Integrating history-length interpolation and classes in language modeling

Hinrich Schütze
Institute for NLP
University of Stuttgart
Germany

Abstract

Building on earlier work that integrates different factors in language modeling, we view (i) backing off to a shorter history and (ii) class-based generalization as two complementary mechanisms of using a larger equivalence class for prediction when the default equivalence class is too small for reliable estimation. This view entails that the classes in a language model should be learned from rare events only and should be preferably applied to rare events. We construct such a model and show that both training on rare events and preferable application to rare events improve perplexity when compared to a simple direct interpolation of class-based with standard language models.

1 Introduction

Language models, probability distributions over strings of words, are fundamental to many applications in natural language processing. The main challenge in language modeling is to estimate string probabilities accurately given that even very large training corpora cannot overcome the inherent sparseness of word sequence data. One way to improve the accuracy of estimation is *class-based generalization*. The idea is that even though a particular word sequence s may not have occurred in the training set (or too infrequently for accurate estimation), the occurrence of sequences similar to s can help us better estimate $p(s)$.

Plausible though this line of reasoning is, the language models most commonly used today do not incorporate class-based generalization. This is partially due to the additional cost of creating classes and using classes as part of the model. But an equally important reason is that most models that integrate class-based information do so by way of a simple interpolation and achieve only a modest improvement in performance.

In this paper, we propose a new type of class-based language model. The key novelty is that we recognize that certain probability estimates are hard to improve based on classes. In particular, the best probability estimate for frequent events is often the maximum likelihood estimator and this estimator is hard to improve by using other information sources like classes or word similarity. We therefore design a model that attempts to focus the effect of class-based generalization on rare events.

Specifically, we propose to employ the same strategy for this that history-length interpolated (HI) models use. We define HI models as models that interpolate the predictions of different-length histories, e.g., $p(w_3|w_1w_2) = \lambda_1(w_1w_2)p'(w_3|w_1w_2) + \lambda_2(w_1w_2)p'(w_3|w_2) + (1 - \lambda_1(w_1w_2) - \lambda_2(w_1w_2))p'(w_3)$ where p' is a simple estimate; in this section, we use $p' = p_{\text{ML}}$, the maximum likelihood estimate, as an example. Jelinek-Mercer (Jelinek and Mercer, 1980) and modified Kneser-Ney (Kneser and Ney, 1995) models are examples of HI models.

HI models address the challenge that frequent events are best estimated by a method close to maximum likelihood by selecting appropriate values for the interpolation weights. For example, if $w_1w_2w_3$ is frequent, then λ_1 will be close to 1, thus ensuring that $p(w_3|w_1w_2) \approx p_{\text{ML}}(w_3|w_1w_2)$ and that the components $p_{\text{ML}}(w_3|w_2)$ and $p_{\text{ML}}(w_3)$, which are unhelpful in this case, will only slightly change the reliable estimate $p_{\text{ML}}(w_3|w_1w_2)$.

Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics, pages 1516–1525,
Portland, Oregon, June 19-24, 2011.

The main contribution of this paper is to propose the same mechanism for class language models. In fact, we will use the interpolation weights of a KN model to determine how much weight to give to each component of the interpolation. The difference to a KN model is merely that the lower-order distribution is not the lower-order KN distribution (as in KN), but instead an interpolation of the lower-order KN distribution and a class-based distribution. We will show that this method of integrating history interpolation and classes significantly increases the performance of a language model.

Focusing the effect of classes on rare events has another important consequence: if this is the right way of using classes, then they should not be formed based on *all events* in the training set, but only based on *rare events*. We show that doing this increases performance.

Finally, we introduce a second discounting method into the model that differs from KN. This can be motivated by the fact that with two sources of generalization (history-length and classes) more probability mass should be allocated to these two sources than to the single source used in KN. We propose a *polynomial discount* and show a significant improvement compared to using KN discounting only.

This paper is structured as follows. Section 2 discusses related work. Section 3 reviews the KN model and introduces two models, the Dupont-Rosenfeld model (a "recursive" model) and a top-level interpolated model, that integrate the KN model (a history interpolation model) with a class model. Section 4 details our experimental setup. Results are presented in Section 5. Based on an analysis of strengths and weaknesses of Dupont-Rosenfeld and top-level interpolated models, we present a new polynomial discounting mechanism that does better than either in Section 6. Section 7 presents our conclusions.

2 Related work

A large number of different class-based models have been proposed in the literature. The well-known model by Brown et al. (1992) is a class sequence model, in which $p(u|w)$ is computed as the product of a class transition probability and an emission probability, $p(g(u)|g(w))p(u|g(u))$, where $g(u)$ is the class of u. Other approaches condition the probability of a class on n-grams of lexical items (as opposed to classes) (Whittaker and Woodland, 2001; Emami and Jelinek, 2005; Uszkoreit and Brants, 2008). In this work, we use the Brown type of model: it is simpler and has fewer parameters. Models that condition classes on lexical n-grams could be extended in a way similar to what we propose here.

Classes have been used with good results in a number of applications, e.g., in speech recognition (Yokoyama et al., 2003), sentiment analysis (Wiegand and Klakow, 2008), and question answering (Momtazi and Klakow, 2009). Classes have also been shown to improve the performance of exponential models (Chen, 2009).

