Handling Anaphoras in multiple languages in the framework of Geometry Constructions

Jeetesh Mangwani & Pankaj Prateek
Advisor: Dr. Amitabh Mukherjee
Dept. of Computer Science and Engineering
Indian Institute of Technology, Kanpur, India
{jeeteshm, pratikkr, amit} @ cse.iitk.ac.in

April 17, 2014

ABSTRACT

In this project, we take up the problem of handling anaphoras in multiple languages in the context of a language-independent interpreter for drawing geometric diagrams. We focus on ruler and compass based construction problems. We start with use cases and motivations on why such a system would be useful and wh ere deploying it would be fruitful. We give a brief tabulated review summary of related research work done on geometry related problems and point out the common trait that they lack the ability to decipher any problem/constraint expressed in a natural language. We, then, move to describe the design of the interpreter and the usage of cross-lingual alignment technique to provide the ability of language-independent interpretation. We briefly mention how the alignment model is utilised to realize the powerful translation feature. This is followed by a detailed account on handling anaphoras while extracting semantics from the translated sentences. We conclude with comprehensive results, difficulties faced and future work.

1 Introduction

1.1 Anaphora

In the traditional literature, anaphoras have been modelled based on formal models for the situation. These include Hans Kemp's Discourse Representation Theory (DRT), and other similar formal models that build up a model for the situation within a suitable logical formalism, such as second-order logics that permit predicates to be abstracted as lambda expressions.

One serious shortcoming of such approaches is that they ignore the goals of the hearer, which strongly influence the anaphora resolution, since only the steps relevant to the goal would be interpreted. Thus, given the sentence such as

If an incendiary bomb drops next to you, don't loose your head. Put it in a bucket and cover it with sand. [11]

A context-aware anaphora resolution system would be hard pressed to argue that the proper antecedent for "it" is "incendiary bomb" and not "your head". At the same time, knowing that this sentence was printed in a pamphlet distributed in Britain during WWII, and assuming that the goal is to minimize bomb damage and that achieving this objective by covering one's head is implausible, one may identify the "it" with the "bomb": an association made quite effortlessly by most human readers.

The main innovation in the task undertaken here is the fact that the sentences which arise while trying to solve a geometry problem constitutes a task context. Thus, there the overarching goal for the discourse is that of solving the problem, and the immediate sub-objective is that of drawing a geometric figure that aids in the process. Thus, the geometric context is defined by a set of already existing geometric objects and their relations. If an interpretation for an anaphora or an ellipsis is such that it cannot be drawn, then it would not be valid. A more complex case is the situation where the construction is possible but if it fails to meet the goal of solving the problem. In the present work, this type of reasoning, which would require the process to be tied in to a logical theorem-prover, is also not being handled; as an example consider the sentence "`Construct a line l such that it makes an angle of 90 degrees at the points where it intersects the parallel line segments AB and CD`"; this construction is not always possible; our system does not deal with such proofs and constructions.

The earliest formalizations of semantics in discourse were driven by truth-value semantics: the speaker was seen to assert the truth of the resulting proposition. In DRT, Kamp and others suggested that the goal was more of encoding the "information" or encyclopedic world state that ensues. However, in many situations, the task constrains (or focuses attention onto) only a part of this world state; thus, task driven models of context are in general more efficient, and also more accurate, than mere informational models.

At the same time, since we wish to be language independent, we do not wish to use rich language models such as a parser etc. All language structures are learned from a small set of semantically tagged NL input.

1.2 Geometry Construction

Majority of students in India write in languages other than English: there are 22 scheduled languages in the Indian Constitution. Therefore, the language independent process discussed in the previous sections becomes a necessary step in handling the input entered by the students to the computer. Integrating this into a tutoring system would not only aid the slow and patient learners, but also reduce the cost of human tutors.

In order to save the stationery required in drawing geometric diagrams as well as to reduce dependence on complex graphics applications, we introduce an interpreter for diagram construction steps expressed in a suitable natural language. This not only simplifies the construction, but also leads to easily understood natural language 'programs'.

