Hw2: Language Modeling across Domains with Various Smoothing Methods

Sam Showalter
University of California, Irvine (showalte)
showalte@uci.edu

Abstract

Language modeling has been prominent in natural language processing (NLP) for decades and takes its roots from probabilistic modeling. However, learning long-range dependencies with n-gram models explicitly is intractable since larger contexts become less and less likely to be observed in data. In turn, n-grams rely on Markovian assumptions; conditioning on the previous n-words in a context is considered a sufficient statistic of future likelihood. This homework explores language modeling across three corpuses from different timeframes and domains with n-gram systems. Several methods of probability smoothing are applied to these models to improve generalization, including Laplacian, add-k , and stupid backoff (Brants et al., 2007) smoothing.

Our findings indicate that across all corpuses, stupid backoff smoothing was most effective at ensuring generalization for n-grams. Moreover, bigram models were best suited for generalizing across all corpuses, though for more homogeneous corpuses (`reuters`, `gutenberg`) benefited from additional context via higher order n-grams (namely, trigrams).

We also applied our tuned n-gram models to each corpus across domains. Using the same set of basic prefixes, we compared generated sentences between these models as well as perplexity scoring. Our high level observations include that the `brown` and `gutenberg` corpuses were most similar lexically, the `brown` corpus appeared to be most diverse, and `reuters` the least. In general, natural sounding sentences scored lower perplexities on n-gram models with $n > 1$, but with poetic or syntactically jumbled sentences, unigram models scored lower values. Due to the length of some generated sentences, extra information is stored in the Appendix 5.

1 Related Work: Language Model Implementations

Language modeling addresses the challenge of generating predictions from sparse data. Language in any dialect is diverse, ambiguous, and generally full of syntax flexibility. As a result, effective methods often make use of intelligent forms of probability smoothing (Chen and Goodman, 1999). Without probabilistic smoothing, n-gram models see some vocabulary combinations as impossible (Jelinek, 1980) and are infinitely surprised when a combination of tokens not witnessed during training is observed at test time.

In addition, a language models capacity is heavily influenced by its n-gram cardinality. Trigram models tend to incorporate higher capacity than bigram models, and so on. Our hyperparameter tuning also focuses on identifying the optimal n-gram cardinality for generalization as well as the correct regularization to assist in the prevention of overfitting.

2 Experimental Setup

Our language model was implemented from scratch and is defined by the following characteristics.

1. Our package can accommodate an arbitrary context size (n), dynamically producing necessary padding and start-of-sentence tokens.

2. Our package can implement add-k smoothing (for an arbitrary k) as well as "stupid" (Brants et al., 2007) backoff smoothing.

3. No centralization of tokens with low counts into `Unk` tokens is undertaken. We made this design choice in an effort to avoid losing meaningful words and speculating wrongly on correct `Unk` thresholds.

With this language model package, we implemented three rounds of hyperparameter tuning. First, we explore the performance of different add-k smoothing methods on the bigram model. Using the results of this ablation, we examine the impact of higher-order n-grams on our performance.

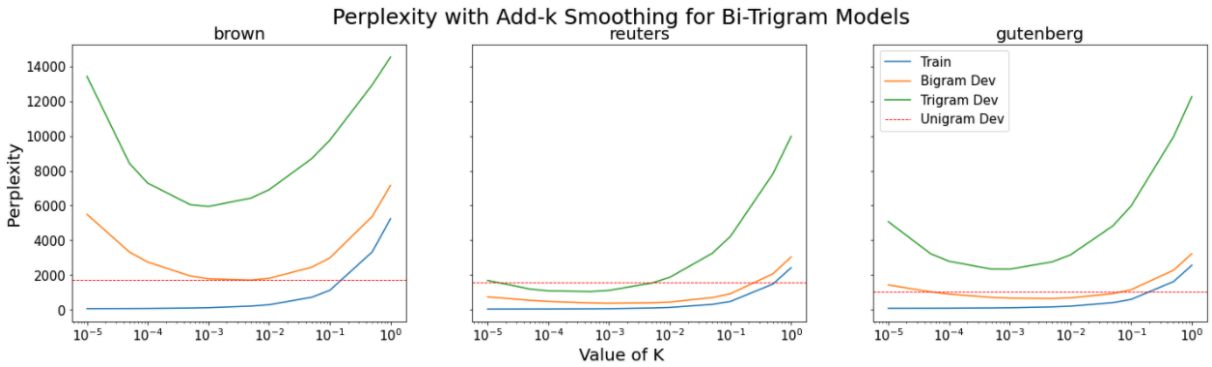

Figure 1: Train and dev-set perplexity for bi- and tri-gram language models with vary k in add-k smoothing. A k=1 represents Laplacian smoothing while dropping k represents weakening the Dirichlet prior based on the data. Referenced is dev-set perplexity for unigram models.

