INTERSPEECH 2009 BRIGHTON

Incremental Dialog Clustering For Speech-to-Speech Translation

David Stallard, Stavros Tsakalidis, Shirin Saleem

BBN Technologies, Cambridge, MA, USA

stallard@bbn.com, stavros@bbn.com, ssaleem@bbn.com

Abstract

Application domains for speech-to-speech translation and dialog systems often contain sub-domains and/or task-types for which different outputs are appropriate for a given input. It would be useful to be able to automatically find such sub-domain structure in training corpora, and to classify new interactions with the system into one of these sub-domains. To this end, We present a document-clustering approach to such sub-domain classification, which uses a recently-developed algorithm based on von Mises Fisher distributions. We give preliminary perplexity reduction and MT performance results for a speech-to-speech translation system using this model.

Index Terms: speech-to-speech translation, dialog clustering, language model adaptation.

10.21437/Interspeech.2009-155

1. Introduction

Application domains for language processing systems often contain separate sub-domains and/or task-types, whose distributions of words, phrases, and word-meanings may differ from one another. The sub-domain or task currently being undertaken often influences which output is correct for a given input. For example, the word "magazine" is likely to have quite a different meaning in a dialog about business, than it would have in a dialog about weapons and ammunition. Sub-domains are relevant to a wide variety of language-processing systems, including systems for automatic speech recognition (ASR), machine translation (MT), language understanding, spoken dialog, and speech-to-speech translation (S2S). Clearly, it would be desirable for such systems to know which type of task or sub-domain they are currently undertaking, so that they can better adapt their internal models to produce the correct output. Knowing the task-type may be useful for other purposes as well, such as actively assisting the user in overcoming problems, etc.

Our approach to the sub-domain classification and recognition problem is carry out unsupervised clustering of existing dialogs, in order to find internal sub-domain structure, estimate sub-domain recognition models, and to determine appropriate sub-corpora for estimating sub-domain-specific language and translation models. We treat the dialog as a dynamically evolving "document" that is updated with each new utterance, to which various Information Retrieval (IR) and document-clustering techniques can be applied. In particular, we apply a leading, recently developed document-clustering algorithm, the Mixture of von Mises Fısher distributions algorithm or moVMF [1]. The movMF has the advantage of producing well-defined probabilities for cluster membership, and for allowing documents to reside in multiple clusters where appropriate.

Our test bed application for this work is a speech-to-speech translation system, reported on in [2]. S2S is an especially "target-rich" environment for context modeling, given the number of component models, which include a separate ASR Language Model (LM), MT phrase translation model, and MT target LM, for both translation directions.

Of course, the idea of using clustering to improve modeling performance, is not new [6]. More recently, there has been work in applying clustering at the word level, using techniques based on Latent Dirichlet Allocation [4]. We believe there is still value in document-level clustering, however, particularly since the sub-domain/task-type is itself a document-level attribute. To our knowledge, the movMF algorithm has not previously been applied to the dialog clustering problem.

In the remainder of the paper, we first review the movMF algorithm, and describe our method for using it. Finally, we give very preliminary experimental results for this work in progress, including results for perplexity reduction and MT improvement.

2. Clustering Algorithms

In this work, the document/dialog is dynamic – that is, it is updated with each new utterance. The final number of words in this document cannot be known until the dialog is complete, and therefore the length of the dialog as it has been observed so far may not always give us good information as to what cluster it belongs to. This argues for a common practice in IR; namely, normalizing the document by its Euclidean length, so that is mapped onto the surface of a (*d-1*)-dimensional unit hypersphere, where *d* is the number of words in the vocabulary.

For the document clustering problem all of "the action" is therefore constrained to take place on the surface of the hypersphere; no points can lie inside or outside the sphere. Model-based clustering techniques that are based on Euclidean distance metrics, such as K-means and multivariate Gaussian mixture models, are likely to be a poor fit for this task, as these algorithms cluster points throughout the d-dimensional hyperspherical volume, and take no cognizance of the hypersurface constraint. They are thus liable to allocate their cluster means, and most of their probability mass, to the forbidden interior and exterior of the hypersphere. Indeed, a recent comprehensive study has shown Euclidean measures to perform very poorly for document clustering [5].