1.3 Objective

To design and implement an interpreter for geometric construction sentences that

- is language-independent (works for English, Hindi at present)
- Receives steps for a geometric construction as input e.g. `"Draw a line segment AB of length 4 cm"`, "Akd B aOr E/¿yA 5AsmFAlkr ek cAp KF"Ece jo phAl KF"cF cAp ko C pr kAVtA ho" etc
- Handles anaphoras using the task context
- Outputs the geometric figure obtained on executing the given sequence of steps

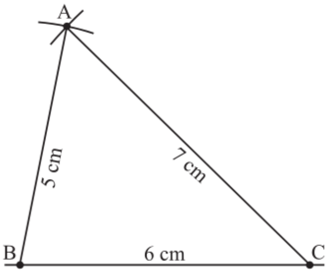
We consider the problem of interpreting sentences that may be generated by students answering questions related to high-school geometry. For instance, consider the following example and its solution (edited) from CBSE NCERT Mathematics for 7th standard:

Example 1:
`Construct a triangle ABC, given that AB = 5 cm, BC = 6 cm and AC = 7 cm.`
Solution

1. `Draw a line segment BC of length 6 cm.`
2. `With B as center, draw an arc of radius 5 cm.`
3. `With C as center, draw an arc of radius 7 cm.`
4. `Mark an intersection point of these arcs as A.`

5. `Join AB.`

6. `Join AC.`



Consider the same problem from the Hindi version of the same book:

udAhrZ 1
ek E/BJΓABC kF rcnA kFEje jbEk AB = 5ΛsmF, BC = 6ΛsmF aOr AC = 7ΛsmF EdyA
ħ

hl

1. 6ΛsmF "lbAI kA ekΛrKAK"X BC KF"Ece

2. B ko Λ"d mAnkr aOr 5ΛsmF E/¿yAΛlkr ek cAp KF"Ece

3. C ko Λ"d mAnkr aOr 7ΛsmF E/¿yAΛlkr ek cAp KF"Ece

4. in cApo"Λ Γ'FEEtQC̈d Eb"dIko AΛs ä"Ekt kFEje

5. AB ko joEwe

6. AC ko joEwe

2 Related work

2.1 Anaphora

Root, Rebecca Louise et al. [11] use the Discourse Representation Theory of Hans Kamp to develop a theory in which each sentence contributes to building a representation of the meaning of the discourse. They introduce “reference markers” for each noun phrase and truth is defined in terms of an embedding of the representation of the model. Along with the formal explanations, it is argued that this approach is attractive from a computational point of view.

Altaf Rahman and Vincent Ng [10] use ensembles from a variety of supervised coreference models. They discuss different methods to apply a model-heterogenous ensemble for coreference resolution: ensemble-based methods significantly outperform the best member of the ensemble.

Nguyen et al. [8] use a filtering algorithm to rerank the output of coreference hypotheses, where the filter is based on relational structures between mentions and relationships. The second approach uses a joint model augmented with a set of relational features derived from semantic relationships of mentions. They claim improved performance of state-of-the-art coreference resolver based on learning.

Veselin Stoyanov et al. [13] build the coreference clusters incrementally, beginning with the most precise rules (termed as sieves by Raghunathan et al. (2010)) and use the available coreference information to guide the less precise sieves. Coreference resolution is inherently a global job. In this context, the approach of Raghunathan et al. (2010) may be taken as a rule-based greedy search for a globally consistent clustering.

2.2 NLP and Parsing

One of the most common techniques today to perform effective parsing is to use Part-of-Speech tagged language banks. Only few languages (about 8-10 of the 600+ languages in wide use) have the privilege of being resource-rich in the sense of having Penn-tree banks and other large-sized standardised corpus. 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 a language.

There has been a significant amount of work done on solving geometry construction problems. Gulwani et. al. [3] propose a method that uses goal-based heuristic to simulate backward deduction to solve a problem expressed in a predefined logical construct. Schreck et. al. [12] talk about the same problem but use CAD methods to deal with constraints. At the same time, Itzhaky et. al. [4] use the number of nondeterministic choices as a measure of a good solution. Ahmed, Umair et. al. [1] look more into using domain-specific measures to minimize parser errors and augment the geometry problem solver, GeoSynth. All these papers have provided us with valuable insights into selecting important and expressive constructs for our intermediate language.

We summarize our observations in terms of following 3 parameters:

- Uses other linguistic/domain knowledge

- Assumes linguistic cues are already translated into logical forms
- Uses Parse knowledge

Paper	**Uses other linguistic/domain knowledge**	**Assumes linguistic clues already traslated into logical forms**	**Uses parse knowledge**
Schreck et. al. [12]	YES	YES	NA
Gulwani et. al. [3]	YES	YES	NA
Itzhaky et. al. [4]	YES	YES	NA
Ahmed et. al. [1]	YES	NO	YES
Zettlemoyer et. al. [14]	YES	NO	YES

Table 1: Related Works

Unlike all previous attempts, we do not assume that the logical forms of the sentences are available, nor we assume the availability of a parser. We use cross-lingual alignment to map the given sentence in natural language to metalanguage. It is therefore applicable to a large range of languages. We demonstrate this for two widely known languages belonging to very different families: Hindi and English.

