International Journal on Recent and Innovation Trends in Computing and Communication ISSN: 2321-8169
Volume: 4 Issue: 6 640 – 643

Processing Natural Language and grading it using Artificial Intelligence

Shiv Kumar Goel
Assistant Prof/Deputy HOD,
MCA Department,
VESIT , Chembur,
Mumbai, India.
shivkumar.goel@ves.ac.in

Ricky Yadav
MCA Department,
VESIT , Chembur,
Mumbai, India.
rickyy465@gmail.com

Abstract: Online Test are conducted now a days to evaluate a person's knowledge regarding a subject. The best way to evaluate a person's understanding regarding a subject is to let him answer the question in his natural language. But the situation is this that we don't have such systems which can examine a users natural language answers.
Hence, I propose a unified neural network architecture approach that can be applied to various natural language processing tasks including part-of-speech tagging, chunking, named entity recognition, and other tasks. Natural language processing is an area of *artificial intelligence* that defines a set of methods and techniques used to automate the translation process between computers and humans or mediate the human-machine communication. Also we will be grading the answer given by the user, so the refined output of NLP will be given for further processing. Once the overall processing of the answer is done we will get the grades out of 10 by applying some formulas to the passage, sentences and there words.

Keywords - *Artificial Intelligence, Natural Language Processing, NLPTasks, Grading.*

*****

1. INTRODUCTION

The Understanding of machines or computers systems is growing day by day, they are becoming smarter and smarter with the help of humans. Technologies like text to speech or speech to text, Gesture Input and output etc. But there is still a field where the distance between the machines and humans is not decreasing i.e processing of Natural Language uttered by humans.

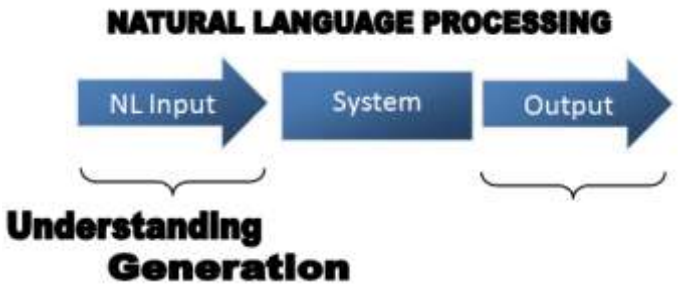
There are many online websites and institutes which Provide Online Assessment Test. These Test can have a verity of type of Questions which people have to answer to assess themselves, like MCQ's (Multiple Choice Questions), Match the Pair, Online Program Compiler and evaluator. But there is no Website which can assess on Answers given by him/her in his natural language.

Following is a Series of NLPTasks and Other Steps taken to generate an Algorithm which will grade the user according to the answers given by him in his natural language and related to specific topic.

2. Natural Language Processing

The field of Natural Language Processing (NLP) aims to convert human language into a formal representation that is easy for computers to manipulate. Current end applications include information extraction, machine translation, summarization, search and human computer interfaces. While complete semantic understanding is still a far distant goal, researchers have taken a divide and conquer approach and identified several sub-tasks useful for application development and analysis. These range from the syntactic, such as part-of-speech tagging, chunking and parsing, to the semantic, such as word sense disambiguation, semantic-role labeling, named entity extraction and anaphora resolution.

Currently, as far as the functionality of NLP is concerned we will be using the various NLPTasks and Other Custom Algorithm to for achieving the Goal.



3. NLPTasks

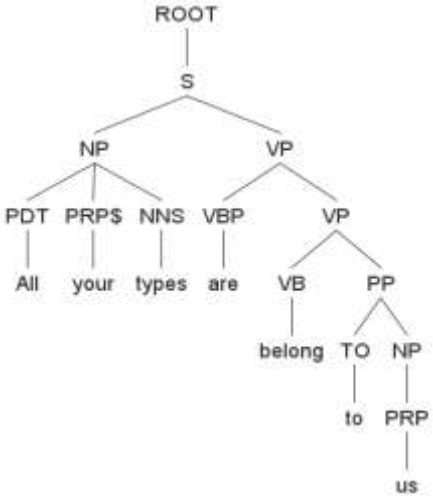
In this section we will be dealing with various NLPTasks which we can use for solving our purpose.

