Relation Extraction for Matrix(type) entities in Introductory programing problems

CS365A: Artificial Intelligence
Project Report
by

Himanshu Shukla (13309)
Kumar Gaurav (12368)

under the guidance of

Prof. Amitabha Mukerjee

Indian Institute of Technology, Kanpur

April 18, 2015

Abstract

Relation Extraction has been an important task in natural language processing since early 1990s and there is no need to specify the use of relation extraction in real life. Relation extraction has been done in lot of fields for example bio-informatics, organisation-affiliation relations, etc. We in this project have tried to extract relations about mathematical entities(matrix) in the specific field of programing problems which are done in elementary Programing courses at various universities. We use statistical machine translation and compiler design concepts solve this problem.

Acknowledgement

This course turned out to be a great learning experience for both of us. The idea of making the students do a project is great as it makes us learn and explore deeper into certain topics which otherwise are not covered in the Classes. We would like to show our gratitude to Prof. Amitabha Mukerjee and the team of CS365, Shashwat Chandra, Triya Bhattacharya, Nimisha Agarwal and Rajesh Shubhankar for helping us out in this project. Further we would like to thank Pankaj Prateek and Prof. Amey Karkare for giving us their valuable time to help us in this project.

Sincerely,
Himanshu Shukla (13309),
Kumar Gaurav (12368).
Indian Institute of Technology, Kanpur.

Contents

1 Introduction

Relations are important features inside any text and these things become more important when we enter into programing domains specially for the beginners who faces a lot of problem while programing. We define relations for the matrix type mathematical entities do extraction for the same in this project however the method that we use can be extended for other type of programing problems.

Relation for the matrix type entities in introductory programing problems are their attributes (size, contains, type, etc.) and operations (sum, sort, rows, etc). We define a domain specific language which we call as Bridging language which is similar to the bridging language defined in Pankaj et. al. 2014. We use Statistical Machine Translation tool MosesDecoder[3] and GIZA++[4] for mapping the natural English statements to metalanguage. The metalanguage is a formal grammar based language which can be parsed using Lex-Yacc compiler software. The Yacc software is also used for analyzing the semantics with reductions. The same can be used for resolving anaphora present in the metalanguage.

2 Motivation

In the programming domain, the problem specification is very important especially if we talk about the introductory programming problems. The problem statements need to be well specified and unambiguous. There a lot of courses running on the massive open on-line course (MOOC) plat-forms in which if the problems specification is not clear and the beginner faces a lot of problem.

Also in all of the IDEs at present data we get only the main function already written and when the student writes the program, he many a times does not use any function which are considered to be good programming practices. Hence she/he can not learn those practices which is the actual motive of such courses. We wanted to automatically generate the header of some functions that the student needs to define in order to solve the problem.

For such changes like ambiguity checking and automatic header generation in the IDE developed by *IIT Kanpur's-SIGPACT*, we wanted to extract relations in the programing problems. Hence our basic motive is to design a system that can take a programming problem and generate a relation tree for it, which will contain relations for the entities, their attributes and operations defined on them.

3 Related Works

There have been a lot of related work in the field of relation extraction. People have used a lot of methods like

- Distant Supervision
- Supervised Learning
- Unsupervised Learning
- Kernel Methods on syntactic Parse Trees.

to extract relations in various fields like bio-informatics, affiliation relations, name-work relations, country- citizenship relation. The important thing with these was the presence of data on the Internet, which could be easily manipulated to use the methods and also the corpus for many such problems has been generated. An important work[5] was done by Pankaj Prateek and Jeetesh Mangwani at IIT Kanpur for automatically solving the construction problems of NCERT level using Statistical Machine Translation. We map our problem to their problem and take help of their method to solve our problem.

