Supplementary Material for: Topic Identification and Discovery on Text and Speech

Chandler May, Francis Ferraro, Alan McCree, Jonathan Wintrode, Daniel Garcia-Romero, and **Benjamin Van Durme**
Human Language Technology Center of Excellence
Johns Hopkins University
cjmay@jhu.edu, ferraro@cs.jhu.edu, alan.mccree@jhu.edu, jcwintr@cs.jhu.edu, dgromero@jhu.edu, vandurme@cs.jhu.edu

1 Speech Systems

The input data to a topic model is a collection of documents, each of which is represented as a vector of word counts. Even in a noisy environment, relatively little work is required to generate such count vectors from a raw corpus. On the other hand, generating count vectors (or soft–count vectors) over content-bearing features from raw audio data requires considerable effort. In this paper, the features we use for speech data are triphone state cluster soft counts. Triphone cluster assignments are latent variables in an ASR system; however, we do not count the inferred values of those latent variables in the original ASR pipeline. Instead, for the sake of computational efficiency, a deep neural network (DNN) with a softmax top layer is trained to predict the triphone state cluster assignments, and we use aggregate triphone state cluster posteriors from this model as soft counts.

We now summarize the essential components of the ASR pipeline (Jurafsky and Martin, 2009).

The raw digital representation of speech is a sequence of samples of an acoustic wave. This sequence is commonly chunked into windows (of, e.g., 10 milliseconds) called *frames* and higher-order features are computed within each window. Commonly-used features are the twelve mel-frequency cepstral coefficients (MFCC) computed over the frame, the frame energy, and estimates of the first- and second-order across-frame derivatives of those thirteen features, for a total of 39 features per frame. These frame-wise feature vectors are used as the observations of states within *phone* models, discussed next.

Phonemes, or *phones*, are small units of sound generated from human speech. For example, the spoken word "yell" consists of three phones, [y], [eh], and [l] (Jurafsky and Martin, 2009). Across several instances of a word, each phone in that word may have varying duration. To accommodate this phenomenon, a phone can be modeled by a left-to-right hidden Markov model (HMM) over states, called *subphones*, that have self-loops. In this model, each state (subphone) corresponds to a fixed-duration frame and emits an acoustic feature vector under a Gaussian or Gaussian mixture model (GMM). Because of the self-loops on the states, a phone in this model can have arbitrarily long duration, as desired.

The phone model consisting of a single phone HMM is specified as a context-independent (CI) phone model, or *monophone* model. To account for an articulatory process called *coarticulation* in which the acoustic realization of phones are altered by their neighboring phones, one or more phone HMMs can be concatenated to the left and right of a monophone HMM, serving as context. Such context-dependent (CD) phone models often contain a single context phone on the left and a single context phone on the right of the central phone, and are accordingly called *triphone* models.

In the traditional ASR setup, after CI phone models (one model per phone) are learned using Gaussian observation models for the subphone states, each phone model is cloned to create one CD phone model for each possible triphone. The parameters for these models are then re-estimated (except the transition matrices are held fixed). In order to improve ASR performance, it is then desirable to increase the number of components in the GMM observation models (where the number of components is initially one). However, learning the raw triphone model is not feasible due to the high dimensionality of the parameter space and the sparsity of the data, so before the number of mixture components is increased, the triphone space is reduced by clustering.

A popular clustering approach constructs clusters in a manner that permits granular, subphone-level context dependencies while facilitating inference on unseen triphones. To fulfill both of these

objectives, a decision tree clustering procedure is applied to each state in the HMM, partitioning the set of triphone models for a given central phone into one or more clusters by clustering the states of the central phone. That is, for each central phone, for each state in the structure of the corresponding monophone model, a decision tree is learned to partition the levels of the state and the observation model parameter vectors of the levels within each cluster are tied together. (One level within each cluster is picked as an exemplar and the parameter vectors of the other levels in that cluster are tied to the exemplar's parameter vector.) Thus future re-estimation of a state's observation model parameters for a given triphone may change the parameters in other states in other triphones as well. Such state tying accommodates subphone-level dependencies while reducing the overall number of parameters.

