Using Finite State Models for the Integration of Hierarchical LMs into ASR Systems

Raquel Justo* and M. Inés Torres

University of the Basque Country
Sarriena s/n, 48940 Leioa, Spain
raquel.justo@ehu.es, manes.torres@ehu.es

Abstract. Through out this work we explore different methods to integrate a complex Language Model (a hierarchical Language Model based on classes of phrases) into an Automatic Speech Recognition (ASR) system. The integration is carried out by means of a composition of the different Stochastic Finite State Automata associated to the specific Language Model. This method is based on the same idea employed to integrate the different knowledge sources involved in the recognition process when a classical word-based Language Model is considered. The obtained results show that this integrated architecture provides better ASR system performance than a two-pass decoder where the complex LM is employed to reorder the N-best list.

Keywords: stochastic finite state models, speech recognition, hierarchical language models.

1 Introduction

Statistical decision theory is applied in a wide variety of problems within pattern recognition framework that aim at minimising the probability of erroneous classifications. The maximization of the posterior probability $P(\bar{w}|\bar{x})$ allows to get the most likely sequence of symbols $\bar{w}$, that matches a given sequence of input observations $\bar{x}$, as shown in eq. (1).

$$\hat{\bar{w}} = \arg\max_{\bar{w}} = P(\bar{w}|\bar{x}) \tag{1}$$

Using the Bayes' decision rule eq. (1) can be rewritten as eq. (2). If we focus on the problem of Automatic Speech Recognition (ASR) the term $P(\bar{w})$ corresponds to the prior probability of a word sequence and it is commonly estimated by a Language Model (LM), whereas $P(\bar{x}|\bar{w})$ is estimated by an Acoustic Model (AM), tipically a Hidden Markov Model (HMM).

$$\hat{\bar{w}} = \arg\max_{\bar{w}} P(\bar{w}|\bar{x}) = \arg\max_{\bar{w}} P(\bar{x}|\bar{w})P(\bar{w}) \tag{2}$$

* This work has been partially supported by the Government of the Basque Country under grant IT375-10, by the Spanish CICYT under grant TIN2008-06856-C05-01 and by the Spanish program Consolider-Ingenio 2010 under grant CSD2007-00018.

J.-F. Martínez-Trinidad et al. (Eds.): MCPR 2011, LNCS 6718, pp. 332–340, 2011.

Nowadays Automatic Speech Recognition (ASR) systems use, mainly, Statistical Language Models (SLMs) in order to represent the way in which the combination of words is carried out in a specific language. Other approaches such as syntactic LMs, including a stochastic component, could also be employed in this kind of applications, i.e. stochastic context free grammars (SCFG)[5,2] or stochastic finite state models [10,11]. Although syntactic models can better model the structure of the language they still present problems regarding automatic inference and integration in ASR systems. In this work we use a syntactic approach, specifically k-Testable in the Strict Sense (k-TSS) LMs. k-TSS languages are a subclass of regular languages and, unlike SCFGs, they can be easily inferred from a set of positive samples by an inference algorithm [4]. Moreover, k-TSS LMs can be represented by Stochastic Finite State Automata (SFSA) allowing an efficient composition of them with other models, i.e. HMMs (in ASR applications).

AT&T laboratories presented an approach that simplifies the integration of different knowledge sources into the ASR system by using finite state models, specifically Stochastic Finite State Transducers (SFST) [10]. The underlying idea is to use a SFST to model each knowledge source, then SFSTs are compounded to obtain an only one SFST where the search of the best word sequence is carried out. Although optimization algorithms [8] can be applied the resulting SFST could still result too memory demanding. One way to solve this problem is the "on-the-fly" composition of SFSTs [3]. In the same way, since k-TSS LMs that can be represented by SFSA are considered in this work, the automaton associated to the LM is compounded with the HMMs representing AMs. Moreover, the idea of "on-the-fly" composition has also been used to obtain less memory demanding approaches.

One of the problems to be faced within the ASR framework is the selection of an appropriate LM. Among SLMs, word n-gram LMs are the most widely used approach, because of their effectiveness when it comes to minimizing the Word Error Rate. Large amounts of training data are required to get a robust estimation of the parameters defining aforementioned models. However there are numerous ASR applications for which the amount of training material available is rather limited. Different approaches can be found in the literature in order to solve this problem [9,12]. In this work, we employ hierarchical LMs based on classes of phrases [7] that has demonstrated to be efficient when dealing with data sparseness problems.

