Int.J.HIT.TRANSC:ECCN.Vol.3: Issue 2A(1) (2017) Page 55-61 ISSN: 0973-6875

Available Online at www.hithaldia.in/locate/ECCN

ORIGINAL CONTRIBUTION

Building NLP Based Calculator

Arpan Halder, Abhirup Paria, Arindam Metya, Shatadru Senguapta* and Sourav Mandal

Haldia Institute of Technology, Haldia, West Bengal-721657

(Received Date: 20th August, 2017; Acceptance Date: 30th September, 2017)

ABSTRACT

We explore the possibilities of writing a Natural Language Processing (NLP) based calculator that can understand sentences asking it to perform a, or even a series of, mathematical calculations. Google already provides services to this effect, and a user can quite easily provide instructions in English like "two plus five". However, we aim to extend this to be able to successfully answer questions like "What is the result of ten divides five multiplies with 2?" As such, we aim to be able to make a system which would then be fully expendable with NLP systems that provide totally speech to text facilities without human intervention, benefiting audiences as wide as students learning mathematics, to executives who are unable to find the time to operate a calculator for some complicated calculation. We have worked on a possible model for this type of a system, which we present as far practicably as we can in this paper.

Key words: NLP, NLPI, NWP

1. INTRODUCTION

A word problem is a mathematical exercise which is presented as natural language texts [13]. The mathematical word problems (MWP) with basic operations (addition, subtraction multiplication and division) are a part of the elementary level. MWPs generally consist of multiple lines with a question sentence at the end. But we did not considered such multiple line text inputs to generate answer. The system takes single line input with complete background knowledge to form mathematical equations which when solved will generate final answer.

Natural Language interface (NLPI) can give us a lot of power over the way computing is conducted. For one, it is an interface that makes the use of computers comparable for people with various degrees of difficulties and various degrees of expertise in handling a computer. Reducing the need for typing complex commands, NLPI reduces unnecessary complexity in using computers. As a first step, we are implementing a calculator that can take commands in English and perform the corresponding calculation which is text-to-text. As a further step, we aim to integrate this with speech-to-text NLP, where the user will be able to speak his commands, which then will be transferred into text and then processed the same way as before.

Therefore, NLP CALCULATOR basically reads the numerical problem as a sentence, then converts the sentence into small tokens as tagger and parser. Next the output of tagger and parser are stored in a text file. After that all the important keywords are extracted from the text file to check from the database where all type of keywords and there corresponding actions are written. Using this database it will create a mathematical expression and solve it and gives us the correct result.

The similar kind of application is already implemented by Google but less efficient to diverse type as our system is. Google just implemented solution for single operation like ``Four divided by two" perfectly but less efficient for multiple operations. But our NLP calculator will solve questions with multiple operation i.e. ``four divided by two plus five" more efficiently. For the example-when we write \textit{``square of two"} on Google, It gives us the output like Figure 1.

