to English syntax but also works with Hindi (see fig. 3). Having a good mapping from the natural language is an important step for minimizing re-sequencing search and other backtracking later.

One assumption here is *proximity* - i.e. the parameters of the object to be plotted occur near to parameter name in the input sentence. Hence the rules have been designed such that for each production rule, the head (leftmost non-terminal) guides the search for the remaining arguments. The system is implicitly type-delimited - types include objects (line, arc etc.) or attributes (markable, constructible). For example, in *Mark a point A on it*, "it" can be a line, arc etc. but not a point since point is not markable. Also, point pairs like AB would define a line segment or a ray, ABC would define an arc, 'l' would define a line.

3.1 Mapper and Sequencer

We learn the mapping from NL input in a particular language to the postfix bridge language by using a training set. The bridge language has a postfix syntax, and we use GIZA++ cross-lingual alignment (Och and Ney, 2003) to obtain an unordered alignment of units (Fig. 3).

Consider the English instruction, *"Draw a line segment BC of length 6 cm."* (sentence E1 in table 1), we have `construct line segment BC length 6 cm` as corresponding bridge language, whereas for the equivalent in Hindi, *"6 se.mi lambAi kA ek rekhAkhaND BC khinchiye."* (sentence H1), we have `6 cm length line segment BC construct`. The sequencer takes these partially ordered set of words and searches through possible reorderings to map it to a sentence acceptable as a bridge grammar terminal subject to type matching. Given any production rule, our DFS-based algorithm recursively expands the leftmost unexpanded nonterminal; on encountering a terminal, trying to fit new non-terminals to some token near its parent's/left sibling's token. In sentence 1 using our assumption we start searching for an object to be constructed near the "construct" word; the "line segment AB" is found. The object may appear on either side of a keyword (as seen in sentence 1 in the Hindi solution in table 1). This heuristic reduces search immensely.

A more complex example is seen in *"Construct a line segment AB such that length of AB is equal to the difference of the length of CD and EF"*. Here, only those parses which demand the construction of AB are retained (as AB is nearer to *construct* than CD and EF); other possibilities (construction of CD and EF) are pruned. Ellipsis ("length of" CD and EF) is handled correctly as there is no other interpretation possible that will provide the length parameter for drawing AB.

3.1.1 Semantics analyzer

At this stage, the system has a bridge language string. However, anaphora have not been removed, since the bridge language retains terms such as *"it", "these", "arc drawn earlier"*. The system maintains the current geometric "context" in terms of the set of plottables already drawn (points, lines, circle, line segments, arcs etc.). The parser now uses Lex-Yacc to generate a "diff" set of plottables which need to be drawn now. All coordinates, radii, lengths and angles, as well as anaphoric referents, are resolved in this step.

In the running example (table 1), after the first step the generated "diff" list would be `Points=`$\{$`B(0,0), C(6,0)`$\}$`, line segments=`$\{$`BC`$\}$

Similarly before the fifth step, the "context" would be: `points=`$\{$`B(0,0), C(6,0), A(1,`$\sqrt{24}$`)`$\}$`, line segments=`$\{$`BC`$\}$`, arcs=`$\{$`(B,5), (C,7)`$\}$ and the "diff" would be `line segments=`$\{$`AB`$\}$

The updated "context" would just append a new line segment "AB" to the context.

3.1.2 Plotter

For each input construction step, the plotter receives a list of primitive objects from the semantic analyzer and plots them on the canvas.

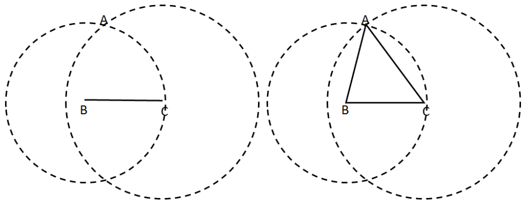

Figure 4: Plot obtained after a) the first four steps, b) all the steps of table 1 problem; the dotted circles represent arcs

3.2 Handling Anaphora: Example

Consider the sentence *With B and C as centers and radii 5 and 7 cm, draw two arcs intersecting each other at A*. Here, "each other" is the anaphora, referring to the arcs being drawn. The arcs have not been constructed yet, and are not in the context but the semantic analyzer identifies them from the object appearing in the "diff" structure.

