BITS - Pilani,Hyderabad Campus

CS F469 IR Assignment - 3

Deadline: 18/11/2018

This assignment is aimed at implementing and understanding IBM Alignment Models and phrase-based translation.

The assignment can be done in groups of **at most 4 (Four) members**. All the group members are expected to contribute to all the aspects of the assignment namely, design, implementation, documentation and testing.

Programming Languages:

Please use Python to code this assignment. External libraries such as **nltk** are allowed in specific parts of the assignment.

Task Description:

1) **IBM Model 1 implementation.**

   Implement IBM Model 1 and the EM algorithm upon two given sentence-aligned corpora (refer to CMS). No external libraries may be used for this portion. The code must output each sentence pair and its most likely alignment.

   Students must also create a parallel corpus of their own between English and a language of their choice containing at least 8 sentences. Run your IBM Model 1 implementation on this dataset. The alignments obtained on your own dataset need not perfectly match the true alignment of words, but you must be able to explain the reasons why the results are imperfect and the model's limitations.

2) **IBM Model 1 and 2 Analysis (using nltk).**

   Use the python nltk library IBMModel1 and IBMModel2 implementations to verify results obtained in the previous task on the given corpus and on your own. Please explain any discrepancies in results obtained, if any.

3) **Phrase based extraction and scoring (using nltk)**

   Use the phrased based translation module in nltk to extract phrases from dataset #2. Use the alignments obtained by your IBM Model 1 implementation as inputs. Once the phrases have been

extracted, calculate the phrase scores for each extracted phrase and rank them in order of descending probability. Do the same on your own dataset. Examine and explain the results obtained in both cases.

Additional Resources:

1. Nltk Installation: https://www.nltk.org/install.html
2. IBM Model implementations can be found in the nltk.translate module : https://www.nltk.org/api/nltk.translate.html
3. Phrase based extraction module : https://www.nltk.org/_modules/nltk/translate/phrase_based.html
4. All relevant formulae can be found in the lecture slides. Please refer to CMS.
5. In case of any queries kindly mail to Ms. Shreya Nimma (f20150951@hyderabad.bits-pilani.ac.in

Deliverables:

The final submission must contain the following documents:

1. **Design Document** – This document should contain the description of the application's architecture along with the major data structures used in the project. Running times must be documented. All results must be presented neatly and thoroughly explained. For the corpus created by your team, the true alignments of the sentences (determined by your own knowledge) should be presented along with the alignments obtained by your code.
2. **Code** – The code should be well commented.
3. **Documentation** – All the classes, functions and modules of the code must be documented. Software that automatically generate such documents can be used – pydoc for Python, Eclipse for Java etc.
4. **README** – The README file should describe the procedure to compile and run your code for various datasets.

Submission Guidelines:

All the deliverables must be zipped and submitted to bphc.information.retrieval@gmail.com latest by **deadline.**

You are expected to demo your application and present your results as per the schedule that will be made available.

Evaluation Criteria for Task :

S.No.	Task	Marks
1.	IBM Model -1 and EM algorithm implementation	5
2.	Own data creation and reporting issues	5
3.	Executing the NLTK IBM model and reporting discrepancies	5
4.	Phrase extractions, translation and comparing results with word based models	10
5.	Viva	5
	Total	**30**

It should be noted that all the assignments would be run through a plagiarism detector and based on the results, the marks would be altered. The final decision lies in the hand of the instructor and only one submission per group would be allowed for one assignment.