3.1. Part-Of-Speech Tagger

A Part-Of-Speech Tagger (POS Tagger) is a piece of software that reads text in some language and assigns parts of speech to each word (and other token), such as noun, verb, adjective, etc., although generally computational applications use more fine-grained POS tags like 'noun-plural'.
Following are some of the Labels for the Parts of the speech:

1. CC Coordinating conjunction
2. CD Cardinal number
3. DT Determiner
4. JJ Adjective
5. JJR Adjective, comparative
6. JJS Adjective, superlative
7. NN Noun, singular or mass
8. NNS Noun, plural
9. NNP Proper noun, singular
10. NNPS Proper noun, plural
11. PDT Predeterminer
12. POS Possessive ending
13. PRP$ Possessive pronoun
14. RB Adverb
15. RBR Adverb, comparative
16. RBS Adverb, superlative

17. UH Interjection
18. VB Verb, base form
19. VBD Verb, past tense
20. VBG Verb, gerund or present participle
21. VBN Verb, past participle
22. VBP Verb, non3rd person singular present



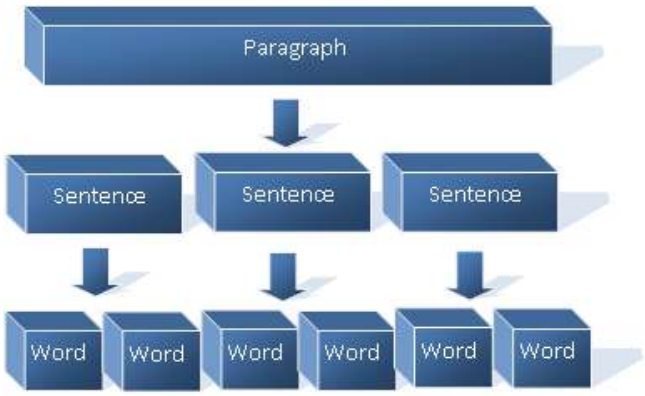
3.2. *Chunking Information*

Chunking refers to an approach for making more efficient use of short-termby grouping information. Chunking breaks up long strings of information into units or chunks. The resulting chunks are easier to commit to memory than a longer uninterrupted string of information.

3.2.1. Chunking Process:

1. Break larger amounts of information into smaller units.
2. Identify similarities or patterns.
3. Organize the information.
4. Group information into manageable units.

Also called shallow parsing, aims at labeling segments of a sentence with syntactic constituents such as noun or verb phrase (NP or VP). Each word is assigned only one unique tag, often encoded as a begin-chunk (e.g. B-NP) or inside-chunk tag (e.g. INP).



3.3. Named Entity Recognition:

Named-entity recognition (NER) (also known as entity identification, entity chunking and entity extraction) is a subtask of information extraction that seeks to locate and classify elements in text into pre-defined categories such as the names of persons, organizations, locations, expressions of times, quantities, monetary values, percentages, etc.

4. Other Tasks

4.1. Confidence of an Entity (TextRazor):

Calculating the Confidence of the paragraph i.e the entity about which the whole paragraph is based on. According to the confidence score of the TextRazor API we can judge that whether the Paragraph is related to our topic or not. Top three confidence values will be taken from the API and will be matched with the key words from the data saved regarding the subject in the database. If minimum of 2 keywords match with the three entities we can say that the paragraph is related.

Example:

In this example the score for java is the highest among all entities we can say that the paragraph talking about java.

4.2. Selection from any one of the Three Samples:

For answers to be Graded we would require the user's answers to be matched with the sample answers stored in the database. This needs to be pre feeded into the system by the Test Question setter.There will be 3 sample answers from which we will be matching the user's answer at pretty basic level of chunking. The answer with maximum number of matching will be selected for further processing.

4.3. Assigning weightage to the keywords(chunks):

A weight will be assigned to the chunks or keywords and saved in the tree like data structure. The weightage will be out of ten. The weight will be according to the importance of the keyword.This needs to be pre feeded into the system by the Test Question setter.

For example:

Java is an object oriented programming language.

