Generalized Interpolation in Decision Tree LM

Denis Filimonov†‡
‡Human Language Technology
Center of Excellence
Johns Hopkins University
den@cs.umd.edu

Mary Harper†
†Department of Computer Science
University of Maryland, College Park
mharper@umd.edu

Abstract

In the face of sparsity, statistical models are often interpolated with lower order (backoff) models, particularly in Language Modeling. In this paper, we argue that there is a relation between the higher order and the backoff model that must be satisfied in order for the interpolation to be effective. We show that in n-gram models, the relation is trivially held, but in models that allow arbitrary clustering of context (such as decision tree models), this relation is generally not satisfied. Based on this insight, we also propose a generalization of linear interpolation which significantly improves the performance of a decision tree language model.

1 Introduction

A prominent use case for Language Models (LMs) in NLP applications such as Automatic Speech Recognition (ASR) and Machine Translation (MT) is selection of the most fluent word sequence among multiple hypotheses. Statistical LMs formulate the problem as the computation of the model's probability to generate the word sequence $w_1 w_2 \dots w_m \equiv w_1^m$, assuming that higher probability corresponds to more fluent hypotheses. LMs are often represented in the following generative form:

$$p(w_1^m) = \prod_{i=1}^{m} p(w_i|w_1^{i-1})$$

In the following discussion, we will refer to the function $p(w_i|w_1^{i-1})$ as a language model.

Note the context space for this function, w_1^{i-1} is arbitrarily long, necessitating some independence assumption, which usually consists of reducing the relevant context to $n-1$ immediately preceding tokens:

$$p(w_i|w_1^{i-1}) \approx p(w_i|w_{i-n+1}^{i-1})$$

These distributions are typically estimated from observed counts of n-grams w_{i-n+1}^{i} in the training data. The context space is still far too large; therefore, the models are recursively smoothed using lower order distributions. For instance, in a widely used n-gram LM, the probabilities are estimated as follows:

$$\begin{aligned}\tilde{p}(w_i|w_{i-n+1}^{i-1}) &= \rho(w_i|w_{i-n+1}^{i-1}) + \\ &\quad \gamma(w_{i-n+1}^{i-1}) \cdot \tilde{p}(w_i|w_{i-n+2}^{i-1})\end{aligned} \quad (1)$$

where ρ is a *discounted* probability[1].

In addition to n-gram models, there are many other ways to estimate probability distributions $p(w_i|w_{i-n+1}^{i-1})$; in this work, we are particularly interested in models involving decision trees (DTs). As in n-gram models, DT models also often utilize interpolation with lower order models; however, there are issues concerning the interpolation which arise from the fact that decision trees permit *arbitrary* clustering of context, and these issues are the main subject of this paper.

[1]We refer the reader to (Chen and Goodman, 1999) for a survey of the discounting methods for n-gram models.

Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics:shortpapers, pages 620–624,
Portland, Oregon, June 19-24, 2011.

2 Decision Trees

The vast context space in a language model mandates the use of context clustering in some form. In n-gram models, the clustering can be represented as a k-ary decision tree of depth $n-1$, where k is the size of the vocabulary. Note that this is a very constrained form of a decision tree, and is probably suboptimal. Indeed, it is likely that some of the clusters predict very similar distributions of words, and the model would benefit from merging them. Therefore, it is reasonable to believe that *arbitrary* (i.e., unconstrained) context clustering such as a decision tree should be able to outperform the n-gram model.

A decision tree provides us with a clustering function $\Phi(w_{i-n+1}^{i-1}) \rightarrow \{\Phi^1, \ldots, \Phi^N\}$, where N is the number of clusters (leaves in the DT), and clusters Φ^k are disjoint subsets of the context space; the probability estimation is approximated as follows:

$$p(w_i|w_{i-n+1}^{i-1}) \approx p(w_i|\Phi(w_{i-n+1}^{i-1})) \qquad (2)$$

Methods of DT construction and probability estimation used in this work are based on (Filimonov and Harper, 2009); therefore, we refer the reader to that paper for details.

Another advantage of using decision trees is the ease of adding parameters such as syntactic tags:

$$\begin{aligned} p(w_1^m) &= \sum_{t_1 \ldots t_m} p(w_1^m t_1^m) = \sum_{t_1 \ldots t_m} \prod_{i=1}^{m} p(w_i t_i | w_1^{i-1} t_1^{i-1}) \\ &\approx \sum_{t_1 \ldots t_m} \prod_{i=1}^{m} p(w_i t_i | \Phi(w_{i-n+1}^{i-1} t_{i-n+1}^{i-1})) \qquad (3) \end{aligned}$$

In this case, the decision tree would cluster the context space $w_{i-n+1}^{i-1} t_{i-n+1}^{i-1}$ based on information theoretic metrics, without utilizing heuristics for which order the context attributes are to be backed off (cf. Eq. 1). In subsequent discussion, we will write equations for word models (Eq. 2), but they are equally applicable to joint models (Eq. 3) with trivial transformations.

