Procesamiento del Lenguaje Natural, Revista nº 45, septiembre 2010, pp 175-182 recibido 30-04-10 revisado 18-05-10 aceptado 20-05-10

Clustering of syntactic and discursive information for the dynamic adaptation of Language Models*

Agrupamiento de elementos sintácticos y discursivos para la adaptación dinámica de modelos de lenguaje

J.M. Lucas, F. Fernández, V. López, J. Ferreiros, R. San Segundo
Grupo de Tecnología del Habla, Universidad Politécnica de Madrid
Avenida Complutense s/n 28040. Madrid
{juanmak, ffm, veronicalopez, jfl, lapiz}@die.upm.es

Resumen: Presentamos una estrategia de agrupamiento de elementos de diálogo, de tipo semántico y discursivo. Empleando Latent Semantic Analysis (LSA) agrupamos los diferentes elementos de acuerdo a un criterio de distancia basado en correlación. Tras seleccionar un conjunto de grupos que forman una partición del espacio semántico o discursivo considerado, entrenamos unos modelos de lenguaje estocásticos (LM) asociados a cada modelo. Dichos modelos se emplearán en la adaptación dinámica del modelo de lenguaje empleado por el reconocedor de habla incluido en un sistema de diálogo. Mediante el empleo de información de diálogo (las probabilidades a posteriori que el gestor de diálogo asigna a cada elemento de diálogo en cada turno), estimamos los pesos de interpolación correspondientes a cada LM. Los experimentos iniciales muestran una reducción de la tasa de error de palabra al emplear la información obtenida a partir de una frase para reestimar la misma frase.
Palabras clave: Adaptación de modelos de lenguaje, Reconocimiento automático de habla, Sistema de diálogo.

Abstract: In this paper we present an approach for clustering dialogue items, both semantic and discursive. We use Latent Semantic Analysis (LSA) to cluster the different dialogue items according to a correlation-based distance. After building a set of groups that make up a partition of the semantic or discursive space, we train a stochastic Language Model (LM) for each group. We use these LM to dynamically adapt the language model used by a speech recognition module included in a Spoken Dialogue System. We use dialogue-based information (namely, the posterior probabilities of the dialogue items that our Dialogue Manager estimates on each dialogue turn) to automatically estimate the interpolation weights among LM. The initial evaluation shows a reduction of the word error rate when using the information of an utterance to rescore the same utterance.
Keywords: Language Model Adaptation, Automatic Speech Recognition, Spoken Dialogue System.

1 Introduction

Statistical language model adaptation has become a current practise in Speech Technology research. Its main goal relies on modifying the language model (LM) used by an automatic speech recognition system (ASR), in order to improve its performance, in terms of speech recognition rate. For instance, we could adapt a general-purpose LM (trained with a high amount of data, usually related to different scopes) to a closed domain (using few data closely related to that domain), pursuing an improvement of the response of a domain-dependent spoken dialogue system (SDS) in which the ASR is included.

There are several approaches to adapt stochastic language models, depending on how the interpolation models can be estimated, the interpolation strategy itself, and the point of the recognition stage the interpolated models are used (Bellegarda, 2004).

The most common language adaptation

* Este trabajo ha sido parcialmente financiado por el Ministerio de Ciencia e Innovación a través de los proyectos TIN2008-06856-C05-05 (SD-TEAM UPM) y DPI2007-66846-C02-02 (ROBONAUTA).

strategies are based on an interpolation (linear, logarithmic, etc.) among the appropriate LM (Kneser and Steinbiss, 1993; Hsu, 2007), or on a re-estimation of the counts of each language unit (usually an n-gram), referred to as *count mixing* (Ljolje et al., 2000). Another adaptation techniques, such as MAP or MLLR, have been proposed (Liu, Gales, and Woodland, 2008), but it has been demonstrated (Bacchiani and Roark, 2003) that the performance of these techniques is similar to the one achieved with linear interpolation, but with a higher computational effort.

