Expert Systems with Applications 40 (2013) 1069–1085

Contents lists available at SciVerse ScienceDirect

Expert Systems with Applications

journal homepage: www.elsevier.com/locate/eswa

On the dynamic adaptation of language models based on dialogue information

J.M. Lucas-Cuesta *, J. Ferreiros, F. Fernández-Martínez, J.D. Echeverry, S. Lutfi

Speech Technology Group, Universidad Politécnica de Madrid, Avenida Complutense 30, 28040 Madrid, Spain

ARTICLE INFO

Keywords:
Spoken dialogue system
Speech recognition
Language models
Dynamic adaptation
Semantic clustering
Dialogue-based information

ABSTRACT

We present an approach to adapt dynamically the language models (LMs) used by a speech recognizer that is part of a spoken dialogue system. We have developed a grammar generation strategy that automatically adapts the LMs using the semantic information that the user provides (represented as dialogue concepts), together with the information regarding the intentions of the speaker (inferred by the dialogue manager, and represented as dialogue goals). We carry out the adaptation as a linear interpolation between a background LM, and one or more of the LMs associated to the dialogue elements (concepts or goals) addressed by the user. The interpolation weights between those models are automatically estimated on each dialogue turn, using measures such as the posterior probabilities of concepts and goals, estimated as part of the inference procedure to determine the actions to be carried out. We propose two approaches to handle the LMs related to concepts and goals. Whereas in the first one we estimate a LM for each one of them, in the second one we apply several clustering strategies to group together those elements that share some common properties, and estimate a LM for each cluster. Our evaluation shows how the system can estimate a dynamic model adapted to each dialogue turn, which helps to significantly improve the performance of the speech recognition, which leads to an improvement in both the language understanding and the dialogue management tasks.

1. Introduction

Statistical language model adaptation has become an area of current interest within the scope of Speech Technology research. Its main goal consists of updating the language model (LM) which an automatic speech recognition system (ASR) makes use of, in order to achieve better recognition rates. The sources of information that could be used to modify the LM can be of any nature. For instance, one might use information available online, which is closely related to the topic that is currently being addressed in the conversation to better match the user utterances. We could also think about adapting the behaviour of the recognition module to the current user, given that different users speak in a different way, not only from the acoustic point of view, but also at a lexical or discourse level.

1.1. Previous work

During the last 20 years there has been a lot of effort in introducing new and up-to-date linguistic information into speech recognizers (see, for instance, Bellegarda, 2001, 2004). We could broadly classify the different approaches into several criteria. For instance, the **nature of the data** that are considered for the adaptation (they could be obtained from external databases through an Information Retrieval process, such as in Martins, Teixeira, & Neto (2010), or the user could provide them, for instance in previous interactions, such as in Kuhn & de Mori (1990)); the **goal of the adaptation** (we could adapt to the topic that is currently being addressed, as shown in Shi et al. (2008), or to the different speakers that use the system, such as in Tur & Stolcke (2007)); or the **degree of knowledge** that the system has on those data (whether the data are previously labelled, or not, respectively carrying out a *supervised*, (Kneser & Steinbiss, 1993) or an *unsupervised* approach, (Bacchiani & Roark, 2003)).

Perhaps more interesting classifications of LM adaptation strategies can be defined by trying to solve three questions: (a) how is the information used to adapt the models obtained; (b) where could the adapted models be applied, and (c) which adaptation strategy is adopted, i.e. how the adaptation data can be used (either switching to the most appropriate model, or generating a new one using the available information).

As regards the **information sources** that could be used, they usually depend on the application domain in which the LM adaptation paradigm is included. One of the most common sources of data for adaptation purposes is the Internet. It is usually queried when trying to adapt the LMs to a specific topic (Lecorvé, Gravier, & Sébillot, 2009; Shi et al., 2008), or to the most recent content,

* Corresponding author. Tel.: +34 915495700x4241; fax: +34 913367323.
E-mail addresses: juanmak@die.upm.es (J.M. Lucas-Cuesta), jfl@die.upm.es (J. Ferreiros), ffm@die.upm.es (F. Fernández-Martínez), jdec@die.upm.es (J.D. Echeverry), syaheerah@die.upm.es (S. Lutfi).

0957-4174/$ - see front matter © 2012 Elsevier Ltd. All rights reserved.
http://dx.doi.org/10.1016/j.eswa.2012.08.029

such as in the case of a Broadcast News transcription domain, as proposed in Federico and Bertoldi (2004), Martins et al. (2010), Saykham, Chotimongkol, and Wutiwiwatchai (2010).

The main advantage in using documents available online to build the dynamic LMs relies on the broad scope of the information on the Internet, in terms of topic coverage (we could find documents regarding almost any topic or domain). However, the variety of topics is also the main weakness of using online information. Indeed, the sparseness of the available data is so high that the LMs tend to be poorly estimated. In an effort to solve this problem, several clustering algorithms have been proposed (Chen, Gauvain, Lamel, Adda, & Adda, 2001; Iyer & Ostendorf, 1999; Iyer, Ostendorf, & Rohlicek, 1994) to group together those elements that share some properties. An interesting idea to fuse the clustering approach with the exploitation of higher level information (not only the information directly provided by the speech recognizer, such as acoustic scores, or the most likely word sequence) is the application of an analysis to extract the semantic relationships between terms, or documents. In this sense, the work presented in Bellegarda (2000) and Bellegarda, Butzberger, Chow, Coccaro, and Naik (1996) proposes the use of Latent Semantic Analysis (LSA, Landauer, Foltz, & Laham, 1998), a tool used in the field of Information Retrieval that can discover semantic relationships between the terms that appear in different documents. A closely related approach consists of the application of Latent Dirichlet Allocation (LDA, Tam & Schultz, 2006), that establishes a probabilistic distribution over the different adaptation models.

It is also useful to consider the information provided by the speech recognizer itself, either to guide the query of the most relevant documents, or to cluster the documents into different groups that will eventually be considered as part of further adaptations. For instance, (Chen et al., 2001) uses the keywords identified by the recognizer to determine the documents to be used for the LM adaptation.

It has also been proven that certain word sequences can predict other sequences, even at a certain distance from the current one. The first sequences are then called *triggers*, since its presence 'triggers up' the likelihood of the so-called *triggered* sequences. That is, it is more likely that the triggered sequence appears after its trigger. (Lau, Rosenfeld, & Roukos, 1993; Rosenfeld, 1994; Rosenfeld & Huang, 1992) have implemented this trigger pair based approach, leading to an improvement in the performance on Large Vocabulary Continuous Speech Recognition systems adapted to different topics.

In Spoken Language Dialogue Systems (SLDS), in which there are several interconnected modules, each one performing a different task (speech recognition, language understanding, dialogue management, and so on), the number of information sources that could potentially be used to adapt the LMs increases. Indeed, we could take into consideration lexical and acoustic information (managed by the speech recognizer), but we could also use either semantic information (the content of the utterance that has been recognized, such as in Gruenstein, Wang, & Seneff (2005), Solsona, Fosler-Lussier, Kuo, Potamianos, & Zitouni (2002), and Visweswariah & Printz (2001)), or discourse or pragmatic information (which is related to the intentions of the user). From a more general point of view, the current research usually defines these dialogue-based approaches as *context-dependent adaptation* (such as in Fügen, Holzapfel, & Waibel (2004) and López-Cózar & Griol (2010)), or *state-dependent adaptation* (Popovici & Baggia, 1997; Riccardi & Gorin, 2000; Riccardi, Potamianos, & Narayanan, 1998). We could even unify all of these information sources, as (Raux, Mehta, Ramachandran, & Gupta, 2010) proposes, by using a statistical framework based on Bayesian Networks (BNs). We will see how our approach also tries to unify the knowledge of the speech recognizer (in terms of lexical features), the language understanding module (semantic features), and the dialogue manager (discourse or user-intention features), but in a rather simple and intuitive way.

As regards the ASR subprocedure or the stage **where the adapted models could be used**, they could be integrated at the decoding stage, as Justo and Torres (2007) proposes, or they could be used in a rescoring stage for improving the initial recognition hypotheses, as in López-Cózar and Callejas (2006), which rescores the word lattice generated during the first pass of the recognizer. Another possibility proposed in López-Cózar and Callejas (2008) and López-Cózar and Griol (2010) relies on using the adapted models for correcting recognition errors after each recognition step, instead of initiating a second recognition step. An accurate error recovery strategy might lead to an improvement in the dialogue efficiency, reducing the number of turns required to complete the actions.

Finally, there are different **adaptation strategies**, each of which suits the nature of each LM best (either *n*-grams, Context-Free Grammars (CFGs), and so on). Bellegarda (2004) identifies two main ways of adapting LMs: model interpolation, and constraint specification. We briefly present each of these strategies below.

1.1.1. Model interpolation

Model interpolation is the most known and widespread strategy of adapting LMs. Its basic idea, together with the definitions that we will use throughout this paper, is presented in Fig. 1.

Language model interpolation simply consists of generating a **context dependent** LM using the **content-specific** LMs generated with the adaptation data. This content-specific model could be related to any domain, application or situation to which we want to adapt the system. The context dependent component is then merged with a **background** LM at each point of the interaction when an adaptation is required, generating the **dynamic** LM that the speech recognizer will use. This way, the background, more general model, that is usually trained with more data (but less specific), keeps the likelihood of a certain amount of word sequences relatively unmodified, while the context dependent component, usually trained with few data, but more specific, and related to the situation (a topic, a speaker, etc.) which we want to adapt to, gives more relevance to those word sequences that better match that situation.

Whereas the background LM is always static (in the sense that it is trained once and remains unmodified during the interaction), the content-specific and the context dependent components of the model could be either static or dynamic, depending on whether the information used for estimating them remains the same, or is periodically updated. In our case, we will keep the content-specific LMs static, but we will estimate the context dependent LM on a turn basis. In that sense, this LM will also be dynamic.

The interpolation between LMs is simple to understand and to implement, and it perfectly suits *n*-grams, the most widespread

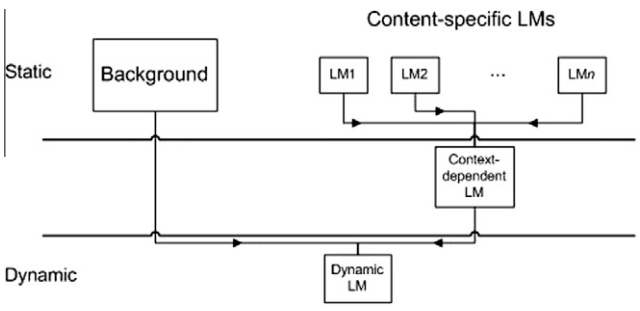

Fig. 1. Interpolation of language models.

approach to build LMs. This simplicity has made interpolation a common adaptation strategy nowadays, providing an important research background.

Looking at the level at which the interpolation is implemented, we could distinguish several approaches to adapt LMs via interpolation, among which the most widely used ones are the merging of LMs, the use of word and sentence caches, and the merging of the counts of the different word sequences.

In the *model merging* approach, the goal is to estimate the probability of observing each word sequence as a mixture between a background, static LM, and one or more content-specific models. As the amount of data to train those content-specific LMs is usually limited, the resulting LMs tend to be poorly estimated. Therefore, the system should obtain the most reliable interpolation weights, giving different relevances to the background LM and each specific component, depending on the conditions under which the adaptation takes place (what the system is adapting to, when the adaptation takes place, and so on).

The most common model merging approach carries out a *linear interpolation* between two or more LMs. Let us consider probabilistic language modelling, and let $P(w|h)$ be the probability to have word w given the previous sequence of words h (usually referred to as the *word history*). Then the probability P_I according to a linear interpolation between two models, a background LM P_B, and the context dependent component P_D (which in our case will include part of the dynamic behaviour), will be

$$P_I(w|h) = (1-\lambda)P_B(w|h) + \lambda P_D(w|h) \quad (1)$$

λ being the interpolation weight between both models, which has to fulfill the condition $0 \leqslant \lambda \leqslant 1$.

