Mutual Information and Perplexity Based Clustering of Dialogue Information for Dynamic Adaptation of Language Models

Juan Manuel Lucas-Cuesta, Fernando Fernández-Martínez,
Tirso Moreno, and Javier Ferreiros

Abstract. We present two approaches to cluster dialogue-based information obtained by the speech understanding module and the dialogue manager of a spoken dialogue system. The purpose is to estimate a language model related to each cluster, and use them to dynamically modify the model of the speech recognizer at each dialogue turn. In the first approach we build the cluster tree using local decisions based on a Maximum Normalized Mutual Information criterion. In the second one we take global decisions, based on the optimization of the global perplexity of the combination of the cluster-related LMs. Our experiments show a relative reduction of the word error rate of 15.17%, which helps to improve the performance of the understanding and the dialogue manager modules.

Keywords: Spoken Dialogue System, Language Models, Dialogue-based Information, Clustering.

1 Introduction

Statistical language model adaptation has become a current issue within the scope of Speech Technology. It aims at modifying the language model (LM) of which a speech recognition system (ASR) makes use, to improve the recognition performance. For instance we can modify a general LM to adapt it to a closed domain, trying to improve the overall response of a domain-dependent system in which the ASR is included.

There are several approaches to adapt LMs, depending on the sources of the adaptation models [3]. Perhaps the simplest one consists of a linear interpolation between LMs [6]. This approach tries to find out an accurate weight to combine a *background* LM, built with more general data, with one or several adaptation LM, usually built with more specific data.

The adaptation LMs could be estimated at each dialogue turn [9]. Dialogue systems that use dialogue-dependent LMs usually consider the semantic information of each utterance. We estimate the LMs using semantic information as well as the user intentions ellaborated by the dialogue manager [8].

To learn more robust models, we group those information elements that share common features (such as the semantics or the word classes) prior to the LM estimation. To discover these relationships, several techniques such as the application of Latent Semantic Analysis (LSA) have been proposed [2].

In this work we propose two clustering techniques, using as clustering criteria two metrics derived from the Information Theory. On the one hand, the Normalized Mutual Information (NMI), previously used for the estimation of parameters of acoustic models for speech recognition [1], or for the adaptation of trigger-based LMs [5]. On the other hand, a minimization of the global perplexity of a LM obtained as the interpolation of all the clusters considered. Our aim is to reach a tradeoff between the *specificity* of having a large number of LMs related to single pieces of information, and the *robustness* of having few LMs, but trained with more data.

The rest of the paper is organized as follows. We first describe our dialogue system (Section 2), and our approaches to cluster dialogue elements (Section 3). The interpolation technique that we apply is shown in Section 4. Finally, the evaluation results are discussed in Section 5, and the conclusions of the work are drawn in Section 6.

2 Baseline Dialogue System

We have designed a user-independent, mixed-initiative dialogue system for controlling household devices. In this work we focus on the control of a Hi-Fi audio system using speech, instead of an infrared remote control.

The Dialogue Manager (DM) is based on a Bayesian Networks (BNs) solution [4] that exploits the causal relationships between the semantics of an utterance (i.e. the *dialogue concepts*), and the intention of the user (i.e. the *goals*). We will refer to both concepts and goals as *dialogue elements*. These elements have been defined by hand using expert knowledge of the application domain.

We have defined a set of 58 concepts that cover all the semantic categories in the application domain. These concepts could be classified into three sets: *actions* (22) to be executed (e.g. to play), *parameters* (16) that can be set up (e.g. the volume), and their corresponding *values* (20). We have also defined 15 goals, according to the available functionality of the Hi-Fi audio system. A concept or a goal is *present* only if it has been extracted from the recognized utterance (by the understanding module), or positively inferred (by the DM).

As an example of our definition of dialogue elements, let us consider the utterance *raise the volume to five.* The understanding module can extract the concepts PARAM_VOL ('volume'), VALUE_VOL ('five'), and ACTION_VOL ('raise'). The dialogue goal that should be inferred is MODIFY_VOLUME.

