Language Modeling with Functional Head Constraint for Code Switching Speech Recognition

Ying Li and Pascale Fung
Human Language Technology Center
Department of Electronic and Computer Engineering
The Hong Kong University of Science and Technology
eewing@ee.ust.hk, pascale@ece.ust.hk

Abstract

In this paper, we propose novel structured language modeling methods for code mixing speech recognition by incorporating a well-known syntactic constraint for switching code, namely the Functional Head Constraint (FHC). Code mixing data is not abundantly available for training language models. Our proposed methods successfully alleviate this core problem for code mixing speech recognition by using bilingual data to train a structured language model with syntactic constraint. Linguists and bilingual speakers found that code switch do not happen between the functional head and its complements. We propose to learn the code mixing language model from bilingual data with this constraint in a weighted finite state transducer (WFST) framework. The constrained code switch language model is obtained by first expanding the search network with a translation model, and then using parsing to restrict paths to those permissible under the constraint. We implement and compare two approaches - lattice parsing enables a sequential coupling whereas partial parsing enables a tight coupling between parsing and filtering. We tested our system on a lecture speech dataset with 16% embedded second language, and on a lunch conversation dataset with 20% embedded language. Our language models with lattice parsing and partial parsing reduce word error rates from a baseline mixed language model by 3.8% and 3.9% in terms of word error rate relatively on the average on the first and second tasks respectively. It outperforms the interpolated language model by 3.7% and 5.6% in terms of word error rate relatively, and outperforms the adapted language model by 2.6% and 4.6% relatively. Our proposed approach avoids making early decisions on code-switch boundaries and is therefore more robust. We address the code switch data scarcity challenge by using bilingual data with syntactic structure.

1 Introduction

In multilingual communities, it is common for people to mix two or more languages in their speech. A single sentence spoken by bilingual speakers often contains the main, matrix language and an embedded second language. This type of linguistic phenomenon is called ”code switching” by linguists. It is increasingly important for automatic speech recognition (ASR) systems to recognize code switching speech as they exist in scenarios such as meeting and interview speech, lecture speech, and conversational speech. Code switching is common among bilingual speakers of Spanish-English, Hindi-English, Chinese-English, and Arabic-English, among others. In China, lectures, meetings and conversations with technical contents are frequently peppered with English terms even though the general population is not considered bilingual in Chinese and English. Unlike the thousands and tens of thousands of hours of monolingual data available to train, for example, voice search engines, transcribed code switch data necessary for training language models is hard to come by. Code switch language modeling is therefore an even harder problem than acoustic modeling.

One approach for code switch speech recognition is to explicitly recognizing the code switch points by language identification first using phonetic or acoustic information, before applying speech recognizers for the matrix and embedded languages (Chan et. al, 2004; Shia et. al,

Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), pages 907–916,
October 25-29, 2014, Doha, Qatar. ©2014 Association for Computational Linguistics

2004; Lyu and Lyu, 2008). This approach is extremely error-prone as language identification at each frame of the speech is necessary and any error will be propagated in the second speech recognition stage leading to fatal and irrecoverable errors.

Meanwhile, there are two general approaches to solve the problem of lack of training data for language modeling. In a first approach, two language models are trained from both the matrix and embedded language separately and then interpolated together (Vu et. al, 2012; Chan et. al, 2006). However, an interpolated language model effectively allows code switch at all word boundaries without much of a constraint. Another approach is to adapt the matrix language language model with a small amount of code switch data (Tsai et. al, 2010; Yeh et. al, 2010; Bhuvanagiri and Kopparapu, 2010; Cao et. al, 2010). The effectiveness of adaptation is also limited as positions of code switching points are not generalizable from the limited data. Significant progress in speech recognition has been made by using deep neural networks for acoustic modeling and language model. However, improvement thus gained on code switch speech recognition remains very small. Again, we propose that syntactic constraints of the code switching phenomenon can help improve performance and model accuracy. Previous work of using part-of-speech tags (Zhang et. al, 2008; Vu et al 2012) and our previous work using syntactic constraints (Li and Fung, 2012, 2013) have made progress in this area. Part-of-speech is relatively weak in predicting code switching points. It is generally accepted by linguists that code switching follows the so-called Functional Head Constraint, where words on the nodes of a syntactic sub tree must follow the language of that of the headword. If the headword is in the matrix language then none of its complements can switch to the embedded language.