In the previous equation, there are two possibilities to include a dynamic behaviour in the LM. In the first one, the interpolation weight λ can be modified in accordance with the variable conditions of the interaction. In the second one, the component P_D itself can also be dynamic to better adapt to the current situation (for instance, switching between several domain-specific LMs). As we said before, in our case, we will consider that both the interpolation weight and the context dependent LM P_D are dynamic.

Depending on how the content-specific LMs, together with their interpolation weights, are estimated, we could find several approaches in the literature. The content-specific LMs can be estimated offline, training them with a certain amount of data. This data could be, for instance, the sentences allowed for each adaptation domain. However, if the number of sentences available to train the specific LMs is reduced, it is usually preferred instead to use the *classes* that each word belongs to (Gruenstein et al., 2005; López-Cózar & Callejas, 2006; Raux et al., 2010). These word classes can be related to the lexical or the morpho-syntactic class of each word, or they could be automatically defined, by means of a clustering algorithm that groups those words that share any relevant feature into the same class.

In order to make the approach more dependent on the current situation of the interaction (the topic, the speaker, the current dialogue turn, etc.), it has been proposed to build even the content-specific LMs using the information gathered up to the current interaction, in the terms of all the previous sequence of words. This approach, referred to as *dynamic cache modelling* (Iyer & Ostendorf, 1999; Jelinek, Merialdo, Roukos, & Strauss, 1991; Kuhn & de Mori, 1990), relies on the fact that, within a specific domain, if a certain word or word sequence has appeared, it is more likely to appear again in a short term. Instead of estimating a LM for the whole content of the cache, it has been proven (Lobacheva, 2000; Rosenfeld, 1994) that using only the content words related to the current topic yields better results, since function words (such as prepositions, articles, and so on) are expected to be common across all the topics.

As regards the interpolation weight, it is usually obtained during a cross-validation step, using data not seen in training (see, for instance, Klakow, 1998; Riccardi & Gorin, 2000; Riccardi et al., 1998; Tur & Stolcke, 2007; Wessel & Baader, 1999), or estimated via some optimization algorithm, such as Expectation Maximization (EM, (Martins et al., 2010)). These approaches keep the interpolation weight constant throughout all the interaction, and they offer good performance, since the weight is optimized to a certain context (either the current topic, the speaker, or the interaction conditions). However, any mismatch between the data used for estimating λ and the test data gives rise to a reduction in the performance figures. Therefore, it would be interesting to modify the interpolation weight according to the ongoing dialogue by trying to keep the model permanently updated with the most recent information gathered by the system. In this sense, we could use the history of each word (i.e. its previous sequence of words) to estimate that weight. This approach has been used, for instance, in Federico (1996), Kneser and Steinbiss (1993), Liu, Gales, and Woodland (2008), and Lobacheva (2000). A more complex scheme can be studied in Yamamoto, Hanazawa, Miki, and Shinoda (2010), in which the interpolation weights between LMs are estimated as a function of the identification of the most relevant words for each topic, those topics being obtained using Conditional Random Fields (CRFs).

Instead of interpolating the models themselves, another possibility relies on merging the frequencies of each word sequence. This approach is known as *count merging* (Federico & Bertoldi, 2004; Ljolje, Hindle, Riley, & Sproat, 2000; Lobacheva, 2000), and it is usually related to well-known adaptation approaches, such as Maximum a Posteriori (MAP) or Maximum Likelihood Linear Regression (MLLR). MAP has been successfully applied to LM adaptation (see, for instance, (Chen et al., 2001; Liu et al., 2008)), although it has been proven in Bacchiani, Riley, Roark, and Sproat (2006), Bacchiani and Roark (2003), and Hsu (2007) that the performance of a MAP-based adaptation system is similar to that achieved with linear interpolation, but requiring a greater computational effort.

1.1.2. Constraint specification

From a different point of view, constraint specification does not consider the dynamic information as different models to be merged somehow with the background, static LM. Instead, it proposes to model both the static and the adaptation LMs as a set of constraints that the dynamic LM needs to satisfy. Constraint specification methods are related to the Maximum Entropy (ME) principle, that has been successfully applied to LM adaptation in Lau et al. (1993) and Rosenfeld (1994). According to this principle, each information source (for instance, the LM specific of a certain domain) can be expressed as a set of constraints. These constraints (usually expressed as marginal distributions over the training data) are related to any relevant feature of the information source. They could be, for instance, the expectation of a certain word sequence (h,w). The resulting set of K linearly independent constraints defines a subspace of functions that are consistent with all the different information sources. This relationship can be expressed as

$$\sum_{\{(h,w)\}} f_k(h,w)P(h,w) = \alpha(\hat{h}_k\hat{w}_k) \quad (2)$$

$P(h,w)$ being a function related to the sequence of words (h,w) in the adapted model (for instance, an n-gram probability $P(w|h)$); $f_k(h,w)$, the set of constraint functions, and $\alpha(\hat{h}_k\hat{w}_k)$, the corresponding value of that function estimated from the data. The idea underlying the ME principle is that, once the constraint set is included, we do not need to assume any additional evidence about the data. Then the best solution is to choose the function that maximizes the entropy.

This function is usually estimated by solving the mentioned set of equations by means of Lagrange multipliers, giving a product of *exponential models* as a result:

$$P(h,w) = \prod_i \mu_i^{f_i(h,w)} \quad (3)$$

The parameters μ_i of this model are then estimated using the Generalized Iterative Scaling algorithm (GIS, Darroch & Ratcliff, 1972; Rosenfeld, 1994), that converges to the ME solution provided that the constraints expressed by the set of functions f_i are consistent.

Constraint specification approaches are also related to the Minimum Discrimination Information (MDI) criterion, in the sense that MDI-based strategies exploit exponential models. MDI approaches have been effectively used for LVCSR applications, as can be seen in Lecorvé et al. (2009) and Rao, Monkowski, and Roukos (1995).

Despite the apparent mismatch between both adaptation approaches (linear interpolation and constraint specification), there have been several efforts to unify them. An interesting idea relies on carrying out a *log-linear interpolation* of LMs, as proposed in Klakow (1998). It consists of optimizing the interpolation weights among the background LM, and one or more content-specific LMs, but at a logarithmic level instead of at a linear one, thus keeping the exponential model formalism from constraint specification.

1.2. Applications of LM adaptation

Language model adaptation has been successfully applied to several tasks, such as Large Vocabulary Continuous Speech Recognition (LVCSR), Broadcast News (BN) transcription, and Spoken Dialogue Systems (SDS). They have also been used as a scoring tool for retrieving information from large databases. For instance, Straková and Pecina (2010) uses LM adaptation to build several LMs, each of which is related to a different domain or topic. With these LMs they estimate the perplexity of a query, in order to retrieve the documents related to the topic that gets the best score.

In the field of LVCSR, as well as in BN transcription, researchers have applied the different approaches we mentioned above. The most common information sources are documents related to the topic that is addressed in the current interaction (Lecorvé et al., 2009; Shi et al., 2008). The topics can be known beforehand or they could be obtained in an unsupervised way by clustering the different words or sequences. The cluster criterion is usually the optimization of an appropriate distance between clusters (Bellegarda, 2000; Chen et al., 2001; Iyer & Ostendorf, 1999). A different *context-dependent* analysis arises when using the most recent information provided by the user of the system (that is, the recognition hypotheses of the previous interactions). The more broadly used adaptation methods are a cache of the last $\mathcal{N}$ words (usually only content words are considered), as proposed in Jelinek et al. (1991) and Kuhn and de Mori (1990), or by using trigger pairs (Lau et al., 1993; Rosenfeld, 1994).

As regards spoken dialogue systems, the greatest effort has been made in adapting the recognition to each dialogue turn, taking into account the information the user provides to the system, and the *state of the dialogue* (Popovici & Baggia, 1997; Riccardi & Gorin, 2000; Visweswariah & Printz, 2001). Traditionally, these adapted systems are defined by a set of states. The dialogue manager moves through the different states according to the information conveyed in the current user utterance, and the previous state of the system. Depending on that information, the system will carry out the most appropriate action on each state. In our case, we will refer to the *system status* as the set of variables and values of our application domain, that define the actions currently being addressed. We use this definition since we do not have explicit states in our dialogue management strategy.

In any case, the purpose of adapting LMs in dialogue systems is either to improve the recognition of the current sentence by adapting the LM to the most likely word sequence that the user can say at each dialogue turn (as in Popovici & Baggia (1997), Wessel & Baader (1999)), or to correct the recognition errors that may arise during a turn, by using higher-level knowledge (for instance, semantic knowledge, or information related to the application domain, such as in López-Cózar & Griol (2010)). The most common adaptation approach is a linear interpolation of LMs, estimating the interpolation weights using a validation set. Other approaches, such as rule-based adaptation (Fügen et al., 2004), or a fusion between interpolation and ME strategies (Visweswariah & Printz, 2001), have been successfully applied for controlling a household robot and for a travel information domain, respectively.

1.3. This work

We propose to adapt the LM used in a speech recognition module that is part of a spoken dialogue system. The underlying adaptation approach is a linear interpolation between a background LM and several content-specific LMs. All of these models are estimated offline (and therefore they are static in accordance to our previous definitions), but the selection of the most suitable models to estimate the context dependent LM takes place at each dialogue turn. Therefore, our context dependent LM is also dynamic. The interpolation weights between the different components of the context dependent LM are also obtained dynamically on a turn basis, using the context of the ongoing dialogue, either the semantics of the utterance (which we will refer to as the **dialogue concepts** that the understanding module extracts from the recognition hypothesis), or the discourse or intention content (which we will refer to as the **dialogue goals**, or actions that the user wants to carry out). We will refer to both concepts and goals as **dialogue elements**.

One of our major claims is that the system itself can feedback the grammar generation engine enough information to obtain accurate interpolation weights that depend on the current recognition result, as well as on the system's contextual information (its knowledge about the dialogue up to the current interaction). This way, the interpolation weights will depend on certain values obtained by the system, such as confidence measures, probabilities, and so on.

We propose several approaches to obtain the LMs to be interpolated: they could be related to each single piece of information (either semantic or discourse based), or they could be related to a group of them. In this regard we will assess several grouping strategies, both supervised (by using expert knowledge) and unsupervised (by carrying out a semi-automatic semantic analysis). Finally, in our evaluation we will measure the improvement achieved not only at the recognition level, but also at the understanding and dialogue management level, and even at the level of the user's experience when interacting with the system. That is, we will determine to what extent an improvement in the LMs leads to an improvement in several modules of the dialogue system.

The rest of the paper is organized as follows. Our baseline Spoken Dialogue System and the main features of its modules are detailed in Section 2. We then explain in Section 3 our different approaches to dynamically modify the LMs of the speech recognizer using the information provided by the language understanding module or the dialogue manager. The results of the evaluation we have carried out are presented in Section 4. Finally, in Section 5 we discuss our approaches, and in Section 6 we present our guidelines for future work.

2. Spoken Dialogue System

We use a user-independent spoken dialogue system (SDS) previously developed in our lab, to control different household devices, such as a TV, a Hi-Fi equipment, a vacuum cleaner, and so on. We have evaluated the performance of the prototype when including the new dynamic LMs, for the task of controlling a Hi-Fi device using speech.

Fig. 2 shows a block diagram of our conversational interface. The system consists of an automatic speech recognition module (ASR), which translates the audio signal into a text hypothesis of what the user has said; a natural language understanding module (NLU), that extracts the semantics of the user's utterance; the dialogue manager (DM), which makes use of the semantic information, together with the information gathered during previous dialogues, to determine the actions that the user wants to carry out, and to provide the user with feedback regarding the ongoing dialogue turn; the context manager (CM), which contains the information of the previous interactions; an execution module, which translates the actions to be carried out into IR commands to be sent to the Hi-Fi equipment; the natural response generator module (NRG), which makes use of the semantic information provided by the dialogue manager to generate a text output, and a text-to-speech module (TTS), that synthesizes the message back to the user.