Our use of classes of lexical n-grams for $n > 1$ has several precedents in the literature (Suhm and Waibel, 1994; Kuo and Reichl, 1999; Deligne and Sagisaka, 2000; Justo and Torres, 2009). The novelty of our approach is that we integrate phrase-level classes into a KN model.

Hierarchical clustering (McMahon and Smith, 1996; Zitouni and Zhou, 2007; Zitouni and Zhou, 2008) has the advantage that the size of the class to be used in a specific context is not fixed, but can be chosen at an optimal level of the hierarchy. There is no reason why our non-hierarchical flat model could not be replaced with a hierarchical model and we would expect this to improve results.

The key novelty of our clustering method is that clusters are formed based on rare events in the training corpus. This type of clustering has been applied to other problems before, in particular to unsupervised part-of-speech tagging (Schütze, 1995; Clark, 2003; Reichart et al., 2010). However, the importance of rare events for clustering in language modeling has not been investigated before.

Our work is most similar to the lattice-based language models proposed by Dupont and Rosenfeld (1997). Bilmes and Kirchhoff (2003) generalize lattice-based language models further by allowing arbitrary factors in addition to words and classes. We use a special case of lattice-based language models in this paper. Our contributions are that we introduce the novel idea of rare-event clustering into language modeling and that we show that the modified model performs better than a strong word-trigram

symbol	denotation
$\sum[[w]]$	$\sum_w$ (sum over all unigrams w)
$c(w_j^i)$	count of w_j^i
$n_{1+}(\bullet w_j^i)$	# of distinct w occurring before w_j^i

Table 1: Notation used for Kneser-Ney.

baseline.

3 Models

In this section, we introduce the three models that we compare in our experiments: Kneser-Ney model, Dupont-Rosenfeld model, and top-level interpolation model.

3.1 Kneser-Ney model

Our baseline model is the modified Kneser-Ney (KN) trigram model as proposed by Chen and Goodman (1999). We give a comprehensive description of our implementation of KN because the details are important for the integration of the class model given below. We use the notation in Table 1.

We estimate p_{KN} on the training set as follows.

$$\begin{aligned} p_{\text{KN}}(w_3|w_1^2) &= \frac{c(w_1^3) - d'''(c(w_1^3))}{\sum[[w]]\ c(w_1^2 w)} \\ &\quad +\gamma_3(w_1^2) p_{\text{KN}}(w_3|w_2) \end{aligned}$$

$$\gamma_3(w_1^2) = \frac{\sum[[w]]\ d'''(c(w_1^2 w))}{\sum[[w]]\ c(w_1^2 w)}$$

$$\begin{aligned} p_{\text{KN}}(w_3|w_2) &= \frac{n_{1+}(\bullet w_2^3) - d''(n_{1+}(\bullet w_2^3))}{\sum[[w]]\ n_{1+}(\bullet w_2 w)} \\ &\quad +\gamma_2(w_2) p_{\text{KN}}(w_3) \end{aligned}$$

$$\gamma_2(w_2) = \frac{\sum[[w]]\ d''(n_{1+}(\bullet w_2 w))}{\sum[[w]]\ n_{1+}(\bullet w_2 w)}$$

$$p_{\text{KN}}(w_3) = \begin{cases} \frac{n_{1+}(\bullet w_3) - d'(n_{1+}(\bullet w_3))}{\sum[[w]]\ n_{1+}(\bullet w)} & \text{if } c(w_3) > 0 \\ \gamma_1 & \text{if } c(w_3) = 0 \end{cases}$$

$$\gamma_1 = \frac{\sum[[w]]\ d'(n_{1+}(\bullet w))}{\sum[[w]]\ n_{1+}(\bullet w)}$$

The parameters d', d'', and d''' are the discounts for unigrams, bigrams and trigrams, respectively, as defined by Chen and Goodman (1996, p. 20, (26)). Note that our notation deviates from C&G in that they use the single symbol D_1 for the three different values $d'(1)$, $d''(1)$, and $d'''(1)$ etc.

3.2 Dupont-Rosenfeld model

History-interpolated models attempt to find a good tradeoff between using a maximally informative history for accurate prediction of frequent events and generalization for rare events by using lower-order distributions; they employ this mechanism recursively by progressively shortening the history.

The key idea of the improved model we will adopt is that class generalization ought to play the same role in history-interpolated models as the lower-order distributions: they should improve estimates for unseen and rare events. Following Dupont and Rosenfeld (1997), we implement this idea by linearly interpolating the class-based distribution with the lower order distribution, recursively at each level. For a trigram model, this means that we interpolate $p_{\text{KN}}(w_3|w_2)$ and $p_{\text{B}}(w_3|w_1 w_2)$ on the first backoff level and $p_{\text{KN}}(w_3)$ and $p_{\text{B}}(w_3|w_2)$ on the second backoff level, where p_{B} is the (Brown) class model (see Section 4 for details on p_{B}). We call this model p_{DR} for Dupont-Rosenfeld model and define it as follows:

$$\begin{aligned} p_{\text{DR}}(w_3|w_1^2) = & \frac{c(w_1^3) - d'''(c(w_1^3))}{\sum[[w]]\ c(w_1^2 w)} \\ + & \gamma_3(w_1^2)[\beta_1(w_1^2) p_{\text{B}}(w_3|w_1^2) \\ & +(1 - \beta_1(w_1^2)) p_{\text{DR}}(w_3|w_2)] \end{aligned}$$