In this work, we propose two ways to incorporate the Functional Head Constraint into speech recognition and compare them. We suggest two approaches of introducing syntactic constraints into the speech recognition system. One is to apply the knowledge sources in a sequential order. The acoustic model and a monolingual language model are used first to produce an intermediate lattice, then a second pass choose the best result using the syntactic constraints. Another approach uses tight coupling. We propose using structured language model (Chelba and Jelinek, 2000) to build the syntactic structure incrementally.

Following our previous work, we suggest incorporating the acoustic model, the monolingual language model and a translation model into a WFST framework. Using a translation model allows us to learn what happens when a language switches to another with context information. We will motivate and describe this WFST framework for code switching speech recognition in the next section. The Functional Head Constraint is described in Section 3. The proposed code switch language models and speech recognition coupling is described in Section 4. Experimental setup and results are presented in Section 5. Finally we conclude in Section 6.

2 Code Switch Language Modeling in a WFST Framework

As code switch text data is scarce, we do not have enough data to train the language model for code switch speech recognition. We propose instead to incorporate language model trained in the matrix language with a translation model to obtain a code switch language model. We propose to integrate a bilingual acoustic model (Li et. al, 2011) and the code switch language model in a weighted finite state transducer framework as follows.

Suppose X denotes the observed code switch speech vector, w_1^J denotes a word sequence in the matrix language, the hypothesis transcript v_1^I is as follows:

$$\begin{aligned}\hat{v}_1^I &= \arg\max_{v_1^I} P(v_1^I|X) \\ &= \arg\max_{v_1^I} P(X|v_1^I)P(v_1^I) \\ &= \arg\max_{v_1^I} P(X|v_1^I)\sum_{w_1^J} P(v_1^I|w_1^J)P(w_1^J) \\ &\cong \arg\max_{v_1^I} P(X|v_1^I)P(v_1^I|w_1^J)P(w_1^J) \quad (1)\end{aligned}$$

where $P(X|v_1^I)$ is the acoustic model and $P(v_1^I)$ is the language model in the mixed language.

Our code switch language model is obtained from a translation model $P(v_1^I|w_1^J)$ from the matrix language to the mixed language, and the language model in the matrix language $P(w_1^J)$.

Instead of word-to-word translation, the transduction of the context dependent lexicon transfer is constrained by previous words. Assume the transduction depends on the previous n words:

$$\begin{aligned} P(v_1^I|w_1^J) &= \prod_{i=1}^{I} P(v_i|v_1^{i-1}, w_1^i) \\ &\cong \prod_{i=1}^{I} P(v_{i-n+1}^{i-1}|w_{i-n+1}^{i}) \\ &= \prod_{i=1}^{I} \frac{P(v_i, w_i|v_{i-n+1}^{i-1}, w_{i-n+1}^{i-1})}{P(w_i|v_{i-n+1}^{i-1}, w_{i-n+1}^{i-1})} \\ &= \prod_{i=1}^{I} \frac{P(v_i, w_i|v_{i-n+1}^{i-1}, w_{i-n+1}^{i-1})}{P(w_i|\sum_{v_i} v_{i-n+1}^{i-1}, w_{i-n+1}^{i-1})} \end{aligned} \quad (2)$$

There are C-level and H-level search networks in the WFST framework. The C-level search network is composed of the universal phone model P, the context model C, the lexicon L, and the grammar G

$$N = P \circ C \circ L \circ G \quad (3)$$

The H-level search network is composed of the state model H, the phoneme model P, the context model C, the lexicon L, and the grammar G

$$N = H \circ P \circ C \circ L \circ G \quad (4)$$

The C-level requires less memory then the H-level search network. We propose to use a weighted finite state transducer framework incorporating the bilingual acoustic model P, the context model C, the lexicon L, and the code switching language models G_{CS} into a C-level search network for mixed language speech recognition. The output of the recognition result is in the mixed language after projection $\pi(G_{CS})$.

$$N = P \circ C \circ L \circ \pi(G_{CS}) \quad (5)$$

The WFST implementation to obtain the code switch language model G_{CS} is as follows:

$$G_{cs} = \mathcal{T} \circ G \quad (6)$$

where T is the translation model

$$P(\tilde{v}_1^L|w_1^J) = \prod_{l=1}^{L} P_l(\tilde{v}_l|w_l) \quad (7)$$

$P_l(\tilde{v}_l|w_l)$ is the probability of w_l translated into $\tilde{v}_l$.