One algorithm recently developed to address this problem is Spherical K-Means (spkmeans) [6]. Spherical K-means is like the standard K-means algorithm, except that it deals with unit-normalized vectors, and uses the cosine as a similarity metric instead of Euclidean distance. It thus carries out clustering explicitly on the hyperspherical surface. Spherical K-means has been shown to perform well for document clustering tasks [6][3]. It is, however, a hard-clustering algorithm in which a given point either resides in a cluster or not, without any notion of graded membership.

More recently, a new document-clustering algorithm, the Mixture of von Mises Fisher distributions algorithm (movMF), has been developed, which is a generative probabilistic mixture model that does allow for graded

membership. The movMF is like the Gaussian mixture model, except that it uses the von Mises Fısher distribution [4] in place of the Gaussian. The von Mises Fisher distribution (vMF) is the analog of the Gaussian distribution for so-called directional statistics, in which only the directions of the vectors matter, and not their magnitudes. It defines a probability density over the unit hypersphere as a function of the cosine of the angle between the input vector and the distribution's mean vector. The density for the complete vMF mixture is given by:

$$P(x \mid \kappa,\mu,\alpha) = \sum_h \alpha_h c_d(\kappa_h) e^{\kappa_h \mu_h^T x}, \quad (1)$$

where α_h is the mixture weight for the h'th vMF distribution, μ_h is the normalized mean vector of this distribution, and κ_h is a concentration parameter governing how tightly the cluster members are distributed about the mean. The coefficients $c_d(\kappa_h)$ are per-distribution normalization terms. As with Gaussian mixture modeling, given a corpus, Expectation Maximization (EM) can be used to estimate these parameters. The following is the complete set of update equations

E-Step:

$$P(h \mid x_i,\kappa,\mu) \leftarrow \frac{\alpha_h f(x_i \mid \kappa_h,\mu_h)}{\sum_j \alpha_j f(x_i \mid \kappa_j,\mu_j)} \quad (2)$$

M-step:

$$\alpha_h \leftarrow \frac{1}{n}\sum_{i=1}^{n} P(h \mid x_i,\kappa,\mu) \quad (3)$$

$$\mu_h \leftarrow \sum_{i=1}^{n} x_i P(h \mid x_i,\kappa,\mu) \quad (4)$$

$$r_h \leftarrow \left\|\mu_h\right\| / n\alpha_h \quad (5)$$

$$\mu_h \leftarrow \mu_h / \left\|\mu_h\right\| \quad (6)$$

$$\kappa_h \leftarrow \frac{r_h d - r_h^3}{1 - r_h^2} \quad (7)$$

3. Estimating the Cluster Model

To carry out our experiments, we used the DARPA TRANSTAC English/Iraqi corpus. This corpus consists of recorded 2-way spoken dialogs between an English speaker playing the role of the soldier, who is known as the Subject Matter Expert (SME) and an Iraqi Arabic speaker, who is known as the Foreign Language Expert (FLE), with a bilingual human interpreter. Dialogs take place on a variety of subjects, including checkpoints, job interviews, house searches, medical conversations, and the like. The English and Iraqi speech is transcribed, and the transcriptions translated into Iraqi and English, respectively.

For purposes of this work, we chose to do clustering only in English, both because of our greater familiarity with that language, and because the increased vocabulary of Arabic due to affixation would likely make it more difficult to cluster. We used both the English of the SME, and the translated English of the FLE, without distinguishing between the two. The resulting dialog clusters can also be used for Arabic language modeling, however, by the reverse procedure: i.e., using the Arabic of the FLE and the translated Arabic of the SME. For the training subset of the corpus, we used 2,748 dialogs, comprising approximately 2.5M word tokens, with 17K unique words.

