International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395 -0056
Volume: 04 Issue: 04 | Apr -2017 www.irjet.net p-ISSN: 2395-0072

EQC: A Novel Approach to applying NLP in Academic Applications

Kanak Sharma[1], Ashish Sharma[2], Mr. Dhananjay Joshi[3]

[1] B.Tech C.E. SVKM's NMIMS, MPSTME, India
[2]B.Tech C.E. SVKM's NMIMS, MPSTME, India
[3]Asst. Prof., Dept. of Computer Engineering, SVKM's NMIMS, MPSTME, India

---

Abstract - Students spend days and nights studying rigorously for examinations following a repetitive futile process. A smart approach to examinations often fetches better results instead of dumb brute force work. Referring to previous year question papers is necessary to better understand what is asked in the exam paper and as a result the students can prepare smartly. But this process is inefficient and very time consuming. Hence a smart and knowledgeable system is needed. With the recent research in Artificial Intelligence and Natural Language Processing, today it is possible to automate such tasks and design easy to use solution for the students of world. EQC is a new and novel entrant in the class of educational software. Its main objective is classifying and analysing contents of examination question papers and aiding test takers as well as test setters in achieving the most out of the education evaluation system and by extension the education system.

***Key Words*: Linguistics, educational, software, questions, classification, analysis.**

1.INTRODUCTION

All Natural Language Processing is a modern and disrupting technology of the modern era. It traces its roots from Artificial Intelligence and machine learning, both of which have been around since a long time back but recently with more developed computing capabilities have gained more popularity. With the computers becoming day by day more closely related to human beings and working with humans interchangeably, it has been never more urgent to recognize the language that humans speak the natural language. Natural Language Processing is a collection of algorithms and techniques to basically recognize what humans say in their natural way. Natural Language Processing Is defined as a domain of computer science in which the algorithms and techniques are used to comprehend and create natural language. Major areas of NLP are to automate classification and pattern discovery in electronic documents and structured and unstructured data. Some widely known examples of natural language processing virtual assistants like Google Now, Siri and Cortana, recognize the language that humans speak and convert them to a textual data and then use it for further process and these all are dependent on a very basic process called as voice recognition in which human voice in the form of audio is converted into text.

2. PURPOSE

After reading, summarizing, categorizing and contemplating upon research work done in the field of Natural Language Processing with primary focus on textual mining/ analysis around academia we feel that the majority of research being done is consumer focused. The textual data being collected is produced by consumers either on social media by users, or by test-takers, in case of examinations. Hence, there should be more research focused on using text mining and NLP techniques to analyze the hosts of this textual data, like Facebook or Twitter or TOEFL. Various social media alerts automatically generated by recommendation algorithms these companies employ. And as far as examination question answer grading is concerned, an analysis of the question paper itself could be done, this can be used to find out how efficient a system is, such as the Friends' posts' recommendation notifications given by Facebook could be improved via sentiment analysis of users posts or a complete Bloom's Taxonomical evaluation of major competitive exams like SAT, TOEFL, IELTS and GRE can be done and the efficiency of them mapping to the careers of their respective test takers can be calculated. Hence, what we would coin as "organizational text mining" is currently the need of the hour, whether it is related to data produced by large social media corporations or other large organizations and bodies, what it says about them as opposed to what it does about their users, needs to be focused on and researched with greater interest and more attention.

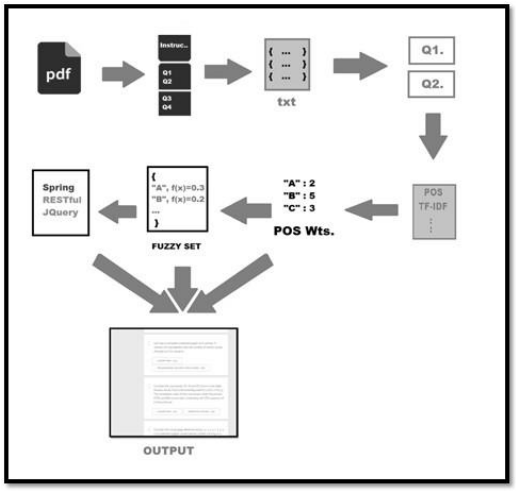
3. SCOPE

EQC is a new and novel entrant in the class of educational software. Same methodologies and techniques have been used recently in applications in the industry, but for examination trend analysis, this is a one of a kind. The EQC project, at the time of writing this report is already far ahead in its development cycle since it bypassed the primary objectives which were set for it that included classification of a GATE question paper to a single subject. The planned development did not include a Web interface, fuzzy logic, a beautiful User Interface, Optical Character Recognition powered by Google's Tesseract engine, support

for any type of question paper. The algorithm has been optimized for working even for generic question papers. Such an academic based project has a lot of scope for further improvement and can to contribute a lot to all the stages in academic evaluation right from syllabus setting to question paper formulation to analysis of previous year question papers by students to determine what the important topics that need to be studied thoroughly are.