3 Design

The interpreter consists of the following components:

- Aligner (training)
- Grammar
- Natural Language (NL) Interpreter
  - NL to Metalanguage Mapper
  - Heuristics-based Parser
  - Semantics Analyser

  - Plotter

- Using Context to handle Anaphoras

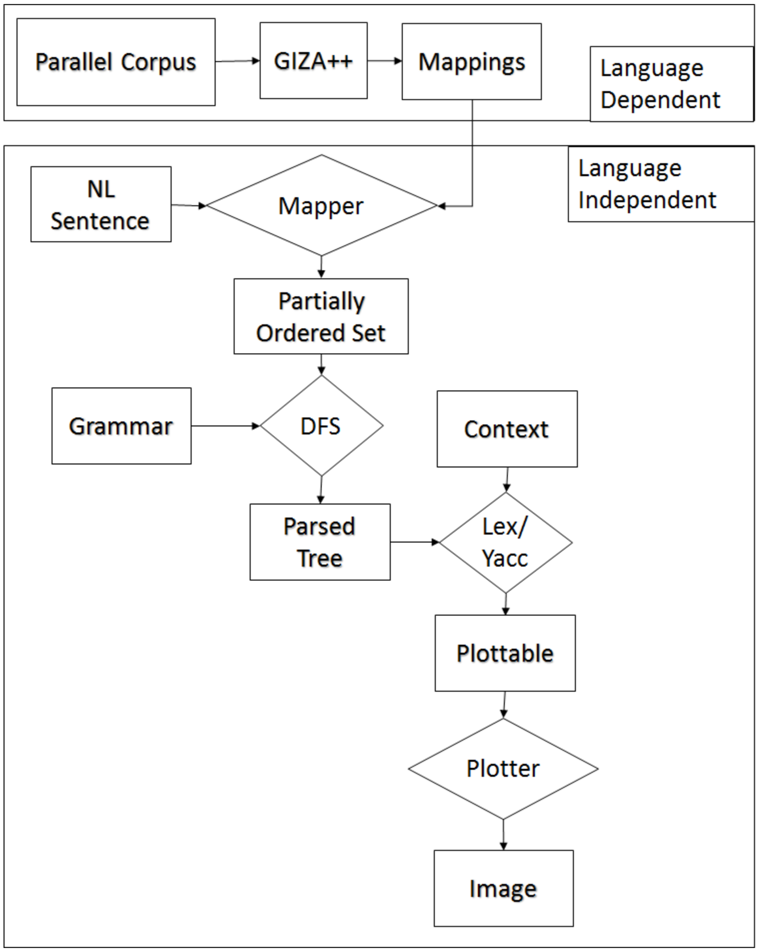


Figure 1: Work Flow

3.1 Aligner

Cross-lingual alignment is the heart of the learning mechanism in this project. It is used to obtain mapping/alignment between a natural language and our carefully designed and structured imperative metalanguage, L0. Since this alignment can be obtained for any natural language, the application is scalable to any number of natural languages.

We have used GIZA++ [9] as the cross-lingual aligner. GIZA++ is a statical machine translation toolkit that is used to train IBM Models 1-5 and an HMM word alignment model.

3.1.1 Cross-Lingual alignment

Cross-Lingual alignment is a technique to statistically align the words of a given pair of languages. It is close to translating one language into another, without any syntactic or semantic knowledge of any of the languages. As an example, for a pair of langauges, given sufficiently many sentences as illustrated in the table 2, a cross-lingual alignment assigns probabilities to the event that a particular source language token is mapped to a target language token.

English	**Hindi**	**Meta Language**
Construct a line AB of length 4 cm	4 Λs mF lMbAI kA ek ΛrKAKΓ'17X AB KF"Ece	`construct lineSegment AB length 4 cm`
With A as center and radius 3 cm, draw an arc	Λk"d A aOr E/¿yA 3 ΛsmFΛlkr ek cAp KF"Ece	`constrcut arc center A radius 3 cm`
With B as center and radius 5 cm, draw an arc cutting the previously drawn arc at C	Λk"d B aOr E/¿yA 5 ΛsmFΛlkr ek cAp KF"Ece jo phΛl KF"cF cAp ko C kAVtA ho	`construct intersectingArc center C radius 5 cm cuts arc previous at C`

Table 2: Sample Corpus

English	**Meta Language**	**Probability**
Construct	`construct`	1.00
Line segment	`lineSegment`	1.00
intersects	`intersect`	0.93
intersect eachother	`intersect`	0.78
cut eachother	`intersect`	0.87
cut	`cut`	0.98
join	`join`	1.00
mark	`mark`	0.98
label	`mark`	0.98
bisector	`bisector`	0.60
interior	`interior`	0.20
exterior	`exterior`	0.20

