

Hub4 Language Modeling Using Domain Interpolation and Data Clustering

Fuliang Weng Andreas Stolcke Ananth Sankar

Speech Technology And Research Laboratory
SRI International
Menlo Park, California

ABSTRACT

In SRI's language modeling experiments for the Hub4 domain, three basic approaches were pursued: interpolating multiple models estimated from Hub4 and non-Hub4 training data, adapting the language model (LM) to the focus conditions, and adapting the LM to different topic types.

In the first approach, we built separate LMs for the closely transcribed Hub4 material (acoustic training transcripts) and the loosely transcribed Hub4 material (LM training data), as well as the North-American Business News (NABN) and Switchboard training data, projected onto the Hub4 vocabulary. By interpolating the probabilities obtained from these models, we obtained a 20% reduction in perplexity and a 1.8% reduction in word error rate, compared to a baseline Hub4-only language model.

Two adaptation approaches are also described: adapting language models to the speech styles correlated with different focus conditions, and building cluster-specific LM mixtures. These two approaches give some reduction in perplexity, but no significant reduction in word error.

Finally, we identify the problems and future directions of our work.

1. Introduction

Statistical language models (LMs) will ideally achieve best performance when the LM training data is drawn from the same underlying source as the test speech. However, in many practical situations not enough training data are available, or the test speech source is constantly changing. For example, Hub4 data consist of speech of different topics and styles such as planned speech and spontaneous speech about politics and the stock market. One way to deal with this problem is to use data from many different sources so as to cover various sublanguages in the test source. Another approach is to adapt the LM to the test speech.

We describe our language modeling work for this Hub4 benchmark along these two directions: interpolating multiple models estimated from Hub4 and non-Hub4 training data, and adapting the LMs to the focus conditions of the test data and to the topic of an article.

Section 2 describes the Hub4 baseline LM. Section 3 discusses the use of non-Hub4 data sources for LM training. Section 4 describes the fourgram LM used in the SRI evaluation system. In Section 5, we discuss adapting LMs to the Hub4 focus conditions. Section 6 presents cluster-specific LM adaptation. Section 7 gives a brief summary of our work and future directions.

2. The Baseline LM

The recognition process in our system consists of three passes. The first two passes, lattice and N-best list generation, use a bigram LM. The N-best lists are rescored using a higher-order N-gram LM.

Two kinds of training material were available from the Hub4 domain. There were about 130 million words of loosely transcribed broadcast shows (Hub4 LM training data), as well as 380,000 words of closely transcribed material for acoustic training. While the first corpus is much larger, it does not always faithfully represent spontaneous speech phenomena (such as disfluencies). In addition, the verbalization of certain acronyms, Internet addresses, etc., is not transcribed accurately. For example, the Hub4 LM training data would have words such as "www.cnn.com", which are actually verbalized as "w. w. dot c. n. n. dot com". For these reasons, we decided to treat these two sets of transcripts separately, and to give more weight to the acoustic training material, relative to the corpus sizes.

To select the recognizer vocabulary, we first included all words occurring at least twice in the acoustic training data. We then added words from the LM training data, in order of frequency, until reaching the target size of 20,000 words. The relatively small vocabulary size was chosen to allow faster experimentation. The out-of-vocabulary rate on the development test set was 2.2%.

To obtain the recognizer bigram LM, we built separate LMs from the Hub4 LM and acoustic training corpora, called H4_LM and H4_AC, respectively. These models were then interpolated linearly so as to optimize the perplexity on the development test data, giving weight 0.7 to H4_LM and 0.3 to H4_AC. As expected, the weight assigned to H4_AC is disproportionately high relative to the sizes of the training data.

We also built a corresponding interpolated trigram model, which was used in the rescoring pass. Table 1 gives perplexities and word error rates (WER) obtained with both bigram and trigram baseline models.

For comparison, we also rescored with a trigram model trained only from Hub4 LM training data (H4_LM). This model has a 17% higher perplexity and slightly higher word error rate than the combined H4_LM and H4_AC model, confirming the advantage of a separate weighting of the acoustic training data.