Thirdly, we restarted this two-step process with stupid backoff smoothing, tuning the n-gram cardinality as well as the backoff parameter λ. More sophisticated methods of smoothing like Katz (Katz, 1987) and Kneser-Ney (Kneser and Ney, 1995) were not attempted due to time constraints.

3 Hyperparameter Tuning

Perhaps the simplest smoothing mechanism for n-gram language models is Laplacian smoothing (MacKay and Peto, 1995). In general, conditional token probabilities given context are derived from frequentist estimates (counts) of instances seen during training. If a token is not seen with a given context at this time, its probability of occurring at test time is set to zero. This is an unreasonable assumption solved by smoothing in general. Laplacian smoothing addresses this by adjusting all count-derived probabilities to the following:

$$p_{laplace}(w_i) = \frac{C(w_i, w_{i-1,\cdots i-n+1}) + k}{C(w_{i-1,\cdots i-n+1}) + kV} \quad (1)$$

where k for Laplacian smoothing is equal to 1 and V refers to the size of the vocabulary. Another way of thinking of Laplacian smoothing is as a conversion from what was a a maximum likelihood estimate into a maximum a-posteriori by adding a Dirichlet prior to the data distribution, defined in this project as a categorical distribution. In this context, k represents the strength of the prior. This adjustment ensures all tokens seen at test time have positive probability density.

Initial experiments with a Laplacian prior yielded poor results, defined in terms of development (dev) set perplexity. We determined this issue to be the result of a prior that was too strong and overpowered the witnessed data. Figure 1 explores the impact of scaling down the strength of our prior, determining an optimal $k = 0.01$ for all datasets. For all corpuses, one can see the transition from underfitting to overfitting.

In addition, Figure 1 includes the same ablation conducted on a trigram model. It appears that the additional capacity of the trigram led it to overfit more easily, resulting in poor generalization relative to the bigram model. Interestingly, the optimal add-k smoothing for bigrams only improved the `brown` model slightly from its unigram baseline, while the other corpuses improve substantially. We believe this is due to the diversity of content in the corpus (poetry, sports, plays, etc.).

Attempting to improve our n-gram generalization performance, we also implemented a more sophisticated smoothing protocol called stupid backoff. Stupid (naive) backoff (Brants et al., 2007) operates by querying an n-gram corpus for a given context-word pair. If no occurrence exist, then the algorithm conducts the same query, but this time on a lower-order n-gram corpus. This iteration continues until a match for the word is found (even if it is a unigram). A heuristic penalty of $\lambda = 0.4$ is recursively applied after each backoff iteration to improve generalization and maintain valid probabilities. If a word is never found, a very low default probability is returned.

3.1 In-Domain Text Analysis: Empirical

Empirically, we found stupid backoff to work well relative to add-k performance. For every corpus, dev set perplexity dropped substantially. However, the `brown` corpus again was quick to overfit when higher-order n-gram models were applied. Trigram models boosted performance modestly for `reuters` and `gutenberg`, but the brown corpus saw a substantial increase in perplexity.