Keyword	Weight
Java	7
ObjectOriented	9
Programming	6
Language	5

 ISSN: 2321-8169
640 – 643

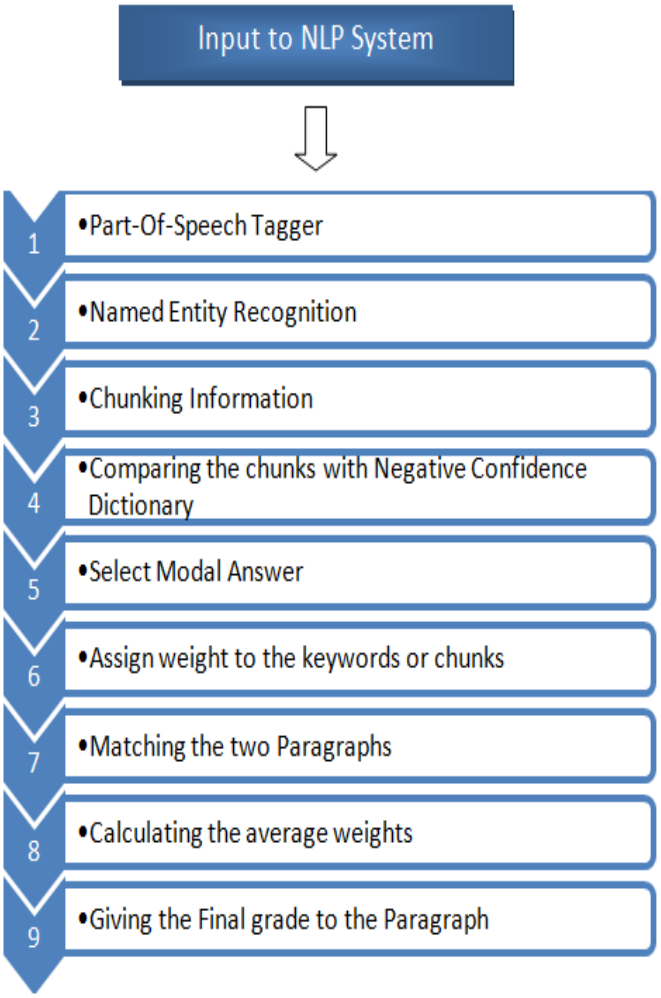
4.4. Negative Confidence Dictionary:

A Dictionary of words will be created with negative confidence. Words like not, neither, nor etc. This dictionary can also be the antonyms of the subjects. This needs to be pre feeded into the system by the Test Question setter.

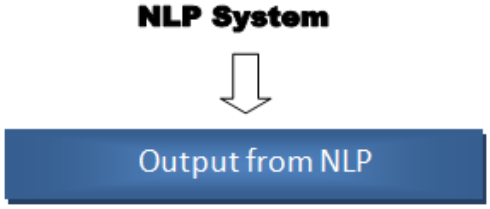
Java is not an object oriented programming language. Java is a Structured programming language.

Negative Dictionary
Not
Structured

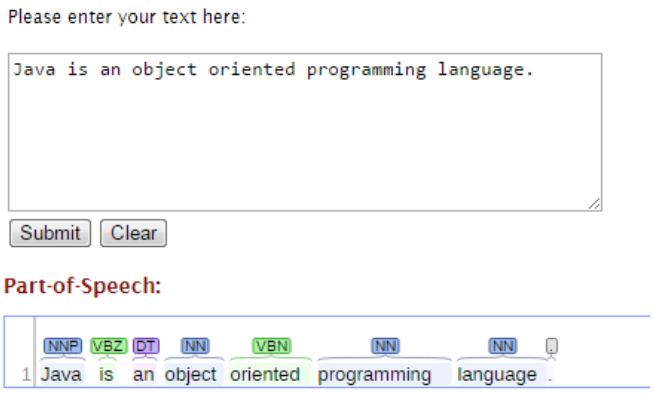
5. *Steps (Algorithm):*



5.1. Step 1(Parts of speech):



Checking the Parts of speech using the Stanford Core NLP. This will conclude that the sentence is in proper formation.



5.2. Step 2(Named Entity Recognition):

Named entity recognition is functionality which will find out the major entities in the sentence.

For Example:

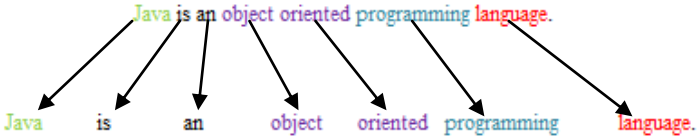
Java is an object oriented programming language.