3. Using Non-Hub4 Training Data

To improve the coverage and robustness of our baseline LM, we used training material from two generally available non-Hub4 databases. The Switchboard corpus [3] contains conversational speech collected over the telephone and can supplement the coverage of various spontaneous speech phenomena. The North American Business News

Model	PPL	WER
H4, bigram (1-pass LM)	242	37.0%
H4, trigram	174	33.8%
H4_LM, trigram	204	34.0%

Table 1: Results of bigram and trigram Hub4 LMs.

(NABN) corpus used in the 1995 CSR evaluation provides additional coverage of business and politics.

As before, we combine the various data sources through linear interpolation of LMs. Separate backoff LMs [5] were estimated for each of the four data sources - Hub4 LM data, Hub4 acoustic data, Switchboard, and NABN - restricting N-grams to the Hub4 20,000 word vocabulary. Word probabilities from the individual models were interpolated linearly. Interpolation weights were optimized by minimizing the perplexity on the Hub4 development data. Thus, the word probability in the combined LM is computed as:

$$\begin{aligned}
 P(w | h) = & .64 * P(w | h, H4_LM) + \\
 & .14 * P(w | h, H4_AC) + \\
 & .16 * P(w | h, NABN) + \\
 & .06 * P(w | h, SWB)
 \end{aligned}$$

where H4_LM, H4_AC, NABN, and SWB are trigram LMs for Hub4 LM data, Hub4 acoustic data, NABN, and Switchboard, respectively.

Table 2 lists the N-best rescoring results for various interpolated LMs, showing the contributions of the individual data sources (rescoring results are given only for the baseline and the full model). As can be seen, adding new data sources consistently improves performance, although the contribution of the Switchboard data is marginal.

4. Fourgram LM Used in the Evaluation System

The LM applied to final rescoring in our evaluation system used all the data sources discussed in section 3. However, for an additional improvement, we decided to replace the trigram model component with the largest weight, H4_LM, with a corresponding fourgram model. Since fourgram models require larger resources to train, we did not consider such a step worthwhile for the other model components, because of their smaller weights. In addition, for the H4_AC and SWB components, there is insufficient training data to improve on the trigram models. (The interpolation weights were not changed from the values previously found for the all-trigram model.)

Model	PPL	WER
H4 (baseline)	174	33.8%
H4 + SWB	172	
H4 + NABN	166	
H4 + NABN + SWB	163	33.4%

Table 2: Results by using multiple data sources.

Model	PPL	WER
H4 + NABN + SWB, 3gram	163	33.4%
H4 + NABN + SWB, 4gram	154	33.1%

Table 3: Improvement due to fourgram modeling.

Table 3 shows that the fourgram LM gives us a small, but statistically significant improvement over the trigram LM.

Summarizing results so far, the fourgram LM incorporating multiple data sources achieves a 20% reduction in perplexity and a 1.8% relative reduction in word error rate on the development data, compared to a baseline trigram LM trained only on Hub4 data.

5. Condition-specific LM

The Hub4 data exhibit a variety of acoustic conditions. For the purposes of the evaluation, the data in the Hub4 benchmark were partitioned into seven different focus conditions [7]. These conditions are correlated with different speaking styles; for example, condition F0 is planned speech while F1 is spontaneous speech. Since speaking style affects language modeling, we can try to exploit this correlation using condition-specific LMs.

Unfortunately, of all the data sources mentioned in Section 3, only Hub4 acoustic training data are annotated with these focus conditions. Therefore, we separated Hub4 acoustic data into different subsets based on the focus conditions, trained separate trigram models for each of them, and interpolated each of the models with a general LM. The resulting model effectively gives extra weight to the data of one condition. During partitioned evaluation, we rescore each utterance using the LM matching the acoustic condition of the utterance.

The perplexity results for the Hub4 development data are summarized in Table 4. The first result uses a general LM that was trained only on Hub4 acoustic training data (H4_AC). The condition-specific LM gives a 5% perplexity reduction in this case.

However, when using the same approach on our best general fourgram LM, the perplexity reduction becomes marginal. This can be partly explained by the fact that the condition-specific training data are several orders of magnitude smaller than the general LM training data. This suggests that we would need an amount of condition-specific training data that is comparable to that of the general LM. However, since extensive hand labeling of more training data is not feasible, we would need automatic methods for sorting the existing unlabeled Hub4 training data by condition.