$$\begin{aligned} p_{\text{DR}}(w_3|w_2) = & \frac{n_{1+}(\bullet w_2^3) - d''(n_{1+}(\bullet w_2^3))}{\sum[[w]]\ n_{1+}(\bullet w_2 w)} \\ + & \gamma_2(w_2)[\beta_2(w_2) p_{\text{B}}(w_3|w_2) \\ & +(1 - \beta_2(w_2)) p_{\text{DR}}(w_3)] \end{aligned}$$

where $\beta_i(v)$ is equal to a parameter α_i if the history (w_1^2 or w_2) is part of a cluster and 0 otherwise:

$$\beta_i(v) = \begin{cases} \alpha_i & \text{if } v \in \mathcal{B}_{2-(i-1)} \\ 0 & \text{otherwise} \end{cases}$$

$\mathcal{B}_1$ (resp. $\mathcal{B}_2$) is the set of unigram (resp. bigram) histories that is covered by the clusters. We cluster bigram histories and unigram histories separately and write $p_{\text{B}}(w_3|w_1 w_2)$ for the bigram cluster model and $p_{\text{B}}(w_3|w_2)$ for the unigram cluster model. Clustering and the estimation of these two distributions are described in Section 4.

The unigram distribution of the Dupont-Rosenfeld model is set to the unigram distribution of the KN model: $p_{\text{DR}}(w) = p_{\text{KN}}(w)$.

The model (or family of models) defined by Dupont and Rosenfeld (1997) is more general than our version p_{DR}. Most importantly, it allows a truly parallel backoff whereas in our model the recursive backoff distribution p_{DR} is interpolated with a class distribution p_{B} that is not backed off. We prefer this version because it makes it easier to understand the contribution that unique-event vs. all-event classes make to improved language modeling; the parameters β are a good indicator of this effect.

An alternative way of setting up the Dupont-Rosenfeld model would be to interpolate $p_{\text{KN}}(w_3|w_1w_2)$ and $p_{\text{B}}(w_3|w_1w_2)$ etc – but this is undesirable. The strength of history interpolation is that estimates for frequent events are close to ML, e.g., $p_{\text{KN}}(\text{share}|\text{cents a}) \approx p_{\text{ML}}(\text{share}|\text{cents a})$ for our corpus. An ML estimate is accurate for large counts and we should not interpolate it directly with $p_{\text{B}}(w_3|w_1w_2)$. For p_{DR}, the discount d''' that is subtracted from $c(w_1w_2w_3)$ is small relative to $c(w_1w_2w_3)$ and therefore $p_{\text{DR}} \approx p_{\text{ML}}$ in this case (exactly as in p_{KN}).

3.3 Top-level interpolation

Class-based models are often combined with other models by interpolation, starting with the work by Brown et al. (1992). Since we cluster both unigrams and bigrams, we interpolate three models:

$$\begin{aligned} & p_{\text{TOP}}(w_3|w_1w_2) \\ = \; & \mu_1(w_1w_2)p_{\text{B}}(w_3|w_1w_2) + \mu_2(w_2)p_{\text{B}}(w_3|w_2) \\ + \; & (1 - \mu_1(w_1w_2) - \mu_2(w_2))p_{\text{KN}}(w_3|w_1w_2) \end{aligned}$$

where $\mu_1(w_1w_2) = \lambda_1$ if $w_1w_2 \in \mathcal{B}_2$ and 0 otherwise, $\mu_2(w_2) = \lambda_2$ if $w_2 \in \mathcal{B}_1$ and 0 otherwise and λ_1 and λ_2 are parameters. We call this the *top-level model* p_{TOP} because it interpolates the three models at the top level. Most previous work on class-based model has employed some form of top-level interpolation.

4 Experimental Setup

We run experiments on a Wall Street Journal (WSJ) corpus of 50M words, split 8:1:1 into training, validation and test sets. The training set contains 256,873 unique unigrams and 4,494,222 unique bigrams. Unknown words in validation and test sets are mapped to a special unknown word u.

We use the SRILM toolkit (Stolcke, 2002) for clustering. An important parameter of the class-based model is size $|\mathcal{B}_i|$ of the base set, i.e., the total number of n-grams (or rather i-grams) to be clustered. As part of the experiments we vary $|\mathcal{B}_i|$ systematically to investigate the effect of base set size. We cluster unigrams ($i = 1$) and bigrams ($i = 2$). For all experiments, $|\mathcal{B}_1| = |\mathcal{B}_2|$ (except in cases where $|\mathcal{B}_2|$ exceeds the number of unigrams, see below). SRILM does not directly support bigram clustering. We therefore represent a bigram as a hyphenated word in bigram clustering; e.g., *Pan Am* is represented as *Pan-Am*.

The input to the clustering is the vocabulary $\mathcal{B}_i$ and the cluster training corpus. For a particular base set size b, the unigram input vocabulary $\mathcal{B}_1$ is set to the b most frequent unigrams in the training set and the bigram input vocabulary $\mathcal{B}_2$ is set to the b most frequent bigrams in the training set.

