COLUMBIA UNIVERSITY – CUNY

Homework 2 - NLP/ML/Web

Jessica Ouyang and Joe Ellis

October 21, 2013

1 TASK – LANGUAGE MODELING

In this problem we are tasked with attempting to modeling a language example of at least 100,000 words and another language model that is created using a seperate corpus of at least 50,000 words. For this experiment we chose to use two famous novels as our language examples, one being "The Scarlet Letter" by Nathaniel Hawthorne and the other being "A Tale of Two Cities" written by Charles Dickens. Both were modeled using different n-gram language models. We also applied Laplacian smoothing and back-off to our models to create more accuracte language models. Finally, we created a combined model trained off of both language examples.

2 EXPERIMENTS AND RESULTS

2.1 TASK 1 – DOMAIN A

1A – DOWNLOAD LANGUAGE TEXT For our first domain we chose to download and use the famous novel "A Tale of Two Cities" by Charles Dickens. The novel had much more than 100,000 words. For our train and test split we decided that we would use 60% of the available data for test, 20% for development, and 20% for evalauation.

1B - FIND N FOR LANGUAGE MODEL After this we had to find the N for the N gram model that would be utilized for this text. The best result in our examples for the proper N-gram model was the tri-gram model. This first example we did not do any smoothing, therefore we evaluated only on the seen grams from the development set, and chose N that provided the lowest perplexity. Average Perplexity per gram was calculated using the formula,

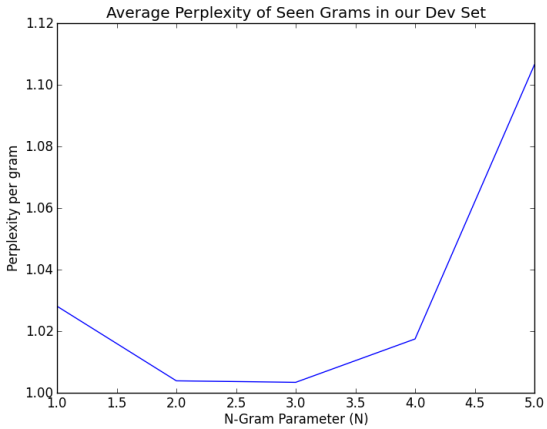

Figure 1: N-Gram Performance for Domain A

$$Perplexity_{ave}(text) = 2^{-\frac{1}{N}\sum_1^N p(w_i)log(p(w_i))} \tag{1}$$

We chose to use this formula, so that different models with differing data sizes can be compared x. We chose tri-gram as the best model, and the graph of the results can be seen in Figure reffig:figure1. The results on the test set for seen grams is $Perplexity(test) = 15.426$

1C – LAPLACE SMOOTHING We also utilized Laplace smoothing to solve for the best value of N with Laplace Smoothing utilized. When we use Laplace smoothing instead of using simply $p(w) = \frac{N(w)}{N}$, we include and extra value of V into the equation where V is the number of distinct N-grams present. Therefore, the new $p(w) = \frac{N(w)+1}{N+V}$. This allows us to classify all of the values seen, but gives small probablities to unseen before grams. The results of this modeling scheme can be seen below, and for this model tri-gram also wins out. However, since many of the n-grams are unseen for $n = 4$ or $n = 5$, the penalty is not as high as before. The results are shown in Figure 2. The result on the test set for all grams with smoothing is, $Perplexity(test) = 2^{-31.25}$, and the total entropy is -31.250.

