BIRLA INSTITUTE OF TECHNOLOGY & SCIENCE, PILANI
FIRST SEMESTER 2019 – 2020
INFORMATION RETRIEVAL (CS F469)
ASSIGNMENT

Due Date: 06/11/2019 **Total Marks: 30**

In this assignment you are required to primarily implement a **Cross Lingual Document Translator,** using Statistical Machine Translation model. Statistical Machine Translation is an empirical machine translation technique using which translations are generated on the basis of statistical models trained on bilingual text corpora.

Important note : Cases of plagiarism would be penalized by awarding zero marks.

Corpus

You are required to use the following corpus for training and validating your model:(Corpus for assignment)

Programming language

You are required to implement this assignment in **Python** only.
Although not mandatory but use of IPython Notebook is recommended, with effective use of Markdown cells, for explaining the code.

Guidelines

- If any data cleaning task is required on the given corpus it can be performed, along with **proper justification** in the design document for the steps taken.
- A statistical model has to be trained for alignment and translation. You are required to **implement your own IBM model and Expectations Maximization (EM) algorithm** covered during the class. No external libraries should be used in this step.
- You should be able to output the results upon request. You will be given few test cases at the time of your demo. Each test case would be foreign/English language document. The translator must generate the corresponding English/foreign translation for the same. (You can be asked to translate in **either direction**.)
- The generated translation would then be compared with an accurate translation using performance metrics such as **cosine similarity** and **Pearson's correlation coefficient**, which is supposed to be implemented inherently into your translator. For this, build a module which takes input two documents of the same language at a time and outputs the cosine similarity and Pearson's correlation coefficient for the two. No external libraries should be used.

Deliverables

1. Well commented code. The purpose and intent of each method, class and module should be mentioned appropriately.
2. Design document describing each and every aspect of assignment should be created which should clearly state all the assumptions made for implementing, limitations, algorithms used

etc. Also show the structure of assignment, by explaining the interdependencies between different modules implemented.

3. Readme file having all the steps for compiling your code and running the translator.
4. Results document tabulating the (average) similarity score and Pearson's correlation coefficient obtained for training and validation sets, for all the model(s) trained.

Innovation

Improving performance of the translator model using higher degree IBM models (2, 3 etc.) or any other optimization technique would be rewarded. You can even use the model trained by you for building a simple retrieval system for CLIR.
Explain this part with **all the details** in the design document.

Interface

The translator can have a simple command line interface (GUI doesn't carry any marks). But the driver code, when run, should have the following interface implemented -

- Show the translated document for given test document(s).
- Compute the cosine similarity and Pearson's correlation coefficient between two documents specified by paths (show them for each doc pair)
- Show the average cosine similarity and Pearson's correlation coefficient for all the test cases.

Submission Details

- You are required to work in a group of exactly five members. **Enter group details** in the Sheet shared herewith **on or before 31st October, 2019**.
- Make a zip file of all the above mentioned mandatory deliverables along with any extra documents and upload it on **Nalanda on or before 6th November, 2019 uptil 11:55 PM** with the zip file named as "Group no. XX" where XX should be substituted with the group number.
- You will have to run the code on your own machine, at the time of the demo. Please ensure the code is working as you expect it to, before coming for demo.

Evaluation Scheme

Task	**Marks**
Implementing Statistical Translational Model	6
Performance on Train and Validation data	3
Performance on Test data	4
Documentation	4
Innovation	5
Viva	8
Total Marks	**30**