*Corresponding Author:shatadru.sengupta@gmail.com

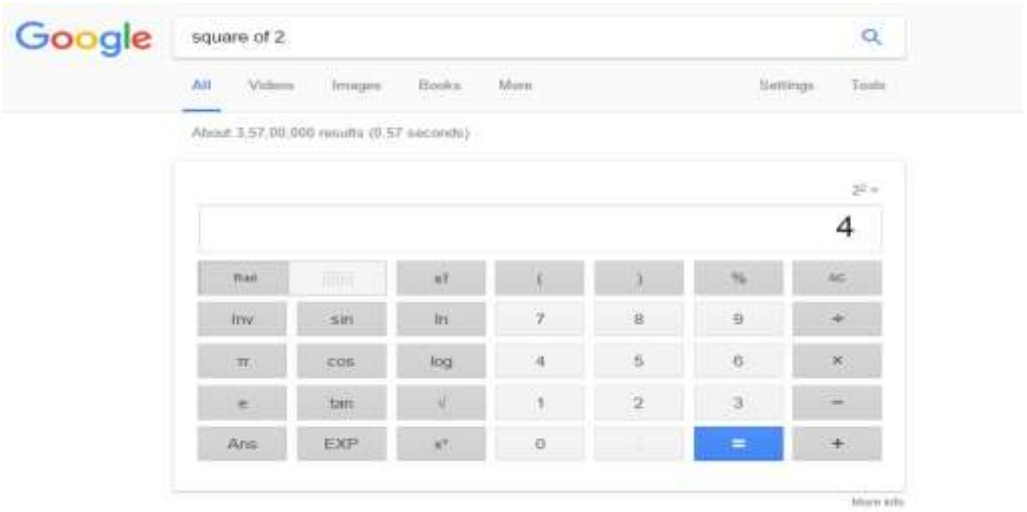

Figure 1: Output of Google NLP calculator application

As a preliminary system, we have built a calculator which is very similar to the calculator that is available with Google. When we type a command like: ``*ten divided by three''*, it gives us the decimal value of 10/3 up to a certain places of decimal (which can of course be altered). We have made a POS based system, using Java, that parses the command issued in English into Structured English, and creates a syntax tree that helps in identifying the various parts of the command. The calculator itself is a simple program that performs various arithmetic following commands issued to it. The POS has the task of taking an English command (like ``*ten divided by three''*) and translating it to a command (correspondingly, 10/3) that the calculator can understand and so that we all can use it by issuing commands directly in English.

Our paper has been arranged as follows: in Section 2, we describe the relevant research works, and in Section 3 we describe the system we have built. In Section 4, we provide the results of running it. Section 5 concludes the paper before heading on the References.

2. RELATED WORK

Solving of MWP is an interesting research problem for the researcher of cognitive science, psychology, and education. [2] identified the MWP as computer solvable. He developed a system named `STUDENT' which was the first try of natural language communication to solve a wide range of algebra word problems. Thereafter, in the decade of 1980 some advancement were done by [6, 10]. `WORDPRO' ([4]), ARITHPRO ([3]) were remarkable systems which were relevant human cognition behavior. [1] developed a simulation program `ROBUST' which solved free-format multi-step arithmetic word problems with irrelevant information.Some other remarkable research works for solving such word problems in recent times are—Equation template based [8], Equation expression tree based [7, 12], Entity, object based [5, 10], and Tag-based [9]. Our objective is not solve complete MWPs like them, rather to solve the single sentence fragments or question sentences containing mathematical queries.

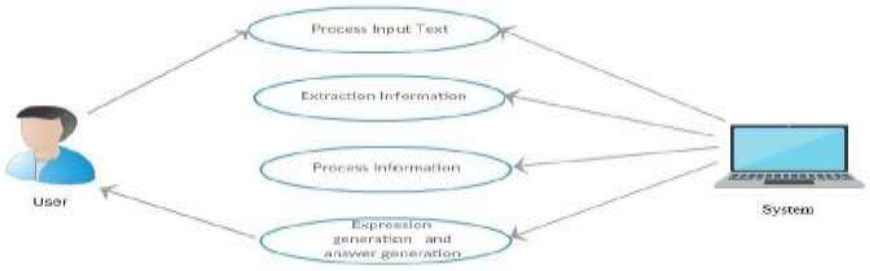

Figure 2: Use-case for the System

3. SYSTEM COMPONENTS

NLP calculator is basically an application software using NLP that solves small mathematical problems (e.g. finding the square root of four). It takes input from the user in simple English, and then processes it into the system and gives the right answer. Figure 2 presents the overall system components.
We have designed a database table (cf. Table 1 to store the important information extracted from the input sentence.

Table 1: Operation index table from NLP text clue(s)

Keyword	Expression
ADDITION	+
DIVIDE	/
DIVIDED	/
MULTIPLICATION	*
MULTIPLIED	*
MINUS	-
PLUS	+
SUBTRACTED	-
SUM	+

3.1 Algorithm

Implementation algorithm is given below:
STEP 1:- Input the sentence.
STEP 2:- POS tagging and Parse the sentence.
STEP 3:- Information extraction in preset templates along with the numbers and the keywords.
STEP 4:- Find operation(s) from the predefined operation table based on extracted keywords and information.
STEP 5:- Generate a mathematical equation using predefined equation template and solve to get the desired result.