Regarding the ASR subprocedure or stage where the interpolated models are to be used, they can be integrated at the decoding stage (Justo and Torres, 2007), or they can be used in a re-scoring stage for improving the initial recognition hypotheses (López-Cózar and Callejas, 2006).

Finally, the language models can be estimated according to different strategies. Instead of merging a word-based LM and a class-based one, or using cache-based LM, that have obtained good results in building more general LM (Iyer and Ostendorf, 1999), we adapt the LM considering the evolution of the dialogue. In fact, people usually adapt their discourse depending on what they want to say, and who they are talking to. Thus a good way to adapt LM is to estimate time-dependent dynamic language models that evolve throughout the dialogue (Riccardi and Gorin, 2000).

State of the art SDS that use dialogue dependent language modeling usually take advantage of the information provided by the natural language understanding module (NLU, (López-Cózar and Callejas, 2006)). Our approach can estimate dynamic LM using both NLU information (i.e. the semantic information or dialogue concepts), and information obtained by the dialogue manager about the speaker's intentions (i.e. the objectives or goals that the user wants to fulfill).

Instead of using isolated concepts or goals to build each LM to interpolate (Lucas, Fernández, and Ferreiros, 2009), we propose to cluster different dialogue items (either concepts or goals) to reduce the number of language models to be considered, and to better estimate them, since each model will be built using a higher number of training sentences. We use a semi-automatic clustering approach, based on Latent Semantic Analysis (LSA, (Landauer, Foltz, and Laham, 1998)), for capturing the semantic relationships among the different dialogue items.

The rest of the paper is organized as follows. We present our baseline spoken dialogue system on Section 2. Section 3 shows our approach to cluster dialogue items, as well as the online interpolation strategy. The main results of our evaluation are presented on Section 4. Finally, the conclusions of our work are discussed on Section 5.

2 Baseline dialogue system

In this section we briefly present the spoken dialogue system we have modified.

We apply our dialogue system in the development of a conversational interface for controling a commercial Hi-Fi audio system using natural language sentences instead of a common infrared remote control.

By means of a mixed-initiative Bayesian Networks approach (Fernández et al., 2005), the Dialogue Manager (DM) infers the relationships between the semantics of an utterance (referred to as *dialogue concepts*) and the intention of the speaker (referred to as *dialogue goals*).

We have defined 58 concepts, which can be classified as *parameters* (16) that can be set up (for instance, the volume of the Hi-Fi device), *values* (20) that the different parameters can take (for instance, a given volume level), and *actions* (22) that the user can execute (for instance, modify the volume). We have also defined 15 different goals, according to the user intention and the available functionality of the Hi-Fi device (for instance, a change on the volume settings). Both dialogue items have been defined using expert knowledge of the application domain.

Our BN-based approach for dialogue management works as follows. First the ASR module extracts a recognition hypothesis, and the natural language understanding module (NLU) parses this hypothesis for extracting the dialogue concepts. Then the DM applies a Bayesian inference procedure to infer the dialogue goals, estimating a posterior probability for each of them, given their evidences (i.e. the presence or absence of each concept). After that, the system has to decide whether it has all the information needed for executing the inferred goals. This is done by another Bayesian inference procedure, which estimates the posterior probability of each concept, given their evidences, and the posterior probabilities of the inferred goals. The result of both inference procedures is used to decide the most suitable dialogue action to perform (either executing the actions the user asked for, or initiating a new dialogue turn to ask the user for any needed information).

3 *Clustering of dialogue items*

In our first approach to use semantic and intention-based information to dynamically adapt LM, we estimated a LM for each dialogue concept and each dialogue goal, and we apply an interpolation between all the LM related to the dialogue items inferred for each sentence.

This approach for dynamically adapting LM (using a LM for each dialogue item) has two main weaknesses. First of all, we had to consider a large number of LM on each interpolation step. Alas, the sparseness of this approach implied that several models had to be estimated using a reduced number of sentences, which could lead to a poor estimation of each LM.