4. IMPLEMENTATION

The implementation of the system is a basic three tier architecture. The first tier, i.e. the client is built on JavaScript which runs on the browser. The front and has been provided in the form of a website. The second tier, i.e. the server is a simple Java Virtual Machine, JVM which can run on a platform independent physical server the third tier slept. The third tier i.e. the database is a simple MySQL database which can again be hosted on any underlying operating system.



4.1 The First Tier

The user first uploads a question paper through the front end website. These question paper goes to the server where questions are parsed out of the question paper and then put into the core EQC system. The system tests for the results in the database or using the tf-idf mechanism and then sends the results in the form of response back to the website.

4.2 The Second Tier

The Core system is a java application which uses the spring framework supplied through the model view controller (MVC) Design pattern. The Spring framework does a fantastic job in getting things easy and correctly configured in the system. Selection of programming language is completely team dependent. One can implement such methodology in any language he/she is comfortable in. Basically the system can be established using either POS Tagging or TF-IDF or both. Part of speech tagging is done using Natural Language Processing tools for tagging each word in a particular question using a part of speech tagger which helps to determine what kind of language construct a word 'i' is. For example "While running DOS on a PC, which command would be used to duplicate the entire diskette?" is tagged as:

While/IN running/VBG DOS/NNP on/IN a/DT PC/NN, /, which/WDT command/NN would/MD be/VB used/VBN to/TO duplicate/VB the/DT entire/JJ diskette/NN?/

Here, all the acronyms written after '/' are Penn Treebank P.O.S. Tags. The list includes tags such as:

- IN - Preposition.
- VBG - Verb, gerund or present participle.
- DT - Determiner.
- WH- Wh-determiner.
- NN - Noun.
- MD - Modal.
- VB - Verb.
- JJ - Adjective.

4.3 The Third Tier

Until now we had extracted all the important words out of the questions. Now the important words are used by the system to find out that the question belongs to which subject in a particular domain of examination. For example a successful output of the question "What is a role of a router in LAN?" will be tagging the question with the subject 'computer networks' in the domain computer science. This is achieved by rigorously querying the database.

The database is stored with the Corpus of the syllabus of every subjects in a given domain. For example for the domain of computer science each subject is stored and related to that each topic of that subject is also stored. Basic structure of the database can be the following:-

DESCRIBE subjects;

	Field	Type	Null	Key	Default	Extra
1	sID	int(11)	NO	PRI	<null>	auto_increment
2	sName	varchar(100)	NO		<null>	

DESCRIBE topics;

	Field	Type	Null	Key	Default	Extra
1	tId	int(11)	NO	PRI	<null>	auto_increment
2	Topics	varchar(500)	YES		<null>	

DESCRIBE mappings;

	Field	Type	Null	Key	Default	Extra
1	mappin...	int(11)	NO	PRI	<null>	auto_increment
2	tId	int(11)	YES	MUL	<null>	
3	sid	int(11)	YES	MUL	<null>	

The Associated Data Definitions are:-

```
CREATE TABLE mappings (
    mappingID INT PRIMARY KEY NOT NULL
AUTO_INCREMENT,
    tId INT,
    sid INT,
    CONSTRAINT mappings_ibfk_1 FOREIGN KEY (tId)
```

```
REFERENCES topics (tId),
    CONSTRAINT mappings_ibfk_2 FOREIGN KEY (sid)
REFERENCES subjects (sID)
  );
  CREATE TABLE subjects (
    sID INT PRIMARY KEY NOT NULL
AUTO_INCREMENT,
    sName VARCHAR(100) NOT NULL
  );
  CREATE TABLE topics (
    tId INT PRIMARY KEY NOT NULL
AUTO_INCREMENT,
    Topics VARCHAR(500)
  );
```

Thus for different domains a separate corpus is designed for storage and further usage by the core system.

We used MySQL database management system which sufficiently provides tools and functionalities to easily and effectively design database. Further MySQL is designed for quick and smooth data reading and writing which speeds up the core system.