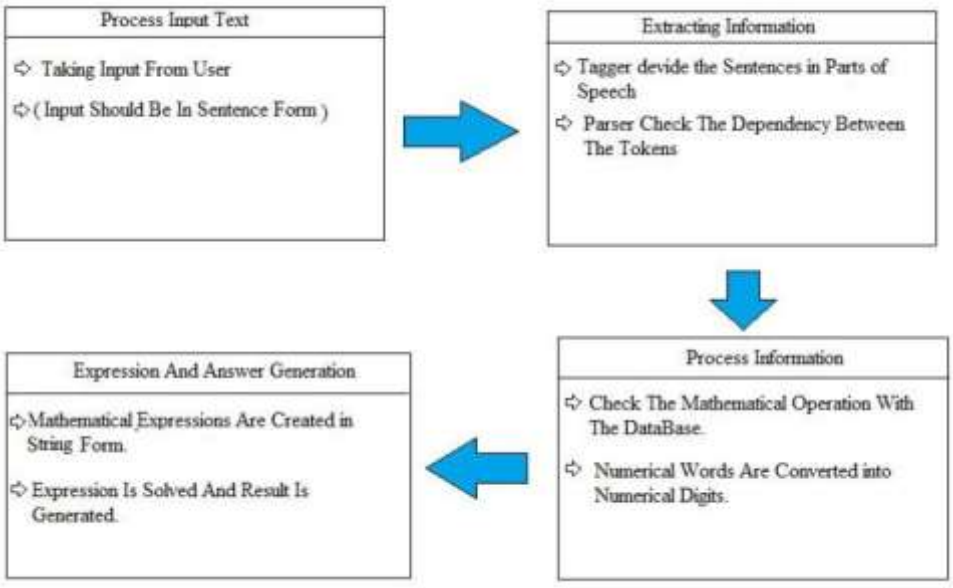

Figure 3: System work-flow

3.2 POS tagging and Parsing

POS tagging is a process deciding what is the type of every token from a text, e.g. NOUN, VERB, DETERMINER, etc. Token can be word or punctuation. Meanwhile shallow parsing or chunking is a process of dividing a text into syntactically related group. We used Stanford CoreNlp[1] toll for the tasks. For e.g., the sentence like ``Square root of four' are tagged as ``Square/NNP root/NN of/IN four/CD" and parsed as ``root (ROOT-0, Square-1) dep (Square-1, root-2) case (four-4, of-3) nmod (root-2, four-4)".

3.3 Equation Formation

Equation are formed based on extracted information and the operation which is rule-based. Using predefined equation template slot filling the equations are simply formed by string concatenation operation. Which is further executed with the help of JavaScript engine to generate final answer.

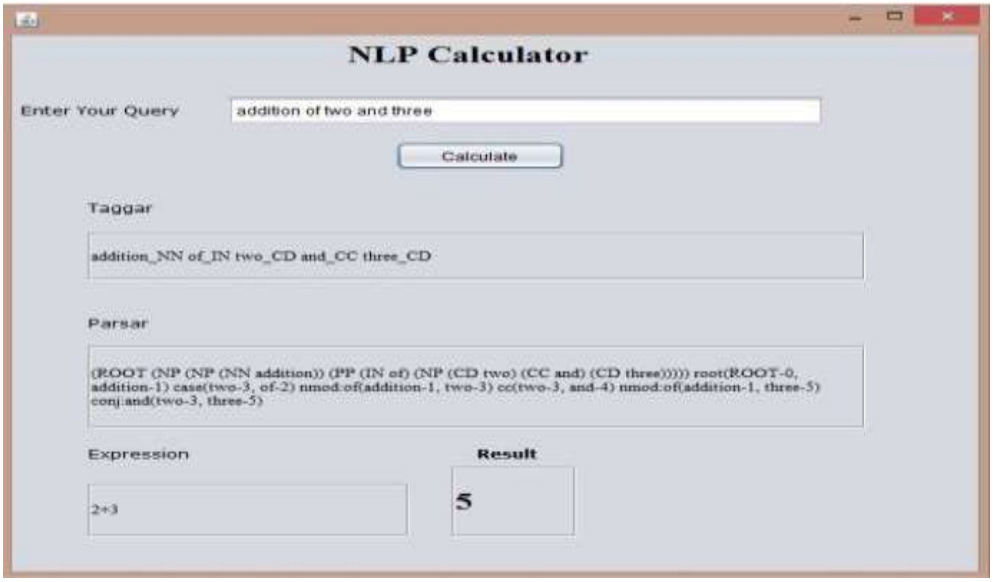

Figure 4: Homepage and solving “addition of two and three”

4. STANDARD DATASETS AND PERFORMANCE EVALUATIONS

We developed a dataset with 100 such problems containing the basic clue(s) of operation on which our system got above 90% accuracy. It used rule-based approach which has a basic problem in terms of generalization. Presently the system can not handle complex problems or the problems with ambiguities. Some of the figures of our system are given with the descriptions. Figure 4 is the example of a homepage of our nlp calculator where a user can give a small mathematical problem in the text field. It will process the mathematical problem and give us the desired output in the result section with the outputs of intermediary steps. For an example, if a user gives an input as *``addition of two and three"*, then it will give 5 as a result.