In an effort for solving these limitations, we propose to apply a clustering strategy over the dialogue items, previously to the estimation of each LM. This clustering of dialogue items pursues two major goals. First of all, it can reduce the number of models to consider at the interpolation stage. Besides, the grouping of dialogue items can lead to a higher robustness of the LM to be interpolated, since each model will be estimated using more data (all the sentences that were used to train the LM for the isolated items that compose each cluster).

3.1 LSA-based clustering

We propose the use of Latent Semantic Analysis (LSA, (Landauer, Foltz, and Laham, 1998)) for performing our clustering. This technique can extract semantic relationships between our dialogue items (either concepts or goals).

The first step of LSA consists of building a co-occurrence matrix. In classical LSA, the co-occurrence is evaluated as the frequency that different words appear in different documents (Bellegarda et al., 1996). In our case, we obtain the frequency that each dialogue item appears on each sentence of our database. That is, our co-occurrence matrix is composed of our training sencences (as columns) and our dialogue items, either concepts or goals (as rows). Then, a Singular Value Decomposition (SVD) is applied over the matrix, transforming it as the product of three matrices: an upper triangular, a diagonal, and a lower triangular. Finally, we reestimate the co-occurrence matrix using the highest values of the diagonal matrix, specifying thus the projected dimension of the cooccurrence vectors.

This last step allows the relationships among dialogue items to be better estimated, in terms of strong variations on the Pearson's r correlation coefficient between them. We use these correlations over the projected vector space to establish which dialogue items can be grouped on each cluster.

LSA tends to cluster together those concepts or goals with a strong semantic relation between them. For instance, a cluster could include the concepts PARAMETER_VOLUME, VALUE_VOLUME, and ACTION_VOLUME all together.

Varying the dimension of the projected space (that is, the number of highest values of the diagonal matrix that will be used for re-estimating the co-occurrence matrix), we obtain different values for the correlations between dialogue items. We use them to establish a hierarchical structure of the clusters following a bottom-up strategy, from the isolated dialogue items, up to the highest layer that could be composed of all the dialogue concepts or goals.

Once we have obtained this cluster hierarchy, we select which clusters will be used during the online interpolation step. We decide that the clusters must form a partition of the dialogue item space, that is, every item (either concepts or goals) must be considered, and must not belong to more than one cluster. Using both restrictions and our correlation-based distance, we assure that the number of clusters takes low values, which makes the LM associated to each cluster more robust. Following our strategy, we consider 10 different clusters when using dialogue concepts, and 4 clusters when using goals.

3.2 LM generation

We have used a database of 516 sentences labeled at the lexical (words), semantic (concepts) and intention (goals) levels to estimate the LM related to each group generated using LSA and our correlation-based distance. Each model consists of a linear interpolation between unigram and bigram models.

We use each sentence of the database which makes reference to a certain dialogue item grouped in a cluster for estimating the LM related to that cluster. This way, the same sentence could be used to estimate different LM if that sentence makes reference to dialogue items that have been classified into different clusters. Consequently, we could assert that, each time two (or more) concepts or goals are classified into the same cluster, the number of sentences that will estimate the LM of the cluster will take a value between the maximum number of sentences among those for the isolated models, and the sum of the sentences that trained all of them.

3.3 Dynamic LM interpolation

As we stated in Section 2, after the understanding and dialogue management stages, the system has a set of available concepts and goals (which are related to one or several group-dependent LM). So once the system determines the models to interpolate, it has to estimate the interpolation weights for each of them, and another general weight for interpolating the dynamic model with a static, background LM.

Therefore, if we rewrite the well-known interpolation equation between probabilistic LM, we will include the dynamic LM in the form of a time dependency of the different interpolation weights. Thus the probability of a word w given its preceding words (its history) h in the interpolated model will be

$$p_T\left(w \mid h\right) = W_B \; p_B\left(w \mid h\right) + (1 - W_B)\; p_D\left(w \mid h\right) \tag{1}$$

being p_B the probability according with the background model, p_D the probability obtained dynamically with the group-based LM, and W_B the interpolation weight between both models.