In this section, we call the WSJ training corpus the *raw corpus* and the cluster training corpus the *cluster corpus* to be able to distinguish them. We run four different clusterings for each base set size (except for the large sets, see below). The cluster corpora are constructed as follows.

- **All-event unigram clustering.** The cluster corpus is simply the raw corpus.
- **All-event bigram clustering.** The cluster corpus is constructed as follows. A sentence of the raw corpus that contains s words is included twice, once as a sequence of the $\lfloor s/2 \rfloor$ bigrams "w_1–w_2 w_3–w_4 w_5–w_6 ..." and once as a sequence of the $\lfloor (s-1)/2 \rfloor$ bigrams "w_2–w_3 w_4–w_5 w_6–w_7 ...".
- **Unique-event unigram clustering.** The cluster corpus is the set of all sequences of two unigrams $\in \mathcal{B}_1$ that occur in the raw corpus, one sequence per line. Each sequence occurs only once in this cluster corpus.
- **Unique-event bigram clustering.** The cluster corpus is the set of all sequences of two bigrams $\in \mathcal{B}_2$ that occur in the training corpus,

one sequence per line. Each sequence occurs only once in this cluster corpus.

As mentioned above, we need both unigram and bigram clusters because we want to incorporate class-based generalization for histories of lengths 1 *and* 2. As we will show below this significantly increases performance. Since the focus of this paper is not on clustering algorithms, reformatting the training corpus as described above (as a sequence of hyphenated bigrams) is a simple way of using SRILM for bigram clustering.

The unique-event clusterings are motivated by the fact that in the Dupont-Rosenfeld model, frequent events are handled by discounted ML estimates. Classes are only needed in cases where an event was not seen or was not frequent enough in the training set. Consequently, we should form clusters not based on all events in the training corpus, but only on events that are rare – because this is the type of event that classes will then be applied to in prediction.

The two unique-event corpora can be thought of as reweighted collections in which each unique event receives the same weight. In practice this means that clustering is mostly influenced by rare events since, on the level of types, most events are rare. As we will see below, rare-event clusterings perform better than all-event clusterings. This is not surprising as the class-based component of the model can only benefit rare events and it is therefore reasonable to estimate this component based on a corpus dominated by rare events.

We started experimenting with reweighted corpora because class sizes become very lopsided in regular SRILM clustering as the size of the base set increases. The reason is that the objective function maximizes mutual information. Highly differentiated classes for frequent words contribute substantially to this objective function whereas putting all rare words in a few large clusters does not hurt the objective much. However, our focus is on using clustering for improving prediction for rare events; this means that the objective function is counterproductive when contexts are frequency-weighted as they occur in the corpus. After overweighting rare contexts, the objective function is more in sync with what we use clusters for in our model.

p_{ML}	maximum likelihood
p_{B}	Brown cluster model
p_{E}	cluster emission probability
p_{T}	cluster transition probability
p_{KN}	KN model
p_{DR}	Dupont-Rosenfeld model
p_{TOP}	top-level interpolation
p_{POLKN}	KN and polynomial discounting
p_{POL0}	polynomial discounting only

Table 2: Key to probability distributions

It is important to note that the same intuition underlies unique-event clustering that also motivates using the "unique-event" distributions $n_{1+}(\bullet w_2^3)/(\sum n_{1+}(\bullet w_2 w))$ and $n_{1+}(\bullet w_3)/(\sum n_{1+}(\bullet w))$ for the backoff distributions in KN. Viewed this way, the basic KN model also uses a unique-event corpus (although a different one) for estimating backoff probabilities.

In all cases, we set the number of clusters to $k = 512$. Our main goal in this paper is to compare different ways of setting up history-length/class interpolated models and we do not attempt to optimize k. We settled on a fixed number of $k = 512$ because Brown et al. (1992) used a total of 1000 classes. 512 unigram classes and 512 bigram classes roughly correspond to this number. We prefer powers of 2 to facilitate efficient storage of cluster ids (one such cluster id must be stored for each unigram and each bigram) and therefore choose $k = 512$. Clustering was performed on an Opteron 8214 processor and took from several minutes for the smallest base sets to more than a week for the largest set of 400,000 items.

To estimate n-gram emission probabilities p_{E}, we first introduce an additional cluster for all unigrams that are not in the base set; emission probabilities are then estimated by maximum likelihood. Cluster transition probabilities p_{T} are computed using add-one smoothing. Both p_{E} and p_{T} are estimated on the raw corpus. The two class distributions are then defined as follows:

$$p_{\text{B}}(w_3|w_1w_2) = p_{\text{T}}(g(w_3)|g(w_1w_2))p_{\text{E}}(w_3|g(w_3))$$
$$p_{\text{B}}(w_3|w_2) = p_{\text{T}}(g(w_3)|g(w_2))p_{\text{E}}(w_3|g(w_3))$$

where $g(v)$ is the class of the uni- or bigram v.