For a given central phone and state, initially all levels are placed in the same cluster. The decision tree is grown by recursively splitting the set of levels models into two groups, where the possible splits are computed from a set of predetermined phonetic questions about the left and right context phones in the triphone HMM (such as, "is the left context phone nasal?"), and the question (split) chosen at a given node in the tree is one that maximizes the increase in log-likelihood of the training data while keeping the size of its child clusters above some lower bound. The splitting procedure stops when no leaf node can be split without violating the cluster size lower bound or when the maximum log-likelihood increase of any split falls below a pre-specified threshold.

Overall, ASR training proceeds by learning CI models using Gaussian observations, cloning those CI models to initialize a set of CD models, re-estimating parameters of the CD models, clustering the CD models, and iteratively increasing the expressibility of the observations by growing the number of GMM components and re-estimating parameters until some stopping criterion is met. Aligned training data is expensive to obtain and often noisy, so this training procedure is performed in an EM framework that also estimates word segmentations and phone alignments (mappings from time intervals in the acoustic data to words and phones within those words, respectively). This approach is called *embedded training*, as each word HMM is embedded in a whole-sentence HMM to estimate soft word segmentations. Embedded training is computationally intensive; after it is complete, to facilitate inference on held-out data, a DNN with a softmax top layer is trained to predict the triphone state cluster of a frame given the acoustic feature vectors of that frame and its context. Commonly, the input to the DNN is a "supervector" consisting of the concatenated acoustic feature vectors for a central frame, four left context frames, and four right context frames (nine frames in total).

2 ASR Training

The ASR system in this study is trained on Parts 1 and 2 of the Fisher English corpus, which comprise 11,699 telephone conversations spanning 1200 hours of audio (Cieri et al., 2004a; Cieri et al., 2005). The acoustic features are MFCC under three transformations: linear discriminant analysis, a maximum likelihood linear transform (MLLT), and continuous (feature-space) maximum likelihood linear regression (CMLLR). The latter transform is used for speaker adaptation. The ASR system is implemented in the KALDI speech recognition toolkit (Povey et al., 2011).

The triphone state cluster DNN is also implemented in KALDI. Its input is a "supervector" of nine frames of acoustic features (a central frame with four left context frames and four right context frames). The acoustic features are MFCC transformed by linear discriminant analysis and MLLT. The DNN comprises five layers with l_2-norm nonlinearities and 10:1 input-to-output ratios (Zhang et al., 2014). It is learned via mini-batch natural gradient descent, using mini-batches of size 512 and a "replicate-train-merge" to parallelize across four instances of the DNN (Povey et al., 2015).

3 Input Representations

We follow Hazen et al. (2007)'s efforts in topic ID on Fisher (Cieri et al., 2004c; Cieri et al., 2004b). We use a copy of Hazen's dataset containing 2060 labeled conversations, of which 1374 are used for training and 686 are reserved for held-out evaluation. However, this copy has two idiosyncrasies. First, our training set is smaller than the original by one conversation (Hazen et al., 2007). Second, we have three conversations in both training and test. We obtained this inexact copy of the data by personal correspondence with the creator, and we

were unable to obtain the exact instance or replicate it.

The data and labels each have their own biases. While the topic occurrences are unbalanced, ranging between 6 and 87 conversations per topic (as explained in the main paper), there is also an issue of *conversation drift*. Conversation drift occurs when a conversation starts off very relevant to a provided prompt or topic, but quickly diverges—or drifts—to topics unrelated to the prompt (in this case, the gold document label). Wintrode (2013) examined conversation drift in Fisher, finding a tremendous amount of classification signal in the first 25% percent of each conversation, and rapidly diminishing returns as more and more of the conversation is considered.