Once the ASR has recognized the input utterance, and the understanding module has extracted the concepts of that utterance, the DM has to identify the goals, using the information available (i.e. the concepts). This task is carried out by means of a *forward inference* procedure (FI), that estimates the posterior probability of each goal, given the available evidences (the presence or absence of each concept in the history of the dialogue). By comparing the resulting probabilities with several predefined thresholds, the DM decides whether a goal is *present* or *absent.*

After the FI process, the DM estimates similar probabilities for the concepts, assuming the inferred goals as new evidences. This task is developed by means of a *backward inference* (BI) procedure. The decision of assuming whether a concept is needed or not is taken by comparing the probabilities against different thresholds. The result of this process is used to carry out the most suitable action (either performing the goals the user has addressed, if the system has the information needed to accomplish them, or asking the user for the wrong or the incomplete information otherwise).

3 Clustering of Dialogue Elements

This section presents the clustering approaches that we have developed to group dialogue elements, as well as the dynamic LM interpolation that we carry out.

Our proposal is a bottom-up, greedy algorithm that builds a hierarchy of clusters, each of which will have a LM associated. The hierarchy will be established from a starting point in which each cluster will be composed of a single dialogue element, to an ending cluster which contains all the dialogue elements (and therefore it could be assimilated to the general, background LM).

We have proposed two algorithms based on the estimation of the perplexity of LMs The first algorithm performs a method that exploits *local* information to decide which elements should be grouped (that is, the metric is obtained by using only those models directly related to the cluster that is potentially eligible). The second one estimates a *global* measure obtained as a contribution of all the models that are present at each step of the algorithm, and chooses the model that optimizes that measure.

3.1 Maximum Mutual Information Criterion

Let us suppose a set of labeled sentences with which we will train two different language models, A and B, each of which is related to a certain dialogue-specific content (for instance, a dialogue concept or a dialogue goal). We could assume that both LMs have a common subset of training sentences (i.e. they share some knowledge, either lexical, semantic, or intention). Let us further assume that we have obtained the perplexities of both models against an additional database.

The perplexity is related to the average number of words between which a model has to decide the most suitable one. We can estimate the perplexity of a model as $pp_A = 2^{H(A)}$, being $H\left(A\right)$ the entropy of that model. In other words, the entropy of the LM A can be obtained as $H\left(A\right) = log_2 pp_A$.

On the other hand, the *mutual information* shared between two random variables can be expressed as $I\left(A;B\right) = H\left(A\right) + H\left(B\right) - H\left(A,B\right)$. Instead of considering the Mutual Information between two LMs, we use the Normalized Mutual Information (NMI), that can be expressed as $NMI\left(A;B\right) = \frac{H(A)+H(B)}{H(A,B)}$.

According to this criterion, we will cluster the elements that maximize the NMI of their related LMs.

We can express the NMI between two models in terms of their perplexity:

$$NMI\left(A,B\right)=\frac{\log_2 pp_A\ pp_B}{\log_2 pp_{AB}} \tag{1}$$

where $pp_{A,B}$ stands for the perplexity of the joint LM, that is, the LM estimated when using the sentences that trained the models A and B (without repeating the common sentences).

This criterion tends to group elements that share common information (i.e. dialogue elements, or sentences that make reference to those elements). It also allows us to reach a tradeoff between low values of perplexity (that tends to lead to better LMs) and the complexity of the models (in terms of information used to estimate them). We use this criterion since we have several elements for which the number of training sentences is so reduced that their LMs give reduced perplexities, but only due to the lack of training data.