4 Rejection Of Methods

- *Rejection of unsupervised or distant learning:* Unsupervised learning can not be used as the amount of data required is very high and there are not much problem statements available for the introductory programing problems over the topic of matrices.
- *Distant learning:* In distant learning we weakly label data by using certain rules and heuristics. The training data is not so big in distant supervision but we require a pool of unlabeled data for testing and make the system learn. Here again is a problem of unlabeled large test data which is not available for this project.
- *Rejection of Methods used for bio-informatics:* The important pattern in for bio-informatics field is that the actions are limited and the entities can have fixed name. Hence this approach seems more promising approach than other approaches mentioned above but the problem is that this pattern

is actually not available with the programming problem as name of matrices are sometimes mentioned and many times not mentioned. size for square matrix is just given as n but the size is a tuple for a matrix and many such case are there.

- *Rejection of Method of Alessandro Moschitti:* Work of Alessandro Moschitti[1] was closest to our problem because it mapped the natural language questions to SQL queries, which is a restricted domain language(related to geolocation queries). We rejected this work because this method received much less accuracy for even small domain as mentioned by the the paper Moschitti et. al.[2].

5 Methodology

We closely follow the methodology adopted by Pankaj et al[5]. We realise the fact that the relation extraction becomes deterministic if English was a restricted language. By restricted we mean it can be generated by a grammar. Also we realise that the statements of programming problems have some what defined structure. Hence conversion of the problems from natural English to a restricted language should be possible. We devise the following strategy to solve the problem.

$$NaturalEnglish \longrightarrow \mathit{Bridging\ Language} \longrightarrow RelationTree.$$

The conversion of English language to Bridge language is done using Statistical Machine Translation that is a very popular technique used for automatic translation of one language to another language and we are essentially doing the same thing. For this we use GIZA++[4] which is word to word aligner and uses HMM (Hidden Morkov Models) for the alignment of words from source language (natural English) to target language (bridging language).

Since we know that the Bridging language would have a defined grammar and because of this fact it becomes easy to parse using compiler generator software. Handling anaphora also becomes possible for this language.We use Lex-Yacc for this. We have assumed that the Bridging language has LALR grammar.These steps are discussed in detail in following sections.

6 Statistical Machine Translation

Statistical Machine Translation is a technique of translating the sentence of one language called source language to the target language. This uses information theory as its base. The sentence is translated according to the probability distribution $P(e|f)$ where e represents the event that sentence translation is e given that the foreign language sentence is f. Finding the best translation $\tilde{e}$ is done by picking up the one that gives the highest probability: $\tilde{e} = arg\max_{e \in e^*}$

$$\tilde{e} = arg\max_{e \in e^*} p(e|f) = arg\max_{e \in e^*} p(f|e)p(e)$$

$P(F|E)$ Translation model
$P(E)$ Language model
The translation is done sentence by sentence by using approximated smoothed n-gram language models. STMs are of three types: Word Based, Phrase Based and Syntax Based. The Word Based alignment model was one of the initially used SMTs and we start with GIZA++ which is the word based alignment model to generate the probability mappings of different word of the English language to the bridging language. GIZA++ uses the HMM for generation of mappings. We do word based alignment because of the fact that we dont have enough corpus to use the phrase based aligners which needs at least a corpus of 1K sentences. More information about the SMTs could be found at the wikipedia page and the following link http://www.statmt.org/.

6.1 Principle of GIZA++

Principle of GIZA++:(`http://essay.utwente.nl/58377/1/scriptie_B_Fournier.pdf`) We generate the alignment by a program called GIZA++ that implements the Hidden Markov Models for generating the alignments. GIZA++ produces a word-level alignment on a sentence aligned parallel corpus. GIZA++ will produce a one-to-many alignment, in which words in the target sentence may only be aligned to a single word in the source sentence. This is illustrated in the following figure. Source: (`http://essay.utwente.nl/58377/1/scriptie_B_Fournier.pdf`)

6.2 Corpus Generation

This is the biggest challenge with the machine translation tools because we need the labeled corpus of both the languages in a sufficient amounts at-least 500 sentences each. We generated the corpus from using programing problems from SIGPACT IIT Kanpur, `http://www.sanfoundry.com/c-programming-examples-matrix/`, asking friends for generating problems, and we ourselves wrote few of the problems, this ensured that the corpus is not biased from our language pattern. Then we wrote the bridge language translation for all the problems ensuring that we use an consistent pattern of operations. For example if a sentence was find minimum of the sum of a and b, then we define an operation minimum which takes a set of arguments either 2 or in the form of a set example matrix. Here the language translation will be find minimum sum a b, sum acts on a and b, and minimum acts on sum of a, b. We have ensured such a consistency in the metalanguage and also mapped all the words like display, print, show to single word print. With such transformations we have ensured that there is a single word map for the all synonyms of a word which helps us to define a grammar for this language.