In order to make use of the text data in the matrix language to recognize speech in the mixed language, the translation model $P(v_1^I|w_1^J)$ transduce the language model in the matrix language to the mixed language.

$$\begin{aligned} P(v_1^I|w_1^J) = \sum_{\tilde{v}_1^L, c_1^L, r_1^K, \tilde{w}_1^K} & P(\tilde{w}_1^K|w_1^J) \\ & \cdot P(r_1^K|\tilde{w}_1^K, w_1^J) \\ & \cdot P(c_1^L, r_1^K, \tilde{w}_1^K, w_1^J) \\ & \cdot P(\tilde{v}_1^K|c_1^L, r_1^K, \tilde{w}_1^K, w_1^J) \\ & \cdot P(v_1^I|\tilde{v}_1^K, r_1^K, \tilde{w}_1^K, w_1^J) \end{aligned} \quad (8)$$

where $P(\tilde{w}_1^K|w_1^J)$ is the word-to-phrase segmentation model, $P(r_1^K|\tilde{w}_1^K, w_1^J)$ is the phrasal reordering model, $P(c_1^L, r_1^K, \tilde{w}_1^K, w_1^J)$ is the chunk segmentation model, $P(\tilde{v}_1^K|c_1^L, r_1^K, \tilde{w}_1^K, w_1^J)$ is the chunk-to-chunk transduction model, $P(v_1^I|\tilde{v}_1^K, r_1^K, \tilde{w}_1^K, w_1^J)$ is the chunk-to-word reconstruction model.

The word-to-phrase segmentation model extracts a table of phrases $\{\tilde{v}_1, \tilde{v}_2, ..., \tilde{v}_K\}$ for the transcript in the embedded language and $\{\tilde{w}_1, \tilde{w}_2, ..., \tilde{w}_K\}$ for the transcript in the matrix language based on word-to-word alignments trained in both directions with GIZA++ (Och and Ney, 2003). The chunk segmentation model performs the segmentation of a phrase sequence $\tilde{w}_1^K$ into L phrases $\{c_1, c_2, ..., c_L\}$ using a segmentation weighted finite-state transducer. Assumes that a chunk c_l is code-switched to the embedded language independently by each chunk, the chunk-to-chunk transduction model is the probability of a chunk to be code switched to the embedded language trained on parallel data. The reconstruction model generates word sequence from chunk sequences and operates in the opposite direction to the segmentation model.

3 Functional Head Constraint

Many linguistics (Abney 1986; Belazi et. al, 1994; Bhatt 1994) have discovered the so-called Functional Head Constraint in code switching. They have found that code switches between a functional head (a complementizer, a determiner, an inflection, etc.) and its complement (sentence, noun-phrase, verb-phrase) do not happen in natural speech. In addition, the Functional Head Constraint is language independent.

In this work, we propose to investigate and incorporate the Functional Head Constraint into code switching language modeling in a WFST framework. Figure 1 shows one of the Functional Head Constraint examples. Functional heads are the roots of the sub trees and complements are part of the sub trees. Actual words are the leaf nodes. According to the Functional Head Constraint, the leave nodes of a sub tree must be in either the matrix language or embedded language, following the language of the functional head. For instance, the third word "東西/something" is the head of the constituents "非常/very 重要的/important 東西/something". These three constituent words cannot be switched. Thus, it is not permissible to code switch in the constituent. More precisely, the language of the constituent is constrained to be the same as the language of the headword. In the following sections, we describe the integration of the Functional Head Constraint and the language model.

We have found this constraint to be empirically sound as we look into our collected code mixing speech and language data. The only violation of the constraint comes from rare cases of borrowed words such as brand names with no translation in the local, matrix language. Borrowed words are used even by monolingual speakers so they are in general part of the matrix language lexicon and require little, if any, special treatment in speech recognition.

In the following sections, we describe the integration of Functional Head Constraint and the language model.

4 Code Switching Language Modeling with Functional Head Constraint

We propose two approaches of language modeling with Functional Head Constraint: 1) lattice-parsing and sequential-coupling (Chapplerler et. al, 1999); 2) partial-parsing and tight-coupling (Chapplerler et. al, 1999). The two approaches will be described in the followed sections.