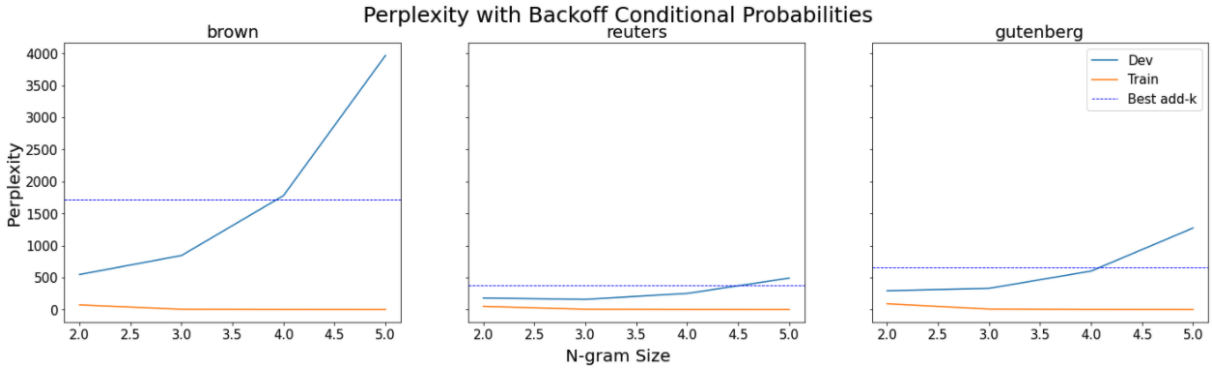

Figure 2: Dev-set perplexity across corpuses with stupid backoff smoothing implemented with $\lambda = 0.4$ and varing number of context sizes (n-gram length). Referenced is current best performance from add-k smoothing (dotted).

Because the performance jump from bigram to trigram models in the other two corpuses was modest, we chose a bigram model implemented with stupid backoff as our tuned model. The baseline unigram performance, along with that of our tuned add-k and backoff models, is shown in Table 1.

Though not shown, varying numbers of λ were tested. We found that $\lambda = 0.4$ was optimal across all corpuses for balancing underfitting (low λ) and overfitting (high λ). With our hyperparameter search complete, we then explored the behavior of our trained bigram models on out of domain data.

3.2 Out-of-Domain Text Analysis: Empirical

An insightful way to explore differences between data corpuses is to train a language model on one and have it run inference on another. We have conducted this exact experiment below in Table 2 by scoring the test set perplexity on each language model - corpus pair. As expected, each model determined its own corpus to be the most natural sounding. However, we also saw noticeable similarity between the `brown` and `gutenberg` corpuses. The test-set perplexity of `gutenberg` when evaluated under the `brown` language model was 841, while `reuters` scored over 2100 in the same setting. Furthermore, When these were evaluated under the `gutenberg` model, the `reuters` corpus was four times more perplexing than `brown`. This led us to believe the lexical characteristics of the `brown` and `gutenberg` corpuses were most aligned.

	brown	reuters	gutenberg
uni.	1514 / 1758	1467 / 1577	981 / 1036
add-k	128 / 1822	72 / 378	124 / 650
b-off	75 / **559**	51 / **180**	92 / **285**

Table 1: Hyperparameter tuning results. Train and dev-set perplexity across all three corpuses with different smoothing. add-k and backoff (b-off) running optimal k=0.01 and n=2 for ngram models

	brown	reuters	gutenberg
brown	**558.59**	2118.33	840.811
reuters	1500.02	**180.312**	2159.68
gutenberg	1098.51	4240.6	**285.342**

Table 2: Test set perpexlity for each language model with stupid backoff smoothing applied. Each model is applied to every corpus.

The `reuters` language model was also unique for how well it generalized to in-domain test data. With a perplexity of 180, it was by far the most successful in-domain bigram model. We feel this is due to the `reuters` corpus being domain specific (finance, business) and therefore more homogeneous than the other corpuses. By contrast, the `brown` corpus was by far the most lexically diverse; its test-set perplexity was over 3 times higher than the other corpuses and nearly 8 times higher than its train-set perplexity. The difficulty of modeling diverse language for downstream tasks is a particularly well explored phenomenon (Ponte and Croft, 1998). To take a closer look at the differences in model and corpus diversity, we fed models human and machine generated sentences and observed perplexity scoring more minutely.

3.3 Qualitative Analysis: Language Scoring

As displayed in Table 3, we have created several sentences (indexed to appendix A) to score under our three language models. Sentences 0, 1, and 2 were created by the authors to appear most natural under the `reuters`, `gutenberg`, and `brown` corpuses, respectively. As expected, the perplexity of our models reflected this intention.

sen.	brown	reuters	gutenberg
0	6836 / 5268	**490** / 1571	4012 / 6455
1	1901 / 1997	1765 / 2670	**519** / 1325
2	**621** / 1081	5685 / 2628	2094 / 1572
3	1011 / **687**	1353 / **1334**	1884 / **1473**
4^g	1270 / 1663	2296 / 2656	**124** / 1271
5^r	**110** / 335	245 / 882	**111** / 361
6^r	1498 / 2288	**846** / 1430	3863 / 4466

Table 3: Perplexity scoring of sentences (bigram / unigram) (indexed in appendix A) by backoff language models. Columns represent the domain of the model used for scoring. Superscripts denote the domain of the model that generated the sentence, if not human.