	p_{DR}						p_{TOP}						
	all events			unique events				all events			unique events		
	$\lvert\mathcal{B}_i\rvert$	α_1	α_2	perp.	α_1	α_2	perp.	$\lvert\mathcal{B}_i\rvert$	λ_1	λ_2	perp.	λ_1	λ_2	perp.
1a	1×10^4	.20	.40	87.42	.2	.4	87.41	1b 1×10^4	.020	.03	87.65	.02	.02	87.71
2a	2×10^4	.20	.50	86.97	.2	.5	86.88	2b 2×10^4	.030	.04	87.43	.03	.03	87.47
3a	3×10^4	.10	.40	87.14	.2	.5	86.57	3b 3×10^4	.020	.03	87.52	.03	.03	87.34
4a	4×10^4	.10	.40	87.22	.3	.5	86.31	4b 4×10^4	.010	.04	87.58	.03	.04	87.24
5a	5×10^4	.05	.30	87.54	.3	.6	86.10	5b 5×10^4	.003	.03	87.74	.03	.04	87.15
6a	6×10^4	.01	.30	87.71	.3	.6	85.96	6b 6×10^4	.000	.02	87.82	.03	.04	87.09

Perplexity of KN model: 88.03

Table 3: Optimal parameters for Dupont-Rosenfeld (left) and top-level (right) models on the validation set and perplexity on the validation set. The two tables compare performance when using a class model trained on all events vs a class model trained on unique events. $|\mathcal{B}_1| = |\mathcal{B}_2|$ is the number of unigrams and bigrams in the clusters; e.g., lines 1a and 1b are for models that cluster 10,000 unigrams and 10,000 bigrams.

Table 2 is a key to the probability distributions we use.

5 Results

Table 3 shows the performance of p_{DR} and p_{TOP} for a range of base set sizes $|\mathcal{B}_i|$ and for classes trained on all events and on unique events. Parameters α_i and λ_i are optimized on the validation set. Perplexity is reported for the validation set. All following tables also optimize on the validation set and report results on the validation set. The last table, Table 7, also reports perplexity for the test set.

Table 3 confirms previous findings that classes improve language model performance. All models have a perplexity that is lower than KN (88.03).

When comparing all-event and unique-event clusterings, a clear tendency is apparent. In all-event clustering, the best performance is reached for $|\mathcal{B}_i| = 20000$: perplexity is 86.97 with this base set size for p_{DR} (line 2a) and 87.43 for p_{TOP} (line 2b). In unique-event clustering, performance keeps improving with larger and larger base sets; the best perplexities are obtained for $|\mathcal{B}_i| = 60000$: 85.96 for p_{DR} and 87.09 for p_{TOP} (lines 6a, 6b).

The parameter values also reflect this difference between all-event and unique-event clustering. For unique-event results of p_{DR}, we have $\alpha_1 \geq .2$ and $\alpha_2 \geq .4$ (1a–6a). This indicates that classes and history interpolation are both valuable when the model is backing off. But for all-event clustering, the values of α_i decrease: from a peak of .20 and .50 (2a) to .01 and .30 (6a), indicating that with larger base sets, less and less value can be derived from classes. This again is evidence that rare-event clustering is the correct approach: only clusters derived in rare-event clustering receive high weights α_i in the interpolation.

This effect can also be observed for p_{TOP}: the value of λ_1 (the weight of bigrams) is higher for unique-event clustering than for all-event clustering (with the exception of lines 1b&2b). The quality of bigram clusters seems to be low in all-event clustering when the base set becomes too large.

Perplexity is generally lower for unique-event clustering than for all-event clustering: this is the case for all values of $|\mathcal{B}_i|$ for p_{DR} (1a–6a); and for $|\mathcal{B}_i| > 20000$ for p_{TOP} (3b–6b).

Table 4 compares the two models in two different conditions: (i) b-: using unigram clusters only and (ii) b+: using unigram clusters and bigram clusters. For all events, there is no difference in performance. However, for unique events, the model that includes bigrams (b+) does better than the model without bigrams (b-). The effect is larger for p_{DR} than for p_{TOP} because (for unique events) a larger weight for the unigram model ($\lambda_2 = .05$ instead of $\lambda_2 = .04$) apparently partially compensates for the missing bigram clusters.

Table 3 shows that rare-event models do better than all-event models. Given that training large class models with SRILM on all events would take several weeks or even months, we restrict our direct

	p_{DR}						p_{TOP}					
	all			unique			all			unique		
	α_1	α_2	perp.	α_1	α_2	perp.	λ_1	λ_2	perp.	λ_1	λ_2	perp.
b-		.3	87.71		.5	86.62		.02	87.82		.05	87.26
b+	.01	.3	87.71	.3	.6	85.96	0	.02	87.82	.03	.04	87.09

Table 4: Using both unigram and bigram clusters is better than using unigrams only. Results for $|\mathcal{B}_i| = 60{,}000$.

	p_{DR}			p_{TOP}		
$|\mathcal{B}_i|$	α_1	α_2	perp.	λ_1	λ_2	perp.
1 $6{\times}10^4$	0.3	0.6	85.96	0.03	0.04	87.09
2 $1{\times}10^5$	0.3	0.6	85.59	0.04	0.04	86.93
3 $2{\times}10^5$	0.3	0.6	85.20	0.05	0.04	86.77
4 $4{\times}10^5$	0.3	0.7	85.14	0.05	0.04	86.74

Table 5: Dupont-Rosenfeld and top-level models for $|\mathcal{B}_i| \in \{60000, 100000, 200000, 400000\}$. Clustering trained on unique-event corpora.

comparison of all-event and rare-event models to $|\mathcal{B}_i| \leq 60,000$ in Tables 3-4 and report only rare-event numbers for $|\mathcal{B}_i| > 60,000$ in what follows.