4.1 Sequential-coupling by Lattice-based Parsing

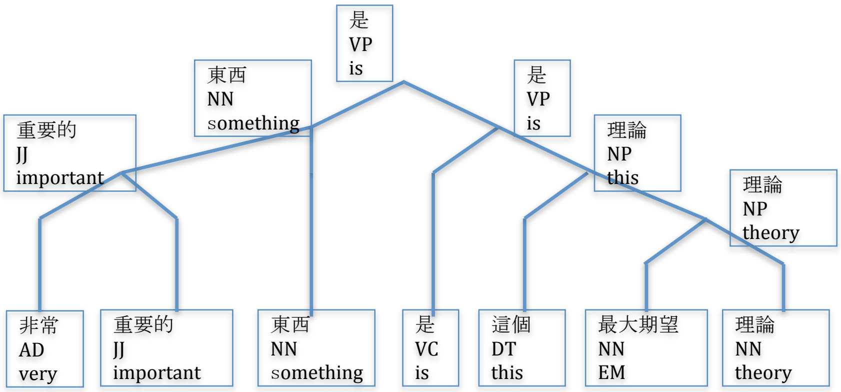
In this first approach, the acoustic models, the code switch language model and the syntactic constraint are incorporated in a sequential order to progressively constrain the search. The acoustic models and the matrix language model are used first to produce an intermediate output. The intermediate output is a lattice in which word sequences are compactly presented. Lattice-based parsing is used to expand the word lattice generated from the first decoding step according to the Functional Head Constraint.

We have reasons to use word lattice instead of N-best hypothesis. The number of hypothesis of word lattice is larger than N-best hypothesis. Moreover, different kinds of errors correspond to the language model would be observed if N-best list is extracted after the first decoding step. The second pass run over the N-best list will prevent the language model with Functional Head Constraint from correcting the errors. In order to obtain a computational feasible number of hypotheses without bias to the language model in the first decoding step, word lattice is used as the intermediate output of the first decoding step.

A Probabilistic Context-Free Grammar (PCFG) parser is trained on Penn Treebank data. The PCFG parser is generalized to take the lattice generated by the recognizer as the input. Figure 2 illustrates a word lattice which is a compact representation of the hypothesis transcriptions of a an input sentence. All the nodes of the word-lattice are ordered by increasing depth.

A CYK table is obtained by associating the arcs with their start and end states in the lattice instead of their sentence position and initialized all the cells in the table corresponding to the arcs (Chaplerler et. al, 1999). Each cell $C_{k,j}$ of the table is filled by a n-tuple of the non-terminal A, the length k and the starting position of the word sequence $w_j...w_{j+k}$ if there exists a PCFG rule $A \rightarrow w_j...w_{j+k}$, where A is a non-terminal which parse sequences of words $w_j...w_{j+k}$. In order to allow all hypothesis transcriptions of word lattice to be taken into account, multiple word sequences of the same length and starting point are initialized in the same cell. Figure 2 mapped the word lattice of the example to the table, where the starting node label of the arc is the column index and the length of the arc is the row index.

The sequential-coupling by lattice-parsing consists of the standard cell-filling and the self-filling steps. First, the cells $C_{k,j}$ and $C_{i-k,j+k}$ are combined to produce a new interpretation for cell $C_{i,j}$. In order to handle the unary context-free production $A \rightarrow B$ and update the cells after the standard cell-filling, a n-tuple of A, i and j is added for each n-tuple of the non-terminal B, the length i and the start j in the cell $C_{i,j}$. The parse trees extracted are associated with the input lattice from the table starting from the non-terminal label of the top cell. After the parse tree is obtained, we re-



Hypotheses: 非常重要的是這個 EM 理論.
非常重要的是這個 EM theory.
非常重要的東西是 this EM theory.
非常重要的東西 is this EM theory.
非常重要的 something is this EM theory. (not permissible)
.
.
.

Figure 1: A Functional Head Constraint example.