Table 3: Sample alignment between English and Metalanguage

Hindi	Meta Language	Probability
lgA dFEjAyr	`construct`	0.90
KF“Ece	`construct`	0.98
rcnA kFEjAyr	`construct`	0.96
bnAiAyr	`construct`	0.98
ArKAKΓ’17X	`lineSegment`	1.00
pr-pr Γ’FEEtQCd	`intersect`	0.98
kAVtA ho	`cut`	0.82
kAXW	`cut`	0.95
EmlAiAyr	`join`	0.98
joEwAyr	`join`	0.98
“aEkt kFEjAyr	`mark`	0.98
mAn lFEjAyr	`mark`	0.98
smEΓ’92BAjk	`bisector`	0.59
a¡yΓ’06tr	`interior`	0.20
bEhBAg	`exterior`	0.20

Table 4: Sample alignment between Hindi and Metalanguage

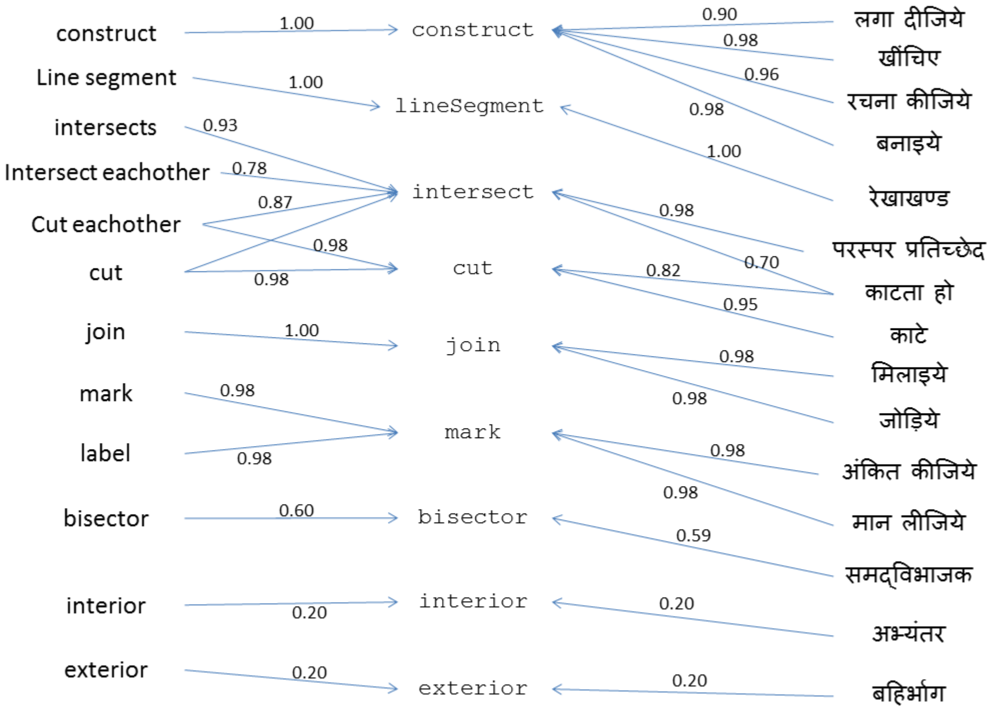


Figure 2: Graphical visualization of the sample alignment

3.2 Grammar

The grammar has been carefully designed to capture the underlying structure of the conventional (imperative) geometry construction steps. Note that there is a mild constraint on the grammar, which is a direct consequence of the assumption that 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 leftmost non-terminal guides the search for the remaining non-terminals. [see section on Proximity]

3.3 Natural Language (NL) Interpreter

3.3.1 NL to Metalanguage Mapper

The interpreter component exploits the alignment model obtained from the Aligner component. Given a sentence expressed in a natural language, we use the alignment model to get the statistically most probable mapping of the NL sentence to a list of words in the language defined by the grammar. This list of words need not conform to a sentence in the language defined by the grammar but still carries some intent of the NL sentence, and hence can be called a partially ordered list of metalanguage words.