To estimate the parameters of the movMF distribution, we process the training corpus into a term-document matrix (TDM) using the following procedure. First, the corpus is preprocessed by contraction expansion, Porter stemming, and stop-word removal. Next, each dialog is transformed into a single word-count vector, without distinguishing between utterances or between SME or FLE roles. Following [6], we remove all words that occurred in less than 0.2% of the documents, or in more than 15% of the documents. We next remove all documents whose remaining total word count was less than 50, so as to avoid short dialogs that would make less reliable candidates for clustering. Finally, words are weighted using the TF.IDF weighting scheme, and the resulting word vectors normalized by their Euclidean length. The resulting TDM comprises 2,316 document vectors, with 3,300 unique words.

The document vectors in this TDM were used to estimate the movMF parameters with EM. We used the movMF 2.0 C++ implementation developed by Sra [9], running on a Linux workstation. An initial κ value of 10 and maximum allowed κ value of 1000 were used. Our primary experiments were done with 30 clusters.

To measure cluster quality, we used two primary metrics. The first, cluster coherency, or H_{avg}, measures how closely the members (or probability-weighted partial members) of a cluster agree with the cluster's mean. It is defined by the following equation, in which D denotes the complete set of documents:

$$H_{avg} = \sum_h \sum_{x \in D} P(h \mid x) \mu_h^T x \quad (8)$$

It is desirable that this value be as high as possible.

The second measure, inter-cluster similarity, or S_{avg}, measures how well non-members of a cluster agree with its mean. It is desirable that this value be as low as possible. In this computation, $P(h|x)$ is replaced by its hardened 0 vs. 1 version, so that a given data point is either in a cluster or out of it. S_{avg} is defined by:

$$S_{avg} = \frac{1}{n(k-1)} \sum_{h=1}^{k} \sum_{x \notin D_h} \mu_h^T x \quad (9)$$

where n denotes the number of documents, k the number of clusters, and D_h denotes the set of documents which are in cluster h. To combine these metrics, we take their ratio, H_{avg}/S_{avg}. Table 1 shows the values obtained for k=30 clusters.

Docs	Words	K	H_{avg}	S_{avg}	Ratio
2316	3301	30	0.46	0.06	7.67

Table 1: Cluster Quality Metrics

We also show below some of the clusters obtained by the above procedure, as represented by the top-ranked words of their cluster means. As can be seen, these clusters correspond to specific dialog types, such as construction, medicine, job recruitment, and municipal services.

1. *construct compani team site design project architectur blueprint civil weld architect plan*
2. *pain patient bleed symptom pressur blood emerg poison glove diabet chest bone stomach wound*
3. *salari thousand unemploi graduat depart health tomorrow applic test serv dinar prefer*
4. *trash garbag citi mayor pick truck pipe power council collect pile remov sheikh bag river*

To estimate cluster-specific language models, we used the original running text of the dialogs, as divided into utterances. Each dialog *x* contributes its n-gram counts to the LM for each cluster *h*, with the counts being weighted by the posterior probability *P(h|x)*, i.e. the dialog's degree of membership in the cluster. The Kneser-Ney method was used for smoothing.

4. Applying the Cluster Model

In an actual online system, the cluster posterior probabilities would be updated after each successive utterance, and this updated distribution used as context when interpreting the next utterance. Our offline experiments follow this strictly causal process. In particular, at each utterance, the system is not allowed to look at the succeeding future utterances of the dialog, which are supposed to be unknown, in order to make decisions about the current utterance. This means that the system cannot be allowed to use the complete word-vector for the test dialog, as this is obtained by summing over all utterances of the dialog. Moreover, even the word counts of the current utterance itself cannot be included, as these words are the very thing the system is trying to decide upon. Instead, at each utterance, the system can only use a dialog vector that was computed from the utterances preceding the current one.

The dialog vector therefore evolves incrementally, beginning with little or no knowledge of the dialog's content, and successively acquiring more information as the dialog proceeds. Each successive state of the dialog vector thus becomes a "document" of its own, a document that is a prefix of the complete one. To generate these document vectors, the words of the partial document are preprocessed in the same way as were the words of the training corpus (stemming, stop-word filtering, etc). The resulting word-count vector is then weighted using the word weights that were calculated when producing the training TDM. Words whose weight is not specified in this set, i.e. those words which were not present in the preprocessed training TDM, are discarded.

