INDIAN INSTITUTE OF TECHNOLOGY, KANPUR

Generating Solutions for basic Speed, Time and Distance Word Problems

by

Varun Kumar Khare

A report submitted for the under-graduate project

under

Arnab Bhattacharya
Department of Computer Science and Engineering

23 November 2017

Abstract

by Varun Kumar Khare

There is a large disparity in the access to proper education at the school level. We seek to build a problem aware online tutoring system for students to help improve the situation. In this project, we have made a complete solution generator which, given a word problem on arithmetic at the levels of classes 6-8, extracts relevant information from the question and solves the problem in a step-by-step manner. Currently, we are handling only basic speed, time and distance problems.

We have tried to mimic the approach children tend to use when asked to solve a word problem. The novel approach ensures that the steps we come up with are intuitive to understand, and conform to the simple formulas known to the school children. We propose a text-to-world mapping to create an undirected graph symbolizing the relationships between the objects mentioned in the text and then using the global function space to generate the answer.

We will conclude with how to extend the system to more linguistically complicated examples and problems involving more than one function.

Acknowledgements

I am grateful to Prof. Arnab Bhattacharya and Prof. Amay Karkare who gave their valuable time and guidance in this project. They were very helpful in helping me correct my mistakes and take further steps.

Contents

Chapter 1

Introduction

1.1 Problem Statement

> Given a preliminary word problem based on speed, time and distance relationship in natural language, generate a step-by-step solution for it.

1.2 Background Theory

We will briefly describing the concepts involved in Natural Language Processing and graph traversal techniques which would form an integral part of the later discussion.

1.2.1 Techniques and Terminology in Natural Language Processing

1.2.1.1 Syntactic Parsing

Within computational linguistics the term is used to refer to the formal analysis by a computer of a sentence or other string of words into its constituents, resulting in a parse tree showing their syntactic relation to each other, which may also contain semantic and other information.[1]

1.2.1.2 NP-Chunking

Noun Phrase chunking, or NP-chunking refers to searching for a group of tokens corresponding to individual noun phrases within the sentence. {*modified from*[5]}

1.2.1.3 Anaphora & Cataphora

Anaphora is the use of an expression whose interpretation depends upon another expression in context (its antecedent or postcedent). In a narrower sense, anaphora is the use of an expression that depends specifically upon an antecedent expression.[1]

Cataphora is the use of an expression or word that co-refers with a later, more specific, expression in the discourse. The preceding expression, whose meaning is determined or specified by the later expression, may be called a cataphor.[1]

We will be using anaphora resolution systems for identifying co-references in the text.

1.2.2 Graph traversal

This refers to visiting every node of a graph. There are several techniques to do this. However knowledge of DFS (Depth-First Search) traversal techniques suffices for the needs of the project.

DFS:A depth-first search (DFS) is an algorithm for traversing a finite graph. DFS visits the child vertices before visiting the sibling vertices; that is, it traverses the depth of any particular path before exploring its breadth.[1]

Chapter 2

Proposed Solution

Our approach to solve mathematical word problems is based on the ability of represent the problem via relationships between various object described in the text. We start out by creating a graph which represents the hypothetical world described in the problem with a global function space defining the rules/association between different types of objects.

We keep on adding nodes in the graph as when a new object (Noun Phrase) appears in the text and depict the association between two objects by an edge between them. If multiple functions are there multiple types of edges need to be created. If a Noun Phrase is an anaphora, all the objects which would have been linked to this noun phrase would instead be linked with the representative mention and node corresponding to the current Noun Phrase won't be added to the graph.

A set of nodes are eligible as arguments for a function *iff* they correspond to the object types used in the functional relationship and are connected by edges corresponding to the function. Each measurable quantity has a value attribute associated to it which stores the value of that quantity.

Once the graph is generated, we need to calculate the value of the query node. To do this, we traverse the graph starting DFS at query node (Top-Down approach) and check if we are able to calculate its value along any one of the computation path. We work recursively to evaluate all the nodes of the computation path from leaf and then finally evaluate the value of the root node. The following figures explain the graph construction.

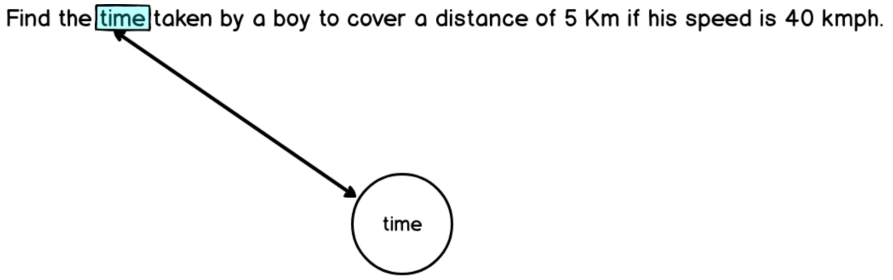

FIGURE 2.1: "the time (**NP**)"