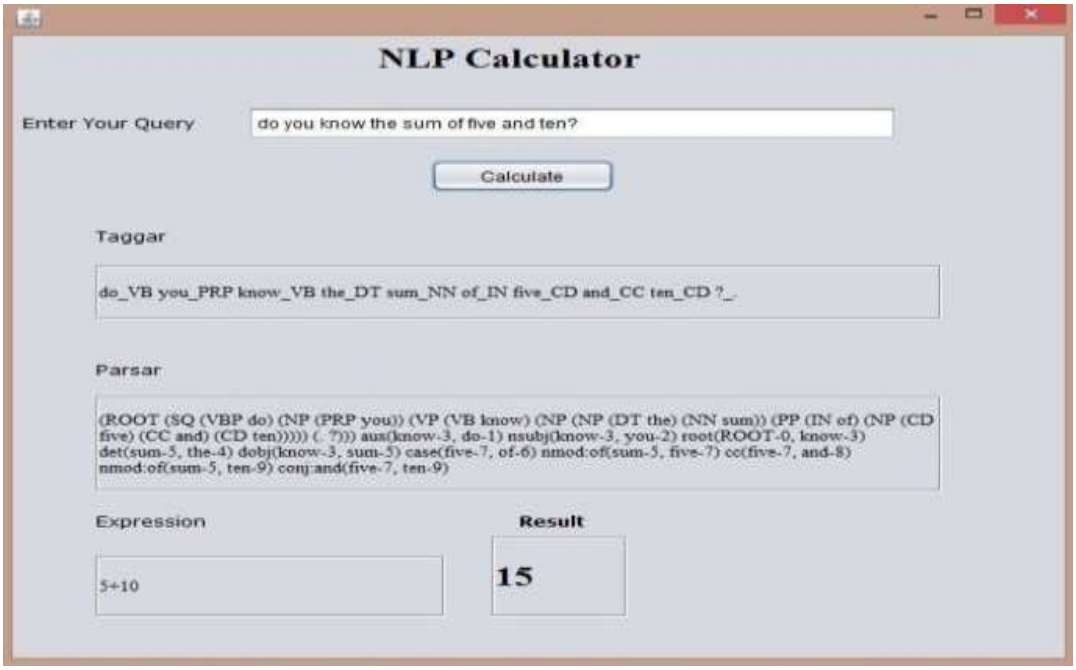

Figure 5: Solving the query " Do you know the sum of five and ten?"

Also nlp calculator can understand human level language to generate the answer (cf. Figure 5). The system also solves multiple queries (cf. Figure 6).

