FILTERING WEB TEXT TO MATCH TARGET GENRES

M. A. Marin, S. Feldman, M. Ostendorf and M. Gupta

University of Washington
Department of Electrical Engineering, Seattle, WA

ABSTRACT

In language modeling for speech recognition, both the amount of training data and the match to the target task impact the goodness of the model, with the trade-off usually favoring more data. For conversational speech, having some genre-matched text is particularly important, but also hard to obtain. This paper proposes a new approach for genre detection and compares different alternatives for filtering web text for genre to improve language models for use in automatic transcription of broadcast conversations (talk shows).

Index Terms— genre, web text filtering, language modeling

1. INTRODUCTION

Research on language modeling has repeatedly shown that larger training sets lead to better performance. Matching the training data to the task – in topic, time epoch, and genre – also benefits performance. Mismatched training and test conditions (e.g. newswire and conversational speech) can lead to an order of magnitude increase in perplexity over matched conditions, and adding such data can hurt performance if the data is not properly weighted [1]. While size cannot compensate for domain mismatch, a large amount of text that is only somewhat matched to the target is typically more useful than a small amount of well matched data. However, finding text that is somewhat matched to conversational speech tasks can be difficult, since written text sources include almost no instances of the sort of disfluencies and filled pauses that are frequently observed in spontaneous speech. Useful transcribed speech resources include the Switchboard and Fisher 2-party telephone conversation collections and the multi-party ICSI meeting corpus, but these are generally very different in topic from many interesting tasks, such as call center interactions, interactive lecture/discussions, and talk shows. Thus, one approach to handling this problem is to build mixture language models that include some components which are style matched and some which are topic matched (e.g. including lecture transcripts and conference proceedings for transcribing lectures [2]).

The web can be a valuable resource in these situations, because it contains a large amount of text from a variety of formal and informal genres. There is the potential to increase the amount of data that is reasonably well matched in both style and topic through filtering. Prior work has used “conversational queries” (high frequency n-grams in conversational speech) to retrieve text from the web, followed by perplexity filtering, as a method for finding data that is better matched to conversational speech [3, 4]. In this paper, we look at a new approach to filtering for genre.

To put our approach in perspective, we begin in Section 2 by summarizing related work on using the web for collecting data to train language models for speech recognition. Then, in Section 3, we discuss a method for modeling genre, which involves modeling part-of-speech (POS) histogram statistics and leverages data from several genres that represent different types of text and speech transcripts. We demonstrate that the model outperforms other standard techniques for genre classification. Sections 4 and 5 describe, respectively, the methods for using the genre model to filter web text and the language model training strategy given a new source of text. Finally, Section 6 summarizes the key findings on collecting text for improved speech recognition.

2. THE WEB AS A RESOURCE FOR LM TRAINING

Work in language modeling has used the web as either a text source or for estimating n-gram counts. Several groups have used online archives from targeted newspapers, television and radio stations. An information-retrieval (IR) approach is used in “just-in-time” language modeling [5], where adaptation data was obtained by submitting content words from initial hypotheses of user utterances as queries to a search engine. Another IR approach uses queries based on metadata (speaker name, topic, and description of the lecture contents) associated with recorded lectures to be transcribed [6]. The retrieved data is used for both vocabulary and language model adaptation. In [7], instead of downloading the actual web pages, the authors retrieved N-gram counts based on page counts provided by the search engine. More recently, web n-gram counts collected by Google for language modeling in machine translation are available from the Linguistics Data Consortium.[1]

In these approaches, the collection strategy generates text (or statistics) typical of a written style, hence not ideally suited for recognition of spontaneous speech. An alternative approach aims at collecting text that is more conversational in nature [3, 4]. The web is searched using queries that are combinations of frequent n-grams in the target data set, which for conversational speech often include the pronoun “I” and various types of fillers, as in:

“yeah I think” + “I mean you know” + “that kind of thing”

Then, the resulting data is filtered to remove junk pages using a threshold on the out-of-vocabulary rate, and further filtered using perplexity as computed from a language model trained on target data. The resulting data was used successfully to improve language modeling on different conversational speech tasks in both English and Mandarin. Other work proposes some modifications to the query generation and filtering stages [8, 9, 10]. In addition to benefiting standard n-gram language models, the web data is useful for more sophisticated parsing language models [11].

A limitation of the above approach is that it requires a non-trivial amount of text from the target domain to identify representative n-grams and to train the model for perplexity filtering. In this work, we look at an alternative method for filtering text for genre that requires less data from the target genre, by modeling more genres and using POS tags rather than words.