To improve the behaviour of the system when interacting with different users, we have also developed an automatic adaptation approach based on the definition and updating of user profiles. These structures contain information related to the speakers, such as their identity, degree of expertise, and their preferences as regards their interactions with the system. To include this user-related information, we have included a speaker identification module, and a profile manager. We group both modules into a User Information Manager (UIM, Lucas-Cuesta, Fernández, Salazar, Ferreiros, & San-Segundo, 2009; Lucas-Cuesta, Fernández-Martınez, Dragos, Lutfi, & Ferreiros, 2011).

Our modification, presented in Section 3, establishes a new module as a feedback loop between the DM, the NLU, and the ASR modules. This new element, the Dynamic LM Generator, or Dynamic Grammar Generator (DGG), will take into account the information provided by the users in their previous utterances to modify dynamically the LMs that the ASR makes use of. We will use this dynamic LM to recognize the current utterance by exploiting the 'contextuality' of human dialogues (that is, the fact that speakers usually tend to make implicit references to ideas they had previously mentioned). In other words, we will use the *contextual information* held by the system in an effort to improve the recognition of the current utterance.

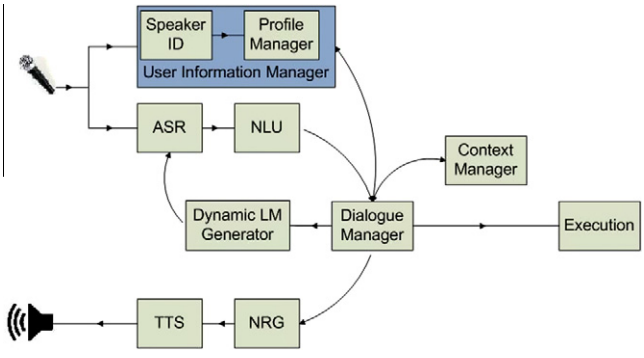

Fig. 2. Block diagram of the spoken dialogue system.

We have designed a multi-goal, mixed-initiative spoken dialogue system based on the use of Bayesian Networks (BN) as the basis of our Dialogue Manager. This approach can exploit the causal relationships between the semantics of an utterance (the dialogue concepts) and the intention of the speaker (the dialogue goals).

We have defined a set of 58 different concepts, divided into *parameters* (16) that can be set up (e.g. the volume of the Hi-Fi device), *values* (20) that the different parameters can take, and *actions* (22) to be carried out (e.g. to increase the volume). We have also defined 15 different goals according to the available functionality of the Hi-Fi device. For instance, a modification in the volume setting. Both dialogue elements have been defined as binary variables (i.e. a concept or a goal is *present* only if it has been observed in the sentence, or positively inferred by the BN).

The BN performs the dialogue management by means of two Bayesian inference algorithms, which can infer the actions that the user wants to perform (i.e. the dialogue goals), and the semantic information, or dialogue concepts, that are needed to achieve those actions, whether available or not (Fernández et al., 2005). The *forward inference* makes use of the concepts referred by the user, and those retrieved from the Context Manager, for inferring the dialogue goals (the actions that the user wants to fulfill). This algorithm estimates the posterior probability of each goal g_i given its available *evidence* e_{g_i}, that is, the presence or absence of each concept, modulated by their confidence score. By comparing the resulting probabilities with several thresholds θ_i (that are set to 0.5, given that the dialogue elements are binary variables), the DM decides whether a goal is present or absent, according to the intention of the user.

After the forward inference process, the DM assumes the inferred goals as a new evidence. Then, and by using the parsed concepts and the already inferred goals, it applies a *backward inference* to estimate the posterior probabilities of the different dialogue concepts c_i given their evidence e_{c_i}. The DM will decide whether a concept is needed or not by comparing the posterior probability of that concept against a predefined threshold. A further analysis of the results of this inference, and the available information (which concepts the user has referred to during this turn, and those ones previously addressed), allows the system to classify each concept as needed, optional, or unnecessary in order to define the most suitable dialogue strategy (Fernández et al., 2005). Once the system has determined which concepts are necessary to carry out an inferred goal, it has to check whether all of these concepts are already available. In this case, the system will send the corresponding IR commands to the Hi-Fi equipment. Otherwise, the system tries to recover the *missing concepts* required to complete the dialogue, using the *Context Manager*.

The mission of the CM consists of solving any lack of information that may arise during the dialogue, using different contextual information handling strategies. It consists of five structures, which can be classified according to the recentness of their contents. The *system status* (the short-term one) stores the current values of the Hi-Fi functionalities (CD track, cassette, volume, and so on). The *task model* is a semantic frame that contains all the information needed to meet a specific dialogue goal. The *application domain model* is made up of that information specific to the application (in our case, the features of the Hi-Fi system, such as the number of CDs, or the number of tracks of a particular CD) The *dialogue history* (a mid-term memory) contains the concepts referred to by the user since the beginning of the current dialogue. An update and attenuation mechanism is applied in such a way that the relevance of this information is permanently re-estimated, coherently to the current state of the dialogue. This mechanism lowers the relevance of those concepts that are no longer referred to by the user, and strengthen those ones more frequently

addressed. If the information stored becomes out of date (that is, if its relevance falls below a certain value), it is disregarded. The last structure of the CM, the *user profile* (the long-term one), stores information of each user since his or her first dialogue.

The CM works as follows. When the DM has to recover a missing concept, it first checks the system status. If it contains such a concept, the system recovers it and carries out the appropriate action, thus finishing the current dialogue. Otherwise, the DM checks the dialogue history. If the system is unable to retrieve the required concepts, the DM finally checks the user profile, that may suggest one or several concepts, depending on its knowledge of the preferences and the privileges of the identified user. If the system is still unable to retrieve a concept using any of the above strategies, it will request the user to provide the missing concepts, initiating a new dialogue turn.

Summarizing, the combined use of the different information resources allows the DM to improve its performance by conducting more efficient dialogues, reducing their number of turns, and trying to reuse any useful information the users may provide during both their ongoing and previous interactions. In the next section we will see how the system can effectively exploit this information to adapt the LMs at each dialogue turn, trying to better recognize what the user has said in the ongoing turn.

3. Dynamic language model generation

In this section we will present the motivation of our work, as well as the different approaches we have developed for the dynamic adaptation of language models based on dialogue-dependent information.

3.1. Motivation

The generation and use of a dynamic LM requires the definition of the temporal aspects on which the LM should depend. These aspects will be closely related to both the application domain in which they will be applied, and the specific characteristics of each user of the system. Therefore, the dynamic generation of LMs is related to the interactions allowed between the user and the system, the phraseology allowed by the system, and the preferences of the users when interacting with the system.

From the point of view of the characteristics of the system (which interactions and phraseology are allowed), the context dependent LM will be determined in turn by the recognition hypothesis, by the semantics of that hypothesis, and by the available actions that could be triggered by these semantics. In other words, the information we will take into account in order to modify dynamically the LM will consist of three knowledge layers: a lexical layer (the vocabulary of the system), a semantic layer (the dialogue concepts that we define) and an intention or action layer, represented by the dialogue goals.

We will thus adapt the LM for the recognition of the current user turn by considering all the information provided by the user up to the preceding one, which has been stored in the Context Manager.

By using these dialogue elements, our approach is able to cover the requirements that any information source must fulfill in order to become part of the knowledge of a Language Model (Ueberla, 1994). Firstly, the recognition system keeps the vocabulary constrained. Secondly, the dialogue elements can be used frequently, since the information they convey is stored and updated on a turn basis, giving us the chance to exploit that information to adapt the LMs at each dialogue turn. Additionally, the computational effort is kept under control, provided that our LM adaptation approach will imply a linear interpolation among a reduced set of models, as is shown later in this section. Moreover, the interpolation weights will be dependent on different metrics estimated by the system at dialogue time, thus avoiding additional requests, either to the user or to more complex sources of information (for instance, a Web request to an information retrieval system). Finally, our approach relies on contextual information automatically collected by the system throughout the user–system interaction, thus avoiding additional system requests.

To sum up, the use of dialogue elements as sources of information for adapting LMs is perfectly feasible. In the rest of this section we will detail the main issues we should tackle to exploit this information effectively.

3.2. Considerations

To estimate an LM we need enough data to model accurately the grammar allowed by our system. If we want to model separately the contribution of each source of information available (words, concepts and goals) we have to solve three problems: (a) how many content-specific components should we define to cover all the characteristics of our system; (b) how much data should we use to train each of these components; and (c) how should we select the different components at each dialogue turn, in order to better adapt to the current context of the dialogue (the current speaker, his or her previous interactions, etc.).

To answer the first question, we propose the management of an LM dependent on each of the aforementioned knowledge layers: a component dependent on dialogue concepts, and another one dependent on goals. More specifically, we will keep a background LM, trained by using a database with more sentences. We will combine this background model with the concept-dependent and/or the goal-dependent models, trained on more specific data, to estimate the dynamic LM at each dialogue turn.

Either we estimate the context dependent LM using concepts, goals, or both, we will split the dialogue element space into different subsets. Each of these subsets will have an associated LM. This way, depending on the information provided by the user during his or her previous interactions, the system can select those LMs more closely related to that information, thus adapting to the current state of the dialogue.

One problem that arises when studying the distribution of dialogue elements in the training sentences is that users tend to refer to several concepts and/or goals in a single utterance. The main statistical parameters of the distribution of dialogue elements per utterance are shown in Table 1.

As we could expect, the sentences in our training corpus refer to several dialogue elements. Keep in mind that our system allows multi-goal interactions, such as in the sentence *Switch the Hi-Fi on, play CD 3, and raise the volume*, in which the user makes reference to three different goals. For this reason, we finally use each sentence to estimate the LMs associated to those dialogue elements (either concepts or goals) that sentence makes reference to. This approach will also imply that the number of sentences to estimate the LM related to a given dialogue element is higher than the number of sentences that makes reference only to that dialogue element. Therefore, the LMs will be estimated with a larger amount of data.

Table 1
Statistical parameters of dialogue element distribution in the training sentences.

	μ	σ
Number of concepts	4.30	2.02
Number of goals	2.25	1.21

Independently of the number of models to be considered, we will combine them by using a linear interpolation approach. First, we will generate offline the LMs associated to the different dialogue elements. Then, at each dialogue, the system will estimate the interpolation weight for each model, as well as the contribution of the background LM, to generate the model that the ASR module will use. In other words, the interpolation weights will depend on the current dialogue. Therefore, if we rewrite the equation for interpolating LMs (1) including the time dependency of the interpolation weights, we could estimate the probability of obtaining a word w given its history h at a time step t as

$$p_I(w|h) = (1 - \lambda_D(t))p_B(w|h) + \lambda_D(t)p_D(w|h) \quad (4)$$

$\lambda_D(t)$ being the dynamic interpolation weight between the background LM p_B and the context dependent component, p_D, dynamically built on each dialogue turn using the LMs related to the dialogue elements addressed by the user.

We present in the next sections our different approaches to obtain the context dependent LM p_D, as well as the dialogue-dependent dynamic interpolation weight λ_D.

3.3. LM based on isolated elements

In our first approach (Lucas-Cuesta, Fernández, & Ferreiros, 2009), we will build a language model for each dialogue element (either concepts, goals, or both). To do so, we first estimate each LM by using all the sentences in which a certain element appears, as was explained before. This estimation takes place off-line, before the user–system interaction.

Once the models for the different dialogue elements have been estimated, we load them into the SDS. Then the dynamic LM estimation is carried out online at each dialogue turn. Once a sentence has been recognized, and the DM has performed both inference mechanisms (that is, once the DM has inferred the user goals using the available concepts), the content-dependent LMs are selected by analyzing the posterior probabilities of the corresponding elements, which were estimated by the BN that carries out the dialogue management (see Section 2). Instead of using the LMs related to all of the elements referred to by the user, the system will select those LM associated to elements that are more relevant for the current turn. That is, those whose posterior probabilities are above several *relevance thresholds*, namely Φ_C for concepts, and Φ_G for goals. These thresholds will be estimated using a validation database.