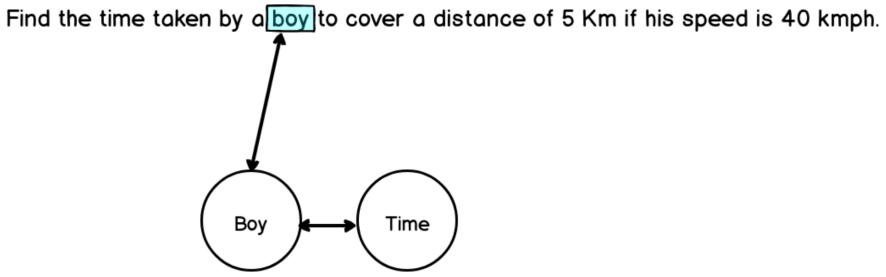

FIGURE 2.2: "a boy (**NP**)"

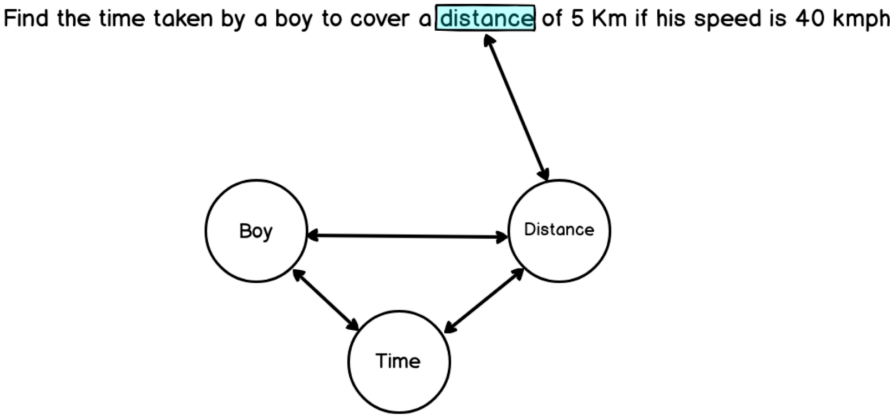

FIGURE 2.3: "a distance (**NP**)"

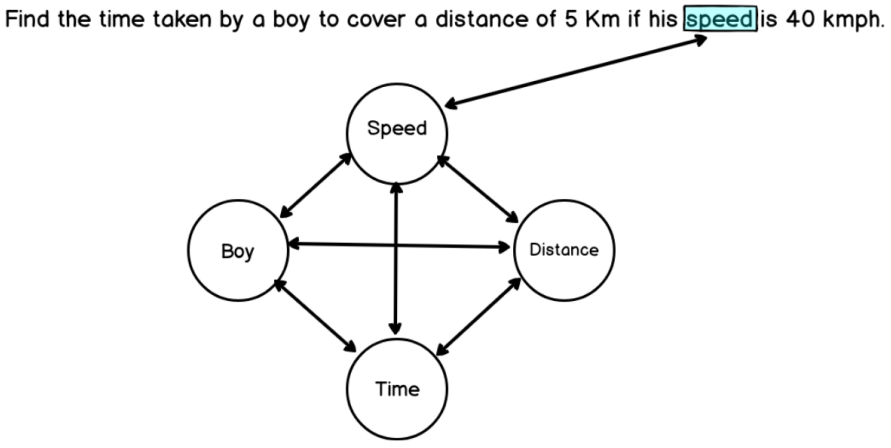

FIGURE 2.4: "his speed (**NP**)"

Chapter 3

Implementation Details

3.1 Approximate Algorithm

1. Start reading the text.
2. Initialize a node in the graph if it has a corresponding noun phrase and is not anaphor
3. Assume global function space
4. Link two objects if they are valid arguments for a function
5. Search in the function space to determine arguments needed to calculate
6. Solve the query recursively

3.2 Structural Ambiguity Resolution

This is done by using the SpaCy's[2] deep learning based syntactic parser. SpaCy parser creates the most probable parse tree representation based on the context of the sentence and the Part-of-Speech tagging. It also gives a dependency parse which tells the subject(*nsubj*),adverb modifiers(*advmod*),object(*pobj*),etc relationships crucial for further disambiguation.

3.3 Anaphora Resolution

Anaphora resolution is done by using the neural network based co-reference resolution[4] system available via stanford CoreNLP [6] toolkit. Additional ambiguity handling was added directly in the code so as to encompass relationships not mentioned in the sentence text but are rather taken for granted. For example

> Rahul travels 5 Km by his car. If he takes 1 hour to travel, find his speed.

As we can see the relationship between distance and time is not explicitly seen in the sentence. It could have been any other distance object if available, but since only one distance was described in the problem, we take for granted the time and speed should be associated to this this object only.

3.4 Quantity entailment[3]

This is the task of assigning value to the associated quantity. The object type is identified by the units following the cardinal value. For example 60 Km/hr can only be associated to a speed object and not a distance object.

