Mathematics Question Prediction using Natural Language Processing (NLP) (K E G O)

Mr. Piyush Thakare
Indira College of Commerce & Science
Pune, India

Mr. Kartikeya Talari
Indira College of Commerce & Science
Pune,India

Dr. Ashwini Shende
Indira College of Commerce & Science
Pune, India,

***Abstract*-Using automated techniques to roll out keywords from the Questions provided from the previous year Question Papers using a beautiful python library RAKE (Rapid Automated Keyword Extraction) and using those keywords to form pattern based upon the scores with the stop words used to the pre texts in the library for Keyword Extraction and pattern recognition for University Questions specifically for Mathematics Question Papers. Every Keyword Extracted is accompanied with the score of the keyword and numeric weightage of the chances of the Keyword to get repeated in the sample. The paper consists of the final accuracy of the model based upon the sample provided to it and the stop-list also does affect the final accuracy of the model with respect to the sample it is processing.**

Keywords-- NLP, RAKE, Keywords, Regularization, Text-Preprocessing

I. INTRODUCTION

The models in Machine Learning (ML) and Deep Learning (DL) are a common method to achieve high accuracy in obtaining patterns and the final predictions which also needs to be accurate at a certain level based on the problem. Predicting the Questions for the next test or examination is one of the key field to use the models for.

Using Keyword extracting tools like RAKE and training a NLP based Machine Learning Model to find of frequently repeated questions and how they change according to the pattern sticking to the syllabus.

II. RELATED WORK

Collect Questions of a particular Subject from a particular University were used in the following steps.

A. Text Preprocessing

Just in case for any unintended noise in the plain text can be removed by automated techniques using the following steps, which will help in removing things like~

- Remove extra whitespaces
- Convert accented characters to ASCII characters
- Expand contractions
- Check for Special Characters
- Lowercase all texts
- Convert number words to numeric form
- Check for numbers

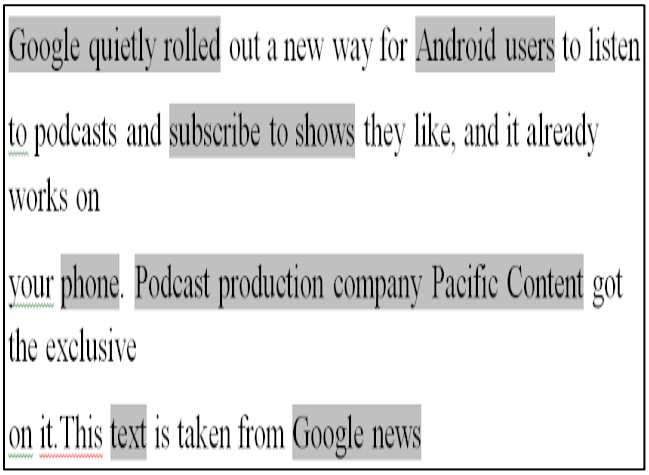

Fig 1: Text preprocessing example

```
In [3]: keywords=rake_object.run(text)
        print("keywords:",keywords)

keywords: [('following function', 4.0), ('simultaneous\nlimit', 4.0), ('iterated limits', 4.0), ('evaluate lim', 4.
0), ('using definition', 4.0), ('cos 0', 4.0), ('sin 0', 4.0), ('test', 1.0), ('existence', 1.0), ('origin', 1.0),
('exits', 1.0), ('discuss', 1.0), ('continuity', 1.0), ('= xy', 1.0), ('= -xy', 1.0), ('fx', 1.0), ('fy', 1.0), ('= 2
xy', 1.0), ('prove', 1.0), ('1', 0), ('=', 0), ('-', 0), ('+', 0), ('6=', 0), ('0', 0), ('[4]\n2', 0), ('=', 0), ('-
1\m/', 0), ('-\m/\n1 -', 0), ('[4]\n3', 0), ('≥', 0), ('<', 0), ('[4]\n4', 0), ('2 -', 0), ('2', 0), ('2 +', 0), ('=
0', 0), ('[4]\n5', 0), ('∂', 0)]
```

Keywords obtained after content processing.

B. Noise Removal

Most of the question papers collected in the form of pdf are converted to simple text for the further operations.

C. Normalization

used to convert the plain text into a more uniform sequence before feeding it to the RAKE model.