The dynamic component of the interpolated model, p_D is also estimated as an interpolation among the LM associated to the groups that the inferred dialogue items belong to:

$$p_D\left(w \mid h\right) = \frac{1}{\sum_{G_i} W_{G_i}} \sum_{G_i} W_{G_i}\; p_{G_i}\left(w \mid h\right) \tag{2}$$

where p_{G_i} is the LM related to the group G_i, and W_{G_i} is the interpolation weight the system assigns to that LM.

Depending on the dialogue items taken into consideration for performing the clustering, the groups will be composed of dialogue concepts or goals.

Several strategies could be applied to estimate the interpolation weights W_{G_i}. Our objective was to automatically obtain them, using information available to the system. For the following analysis, let N_G be the number of clusters considered during the LM adaptation, n_i the number of items (concepts or goals) classified into the same cluster G_i, n the total number of dialogue items obtained from the input utterance, $p_f\left(g_i = 1 \mid e_{g_i}\right)$ the posterior probability that the goal g_i is present on the utterance under analysis given its evidence e_{g_i}, and $p_b\left(c_i \mid e_{c_i}\right)$, the posterior probability assigned by the DM to the concept c_i given its evidence e_{c_i}.

We have evaluated five different strategies to estimate W_{G_i}, which benefits and drawbacks we present below:

- **A0**. In this first approach we assign the same interpolation weight to each language model to be considered on each dialogue turn. That is, the weight assigned to the model associated to each cluster will be $1/N_G$. Whilst the best characteristic of this strategy is its easiness to obtain the interpolation weights, it does not take into account neither the number of items belonging to each considered cluster, nor the posterior probabilities of each dialogue item, estimated by the Dialogue Manager.
- **A1**. To take into account the proportion of dialogue items in each cluster, our second approach consists of using as interpolation weights the frequency of dialogue items of a sentence that belongs to each group. With this approach the interpolation weight will be $W_{G_i} = n_i/n$. Whilst this method is still easy to apply, and it is aware of the distribution of items into each cluster, it does not consider the confidence measures (that is, the posterior probabilities) that the dialogue manager obtains.
- **A2**. A way to include these confidence measures into the interpolation weights

consists of estimating them as the average of the posterior probabilities of the items of each cluster. That is, W_{G_i} will take the value

$$W_{G_i} = \frac{1}{n_i} \sum_{c_i \in G_i} p_b \left(c_i \mid e_{c_i} \right) \quad (3)$$

if we consider concept-based clustering, and

$$W_{G_i} = \frac{1}{n_i} \sum_{g_i \in G_i} p_f \left(g_i \mid e_{g_i} \right) \quad (4)$$

if we consider goal-based clustering.

As its main advantage, this method includes the knowledge obtained by the dialogue manager (i.e. the posterior probabilities of the dialogue items), but it does not take into account the distribution of them among the clusters, which may lead to a reduction of the relevance of a model with more items if its probabilities are reduced.

- **A3**. To partially avoid the effect of the previous approach, we could take the maximum of the posterior probabilities of the dialogue items of a cluster as the interpolation weight, being thus them

$$W_{G_i} = \max_{c_i \in G_i} p_b \left(c_i \mid e_{c_i} \right) \quad (5)$$

for concept clustering, and

$$W_{G_i} = \max_{g_i \in G_i} p_f \left(g_i \mid e_{g_i} \right) \quad (6)$$

for goal clustering.

Again, this method is really simple to implement. However, it has the same drawbacks than the **A2** approach: it does not take into account the number of items on each cluster. Even worse, it only considers the posterior probability of a single item, discarding the probabilities of the rest of items obtained from the utterance. Therefore, as the decision relies on just one item, it could cause that the LM related to a cluster with several items with reduce probabilities, and an item with high probability becomes more relevant than clusters with relatively high posterior probabilities.

- **A4**. In an effort to solve both problems (i.e. to use posterior probabilities and to consider the distribution of dialogue items into the clusters) we estimate the interpolation weights as the sum of the posterior probabilities of the items belonging to each cluster. Therefore, we will use the expression

$$W_{G_i} = \sum_{c_i \in G_i} p_b \left(c_i \mid e_{c_i} \right) \quad (7)$$

for concept clustering, and

$$W_{G_i} = \sum_{g_i \in G_i} p_f \left(g_i \mid e_{g_i} \right) \quad (8)$$

for goal clustering.