As another example, table 1 step 4, the words *"these arcs"* guide us to fetch the last drawn arcs. The "intersection point" applied on "these arcs" yields us a markable object, which when combined with *"mark"* completes the parse.

By the fourth step, we obtain the plot in figure 4a.

After the sixth and final step, we obtain the plot in figure 4b.

4 Corpus Collection and Results

Our corpus contains 360 sentences in Hindi and English, collected from geometry construction problems, in NCERT textbooks of class 6^{th} to 9^{th}. Bridge language mappings were generated manually, a training subset from which was used to train the cross lingual alignment mappings.

The corpus is available at (Kewalramani and Mangwani, 2014) [Unique Tokens: English: 181; Hindi: 169; Bridge Language: 110]

We note here that our aim in this work was to handle very primitive objects of geometric constructions, and to validate a low knowledge multilingual model. Thus, we have not modelled conceptual hierarchies such as triangles, hypotenuse, parallelogram etc. Aso, angles to be drawn using ruler and compass, such as 75 degrees or 22.5 degrees have not been implemented. Constructs such as '='(equals) or '+'(sum) have not been implemented (eg. *Draw AC such that AC = AX + YB*). This level of modelling corresponds to roughly a class VI standard NCERT text.

However, we find that the system also has respectable parse percentages, so we report these as

Standard/ Grade	Corpus Size	Number of successful parses	Percentage
English			
6^{th}	86	82	95.34%
6^{th} to 9^{th}	225	190	84.44%
Hindi			
6^{th}	96	89	92.7%
6^{th} to 9^{th}	250	185	74%

Table 3: Results

well (table 3)

We need to provide for unspecified radii, lengths and angle measures e.g. *"With A and B as centers and a suitable radius, draw two arcs intersecting each other at point C"*.
Radii, lengths etc. are supplied randomly in such cases, obeying the constraints which might have been put on such values in this or the previous step (e.g. *"Construct a line segment with length greater than length of AB"*). However, we note that in general findings such solutions is non-trivial in a planning point of view (Schreck et al., 2012)

4.1 Failure Cases

Whole sentence interpretation on the class VI test set was 82 out of 86 (English) and 89 of 96 (Hindi). In most of these sentences (e.g. sentence 5 in table 4), the anaphora interpretation is correct, but the sentence is not being handled due to the entity *"chord"* which has not been implemented in the semantic analyzer.

Phrasal anaphora, e.g. *"With center at Q, draw an arc to cut the arc drawn earlier at R"*, is being handled since *"earlier"* and *"previous"* are both mapped to \`\`previous" in the bridge language and are handled by the semantic analyzer.

In addition to anaphora, some ellipsis are also handled. In the sentence, *"Construct PQ such that the length of PQ is twice that of AB"*, the ellipsis "length of" AB is handled at the bridge parsing stage, as there is no other scenario possible.

However, in sentence 2 in table 4, the system draws both A and B on the same line since arms of an angle have not been implemented.

Sentence 1 in the table 4 is an example of a ellipsis failure. The perpendicular bisector of OB not being handled. The system would also fail for sentences such as *Join AB and BC* - but such sentences do not occur in the class 6^{th} corpus.

English	Hindi
Draw the perpendicular bisectors of OA and OB (1x)	OA और OB का समद्विभाजक खींचिए (2x)
Take a point A on one of its arms and B on the other(1x)	
With A and B as center, and 4 and 5 cm as radius, draw two arcs intersecting each other at C (1x)	A और B को केंद्र मानकर और 4 और 5 सेमी त्रिज्या लेकर दो चाप खींचिए जो परस्पर C पर काटती हो (1x)
With A and B as centers and a radius greater than AP, construct two arcs which cut each other at Q (1x)	
Draw any two of its chords (1x)	इसकी दो जीह्वा खींचिए (1x)
Divide it into 4 parts (passes)	इसके चार भाग कीजिये (2x)

Table 4: Failure Cases: The number after the sentences indicate the number of similar structured sentences in the corpus

Sentence 3 has a parallel ellipsis: but due to "B" being linguistically near to *centers* while "4" is nearer to *radius*, the proximity heuristic results in aligning B with radius 4cm and A with 5cm. We observe that for *"With centers as A and B and radius as 4 cm and 5 cm, draw two arcs intersecting each other at C"*, out present heuristic captures the correct intent.