Sentence 0, a comment on the financial forecast of Dow Chemical, was given a perplexity score of 490 by the `reuters` model, 8x and 10x smaller than the scores given by the `gutenberg` and `brown` datasets, respectively.

By contrast, sentence 1 - a Shakespearean question of temptation - appears most natural under the `gutenberg` language model, though the contrast between other models is not as stark as sentence 0. Sentence 2, an informal football discussion and seen as most natural by the `brown` language model, was scored 10x more perplexing by `reuters` but only 3x more perplexing by `gutenberg`, providing further evidence that the `brown` and `gutenberg` corpuses are most similar. Lastly, sentence 3 is most unique because the unigram model for all corpuses scored a lower perplexity.

A short but modern poetic phrase, sentence 3 seemed most natural under the unigram model because unigrams do not model context. Since all of the words in the phrase are relatively common independently (but not in their present order), n-gram models from all domains saw this phrase as unnatural. This demonstrates one of the fundamental challenges of language modeling across domains. The definition of what is a "natural" language phrase is difficult to model holistically.

Aside from human-generated sentences, we also included three machine generated sentences to compare the `reuters` and `gutenberg` corpuses. All sentences were seeded with the prefix `It was like`. Sentence 4^g was generated by the `gutenberg` corpus and also categorized as most natural under the same. Sentence 6^r depicts the same trend for the `reuters` corpus. Alternatively, 5^r was generated by the `reuters` corpus but scored as more natural under the `brown` and `gutenberg` corpuses. This is likely due to how short the sentence is; with only a few tokens, the sentence also contains few exclusively economic terms. These two features together made 5^r appear unnatural under the domain-specific `reuters` corpus, which is sensitive to business terms.

Prefix: **`The forecast`**

Brown: **`The forecast`** `by an age could flash gangling man found to suppose we have established rapport and payments shall be adjusted Richards of Russia does the at Portsmouth Haverhill and wild`

Reuters: **`The forecast`** `on stocks and preferred their trade has led some 11 Pechiney this CBT floor price index has pared severely impair efficiency to elaborate`

Gutenberg: **`The forecast`** `of Israel 21 say is falleth virtue summon all our servants and his own hands moment they thank them Why did you`

Prefix: **`Who hath`**

Brown: **`Who hath`** `according to whatever substantial ups unconsciously keeping the could pilots took pride or disquietude he told between eleven at the synergism between 1930`

Reuters: **`Who hath`** `QTR NET Shr 28 producing area`

Gutenberg: **`Who hath`** `failed of up into the man shall hiss What chinn value not your game with thine By at once more in them will Jerusalem saying The mound torrid suns`

3.4 Qualitative Analysis: Text Generation

In our final set of experiments, we qualitatively assess the performance of our language models by seeding them with several prefixes and observing the output. Shown above in examples 3.3 and 3.3, we seeded sentences with `The forecast`

and `Who hath`, respectively. Other experiments like this were completed and can be found in our supporting materials. We highlight these two prefixes to demonstrate the domain adaptation of our models even when given an out-of-domain prefix. For example, the prefix `Who hath` originates from old-English often found in the `gutenberg` corpus, forcing the other models to adapt. The `reuters` model converts this prefix into a question about stocks and other assets. Additionally, the sentence also appears most unnatural to a human reader. The `brown` corpus does a much better job of generating a natural sentence, but still lacks compared to `gutenberg` which produced a near-biblical phrase.

As a comparison, we also fed each model a more general seed, `The forecast`, one that could yield multiple interpretations. The language models took advantage of this ambiguity to resolve the prefix into its domain. The `reuters` model discussed the forecast of basic and preferred stocks, `brown` quoted something that vaguely resembles historical prose, and again `gutenberg` interprets the forecast as "Israel 21," another pseudo-biblical passage. Though qualitative, the extent to which each generated sentence appears domain specific seems indicative of the lexical homogeneity of the training corpus. From this lens, the `brown` corpus appears the most linguistically diverse, and `reuters` the least, mirroring quantitative findings.

Included in appendix B, generated sentences from unigram models loosely replicate the findings of our tuned n-gram models, with a few notable exceptions. First, unigram models are poor at conjugating phrases and developing coherent thoughts due to their lack of conditioning on history and directly sampling from the marginal distributions of words. N-gram models prove far more coherent even with a single word context, though long phrases tend to progressively lose meaning.