Model	General	By condition
H4_AC, 3gram	379	361
H4 + SWB + NABN, 4gram	154	151

Table 4: Perplexity results for condition-specific LM

6. Clustering Algorithms for Training Data

The Hub4 domain consists of speech of different topics and styles. Ideally, if we train topic- and style-specific LMs and correctly identify them during testing, we expect to improve the performance of our LM. Since the Hub4 LM data are a collection of unlabeled news articles, it is not possible to train topic- or style-specific LMs directly. Therefore, our approach here is to group the training data into subsets with coherent LM characteristics, which could subsume categories such as topic or style. To perform this grouping, we use an unsupervised hierarchical cluster algorithm and distance measure based on log likelihood. The text units clustered are articles, since we expect characteristics such as topic to be mostly constant within articles.

6.1. Agglomerative Clustering Algorithm

The algorithm forms clusters in a bottom-up manner, as follows:

1. Initially, put each article in its own cluster.
2. Among all current clusters, pick the two clusters with the smallest distance.
3. Replace these two clusters with a new cluster, formed by merging the two original ones.
4. Repeat the above two steps until there is only one remaining cluster in the pool.

Thus, the agglomerative clustering algorithm will result in a binary cluster tree with single article clusters as its leaf nodes and a root node containing all the articles.

In the clustering algorithm, we use a distance measure based on log likelihood. For articles A and B , the distance is defined as

$$d(A, B) = LL(A) + LL(B) - LL(A \cup B) \quad (1)$$

The log likelihood $LL(X)$ of an article or cluster X is given by a unigram model:

$$\begin{aligned} LL(X) &= \log \prod_{w \in X} p_X(w)^{c_X(w)} \\ &= \sum_{w \in X} c_X(w) \log c_X(w) - N_X \log N_X \end{aligned}$$

Here, $c_X(w)$ and $p_X(w)$ are the count and probability, respectively, of word w in cluster X , and N_X is the total number of words occurring in cluster X .

Notice that this definition is equivalent to the weighted information loss after merging two articles:

$$d'(A, B) = (N_A + N_B)H(A \cup B) - (N_A H(A) + N_B H(B)) \quad (2)$$

where

$$H(X) = - \sum_{w \in X} P_X(w) \log P_X(w) \quad .$$

To avoid expensive log likelihood recomputation after each cluster merging step, we define the distance between two clusters with multiple articles as the maximum pairwise distance of the articles from the two clusters:

$$d(C_1, C_2) = \max_{A \in C_1, B \in C_2} d(A, B) \quad (3)$$

where C_1 and C_2 are two clusters, and A, B are articles from C_1 and C_2 , respectively.

Once a cluster tree is created, we must decide where to slice the tree to obtain disjoint partitions for building cluster-specific LMs. This is equivalent to choosing the total number of clusters. There is a tradeoff involved in this choice. Clusters close to the leaves can maintain more specifics of the word distributions. However, clusters close to the root of the tree yield LMs with more reliable estimates, because of the larger amount of data.

We roughly optimized the number of clusters by evaluating the perplexity of the Hub4 development test set. We created sets of 1, 5, 10, 15, and 20 article clusters, by slicing the cluster tree at different points. A backoff trigram model was built for each cluster, and interpolated with a trigram model derived from all articles for smoothing, to compensate for the different amounts of training data per cluster. Then, the set of LMs that maximizes the log likelihood of the Hub4 development data was selected. Given a cluster model set $LM = \{LM_i\}$, the test set log likelihood was obtained as an approximation to the mixture-of-clusters model:

$$\begin{aligned} P(w | LM) &= \sum_i P(LM_i) * P(w | LM_i) \\ &\approx P(LM_{i^*}) * P(w | LM_{i^*}) \\ &\propto P(w | LM_{i^*}) \end{aligned}$$

where

$$i^* = \operatorname{argmax}_i P(LM_i | A) \quad ,$$

and $P(LM_i)$ and $P(LM_i | A)$ are the prior and posterior cluster probabilities, respectively.

In training, A is the reference transcript for one story from the Hub4 development data. During testing, A is the 1-best hypothesis for the story, as determined using the standard LM.