This part took most of the time as replacing the synonyms and maintaining the consistency of the maps was a big problem.

6.3 Filter

The mapping systems consists of two major phases. First, the natural English problems is converted to generalized English problems by converting variables, numbers and size descriptors into simple words like $varNum$, $varSymbol$ and $varSize$. Using this technique, the alignment obtained using GIZA++ is improved heavily. When we first used the GIZA directly with the English corpus then the mappings generated were really bad because of the sentences contained the phrases like you are given ,write a program, etc. Hence we wrote a program to get rid of such unwanted words. The program is earlier trained using the corpus which contained only relevant words that contribute to bridge language generation. After doing this we found that the probability mappings improved.

6.4 Bridge Language

The filtered statement after the above step, is then fed to the trained GIZA++ system that produces the required mapping. The GIZA++ system is trained using a parallel corpus that was also filtered using the same system to ensure good alignment

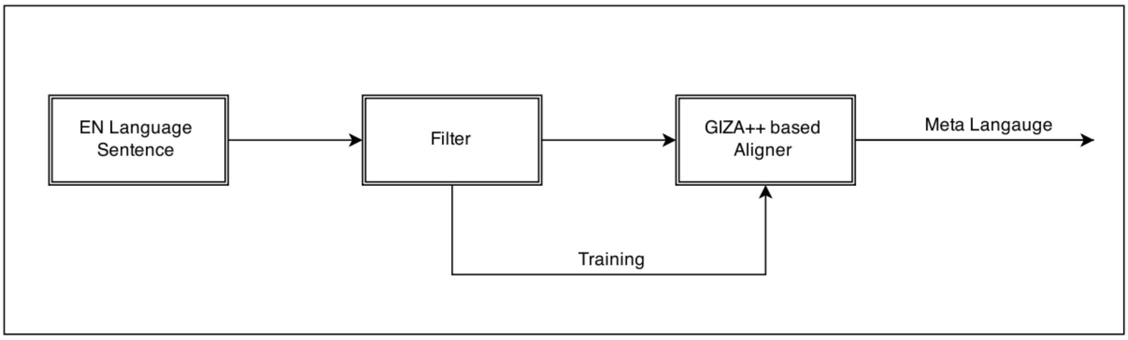


Figure 1: Systematic diagram to show the flow from English to Bridge Language.

results. After this step, a statement is generated which is the bridge-language mapping of the English sentence at the beginning.

7 Semantic Analysis

7.1 Semantic Analyser

After the sentence has been aligned and mapped to the metalanguage, the control is passed over to a LALR parser generated using Lex-Yacc. The parser parses the obtained sentences using the rules mentioned in the parser and the parser derives the attributes present in the sentences simultaneously. Depending on the rule getting reduced within the grammar the parser assigns and maps values and attributes mentioned in the sentence with the corresponding entities(here matrices). For example consider a simplified rule:

$$sentence \rightarrow INPUT \;\; MATRIX \;\; SIZE \;\; VARSIZE$$

In the above case, during reduction of this rule, the semantic analyzer will relate the MATRIX mentioned within the rule to the size i.e. the *varSize* attribute. Using these reductions, over the rules, we finally get the required relationship maps.

8 Experiments

We have tried two types of Statistical Machine Translation, they are:

- Word Alignment between Filtered language and Bridging Language using GIZA++

- Phrase-Based Translation between Filtered Language and Bridging Language using MOSESDECODER[3]

We will discuss the results with GIZA++ in detail and that with mosesdecoder will not be discussed in detail as they are not significant due to lack of data but still mosesdecoder results show that in future this tool can be of great importance.