[1] http://www.ldc.upenn.edu/Catalog/docs/LDC2006T13/readme.txt

3. GENRE CLASSIFICATION ALGORITHM

The variation in word usage associated with topic dynamics in language is well known and often modeled. However, genre or register is also known to have a significant effect.[2] Biber [12] provides statistics showing differences in the frequency of use of different POS or syntactic structures for fiction vs. exposition, finding differences in passive vs. past tense forms of verbs and subordination vs. prepositional usage of function words such as *until, before,* and *as*, for example. Others have shown part-of-speech differences associated with different types of conversational speech, news text and email [1, 13]. Not surprisingly, filled pauses and pronouns are more frequent in spoken language than in written language; long noun phrases are more common in written language. Of course, the differences are much more complex than can be characterized by POS sequences, as evidenced by the relative lack of success in using a POS n-gram to "select" more conversational news broadcasts [1] and the usefulness of a variety of text features in genre detection [14, 15, 16]. However, working with POS tags as features has the advantage of a much lower dimensionality than would be needed when using words.

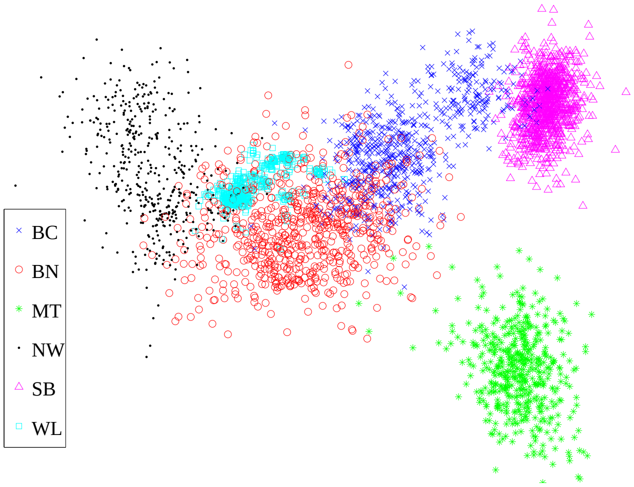
Our general approach to genre modeling, described in more detail in [17], involves the following:

1. **POS tagging**
   Tag the input word sequence, resulting in the l-length sequence p, based on a set of K tags.
2. **Feature extraction**
   - Estimate *Sliding Window POS Histograms* $h_j \in R^K$ using a w-length window $\{p_j, \ldots, p_{j+w-1}\}$.
   - Compute mean and variance *Histogram Statistics* of $\mathcal{H} = \{h_1, \ldots, h_{l-w+1}\}$.
   - *Normalize* the elements of the resulting $2K$-dimensional vector to have zero mean and unit variance.
   - Transform the normalized vector to a reduced dimension using *principal components (PC) analysis.*
3. **Classification or Scoring**
   Use quadratic discriminant analysis (QDA), i.e. Gaussian models with class-dependent full covariances, to compute a score for each class and maximize over all classes.

We start from text or speech transcripts with punctuation and case, as in [15]. Even though this information is discarded in training speech language models, retaining it leads to more reliable POS tags and higher accuracy genre detection (e.g. there are more questions in conversational speech than in newswire).

Parameters of the approach include the tag set, the sliding window width, and the final feature vector dimension. The POS tags are a modified version of the Penn Treebank set [18], with some collapsed noun and verb forms, four new different punctuation markers for periods, commas, colons, and quotes, and a few added words that tend to be indicative of conversational or informal speech (such as pronouns: "I", "you", and "we", adverbs: "so", "well", and other common words: "yeah", "ok", and "uh") for a total of K = 36 POS tags. The moving window width is $w = 5$. The final feature dimension is determined by discarding all the PC dimensions with variance below 1% of the maximum PC variance. The choice of QDA over linear (shared covariance) models or naive Bayes is due to the difference in distribution shapes observed in visualizations of the classes, as shown in Figure 1 which illustrates the PCA projection from one particular training/test split onto the top two ranked components.

Fig. 1. 2D PCA projection for the POS histogram features.



We conducted experiments using documents from six distinct genre classes: broadcast news (BN, 671 docs), broadcast conversations (BC, 698 docs), meetings (MT, 493 docs), news wire (NW, 471 docs), switchboard (SB, 890 docs), and weblogs (WL, 543 docs). The majority of the documents are 600 to 1000 words in length. We used the Ratnaparkhi maximum entropy tagger [19]. To estimate variance, the data set for each class is randomly split 75/25 into training/test sets, and the random split is repeated 50 times.