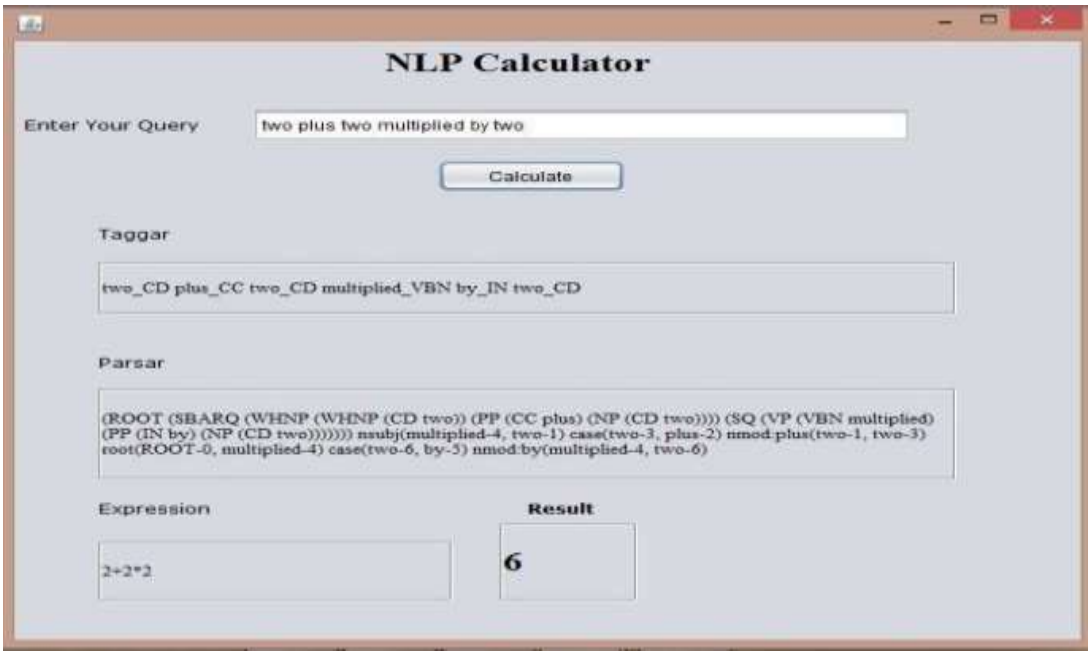

Figure 6: Solving the query with multiple operations

5. CONCLUSION

This paper is a demonstration of the possible ways in which we can convert natural language into solvable arithmetic equation. We select a sentence or a fragment of sentence which contains complete mathematical background for equation formation. The system understands all relevant numbers, keywords and the associated operations. Natural Language (NL) generation and analysis could revolutionize our individual, institutional, and national ability to enter, access, summarize, and translate textual reformation. It can make interaction with machines as easy as interaction between human individuals. Although, a computer's linguistic proficiency may never be as great as a human. However, the existence and use of current NL products and cited market projections suggest that investment in this technology should lead to useful spin-offs in the near future. As an immediate extension we would like to extend the work with more functionalities eliminating the present constraints.

References

[1] Bakman, Y.: Robust understanding of word problems with extraneous information. arXiv preprint math/0701393 (2007).

[2] Bobrow, D.G.: Natural language input for a computer problem solving system (1964)

[3] Dellarosa, D.: A computer simulation of childrens arithmetic word-problem solving. Behavior Research Methods, Instruments, & Computers 18(2), 147–154 (1986)

[4] Fletcher, C.R.: Understanding and solving arithmetic word problems: A computer simulation. Behavior Research Methods 17(5), 565–571 (1985)

[5] Hosseini, M.J., Hajishirzi, H., Etzioni, O., Kushman, N.: Learning to solve arithmetic word problems with verb categorization. In: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, October 25-29, 2014, Doha, Qatar, A meeting of SIGDAT, a Special Interest Group of the ACL. pp. 523–533 (2014), http: //aclweb.org/anthology/D/D14/D14-1058.pdf

[6] Kintsch, W., Greeno, J.G.: Understanding and solving word arithmetic problems. Psychological review 92(1), 109 (1985)

[7] Koncel-Kedziorski, R., Hajishirzi, H., Sabharwal, A., Etzioni, O., Ang, S.D.: Parsing algebraic word problems into equations. TACL 3, 585–597 (2015), https://tacl2013.cs.columbia.edu/ojs/index.php/tacl/article/view/692

[8] Kushman, N., Zettlemoyer, L., Barzilay, R., Artzi, Y.: Learning to automatically solve algebra word problems. In: Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics, ACL 2014, June 22-27, 2014, Baltimore, MD, USA, Volume 1: Long Papers. pp. 271–281

[9] Liang, C., Tsai, S., Chang, T., Lin, Y., Su, K.: A meaning-based english math word problem solver with understanding, reasoning and explanation. In: COLING 2016, 26th International Conference on Computational Linguistics, Proceedings of the Conference System Demonstrations, December 11-16, 2016, Osaka, Japan. pp. 151–155 (2016), http: //aclweb.org/anthology/C/C16/C16-2032.pdf

[10] Mandal, S., Naskar, S.K.: Towards generating object-oriented programs automatically from natural language texts for solving mathematical word problems. In: Natural Language Processing and Information Systems - 22nd International Conference on Applications of Natural Language to Information Systems, NLDB 2017, Li`ege, Belgium, June 21-23, 2017, Proceedings. pp. 222–226 (2017), https://doi.org/10.1007/978-3-319-59569-6_ 26

[11] Riley, M.S., et al.: Development of children's problem-solving ability in arithmetic. (1984)

[12] Roy, S., Roth, D.: Solving general arithmetic word problems. In: Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, EMNLP 2015, Lisbon, Portugal, September 17-21, 2015. pp. 1743–1752 (2015), http://aclweb.org/ anthology/D/D15/D15-1202.pdf

[13] Verschaffel, L., Greer, B., De Corte, E.: Making sense of word problems. Lisse Swets and Zeitlinger (2000)