4 Representation Learner Implementation

We learn the mi-vector model by alternating maximum a posteriori inference of the mi-vectors $\boldsymbol{\theta}^{(d)}$ and maximum likelihood inference of the subspace basis $\boldsymbol{H}$, renormalizing $\boldsymbol{H}$ each time, as in McCree and Garcia-Romero (2015). The background vector $\mathbf{m}$ is augmented with a small backoff to the uniform distribution.

We use our own C++ implementation of mean-field variational inference for SAGE. We initially used Eisenstein et al. (2011)'s publicly released code[1] but found it too inefficient for our experiments. Our implementation, available online,[2] is orders of magnitude faster and more memory efficient. We use L-BFGS for MAP topic estimation and Newton-Raphson hyperparameter optimization. For models up-to size $K = 100$ (which we found to be the feasible upper-limit), we verified, both intrinsically and in our downstream classification task, that our implementation and the publicly available code were mutually competitive.

LDA is learned via Gibbs sampling, in which hyperparameters are optimized every 25 iterations, after a 200-iteration burn-in.

5 LSA Variations

For brevity and visual clarity, in the main paper we only report one variation of LSA on each dataset. Specifically, we weight the word counts by tf-idf, mean-center that data matrix, and compute the lower-dimensional representation by SVD. Similarly, we normalize the triphone state cluster soft counts using the l_2 norm, mean-center that data matrix, and compute the lower-dimensional representation by SVD. These implementations were chosen by lowest overall cross-validation–estimated topic ID error and test-set V-measure. In this section of the supplement we report performance of other implementations we tried. We name these implementations according to their preprocessing recipes, so (for example) in ctr-l2-SVD we mean-center the data matrix, l_2-normalize the centered matrix, and compute the lower-dimensional representation via SVD.

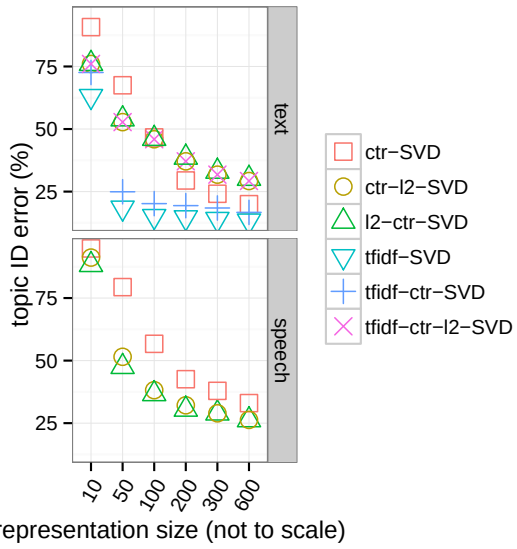

Figure 1: Topic ID error (%) on the test set for variations of LSA at dimensions $K \in \{10, 50, 100, 200, 300, 600\}$.

Topic ID error computed on the test set is plotted with respect to representation dimension K in Figure 1. This is the full-supervision setting.

We report cross-validation estimates of the topic ID error (over the training set) for the full-supervision and limited-supervision settings in Figure 2 for $K = 10$, Figure 3 for $K = 100$, and Figure 4 for $K = 600$.

Finally, we report V-measure with respect to representation dimension in Figure 5.

[1] https://github.com/jacobeisenstein/SAGE

[2] https://github.com/fmof/sagepp

References

Christopher Cieri, David Graff, Owen Kimball, Dave Miller, and Kevin Walker. 2004a. Fisher english training speech part 1 speech LDC2004S13. DVD.