D. *Removing Redundant Text – Removing Redundant texts include removing the Header and Footer content from the RAW question papers obtained from the university and the repeated formats from the Question Papers.*

E. *Stemming – reducing the words to their original form(lemma) by removing the prefixes and suffixes.*

F. *Lemmatization – converting the words into their base forms*

After obtaining most from the text obtained from the RAW question papers of the University, we move on to putting the same texts(questions) to work with in the RAKE library.

As Keywords or entities are condensed form of the content these can be widely used to define questions and tags within the information to be processed.
Information retrieval can be done with various methods and one of the most popular is ***POS tagging*** *etc. But since overcoming manual work and converting the same to an Automated system is always a good thing which in result also increases the accuracy and decreasing time-complexity over space.*
Since the open source nature of python and RAKE adding various ways of training and testing data over the period of time is possible, in rake the fao_test and fao_train data replaced with the Questions obtained from the University Portal.

After several epochs and training prequels the model was capable to fetch meaning full keywords from the data put towards it.
Considering following example to understand the background to the work of the Training and Testing data and how it works with the RAKE library.

```
stoppath = "SmartStoplist.txt"

rake_object = rake.Rake(stoppath)
```

According to the above sample text it is noted that the keywords obtain come with the score for that particular keyword according to the algorithm.

All words listed in the SmartStopList.txt are treated as phrase boundaries for the questions and which can be easily changed from the inside text files which also include various other StopWord lists.

First, this helps obtain candidates that consist of one or more non-stopwords, such as 'compatibility,' 'systems,' 'linear constraints,' 'set,' 'natural numbers,' and 'criteria' in this text which is useful for the model to obtain the keywords from the plain text accordingly.

Second, RAKE computes the properties of each candidate obtained from the data or text given, which is the addition of all the scores given for each of its words in the process. The words are scored according to their frequency at what they repeat and the typical length of a candidate phrase in which they appear.

```python
from __future__ import absolute_import
from __future__ import print_function
import six
__author__ = 'a_medelyan'

import rake
import operator
import io

# EXAMPLE ONE - SIMPLE
stoppath = "data/stoplists/SmartStoplist.txt"

# 1. initialize RAKE by providing a path to a stopwords file
rake_object = rake.Rake(stoppath, 5, 3, 4)

# 2. run on RAKE on a given text
sample_file = io.open("data/docs/fao_test/w2167e.txt", 'r',encoding="iso-8859-1")
text = sample_file.read()

keywords = rake_object.run(text)

# 3. print results
print("Keywords:", keywords)
```

Fig 3: code snippet of rake.py

After obtaining the keywords, based on there scores now it is easy to know which keyword or concept in the set of questions has been repeated for the number of times. And due to the training data which was *completely based on the questions which were used to find out the keywords, we also get to notice the pattern at which the questions has been repeated.*

III. RESULT-

After comparing the scores obtained from the testing and the training data, the model performed at the following results

E

```
Keyword:  minimal generating sets , score:  8.66666666667
Keyword:  linear diophantine equations , score:  8.5
Keyword:  minimal supporting set , score:  7.66666666667
Keyword:  minimal set , score:  4.66666666667
Keyword:  linear constraints , score:  4.5
Keyword:  upper bounds , score:  4.0
Keyword:  natural numbers , score:  4.0
Keyword:  nonstrict inequations , score:  4.0
```

```
$ python evaluate_rake.py data/docs/fao_test/ 10
...
Precision 4.44 Recall 5.17 F-Measure 4.78
```

Precision states the percentage of correct keywords among those extracted.

Recall shows the percentage of correctly extracted keywords among all correct ones, and F-measure is the combination of both the correct keywords from the extracted and Recall values.

CONCLUSION

The final overall score obtained from the test on the model shows us the overall usage accuracy with Mathematics question paper data set. The model will be better for the successful predictions of Questions from the dataset beneficial for not only students but for the administration as well to observe the trends and pattern for various Subjects based on the keyword scores.

FUTURE SCOPE

Using the model in obtaining patterns over the time can be more precise, with more data more regularization will lead to making the model more accurate. It will be interesting to work with more datasets which will include not just texts but figures, symbols and numbers while using techniques like image to text processing it'll completely change how the model will react and how it will process the data before feeding the model.

REFERENCES

[1] https://www.airpair.com/nlp/
[2]https://towardsdatascience.com/textrank-for-[3]keyword-extraction-by-python-c0bae21bcec0
[4] https://pypi.org/project/word2number/