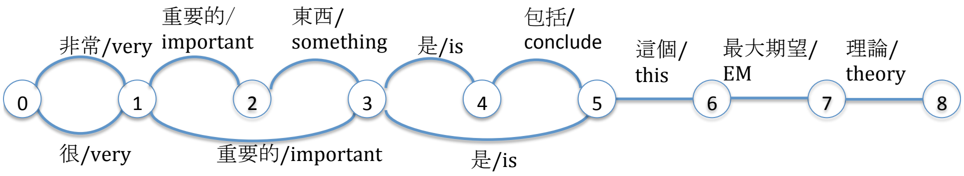
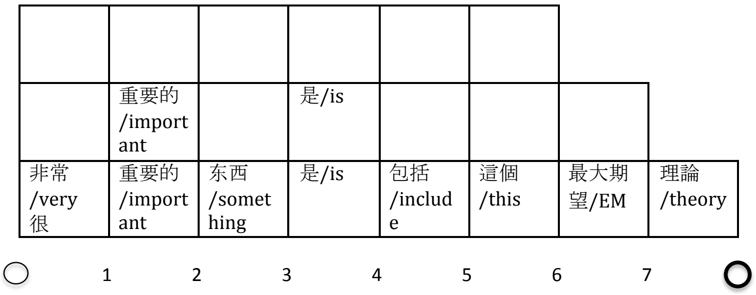


Figure 2: An example word lattice in the matrix language.



	重要的 /import ant		是/is				
非常 /very 很	重要的 /import ant	东西 /somet hing	是/is	包括 /includ e	這個 /this	最大期 望/EM	理論 /theory

Figure 3: The mapping of the example word lattice to the table.

cursively enumerate all its subtrees. Each subtree is able to code-switch to the embedded language with a translation probability $P_l(\tilde{v}_l|w_l)$.

The lattice parsing operation consists of the an encoding of a given word sequence along with a parse tree (W, T) and a sequence of elementary model actions. In order to obtain a correct probability assignment $P(W, T)$ one simply assign proper conditional probabilities to each transition in the weighted finite states.

The probability of a parse T of a word sequence $WP(W, T)$ can be calculated as the product of the probabilities of the subtrees.

$$P(W,T) \quad = \quad \prod_{k=1}^{n+1} [P(wk|W_{k-1}T_{k-1}) \tag{9}$$

Where $W_k = w_0...w_k$ is the first k words in the sentence, and (W_k, T_k) is the word-and-parse k-prefix. The probability of the n-tuple of the non-terminal A, the length i and the starting position j is the probability of the subtree corresponding to A parsing throughout the sequence $w_j...w_{j+i-1}$. The probability of the partial parsing is the product of probabilities of the subtree parses it is made of. The probability of an n-tuple is the maximum over the probabilities of probable parsing path.

The N most probable parses are obtained during the lattice-parsing.

The probability of a sentence is computed by adding on the probability of each new context-free rule in the sentences.

4.2 Tight-coupling by Incremental Parsing

To integrate the acoustic models, language model and the syntactic constraint in time synchronous decoding, an incremental operation is used in this approach. The final word-level probability assigned by our model is calculated using the acoustic models, the matrix language model, the structured language model and the translation model. The structured language model uses probabilistic parameterization of a shift-reduce parse (Chelba and Jelinek, 2000). The tight-coupled language model consists of three transducers, the word predictor, the tagger and the constructor. As shown in Figure 3, $W_k = w0...wk$ is the first k words of the sentence, T_k contains only those binary subtrees whose leaves are completely included in W_k, excluding w_0 =<s>. Single words along with their POS tag can be regarded as root-only trees. The exposed head h_k is a pair of the headword of the constituent W_k and the non-terminal label. The exposed head of single words are pairs of the words and their POS tags.

Given the word-and-parse (k-1)-prefix $W_{k-1}T_{k-1}$, the new word w_k is predicted by the word-predictor $P(w_k|W_{k-1}T_{k-1})$. Taking the word-and-parse $k-1$-prefix and the next word as input, the tagger $P(t_k|w_k, W_{k-1}T_{k-1})$ gives the POS tag t_k of the word w_k. Constructor $P(p_i^k|W_kT_k)$ assigns a non-terminal label to the constituent W_{k+1}. The headword of the newly built constituent is inherited from either the headword of the constituent W_k or the next word w_{k+1}.

$$\begin{aligned}
&P(w_k|W_{k-1}T_{k-1})\\
&= P(w_k|[W_{k-1}T_{k-1}])\\
&= P(w_k|h_0, h_{-1}) && (10)\\
&P(t_k|w_k, W_{k-1}T_{k-1})\\
&= P(t_k|w_k, [W_{k-1}T_{k-1}])\\
&= P(t_k|w_k, h_0.tag, h_{-1}.tag) && (11)\\
&P(p_i^k|W_kT_k)\\
&= P(p_i^k|[W_kT_k])\\
&= P(p_i^k|h_0, h_1) && (12)
\end{aligned}$$