3.2 Minimum Perplexity Criterion

We could consider the NMI criterion as a local one, since the decision of which is the optimum group at each step of the algorithm is taken by considering only the mutual information between those elements that are to be merged, and the resulting cluster. We have also implemented a clustering strategy based on a global criterion, that is, in which the decision on which elements to cluster depends on a metric obtained from all the clusters considered at each step of the algorithm. This criterion is based on a linear interpolation between the LMs related to the clusters that are considered at each step of the algorithm. Then the system estimates the perplexity of the resulting LM. The cluster selected is the one that minimizes the perplexity of the global model.

We assign the same interpolation weight to each LM. That is, if at a certain step of the algorithm there are N_S clusters, the LM related to each model will have an interpolation weight of $1/N_S$.

Therefore, if we represent the probability of obtaining a word w given its history h with the LM related to cluster S_k as $p_{S_k}\left(w \mid h\right)$, the corresponding probability in the global, artificial model, p_G, at a certain iteration of the algorithm, can be obtained as

$$p_G\left(w \mid h\right)=\frac{1}{N_S}\left[p_{S_{ij}}\left(w \mid h\right)+\sum_{\substack{k=1,\\ k\neq i,j}}^{N_S} p_{S_k}\left(w \mid h\right)\right] \tag{2}$$

Once the system obtains the perplexity of p_G, the process is repeated for each available combination ij of elements to be grouped (i.e. for each potential cluster). As a result the algorithm obtains a set of global LMs related to all the potential clusters. The algorithm selects as the new cluster to be included in the hierarchy the one that obtains the lowest perplexity among all of them. The rest of the potential clusters are disregarded in the current step of the algorithm. Nevertheless, they could be considered as potential clusters in further iterations.

The global perplexity minimization criterion is similar to the NMI-based one in the sense that both criteria allows us to obtain groups of elements that share common information. With the NMI metric the system groups those elements that share a high amount of common sentences (i.e. strongly related from the point of view of vocabulary and semantics). In the global perplexity one, the result is similar, but from the model robustness' perspective. That is, the elements that are clustered together are those ones that lead to a better estimated LM. The main difference between both criteria is related to the computing time. The global perplexity minimization one has a higher computational complexity since it has to estimate a higher number of models at each iteration (not only the LM related to the cluster that is included to the hierarchy, but also the specific models and the global one for each potential cluster).

3.3 Estimating a Correction Function

After carrying out some initial clustering experiments, we found that both the NMI and the global perplexity criteria have a main drawback. The cluster hierarchies that are obtained are unbalanced, in the sense that after the first grouping, a cluster with a high number of sentences is obtained. The rest of elements tend to join that cluster instead of building more specific groups. In order to reach a tradeoff between the perplexity of each LM and their complexity (in terms of the number of sentences that will train the corresponding LM, and the number of elements into each cluster), we propose to obtain a complexity correction function that will take a positive value.

The motivation of defining a correction function is to enable the clustering of those elements which have a strong lexical or semantic relationship, even though the related LMs are trained with a reduced number of sentences. This fact will avoid the generation of a too general model with which the rest of elements are progressively joined. In other words, the system can keep an important degree of specificity in the early steps of the clustering algorithm.

Taking into account that we want to optimize the criterion metric, the correction function is applied in two different ways, depending on the chosen criterion for the clistering. In the case of the the NMI measure (which is a maximization function), we will apply the function as a division factor prior to decide which elements to cluster. In a similar fashion, the global perplexity metric (minimization function) will be multiplied by the correction factor.

We will make the correction function dependent on the main features of each cluster, namely the number of dialogue elements that form each cluster, and the number of sentences with which the LM associated to the cluster will be estimated.

The number of elements joined in a given cluster S_i, which we denote as N_{S_i}, will model the complexity of the clusters. It is used to allow those clusters with few elements to be joined among them, avoiding thus the tendency to join a cluster with more elements, which in turn leads to less specific LMs, especially in the initial steps of the clustering algorithm.