This approach could be considered as the most balanced one, in terms of giving importance to both the knowledge of the Dialogue Manager (i.e. the posterior probabilities), and the number of items belonging to each cluster.

Independently to the approach for obtaining W_{G_i}, the posterior probabilities obtained by the DM can be interpreted as a kind of confidence measure that the system has about the presence or absence (or the need) of the goals (and concepts). We include two degrees of freedom to be tuned during a validation stage. These parameters are the probability thresholds for concepts, Φ_C, and goals, Φ_G. The dynamic grammar generator will only take into account those items which posterior probabilities are above the corresponding threshold. As will be shown on Section 4, the use of those items with low probabilities may lead to a reduction of the performance of the dynamic LM.

4 Evaluation

This section presents the database we have used to evaluate the behaviour of the dynamically adapted LM, as well as the results of the different experiments we have carried out.

We have used a proprietary database called HIFI-MM1. This database is composed of 100 different sentences spoken by 13 speakers (7 male, 6 female), giving a total of 1300 sentences related with the application domain. By means of a k-fold approach we have split the database into ten different folds, each one with 130 sentences picked up randomly from the database.

Each sentence of the database has been manually labeled with its appropriate dialogue items. On average, each sentence makes reference to 4.31 concepts and 2.17 goals.

With the folds in which we split the database, we build three different sets: a *training* one, composed of eight folds (1040 sentences), and a *validation* and a *test* sets, each one with one fold (130 sentences).

Using round-robin we develop ten experiments. On each one we use the training subset to build the background LM, whilst the validation subset served us to tune the different parameters: LM weight (LMW), inter-word penalty (IWP), and concept and goal thresholds, Φ_C and Φ_G, as well as the interpolation weight with the background LM, W_B.

Using the test subset to evaluate the performance of the ASR, the baseline results (without using dynamic LM interpolation) shows a word error rate of 5.33 %.

We have evaluated the clustering approach using slots and goals separately, that is, using only semantic-based or only intention-based information for estimating the dynamic LM. As we stated in Section 3, the number of groups taken into consideration on each approach are 10 (when using concept-based grouping), and 4 (goal-based grouping).

Finally, we emphasize that we have analyzed the results of the recognition process when rescoring an utterance with the information obtained from that utterance. We will further use this results as an oracle, or an upper bound of the performance of our LM adaptation approach.

4.1 Concept-based clustering

We have tested the performance of the speech recognition system when using only the LM associated to the clusters built from dialogue concepts. In this first experiment, we have used the validation set to estimate the interpolation weight with the background LM, W_B, and the relevance threshold for concepts, Φ_C.

One of our goals was to determine which approach for obtaining the interpolation weights among the LM associated to the clusters, W_{G_i}, was the most appropriate. Table 1 shows the results of this experiment, in terms of Word Error Rate (WER) as well as the relative improvement (both in %) with respect to the baseline performance (5.33 %, see the previous Section).

App.	W_B	Φ_C	WER (%)	Rel. impr. (%)
A0	0.84	0.75	4.84	9.27
A1	0.83	0.55	4.67	**12.36**
A2	0.83	0.46	4.78	10.39
A3	0.81	0.45	4.78	10.39
A4	0.85	0.47	4.90	8.15

Table 1: WER when using concept-based clustering

It is interesting to remark that the best result is obtained with the approach **A2** to obtain the interpolation weights of the cluster-dependent LM (that is, the frequency of concepts belonging to each cluster), which was considered a priori as a weak approach (see Section 3.3), while the strategy **A4** (i.e. the sum of posterior probabilities) achieves the lesser improvement. This could be due to the relatively high number of concepts belonging to each cluster (an average value of 5.8), which may cause that the weights of those clusters with few concepts arise, modulating thus the relevance of the LM.

