Assignment -01 : Language modelling

1 Problem statement

To experiment and implement the best language model for the datasets Brown Corpus(D1) and Gutenberg Corpus(D2) and evaluate them in following settings: S1: D1 as Train and D1 as Test, S2: D2 as Train and D2 as Test, S3: D1 and D2 as Train and D1 as Test, S4: D1 and D2 as Train and D2 as Test Perplexity is to be used as metric for the comparisons and Using the best model, we have to generate a sentence of 10 tokens.

2 Introduction and Methodology

Language models finds its application in various fundamental tasks such as assigning probability to a sentence and various other advanced tasks such as spelling correction, machine translation etc. In general,bigram based models are syntactic whereas trigram based models are both semantic and syntactic in nature. For our comparisons, we have used Katz backoff method to implement bigram based model and Kneser-Ney smoothing method to implement both bigram based and trigram based models. In both the methods, our language model remains a true probability distribution. Katz backoff involves discounting along with backoff and Kneser-Ney involve discounting along with interpolation.

3 DataSet

After reading brown corpus(D1) and gutenberg corpus(D2) into a list of sentences, a start token is introduced at start and an end token is inserted at end of each sentence. This is helpful because when we create the word sequence using the list of sentences, sequence may cross many sentence boundaries and hence, it can be avoided. Next, we split each of the list of sentences in 80:20 train-test spilt. Now training and test data is being sent to the language model to train and evaluate as per the settings S1, S2, S3, S4 mentioned in problem statement. Further, 20 percent of training data is kept as heldout set to train our hyperparameters while designing our language models.

4 Evaluation metric

Our language models will be evaluated and compared on the basis of perplexity metric. Minimizing perplexity is equivalent to maximizing the test set probability according to the language model. Hence, smaller the perplexity,better is the model.

5 Implementation

Training data being obtained as per the settings and after further splitting into training set and heldout set is converted into word sequences and unigram dictionary is obtained. Also, test data is obtained as per the settings. Now to handle unseen unigrams which may appear in test data, we replace 10 percent of unigrams in training data having count equal to one by ‘UNK’ and modified unigram dictionary is obtained. Also, bigram and trigram dictionaries are obtained for training set. Test data and heldout data are also converted into word sequences and the unigram which are not present in training set ungram dictionary are replaced by ’UNK’.

5.1 Katz backoff based bigram model

In Katz backoff, we rely on discounted probability, if we have seen the bigram before. Otherwise, we recursively backoff to katz probability for unigram. Discount ’d’ is considered as hyperparameter and best posssible value for ‘d’ that gives perplexity measure on heldout data. Here, no context of unigram is captured and probability of unigram is directly obtained from count of unigram. Hence, probability of test corpus is obtained and perplexity is thus obtained to evaluate our language model.

5.2 Kneser-Ney based bigram model

Probability of any bigram is represented as the bigram probabilty with some discount 'd' interpolated with unigram probabilty.Although, Church and Gale(1991) observed by experimenting that best value of 'd' is .75, We have still considered 'd' as hyperparameter and obtained best posssible value for 'd' that gives best perplexity measure on heldout data. Also, probability of unigram is given by the number of bigrams it was seen as novel continuation divided by total number of bigram types. This captures the context of unigram. Hence, probability of test corpus is obtained and perplexity is thus obtained to evaluate our language model.

5.3 Kneser-Ney based trigram model

Just like Kneser-Ney based bigram model, here probability of any trigram is represented as the trigram probability with some discount interpolated with recursive Kneser-ney bigram and unigram probabilities. Here, we have two hyperparameters :- d1, which corresponds to trigram probability discount and d2, which corresponds to bigram probability discount. d2 is kept fixed at the best dicount obtained from Kneser-Ney based bigram model and d1 is tuned to give best perplexity on heldout data. Also, we have used normal count for trigram and continuation count for bigram and unigram. This helps us in capturing context and thus improves the perplexity. While finding probability of test corpus, if for a particular trigram, its bigram of first two second words is not present in training corpus, then we direcly find the recursive Kneser-ney bigram probability. This does not affect probability distribution, because that bigram would not be present in any of the training set trigram. Hence, probability of test corpus is obtained and perplexity is thus obtained to evaluate our language model.

Probabilty distribution of all these three models were verified to ensure correctness of implementation.

5.4 Random sentence generation

From the Kneser-Ney language model, 5 random sentences of length 10 were generated. We take a word at random from list of words ['The','He','This','His','Her','Our','An'] to intialize the sentence. Further associated trigram is picked at random and its probability is compared with the random number between 0 and 1. If it is greater than it, append the first two words of this trigram to a sentence and process is repeated with the last word.

6 Discount tuning and Results

Discount tuning curves implemented on held out corpus for different models have been shown in Figure 1,2,and 3. From the curves,corresponding discounts which gives the minimum perplexity have been selected and shown in Table 1.1. With these discounts, we evaluated our models on test corpus as per the settings and perplexity obtained have been shown in Table 2.