It is important to emphasize that these thresholds do not have to be the same as those that the DM considers to decide whether a goal is active (during the forward inference) or whether a concept should be present (during the backward inference). During the estimation of these thresholds we found that Φ_C tends to take values below the dialogue thresholds θ_i, whereas the optimum value of Φ_G is more similar or slightly greater than the dialogue thresholds.

Once the system has selected the LMs to generate the context dependent LM p_D (i.e. those corresponding to elements whose posterior probabilities rise above the appropriate relevance threshold), it has to obtain the interpolation weights between all the models. These weights are estimated online as a function of the posterior probabilities of the different dialogue elements.

The last step consists of interpolating the context dependent LM with the background one, p_S, to generate the dynamic LM that the ASR will finally use. That is, we have to obtain the interpolation weight λ_D between both LMs. We obtain this weight at a validation stage.

We have considered three different situations: using only intention dependent information (goals), using only semantic information (concepts) or merging both sources of information into a single context dependent LM. We detail each of these approaches below.

3.3.1. Concept-based LM interpolation

First of all we have interpolated only concept-based models with the background LM. This way, we can consider up to 58 different LMs, one for each dialogue concept, to build the context dependent LM. We will denote this by making $p_D(w|h) = p_C(w|h)$. The relevance threshold Φ_C introduced before will be used to select online (at each dialogue turn) which models will be used.

To obtain the model p_C, the system will apply a linear interpolation between the LMs associated to those concepts whose posterior probability (according to the backward inference procedure) rises above the relevance threshold Φ_C. If we denote the set of 58 dialogue concepts by $\mathcal{C}$; by $p_{c_i}(w|h)$, the LM related to concept c_i; by $p_b(c_i|e_{c_i})$, the posterior probability of the concept c_i given its evidence e_{c_i} according to the backward inference, and by $\hat{\mathcal{C}} = \{c_i \in \mathcal{C}; p_b(c_i|e_{c_i}) > \Phi_C\}$, the subset of concepts whose posterior probability is above the relevance threshold at a given dialogue turn, then the context dependent LM will be estimated as

$$p_C(w|h) = \sum_{\forall \hat{c}_i \in \hat{\mathcal{C}}} w_{\hat{c}_i} p_{\hat{c}_i}(w|h) \quad (5)$$

The interpolation weights $w_{\hat{c}_i}$ associated to each LM considered, are also obtained automatically at each dialogue turn, as a function of the posterior probabilities obtained by the DM. This way, the higher probability the system obtains for a given concept, the higher the interpolation weight of that model will be. To ensure that we are working with true probabilities we constrain the interpolation weights to sum 1. Therefore we include the summation of the posterior probabilities of all the concepts considered as a normalization constant. Eq. (5) then becomes

$$p_C(w|h) = \frac{1}{\sum_{\forall \hat{c}_i \in \hat{\mathcal{C}}} p_b(\hat{c}_i|e_{\hat{c}_i})} \sum_{\forall \hat{c}_i \in \hat{\mathcal{C}}} [p_b(\hat{c}_i|e_{\hat{c}_i}) p_{\hat{c}_i}(w|h)] \quad (6)$$

3.3.2. Goal-based LM interpolation

In a similar way, we use only the information related to the discourse or intention level. That is, we interpolate the LMs associated to the different dialogue goals. We consider up to 15 different goal-based LMs. In this approach, the context dependent LM in Eq. (4) becomes $p_D(w|h) = p_G(w|h)$, the G standing for the goal-dependent models.

We have used the posterior probabilities of the dialogue goals, obtained during the forward inference, to decide which models should be interpolated, together with the interpolation weights between these LMs. The decision was made by comparing the probabilities of each goal against the relevance threshold Φ_G, and considering only the LM related to those goals which posterior probability took a value above that threshold.

Let $\mathcal{G}$ be the set of 15 dialogue goals; $p_{g_i}(w|h)$ be the LM related to goal g_i; $p_f(g_i|e_{g_i})$ be the posterior probability that the goal g_i is present in the utterance under analysis, given its evidence e_{g_i}, according to the forward inference. We now denote the subset of dialogue goals whose posterior probabilities are above Φ_G as $\hat{\mathcal{G}} = \{g_i \in \mathcal{G}; p_f(g_i|e_{g_i}) > \Phi_G\}$. Using these definitions, and taking also into account the constraint of the interpolation weights (to fall between 0 and 1, and to sum 1), the context dependent LM based on goals is obtained as:

$$p_G(w|h) = \frac{1}{\sum_{\forall \hat{g}_i \in \hat{\mathcal{G}}} p_f(\hat{g}_i|e_{\hat{g}_i})} \sum_{\forall \hat{g}_i \in \hat{\mathcal{G}}} [p_f(\hat{g}_i|e_{\hat{g}_i}) p_{\hat{g}_i}(w|h)] \quad (7)$$

Again, by estimating the interpolation weights as a function of the posterior probabilities we give more relevance to the goals best

scored by the Dialogue Manager, making the LM related to those goals to have more importance in the context dependent LM.

3.3.3. Concept and goal combination

Finally, we also considered combining the isolated language models related to both dialogue elements (concepts and goals). In this approach, the context dependent LM will also be an interpolation between a concept-specific LM, and a goal-dependent one, as depicted in Fig. 3.

In order to include the dependency on both dialogue elements, we define an additional interpolation step, in which we obtain p_D as a linear interpolation of a concept-dependent model, p_C, and a goal-dependent one, p_G, as previously defined. The estimation of the context dependent LM will thus become

$$p_D(w|h) = \frac{1}{w_C + w_G}(w_C p_C(w|h) + w_G p_G(w|h)) \tag{8}$$

where p_C and p_G are the interpolated LMs presented in Eqs. (6) and (7) respectively, and w_C, w_G are the weights assigned to each of these models.

Instead of estimating both interpolation weights by means of an estimation approach (such as Expectation Maximization) or during a validation step, we obtain them as a function of the number of dialogue elements considered, their posterior probabilities, and the relevance thresholds defined in the previous sections. We will explain the expression of both interpolation weights with the help of Fig. 4. For the sake of simplicity we will only consider dialogue concepts; the expressions for goals can be obtained similarly.

Let us suppose a certain dialogue turn in which 5 concepts have been extracted, with posterior probabilities $p_b(c_i|e_{c_i})$, obtained by the backward inference. In the example, the relevance threshold Φ_C has been set to 0.5. Consequently, we can see that the LMs associated to the concepts c_2 and c_3 will not be considered for the adaptation of the LM, since their posterior probabilities are not above Φ_C. Therefore, the set of concepts to be used is $\hat{C} = \{c_1, c_4, c_5\}$. Let us define that number of concepts (i.e. the cardinality of $\hat{C}$) as $\hat{N}_C = |\hat{C}|$.

Now let us define the *Amount of Presence* of a concept $\hat{c}_i$ as the difference between its posterior probability, and the relevance threshold:

$$AP(\hat{c}_i) = p_b(\hat{c}_i|e_{\hat{c}_i}) - \Phi_C \tag{9}$$

The values of $AP(\hat{c}_i)$ will always fall between 0 (when the posterior probability of $\hat{c}_i$ is equal to Φ_C), and $(1 - \Phi_C)$ (when $p_b(\hat{c}_i|e_{\hat{c}_i}) = 1$).

Instead of considering the amount of presence of each concept (which is an absolute difference between their posterior probabilities and the threshold), we want to obtain the interpolation weights as a function of a relative comparison between the amount of presence of the different dialogue elements. Therefore we will

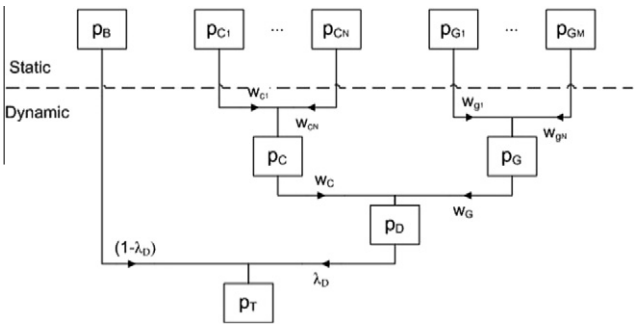

Fig. 3. Language model dependencies for building the dynamic LM.

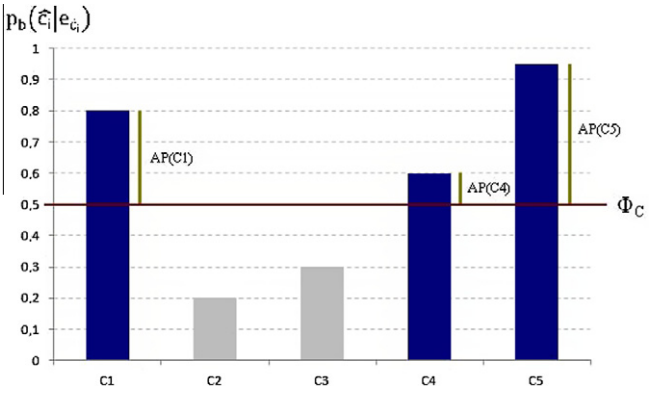

Fig. 4. Example of the obtention of the interpolation weight w_C.

consider the accumulated amount of presence for all the considered concepts, defined as the sum of the individual amounts of presence:

$$AP(\hat{C}) = \sum_{\forall \hat{c}_i \in \hat{C}} AP(\hat{c}_i) \tag{10}$$

The boundaries of the accumulated amount of presence are 0 (when the posterior probabilities of *all* the concepts in $\hat{C}$ are equal to Φ_C), and $(1 - \Phi_C)\hat{N}_C$ (when all of these probabilities are equal to 1).

Taking the previous fact into account, we could obtain the interpolation weight w_C as the quotient between the accumulated amount of presence for the concepts extracted in the current turn, and the maximum value allowed for that accumulated amount of presence: $w_C = AP(\hat{C}) / \max AP(\hat{C})$.

If we substitute the expressions of the accumulated amount of presence, we obtain the expressions of the interpolation weight w_C (and the corresponding one for dialogue goals w_G, that can be easily derived):

$$\begin{aligned} w_C &= \frac{1}{(1-\Phi_C)\hat{N}_C} \sum_{\forall \hat{c}_i \in \hat{C}} [p_b(\hat{c}_i|e_{\hat{c}_i}) - \Phi_C] \\ w_G &= \frac{1}{(1-\Phi_G)\hat{N}_G} \sum_{\forall \hat{g}_i \in \hat{G}} [p_f(\hat{g}_i|e_{\hat{g}_i}) - \Phi_G] \end{aligned} \tag{11}$$

As we stated previously, Φ_C and Φ_G are the respective thresholds for considering the concept or goal LM to be interpolated; $\hat{N}_C$, $\hat{N}_G$ are the number of concepts and goals (extracted or inferred from the input utterance) whose posterior probabilities are above the corresponding threshold (they are the cardinality of the subsets $\hat{C}$ and $\hat{G}$, defined previously), and $p_b(c_i|e_{c_i}), p_f(g_i|e_{g_i})$, are the posterior probabilities resulting from both inference processes of each concept c_i and each goal g_i of the utterance given their respective evidences, e_{c_i} and e_{g_i}.

Using these expressions, we give more relevance to those dialogue elements with higher posterior probabilities, also assuring that w_C and w_G take always a value between 0 and 1, whatever the posterior probabilities are.

3.4. LM based on the clustering of dialogue elements

Despite its soundness and benefits, having a single LM for each dialogue element has two main weaknesses. On the one hand, the system has to consider a large number of LM into consideration at each interpolation step (that is, at each dialogue turn). On the other hand, the division of our database into different subsets may cause the training sentences to be sparsed over all the LMs, which implies that several of these models could be estimated with a

drastically reduced number of sentences. This may lead to a poor estimation of the corresponding models.

In an effort to solve both limitations, we propose to apply a clustering strategy over the dialogue elements (either concepts, goals or both), building several groups of elements. After grouping the elements, a model is estimated for each group. At each dialogue turn the system will decide which group-based LMs have to be used, and it will calculate the interpolation weights between them.