We can see that, independently to the approach to obtain W_{G_i}, the interpolation weight between the dynamic LM and the background one, W_B, takes very close values, always above 0.8. This means that the concept-based clustering can lead to an improvement even with a small contribution of the dynamic language model (always below 20 %).

The values of the relevance threshold, Φ_C, on any of the strategies to estimate W_{G_i}, imply that the system can take advantage of those concepts with middle posterior probabilities, not only using the best rated ones. Thus an important knowledge relies on concepts which our approach could consider as optionals to solve the current dialogue.

Finally, despite the results are not significant, since the confidence intervals (up to 0.52 %) still overlap, the improvement tendency is very promising, reaching an improvement of up to 12.36 % in the best case (boldfaced in Table 1).

4.2 Goal-based clustering

In our second experiment we took into account the clusters generated using only dialogue goals (i.e. discursive or intention-based information). We used the validation set to optimize the values of W_B and Φ_G, that is, the relevance threshold for goals. The results of this experiment, presented in table 2, shows the WER reduction when using each of the strategies for estimating the interpolation weights among the cluster-dependent LM presented in Section 3.3.

App.	W_B	Φ_G	WER (%)	Rel. impr. (%)
A0	0.77	0.53	4.85	8.99
A1	0.77	0.51	4.82	9.55
A2	0.78	0.41	4.75	10.96
A3	0.78	0.42	4.79	10.11
A4	0.76	0.46	4.70	**11.80**

Table 2: WER when using goal-based clustering

Despite the dialogue goals contain a more integrated information than the dialogue concepts, the best performance (with a relative improvement of 11.80 %, boldfaced in Table 2) keeps below the one obtained using concept-based clustering. This could happen because the goal-based grouping could not be the optimum one. In any case, the differences are not significant.

The weight of the background model W_B takes rather similar values than in the previous experiment, which means that, as we stated before, the goal-based clustering could be useful for improving the performance of the recognition.

Likewise, and despite that the relevance of the goals should be greater than the concept-based one (since dialogue goals carry more information), the values of the relevance threshold for goals, Φ_G, are pretty similar to those of the concept threshold, Φ_C, on any of the strategies for obtaining the interpolation weights of each dynamic LM, W_{G_i}. Even the goals with intermediate posterior probabilities carry important information in terms of adapting language models.

With goal-based clustering, the best performance is achieved when using the approach **A4** to estimate W_{G_i}. As we hypothesized, it reaches the best tradeoff between the number of items in a single cluster, and their posterior probabilities.

5 *Conclusions*

We present a semi-automatic approach for clustering semantic or discursive items (i.e. dialogue concepts or goals) as a previous step in the dynamic interpolation of language models. We exploit the semantic relationships, inferred by LSA, in order to build the clusters.

We have tested five different strategies in order to estimate the interpolation weights among the LM interpolated on each dialogue turn. The experiments we have carried out (see Section 4) shows that, since the approach **A4** (i.e. estimating the interpolation weights as the sum of the posterior probabilities of items of the same group) takes into account every posterior probability of items in a cluster (without averaging them), it can achieve a tradeoff between the information provided by the Dialogue Manager (that is, the posterior probabilities of each dialogue item) and the complexity of each group (as the number of items inferred that have been classified into the same cluster).

The difference of the improvement of using either dialogue concepts or goals is not significant. However, it presents a tendency of achieving a higher performance when using semantic information. As our dialogue goals can be considered as a source of more integrated and reliable information, this may imply that the set of goal-based clusters could be better estimated.

We are currently studying several approaches to merge both dialogue items (concepts and goals) into a single hierarchical structure of clusters. We are testing different weights to each component of a cluster, provided that the dialogue goals could be more reliable than the concepts.

In this work we have focused on rescoring each utterance with the knowledge obtained from the same utterance. However, the final function of the system consists of using the information of the previous dialogue turns (either the concepts provided by the user, the inferred goals, or both) to improve the recognition of the current utterance.