The exact object is searched according to the dependency parsing within a window size of 6 words (a size of 6 usually defines the context of the current word). We search for the nearest ancestor associated to the cardinal value by traversing the dependency tree within this window. If the ancestor is an anaphor the value is allotted to the representative mention instead.

3.5 Quantity Association

We base this on the assumption that quantities mentioned far-off from each other rare less likely to be connected together until explicit referencing is done. So we only associate quantities that fall in the same sentence. This default case behaviour is over ruled in case of co-referencing when linking occurs with a distant object.

Chapter 4

Experimentation and Conclusion

4.1 Testing

Since the project is still in its infancy it wasn't possible for us to test it on a huge set of corpora. The problems that it could understand were restricted and testing needed a controlled environment. The biggest challenge that we faced was handling prior world knowledge in the system. Currently, we only took those problem which didn't need this world-knowledge for evaluation. This assumption seems a bit heavy so, we came up with an approach to incorporate it and have left it under future work for testing and validating its effectiveness.

We also assumed that the problems didn't need the knowledge of monotonic nature of the quantities and the association of words to modifications in the values of quantities like *Late* w.r.t time refers to addition of time to certain time object similarly *longer* w.r.t distance implies higher magnitude.

Owing to a deterministic algorithm for query solving, if the system was able to understand the language it gives the correct answer. So, we took a set of 11 different type of problems that needed only *Distance = Speed*Time* as relationship. Out of these we were able to solve 7 problems soem of which are mentioned below. The source code for the experiment is available at .

Some test problems on which the model worked successfully:-

1. How much time Raman will take to cover a distance of 400 km if he runs at a speed of 20 kmph ?
2. The distance between two stations is 240 km. A train takes 4 hours to cover that distance. Calculate the speed of the train.
3. A person runs a distance of 60 km in 5 hours. What is his speed ?
4. Nancy travelled a distance of 455 km by car in 10 hours. Find the speed of the car.
5. Ramesh travels at a speed of 30 kmph for a time of 2 hours to travel a certain distance. How much time will he take to cover that distance if his new speed is 20 kmph?
6. If it takes 3 hours to drive a distance of 192 km on a motorway, what would be your speed?

4.2 Problems as challenges

These were the rest of the problems which were based solely on distance speed and time relationship but could not be solved because of unavailability of word manipulations or monotonic nature of the quantities.

1. Speed of a train is 20 meters per second. It can cross a pole in 10 seconds. What is the length of train ?
2. A car moves with a speed of 40 km/h for 15 hours and then with a speed of 60 km/h for the next 15 hours. What is the total distance covered by the car?
3. Ram walks at a speed of 12 km/h to cover a distance of 5 Km between his school and home. He arrived his school 10 minutes LATE. Find the usual time he takes to cover the distance between his school and home?

4.3 Conclusion

Our system was able to generate step-by-step solutions to the above problems and owing to the top-down approach in evaluating the answer our model produced steps that were intuitive to understand and could be easily given as hints if incorporated in the tutoring system. The graph that is generated for the problem uniquely determines the difficulty level of the problem i.e. a problem with longer computation path in the DFS tree (the one from which answer is evaluated) is more difficult. So this measure can be used for objectively identifying the difficulty of the problem.

We also make a point that even though the approach is general and graph construction can be closely associated to text-to-image problem but we intentionally remove a lot of details from the sentence to simplify the problem as they are not needed to evaluate the answer.

Chapter 5

Future Work

5.1 Future Tasks

We propose the following methods to overcome the challenges mentioned earlier and would be following our future work in this direction to improve the system,

1. Automating the graph generation for any type of problem
   (a) Graph Neural Nets are highly effective in handling word to graph manipulations. They should work out better in handling linking between different objects.
   (b) The monotonic and sequential nature of the quantities can be derived using Semantic Role Labeller. This would be helpful in adding word-to-quantity manipulations in general.
2. Synonymous descriptions like associating length to distance, age to time can be worked by training a ML classifier to associate these labels to noun-phrases. Semantic similarity or paraphrase matching systems might work out better in these.
3. There are some relationships which are not given by function space but rather by ownership like a distance travelled by a car is owned by the agent(car). Such relationships can help in handling ambiguity in associating quantities together.

Bibliography

[1] Wikipedia https://en.wikipedia.org/wiki/Main_Page

[2] SpaCy. *http://spacy.io.*

[3] Subhro Roy, Tim Vieira, Dan Roth *Diss. University of Illinois at Urbana-Champaign.* [*Reasoning about Quantities in Natural Language*]. 2017.

[4] Clark, Kevin and Manning, Christopher D. *Empirical Methods on Natural Language Processing.* [*Deep Reinforcement Learning for Mention-Ranking Coreference Model*]. 2016.

[5] *Natural Language ToolKit.* https://www.nltk.org

[6] Manning, Christopher D., Mihai Surdeanu, John Bauer, Jenny Finkel, Steven J. Bethard, and David McClosky. *The Stanford CoreNLP Natural Language Processing Toolkit* [*Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations*] 2014.