The idea underlying the grouping of dialogue elements is to reach a tradeoff between having a higher number of more specific models (**specificity**), but trained with a more reduced number of sentences, and having LMs trained with more data (**robustness**), but also more generalistic.

This clustering of elements can solve the weaknesses of the previous approach. A group-based approach could reduce the number of models to be considered during the interpolation, and may lead to a more robust estimation of the LMs, provided that each model will be estimated using more data. The minimum number of sentences to train the LM related to a cluster will be, in the worst case, equal to the greatest number of sentences related to the elements that belong to that cluster.

We have proposed and evaluated three different approaches to cluster dialogue elements. The simplest one consists of using expert knowledge to generate the groups. It will be presented in Section 3.4.1. We have also applied two strategies of semantic clustering, detailed in Section 3.4.2. The strategy to estimate the interpolation weights will be explained in Section 3.4.4.

3.4.1. Expert clustering

Our first approach consisted of a classification of the dialogue elements in accordance with the application domain (in our case, the control of a Hi-Fi device using speech). This expert-based (or domain-based) clustering takes into account the available functionality of the Hi-Fi, namely, controlling the 3 CD player, the 2 cassette player (one of which includes the recording function), the radio tuner, and the amplifier (made up of the volume and the equalization functions). We also define another *Rest* group, in which we classify the remaining elements, which do not fall into function-specific groups, or elements that could be shared by two or more groups.

We apply an additional restriction to build the groups. The clusters to be generated must form a partition of the dialogue element space (either of concepts or goals). That is, every dialogue element will be considered and classified in a cluster, and no dialogue elements could be classified into more than one group. The number of dialogue elements in each of the expert groups is presented in Table 2.

As a second evaluation of this clustering strategy, we did not consider the *Rest* group, since too many and rather heterogeneous dialogue elements were classified into it, so the use of that group may lead to an interpolation between two generalistic LMs (the background one and that related to the *Rest* cluster).

We only considered this approach as an initial assessment of a group-dependent dynamic LM generation. Despite this method is easy to implement, it lacks a formal criterion to classify the different dialogue elements. It also makes the classification decision dependent on the application domain. To overcome both limitations we propose the application of an automatic clustering based on semantic criteria to build the clusters of dialogue elements.

Table 2
Number of dialogue elements of each expert cluster.

Group	No. of concepts	No. of goals
CD	10	3
Cassette	6	2
Radio	9	2
Amplifier	9	2
Rest	24	6

3.4.2. Semi-automatic semantic hierarchical clustering

To overcome the limitations of the previous clustering strategy, we propose a clustering algorithm based on the application of a tool that could emphasize the relationships between the dialogue elements. We will use an adaptation of the Latent Semantic Analysis paradigm (LSA, Landauer et al., 1998) to carry out the clustering of dialogue elements.

In classic LSA, the first step consists of building a co-occurrence matrix W, its rows being a set of documents under analysis, and its columns, the words among which the semantic relationships are to be discovered (Bellegarda, 2000; Bellegarda et al., 1996). As we want the semantic patterns between dialogue elements to arise, we propose to build the matrix using the frequency that each concept and/or goal appears on each labelled sentence in our database. This way, the rows of our co-occurrence matrix will be the different sentences, whereas each column will reflect the occurrence of each dialogue element in that sentence.

The next step of the algorithm consists of applying a Singular Value Decomposition (SVD) over this matrix. This transformation allows us to represent the co-occurrence matrix as the product of three matrices: $W = USV^T$, where U and V are column-orthonormal matrices, and S is a diagonal matrix of singular values. The goal of SVD is to discover a projected space in which the semantic relationships arise. This projected space usually keeps only the highest k singular values of S, that are sorted in a decreasing order, making the rest of values equal to 0. This way we could obtain an approximation of the co-occurrence matrix $\hat{W}$ using only these k singular values.

This projection on a space of reduced dimension allows the relationships between the dialogue elements to be better inferred, in terms of some distance metric. We propose to use the Pearson r correlation coefficient as this distance, since that correlation gives us a good hint of the relationships between the elements. That is, two or more elements highly correlated between themselves imply that they tend to appear together in a sentence, which indicates that they could potentially belong to the same cluster.

This semi-automatic approach based on LSA, together with the Pearson correlation distance, tends to cluster those concepts or goals with a strong semantic relationship between them. For instance, a cluster could include the three dialogue concepts related to the volume: the volume parameter itself (VOLUME_PARAMETER), its value (VOLUME_VALUE), and the action to be carried out (VOLUME_ACTION) all together.

The different values k for the dimension of the projected space (that is, the number of the highest eigenvalues chosen to re-estimate the co-occurrence matrix) can obtain different estimations for the correlation between dialogue elements. When this number is high, the estimation is more accurate, but the relationships among dialogue elements are more strict. In this sense, the result of the clustering is a large number of clusters, each of which has few dialogue elements. As k decreases, the estimation of the correlation becomes less precise, but more relationships among dialogue elements arise. Therefore, by varying the dimension of the projected space, we can establish a tree-like structure of clusters. We thus define a hierarchical structure by means of a bottom-up strategy, from the isolated dialogue elements, up to a top leaf, that could be made up of a single cluster with all the dialogue elements. An example of a hypothetical tree of clusters that could generate our LSA-based approach can be seen in Fig. 5.

Our LSA-based approach yields a total number of clusters of 100 (when using dialogue concepts), 25 (when considering only dialogue goals), and 116 (when clustering both elements together).

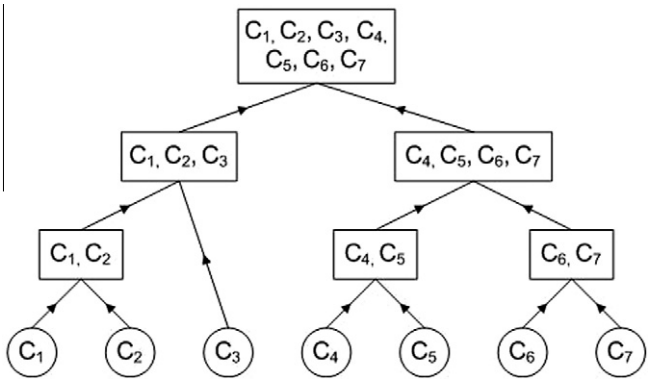

Fig. 5. Example of a cluster tree for dialogue concepts.

However, we do not consider the cluster that contains all of the dialogue elements, because the LM related to that cluster is trained with all of the sentences of the database, giving a too general LM as a result, close to the background one.

We could consider keeping the full hierarchy of clusters, allowing a given dialogue element (and, therefore, all the sentences that make reference to that element) the possibility of contributing to more than one LM. This approach has one main drawback. If we consider a full hierarchy, we may estimate more LMs than the number of dialogue elements. In our case, for instance, we should consider 25 LMs for the goal-based approach, whereas the number of dialogue goals is 15. Therefore, some criteria to select the most feasible clusters are needed. The underlying idea, as presented before, is to reach a reasonable tradeoff between the specificity of the resulting LMs (the lower the number of elements that belong to each cluster, the more specific the associated LM will be), and their robustness (the greater the number of elements in each cluster, the greater the number of sentences to train the associated LM).

In these conditions, a method to prune the cluster tree is needed. This pruning strategy is carried out by taking into account the data we use to estimate the LM associated to each cluster. Our pruning strategy discards the clusters made up of the dialogue elements that appear in a number of sentences too reduced to train the related LM accurately. On a first approach to prune the cluster tree, we establish that the number of sentences that trains the model related to a cluster should be, at least, a given percentage of the number of sentences that train the model of the parent cluster in the hierarchy tree. We will determine the optimum percentage during a cross-validation step.

As the number of clusters after this first pruning was still relatively large, we decided to include another, more restrictive condition: the system will not consider any cluster whose model should be trained with a number of sentences below a certain percentage of the total number of sentences in our database. As with the previous pruning, the final percentage was estimated during a cross-validation step.

Finally, we decided to study two different strategies, related to the number of layers in the cluster hierarchy that we keep. On the first one, which we will refer to as *single-level approach*, we force each dialogue element to belong to one and only one cluster. However, the reduction in the number of clusters with this strategy was hard. Consequently, the LMs related to the resulting clusters could be too generalistic. In an effort to keep the number of clusters constrained, but also keeping the specificity of the LMs, we apply a less restrictive strategy, which we call the *multi-level approach*. In this approach, we keep several levels of the cluster tree, allowing a dialogue element to be part of several clusters. Fig. 6 shows an example of the cluster selection following both strategies, applied to the cluster tree presented in Fig. 5.

As we said, the *single-level approach* implies that every dialogue element is taken into account, and each one of them will only be considered as belonging to a single cluster. By applying this restriction, we assure that the number of clusters is low, which implies that the LM associated to each cluster are trained with a larger number of sentences than in the isolated approach (3.3), thus making them more robust. In the example shown in Fig. 6(a), we can see that every concept belongs to one and only one cluster. Following this strategy, the system will consider 10 different clusters, when using only concepts, and 4 clusters for the goal-based clustering.

In the *multi-level approach*, we allow the dialogue elements to belong to more than one cluster (and, therefore, to be considered to estimate more than one LM). As the example of Fig. 6(b) shows, the concept c_4 belongs to three different clusters (a cluster made up only of c_4 itself, a cluster with c_4 and c_5, and a third one which contains 4 concepts).

In any case, we have also considered the full hierarchy of clusters for our initial experiments. Table 3 shows the number of clusters (and, therefore, the number of LMs estimated) for our semi-automatic semantic hierarchical clustering, either we use the single-level approach or the multi-level one. In the latter case, we also distinguish whether we apply a pruning strategy or not, and which pruning strategy (the initial or the more restrictive one).

As we can see, the conditions of the cluster selection strategy hardly affect the number of LMs to be estimated when considering only dialogue goals (14 or 9 vs. 15, when estimating a model for each goal). However, we can achieve a significant reduction in the number of LMs in the case of considering only concepts (we estimate less than half the number of concepts, for the initial pruning, and only a quarter of the total number of models of the hierarchy, for the restrictive one), and when estimating LMs related to clusters that group both dialogue elements together. In any case, the most important reduction in terms of LMs to be estimated is achieved when using the single-level approach to select clusters (we consider about one third of LMs for goal-based LM, and about one sixth of LMs for concept-based modelling).

3.4.3. Perplexity-based automatic hierarchical clustering

We have proposed two different 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.

Let us suppose a set of labelled 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.

As we know, 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 $H(A)$ the entropy of that model. In other words, the entropy of the LM A can be obtained as $H(A) = log_2 pp_A$.

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

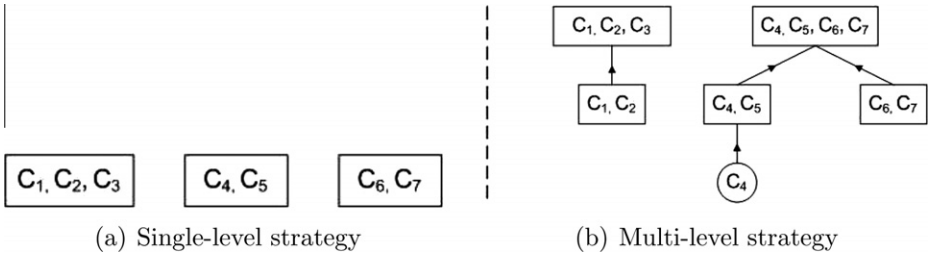

Fig. 6. Clusters chosen with the two selection strategies.

Table 3
Number of clusters considered for each LSA-based grouping strategy.

Clustered elements	Single-level	Multi-level		
		No pruning	Initial pruning	Restrictive pruning
Concepts	10	100	25	16
Goals	4	25	14	9
Concepts and goals	–	116	38	21

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

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.

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 $1/N_S$.

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 systems 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.

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.

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.

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

$$CF = N_{S_i} \ln \left[\frac{\sqrt{(n_{AB} - n_B)(n_{AB} - n_A)}}{n_A + n_B - n_{AB}} + \mathcal{K}_0 \right] \tag{12}$$

where $\mathcal{K}_0$ is a constant that assures that the logarithm always takes a positive value. This constant is needed since the first factor of the logarithm can take a value below 1.