The correction criterion will also take into account the number of sentences n_A and n_B that have been used to train the LMs related to the clusters to be joined, as well as the number of sentences of the resulting cluster, n_{AB}. We use the number of sentences as a value that can measure both the complexity of the model and also its robustness (the larger the number of sentences to train a LM, the better it will be estimated).

The correction function will consider the number of sentences in the sense of favoring the union of those elements that share a large number of common sentences and a reduced number of different sentences.

The situation in which the correction function reaches its maximum value arises when there are not any sentence in common between both models. In other words, a lexical or semantic relationship between both clusters A and B is too weak or inexistent, and therefore both clusters should not be joined in the current step of the algorithm. This situation arises when $n_{AB} = n_A + n_B$.

A final restriction that we apply to the correction function is that the contribution of the number of sentences is measured on a logarithmic scale. We decide that since the number of sentences with which the LMs are trained is about two orders of magnitude over the entropy of the models (which is also a logarithmic magnitude).

Taking these conditions into account, the expression of the correction function CF for joining two clusters A and B into a single cluster AB is

$$CF = N_{S_i} \ \ln \left[\frac{\sqrt{(n_{AB} - n_B) \ (n_{AB} - n_A)}}{n_A + n_B - n_{AB}} + \mathcal{K}_0 \right] \tag{3}$$

where $\mathcal{K}_0$ is a constant that assures that the logarithm takes a positive value.

We finally apply a pruning process to the cluster hierarchies obtained. The idea is to keep these LMs that are trained with a sufficient number of sentences, and also assuring that each LM is related to a specific content (i.e. we try to reach a tradeoff between *robustness* and *specificity* of the LMs. The number of LMs to be considered are 10 (when using goal-based information), 23 (when considering concepts), and 25 (when grouping both dialogue elements).

4 Dynamic Language Model Generation

We have included a new module as a feedback loop between the ASR, the NLU, and the DM modules. This new element, the Dynamic LM Generator, will consider the information provided by the user in the current and the previous utterances to dynamically modify the LMs that the ASR makes use of.

We first estimate the LM related to each cluster. Instead of keeping a LM for each dialogue element, as we proposed in [7], we consider that keeping 73 LMs is a suboptimal approach, since several of these models are poorly estimated, due to the limited amount of sentences that make reference to those elements. Therefore, we proposed to group the dialogue elements in a hierarchical cluster structure, according to the semantic relationships among them ([8]). Our aim is to reach a tradeoff between the *specificity* of having a large number of LMs related to single pieces of information, and the *robustness* of having few LMs, but trained with more data.

At each dialogue turn, once a sentence has been recognized, and the DM has developed both forward and backward inferences, the posterior probabilities of concepts and goals are used to decide which LMs will be interpolated. We base this decision on the comparison of the posterior probabilities of the dialogue elements against different *relevance thresholds*, Φ_C for concepts and Φ_G for goals. We find the optimal values for Φ_C and Φ_G at a validation stage. We perform the LM adaptation by means of a linear interpolation between a background LM, p_B, and a *context-dependent* LM, p_D. The probability of a word w given its preceding words (its history) h in the interpolated model will then be

$$p_I\left(w \mid h\right) = \left(1 - \lambda_D\right) p_B\left(w \mid h\right) + \lambda_D \, p_D\left(w \mid h\right) \tag{4}$$

being λ_D the interpolation weight between the background LM and the context-dependent LM, p_D. This model is also built by interpolating the LMs related to clusters to which the dialogue elements belong to. The interpolation weights are obtained as functions of the posterior probabilities of each dialogue element, and also as a function of the number of elements on each cluster.

By using the summation of posterior probabilities we can achieve a tradeoff between the contribution of the number of elements belonging to each cluster, and their posterior probabilities, giving more relevance to those clusters to which more dialogue elements belong to, or to those ones with the dialogue elements with greater posterior probabilities.

5 Experimental Setup

This section presents the database that we have used to assess the adapted system, and the evaluation results.