For example, in our problem, we obtain the following partially ordered set of words for the 6 steps in English:

1. "`construct line segment BC length 6 cm`"
2. "`B center construct arc radius 5 cm`"
3. "`C center construct arc radius 7 cm`"
4. "`mark intersection point these arcs A`"
5. "`join AB`"
6. "`join AC`"

Similarly, following is the set for the 6 steps expressed in Hindi:

1. "`6 cm length line segment BC construct`"
2. "`B center 5 cm radius arc construct`"
3. "`C center 7 cm radius arc construct`"
4. "`these arcs intersection point A mark`"
5. "`AB join`"
6. "`AC join`"

3.3.2 Heuristics-based Parser

At this stage, the parser takes the partially ordered set of words generated in the previous step and tries to map it to a sentence formed by the grammar. This is done by performing a DFS on the grammar tree and trying to fit every token in the set with some grammar terminal. Given any production rule, our DFS-based algorithm recursively expands the leftmost unexpanded nonterminal in the rule; on encountering a terminal, it tries to fit the terminal to some token near its parent's/left sibling's token. The subtrees which are not satisfied are pruned and search resumes with the next available production rule. The search is aided by the following heuristics:

1. **Proximity**
   Proximity plays an important role in our parsing method e.g. we have assumed that the object that needs to be constructed would be near the "construct" word,

and the length of a line segment would be near its name. Consider, as an example, “`Construct a line segment AB of length 5 cm`”. Here, using our assumption we start searching for the object to be constructed near the “construct” word; it comes out to be the line segment AB. Note that we have to search on both the sides of the keyword (For corresponding hindi sentence, the set of words is “`5 cm length line segment BC construct`”). This assumption is justified by the structure of the sentences in the natural languages, and specifically the geometry construction steps. This reduces the parsing complexity many-fold. As another example, “`Construct a line segment AB such that length of AB is equal to the difference of the length of CD and length of EF`”. Here, only those parse trees which demand the construction of AB are retained (as AB is nearer to “construct” than CD and EF); remaining trees which involve the construction of CD or EF are pruned.

2. **Type system**
   The system is implicitly type-defined. For example, consider a statement like “`Mark a point A on it`”. Here “it” can be a line, line segment, circle etc. but not a point, as marking a point on a point is not permitted. Also, point pairs like AB would define a line segment or a ray, ABC would define an arc, ’l’ would define a line (according to the conventions in the existing geometry texts). The grammar and the semantic analyser together form the type system.

3. **Validity of the output**
   The search is also guided by the output produced at this stage. If the output produced seems correct, i.e., it uses most of the tokens in the set and conforms to some sentence generated by the grammar, it is passed to the semantic analyser. The semantic analyser then checks whether the output represents a valid intent to draw, i.e., a valid plottable object can be inferred out of it. If not, the parser tries to correct the parse tree starting from the rightmost leaf; this procedure is repeated until either we obtain something plottable or exhaust all the possibilities.

For example, in our problem, the output sentences in the language defined by the grammar would be

1. “`construct line segment BC length 6 cm`”
2. “`construct arc center B radius 5 cm`”
3. “`construct arc center C radius 7 cm`”
4. “`mark intersection point these arcs A`”
5. “`join AB`”
6. “`join AC`”

Note that, the output sentences for the hindi sentences would be the same, as the underlying grammar in which the parser tries to fit the sentences is the same.

3.3.3 Semantics Analyser

This stage of the parser uses Lex-Yacc to generate a set of primitive objects like points, lines, arcs etc. which need to be plotted in the last input construction step. All coordinates, radii, lengths and angles are resolved in this step. Anaphora are also resolved in this stage. This is detailed in the later sections.

The semantics analyser maintains two lists of plottables, one corresponding to the "context" (which is derived from the objects which were plotted in the previous construction steps) and "diff", which is generated from the current construction step and stores the objects which are to be plotted at this stage. At the end of the plotting step, the "diff" is appended to the "context". We put a restriction that is an object requires some primitive objects to be plotted, we require that such objects are either in the running context or in the "diff" list, e.g., a line segment construction would require the two points, which determine the line segment, to be plotted first. So these points with their coordinates are also stored in the "diff" list.

In the running example, after the first step, the generated diff list would be as follows:
`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 after the fifth step would be
`points={B(0,0), C(6,0), A(1,`$\sqrt{}$`24)}, line segments={BC, AB}, arcs={(B,5), (C,7)}`

The objects are arranged in the "context" and the "diff" list such that the last plotted object is at the end of the respective list. For a point which occurs for the first time in the construction step, and is independent of the running context, we assign a coordinate randomly.

3.3.4 Plotter

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

3.4 Using Task Context to handle Anaphora

An anaphora is a word or phrase which points back to a previously referred linguistic or semantic object. Thus the anaphora and its antecedent both corefer to the same referent.

The semantics analyser uses the context (a list of objects which were plotted in the previous construction steps) to resolve the anaphoras. For example, consider the sentence "`Mark a point M on it`". Here "it" would refer to the most recently plotted markable object (objects on which a point can be marked e.g. lines, line segments, arcs, circles etc.). We fetch such markable object from the context to resolve the anaphora.