We finally prune the resulting hierarchy following the same stragegy mentioned for the NMI-based approach. 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).

3.4.4. Obtaining the interpolation weights of each cluster

Once the system has determined which LMs to interpolate at each dialogue turn, it has to estimate the relevance of each model in the context dependent model, p_D, to be interpolated with a background LM, p_B.

In Section 3.3 we proposed that the system could obtain feasible interpolation weights using its own estimations of the dialogue management procedure (in our case, the posterior probabilities for dialogue elements, given by the BNs). When considering the interpolation of LMs related to clusters of dialogue elements, we should include an additional restriction: the number of models to interpolate does not have to be equal to the number of dialogue elements, since each of them could be part of different clusters (when considering several layers of the hierarchy of clusters), and several elements could belong to the same cluster, thus generating a single LM for all of them.

To include this constraint, we slightly modify the criterion for the estimation of the interpolation weights. The system still uses the posterior probabilities of the dialogue elements, but it will also take into account the number of elements addressed by the user that belong to the different clusters. This way, the relevance of the LM associated to a given cluster will depend not only on the reliability of the elements of that cluster, but also on the *representativity* of each cluster in comparison with the rest. That is, the system should modify the relevance of a cluster depending on the number of positively inferred elements belonging to that cluster.

Previously we have proposed several strategies to obtain the interpolation weights (Lucas-Cuesta, Fernández, López, Ferreiros, & San-Segundo, 2010). We have decided to use the summation of the posterior probabilities of the elements belonging to each cluster. For the sake of simplicity, we will present the expressions of the interpolation weights of the approach including both concepts and goals. The corresponding expressions for concept-based or goal-based clustering could be easily derived.

Let us suppose that, at a certain point in the interaction, the system has already obtained the posterior probabilities of the different dialogue elements ($p_b(c_i|e_{c_i})$ for concept c_i, and $p_f(g_j|e_{g_j})$ for goal g_j, given their respective evidences, e_{c_i}, and e_{g_j}).

We will only consider those dialogue elements whose posterior probabilities rise above the corresponding thresholds Φ_C and Φ_G. We will denote by N_{C_i} and N_{G_i} the number of concepts and goals of cluster S_i with posterior probabilities above the relevance thresholds. Let us also suppose that those dialogue elements belong to N_S different clusters.

If we want to give the LM related to each cluster a relevance dependent on both the number of elements belonging to that cluster, and their posterior probabilities, we could consider to sum the contributions of the elements of the cluster (i.e. their posterior probabilities). Therefore, the interpolation weight w_{S_i} of the cluster S_i will be equal to

$$w_{S_i} = \frac{1}{k}\left[\sum_{j=1}^{N_{C_i}} p_b(c_j|e_{c_j}) + \sum_{j=1}^{N_{G_i}} p_f(g_j|e_{g_j})\right] \tag{13}$$

To assure that the sum of the interpolation weights is equal to one, we have to include a normalization constant k, which will be equal to the summation of the contribution of all the considered clusters:

$$k = \sum_{i=1}^{N_S}\left[\sum_{j=1}^{N_{C_i}} p_b(c_j|e_{c_j}) + \sum_{j=1}^{N_{G_i}} p_f(g_j|e_{g_j})\right] \tag{14}$$

Using the summation of posterior probabilities allows us to 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.

4. Evaluation

In this section we present the evaluation that we have carried out when using the different approaches to include dialogue-based information for modifying the LMs dynamically. We have evaluated several parameters in order to assess the performance of the system with the different grammar generation strategies. First of all, we have measured the improvement in the recognition performance, in terms of Word Accuracy. Additionally, we will show how our approaches improve the extraction of the semantics from the input utterance (in terms of Concept Accuracy), as well as the inference of the goals of the user (we assess this performance by estimating the Goal Accuracy).

We have used three databases to assess the performance of our dynamic LM adaptation approaches. The three databases have been labelled at the three knowledge layers: lexical (words), semantic (concepts), and intention (goals).

The first text database comprises 747 sentences. We use them to train the background component of the LM. Keeping in mind that reduced amount of data, our baseline LM is a bigram that has been built using a smoothing of the *n*-gram counts, plus a linear interpolation between the bigram and the unigram components (Manning & Schütze, 2002). All of the LMs that we estimate and interpolate at the adaptation step follow the same modelling approach.

We have another text database, comprising 516 fully labelled sentences. We use these sentences to train the LMs related to each of our approaches, the isolated and the clustered strategies.

The recorded database we have used to evaluate our strategies is called HIFI-MM1. It consists of 1300 sentences spoken by 13 different speakers (7 male, 6 female) addressing the Hi-Fi equipment. Each of them spoke 100 different sentences. We use this database to estimate the improvement in the recognition rates, as well as the understanding and dialogue metrics, when adapting the LMs dynamically.

The evaluation that we have carried out consists of using the information conveyed by an utterance (the extracted dialogue concepts and the inferred dialogue goals) to modify the LM and recognize the same sentence again. This way we can estimate an upper limit to the performance of the LM-adaptive system.

To increase the reliability of the results, we have performed a *k*-fold approach. We have randomly divided the database into 10 folds, each of them comprising 130 sentences. We use nine of the folds to adjust the different degrees of freedom of the system (relevance thresholds, and the interpolation weights λ_D), whereas the tenth fold is kept for the evaluation itself, using the values of the parameters obtained during the validation step.

The initial performance of our baseline system (i.e. without the adaptation of LMs) was of 94.67% for Word Accuracy (WAcc), 86.63% for Concept Accuracy (CAcc), and 73.80% for Goal Accuracy (GAcc). All of these measures were obtained using the HIFI-MM1 database and the *k*-fold approach. We will use them to discuss the improvements in the different approaches that we have applied.

4.1. Using an LM for each dialogue element

As we presented in the previous Section, our first strategy consisted of estimating an LM for each dialogue element (either concepts, goals or both). During a validation step, we estimated the values of the degrees of freedom of our strategy (the relevance threshold for concepts Φ_C, for goals Φ_G, or both, as well as the

interpolation weight λ_D between the background and the context dependent LMs). During the test step we kept these values and estimated the performance of the system. The values of our degrees of freedom, as well as the results of this evaluation in terms of the three performance figures aforementioned, can be seen in Table 4.

All the relevance thresholds take values of about 0.5. This means that even those dialogue elements that could not be considered as active or needed by the DM (i.e. those ones whose posterior probability falls below the decision threshold θ_i of the DM, see Section 2) conveys enough information to help building reliable LMs.

The interpolation weight λ_D takes values of between 0.1 and 0.13 for all the approaches. That is, it is enough to slightly modify the background LM (keeping at least 87% of its relevance in the dynamic model) to achieve better results in the three modules evaluated.

The Word Accuracy yields a maximum of 14.82% of relative error reduction over the baseline, when considering the combination of concepts and goals to adapt the LMs. This fact could be explained by taking into account the characteristics of our experimental setup. We are using the information conveyed in a single utterance (in terms of dialogue concepts and/or goals) to estimate a model that will be used to recognize the same utterance again. Therefore, the dynamic component of this adapted LM will only depend on those dialogue elements that appear in the same sentence, without any other concept or goal, that may influence the models when considering the different grouping strategies. That is, the isolated strategy is the one that achieves the highest specificity (in terms that each LM is related to a single piece of information, as opposed to the cluster-dependent LMs).

As regards the understanding and dialogue management metrics, our approach outperforms the baseline setup, with a maximum relative improvement of 5.46% (for Concept Accuracy) when using goal-dependent information, and 2.14% (for Goal Accuracy) using both dialogue elements. That is, the dynamic adaptation of the LMs of the speech recognizer also helps to better extract the semantic information (the task of the understanding module), which also gives rise to a better result when inferring the actions that the user wants to carry out (the task of the DM).

We can also see that considering concepts and goals together tends to yield better results than using each type of dialogue element separately. However, we could expect that better results were reached when considering only the goal-based adaptation, given the number of dialogue elements (58 concepts and 15 goals), and the number of labelled sentences (516) that we use to estimate the models related to each dialogue element. As the number of concepts is greater than the number of goals, the LMs associated to the concepts are usually trained with fewer sentences than the goal-based ones. In fact, the average number of sentences for each concept is 40.38, whereas the average number of sentences that make reference to each goal is 77.47. That is, the concept-based models are more poorly estimated than the goal-dependent LMs, which implies a slightly worse performance when using the concept-based approach. In any case, the differences in performance are not statistically significant.

Table 4
Parameters and performance of the dynamic LM estimation (one LM for each dialogue element).

Strategy	Φ_C	Φ_G	λ_D	WAcc	CAcc	GAcc
Baseline				94.67	86.63	73.80
Concept-based	0.53	–	0.13	95.22	87.32	74.15
Goal-based	–	0.43	0.10	95.27	87.36	74.19
Concept and goal merging	0.57	0.46	0.10	95.46	87.29	74.36

4.2. Using expert clustering

Our next experiment evaluated the performance of the dialogue system when using a grouping criterion dependent on the final application (see Section 3.4.1). We have assessed the same metrics as when considering a single LM for each dialogue element. Table 5 shows the values of the relevance thresholds Φ_C and Φ_G, the interpolation weight λ_D, and the results of the evaluation, for the two grouping approaches that we propose.

When using expert clustering, the interpolation weight λ_D takes values slightly higher than in when using isolated LMs. Its value rises up to about 0.20, which means that even by keeping 80% of the background LM, it is enough to yield a slight improvement in the performance of the dialogue system. As regards the relevance thresholds, the most remarkable difference is seen when not considering the *Rest* cluster. In this case, the system needs to keep more dialogue concepts than those considered as needed by the DM (that is, those ones whose posterior probabilities are above the decision threshold θ_i, see Section 2). This behaviour could be explained by looking at the distribution of sentences in each LM considered, together with the number of concepts (58) and goals (15) defined in our domain. As we established in the previous experiment, the concept-based LMs are trained with a reduced number of sentences, thus resulting in more poorly estimated LMs when compared to the goal-based ones. In this situation, the system tends to consider even the models related to those concepts with reduced posterior probabilities (according to the backward inference of the DM) in order to obtain a dynamic LM as robust as possible.

Despite the slight improvement in the three metrics we have considered (which yields their maximum values for the full grouping taking into account goal-based information only; the relative improvement is 8.63% for Word Accuracy, 5.01% for Concept Accuracy and 2.02% for Goal Accuracy), none of them outperforms the results of the previous experiment. This behaviour could be explained by taking into account that, when grouping several elements into the same cluster, it is possible that a single utterance contains elements that belong to different clusters. For instance, the sentence *Play track 5 of CD 2 and raise the volume* makes reference to two different expert clusters: the *CD* one and the *VOLUME* one. In more complex utterances, the number of clusters to be considered may be even higher (remember the average number of dialogue concepts and goals referred to in each utterance, see Section 3.2). Under these circumstances, the system could consider clusters which contains several dialogue elements that were not referred in the utterance. Therefore, the Dynamic Grammar Estimator may interpolate the background LM with a too generalistic context-dependent LM, that is, the result of the adaptation could be an LM with insufficient discriminative strength.

4.3. Using hierarchical single-level LSA clustering

We then assessed the performance of the dynamic estimation of the LMs considering the first strategy of LSA-based clustering (that is, selecting a single level in the cluster tree, see Section 3.4.2). We estimate the interpolation weights between the different LMs to be obtained as the sum of the posterior probabilities of the dialogue elements that belong to each cluster (see Section 3.4.4). The values of the relevance thresholds and λ_D, as well as our performance figures, can be seen in Table 6.

It is interesting to note that, when considering the LSA-based clustering, the system gives more relevance to the context-dependent LM when using intention-based information (i.e. dialogue

Table 5
Parameters and performance of the dynamic LM estimation (expert grouping of dialogue elements).