Our proprietary database comprises 1300 different sentences, uttered by 13 speakers (7 male, 6 female), giving a vocabulary of 391 words. Each sentence has been manually labeled with its appropriate concepts and goals. By means of a k-fold approach we have split the database into ten folds (each one with 130 sentences picked up randomly from the database), with which we build three 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 set to build the LMs, whereas the validation set is used to adjust the parameters of the system.

We have evaluated the word error rate (WER) of the speech recognizer, the concept error rate (CER) of the understanding module, and the goal error rate (GER, that is, the percentage of errors in the inference of goals).

Throughout the evaluation we have assessed the performance of the system when using the concepts and goals extracted from an utterance to dynamically adapt the LM, and use it to recognize again the same sentence. This way we can estimate an upper bound of the performance of our system.

5.1 Using the NMI Criterion

In our first experiment we consider the clustering strategy based on maximum normalized mutual information (NMI). Table 1 shows the results of the evaluation in terms of WER, CER and GER, when considering only concept-dependent information, only goal-dependent information, or when merging both elements for the clustering. We also include the performance of the baseline system (i.e. with the background, static LM).

Table 1. Performance of the NMI-based language modeling

Clustering approach	WER (%)	CER (%)	GER (%)
Baseline	5.33	13.37	26.20
Concepts	4.82	12.73	25.67
Goals	4.84	12.68	**25.53**
Both	4.70	12.66	25.71

The interpolation weight λ_D takes values of about 0.15. That is, it is enough to slightly modify the LM (keeping a 85% of the background LM) to achieve improvements in the three metrics considered. The improvements reach a maximum relative value (in terms of error reduction) of 11.80% WER and 5.34% CER (both when considering the clustering of both dialogue elements together). On the other hand, the maximum relative error reduction in Goal Error Rate (2.56%) is reached when considering only dialogue goals. The main reason for this behaviour is that using only goal-based information (that is, the more integrated source of information that the system considers) implies a reduction of the insertions of goals into the hypothesis, which are the most important source of errors. In any case, the size of our database makes that the improvements in GER are not statistically significant.

5.2 Using the Minimum Perplexity Criterion

We next evaluate the performance of the adapted system when using the Minimum Global Perplexity criterion. Table 2 shows the results of the evaluation of this strategy.

Table 2. Performance of the Minimum Perplexity-based language modeling

Clustering approach	WER (%)	CER (%)	GER (%)
Baseline	5.33	13.37	26.20
Concepts	**4.52**	**12.54**	25.60
Goals	4.60	12.59	25.64
Both	4.58	12.66	25.64

The interpolation weight λ_D between the background LM and the context-dependent one (i.e. the generated using the LMs associated to the clusters considered) takes a value of about 0.21. Using this clustering strategy, the relevance of the context-dependent component is higher than with the NMI-based clustering approach. This fact implies that the LMs obtained with the Maximum Global Perplexity criterion tend to be better estimated. This leads to a slightly better performance of the system (with maximum relative error reduction of 15.17% for Word Error Rate, and 6.28% for Concept Error Rate, both when considering concept-based clustering). The improvement of the WER is statistically significant with confidence intervals of 90%. As regards the dialogue performance, the GER also tends to decrease (up to a maximum of 2.29% of relative reduction). However, this value is not statistically significant.

Merging both dialogue elements cannot outperform the strategies of using the elements separately. This could happen due to the fact that the goals are inferred using the concepts. Therefore, using both sources of information may cause the estimation of LMs with redundant information. This redundancy could cause the reduction of the performance observed. In any case, the differences between the performance of the clustering strategies are not significant.

6 Conclusions

We have presented two strategies to cluster dialogue-based information that is used to generate content-specific language models. The first approach is based on a local criterion that considers the Normalized Mutual Information (NMI) to decide which elements to cluster at each step of the algorithm. The second one is based on a global criterion that tries to minimize the perplexity of a model obtained as a linear interpolation of the LMs related to the clusters considered. The LMs obtained are interpolated at each turn with a background LM to dynamically adapt the model to be used by the recognizer.