5. WORKING

Thus, finally the user has uploaded the paper, the system has extracted questions from it and filtered out the stop words and now it has the important words with it and off course, the database is well designed with the subject Corpus fed into it well in advance. Now recursively each word of the question is taken and fired against a SQL Query which queries out the respective subject that this important word is mapped to in the database. Subsequently for n important words in the question we get n respective subject mappings of that important keyword. And clearly and very simply the subject that has the highest frequency of the mappings found out is the most probable subject category that this question falls into. For example, the system executes and results the final output in the form of a fuzzy set as:

The problem of determining whether there exists a cycle in an undirected graph is in?

```
{"Algorithms":0.3333333333333333,"Compiler
Design":0.08333333333333333,"Computer
Networks":0.25,"Discrete
Mathematics":0.16666666666666666,"Programming and Data
Structures ":0.16666666666666666}
```

Here each numeric digit is its fuzzy membership value of the respective member of the fuzzy set f(x).

$$f(x) = \frac{W_x}{\sum W_i}$$

Here the numerator is the priority based weight of item x of the fuzzy set and the denominator is the sum of all weights of all items. Here, fuzzy logic has been implemented in a way as to find the probability of each subject. And the whole question paper is then visualized using eye catching and intuitive pie charts and other data visualizations methods.

Term frequency - Inverse document frequency or TF-IDF weight is a weight often used in information retrieval and text mining. This weight is a statistical measure used to evaluate how important a word is to a document in a collection or corpus. Term Frequency, measures how frequently a term occurs in a document. Since every document is different in length, it is possible that a term would appear much more times in long documents than shorter ones. Thus, the term frequency is often divided by the document length as a way of normalization:

$$\mathrm{TF(t)} = \frac{n_k}{\sum n_i}$$

Inverse Document Frequency measures how important a term is. While computing TF, all terms are considered equally important. However it is known that certain terms, such as "is", "of", and "that", may appear a lot of times but have little importance. Thus we need to weigh down the frequent terms while scale up the rare ones, by computing the following:

$$\mathrm{IDF(t)} = \ln \frac{N}{n_t}$$

Then the TF-IDF is computed by summing the TF-IDF for each query term; many more sophisticated ranking functions are variants of this simple model.TF-IDF can be successfully used for stop-words filtering in various subject fields including text summarization and classification. Both of these approaches are used only to remove the stop words from the Corpus. Because in the natural language there are so many words that are useless for the algorithm we need to perform either of these two steps as preprocessing. Moreover it adds to the noise when the natural language from the PDF of the document is applied with optical character recognition or any basic text extraction software. search methods to extract all the text that is extractable from the input source. For example if the system is given PDF scanned document as input then a basic optical character recognition is applied to the PDF which results into a very noisy data output through which the key question statements are to be taken out. This is where Term Frequency – Inverse Domain Frequency (TF-IDF) and Part of Speech (POS) tagging approaches come into the picture. POS tagging simply works by entering the language construct and by that method majority of the punctuation marks and stop words like in, there, his, of, the, a etc. are detected and thereafter removed- accordingly.

The TF-IDF approach, on the other hand, works statistically and it finds more important word in the given document out of the given set of documents. What is more important if it is used the basic element of TF-IDF approach is counting words in the document and the frequency of occurrence of that word in the set of documents.

6. RESULTS

To analyze the given question paper we have used the confusion matrix. Confusion matrix is a statistical measure to assess the quality of categorical data. It compares the predicted value and the actual values for how many are correctly classified and how many are not.

A sample data of 200 questions was taken which was labeled with the correct question category and then fed to EQC model on these sample questions and obtained the classified final values. The confusion Matrix was made with the help of R programming language which is an optimal tool for data analysis. Table 1 shows the finally obtained confusion matrix. We see that on the diagonal of the table the values are shown which are correctly classified. We can further calculate the classical values of true positive and false negative using the obtained confusion matrix. We try to understand these metrics using a classical example of disease prediction:

- True positives (TP): These are cases in which we predicted yes (they have the disease), and they do have the disease.
- True negatives (TN): We predicted no, and they don't have the disease.
- False positives (FP): We predicted yes, but they don't actually have the disease. (Also known as a "Type I error.")
- False negatives (FN): We predicted no, but they actually do have the disease. (Also known as a "Type II error.").