4 Discussion and Conclusion

Considered together, our quantitative and qualitative findings yield several insights. Regarding the `brown` corpus, we found it surprising that our add-k smoothing struggled to even slightly outperform the unigram baseline compared to the other corpuses. After further examination, we feel this is because this corpus is far more linguistically diverse than its counterparts. This is confirmed by the tendency for language models to overfit to the `brown` corpus and the discovery that this corpus includes many different forms of language, from sports to poetry. Even with a more sophisticated smoothing method, stupid backoff, n-gram models struggled to fit well to the data and not 2.

Most likely, this linguistic diversity was part of the reason why the `brown` corpus generalized better to the `gutenberg` texts than `reuters`. Full of domain-specific terminology, the `reuters` corpus was considered an almost completely foreign dialect when scored by other models. This observation is partially true, with many terms in the `reuters` unambiguously financial and business related. Though each corpus observes some domain-specificity (e.g. biblical references and old-English from `gutenberg`), the concentration of domain-specific terms is most pronounced in `reuters`. Evidence of this homogeneity can be seen with the incredibly low perplexity values `reuters` had on its testing corpus relative to other models. However, this excellent in-domain generalization came at the cost of poor out-of-domain adaptation.

In a sense, the `reuters` model intended to capture business-centric phrases more so than English. In `reuters`-generated phrases that included little business jargon, the `brown` and `gutenberg` corpuses scored the statement as more natural. At the same time, it proved nearly impossible to generate a sentence using the `brown` and `gutenberg` models that scored a lower perplexity with `reuters`. Generally, this lends the insight that language models are particularly useful when applied to domain-specific language, but may subsequently struggle to generalize in out-of-domain tasks. In several cases, the in-domain benefit may justify this trade-off.

5 Statement of Collaboration

I solicited help from Anthony Chen about issues I was having with my smoothing techniques and the assignment in general. Beyond this discussion and perusing Campus Wire, I completed this assignment independently.

References

Thorsten Brants, Ashok C Popat, Peng Xu, Franz J Och, and Jeffrey Dean. 2007. Large language models in machine translation. *ArXiv*.

Stanley F Chen and Joshua Goodman. 1999. An empirical study of smoothing techniques for lan-

guage modeling. *Computer Speech & Language*, 13(4):359–394.

Frederick Jelinek. 1980. Interpolated estimation of markov source parameters from sparse data. In *Proc. Workshop on Pattern Recognition in Practice, 1980*.

Slava Katz. 1987. Estimation of probabilities from sparse data for the language model component of a speech recognizer. *IEEE transactions on acoustics, speech, and signal processing*, 35(3):400–401.

Reinhard Kneser and Hermann Ney. 1995. Improved backing-off for m-gram language modeling. In *1995 international conference on acoustics, speech, and signal processing*, volume 1, pages 181–184. IEEE.

David JC MacKay and Linda C Bauman Peto. 1995. A hierarchical dirichlet language model. *Nat. Lang. Eng.*, 1(3):289–308.

Jay M Ponte and W Bruce Croft. 1998. A language modeling approach to information retrieval. In *Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval*, pages 275–281.

A Sentence Reference

Below is a reference for table with generated sentences. As noted above, the sentences indexed in 3 with an additional superscript were generated by the `reuters` model while g references `gutenberg`.

index	sentence
0	This third quarter fiscal forecast for DOW Chemical is bearish according to financial analysts.
1	Who hath such scuples as to remain untrodden by the perils of temptation?
2	Hey did hear about that fight the football team got in to after practice?
3	Walked alone did she, on to tomorrow.
4^g	It was like fire sent them and new psychological influences set in stone brought the Holy Ghost
5^r	It was like the workforce body
6^r	It was like a loss for Bahrain Oman and current residents are packing said company spokesman

B Unigram Generated Sentences

Prefix: **`The forecast`**

Brown: **`The forecast`** `blanket royalty was least is limited story cases heading present be payments movement its two possible the unmistakably`

Reuters: **`The forecast`** `sales late and practices`

Gutenberg: **`The forecast`** `point blossoms blossoms out won mark is natured incommodiously`

Prefix: **`Who hath`**

Brown: **`Who hath`** `the automatically the efficiency`

Reuters: **`Who hath`** `to Ray agency was`

Gutenberg: **`Who hath`** `said she Till the the at there and the news you and hands mouse shall 28 strength entrance assistants my from`