Tuned discounts for our language models			
Training corpus vs methods	Knesar-Ney (bi-gram)	Katz backoff (bi-gram)	Knesar-Ney (tri-gram)
Brown	.8	.7	.95
Gutenberg	.75	.7	.9
Brown and Gutenberg	.8	.65	.9

Table 1: Tuned discounts for the different methods corresponding to different training corpus

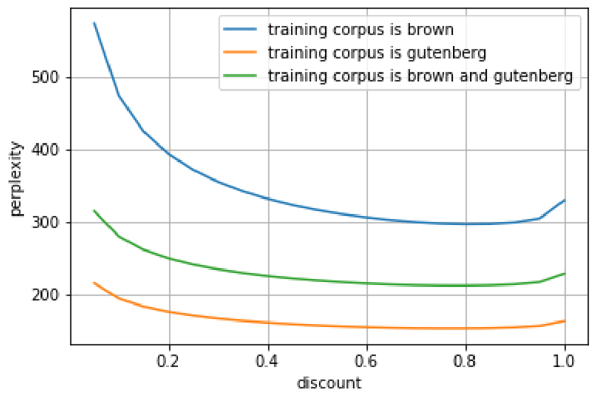

Figure 1: Perlexity vs discount for heldout corpus in Kneser-Ney based bigram model.

Following random sentences with brown corpus of length 10 were generated:

- Our companion was barren of histochemical techniques and fermentation extracts

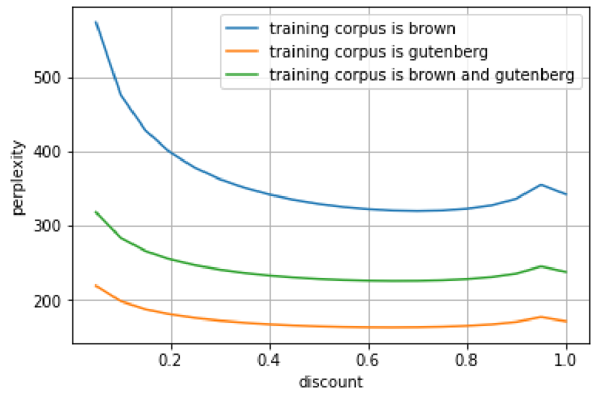

Figure 2: Perlexity vs discount for heldout corpus in Katz backoff based trigram model.

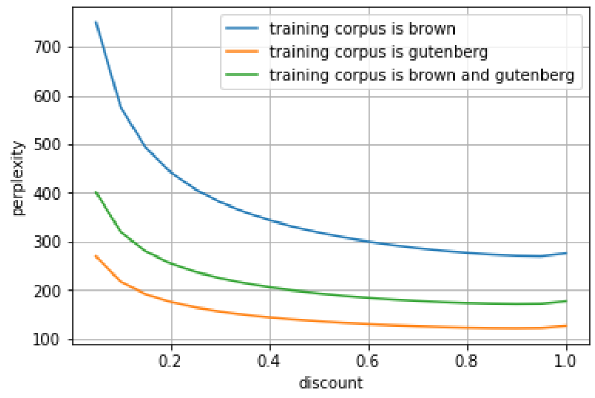

Figure 3: Perlexity vs discount for heldout corpus in Kneser-Ney based trigram model.

Perplexity comparisons			
Perplexity under different settings vs language models	Katz backoff (bi-gram)	Knesar-Ney (bi-gram)	Knesar-Ney (tri-gram)
S1	320	297	270
S2	166	157	120
S3	417	379	338
S4	175	167	129

Table 2: Perplexity comparisons among different language models under give settings

- He used the fatal historical elegance . Absolution for corny
- Our American Cousin Emma is agreed that Regular Army
- An interested sitter attach to brush his wits . Smarter

7 Observations and Summary

Following are the observations made while implementing and after impementation.

- In both Kneser-Ney and Katz backoff, tuned discount was found to be very close to 0.75. This was expected as suggested by Church and Gale(1991) by teir experimentation.
- In Katz backoff based bigram model, perplexity was found to be higher because when a bigram is not found, it backoff to lower order unigram and and probability of unigram is directly calculated from the count. Hence, it does not capture the type of context in which it has occured. Hence, it may assign more probabilities to unlikely bigrams. For eg, ‘reading kong’ could be assigned more probability than ‘reading glasses’ if both the bigrams are not present in training corpus.
- To handle this issue and to get better estimates of probabilities, we tried Kneser-ney based bigram model which captures the conext of unigram as well as mix the probabilities of bigram and unigram. Hence, we obtained better perplexity forn all the settings.
- To further improve our perplexity, we implemented trigram based Kneser-ney model as trigram model captures semantic relations in a better way. Perplexity was found to be improved by margin of 25 percent in settings S2 and S4 and by margin of 10 percent in settings S1 and S3. Hence, we propose ‘Kneser-ney based trigram model’ as bst model among those implemented.
- Also, it is observed that perplexity is very high when test corpus is from brown as compared to when test corpus is from gutenberg. This may be happening because brown coupus may have more long distant relationship and more syntactic variations than gutenberg corpus. Hence, our bigram,trigram models are not good enough to predict the probability of sentence when it is from brown corpus.

8 Github Link

Github Link : https://github.com/shubham-IISc/Language-model-Assignment-1.git