Natural Language Processing, 2018: Report on Assignment-1

Sqn Ldr Abhimanyu Vatta
M.Tech. First Year, CSA, IISc
SR No: 15251
abhimanyuv@iisc.ac.in, v.manyu@gmail.com

1 Problem Statement - Assignemnt 1

Primary task involves developing a language model using two NLP training corpus namely Brown and Gutenberg Corpus. Assignment further is divided into two major tasks as follows:

1.1 Task-1

In this task, each dataset needs to be divided into training, development and test sets. Training splits' notation is D1-Train/D1-Test (for guttenberg) and D2-Train/D2-Test (for brown). We have to implement and build a Language Model with the following settings:

- S1: Train: D1, Test: D1
- S2: Train: D2, Test: D2
- S3: Train: D1 & D2, Test: D1
- S4: Train: D1 & D2, Test: D2

1.2 Task-2

Using the Language Model developed as Task-1, we need to develop a script with the name $generate\_sentence.sh$ which will generate a sentence of 10 tokens.

2 Development of N-Gram Language Model

As a solution of the Task-1, Trigram Language Model is implemented. To enhance the Development task and tuning of hyper parameters, size of training, development and test set is selected in 80:10:10 ratio for both of the corpus.

References include book and slides of Dan Jurafsky and Christopher Manning (Dan Jurafsky, 2012), along with the work of Favian Contreras (Contreras, 2014) during the development of solutions against Assignment-1.

NLTK library for the Brown and Guttenberg corpus have not ben used, instead pre-processing has been carried out using regular expressions. Pre-processing of the corpus involves following subtasks:

- Reading of files
- Identifying the period punctuations (.?!) as end of sentence
- Addition of start and end symbol accordingly ($\langle s \rangle$, $\langle /s \rangle$)
- Removing Book Titles (identifying text in between []), Chapter numbers and other words which are not part of a sentence.
- Tokenization of the sentences
- Removal of numerics
- Removal of single characters

3 Smoothing Techniques

Three type of smoothing techniques are used in aforesaid Language model as a learning exercise.

3.1 $\langle UNK \rangle$

Occurrence of words for less than or equal to 5 times in the whole corpus was replaced by $\langle UNK \rangle$. Post this the model was trained onto trigram probabilities with $\langle UNK \rangle$ tokens occurring as words.

3.2 K-Smoothing

Originally, Laplase smoothing was tested in the Language Model but to make evaluation more responsive, k-smoothing was used with k set to 0.004 after trying various values like .1, 0.01, 0.05 etc.

4 Observations

Language Model is tested with different combinations of the bi/tri-grams, unknown tokens threshold and K-Smoothing with difffenrent values of K.

Bi-gram vs Tri-gram Probabilities - Tri-gram model predicts the contextual relations better as expected and also brings down the perplexity which is in direct relation to the observation.

Inclusion of the UNK token decreases the overall entropy and thus perplexity as rarely occurring words (less than or equal to 5) in a corpus are all clubbed together for better probability estimate and also assigning better chance of word/set of words(tri-gram) encountered in Test set which does not occurs in Training set vocabulary size. It brought down the vocabulary size down to one-third size in both Guttenberg and Brown Corpus.

Since the vocabulary size was large, Laplace smoothing leads to large changes in frequqency data of n-grams. K-smoothing has shows good improvement due to reduction in vocabulary size by factor of 0.004.

5 Generation of Sentence

Script file $generate\_sentence.sh$ is developed as a wrapper over original python call of the sentence generation module.

6 Result

Following are the results derived by running program on training and testing models as per subtasks defined in Section 1.1 -

- **S1**: Train: D1, Test:D1
  Sentence: the heavens hath left to guilty shame , who was
  Perplexity: 629.427062636204

- **S2**: Train: D2, Test:D2
  Sentence: with progressivism the religion of democracy , into a three
  Perplexity: 1236.83569865428

- **S3**: Train: D1 & D2, Test:D1
  Sentence: am i impressed with its black , i recall all
  Perplexity: 54.81045377172605

- **S4**: Train: D1 & D2, Test:D2
  Sentence: the sandburg goat herd increased , and in his field
  Perplexity: 86.21266986318547

7 Conclusion

It can be observed and concluded that S3 model gives best performance in terms of intrinsic evaluation i.e. Perplexity measure. Models S1 and S2 performs well in sentence generation as both of the model uses Training as well as Testing set from the same corpus. (Punctuation comma (,) is retained as it was observed that it helped retain meaning of a sentence).

8 Usage

This solution contains three python files $LM.py$ along with two corpus guttenberg and brown. The usage of this Language Model interface is:
LM.py [-t] [-t T] [-m] [-m M]

- -t: Set 1 for Language Model Generation & Evaluation.
  Set 2 for generation of 10 words sentence.
  (Default value set to 1).

- -m: Set S1/S2/S3/S4 for respective task.
  (Default value set to S3).

8.1 Sentence Generation with model S1

python.exe LM.py -t 2 -m S1

References

Favian Contreras. 2014. N-grams. https://github.com/BigFav.

Christopher Manning Dan Jurafsky. 2012. Natural language processing, stanford. https://web.stanford.edu/~jurafsky/NLPCourseraSlides.html.