As we can see in Table 5, the trends observed in Table 3 continue as $|\mathcal{B}_i|$ is increased further. For both models, perplexity steadily decreases as $|\mathcal{B}_i|$ is increased from 60,000 to 400,000. (Note that for $|\mathcal{B}_i| = 400000$, the actual size of $\mathcal{B}_1$ is 256,873 since there are only that many words in the training corpus.) The improvements in perplexity become smaller for larger base set sizes, but it is reassuring to see that the general trend continues for large base set sizes. Our explanation is that the class component is focused on rare events and the items that are being added to the clustering for large base sets are all rare events.

The perplexity for p_{DR} is clearly lower than that of p_{TOP}, indicating the superiority of the Dupont-Rosenfeld model.[1]

[1]Dupont and Rosenfeld (1997) found a relatively large improvement of the "global" linear interpolation model – p_{top} in our terminology – compared to the baseline whereas p_{top} performs less well in our experiments. One possible explanation is that our KN baseline is stronger than the word trigram baseline they used.

6 Polynomial discounting

Further comparative analysis of p_{DR} and p_{TOP} revealed that p_{DR} is not uniformly better than p_{TOP}. We found that p_{TOP} does poorly on frequent events. For example, for the history $w_1w_2 =$ *cents a*, the continuation $w_3 =$ *share* dominates. p_{DR} deals well with this situation because $p_{\text{DR}}(w_3|w_1w_2)$ is the discounted ML estimate, with a discount that is small relative to the 10,768 occurrences of *cents a share* in the training set. In the p_{TOP} model on the last line in Table 5, the discounted ML estimate is multiplied by $1 - .05 - .04 = .91$, which results in a much less accurate estimate of $p_{\text{TOP}}(\text{share}|\text{cents a})$.

In contrast, p_{TOP} does well for productive histories, for which it is likely that a continuation unseen in the training set will occur. An example is the history *in the* – almost any adjective or noun can follow. There are 6251 different words that (i) occur after *in the* in the validation set, (ii) did not occur after *in the* in the training set, and (iii) occurred at least 10 times in the training set. Because their training set unigram frequency is at least 10, they have a good chance of being assigned to a class that captures their distributional behavior well and $p_{\text{B}}(w_3|w_1w_2)$ is then likely to be a good estimate. For a history with these properties, it is advantageous to further discount the discounted ML estimates by multiplying them with .91. p_{TOP} then gives the remaining probability mass of .09 to words w_3 whose probability would otherwise be underestimated.

What we have just described is already partially addressed by the KN model – $\gamma(v)$ will be relatively large for a productive history like $v =$ *in the*. However, it looks like the KN discounts are not large enough for productive histories, at least not in a combined history-length/class model. Apparently, when incorporating the strengths of a class-based model into KN, the default discounting mechanism does not reallocate enough probability mass

from high-frequency to low-frequency events. We conclude from this analysis that we need to increase the discount values d for large counts.

We could add a constant to d, but one of the basic premises of the KN model, derived from the assumption that n-gram marginals should be equal to relative frequencies, is that the discount is larger for more frequent n-grams although in many implementations of KN only the cases $c(w_1^3) = 1$, $c(w_1^3) = 2$, and $c(w_1^3) \geq 3$ are distinguished.

This suggests that the ideal discount $d(x)$ in an integrated history-length/class language model should grow monotonically with $c(v)$. The simplest way of implementing this heuristically is a polynomial of form ρx^r where ρ and r are parameters. r controls the rate of growth of the discount as a function of x; ρ is a factor that can be scaled for optimal performance.

The incorporation of the additional polynomial discount into KN is straightforward. We use a discount function $e(x)$ that is the sum of $d(x)$ and the polynomial:

$$e(x) = d(x) + \begin{cases} \rho x^r & \text{for } x \geq 4 \\ 0 & \text{otherwise} \end{cases}$$

where $(e, d) \in \{(e', d'), (e'', d''), (e''', d''')\}$. This model is identical to p_{DR} except that d is replaced with e. We call this model p_{POLKN}. p_{POLKN} directly implements the insight that, when using class-based generalization, discounts for counts $x \geq 4$ should be larger than they are in KN.

We also experiment with a second version of the model:

$$e(x) = \rho x^r$$

This second model, called p_{POL0}, is simpler and does not use KN discounts. It allows us to determine whether a polynomial discount by itself (without using KN discounts in addition) is sufficient.

Results for the two models are shown in Table 6 and compared with the two best models from Table 5, for $|\mathcal{B}_i| = 400{,}000$, classes trained on unique events. p_{POLKN} and p_{POL0} achieve a small improvement in perplexity when compared to p_{DR} (line 3&4 vs 2). This shows that using discounts that are larger than KN discounts for large counts is potentially advantageous.