In case of the problem we consider, suppose we combine the fourth, fifth and the sixth step into "`With B and C as centers and radii 5 and 7 cm, draw two arcs intersecting eachother at A`". Here, "arcs" is the antecedent and "eachother" is the anaphora, both corefering to the same referent i.e. the arcs being drawn. Since the arcs have not been constructed yet, they are not in the task context. In this case, the semantic analyser comes to our rescue by providing for such sentences in the grammar.

At the same time, in the original fourth step, "`mark intersection point these arcs A`", 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 following plot (Note that the dotted circles represent arcs):

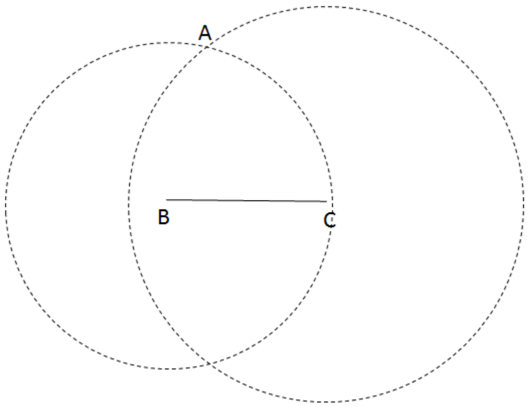


Figure 3: Plot obtained after interpreting first four steps

If we join the fifth and the sixth step into "`Join AB and AC`", we get a case of ellipsis: the intent is to join AC too. Such cases are not handled here and are planned to be addressed in future.

By the sixth and the final step, we obtain the following plot:

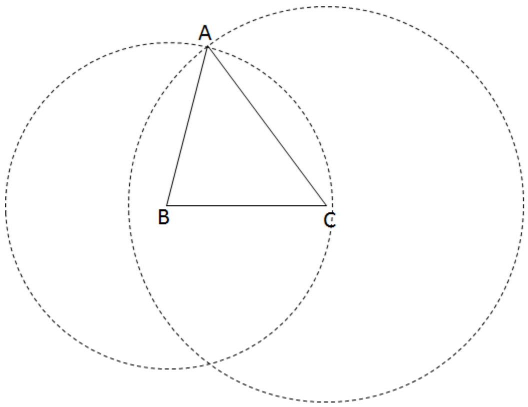


Figure 4: Plot obtained after interpreting all the steps

3.5 Language Independence

Given a corpus in the specified format, the system generates a probabilistic mapping between the words in the language and the words in the meta language using cross-lingual alignment. This is then used in later stages to convert the presented sentence in the language to the meta language. After this conversion, the system behaves the same independent of the natutal language specified.

4 Parsing metalanguage to generate plot

The first input sentence in the English version of our problem is "`Construct a line segment BC of length 6 cm`". Using the English-to-Metalanguage map, we translate this sentence tokenwise to "`construct line segment AB length 7 cm`" where "a" and "of" map to empty words. Similarly, for the Hindi input sentence "7 ΛsmF lMbAI kA ek ΛrKAKΓ'17X AB KFEce", using the Hindi-to-Metalanguage map, we obtain the translation "7 cm length line segment construct", where "kA" and "ek" map to empty word.

Using heuristics-based parsing, we obtain the same parse tree for both of these translations:

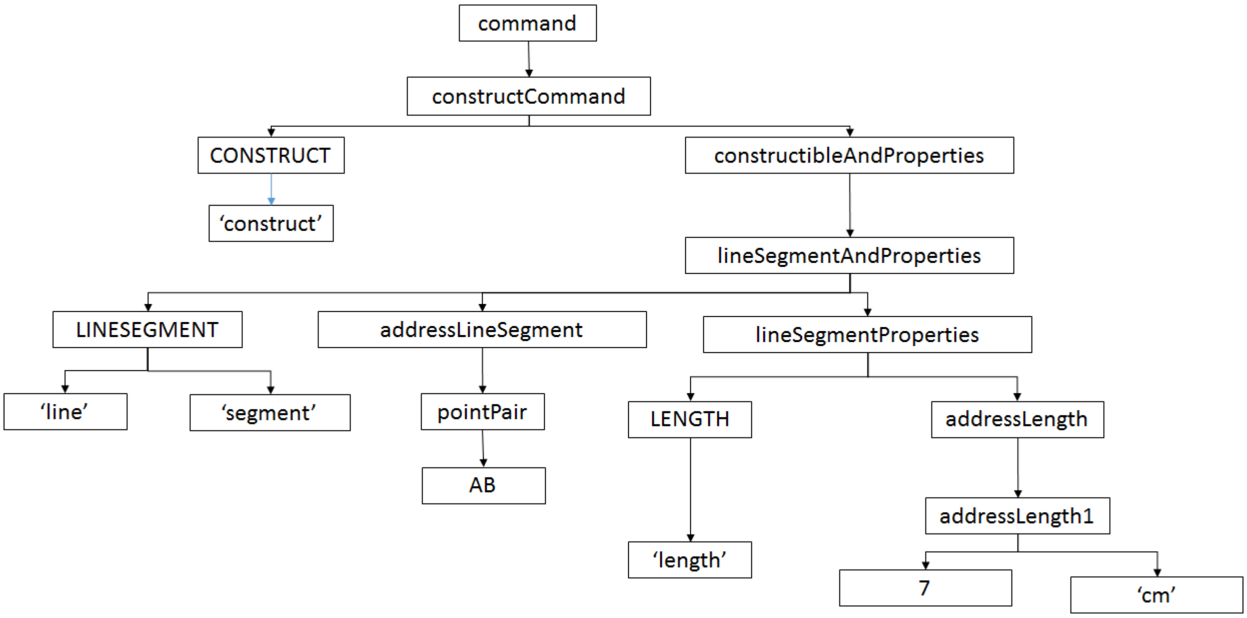


Figure 5: Sample Parse Tree

The semantic analyser, which is, in reality, a lex-yacc parser, when presented with pre-order string of this parse tree, outputs a list of primitive objects to be plotted:

`points={B(0,0), C(6,0)}, line segments={BC}`

The plotter is a simple program which plots these geometric objects.

This approach has been shown to work for simple construction steps.

5 Corpus Collection and Testing

The corpus contains around 350 sentences from the geometry construction field in both the languages. These were collected from the NCERT textbooks of standards 6^{th} to 9^{th}. English-Metalanguage and Hindi-Metalanguage sentence pairs were generated manually which were then used to train the cross lingual alignment software

The corpus has been formatted as follows:
For every sentence in natural language,

- First line corresponds to the natural language sentence.
- The second line corresponds to the intended meta langauge sentence.
- The third line is an empty line.

The corpus is available at [2]

To get an almost perfect probabilistic map between the parallel corpus, we impose a mild constraint: let the map be from language A to B; we design the corpus such that every word of the language A either maps to exactly one word of language B or to the empty word. Note that this is not necessary. It has been done so as to obtain the best possible map for a given corpus size under corss-lingual alignment. Poorer quality of map leads to considerable increase in the possible translations that the parser would have to parse.

6 Results

1. Number of sentences in the corpus: 360 each in English-to-Metalanguage and Hindi-to-Metalanguage corpus

2. Number of unique tokens in each of the three languages are given below

Language	**Number of unique tokens**
English	181+
Hindi	169+
Metalanguage	110+

Table 5: Unique tokens in different languages

3. The implementation demonstrates the performance of the system for atomic construction statements like "`construct a line segment AB of length 5 cm`" etc. Statements like "`Construct an equilateral triangle of side length 5 cm`", which involve more than one atomic step, have not been implemented.

   The chapter on Geometry from CBSE NCERT Mathematics textbook for 6^{th} standard contains only the atomic construction steps, while chapters from 7^{th}, 8^{th} and 9^{th} standard contain construction steps which require two or more applications of simple ruler-compass based (atomic) construction step.

   Tabulated below is the performance of curent implementation. We are confident that the accuracy is bound to increase once the grammar and implementation is full-fledged.

Standard/Grade	Corpus Size	Number of successful parses	Percentage
6^{th}	84	77	91.67 %
6^{th} to 9^{th}	225	172	76.44 %

Table 6: Unique tokens in different languages

6.1 Failure Cases

1. It is evident that sentences which are forbidden by the grammar can not be handled by the interpreter. This shortcoming can be easily removed by making the grammar more expressive, keeping in mind that it doesn't become ambiguous.

2. In cases such as "`With A and B as centers, and radius as 4 and 5 cm, draw two arcs intersecting eachother at C`", we see the "B" is nearer to the paramter "centers", while "4" is nearer to the parameter "radius". This guides the heuristics-based parser to draw two arcs with centers B and A and radius as 4 and 5 cm respectively; but the intent is to draw two arcs with centers A and B and radius as 4 and 5 cm respectively.

   Clearly, if the sentence were "`With centers as A and B, and radius as 4 and 5 cm, draw two arcs intersecting eachother at C`", we would have captured the correct intent.

6.2 Difficulties & Resolutions

- **Anaphoras** Dealing with anaphoras like "this", "it", "previous", "last" etc in sentences like "`Mark any point M on it`", "ek sEvDAjnk E/¿yAΛkr EpCΛ crZ vAΛ cAp ko EbΓ'06dΓA pr kAX:" etc.
  Anaphora are resolved by using the context (set of objects drawn in the previous steps).