Entities	Topic
Java	Programming Language
Object Oriented	Programming Language Type
Programming	Language type
Language	Medium to communicate

5.3. Step 3(Chunking Information):

Chunking is a technique which will segregate the paragraph into smaller parts. The paragraphs will be converted to sentences and sentences will be converted to words.

For Example:



5.4. Step 4(Comparing the chunks with Negative Confidence Dictionary):

In this stage we compare the negative confidence Dictionary words with the words in the sentence of the user. If the words match the whole sentence will be neglected from further processing. Like this we will eliminate the false sentence regarding the subject.

For Example:

Java is not an object oriented programming language. Java is a Structured programming language.

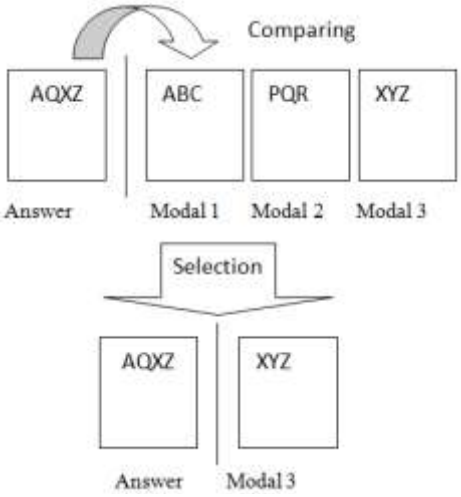
Negative Confidence
Not
Structured

Here the two negative Confidence words are present in the sentences, so the two sentences will be ignored and weights will be calculated from it.

5.5. Step 5(Select Modal Answer):

There will be sample answers in our database previously saved for the question. From the three answers we will choose one of them for further processing.

For Example:



In this Example modal answer 3 is selected for further processing.

5.6. Step 6(Assign weight to the keywords or chunks):

Weights are assigned to the keywords and saved in the database. The Database will be Pre populated with keywords and its associated weight according to the importance of the word in the sentence.

For example:

Java is an object oriented programming language.

Keyword	Weight
Java	7
Object Oriented	9
Programming	6
Language	5

5.7. Step 7(Matching the two Paragraphs i.e user's Answer & modal answer):

Matching the two paragraphs i.e answer given by user and modal answer. All the chunks with different weight which are matching will be added to a variable total weight variable. Also a counter will also be used to determine the number of chunks found.

Note: A word once used in a sentence will counted only once in the whole paragraph to eliminate the repetition of words.

Total Weight = (Addition of all weight of the chunks (keywords) found).

Chunk Counter = Count of all the chunks or keywords found.

5.8. Step 8(Calculating the average weights):

Total Weight of all the chunks will be divided from the count of chunks to get the average weight of the passage.

$$\textbf{Average Weight:} \quad \frac{\text{Total of weights of all Chunks}}{\text{Chunk Count}}$$

5.9. Step 9(Giving the Final grade to the Paragraph):

Final grade of the paragraph will be calculated by dividing the average weight with 10.

$$\textbf{Final Grade:} \quad \frac{\text{Average Weight}}{10}$$

6. CONCLUSION

In this paper, we addressed the topic of NATURAL LANGUAGE PROCESSING in Subjective Answering during Online Test. We discussed various NLPTasks and other methods using which we can use to check if the answers given by the layman user is grammatically correct, is subject related and we can Grade them accordingly. Grading will be done according to the weights provided to the system by the Paper setter.

From this we can conclude that we can give grades to the user according to the answer he provides. But the grades will not be 100 percent genuine or correct .So will it ever be genuine? The answer lies in this question -

"Will a computer program ever be able to convert a piece of English text into a programmer friendly data structure that describes the meaning of the natural language text?"

Unfortunately, no consensus has emerged about the form or the existence of such a data structure. Until such fundamental Artificial Intelligence problems are resolved, computer scientists must settle for the reduced objective of extracting simpler representations that describe limited aspects of the textual information.

7. REFERENCES

[1] http://nlp.stanford.edu/
[2] https://www.textrazor.com/
[3] http://nlp.stanford.edu:8080/corenlp/process
[4] http://www.cis.strath.ac.uk/cis/research/publications/papers/strath_cis_publication_320.pdf - by Gobinda G. Chowdhury Dept. of Computer and Information Sciences University of Strathclyde, Glasgow.
[5] https://sergey-tihon.github.io/Stanford.NLP.NET/