Note that the above procedure will produce some dialog vectors that have a length of zero. In fact, on average the first 3 utterances of test-set dialogs produce 0-vectors, since their preceding utterances consist exclusively of highly common words (e.g. greetings), which were excluded from the training TDM. By definition, such 0-vectors carry no information about the dialog, and cannot be assigned to any cluster. However, once the first non-zero vector is produced, all subsequent vectors in that dialog are guaranteed to be non-zero.

At test time, we consider each utterance of the dialog in turn. If the current utterance corresponds to the 0-vector, the system defaults to the background LM alone. Otherwise, the posterior cluster probabilities *P(h|x)* are computed for the utterance, using equation (10) below, which is simply a restatement of the E-step in equation (2):

$$P(h \mid x, \Theta) = \frac{\alpha_h f(x \mid \kappa_h, \mu_h)}{\sum_l \alpha_l f(x \mid \kappa_l, \mu_l)} \qquad (10)$$

These posterior probabilities are then used as the weights for the corresponding sub-LMs for the clusters. To produce the complete LM, the weighted ensemble of cluster LMs is further interpolated with a background LM that is estimated from the entire training corpus, using a weight 0.4/0.6 for the clusters vs. the background LM.

5. Results and Discussion

To measure the effect of on the cluster models on performance, we used the 243 dialogs of the held-out test set, which contained a total of 20,769 utterances. We first measured perplexity on the SME (English) portion of the held-out set, comparing the baseline general model, trained on the combined English for both SME and FLE roles, with the interpolated cluster LM model, trained on the same set but using the dialog cluster information. Only the fair, causal update procedure outlined above was used for this test. Results are shown in Table 2.

Baseline	Cluster LM	Reduction
58.5	42.5	27.3%

Table 2: English Perplexity Results

We also measured the perplexity for the FLE (Iraqi) portion of the held-out set, as shown in Table 3. Here, we measure perplexity for 1) the baseline (non-clustered) model, 2) a static cluster model with sub-LM's weighted by their prior mixture weights, 3) a dynamic cluster model in which sub-LM's weighted by the posterior probabilities of the various clusters, and 4) a "cheating" version of the dynamic cluster model, in which the weights used are those of the final utterance of the given dialog. We give these numbers for clusters generated by both the movMF and spkmeans algorithms.

	movMF	spkmeans
Baseline	279	279
Prior	275	275
Posterior	224	223
Cheating	220	219

Table 3: Iraqi Perplexity Results

As can be seen, the reduction in perplexity is substantial, and is approximately equal for both movMF and spkmeans. The "cheating" version of posterior clustering, i.e. looking into the future at the final-state dialog vector, does help reduce perplexity, but only very slightly.

In Table 4, we show the corresponding effects on Iraqi Word Error Rate (WER) for each of these conditions. Again we see that "cheating" does not actually help. A small improvement in WER is seen for the posterior-weighted cluster LM, with spkmeans being slightly better than moVMF.

	movMF	spkmeans
Baseline	29.2	29.2
Prior	29.0	29.1
Posterior	28.9	28.8
Cheating	28.9	28.8

Table 4: Iraqi WER

To assess the effect on bottom-line performance, we compared Arabic-to-English MT performance comparing the baseline general model as the target LM, vs. the dialog clustered model as target LM. Results for BLEU and 100-TER are shown in Table 5.

Target LM		BLEU	100-TER
Baseline		34.6	50.2
movMF	Prior	34.9	50.7
	Posterior	35.2	50.8
	Cheating	35.2	50.8
spkmeans	Prior	34.9	50.7
	Posterior	35.2	50.8
	Cheating	35.3	50.8

Table 5: Effect of Clustering on MT Performance

A modest gain is seen for use of the cluster LM on BLEU and TER. Once again, the movMF and spkmeans algorithms are very close in performance, indicating that the soft-assignment property of the movMF is not, as of yet, providing any performance benefit.