Heuristic failures are also encountered. In sentence 4, the mapper generates the bridge sentence to `cut a at Q`, meaning to cut a line 'a' at a point Q, or `construct AP`, both of which are incorrect parses.

Sentence 6 parses correctly in English but not in Hindi. This can be attributed to the lack of a good probabilistic map between Hindi and bridge language, which is constrained by the size of the corpus.

4.2 Conclusion

In this work we have considered the interpretation of language driven by the imperatives of a text to geometric construction task. We attempt this without language models such as parsers, wordnet etc. Rather than trying to construct formal models of interpretation, the system learns to infer plottables from a small domain-specific bridge-language annotated corpus. Despite these constraints, it is able to execute above 90% of unseen sentences from two very different languages.

One major lacuna in this work is its dependence on heuristics such as proximity. Knowledge relating to language models - e.g. that argument structures in Hindi are less determined by position than in English - is implicit in the learned mappings from text to bridge language. However, this knowledge is not modeled in the analysis. Such a model would be very important for the work to be generalized to new domains, and is one of our main goals in the immediate future.

Finally, the work also has potential as a part of an intelligent tutoring system. In using such a system for student input, a number of other aspects would need to be considered. Much of the limitation in the present system arises from incompleteness in the set of objects that can be handled by the drawing system. In order to evaluate its effectiveness on more informal, and possibly incorrect input, we would like to deploy the system online to enable students to interact with it.

However, the fact that this much can be achieved without using any syntactic or semantic prior knowledge, makes it language agnostic, as demonstrated here with a two-language implementation. This in itself is of direct value for many languages.

References

Umair Z Ahmed, Arpit Kumar, Monojit Choudhury, and Kalika Bali. 2012. Can modern statistical parsers lead to better natural language understanding for education? In *Computational Linguistics and Intelligent Text Processing*, pages 415–427. Springer.

Sumit Gulwani, Vijay Anand Korthikanti, and Ashish Tiwari. 2011. Synthesizing geometry constructions. In *ACM SIGPLAN Notices*, volume 46, pages 50–61. ACM.

Jerry R Hobbs. 1978. Resolving pronoun references. *Lingua*, 44(4):311–338.

Shachar Itzhaky, Sumit Gulwani, Neil Immerman, and Mooly Sagiv. 2012. Solving geometry problems using a combination of symbolic and numerical reasoning. Technical report, Technical report, Tel Aviv University.

Hans Kamp, Josef Van Genabith, and Uwe Reyle. 2011. Discourse representation theory. In *Handbook of philosophical logic*, pages 125–394. Springer.

Pankaj Prateek Kewalramani and Jeetesh Mangwani. 2014. Iitk annotated hindi-english geometry corpus (http://cse.iitk.ac.in/users/grounded-lang/pratikkr/).

Shalom Lappin and Herbert J Leass. 1994. An algorithm for pronominal anaphora resolution. *Computational linguistics*, 20(4):535–561.

Ruslan Mitkov. 2014. *Anaphora resolution*. Routledge.

Anirban Mukherjee and Utpal Garain. 2008. A review of methods for automatic understanding of natural language mathematical problems. *Artificial Intelligence Review*, 29(2):93–122.

Truc-Vien T. Nguyen and Massimo Poesio. 2012. Relational structures and models for coreference resolution. In *Proceedings of the 24rd International Conference on Computational Linguistics*, COLING '12, December.

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

Karthik Raghunathan, Heeyoung Lee, Sudarshan Rangarajan, Nathanael Chambers, Mihai Surdeanu, Dan Jurafsky, and Christopher Manning. 2010. A multipass sieve for coreference resolution. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, pages 492–501. Association for Computational Linguistics.

Altaf Rahman and Vincent Ng. 2011. Ensemble-based coreference resolution. In *Proceedings of the 22nd IJCAI Volume Three*, pages 1884–1889.

Pascal Schreck, Pascal Mathis, and Julien Narboux. 2012. Geometric construction problem solving in computer-aided learning. In *Tools with Artificial Intelligence (ICTAI), 2012 IEEE 24th International Conference on*, volume 1, pages 1139–1144. IEEE.

Veselin Stoyanov and Jason Eisneree. 2012. Easy-first coreference resolution. In *Proceedings 24rd COLING '12*, Dec.

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*.