TABLE 1. CONFUSION MATRIX

Pred	CN	DAA	DBMS	DIS	DS	EM	OS	TCS
CN	20	00	00	00	00	00	00	00
DAA	00	30	00	00	00	00	00	00
DBMS	10	00	10	00	00	00	00	00
DIS	00	00	00	10	00	00	00	00
DS	00	00	00	00	30	00	00	00
EM	10	00	00	00	00	10	10	00
OS	00	00	00	00	00	00	30	00
TCS	00	00	00	00	00	00	00	30

Next, we go on calculating the Overall statistics of the system. We see that the classification accuracy of EQC system results out to be 85% correct. This means that if in a question paper the system is given 100 questions, our system can classify 85 questions correctly. Table 2 shown below gives the overall statistic.

TABLE 2. OVERALL STATISTICS

Accuracy	0.85
95% CI	(0.6211, 0.9679)
No Information Rate	0.2
P-Value [Acc > NIR]	7.978e-10
Kappa	0.8271
Mcnemar's Test P-Value	NA

7. CONCLUSION

This work helps to analyze the question papers in order to better prepare for the examination using keyword matching and passing techniques of core NLP. The linguistic features such as TF-IDF and POS tagging along with a corpus-based approach are used for question classification from the given input document. The Corpus used in the system can be updated according to specified domain by adding new subjects as and when needed. The POS tagging used, extracts important keyphrases which are used for classification and further we analyse the paper based on the classification. The finally obtained accuracy was 85% and we believe this can be further improved.

ACKNOWLEDGEMENT

The authors wish to acknowledge the contributions of their colleagues Nikhil Vyas and Arpit Bapna towards this paper.

REFERENCES

[1] Yasunari Maeda, Hideki Yoshida, and Toshiyasu Matsushima. "Document classification method with small training data," in Proc. ICCAS-SICE, 2009.

[2] Hao Wang and Jorge A. Castanon. "Sentiment Expression via Emoticons on Social Media" in Proc. IEEE International Conference on Big Data, 2015.

[3] Shweta Patil and Sonal Patil. "Intelligent Tutoring System for Evaluating Student Performance in Descriptive Answers Using Natural Language Processing." International Journal of Science and Research, 2014.

[4] Siddhartha Ghosh and Dr. Sameen S Fatima. "Design of an Automated Essay Grading (AEG) system in Indian Context." International Journal of Computer Application, vol.1, No.11, 2010.

[5] Deepali K. Gaikwad and C. Namrata Mahender. "A Review Paper on Text Summarization". International Journal of Advanced Research in Computer and Communication Engineering, Vol. 5, Issue 3, Mar. 2016.

[6] Tushar Ghorpade and Lata Ragha. "Featured Based Sentiment Classification for Hotel Reviews using NLP and Bayesian Classification" presented at the International Conference on Communication, Information & Computing Technology (ICCICT), Mumbai, India, Oct. 2012.

[7] Bhumika, Prof Sukhjit Singh Sehra and Prof Anand Nayyar. "A Review Paper On Algorithms Used For Text Classification". International Journal of Application or Innovation in Engineering & Management, Vol. 2, Issue 3, March 2013.

[8] Mita K. Dalal and Mukesh A. Zaveri. "Automatic Text Classification: A Technical Review", 2011.

[9] N. Moratanch and Dr. S. Chitrakala. "A Survey on Abstractive Text Summarization", in Proc. International Conference on Circuit, Power and Computing Technologies, 2016.

[10] Urmila Shrawankar and Kranti Wankhede. "Construction of News Headline from Detailed News Article", 2016.

[11] Manju Khari, Amita Jain, Sonakshi Vij and Manoj Kumar. "Analysis of Various Information Retrieval Models", 2016.

[12] B. Azvine, Z. Cui, D.D. Nauck and B. Majeed. "Real Time Business Intelligence for the Adaptive Enterprise", 2006.

[13] James Benhardus. "Streaming Trend Detection in Twitter", 2013. Benhardus, James, and Jugal Kalita. "Streaming trend detection in twitter." International Journal of Web Based Communities, pp. 122-139, 2013.

[14] Steven Bird. "NLTK: The Natural Language Toolkit", Proc. COLING/ACL on Interactive presentation sessions, pp. 69-72, 2006.

[15] Chetan Botre, Saad Patel, Shrinivas Kunjir and Swapnil Shinde. "NoteMate – A Note Making System Using OCR and Text Mining" in International Journal of Advanced Research in Computer Science and Software Engineering, Volume 5, Issue 3, Mar. 2015.

[16] https://en.wikipedia.org/wiki/Natural_language_processing (last accessed Mar 2017)

[17] https://www.tutorialspoint.com/artificial_intelligence/artificial_intelligence_natural_language_processing.htm (last accessed Mar 2017)