The probability of a parse tree T $P(W, T)$ of a word sequence W and a complete parse T can be calculated as:

$$\begin{aligned}
P(W,T) \quad = \quad & \prod_{k=1}^{n+1} [P(w_k|W_{k-1}T_{k-1})\\
& P(t_k|W_{k-1}T_{k-1}, w_k)\\
& P(T_k|W_{k-1}T_{k-1}, w_k, t_k)](13)
\end{aligned}$$

$$\begin{aligned}
&P(T_{k-1}^k|W_{k-1}T_{k-1}, w_k, t_k)\\
&= \prod_{i=1}^{N_k} P(p_k|W_{k-1}T_{k-1}, w_k, t_k, p_1^k...p_{i-1}^k)
\end{aligned} \tag{14}$$

Where w_k is the word predicted by the word-predictor, t_k is the POS tag of the word w_k predicted by the tagger, $W_{k-1}T_{k-1}$ is the word-parse (k - 1)-prefix, T_{k-1}^k is the incremental parse structure that generates $T_k = T_{k-1}||T_{k-1}^k$ when attached to T_{k-1}; it is the parse structure built on top of T_{k-1} and the newly predicted word wk; the $||$ notation stands for concatenation; N_{k-1} is the number of operations the constructor executes at

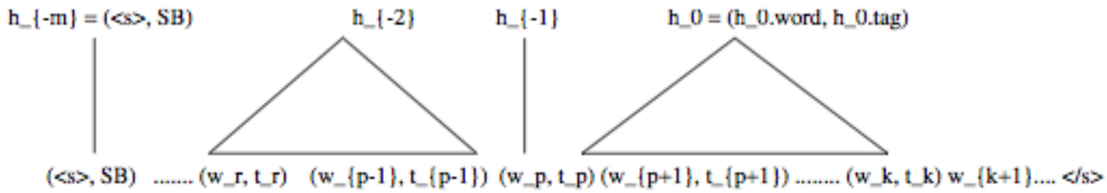

h_{-m} = (<s>, SB)
h_{-2}
h_{-1}
h_0 = (h_0.word, h_0.tag)
(<s>, SB) .... (w_r, t_r) (w_{p-1}, t_{p-1}) (w_p, t_p) (w_{p+1}, t_{p+1}) .... (w_k, t_k) w_{k+1}.... </s>

Figure 4: A word-and-parse example.

position k of the input string before passing control to the word-predictor (the N_k th operation at position k is the null transition); N_k is a function of T ; p_i^k denotes the i th constructor action carried out at position k in the word string.

The probability models of word-predictor, tagger and constructor are initialized from the Upenn Treebank with headword percolation and binarization. The headwords are percolated using a context-free approach based on rules of predicting the position of the headword of the constituent. The approach consists of three steps. First a parse tree is decomposed to phrase constituents. Then the headword position is identified and filled in with the actual word percolated up from the leaves of the tree recursively.

Instead of the UPenn Treebank-style, we use a more convenient binary branching tree. The parse trees are binarized using a rule-based approach.

The probability models of the word-predictor, tagger and constructor are trained in a maximization likelihood manner. The possible POS tag assignments, binary branching parse, non-terminal labels and the head-word annotation for a given sentence are hidden. We re-estimate them using EM algorithm.

Instead of generating only the complete parse, all parses for all the subsequences of the sentence are produced. The headwords of the subtrees are code switched to the embedded language with a translation probability $P_l(\tilde{v}_l|w_l)$ as well as the leaves.

4.3 Decoding by Translation

Using either lattice parsing or partial parsing, a two-pass decoding is needed to recognize code switch speech. A computationally feasible first pass generates an intermediate result so that the language model with Functional Head constraint can be used in the second pass. The first decoding pass composes of the transducer of the universal phoneme model P, the transducer C from context-dependent phones to context-independent phones, the lexicon transducer L which maps context-independent phone sequences to word strings and the transducer of the language model G. A T3 decoder is used in the first pass.

$$ASR_1 = P \circ C \circ L \circ G \qquad (15)$$

Instead of N-best list, word lattice is used as the intermediate output of the first decoding step.