- **Underspecified Parameters** We also need to provide for unspecified radii, lengths and anlge measures e.g. "`With A and B as centers and a suitable radius, draw two arcs interseting eachother 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`")

- **Probabilistic Mapping** Since the mapping is statistical in nature, there might be various close alignments for the same source-language word. This forces us to resort to try out different possible alignments and look for the one that fits e.g. for the sentence "`Construct AB of length 7.8cm`", we get the following alignments with their corresponding probabilites (we assume that word-wise alignments are independent of eachother)

Mapped meta-language sentence	**Probability**
`construct AB any length 7.8cm`	0.716853
`construct AB lineSegment length` 7.8cm	0.21081
`construct AB angle length 7.8cm`	0.0723206
`construct AB center length 7.8cm`	1.90645e-06

Table 7: Map to possible metalanguage sentences alongwith their probabilities

After incorporating the assumption that the every word in the sentence should map to either a word in the meta language or some null word, the map comes out to be nearly perfect. Even less frequent words like "equilateral", "trapezium" etc. map with very high probabilities with their meta langauge counterparts. This gives increased confidence in the correctness of meta langauge sentence produced from a given natural language sentence.

OA aOr OB kA smEГ’92BAjk KF"Ece A aOr B ko Ed mAnkr aOr 4 aOr 5 smF E/¿yA
Nkr do cAp KF"Ece jo pr-pr C pr kAVtF ho iskF do jFГ’9FA KF"Ece isk cAr BAg kFEjy

6.3 Future Work

1. Enrich the corpus as well as grammar to cover larger domains of expressions e.g. having larger number of primitive steps, providing for higher-level primitives, specifying relative distances, dealing with anaphoras and references

2. Expand the functionality of the system to work for all kinds of construction steps claimed so far

3. Deploy the system on a webserver to demonstrate its functionality and work towards making it more robust

4. In order to capture colloquial (non-academic) ways of expressing construction steps, we need to develop a user interface that augments the corpus with input sentences

Annotated References

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

> Focusing on the educational domain, this paper seeks to build robust domain specific natural language understanding modules by suitable preprocessing of the text and use of domain knowledge to deal with parser errors. The paper starts with showing that even off-the-shelf parsers do not perform up to the mark on texts with domain-specific vocabulary and long sentences. The paper also talks about GeoSynth, built using state-of-the-art parsing technology. GeoSynth is a system which can automatically solve geometric constructions word problems.

[2] Parallel Corpus for Geometry Constructions. http://cse.iitk.ac.in/users/pratikkr/cs499/index.html.

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

> Aiming to solve ruler and compass based geometry construction problems, the tool described in this paper claims to have synthesized construction steps for problems posed from high-school textbooks and examination papers. It focuses on three important insights viz. reduction of symbolic reasoning to concrete reasoning using verification based on random testing, augmenting the instruction set with higher level constructs and simulating backward deduction used by humans by using a goal-based heuristic.

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

> This paper combines deductive, numeric and inductive reasoning to solve geometric problems. Using a mixture of these methods, several problems are solved which, otherwise, wouldn't have been solved using the methods individually. The paper uses the number of nondeterministic choices as a measure of how close a partial program is to the solution, claiming this to be useful in grading and providing hints to the students. The implementation takes an average of few seconds on 18 Scholastic Aptitude Test geometry problems and 11 other ruler and compass-based geometry construction problems.

[5] Dan Jurafsky and James H Martin. *Speech & Language Processing*. Pearson Education India, 2000.

> An anaphora is a "REFERRING EXPRESSION" that "COREFERS" to some entity already introduced in the discourse context. Its use depends on the speaker's belief about the role of the referent within the hearer's set of beliefs (pg 666). In our case, the "context" constitues that of the drawing interpreter's.

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

[7] Ruslan Mitkov. *Anaphora resolution*, volume 134. Longman London, 2002.

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

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

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

[11] Rebecca Louise Root. *The semantics of anaphora in discourse*. PhD thesis, University of Texas at Austin, 1986.

[12] Pascal Schreck, Pascal Mathis, and Julien Narboux. 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, 2012.

> This paper introduces CAD methods to deal with Constraint Satisfaction Problems related to geometry, at the same time claims to have met the requirement met by the educational domain. The methods show that results must be construction programs to take into account particular cases. A knowledge-based system implemented in Prolog has also been presented.

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

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

With the objective of mapping natural language sentences to their counterpart lambda-calculus encoding, this paper introduces an algorithm which learns from the input set of natural language sentences labeled with lambda calculus expressions. The algorithm attempts to infer the underlying distribution over semantic and syntactic analyses conditioned on the input sentences and outputs a log-linear model.