This kind of complex LMs, integrating different knowledge sources, entail an additional problem regarding the integration of them into the ASR system. One of the ways employed to solve this problem is to use a two-pass decoder, that is, first, a list of the N-best hypothesis is obtained from a classical decoder that considers a word-based LM. Then, the complex LM of choice is employed to reorder the list and to obtain the best word sequence. This decoupled architecture allows the recognition process to be carried out without any change in the decoder. However, it does not permit to take advantage of all the potential of the model because the recognition process is not guided by the LM of choice.

Alternatively, an integrated architecture which employs a one-pass decoder could be considered. This kind of integration is based on the use of SFSA associated to the LM. In this work in order to integrate hierarchical LMs into the ASR system we propose to use the same idea employed to integrate different knowledge sources in an ASR system. That is, the integration is carried out by doing an "on-the-fly" composition of the different SFSA associated to the different knowledge sources in the hierarchical LM.

2 A Hierarchical Language Model Based on Classes of Phrases

In this section we present the LMs employed in this work: a word-based LM M_w, a hierarchical LM based on classes of phrases M_{sw} and an interpolated LM, M_{hsw}, fully described and formulated in [7]. These models are defined within the Stochastic Finite State framework, specifically we use k-TSS LMs.

Thus, under the k-TSS formalism the probability of a sequence of N words ($\bar{w} = w_1, \ldots, w_N = w_1^N$) is obtained considering the history of previous $k_w - 1$ words as shown in eq. (3), when considering a classical word based model (M_w).

$$P(\bar{w}) \simeq P_{M_w}(\bar{w}) = \prod_{i=1}^{N} P(w_i | w_{i-k_w+1}^{i-1}) \tag{3}$$

On the other hand, the probability of a word sequence ($\bar{w}$) using the M_{sw} model is given in the equation below.

$$P(\bar{w}) = \sum_{\forall \bar{c} \in \mathcal{C}^*} \sum_{\forall s \in \mathcal{S}(\bar{w})} P(\bar{w} | s, \bar{c}) P(s | \bar{c}) P(\bar{c}) \tag{4}$$

where $\mathcal{C}^*$ is a set of all the possible class sequences ($\bar{c}$) given a priori defined set of classes made up of phrases. s is a segmentation of a word sequence $w_1, \ldots . w_N$ in M phrases and can be understood as a vector of M indexes. The set of all possible segmentations of a word sequence $\bar{w}$ is denoted by $S(\bar{w})$.

The third term involved in eq (4) can be calculated as a product of conditional probabilities and it is approached by a class k-TSS model. The SFSA associated to the model can be inferred from a classified corpus and provides the probability for each class sequence as eq (5) shows.

$$P(\bar{c}) = \prod_{i=1}^{T} P(c_i | c_1^{i-1}) \simeq P(c_i | c_{i-k_c-1}^{i-1}) \tag{5}$$

where $k_c - 1$ stands for the maximum length of the considered class history.

To estimate the probability of the second term in eq (4) we assume that the segmentation probability is constant, that is, $P(s|\bar{c}) \simeq \alpha$.

Finally, $P(\bar{w}|s,\bar{c})$ is estimated considering that given a sequence of classes $\bar{c}$ and a segmentation s, the probability of a phrase given a class c_i depends exclusively on this c_i class and not on the previous ones

$$P(\bar{w}|s,\bar{c}) \simeq \prod_{i=1}^{T} P(w_{a_{i-1}+1}^{a_i}|c_i) \tag{6}$$

The term $P(w_{a_{i-1}+1}^{a_i}|c_i)$ represents the probability of a sequence of words, which is the phrase corresponding to the segmentation indexes $(a_{i-1}+1, a_i)$, given the class of this phrase. To estimate this probability, a k-TSS model, represented by an SFSA, can be used for each class, as shown in eq (7).

$$P(w_{a_{i-1}+1}^{a_i}|c_i) \simeq \prod_{j=a_{i-1}+1}^{a_i} P(w_j|w_{j-k_{cw}+1}^{j-1}, c_i) \tag{7}$$

where $k_{cw}-1$ stands for the maximum length of the word history that is considered in each class c_i.

Summing up, N_c+1 (where N_c is the considered number of classes) SFSA are needed to represent the M_{sw} model: one for each class considering the relations among words inside the classes and an additional one that considers the relations among classes.