		α_1/λ_1	α_2/λ_2	ρ	r	perp.
1	p_{TOP}	.05	.04			86.74
2	p_{DR}	.30	.70			85.14
3	p_{POLKN}	.30	.70	.05	.89	85.01
4	p_{POL0}	.30	.70	.80	.41	84.98

Table 6: Results for polynomial discounting compared to p_{DR} and p_{TOP}. $|\mathcal{B}_i| = 400{,}000$, clusters trained on unique events.

	tb:l	model	$\lvert\mathcal{B}_i\rvert$		perplexity val	test
1	3	p_{KN}			88.03	88.28
2	3:6a	p_{DR}	$6{\times}10^4$	ae b+	87.71	87.97
3	3:6a	p_{DR}	$6{\times}10^4$	ue b+	85.96	86.22
4	3:6b	p_{TOP}	$6{\times}10^4$	ae b+	87.82	88.08
5	3:6b	p_{TOP}	$6{\times}10^4$	ue b+	87.09	87.35
6	4	p_{DR}	$6{\times}10^4$	ae b-	87.71	87.97
7	4	p_{DR}	$6{\times}10^4$	ue b-	86.62	86.88
8	4	p_{TOP}	$6{\times}10^4$	ae b-	87.82	88.08
9	4	p_{TOP}	$6{\times}10^4$	ue b-	87.26	87.51
10	5:4	p_{DR}	$2{\times}10^5$	ue b+	85.14	85.39
11	5:4	p_{TOP}	$2{\times}10^5$	ue b+	86.74	86.98
12	6:3	p_{POLKN}	$4{\times}10^5$	ue b+	85.01	85.26
13	6:4	p_{POL0}	$4{\times}10^5$	ue b+	84.98	85.22

Table 7: Performance of key models on validation and test sets. tb:l = Table and line the validation result is taken from. ae/ue = all-event/unique-event. b- = unigrams only. b+ = bigrams and unigrams.

The linear interpolation $\alpha p + (1 - \alpha) q$ of two distributions p and q is a form of linear discounting: p is discounted by $1 - \alpha$ and q by α. See (Katz, 1987; Jelinek, 1990; Ney et al., 1994). It can thus be viewed as polynomial discounting for $r = 1$. Absolute discounting could be viewed as a form of polynomial discounting for $r = 0$. We know of no other work that has explored exponents between 0 and 1 and shown that for this type of exponent, one obtains competitive discounts that could be argued to be simpler than more complex discounts like KN discounts.

6.1 Test set performance

We report the test set performance of the key models we have developed in this paper in Table 7. The experiments were run with the optimal parameters

on the validation set as reported in the table referenced in column "tb:l"; e.g., on line 2 of Table 7, $(\alpha_1, \alpha_2) = (.01, .3)$ as reported on line 6a of Table 3.

There is an almost constant difference between validation and test set perplexities, ranging from +.2 to +.3, indicating that test set results are consistent with validation set results. To test significance, we assigned the 2.8M positions in the test set to 48 different bins according to the majority part-of-speech tag of the word in the training set.[2] We can then compute perplexity for each bin, compare perplexities for different experiments and use the sign test for determining significance. We indicate results that were significant at $p < .05$ ($n = 48$, $k \geq 32$ successes) using a star, e.g., $3 <^* 2$ means that test set perplexity on line 3 is significantly lower than test set perplexity on line 2.

The main findings on the validation set also hold for the test set: (i) Trained on unique events and with a sufficiently large $|\mathcal{B}_i|$, both p_{DR} and p_{TOP} are better than KN: $10 <^* 1$, $11 <^* 1$. (ii) Training on unique events is better than training on all events: $3 <^* 2$, $5 <^* 4$, $7 <^* 6$, $9 <^* 8$. (iii) For unique events, using bigram and unigram classes gives better results than using unigram classes only: $3 <^* 7$. Not significant: $5 < 9$. (iv) The Dupont-Rosenfeld model p_{DR} is better than the top-level model p_{TOP}: $10 <^* 11$. (v) The model POL0 (polynomial discounting) is the best model overall: Not significant: $13 < 12$. (vi) Polynomial discounting is significantly better than KN discounting for the Dupont-Rosenfeld model p_{DR} although the absolute difference in perplexity is small: $13 <^* 10$.

Overall, p_{DR} and p_{POL0} achieve considerable reductions in test set perplexity from 88.28 to 85.39 and 85.22, respectively. The main result of the experiments is that Dupont-Rosenfeld models (which focus on rare events) are better than the standardly used top-level models; and that training classes on unique events is better than training classes on all events.

[2] Words with a rare majority tag (e.g., FW 'foreign word') and unknown words were assigned to a special class OTHER.

7 Conclusion

Our hypothesis was that classes are a generalization mechanism for rare events that serves the same function as history-length interpolation and that classes should therefore be (i) primarily trained on rare events and (ii) receive high weight only if it is likely that a rare event will follow and be weighted in a way analogous to the weighting of lower-order distributions in history-length interpolation.

We found clear statistically significant evidence for both (i) and (ii). (i) Classes trained on unique-event corpora perform better than classes trained on all-event corpora. (ii) The p_{DR} model (which adjusts the interpolation weight given to classes based on the prevalence of nonfrequent events following) is better than top-level model p_{TOP} (which uses a fixed weight for classes). Most previous work on class-based models has employed top-level interpolation. Our results strongly suggest that the Dupont-Rosenfeld model is a superior model.

A comparison of Dupont-Rosenfeld and top-level results suggested that the KN discount mechanism does not discount high-frequency events enough. We empirically determined that better discounts are obtained by letting the discount grow as a function of the count of the discounted event and implemented this as polynomial discounting, an arguably simpler way of discounting than Kneser-Ney discounting. The improvement of polynomial discounts vs. KN discounts was small, but statistically significant.