Instead of using Pearson's r correlation coefficient as the distance metric for generating the clusters, we are now evaluating another different metrics, such as a cosine-based metric, well-known in research using

LSA (Bellegarda et al., 1996). Furthermore, instead of using the raw data when building the co-occurrence matrix, we want to perform a preprocessing stage, trying to smooth the values in this matrix. This smoothing strategies have proven to achieve good performance when clustering words.

Additionaly, we are performing a fully automatic clustering strategy, basing it on some performance measure, such as the reduction of the perplexity of a clustered model in comparison with the isolated LM. This way, we will automatize the offline generation of language models to be used online.

Finally, we could also use the full hierarchical structure of clusters that LSA generates. This way, instead of having a single level of clusters, we will use several LM, even when considering only one concept and/or goal. To obtain more significant results we are labeling a database which comprises a higher number of sentences and human-machine dialogues for training the LM associated to each cluster, estimating more robust models and, eventually, making the results of our approach more significant.

References

Bacchiani, M. and B. Roark. 2003. Unsupervised Language Model Adaptation. In *Proceedings of the International Conference on Acoustic, Speech and Signal Processing (ICASSP03)*, volume I, pages 224–227.

Bellegarda, J.R. 2004. Statistical language model adaptation: review and perspectives. *Speech Communication*, 42:93–108.

Bellegarda, J.R., J.W. Butzberger, Y.L. Chow, N.B Coccaro, and D. Naik. 1996. A Novel Word Clustering Algorithm Based on Latent Semantic Analysis. In *Proceedings of the International Conference on Acoustic, Speech and Signal Processing (ICASSP96)*, volume I, pages 172–175.

Fernández, F., J. Ferreiros, V. Sama, J.M. Montero, R. San-Segundo, and J. Macías-Guarasa. 2005. Speech Interface for Controlling a Hi-Fi Audio System Based on a Bayesian Belief Networks Approach for Dialog Modeling. In *Proceedings of the 9th European Conference on Speech Communication and Technology (INTERSPEECH05)*, pages 3421–3424.

Hsu, B-J. 2007. Generalized Linear Interpolation of Language Models. In *Proceedings of the International Conference of Automatic Speech Recognition and Understanding (ASRU07)*, pages 136–140.

Iyer, R.M. and M. Ostendorf. 1999. Modeling Long Distance Dependence in Language: Topics Mixtures Versus Dynamic Cache Models. *IEEE Transactions on Speech and Audio Processing*, 7(1):30–39.

Justo, R. and M.I. Torres. 2007. Word Segments in Category-Based Language Models for Automatic Speech Recognition. *Lecture Notes in Computer Science*, 4477:249–256.

Kneser, R. and V. Steinbiss. 1993. On the Dynamic Adaptation of Stochastic Language Models. In *Proceedings of the International Conference on Acoustic, Speech and Signal Processing (ICASSP93)*, volume II, pages 586–589.

Landauer, T.K., P.W. Foltz, and D. Laham. 1998. An Introduction to Latent Semantic Analysis. *Discourse Processes*, 25:259–284.

Liu, X., M.J.F. Gales, and P.C. Woodland. 2008. Context Dependent Language Model Adaptation. In *Proceedings of the 8th International Conference on Speech Communication and Technology (INTERSPEECH08)*, pages 837–840.

Ljolje, A., D.M. Hindle, M.D. Riley, and R.W. Sproat. 2000. The AT&T LVCSR-2000 System. In *NIST LVCSR Workshop*.

López-Cózar, R. and Z. Callejas. 2006. Combining language models in the input interface of a spoken dialogue system. *Computer, Speech and Language*, 20:420–440.

Lucas, J.M., F. Fernández, and J. Ferreiros. 2009. Using Dialogue-Based Dynamic Language Models for Improving Speech Recognition. In *Proceedings of the 9th International Conference on Speech Communication and Technology (INTERSPEECH09)*, pages 2471–2474.

Riccardi, G. and A.L. Gorin. 2000. Stochastic Language Adaptation over Time and State in Natural Spoken Dialog Systems. *IEEE Transactions on Speech and Audio Processing*, 8(1):3–10.