3 Backoff Property

Let us rewrite the interpolation Eq. 1 in a more generic way:

$$\begin{aligned} \tilde{p}(w_i|w_1^{i-1}) &= \rho_n(w_i|\Phi_n(w_1^{i-1})) + \qquad (4) \\ &\quad \gamma(\Phi_n(w_1^{i-1})) \cdot \tilde{p}(w_i|BO_{n-1}(w_1^{i-1})) \end{aligned}$$

where, ρ_n is a *discounted* distribution, Φ_n is a clustering function of order n, and $\gamma(\Phi_n(w_1^{i-1}))$ is the backoff weight chosen to normalize the distribution. BO_{n-1} is the *backoff* clustering function of order $n-1$, representing a reduction of context size. In the case of an n-gram model, $\Phi_n(w_1^{i-1})$ is the set of word sequences where the last $n-1$ words are w_{i-n+1}^{i-1}, similarly, $BO_{n-1}(w_1^{i-1})$ is the set of sequences ending with w_{i-n+2}^{i-1}. In the case of a decision tree model, the same backoff function is typically used, but the clustering function can be arbitrary.

The intuition behind Eq. 4 is that the backoff context $BO_{n-1}(w_1^{i-1})$ allows for more robust (but less informed) probability estimation than the context cluster $\Phi_n(w_1^{i-1})$. More precisely:

$$\forall_{w_1^{i-1}, W} \; : \; W \in \Phi_n(w_1^{i-1}) \Rightarrow W \in BO_{n-1}(w_1^{i-1}) \qquad (5)$$

that is, every word sequence W that belongs to a context cluster $\Phi_n(w_1^{i-1})$, belongs to the same backoff cluster $BO_{n-1}(w_1^{i-1})$ (hence has the same backoff distribution). For n-gram models, Property 5 trivially holds since $BO_{n-1}(w_1^{i-1})$ and $\Phi_n(w_1^{i-1})$ are defined as sets of sequences ending with w_{i-n+2}^{i-1} and w_{i-n+1}^{i-1} with the former clearly being a superset of the latter. However, when Φ can be arbitrary, e.g., a decision tree, that is not necessarily so.

Let us consider what happens when we have two context sequences W and W' that belong to the same cluster $\Phi_n(W) = \Phi_n(W')$ but different backoff clusters $BO_{n-1}(W) \neq BO_{n-1}(W')$. For example: suppose we have $\Phi(w_{i-2}w_{i-1}) = (\{\text{on}\}, \{\text{may,june}\})$ and two corresponding backoff clusters: $BO' = (\{\text{may}\})$ and $BO'' = (\{\text{june}\})$. Following *on*, the word *may* is likely to be a month rather than a modal verb, although the latter is more frequent and will dominate in BO'. Therefore we have much less faith in $\tilde{p}(w_i|BO')$ than in $\tilde{p}(w_i|BO'')$ and would like a much smaller weight γ assigned to BO', but it is not possible in the backoff scheme in Eq. 4, thus we will have to settle on a compromise value of γ, resulting in suboptimal performance.

We would expect this effect to be more pronounced in higher order models, because viola-

tions of Property 5 are less frequent in lower order models. Indeed, in a 2-gram model, the property is never violated since its backoff, unigram, contains the entire context in one cluster. The 3-gram example above, $\Phi(w_{i-2}w_{i-1}) = (\{\text{on}\}, \{\text{may,june}\})$, although illustrative, is not likely to occur because *may* in w_{i-1} position will likely be split from *june* very early on, since it is very informative about the following word. However, in a 4-gram model, $\Phi(w_{i-3}w_{i-2}w_{i-1}) = (\{\text{on}\}, \{\text{may,june}\}, \{\texttt{<unk>}\})$ is quite plausible.

Thus, arbitrary clustering (an advantage of DTs) leads to violation of Property 5, which, we argue, may lead to a degradation of performance *if* backoff interpolation Eq. 4 is used. In the next section, we generalize the interpolation scheme which, as we show in Section 6, allows us to find a better solution in the face of the violation of Property 5.

4 Linear Interpolation

We use linear interpolation as the baseline, represented recursively, which is similar to Jelinek-Mercer smoothing for n-gram models (Jelinek and Mercer, 1980):

$$\begin{aligned}\tilde{p}_n(w_i|w_{i-n+1}^{i-1}) &= \lambda_n(\phi_n)\cdot p_n(w_i|\phi_n) + \\ &\quad (1-\lambda_n(\phi_n))\cdot \tilde{p}_{n-1}(w_i|w_{i-n+2}^{i-1})\end{aligned} \quad (6)$$

where $\phi_n \equiv \Phi_n(w_{i-n+1}^{i-1})$, and $\lambda_n(\phi_n) \in [0,1]$ are assigned to each cluster and are optimized on a held-out set using EM. $p_n(w_i|\phi_n)$ is the probability distribution at the cluster ϕ_n in the tree of order n. This interpolation method is particularly useful as, unlike count-based discounting methods (e.g., Kneser-Ney), it can be applied to already smooth distributions p_n[2].