Note that $P(w | LM)$ depends on the smoothing weights used to compute $P(w | LM_i)$, which in turn determine which cluster a story is assigned to, which in turn determines the best smoothing weights. Therefore, we jointly optimize smoothing and cluster assignment in an iterative procedure. First, the posterior probabilities of the smoothed cluster LMs given reference transcripts for a story were calculated. Then, stories with the highest posterior probability of a *same* cluster LM were merged. The interpolation weight for the cluster LM and the general LM was tuned by maximizing the likelihood of the segments in the story cluster corresponding to the cluster LM. These steps were iterated until all cluster assignments became stable and the interpolation weights converged.

6.2. Practical Considerations

If the number of articles in a training data is N , the pairwise distance computation in the clustering algorithm will take $O(N^2)$ steps and $O(N^2)$ memory units. There are a total of 120,000 articles in the Hub4 LM training data, making this a challenging computational task. Applying the standard clustering algorithm would require roughly 60 GB of disk space to store the cluster tree and associated data, which was infeasible. To overcome this problem, we used a modified, two-stage agglomerative clustering algorithm.

In the first stage, 120,000 articles were divided into 20 sets of roughly equal sizes (about 6,000 articles for each set). We built a cluster tree for each set, resulting in 20 trees with 125,000 articles as their leaf nodes. For each of the 20 trees, 250 cluster nodes were then chosen for second-stage clustering. Therefore, for the 20 sets, there were a total of 5000 leaf clusters in the second stage. We used 250 clusters for each set so that the first stage did not enforce too many suboptimal clusters, since this stage did not consider data similarity across partitions. In the second stage, these 5000 clusters were further clustered into a super-cluster tree.

Using this approach, and choosing the optimal number of clusters in the second stage by the method described earlier, we obtained a set of 10 clusters. Manual inspection revealed that some are indeed strongly centered around topics, such as the O. J. Simpson trial, or stock market issues. Others do not seem to correspond to topics, but of course they might capture other distributional characteristics that are simply less obvious on inspection. Alternatively, clusters may represent collections of disparate topics that would reveal themselves on closer inspection, or if we had chosen a finer cluster granularity.

We also note that the clustering algorithm is suboptimal in several ways. One reason for suboptimality is the two-stage approximation described above. Even the single-stage algorithm may make suboptimal cluster choices because of its greedy nature. Finally, the distance measure used is only an approximation of an ideal measure. The likelihood is derived from unigram statistics, and a distance between higher-level clusters is approximated by the maximal pairwise distance among cluster members.

We were able to explore only some of the possible variations of the cluster model that suggest themselves. So far, these have not yielded significantly different results, as described in the Section 6.3. In particular, we experimented with k -means style clustering as a way to further optimize the cluster membership assignment once the number of clusters was chosen. In addition, the unigram likelihood distance was replaced with a corresponding measure based on bigrams.

In the k -means clustering, for each article in the Hub4 LM data, its perplexity was measured with respect to each of 10 cluster LMs previously computed using the greedy algorithm described earlier. The membership of articles in clusters was then recomputed by picking the lowest perplexity model for each article, and the cluster models were recomputed based on the new memberships. This process was iterated until two consecutive iterations resulted in identical memberships, the average log likelihood did not change, or disk space was exhausted.

6.3. Results and Discussion

Experiments were conducted on the Hub4 development data to compare the performance of the single, general LM to that of a cluster

Model	General LM		Cluster LM	
	PPL	WER	PPL	WER
H4.LM, trigram	195	34.0%	184	
H4 + SWB + NABN, trigram	156	33.4%	154	
H4 + SWB + NABN, 4gram	147	33.1%	146	33.0%

Table 5: Results of cluster LMs.

LM.

Results for the basic clustering algorithm are shown in Table 5. We compared perplexities of the cluster model to nonclustered models based on various training corpora. In all cases, only Hub4 LM training data were clustered, whereas the general LM component varied across experiments.

As expected, the largest relative improvements are obtained when the general LM is based on the same data (H4.LM) as the cluster models, yielding a 5% perplexity reduction. The improvements become successively smaller as more data are added to the general LM, similar to the results for the condition-specific LMs (see Section 5). We observed no significant word error rate reduction over the full fourgram LM. This suggests extending the clustering to the non-Hub4 data sources.

We also computed perplexity of the cluster LMs trained using the k -means approach. Results showed no significant improvement on perplexity and word error rate. On inspection, the new clusters did not seem to be more topic-coherent than the original ones. However, note that the k -means stage is based on the original 10 clusters. If one article does not belong to any cluster with a distinct topic, it will likely be assigned to a cluster with many topics. So it makes sense that k -means iterations will not improve clusters that are already incoherent.