There are two baselines for comparison: one with POS trigram features as in [20] and the other with word-based features as in [16]; both use a naive Bayes model implemented with the Rainbow toolkit [21]. For the POS trigram and word frequency methods, we pruned all but the top 1000 and 10000 dimensions, respectively, maximizing information gain and choosing the thresholds empirically.

Classification results are in Table 1, averaged over the 50 training/test splits. The use of POS histogram statistics and QDA gives significantly better performance than both baselines.

	% correct	% std
QDA with POS histograms	98.45	0.44
naive Bayes with word unigrams	95.19	0.52
naive Bayes with POS trigrams	89.31	0.85

Table 1. Classification Performance.

The confusion matrix for QDA is shown in Table 2. The (ith,jth) entry indicates the percent of documents from class i that were classified as class j, on average. Note that the broadcast conversations are never confused with other conversational speech genres (meetings and Switchboard telephone conversations). For both baselines, there are many more types of errors, but most notably newswire is frequently labeled as the weblog class. For the POS naive Bayes model, broadcast news is also frequently labeled as weblog data. For word-based features, this result makes some sense, given that a lot of the weblog data is topically similar to the newswire data, though further pruning of words feature hurts performance.

[2] We use the term genre loosely here, differentiating between texts associated with spoken vs. written forms, formal vs. informal, pre-planned vs. spontaneous, etc., without enumerating these characteristics.

	Recognized class					
Ref	BC	BN	MT	NW	SB	WL
BC	97.4	2.6	0	0	0	0
BN	0.7	99.3	0	0	0	0
MT	0	0	100	0	0	0
NW	0	0.4	0	99.6	0	0
SB	0.1	0	0.3	0	99.5	0
WL	0	4.2	0	1.3	0	94.5

Table 2. Confusion matrix for QDA.

4. WEB TEXT FILTERING

We applied the genre detection method outlined in Section 3 to the problem of genre-specific filtering. First, data was collected from targeted web resources (specifically the Cable News Network (CNN) archives, containing both news reporting and talk shows), as well as using the frequent n-gram query method of [4]. The data was preprocessed to remove HTML tags and other formatting, and the documents were segmented so the average document length was about 600-1000 words. Approximately 5400 transcripts of news and talk shows were retrieved from the CNN archives (yielding 18k documents after segmenting), and approximately 1900 web documents (21k after segmenting). Initial inspection of the frequent n-gram web documents suggested that most originated from weblogs.

We performed genre classification using the QDA classifier as described in the previous section, and using all 6 classes for learning the PCA transform. Roughly 36% of the CNN documents are classified as BC, and roughly 50% have probability of being from the BC class as greater than 0.1. For the frequent n-gram web data, only 9% of the segmented documents are classified as BC, and 20% had a probablity of the BC class as greater than 0.1. Anecdotal inspection indicates that the data classified as BC indeed represent a more informal style, but the total amount was much lower than expected, given that the data collection approach was intended to target the conversational style. Surprisingly, only 9% of the segmented documents are labeled as weblogs (or 12% of full documents), which is not consistent with our initial observations. On the other hand, 68% are classified as BN, consistent with the overlap of the weblog and BN classes, as seen in Figure 1. A possible problem is that weblogs are highly variable, and our training examples reflected only a few sources. We also noted that some informal web data was incorrectly classified as newswire, which may be related to punctuation usage.

While we felt confident that the BC-filtered web data was a reasonable match to the target task, the resulting amount of data was small and therefore expected to have little impact in genre-adaptive language modeling. Therefore, we focused on filtering the CNN data. We used the 0.1 probability threshold for BC, resulting in a subset of about 7.8M words of text.

5. GENRE-ADAPTIVE LANGUAGE MODELING

We applied the filtering technique described in Section 4 to a recognition task for broadcast conversation data. The state-of-the-art SRI multipass broadcast news recognizer [22] was used as the baseline.

The baseline language model was generated from about twenty different components, including large, older corpora such as the North American Business News corpus (700M words), the Hub4 training set (150M words), TDT2 (21M words) and TDT4 (13M words). Newer corpora such as a collection of Business Week articles (5M words) and BBC transcripts (22M words) were also used. Additionally, a few corpora from GALE-related Broadcast News and Broadcast Conversations sources, both English source and English translations of Mandarin and Arabic, were used to provide data that was better matched in style as well as epoch to the evaluation data. The interpolated version of modified Kneser-Ney discounting [23] was used for all the component models in the baseline, as well as the components created for genre-specific adaptation. A heldout dataset of approximately 120K words, consisting of broadcast conversations transcribed with closed captions, was used to tune the vocabulary selection (following the procedure in [22]) and estimate LM mixture weights. The final vocabulary was 39K words, which is lower than that for broadcast news, consistent with other observations of lower effective vocabulary sizes in conversational speech.