Finally, an interpolated model (M_{hsw}) is defined, here, as a linear combination of a word-based LM, M_w, and a hierarchical LM based on classes of phrases, M_{sw}. Using such a model the probability of a word sequence is given by eq. (8).

$$P_{M_{hsw}} = \lambda P_{M_w}(\bar{w}) + (1-\lambda) P_{M_{sw}}(\bar{w}) \tag{8}$$

3 Integration of Complex LMs into an ASR System

The goal of an ASR system is to obtain the most likely word sequence given the acoustic signal uttered by the speaker. In this work, all the models involved in the decoding process (acoustic models AM, language model LM and lexical model) were integrated into the SFSA framework. Thus, the problem of finding the most likely word sequence would be solve by finding the most likely path in the search network obtained by doing the composition of all the automata representing the models. However, a static composition of all the automata can cause computation problems regarding memory allocation when large vocabularies are employed. Instead of carrying out such a composition where different parts of the network are replicated, the composition of different models could be done on demand at decoding time. Fig. 1 illustrates the search network built to carry out this kind of integration when a classical M_w model is employed.

A vocabulary of two words w1="no" and w2 = "nada" has been employed. In order to obtain the transition probabilities among different nodes s_i of the network, the SFSA associated to each model has to be consulted when required. Specifically, the transition probabilities among words (red arrows in Fig. 1) are calculated turning to the SFSA associated to the word k-TSS LM (M_w).

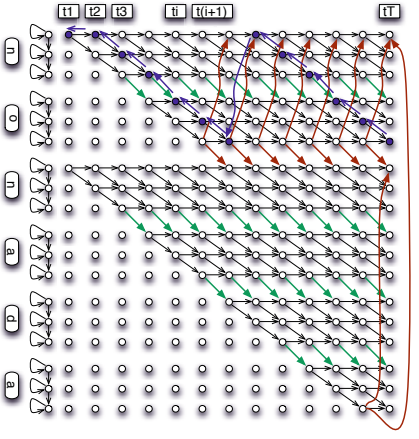

Fig. 1. Search network for a word 1-TSS model

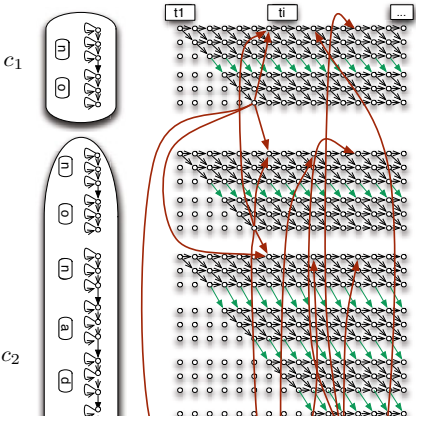

Fig. 2. Search network for M_{sw}, whith $k_c = 1$ and $k_{cw} = 1$

However, in this work, we want to integrate in the ASR system a LM considering different knowledge sources, the M_{sw} model. In order to do this we used different architectures for comparison purposes. In the first one, decoupled architecture shown in Fig. 3, the recognition process is carried out using a two-pass decoder that considers a standard word-based LM (M_w). The output of the ASR system is a word-graph from which the N-best list is commonly extracted. However, the obtention of the word graph entails prohibitive computational costs or coarse approaches due to very restrictive assumptions [6]. Thus we do not obtain the real N-best list but an approach of it. Then, the M_{sw} model is employed to provide a new score to the obtained hypothesis and to reorder them in terms of this new score. Thus, we finally obtain a new best hypothesis which is considered the output of the system when the M_{sw} model is used. Although, this architecture tries to simulate the integration of the model into the ASR system

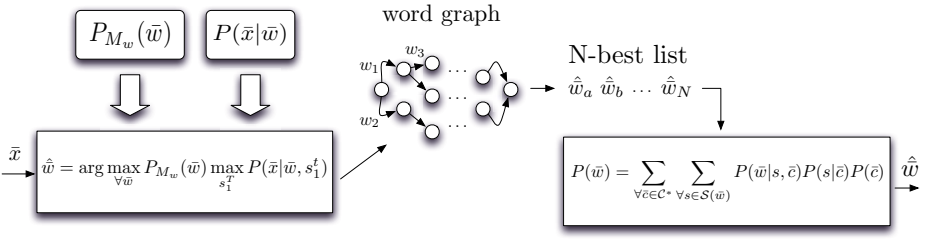

Fig. 3. Decoupled architecture for an ASR system considering a M_{sw} model

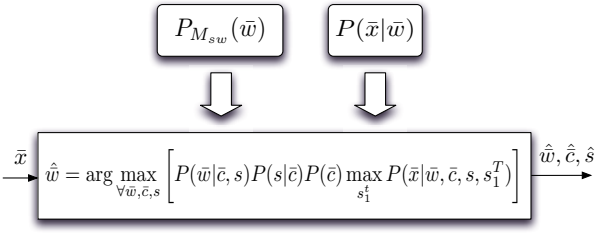

Fig. 4. Integrated architecture for an ASR that considers a M_{sw} model

the recognition process is not guided by the LM of choice, so the obtained result is limited by the best result a M_w model could provide using a word graph.