5 Generalized Interpolation

We can unwind the recursion in Eq. 6 and make substitutions:

$$\begin{aligned}\lambda_n(\phi_n) &\rightarrow \hat{\lambda}_n(\phi_n) \\ (1-\lambda_n(\phi_n))\cdot\lambda_{n-1}(\phi_{n-1}) &\rightarrow \hat{\lambda}_{n-1}(\phi_{n-1}) \\ &\vdots\end{aligned}$$

$$\begin{aligned}\tilde{p}_n(w_i|w_{i-n+1}^{i-1}) &= \sum_{m=1}^{n}\hat{\lambda}_m(\phi_m)\cdot p_m(w_i|\phi_m) \quad (7)\\ \sum_{m=1}^{n}\hat{\lambda}_m(\phi_m) &= 1\end{aligned}$$

Note that in this parameterization, the weight assigned to $p_{n-1}(w_i|\phi_{n-1})$ is limited by $(1-\lambda_n(\phi_n))$, i.e., the weight assigned to the higher order model.

Ideally we should be able to assign a different set of interpolation weights for every eligible combination of clusters $\phi_n, \phi_{n-1}, \ldots, \phi_1$. However, not only is the number of such combinations extremely large, but many of them will not be observed in the training data, making parameter estimation cumbersome. Therefore, we propose the following parameterization for the interpolation of decision tree models:

$$\tilde{p}_n(w_i|w_{i-n+1}^{i-1}) = \frac{\sum_{m=1}^{n}\lambda_m(\phi_m)\cdot p_m(w_i|\phi_m)}{\sum_{m=1}^{n}\lambda_m(\phi_m)} \quad (8)$$

Note that this parameterization has the same number of parameters as in Eq. 7 (one per cluster in every tree), but the number of degrees of freedom is larger because the the parameters are not constrained to sum to 1, hence the denominator.

In Eq. 8, there is no explicit distinction between higher order and backoff models. Indeed, it acknowledges that lower order models are *not* backoff models when Property 5 is not satisfied. However, it can be shown that Eq. 8 reduces to Eq. 6 if Property 5 holds. Therefore, the new parameterization can be thought of as a generalization of linear interpolation. Indeed, suppose we have the parameterization in Eq. 8 and Property 5. Let us transform this parameterization into Eq. 7 by induction. We define:

$$\Lambda_m \equiv \sum_{k=1}^{m}\lambda_k \;;\; \Lambda_m = \lambda_m + \Lambda_{m-1}$$

where, due to space limitation, we redefine $\lambda_m \equiv \lambda_m(\phi_m)$ and $\Lambda_m \equiv \Lambda_m(\phi_m)$; $\phi_m \equiv \Phi_m(w_1^{i-1})$, i.e., the cluster of model order m, to which the sequence w_1^{i-1} belongs. The lowest order distribution p_1 is not interpolated with anything, hence:

$$\Lambda_1\tilde{p}_1(w_i|\phi_1) = \lambda_1 p_1(w_i|\phi_1)$$

Now the induction step. From Property 5, it follows that $\phi_m \subset \phi_{m-1}$, thus, for all sequences in $\forall_{w_1^n} \in$

[2] In decision trees, the distribution at a cluster (leaf) is often recursively interpolated with its parent node, e.g. (Bahl et al., 1990; Heeman, 1999; Filimonov and Harper, 2009).

	n-gram		DT: Eq. 6 (baseline)		DT: Eq. 8 (generalized)	
order	Jelinek-Mercer	Mod KN	word-tree	syntactic	word-tree	syntactic
2-gram	270.2	261.0	257.8	214.3	258.1	214.6
3-gram	186.5 (31.0%)	174.3 (33.2%)	168.7 (34.6%)	156.8 (26.8%)	168.4 (34.8%)	155.3 (27.6%)
4-gram	177.1 (5.0%)	161.7 (7.2%)	164.0 (2.8%)	156.5 (0.2%)	155.7 (7.5%)	147.1 (5.3%)

Table 1: Perplexity results on PTB WSJ section 23. Percentage numbers in parentheses denote the reduction of perplexity relative to the lower order model of the same type. "Word-tree" and "syntactic" refer to DT models estimated using words only (Eq. 2) and words and tags jointly (Eq. 3).