9 Results

9.1 Probability Mapping of GIZA++

The following figure shows a small set of probability mapping between words of filtered language to words of bridging language.
Suppose the sentence is *"Write a program to accept a 2d-array of integers with size M*N."*. Then the following table illustrates how the sentence looks in filtered language and Bridging language.

S No.	Word in Reduced Natural English	Word in Bridging Language	Probability
1	Accept	Input	1
3	2d-array	matrix	.97
4	of	of	1
5	integers	integers	1
6	size	size	1
7	varSize	varSize	1

The following sentence :
*"Write a program to input a two-dimensional array of integers of size M*N"* generated the following bridge language sentence *"input integer matrix integers size varSize."*. This happened due the fact that the term $two-dimensional$ was mapped with a probability 0.87 to integers and 0.13 with matrix hence the integer before matrix is because of that. This is an example of outlier in the mapping.

9.2 Parser Results

Running semantic analyser/parser on the above obtained bridge language gave the following parse tree. While generating the above parse tree, the parser also generated an output "integers(size_0)", suggesting that the first mentioned size attribute

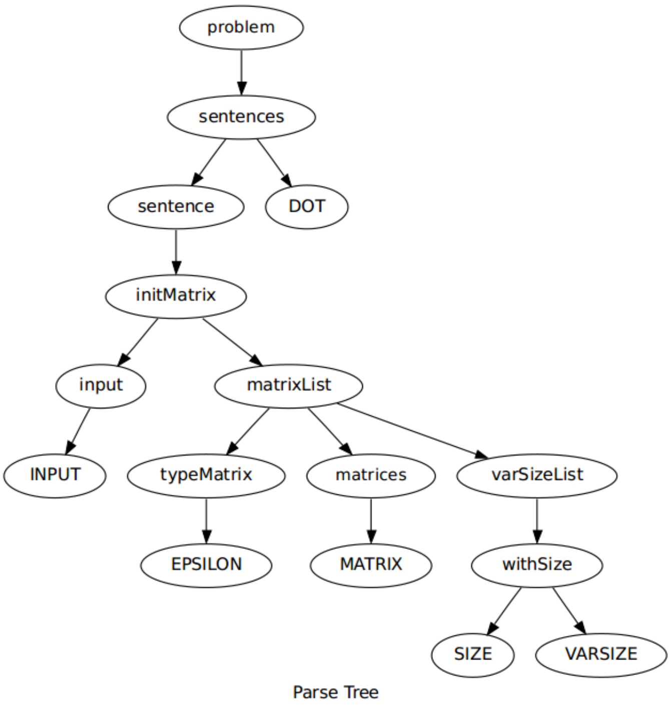


Figure 2: Obtained parse tree for the given bridge language

is linked to the mentioned matrix and is an integer matrix. The graph below explains the mentioned relationship.

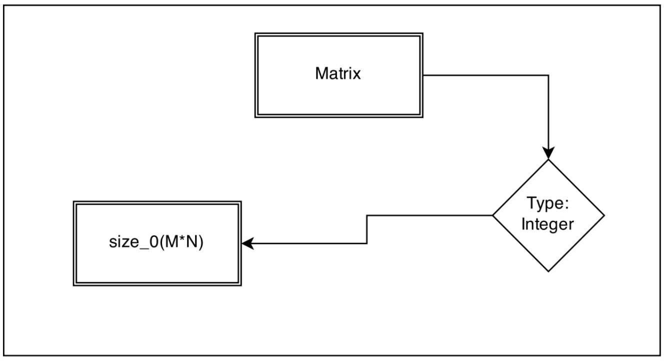
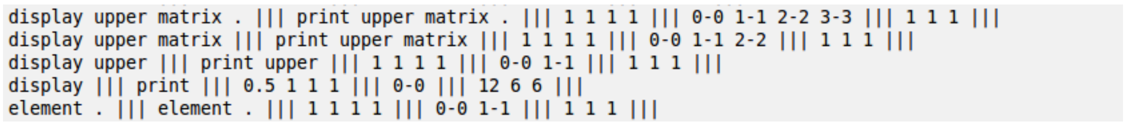