1D – BACK-OFF Finally, we addressed the issue of unseen grams during training using the Back-Off parameters. For, Back-Off we use $p(w)$ as a linear interpolation of the values of each condition probablity that makes up $p(w)$. Therefore, the equation for back-off is as follows,

$$p(w) = \lambda_0 p(w_0) + \lambda_1 p(w_1|w_0) + \lambda_2 p(w_2|w_1, w_0). \tag{2}$$

We would have used 4-gram for back-off training, however once I got to 4-gram language representation 98% of my grams were unseen. So most of the values would be unseen for this case, and therefore we chose to only perform back-off from 3. We attempted to use a

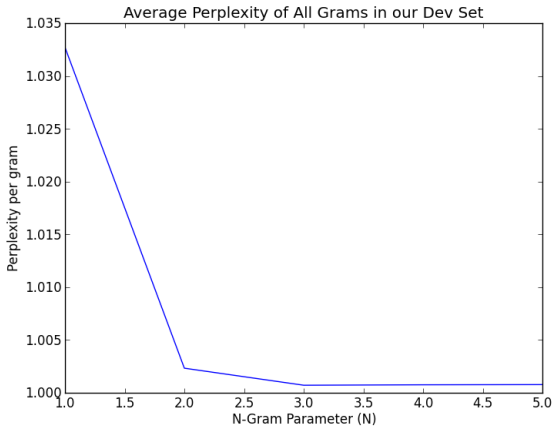

Figure 2: N-Gram Performance for Domain A

Table 1: lambda values and perplexity scores

λ_0	λ_1	λ_2	$Perplexity_{ave}(dev)$
0	0.5	0.5	1.0075
0.33	0.33	0.33	1.0181
0.2	0.4	0.4	1.0145
0.5	0.25	0.25	1.0223
0.25	0.5	0.25	1.0171
0.25	0.25	0.5	1.0148
0.1	0.2	0.7	1.0089
0.0	0.3	0.7	1.0055

variety of different lambdas for the language model and the results can be seen below in Table reftab:table1. We have chosen our λ wieghting scheme as $\lambda = (0.0, 0.3, 0.7)$, and this is because we wanted all λs to be non-zero, but the unigram model was not useful for this problem. The perplexity score on our test dataset using these parameters was $Perplexity(test) = 2^{241.31}$.

2.2 Task 2

2a - Download another Domain For our second domain, we utilized the famous novel "The Scarlet Letter" by Nathaniel Hawthorne. This book contains over 50,000 words and the train and test split of the data was the same as was used above.

2b - Find Optimal Language Model The same tests were used as those described above to find the optimal language model for this text, and we received similar results. Once again the highest performing model was the tri-gram model with smoothing. This model appears to work well on many datasets, however it would be useful to have more training data as many

Table 2: lambda values and perplexity scores for linearly combined model

λ_0	λ_1	$Perplexity(dev)$
0.5	0.5	$2^{88.62}$
0.2	0.8	$2^{84.42}$
0.3	0.7	$2^{85.88}$

of the grams remain unseen even in this model. The perplexity of this model on the test-set on the seen grams is $Perplexity(test) = 2^{0.2627} = 1.2033$. When we utilized the $\lambda = (0.0, 0.6, 0.4)$ parameters and performed back-off ad smoothing on the set we received a perplexity score on our dataset of $Perplexity(test) = 2^{113.5347}$

2C-CROSS DOMAIN ANALYSIS We also attempted to see how our first domain model, the model that was trained on Dickens performed on The Scarlet Letter. The performance was similar but slightly degraded, and the resulting average entropy per n-gram is -0.0120, which over the 18,652 trigrams testsed gives us a perplexity of, $Perplexity(crossdomain) = 2^{108.0305}$.

2D- INTERPOLATED MODEL Finally, we also tested on the The Scarlet Letter an interpolated model, that allowed for us to utilized both trained models for modeling. For this model we linearly combined the probablilty received from both models using λ parameters and the results on the development set can be seen below for multiple λ. We have λ_0 corresponding to "A Tale of Two Cities" trained model and λ_1 corresponds to "The Scarlet Letter" trained model. These values outperform just the Scarlet Letter Model by itself on the test data, greatly.

2E - INTERPOLATED MODEL PERFORMANCE The testing was performed on the "The Scarlet Letter" test set, and the resulting perplexity using interpolation parameters ($\lambda_{int} = (0.2, 0.7)$ is $Perplexity(test) = 2^{77.39}$. This shows just how useful it is to have multiple models and broaden our training data. The interpolated model outperforms both singluar models by itself.

3 LINGUISTIC ORIENTED SECTION