The genre adaptation approach used here was mixture modeling. While experiments in [24] show that slightly better results can be obtained with unigram marginal constraints, we chose the simpler mixture models since this work focuses on data selection methods. Two mixture models were generated, a multiword bigram LM used in the initial decoding, and a regular 5-gram LM used for lattice expansion and rescoring. The mixture weights were optimized using an EM algorithm that minimized the perplexity of the heldout set. Each baseline mixture model was trained in two stages, first combining the various component models into two intermediate mixtures, then generating a single final model from the intermediate mixtures. The baseline language model was tuned for broadcast news transcription, and we retuned the mixture weights for BC for more direct comparison to subsequent experiments. Without adding new data, the BC tuning did not improve performance. The BC-tuned baseline first-pass LM has 18M bigrams, and the rescoring language model has 26M bigrams, 30M trigrams, 17M 4-grams and 19M 5-grams. There was only a small increase with the added web-collected genre data.

Two sets of experiments were performed using the targeted (CNN) web data. In the first set, summarized in table 3, we added different data sources to the final mixture models generated for the baseline and compared their impact on ASR performance. The corpora used were: the CNN web corpus, the Fisher corpus of telephone conversations, and the Google n-grams corpus. The CNN data, which is known to contain some BC data and is temporally close to the target task, gives a slight improvement both in first pass and second pass decoding. There is a slight gain (compared to the BC-tuned baseline) from adding the Fisher data in the first pass, but the gain is lost in the second pass. This behavior is consistent with the lack of confusion between BC and Switchboard in the genre classification experiments, but without these experiments the data may have been expected to be useful. The Google n-grams, representing unfiltered web text, provide no improvement in the first pass, consistent with our hypothesis that general web text is not useful for spontaneous speech genres. This is also reflected by the weights each corpus receives in the first pass mixture LM: both the Fisher and CNN components receive a non-trivial weight, and they give gains over the BC baseline. On the other hand, the Google data does not improve the perplexity of the tuning set, so it receives little weight. Given the very low weight received in the first-pass mixture, second pass decoding was not run for the Google data.

To analyze the impact of the BC genre match, we used a filtered subset consisting of only half of the documents and a same-size subset of randomly-selected documents, as shown in table 4. The accuracy obtained after filtering the CNN corpus is only marginally better than that obtained using a random subset. This is not entirely unexpected, given the large number of documents classified as BC-

	% Accuracy		First Pass
Method	First Pass	Final Pass	Weight
baseline, BN tuned	71.0	80.9	n/a
baseline, BC tuned	70.7	80.8	n/a
base + Full CNN	71.1	81.1	0.1275
base + Fisher	70.9	80.7	0.1419
base + Google n-grams	70.7	n/a	0.0011

Table 3. ASR accuracy when different types of LM training sources.

like. In fact, filtering yields slightly better first-pass results than the full corpus but, in the final pass, the larger amount of data in the full corpus trumps the better genre match of the filtered corpus.

	% Accuracy	
Method	First Pass	Final Pass
baseline, BC tuned	70.7	80.8
base + Full CNN	71.1	81.1
base + Random CNN Subset	71.1	80.9
base + Filtered CNN Subset	71.2	81.0

Table 4. ASR Accuracy, various mixture models.

6. DISCUSSION

In summary, this work confirms that unfiltered web data is not always useful for language model training, depending on the genre of the target task. Further, we find that intuitions about conversational speech from one genre being generally useful for other conversational genres do not hold. The new model of genre explored here is very successful for genre classification and prediction of genre differences that match the experimental ASR findings. Applying the model on web data collected with previously proposed methods shows that only a small percentage of the collection match the target genre. In the experiments here, the best results are obtained with a matched source web text collection. Further genre filtering is somewhat useful in the first pass recognition stage, but gains disappear with 5-gram rescoring, perhaps because the larger data sources are more important in those cases. However, we hypothesize that the genre filtering will be useful with a broader initial collection.