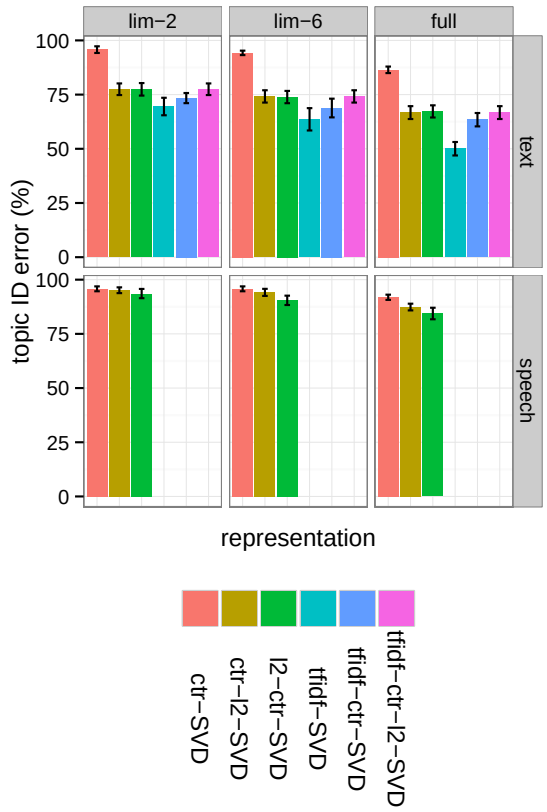

Figure 2: CV topic ID error (%) for variations of LSA of size $K = 10$.

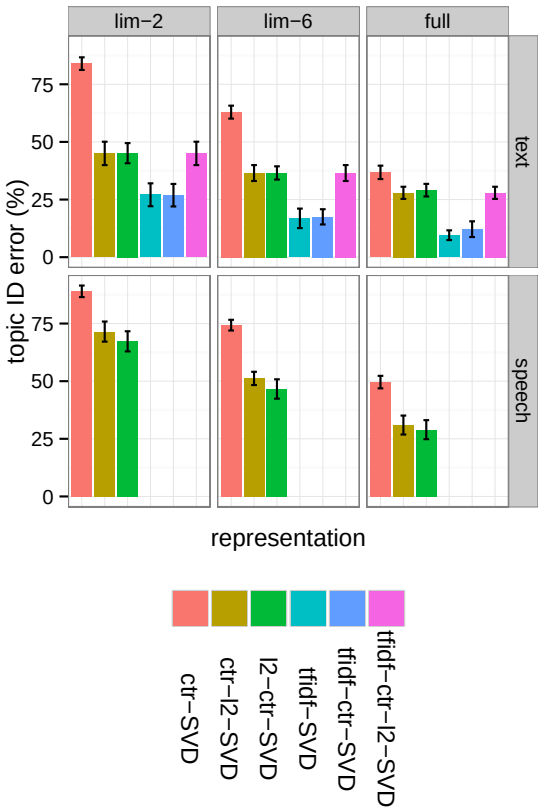

Figure 3: CV topic ID error (%) for variations of LSA of size $K = 100$.

Christopher Cieri, David Graff, Owen Kimball, Dave Miller, and Kevin Walker. 2004b. Fisher english training speech part 1 transcripts LDC2004T19. Web Download.

Christopher Cieri, David Miller, and Kevin Walker. 2004c. The fisher corpus: a resource for the next generations of speech-to-text. In *International Conference on Language Resources and Evaluation (LREC)*, pages 69–71.

Christopher Cieri, David Graff, Owen Kimball, Dave Miller, and Kevin Walker. 2005. Fisher english training part 2, speech LDC2005S13. DVD.

Jacob Eisenstein, Amr Ahmed, and Eric P. Xing. 2011. Sparse additive generative models of text. In *Proceedings of the 28th International Conference on Machine Learning (ICML)*, pages 1041–1048.

Timothy J. Hazen, Fred Richardson, and Anna Margolis. 2007. Topic identification from audio recordings using word and phone recognition lattices. In *IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU)*, pages 659–664.

Daniel Jurafsky and James H. Martin. 2009. *Speech and Language Processing*. Pearson Education, Inc., 2nd edition.