In future work, we would like to find a theoretical justification for the surprising fact that polynomial discounting does at least as well as Kneser-Ney discounting. We also would like to look at other backoff mechanisms (in addition to history length and classes) and incorporate them into the model, e.g., similarity and topic. Finally, training classes on unique events is an extreme way of highly weighting rare events. We would like to explore training regimes that lie between unique-event clustering and all-event clustering and upweight rare events less.

Acknowledgements. This research was funded by Deutsche Forschungsgemeinschaft (grant SFB 732). We are grateful to Thomas Müller, Helmut Schmid and the anonymous reviewers for their helpful comments.

References

Jeff Bilmes and Katrin Kirchhoff. 2003. Factored language models and generalized parallel backoff. In *HLT-NAACL*.

Peter F. Brown, Vincent J. Della Pietra, Peter V. de Souza, Jennifer C. Lai, and Robert L. Mercer. 1992. Class-based *n*-gram models of natural language. *Computational Linguistics*, 18(4):467–479.

Stanley F. Chen and Joshua Goodman. 1996. An empirical study of smoothing techniques for language modeling. *CoRR*, cmp-lg/9606011.

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

Stanley F. Chen. 2009. Shrinking exponential language models. In *HLT/NAACL*, pages 468–476.

Alexander Clark. 2003. Combining distributional and morphological information for part of speech induction. In *EACL*, pages 59–66.

Sabine Deligne and Yoshinori Sagisaka. 2000. Statistical language modeling with a class-based *n*-multigram model. *Computer Speech & Language*, 14(3):261–279.

Pierre Dupont and Ronald Rosenfeld. 1997. Lattice based language models. Technical Report CMU-CS-97-173, Carnegie Mellon University.

Ahmad Emami and Frederick Jelinek. 2005. Random clustering for language modeling. In *ICASSP*, volume 1, pages 581–584.

Frederick Jelinek and Robert L. Mercer. 1980. Interpolated estimation of Markov source parameters from sparse data. In Edzard S. Gelsema and Laveen N. Kanal, editors, *Pattern Recognition in Practice*, pages 381–397. North-Holland.

Frederick Jelinek. 1990. Self-organized language modeling for speech recognition. In Alex Waibel and Kai-Fu Lee, editors, *Readings in speech recognition*, pages 450–506. Morgan Kaufmann.

Raquel Justo and M. Inés Torres. 2009. Phrase classes in two-level language models for ASR. *Pattern Analysis & Applications*, 12(4):427–437.

Slava M. 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 *ICASSP*, volume 1, pages 181–184.

Hong-Kwang J. Kuo and Wolfgang Reichl. 1999. Phrase-based language models for speech recognition. In *European Conference on Speech Communication and Technology*, volume 4, pages 1595–1598.

John G. McMahon and Francis J. Smith. 1996. Improving statistical language model performance with automatically generated word hierarchies. *Computational Linguistics*, 22:217–247.

Saeedeh Momtazi and Dietrich Klakow. 2009. A word clustering approach for language model-based sentence retrieval in question answering systems. In *ACM Conference on Information and Knowledge Management*, pages 1911–1914.

Hermann Ney, Ute Essen, and Reinhard Kneser. 1994. On structuring probabilistic dependencies in stochastic language modelling. *Computer Speech and Language*, 8:1–38.

Roi Reichart, Omri Abend, and Ari Rappoport. 2010. Type level clustering evaluation: new measures and a pos induction case study. In *Proceedings of the Fourteenth Conference on Computational Natural Language Learning*, pages 77–87.

Hinrich Schütze. 1995. Distributional part-of-speech tagging. In *EACL 7*, pages 141–148.

Andreas Stolcke. 2002. SRILM - An extensible language modeling toolkit. In *International Conference on Spoken Language Processing*, pages 901–904.

Bernhard Suhm and Alex Waibel. 1994. Towards better language models for spontaneous speech. In *International Conference on Spoken Language Processing*, pages 831–834.

Jakob Uszkoreit and Thorsten Brants. 2008. Distributed word clustering for large scale class-based language modeling in machine translation. In *Annual Meeting of the Association for Computational Linguistics*, pages 755–762.

E.W.D. Whittaker and P.C. Woodland. 2001. Efficient class-based language modelling for very large vocabularies. In *ICASSP*, volume 1, pages 545–548.

Michael Wiegand and Dietrich Klakow. 2008. Optimizing language models for polarity classification. In *ECIR*, pages 612–616.

T. Yokoyama, T. Shinozaki, K. Iwano, and S. Furui. 2003. Unsupervised class-based language model adaptation for spontaneous speech recognition. In *ICASSP*, volume 1, pages 236–239.

Imed Zitouni and Qiru Zhou. 2007. Linearly interpolated hierarchical n-gram language models for speech recognition engines. In Michael Grimm and Kristian Kroschel, editors, *Robust Speech Recognition and Understanding*, pages 301–318. I-Tech Education and Publishing.

Imed Zitouni and Qiru Zhou. 2008. Hierarchical linear discounting class n-gram language models: A multi-level class hierarchy approach. In *International Conference on Acoustics, Speech, and Signal Processing*, pages 4917–4920.