Figure 3: Relationship obtained on running semantic analyzer

9.3 Results using Moses

The results that we got using mosesdecoder which is a phrase based SMT tool were not good as it was able to map only one word statements as shown and failed for

the sentences more than 1 hence mosesdecoder can not be used for this but this is a positive indication that it can be of great use when we will have enough training corpus. Note that the language model was built using KENLM which comes inbuilt with mosesdecoder, one can as well use SRILM and IRSTLM for better results.
The picture shows the phrase-table generated by mosesdecoder for the train data.



```
display upper matrix . ||| print upper matrix . ||| 1 1 1 1 ||| 0-0 1-1 2-2 3-3 ||| 1 1 1 |||
display upper matrix ||| print upper matrix ||| 1 1 1 1 ||| 0-0 1-1 2-2 ||| 1 1 1 |||
display upper ||| print upper ||| 1 1 1 1 ||| 0-0 1-1 ||| 1 1 1 |||
display ||| print ||| 0.5 1 1 1 ||| 0-0 ||| 12 6 6 |||
element . ||| element . ||| 1 1 1 1 ||| 0-0 1-1 ||| 1 1 1 |||
```

Figure 4: Screen-shot of Phrase Table generated by mosesdecoder

10 Analysis of Results

We found that the problem of extraction of mathematical entities and their relations in a restricted domain such as programing problems can actually be solved to a great extent using these techniques. Due to closeness of bridge language to the actual English, GIZA++ is giving pretty good results on mapping and we believe that once the corpus is expanded phrase based mapping tools like mosesdecoder could be extremely helpful.

The uncertainty is only involved in the mapping part while the rest of semantic part is bound to be true given the mapping is correct because they are rule based reductions. Due to these reasons, results on small corpus are good when sentences from similar domain to that of training data are given. But due to over fitting effect, the results are poor when the problem statements deviate from training data set.

11 Future Prospects

- *Anaphora resolutions and removal:* The system maintains the current sentence context[5] in a map containing all the entities whose values and attributes(if any) has been found till the previous sentence. The parser also maintains a diff[5] set of entities which are mentioned in the new sentence but the sentence itself has not been reduced yet. For example, consider a problem in metalanguage below: matrix A size 4*3.find determinant given matrix. In the above case, in general the parser would have not known the relation between words A and given unless rewritten as A. But using the context sets the parser when encountered with given immediately looks back in the context set and assigns A as a possible candidate for the assignment find determinant.

- Currently we are having a short corpus of about 130 natural language sentences so we are using only the word based SMT i.e GIZA++, in future we hope to extend the corpus and generate the translation using mosesdecoder and also use Phrasal, the toolkit by the Stanford NLP group.

- We also hope to extend this bridge-language for other matrix entities like arrays, strings, vectors, etc.

- We are making the code public so that anybody can use it and contribute to this problem.

Bibliography

[1] Alessandra Giordani and Alessandro Moschitti. Semantic mapping between natural language questions and sql queries via syntactic pairing. In *Natural Language Processing and Information Systems*, pages 207–221. Springer, 2010.

[2] Alessandra Giordani and Alessandro Moschitti. Generating sql queries using natural language syntactic dependencies and metadata. In *Natural Language Processing and Information Systems*, pages 164–170. Springer, 2012.

[3] Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, et al. Moses: Open source toolkit for statistical machine translation. In *Proceedings of the 45th annual meeting of the ACL on interactive poster and demonstration sessions*, pages 177–180. Association for Computational Linguistics, 2007.

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

[5] Kewalramani Pankaj Prateek, Jeetesh Mangwani, Amey Karkare, Sumit Gulwani, and Amitabha Mukerjee. Anaphora without syntax-a multi-lingual approach for geometry constructions. 2014.