Alan McCree and Daniel Garcia-Romero. 2015. DNN senone MAP multinomial i-vectors for phonotactic language recognition. In *Interspeech*. To appear.

Daniel Povey, Arnab Ghoshal, Gilles Boulianne, Lukáš Burget, Ondřej Glembek, Nagendra Goel, Mirko Hannemann, Petr Motlíček, Yanmin Qian, Petr Schwarz, Jan Silovský, Georg Stemmer, and Karel Veselý. 2011. The kaldi speech recognition toolkit. In *IEEE 2011 Workshop on Automatic Speech Recognition and Understanding*. IEEE Signal Processing Society.

Daniel Povey, Xiaohui Zhang, and Sanjeev Khudanpur. 2015. Parallel training of deep neural networks with natural gradient and parameter averaging. In *International Conference on Learning Representations (ICLR) Workshop*.

Jonathan Wintrode. 2013. Leveraging locality for topic identification of conversational speech. In *Interspeech*, pages 1579–1583.

Xiaohui Zhang, Jan Trmal, Daniel Povey, and Sanjeev Khudanpur. 2014. Improving deep neural network acoustic models using generalized maxout networks. In *International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pages 215–219.

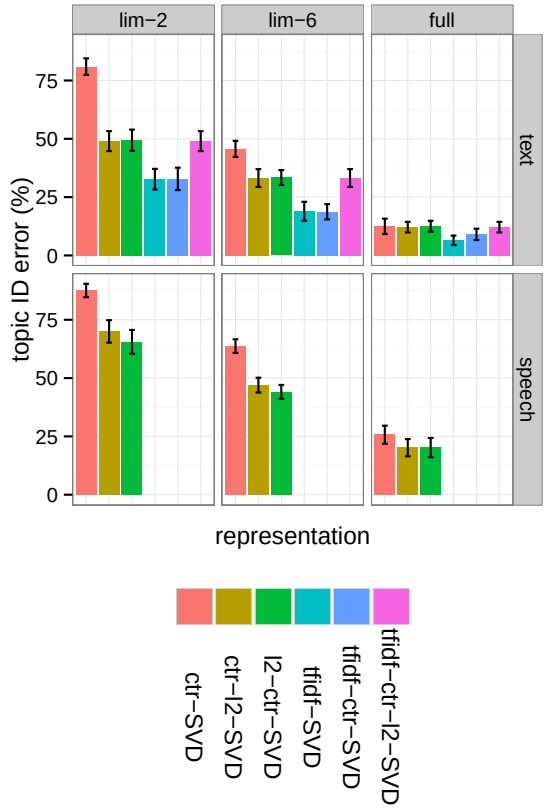

Figure 4: CV topic ID error (%) for variations of LSA of size $K = 600$.

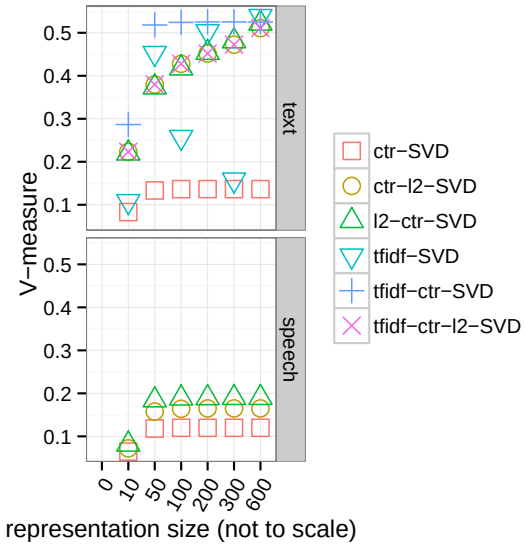

Figure 5: V-measure on the Fisher English text and speech data, respectively, for variations of LSA at selected dimensions.