ϕ_m, we have the same distribution:

$$\begin{aligned}
&\lambda_m p_m(w_i|\phi_m) + \Lambda_{m-1}\tilde{p}_{m-1}(w_i|\phi_{m-1}) = \\
&= \Lambda_m \left(\frac{\lambda_m}{\Lambda_m} p_m(w_i|\phi_m) + \frac{\Lambda_{m-1}}{\Lambda_m} \tilde{p}_{m-1}(w_i|\phi_{m-1}) \right) \\
&= \Lambda_m \left(\hat{\lambda}_m p_m(w_i|\phi_m) + (1-\hat{\lambda}_m) \tilde{p}_{m-1}(w_i|\phi_{m-1}) \right) \\
&= \Lambda_m \tilde{p}_m(w_i|\phi_m) \; ; \hat{\lambda}_m \equiv \frac{\lambda_m}{\Lambda_m}
\end{aligned}$$

Note that the last transformation is because $\phi_m \subset \phi_{m-1}$; had it not been the case, $\tilde{p}_m$ would depend on the combination of ϕ_m and ϕ_{m-1} and require multiple parameters to be represented on its entire domain $w_1^n \in \phi_m$. After n iterations, we have:

$$\sum_{m=1}^{n} \lambda_m(\phi_m) p_m(w_i|\phi_m) = \Lambda_n \tilde{p}_n(w_i|\phi_n); \text{ (cf. Eq. 8)}$$

Thus, we have constructed $\tilde{p}_n(w_i|\phi_n)$ using the same recursive representation as in Eq. 6, which proves that the standard linear interpolation is a special case of the new interpolation scheme, which occurs when the backoff Property 5 holds.

6 Results and Discussion

Models are trained on 35M words of WSJ 94-96 from LDC2008T13. The text was converted into speech-like form, namely numbers and abbreviations were verbalized, text was downcased, punctuation was removed, and contractions and possessives were joined with the previous word (i.e., *they 'll* becomes *they'll*). For syntactic modeling, we used tags comprised of POS tags of the word and its head, as in (Filimonov and Harper, 2009). Parsing of the text for tag extraction occurred after verbalization of numbers and abbreviations but before any further processing; we used an appropriately trained latent variable PCFG parser (Huang and Harper, 2009). For reference, we include n-gram models with Jelinek-Mercer and modified interpolated KN discounting. All models use the same vocabulary of approximately 50k words.

We implemented four decision tree models[3]: two using the interpolation method of (Eq. 6) and two based on the generalized interpolation (Eq. 8). Parameters λ were estimated using the L-BFGS to minimize the entropy on a heldout set. In order to eliminate the influence of all factors other than the interpolation, we used the same decision trees. The perplexity results on WSJ section 23 are presented in Table 1. As we have predicted, the effect of the new interpolation becomes apparent at the 4-gram order, when Property 5 is most frequently violated. Note that we observe similar patterns for both word-tree and syntactic models, with syntactic models outperforming their word-tree counterparts.

We believe that (Xu and Jelinek, 2004) also suffers from violation of Property 5, however, since they use a heuristic method[4] to set backoff weights, it is difficult to ascertain the extent.

7 Conclusion

The main contribution of this paper is the insight that in the standard recursive backoff there is an implied relation between the backoff and the higher order models, which is essential for adequate performance. When this relation is not satisfied other interpolation methods should be employed; hence, we propose a generalization of linear interpolation that significantly outperforms the standard form in such a scenario.

[3]We refer the reader to (Filimonov and Harper, 2009) for details on the tree construction algorithm.

[4]The higher order model was discounted according to KN discounting, while the lower order model could be either a lower order DT (forest) model, or a standard n-gram model, with the former performing slightly better.

References

Lalit R. Bahl, Peter F. Brown, Peter V. de Souza, and Robert L. Mercer. 1990. A tree-based statistical language model for natural language speech recognition. *Readings in speech recognition*, pages 507–514.

Stanley F. Chen and Joshua Goodman. 1999. An empirical study of smoothing techniques for language modeling. *Computer Speech & Language*, 13(4):359–393.

Denis Filimonov and Mary Harper. 2009. A joint language model with fine-grain syntactic tags. In *Proceedings of the EMNLP*.

Peter A. Heeman. 1999. POS tags and decision trees for language modeling. In *Proceedings of the Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora*, pages 129–137.

Zhongqiang Huang and Mary Harper. 2009. Self-Training PCFG grammars with latent annotations across languages. In *Proceedings of the EMNLP 2009*.

Frederick Jelinek and Robert L. Mercer. 1980. Interpolated estimation of markov source parameters from sparse data. In *Proceedings of the Workshop on Pattern Recognition in Practice*, pages 381–397.

Peng Xu and Frederick Jelinek. 2004. Random forests in language modeling. In *Proceedings of the EMNLP*.