Grouping	All groups						Without *Rest* group					
Strategy	Φ_C	Φ_G	λ_D	WAcc	CAcc	GAcc	Φ_C	Φ_G	λ_D	WAcc	CAcc	GAcc
Baseline				94.67	86.63	73.80				94.67	86.63	73.80
Concept-based	0.44	–	0.20	94.98	87.20	74.22	0.33	–	0.20	95.07	87.21	74.28
Goal-based	–	0.44	0.21	95.13	87.30	74.33	–	0.52	0.18	94.98	87.00	74.12

Table 6
Parameters and performance of the dynamic LM estimation (LSA-based single-level clustering of dialogue elements).

Strategy	Φ_C	Φ_G	λ_D	WAcc	CAcc	GAcc
Baseline				94.67	86.63	73.80
Concept-based	0.47	–	0.15	95.10	87.13	74.08
Goal-based	–	0.46	0.24	95.30	87.41	74.47

goals). That is, the interpolation weight λ_D is higher within this strategy. The main reason for this behaviour lies in both the more reduced number of dialogue goals (and thus the more robust the goal-dependent LMs are), and in the character of the dialogue goals. Indeed, as our application only considers 15 dialogue goals, but 58 concepts, each goal can be associated to several concepts (for instance, the goal MODIFY_VOLUME with the concepts VOLUME_PARAMETER, VOLUME_VALUE and VOLUME_ACTION). In other words, the inference procedure could be seen as an integration of information, from a sparse source (the concepts) to a more concentrated one (the goals). Then, the more integrated the information is, the more reliable the context dependent LM will be, and thus the more relevance the system will give to that component (and the lower the value of λ_D).

Despite the Word Accuracy of this approach is above that corresponding to the expert partitioning approach, it cannot outperform the isolated approach. The best result (11.82% of relative error reduction) is achieved when using goal-based information (which implies considering up to 4 different clusters of goals). The Concept Accuracy achieves similar results to the isolated approach: the relative error reduction takes a maximum value of 5.83% when using goal-based information. More interestingly, the Goal Accuracy takes its maximum value over all the dynamic LM estimation approaches, with a relative error reduction of 2.56% when considering also goal-based clustering.

4.4. Using hierarchical multi-level LSA clustering

We finally measured the performance when keeping several levels of the cluster hierarchies of dialogue elements (either concepts, goals, or both). As we said before, we estimate the interpolation weights between the cluster-dependent LMs as the summation of the posterior probabilities of the elements belonging to each cluster.

Table 7 shows the values of the relevance thresholds for concepts Φ_C and goals Φ_G, the interpolation weight λ_D between the background LM and the context-dependent one, and the results of the evaluation carried out, when considering each of the proposed strategies (keeping the full tree of clusters, that is, from the isolated elements up to a model that contains all the dialogue elements, and the two pruning strategies proposed in Section 3.4).

As in the previous experiments, the values of both relevance thresholds fall within the intermediate region of confidence, which means that even those concepts and goals with posterior probabilities that, according to the inference procedure, should not be considered as needed or active (see Section 2) may help to better recognize the current sentence. In a similar way, the interpolation weights λ_D for all the pruning strategies take values between of 0.1 and 0.2: the dynamic modification of the LMs can improve the performance of the speech recognizer even considering only a 10% of the context-dependent LM.

The performance of the multi-level approach is similar to that obtained with the single-level clustering (Section 4.3). This makes sense, provided that the generation of clusters remains the same, and the only difference lies in which clusters were selected to generate an associated LM. In any case, we considered the development of the multi-level strategy as a way of establishing a tradeoff between the robustness of the models, and the number of LMs (and thus their specificity).

The maximum values of each metric are reached when using goal-based information, and applying one of the pruning strategies (the initial one for Word Accuracy, with a relative error reduction of 13.51%; the restrictive one for Concept and Goal Accuracy, with error reductions of 5.31% and 2.02%, respectively). This is also consistent with the proposed approach, and the databases we have used. When using the full hierarchy, we consider more clusters than dialogue elements (see Section 3.4). This can give rise to an overtraining of the context-dependent LM, provided that the system makes use of the same sentences to train different LMs. This situation is partially avoided with the initial pruning strategy. However, a too restrictive pruning may lead to the opposite situation: the number of clusters becomes too reduced, so the context-dependent LM that is generated at each dialogue turn tends to be too general, thus decreasing the performance.

Table 7
Parameters and performance of the dynamic LM estimation (LSA-based multi-level clustering of dialogue elements).

Hierarchy	Strategy	Φ_C	Φ_G	λ_D	WAcc	CAcc	GAcc
Baseline					94.67	86.63	73.80
Full	Concept-based	0.52	–	0.10	95.28	87.32	74.26
	Goal-based	–	0.51	0.10	95.30	87.23	74.15
	Concept-based	0.49	–	0.10	95.21	87.27	74.22
Initial pruning	Goal-based	–	0.46	0.10	95.39	87.25	74.12
	Concept & goal merging	0.54	0.61	0.19	95.21	87.34	74.29
	Concept-based	0.48	–	0.14	95.07	87.05	74.22
Restrictive pruning	Goal-based	–	0.47	0.19	95.27	87.34	74.33
	Concept & goal merging	0.53	0.53	0.11	95.13	87.27	74.26

4.5. Using automatic perplexity-based hierarchical clustering

In our first experiment we consider the clustering strategy based on maximum normalized mutual information (NMI). Table 8 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 dialogue elements for the clustering. We also include the performance of the baseline system.

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.

We next evaluate the performance of the adapted system when using the Minimum Global Perplexity criterion. Table 9 shows the results of the evaluation of this strategy.

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 marginally significant with confidence intervals of 90%.

Merging both dialogue elements into the clustering 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.

4.6. Results with HIFI-PM2

The results obtained with the previous database are promising (since the recognition performance tends to improve), but none of the results is statistically significant with confidence intervals of 95%. This behaviour moved us to assess the performance of our dynamic LM adaptation approaches with another database, more realistic from a dialogue point of view. This database, referred to as HIFI-PM2, comprises dialogues of 40 speakers, 20 male and 20 female, that developed different interaction scenarios with our system. The database comprises 9162 sentences that yielded a recognition performance of 72.20 ± 0.51% with the baseline setup of the system.

Table 8
Performance of the NMI-based language modelling.

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

Table 9
Performance of the Minimum Perplexity-based language modelling.

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

Table 10
Performance of the recognition system for HIFI-PM2 database.

System configuration	Φ_G	λ_D	Word Accuracy (%)
Baseline	–	0	72.20
Isolated model, goals	0.52	0.22	74.50
Perplexity-based clustering, goals	0.48	0.19	73.99

We have assessed the recognition performance only with these approaches that yielded the best results when using HIFI-MM1 database. That is, the adaptation based on using a specific LM for each dialogue goal (i.e. 15 content-specific models), and when considering the automatic clustering strategy of dialogue goals based on the minimization of a global perplexity (keeping thus 10 LMs).

The degrees of freedom of the experiments were the relevance threshold for dialogue goals Φ_G and the interpolation weight λ_D between the static, background model p_B and the context dependent model p_D. The results of this offline evaluation are presented in Table 10.

The value of Φ_G shows that it is enough to use only those dialogue goals with middle and high posterior probabilities to the estimation. In other words, the system does not need to be too strict in selecting the goals to be interpolated (which happens with higher values of the threshold). It also does not have to be too permissive, allowing goals with reduced posterior probability (and thus the goals that are not inferred by the Bayesian networks from the user's utterance) to be included in the dynamic LM.

On the other hand, the contribution of the context-dependent model p_D is reduced enough so as to allow the speech recognizer to still take into account expressions not directly related to the current one. That is, it is enough to keep about a 75% of the contribution of the background, static LM p_B to obtain a dynamically adapted LM that can better recognize a given sentence.

However, the most important achievement is that the system itself is able to obtain an accurate estimation of the interpolation weights between the different content-specific components of the context-dependent LM. In other words, the system can effectively merge the LMs related to the pieces of information that it considers without training any interpolation weight.

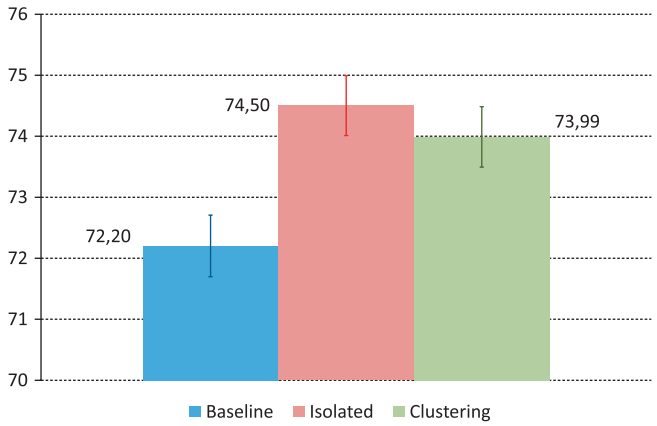

Fig. 7. Speech recognition performance for HIFI-PM2 database.

Finally, we can see that all the strategies proposed clearly outperform the baseline (with relative reduction of Word Error Rate of 8.27% and 6.44% for each approach). This improvement is statistically significant when comparing any of the strategies against the baseline. However, the differences between the different strategies for dynamically estimating the LM are not significant. We present the results of this evaluation, together with the corresponding confidence intervals, in Fig. 7.

5. Conclusions

We have presented an approach to make use of semantic and intention-dependent information as knowledge sources to dynamically modify the LMs that a speech recognizer (being part of a spoken dialogue system) makes use of. The adaptation of the LM is performed by means of a linear interpolation between a background LM and one or several models related to the different dialogue elements, either semantic (concepts) or intention-dependent (goals).

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 different confidence scores and posterior probabilities obtained by the understanding module and the Dialogue Manager. 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.

As regards how to build the LMs that are used at the interpolation step, we propose two ways to proceed. In the first one, each LM is associated to a single dialogue element. Once the system has inferred the elements that the speaker has addressed in the current utterance, the corresponding LMs are interpolated. The second approach follows a clustering strategy before training the LMs, in such a way that different dialogue elements may be part of the same cluster, and thus be associated to the same LM. We have proposed an expert strategy for grouping dialogue elements together, as well as a hierarchical semi-automatic semantic clustering approach, based on an adaptation of the Latent Semantic Analysis (LSA) framework.

The results of our evaluation show how the system can estimate sensible dynamic models at each dialogue turn, and more importantly, how the improvement of these LMs (used by the speech recognizer) can increase the performance of other modules of the system (the speech understanding and the dialogue manager).

We have obtained significant improvements when considering a database of actual dialogues, thus demonstrating that our dynamic approach to estimate LMs is able to obtain reliable LMs at dialogue time, without any offline estimation of the interpolation weights. That is, our system can obtain accurate enough LMs by taking under consideration the elements of information inferred by the dialogue manager (i.e. the dialogue goals that the user wants to carry out).

We have seen how our clustering approaches tend to outperform the results of Goal Accuracy when compared to the isolated element-based LM. A remarkable conclusion is that the highest improvements in terms of Goal Accuracy are not always reached with the configuration that reaches the best recognition performance (see, for instance, the two clustering strategies), though the differences are not significant. This behaviour could be explained by the nature of the errors in the different subsystems. We have studied the number of insertions, substitutions, and deletions for the different approaches, and we have checked the varying performance aforementioned. For instance, when considering the single-level semantic clustering, the number of insertions of the optimum strategy from the dialogue point of view (using goal-based information) is 24.14% greater than the number of insertions of the recognition-optimum approach. In the case of the multi-level clustering, the relative difference between number of insertions for the Word Accuracy optimum and the Goal Accuracy optimum, reaches 12.5%. When the number of insertions in the recognition hypothesis becomes higher, the word error rate increases (i.e. the performance worsens). However, the understanding module and the DM can take advantage of this overinformation to extract the proper dialogue concepts and to infer the goals addressed by the user.

6. Future work

We will now present some of the current research guidelines that we are currently developing with regard to the dynamic adaptation of LMs.

We are aware that the databases that we have used are somewhat 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).