One of the reasons our clustering approach has not yielded significant improvements so far may be cluster size. As described in Section 6.1, we varied the number of clusters between 1 and 20, partly because of computational constraints. This results in very large cluster sizes, not allowing the cluster models to be highly specialized, e.g., to a specific subtopic. Related approaches by other researchers have effectively used much smaller amounts of target-specific training data for LM adaptation purposes. For example, [6] reported small but significant improvements by using the 50 closest articles to the segment hypotheses within a story boundary. Similarly, [4] reported improvements by interpolating an in-domain LM with out-of-domain articles, which were weighted by a distance metric defining how similar they were to the target domain. However, since these approaches also used different distance measures, more careful examination is needed to assess the effect of cluster size on the performance of the adapted LM.

Another general issue in clustering is the choice of distance metric. In particular, it is not clear whether the distance metric should measure similarity of lower- or higher-order statistics. The main argument for using higher order statistics (as opposed to just unigrams) is that our eventual LM uses trigrams and fourgrams, so lower-order similarity of potential training material may not be as relevant to

the performance of the final model. For example, unigrams might not capture differences in style reflected in word collocations. In addition, isolated words are often ambiguous, and may be indicative of a topic only when used in conjunction with nearby words. For example, the word “drug” could be used to describe abuse by athletes for performance enhancement, or to refer to cases of recreational drug use by sports celebrities. To disambiguate such uses one would probably have to look at higher-order statistics of content words (and not just longer N-grams) [1]. However, the use of higher-order statistics could subject the similarity measure to too much noise, given that they are unreliable when collected on small samples.

Alternatively, the unigram distance metric can be made even less sensitive to stylistic and grammatical differences by omitting function and other high-frequency words from the similarity assessment. This is common practice in information retrieval (IR), and was also used for LM adaptation [6] and topic identification [2]. This approach focuses the similarity measure on semantic aspects and makes it less sensitive to syntactic features.

So far, we have experimented with only a few of the many choices on this continuum. Besides the plain unigram distance, we also tried using both bigram distance and a modified unigram distance that ignored a list of “stop-words” commonly used in information retrieval. Neither of these resulted in significant differences. Obviously, further investigation into the use of higher-order features for distance measures is needed.

7. Summary

In our language modeling experiments with the Hub4 task, we investigated the use of non-Hub4 data for increased LM robustness, as well as various forms of LM adaptation. The only significant improvements over a standard Hub4 trigram model were due to out-of-domain training data, namely from the NABN and Switchboard corpora. It was also advantageous to treat the acoustic training transcripts separately from the LM training texts, giving them disproportionate weight in the overall LM. As expected and found by others, a fourgram LM gives some improvement over the standard trigram model.

Adaption of the LM to the acoustic focus condition seems to work in principle, but suffers from the lack of a sufficient amount of labeled training data. Unsupervised clustering of the training data is another promising approach. We plan to continue developing this technique by optimizing some of its many parameters, such as cluster number and size, choice of distance metric, use of higher-order features and semantic versus syntactic similarity.

References

1. J. Cowie, L. Guthrie, and J. Guthrie. Lexical disambiguation using simulated annealing. In *Proceedings of 1992 DARPA Workshop on Speech and Natural Language*, 1992.
2. L. Gillick et al. Application of large vocabulary continuous speech recognition to topic and speaker identification using telephone speech. In *Proceedings of ICASSP-93*, 1993.
3. J. J. Godfrey, E. C. Holliman, and J. McDaniel. SWITCHBOARD: Telephone speech corpus for research and development. In *Proceedings ICASSP*, vol. I, pp. 517–520, San Francisco, 1992.
4. R. Iyer, M. Ostendorf, and H. Gish. Using out-of-domain data to improve in-domain language models. Technical Report ECE-97-001, ECE Department, Boston University, 1997.
5. S. M. Katz. 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), 1987.
6. S. Sekine, A. Borthwick, and R. Grishman. NYU language modeling experiment for 1996 CSR evaluation. In these proceedings.
7. R. Stern. Specification of the 1996 Broadcast News evaluation. In these proceedings.