7. REFERENCES

[1] R. Iyer and M. Ostendorf, "Relevance weighting for combining multi-domain data for n-gram language modeling," *Computer Speech and Language*, **13**(3): 267–282, 1999.

[2] C. Fuegen et al., "The ISL RT-06S speech-to-text system," in *Machine Learning for Multimodal Interaction: Lecture Notes in Computer Science 4299*, S. Renals, S. Bengio, and J. Fiscus, Eds., pp. 407–418. Springer, Berlin/Heidelberg, 2006.

[3] I. Bulyko, M. Ostendorf, and A. Stolcke, "Getting more mileage from web text sources for conversational speech language modeling using class-dependent mixtures," in *Proc. HLT/NAACL*, 2003, pp. 7–9.

[4] I. Bulyko et al., "Web resources for language modeling in conversational speech recognition," *ACM Trans. on Speech and Language Processing*, **5**(1):1–25, 2007.

[5] A. Berger and R. Miller, "Just-in-time language modeling," in *Proc. ICASSP*, 1998, pp. II:705–708.

[6] C. E. Liu, K. Thambiratnam, and F. Seide, "Online vocabulary adaptation using limited adaptation data," in *Proc. Interspeech*, 2007, pp. 1821–1824.

[7] X. Zhu and R. Rosenfeld, "Improving trigram language modeling with the World Wide Web," in *Proc. ICASSP*, 2001, pp. I:533–536.

[8] A. Sethy, P. Georgiou, and S. Narayanan, "Building topic-specific language models from webdata using competitive models," in *Proc. Interspeech*, 2005, pp. 1293–1296.

[9] R. Sarikaya, A. Gravano, and Y. Gao, "Rapid language model development using external resources for new spoken dialog domains," in *Proc. ICASSP*, 2005, vol. I, pp. 573–576.

[10] T. Ng, M. Ostendorf, M.-Y. Hwang, M. Siu, I. Bulyko, and X. Lei, "Web-data augmented language models for Mandarin conversational speech recognition," in *Proc. ICASSP*, 2005, vol. I, pp. 89–593.

[11] W. Wang, A. Stolcke, and M. Harper, "The use of a linguistically motivated language model in conversational speech recognition," in *Proc. ICASSP*, 2004, vol. I, pp. 261–264.

[12] D. Biber, "Using register-diversified corpora for general language studies," *Computational Linguistics*, vol. 19, no. 2, pp. 219–242, 1993.

[13] S. Schwarm, I. Bulyko, and M. Ostendorf, "Adaptive language modeling with varied sources to cover new vocabulary items," *IEEE Trans. Speech and Audio*, **12**(3):334–342, 2004.

[14] Y.-B. Lee and S. H. Myaeng, "Text genre classification with genre-revealing and subject-revealing features," in *Proc. ACM SIGIR*, 2002, pp. 145–150.

[15] B. Kessler, G. Numberg, and H. Schütze, "Automatic detection of text genre," in *ACL-35*, 1997, pp. 32–38.

[16] E. Stamatatos, N. Fakotakis, and G. Kokkinakis, "Text genre detection using common word frequencies," in *COLING*, 2000, pp. 808–814.

[17] S. Feldman et al., "Part-of-speech histogram features for genre classification of text," in *Proc. ICASSP*, 2009.

[18] M. P. Marcus, M. A. Marcinkiewicz, and B. Santorini, "Building a large annotated corpus of English: the Penn Treebank," *Computational Linguistics*, vol. 19, no. 2, pp. 313–330, 1993.

[19] A. Ratnaparkhi, "A maximum entropy part-of-speech tagger," in *Proc. Empirical Methods in Natural Language Processing Conference*, 1996, pp. 133–141.

[20] M. Santini, "A shallow approach to syntactic feature extraction for genre classification," *CLUK 7: The UK special-interest group for computational linguistics*, 2004.

[21] A. K. McCallum, "BOW: A toolkit for statistical language modeling, text retrieval, classification and clustering," http://www.cs.cmu.edu/ mccallum/bow, 1996.

[22] A. Stolcke et al., "Recent innovations in speech-to-text transcription at SRI-ICSI-UW," *IEEE Trans. Audio, Speech and Language Processing*, **14**(4):1729–1744, 2006.

[23] S. Chen and J. Goodman, "An empirical study of smoothing techniques for language modeling," *Computer Speech and Language*, **13**(4):359–394, 1999.

[24] W. Wang and A. Stolcke, "Integrating MAP, marginals, and unsupervised language model adaptation," in *Proc. Interspeech*, 2007, pp. 618–621.