Inspection of the movMF output shows that in fact the algorithm assigns very sharp posteriors to both training and test vectors in our corpus. In particular, the posteriors that the movMF EM algorithm computes for training vectors are exclusively 0 vs. 1, meaning training vectors are hard-assigned to clusters in the final result. More relevantly for our purposes, many of the posteriors computed by inference on the test vectors are 0 vs. 1 as well. Further inspection of movMF output shows that the final κ-values (concentrations) computed by EM are very high, another sign of de facto hard clustering. This is in line with observations in [1], but for our purposes may indicate overfitting. Initial experiments with reducing the dimensionality of the vocabulary show less tendency towards such hard clustering, and this is a path for further investigation.

Throughout these experiments, we also find that "cheating" in the form of looking ahead to the vector formed from the final state of the dialog, helps only slightly if it all. One question is how much the average dialog vector differs from the final vector for the dialog. To determine this, we computed for each non-zero vector a "classification vector", defined as a linear combination of the corresponding cluster means, weighted by $P(h|x)$. We computed the average cosine between the dialog's final classification vector and the current classification vector. The value obtained was 0.88, indicating substantial average agreement. Of course, as the dialog progresses, the succeeding dialog vectors converge to the final one, and the average agreement is surely skewed upwards by this fact. An interesting question, which we have not yet explored, is how much the difference between early and later dialog vectors affects the performance of the clustering model.

6. Conclusions

We have presented an incremental dialog clustering model which applies the Spherical K-Means and Mixture of Von Mises Fisher Distributions algorithms to cluster training-set dialogs on the unit hypersphere. The dialog clusters are used to estimate cluster-specific language models, which are used for speech recognition and machine translation in a speech-to-speech translation system. Substantial perplexity reductions were achieved, as well as modest end-performance gains in terms of WER and BLEU. So far, little benefit is seen from the soft-clustering properties of movMF over spkmeans, but this may be because of the highly concentrated clusters that movMF actually produces in its final result. It is possible that an alternative parameter estimation method to EM, such as Gibbs sampling, would improve this situation. Avenues for further investigation include other clustering methods, such as co-clustering of words and dialogs, dimensionality reduction prior to clustering, and applying the dialog clustering to the SMT translation model as well.

7. Acknowledgements

We wish to thank Suvrit Sra for making his movMF implementation publically available, and for email discussions. We also thank Ivan Bulyko for help with his LM toolkit

8. References

[1] A. Banerjee, I. Dhillon, J. Ghosh, and S. Sra, "Clustering on the Unit Hypersphere using von Mises-Fisher Distributions," *The Journal of Machine Learning Research*, Vol. 6, pp. 1345-1382, 2005.

[2] D. Stallard, C. Kao, K. Krstovski, D. Liu, P. Natarajan, R. Prasad, S. Saleem, K. Subramanian. "Recent Improvements and Performance Analysis of ASR and MT in a Speech-to-Speech Translation System," in *Proceedings of ICASSP,* 2008.

[3] R. Iyer and M. Ostendorf, "Modeling long distance dependencies in language: Topic mixtures versus dynamic cache model," *IEEE Transactions on Speech and Audio Processing*, 7, January 1999.

[4] D. Blei, A. Ng, and M. Jordan, "Latent Dirichlet Allocation," Journal of Machine Learning Research, 3:pp. 993 – 1022, 2003.

[5] A. Strehl, J. Ghosh, and R. Mooney, "Impact of Similarity Measures on Web-Page Clustering," in *AAAI Workshop on AI for Web Search*, 58-64.

[6] I. Dhillon and D. Mohda, "Concept Decompositions for Large Sparse Text Data Using Clustering," *Machine Learning*, 42(1), pp. 143-175.

[7] S. Zhong and J. Ghosh, "Generative Model-Based Document Clustering: A Comparative Study," *Knowledge and Information Systems*, Vol. 8, Issue 3. pp. 377-384.

[8] R. Fisher, "Dispersion on a Sphere," *Proceedings Royal Society London* Ser. A., 217 pp 295-305, 1954.

[9] S. Sra, "Clustering Data Using Mixtures of Von Mises Fisher Distributions," November 2007. [Online]. Available: http://www.kyb.tuebingen.mpg.de/bs/people/suvrit/work/progs/movmf/