On the other hand and taking advantage of the use of stochastic finite state models, we propose in this work to integrate complex LMs into a one step decoder as shown in Fig. 4. In this architecture the decoder was modified to be able to integrate the M_{sw} model in the recognition process.

The M_{sw} model can be represented by different SFSA, a SFSA that captures the relations among the classes and N_c (where N_c is the size of the class vocabulary) additional SFSA to consider the relations among the words inside each class. Under the approach proposed in this work an "on-the-fly" composition of the automata could be done at decoding time in the same way the composition of the automata associated to lexical and word-based language model (M_w) is carried out in the standard decoder.

Let us show an example to illustrate this method. We assume that Fig. 5 and 6 represent the automata considering the relations among classes and the specific automaton associated to the class c_2 respectively. Fig. 2 shows the search network for this example when M_{sw} is considered.

When the probability of a transition is needed the corresponding automata have to be considered. Specifically the probabilities of the transitions among words (red arrows) are obtained in the way described below:

Let us focus on the word sequence "$ no nada más gracias". There are different paths in the search network associated to the different segmentations and classifications for this word sequence.

If we consider one of those paths, the class sequence "$c_\$ c_2 c_2$" and the segmentation "no nada más - gracias", the associated probability is obtained according to eq. (5), (6) and (7) as follows:

$$\begin{aligned} &P(\$ \text{ no nada más gracias}) = P(c_\$)P(\$|c_\$)P(c_2|c_\$)P(\text{no}|c_2)\cdot \\ &\cdot P(\text{nada}|\text{no},c_2)P(\text{más}|\text{nada},c_2)P(c_2|c_\$ c_2)P(\text{gracias}|c_2) \end{aligned} \tag{9}$$

where $P(\$|c_\$) = 1$ y $P(c_\$) = 1$. $P(c_2|c_\$)$ is the probability of the transition labeled with c_2 in the SFSA of Fig. 5 (red transition). $P(\text{no}|c_2)$ is obtained from the transition labeled with the word "no" in the automaton associated to c_2 class of Fig. 6 (blue transition), $P(\text{nada}|\text{no},c_2)$ is obtained from the green transition

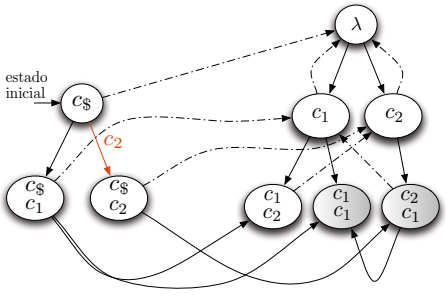

Fig. 5. Class k-TSS model with a value $k_c = 3$

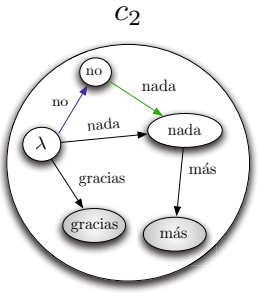

Fig. 6. k-TSS model associated to c_2 class with a value $k_{cw} = 2$

in Fig. 6 and so on. However, to obtain the probability $P(c_2|c_\$c_2)$ it is necessary to consider again the automaton of Fig. 5. That is, the probabilities associated to the automaton of Fig. 5 have to be considered when a final state is reached in the specific automaton of a class and transitions among classes are needed.

Moreover, the use of stochastic finite state framework allows to integrate into a one step decoder the hybrid model M_{hsw} defined like the linear combination of a M_w and a M_{sw} model.

4 Experimental Results

The experiments described in this section were carried out over DIHANA [1] corpus. This corpus consists of 900 human-machine dialogues in Spanish. 225 speakers ask by telephone for information about long-distance train timetables, fares, destinations and services. It comprises 5,590 different sentences to train the LM with a vocabulary of 865 words. The test set (a subset of the whole test set) includes 400 spoken utterances. This task has intrinsically a high level of difficulty due to the spontaneity of the speech and the problematic derived from the acquisition of large amount of training data. Thus, data sparsity is a problem that need to be faced.