Instead of training the most accurate interpolation weights, one of our main claims is that the system can estimate accurate interpolation weights dynamically using the posterior probabilities obtained by the DM. This way, the more confident the system is when inferring a given concept or goal, the more relevant the LM associated to that dialogue element will be in the dynamic LM estimated at that turn.

The evaluation results show that these clustering strategies lead to an estimation of LMs which can improve the recognition performance. More importantly, the improvement of these LMs (used by the speech recognizer) tends to improve the performance of other modules of the system (the speech understanding and the DM). We have also seen that the clustering based on the minimization of the perplexity tends to obtain better LMs (from both the specificity and the robustness points of view) than the NMI-based one.

We are aware that the databases that we have used are limited. We are now acquiring and preparing new data to train the LMs related to the different dialogue elements. This way we have to label this data at the three levels of information (lexical, semantic, and user intention).

We are now working on another interpolation strategy for the Minimum Perplexity approach. Instead of using the same weight for all the LMs, we will make them dependent on the complexity of each cluster.

We are also defining a strategy to adjust dynamically the weight λ_D between the background and the context-dependent LMs, instead of obtaining it at a validation stage.

We are also applying our adaptation paradigm to other information sources, such as the knowledge that the system has about the users, taking into account that each speaker may express their ideas in different ways. The system could take advantage of this information once it identifies the speaker, to adapt the LMs to the characteristics of each user.

Acknowledgements. This work has been partially supported by the Spanish Ministry of Science and Innovation, under contracts TIN2008-06856-C05-05 (SD-TEAM UPM) and DPI2010-21247-C02-02 (INAPRA), and by the Spanish Ministry of Education, under contract AP2007-00463 (FPU Grant).

References

1. Bahl, R.L., Brown, P.F., de Souza, P.V., Mercer, R.L.: Maximum Mutual Information Estimation of Hidden Markov Model Parameters for Speech Recognition. In: Proc. ICASSP, pp. 49–52 (1986)
2. Bellegarda, J.R., Butzberger, J.W., Chow, Y.L., Coccaro, N.B., Naik, D.: A Novel Word Clustering Algorithm Based on Latent Semantic Analysis. In: Proc. ICASSP, vol. I, pp. 172–175 (1996)
3. Bellegarda, J.R.: Statistical language model adaptation: review and perspectives. Speech Comm. 42, 93–108 (2004)
4. Fernández, F., Ferreiros, J., Sama, V., Montero, J.M., San-Segundo, R., Macías-Guarasa, J.: Speech Interface for Controlling a Hi-Fi Audio System Based on a Bayesian Belief Networks Approach for Dialog Modeling. In: Proc. INTERSPEECH, pp. 3421–3424 (2005)
5. GuoDong, Z., KimTeng, L.: Interpolation of n-gram and mutual-information based trigger pair language models for Mandarin speech recognition. Comp. Speech & Lang. 13, 125–141 (1999)
6. Kneser, R., Steinbiss, V.: On the dynamic adaptation of stochastic language models. In: Proc. ICASSP, vol. II, pp. 586–589 (1993)
7. Lucas-Cuesta, J.M., Fernández, F., Ferreiros, J.: Using Dialogue-Based Dynamic Language Models for Improving Speech Recognition. In: INTERSPEECH, pp. 2471–2474 (2009)
8. Lucas-Cuesta, J.M., Fernández, F., López, V., Ferreiros, J., San-Segundo, R.: Clustering of syntactic and discursive information for the dynamic adaptation of Language Models. In: SEPLN, vol. 45, pp. 175–182 (2010)
9. Solsona, R.A., Fosler-Lussier, E., Kuo, H.K.J., Potamianos, A., Zitouni, I.: Adaptive Language Models for Spoken Dialogue Systems. In: Proc. ICASSP, vol. I, pp. 37–40 (2002)