The semantic clustering we have proposed is based on the distance between clusters dependent on the Pearson correlation coefficient between dialogue elements. We are now using other distances, more closely related to the Information Retrieval field, such as a cosine distance among the feature vectors of the LSA matrix.

We are also defining a strategy to adjust dynamically the interpolation weight λ_D between the background LM and the context dependent one, instead of obtaining it at a validation stage. We are defining a variation range in the environment of an initial interpolation value, giving more relevance to the context dependent LM when the posterior probabilities of the dialogue elements increase, and vice versa.

For the evaluation presented in this paper, we have used the information conveyed in each utterance to modify the LM, performing an additional recognition (followed by the extraction of the semantics and the inference of the dialogue goals). Another application of our approaches will consist of modifying the LMs dynamically at each dialogue turn using the information provided during the previous turns, in order to better recognize the next utterance that the speaker might say to the system. We could also use the information retrieved to the user via the language generation, trying to adapt the LMs to what the system asks to the users.

We are also setting up an evaluation of the adaptive system with real users, in an effort to measure the performance figures presented here, as well as other metrics regarding the dialogue efficiency, such as task completion, number of turns needed to fulfill a task, and so on.

Finally, we may also think about applying the adaptation of LMs using other sources of information. For instance, the knowledge that the system has on the users. Provided that each user might express his or her ideas in different ways (not only in terms of prosodic patterns, but also from the lexical and rhetorical point of view), and even with disparate emotional content, the system could take advantage of this information once it has identified the speaker, to adapt the LMs (indirectly improving the performance of the full dialogue system) to the current 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), DPI2007-66846-C02-02 (ROBONAUTA) and

DPI2010-21247-C02-02 (INAPRA), and by the Spanish Ministry of Education, under contract AP2007–00463 (FPU Grant).

References

Bacchiani, M., Riley, M., Roark, B., & Sproat, R. (2006). MAP adaptation of stochastic grammars. *Computer Speech and Language, 20*, 41–68.

Bacchiani, M., & Roark, B. (2003). Unsupervised language model adaptation. In *Proceedings of the international conference on acoustic, speech and signal processing (ICASSP)* (Vol. I, pp. 224–227).

Bellegarda, J. R. (2000). Exploiting latent semantic information in statistical language modeling. *Proceedings of IEEE 88*(8), 1279–1296.

Bellegarda, J. R. (2001). An overview of statistical language model adaptation. In *Proceedings of the international workshop on adaptation methods for speech recognition* (pp. 165–174).

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

Bellegarda, J. R., Butzberger, J. W., Chow, Y. L., Coccaro, N. B., & Naik, D. (1996). A novel word clustering algorithm based on latent semantic analysis. In *Proceedings of the international conference on acoustic, speech and signal processing (ICASSP)* (Vol. I, pp. 172–175).

Chen, L., Gauvain, J. L., Lamel, L., Adda, G., & Adda, M. (2001). Using information retrieval methods for language model adaptation. In *Proceedings of the 7th European conference on speech communication and technology (EUROSPEECH)* (pp. 255–258).

Darroch, J. N., & Ratcliff, D. (1972). Generalized iterative scaling for log-linear models. *The Annals of Mathematical Statistics, 43*(5), 1470–1480.

Federico, M. (1996). Bayesian estimation methods for n-gram language model adaptation. In *Proceedings of the international conference on spoken language processing (ICSLP)* (pp. 240–243).

Federico, M., & Bertoldi, N. (2004). Broadcast news LM adaptation over time. *Computer Speech and Language, 18*, 417–435.

Fernández, F., Ferreiros, J., Sama, V., Montero, J. M., San-Segundo, R., & Macías-Guarasa, J. (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 (INTERSPEECH)* (pp. 3421–3424).

Fügen, C., Holzapfel, H., & Waibel, A. (2004). Tight coupling of speech recognition and dialog management – dialog-context dependent grammar weighting for speech recognition. In *Proceedings of the international conference on spoken language processing (ICSLP)* (Vol. I, pp. 169–172).

Gruenstein, A., Wang, C., & Seneff, S. (2005). Context-sensitive statistical language modeling. In *Proceedings of the 5th international conference on speech communication and technology (INTERSPEECH)* (pp. 17–20).

Hsu, B. J. (2007). Generalized linear interpolation of language models. In *Proceedings of the IEEE workshop on automatic speech recognition and understanding (ASRU)* (pp. 136–140).

Iyer, R., Ostendorf, M., & Rohlicek, J. R. (1994). Language modeling with sentence-level mixtures. In *Proceedings of the ARPA HLT workshop* (pp. 82–87).

Iyer, R. M., & Ostendorf, M. (1999). Modeling long distance dependence in language: Topic mixture versus dynamic cache models. *IEEE Transactions on Speech and Audio Processing 7*(1), 30–39.

Jelinek, F., Merialdo, B., Roukos, S., & Strauss, M. (1991). A dynamic language model for speech recognition. In *Proceedings of the DARPA workshop on speech and natural language* (pp. 293–295).

Justo, R., & Torres, M. I. (2007). Word segments in category-based language models for automatic speech recognition. *Lecture Notes in Computer Science, 4477*, 249–256.

Klakow, D. (1998). Log-linear interpolation of language models. In *Proceedings of the international conference on spoken language processing (ICSLP)* (Vol. 5, pp. 1695–1698).

Kneser, R., & Steinbiss, V. (1993). On the dynamic adaptation of stochastic language models. In *Proceedings of the international conference on audio speech and signal processing (ICASSP)* (Vol. II, pp. 586–589).

Kuhn, R., & de Mori, R. (1990). A cache-based natural language model for speech recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 12*(6), 570–583.

Landauer, T. K., Foltz, P. W., & Laham, D. (1998). An introduction to latent semantic analysis. *Discourse Processes, 25*, 259–284.

Lau, R., Rosenfeld, R., & Roukos, S. (1993). Trigger-based language models: A maximum entropy approach. In *Proceedings of the IEEE international conference on acoustic, speech and signal processing (ICASSP)* (Vol. 2, pp. 45–48).

Lecorvé, G., Gravier, G., & Sébillot, P. (2009). Constraint selection for topic-based MDI adaptation of language models. In *Proceedings of the 9th international conference on speech communication and technology (INTERSPEECH)* (pp. 368–371).

Liu, X., Gales, M. J. F., & Woodland, P. C. (2008). Context dependent language model adaptation. In *Proceedings of the 8th international conference on speech communication and technology (INTERSPEECH)* (pp. 837–840).

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

Lobacheva, Y. (2000). *Discourse mixture language modeling*. Master's thesis. College of Engineering, Boston University.

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

López-Cózar, R., & Callejas, Z. (2008). ASR post-correction for spoken dialogue systems based on semantic, syntactic, lexical and contextual information. *Speech Communication, 50*, 745–766.

López-Cózar, R., & Griol, D. (2010). New technique to enhance the performance of spoken dialogue systems based on dialogue states-dependent language models and grammatical rules. In *Proceedings of the 10th international conference on speech communication and technology (INTERSPEECH)* (pp. 2998–3001).

Lucas-Cuesta, J. M., Fernández, F., & Ferreiros, J. (2009). Using dialogue-based dynamic language models for improving speech recognition. In *Proceedings of the 9th international conference on speech communication and technology (INTERSPEECH)* (pp. 2471–2474).

Lucas-Cuesta, J. M., Fernández, F., López, V., Ferreiros, J., & San-Segundo, R. (2010). Clustering of syntactic and discursive information for the dynamic adaptation of Language Models. *Revista de la Sociedad Española de Procesamiento del Lenguaje Natural, 45*, 175–182.

Lucas-Cuesta, J. M., Fernández, F., Salazar, J., Ferreiros, J., & San-Segundo, R. (2009). Managing speaker identity and user profiles in a spoken dialogue system. *Revista de la Sociedad Española de Procesamiento del Lenguaje Natural, 43*, 77–84.

Lucas-Cuesta, J. M., Fernández-Martínez, F., Dragos, G., Lutfi, S., & Ferreiros, J. (2011). Evaluation of a user-adapted spoken language dialogue system: Measuring the relevance of the contextual information sources. In *Proceedings of the 3rd international conference on agents and artificial intelligence (ICAART)* (pp. 218–223).

Manning, C. D., & Schütze, H. (2002). *Foundations of statistical natural language processing*. The MIT Press.

Martins, C., Teixeira, A., & Neto, J. (2010). Dynamic language modeling for European Portuguese. *Computer Speech and Language, 24*, 750–773.

Popovici, C., & Baggia, P. (1997). Specialized language models using dialogue predictions. In *Proceedings of the international conference on audio speech and signal processing (ICASSP)* (Vol. 2, pp. 815–818).

Rao, P. S., Monkowski, M. D., & Roukos, S. (1995). Language model adaptation via minimum discrimination information. In *Proceedings of the international conference on audio speech and signal processing (ICASSP)* (pp. 161–164).

Raux, A., Mehta, N., Ramachandran, D., & Gupta, R. (2010). Dynamic language modeling using bayesian networks for spoken dialog systems. In *Proceedings of the 10th international conference on speech communication and technology (INTERSPEECH)* (pp. 3030–3033).

Riccardi, G., & Gorin, A. L. (2000). Stochastic language adaptation over time and state in natural spoken dialog systems. *IEEE Transactions on Speech and Audio Processing, 8*(1), 3–10.

Riccardi, G., Potamianos, A., & Narayanan, S. (1998). Language model adaptation for spoken language system. In *Proceedings of the international conference on spoken language processing (ICSLP)* (pp. 2327–2330).

Rosenfeld, R. (1994). *Adaptive statistical language modeling: A maximum entropy approach*. Ph.D. thesis. School of Computer Science, Carnegie Mellon University.

Rosenfeld, R., & Huang, X. (1992). Improvements in stochastic language modeling. In *Proceedingso of the DARPA workshop on speech and natural language* (pp. 107–111).

Saykham, K., Chotimongkol, A., & Wutiwiwatchai, C. (2010). Online temporal language model adaptation for a thai broadcast news transcription system. In *Proceedings of the 7th international conference on language resources and evaluation (LREC)* (pp. 1690–1694).

Shi, Q., Chu, S. M., Liu, W., Kuo, H. K., Liu, Y., & Qin, Y. (2008). Search and classification based language model adaptation. In *Proceedings of the 8th international conference on speech communication and technology (INTERSPEECH)* (pp. 1578–1581).

Solsona, R. A., Fosler-Lussier, E., Kuo, H. K. J., Potamianos, A., & Zitouni, I. (2002). Adaptive language models for spoken dialogue systems. In *Proceedings of the international conference on audio speech and signal processing (ICASSP)* (No. I, pp. 37–40).

Straková, J., & Pecina, P. (2010). Czech information retrieval with syntax-based language models. In *Proceedings of the 7th international conference on language resources and evaluation (LREC)* (pp. 1359–1362).

Tam, Y. C., & Schultz, T. (2006). Unsupervised language model adaptation using latent semantic marginals. In *Proceedings of the 8th international conference on speech communication and technology (INTERSPEECH)* (pp. 2206–2209).

Tur, G., & Stolcke, A. (2007). Unsupervised language model adaptation for meeting recognition. In *Proceedings of the international conference on audio speech and signal processing (ICASSP)* (No. IV, pp. 173–176).

Ueberla, J. (1994). *Analyzing and improving statistical language models for speech recognition*. Ph.D. thesis. School of Computing Science, Simon Fraser University.

Visweswariah, K., & Printz, H. (2001). Language models conditioned on dialog state. In *Proceedings of the 7th European conference on speech communication and technology (EUROSPEECH)* (pp. 251–254).

Wessel, F., & Baader, A. (1999). Robust dialogue-state dependent language modeling using leaving-one-out. In *Proceedings of the international conference on audio speech and signal processing (ICASSP)* (pp. 741–744).

Yamamoto, H., Hanazawa, K., Miki, K., & Shinoda, K. (2010). Dynamic language model adaptation using keyword category classification. In *Proceedings of the 10th international conference on speech communication and technology (INTERSPEECH)* (pp. 2426–2429).