Different LMs and methods of integration were evaluated in terms of *Word Error Rate* (WER). First of all we used a word k-TSS LM M_w (with a value $k_w = 3$). This model was integrated into the ASR system and evaluated in terms of WER. Then M_{sw} ($k_c = 2$ and $k_{cw} = 2$) and M_{hsw} models were considered and also integrated into the ASR system using the one-pass decoder. On the other hand, M_{sw} with the same features was considered but in this case the recognition process was carried out by using the decoupled architecture (two-pass decoder). The obtained results are given in Table 1.

As Table 1 shows the integration carried out by means of one-pass decoder provides better results than the integration using a two-step decoder. In fact, M_{sw} model significantly outperforms M_w (improvement of 8.7%), when using

Table 1. WER results for M_w, M_{sw} and M_{hsw} models using different architectures

	M_w	M_{sw} (one-pass)	M_{hsw} (one-pass)	M_{sw} (two-pass)
WER	16.81	15.18	14.23	15.74

the first one, whereas the two pass decoder provides an improvement of 6.4% when rescoring with this M_{sw} model. Moreover, in the two-pass decoder there are two LMs involved (M_w in the first step and M_{sw} in the second step), thus it should be compared with the results obtained with the one pass-decoder and the interpolation of both models (M_w and M_{sw}), that is M_{hsw} model, which provides an improvement of 14.2%. These differences in the system performance could occur due to the fact that the more complex LM (M_{sw}) guides the recognition process when the integrated architecture is considered, while the recognition is guided by the M_w model in the decoupled architecture. Thus, the obtained results proof that a way of integrating the LMs into the ASR system is needed in order to evaluate the system performance for different LMs.

5 Conclusions

In this work we explore different methods to integrate a hierarchical LM based on classes of phrases into an ASR system. The LM is defined within the Stochastic Finite State framework, thus it can be represented by means of different SFSA. The integration is carried out employing an "on-the-fly" composition of the different SFSA associated to the model. WER results are obtained for this integrated architecture (one-pass decoder) and for a decoupled one (two-pass decoder). The obtained results show that the integrated architecture provides significantly better results than those obtained with the decoupled architecture.

References

1. Benedí, J., Lleida, E., Varona, A., Castro, M., Galiano, I., Justo, R., López, I., Miguel, A.: Design and acquisition of a telephone spontaneous speech dialogue corpus in Spanish: DIHANA. In: Proceedings of LREC 2006, Genoa, Italy (May 2006)
2. Benedí, J.M., Sánchez, J.A.: Estimation of stochastic context-free grammars and their use as language models. Computer Speech & Language 19(3), 249–274 (2005)
3. Caseiro, D., Trancoso, I.: A specialized on-the-fly algorithm for lexicon and language model composition. IEEE Transactions on Audio, Speech & Language Processing 14(4), 1281–1291 (2006)
4. García, P., Vidal, E.: Inference of k-testable languages in the strict sense and application to syntactic pattern recognition. IEEE Transactions on Pattern Analysis and Machine Intelligence 12(9), 920–925 (1990)
5. Jurafsky, D., Wooters, C., Segal, J., Stolcke, A., Fosler, E., Tajchman, G., Morgan, N.: Using a stochastic context-free grammar as a language model for speech recognition. In: Proceedings of ICASSP 1995, pp. 189–192. IEEE Computer Society Press, Detroit (1995)

6. Justo, R., Pérez, A., Torres, M.I.: Impact of the approaches involved on word-graph derivation from the asr system. In: Proceedings of the IbPRIA 2011, Las Palmas de Gran Canaria, Spain, June 8-10 (2011) (to be published in LNCS)
7. Justo, R., Torres, M.I.: Phrase classes in two-level language models for asr. Pattern Analysis & Applications 12(4), 427–437 (2009)
8. Mohri, M., Riley, M.: A weight pushing algorithm for large vocabulary speech recognition. In: Proceedings of INTERSPEECH 2001, Aalborg, Denmark, September 2001, pp. 1603–1606 (2001)
9. Niesler, T., Whittaker, E., Woodland, P.: Comparison of part-of-speech and automatically derived category-based language models for speech recognition. In: ICASSP 1998, Seattle, pp. 177–180 (1998)
10. Pereira, F., Riley, M.D.: Speech recognition by composition of weighted finite automata. In: Finite-State Language Processing, pp. 431–453. MIT Press, Cambridge (1996)
11. Torres, M.I., Varona, A.: k-TSS language models in speech recognition systems. Computer Speech and Language 15(2), 127–149 (2001)
12. Zitouni, I.: Backoff hierarchical class n-gram language models: effectiveness to model unseen events in speech recognition. Computer Speech and Language 21(1), 99–104 (2007)