The language model G_{CS} of the transducer in the second pass is improved from G by composing with the translation model $P_l(\tilde{v}_l|w_l)$. Finally, the recognition transducer is optimized by determination and minimization operations.

$$ASR_2 = P \circ C \circ min(det(L \circ min(det(\pi(G_{CS}))))) \qquad (16)$$

5 Experiments

5.1 Experimental Setup

The bilingual acoustic model used for our mixed language ASR is trained from 160 hours of speech from GALE Phase 1 Chinese broadcast conversation, 40 hours of speech from GALE Phase 1 English broadcast conversation, and 3 hours of in-house nonnative English data. The acoustic features used in our experiments consist of 39 components (13MFCC, 13MFCC, 13 MFCC using cepstral mean normalization), which are analyzed at a 10msec frame rate with a 25msec window size. The acoustic models used throughout our paper are state-clustered crossword tri-phone HMMs with 16 Gaussian mixture output densities per state. We use the phone set consists of 21 Mandarin standard initials, 37 Mandarin finals, 6 zero initials and 6 extended English phones. The pronunciation dictionary is obtained by modifying Mandarin and English dictionaries using the phone set. The acoustic models are reconstructed

Table 1: *Code switching point detection evaluation (Precision/Recall/F-measure)*

	Lecture speech	Lunch conversation
MixedLM	0.61/0.64/0.64	0.54/0.63/0.58
InterpolatedLM	0.62/0.66/0.64	0.55/0.63/0.58
AdaptedLM	0.63/0.71/0.67	0.54/0.63/0.58
Sequential coupling	0.66/0.71/0.68	0.55/0.70/0.61
Tight coupling	**0.68/0.71/0.70**	**0.56/0.70/0.62**

by decision tree tying. We also collected two speech databases with Chinese to English code switching - namely, 20 hours of lecture speech corpus (Data 1) and 3 hours of lunch conversation corpus (Data 2). 18 hours of Data 1 is used for acoustic model adaptation and 1 hour of data are used as the test set (Test 1). 2 hours of Data 2 containing 2389 utterances is used to adapt the acoustic model and 280 utterances are used as the test set (Test 2). To train the parser, we use Chinese Treebank Version 5.0 which consists of 500 thousand words and use the standard data split (Petrov and Klein, 2007).

For the language models, transcriptions of 18 hours of Data 1 are trained as a baseline mixed language model for the lecture speech domain. 250,000 sentences from Chinese speech conference papers, power point slides and web data are used for training a baseline Chinese matrix language model for the lecture speech domain (LM 1). Transcriptions of 2 hours of Data 2 are used as the baseline mixed language model in the lunch conversation domain. 250,000 sentences of the GALE Phase 1 Chinese conversational speech transcriptions are used to train a Chinese matrix language model (LM 2). 250,000 of GALE Phase 1 English conversational speech transcription are used to train the English embedded language model (LM 3). To train the bilingual translation model, the Chinese Gale Phase 1 conversational speech transcriptions are used to generate a bilingual corpus using machine translation. For comparison, an interpolated language model for the lunch conversation domain is trained from interpolating LM 2 with LM 3. Also for comparison, an adapted language model for lecture speech is trained from LM 1 and transcriptions of 18 hours of Data 1. An adapted language mode l for conversation is trained from LM 2 and 2 hours of Data 2. The size of the vocabulary for recognition is 20k words. The perplexity of the baseline language model trained on the code switching speech transcription is 236 on the lecture speech and 279 on the conversation speech test sets.

5.2 Experimental Results

Table 1 reports precision, recall and F-measure of code switching point in the recognition results of the baseline and our proposed language models. Our proposed code switching language models with functional head constraint improve both precision and recall of the code switching point detection on the code switching lecture speech and lunch conversation 4.48%. Our method by tight-coupling increases the F-measure by 9.38% relatively on the lecture speech and by 6.90% relatively on the lunch conversation compared to the baseline adapted language model.

The Table 2 shows the word error rates (WERs) of experiments on the code switching lecture speech and Table 3 shows the WERs on the code switching lunch conversations. Our proposed code switching language model with Functional Head Constraints by sequential-coupling reduces the WERs in the baseline mixed language model by 3.72% relative on Test 1, and 5.85% on Test 2. Our method by tight-coupling also reduces WER by 2.51% relative compared to the baseline language model on Test 1, and by 4.57% on Test 2. We use the speech recognition scoring toolkit (SCTK) developed by the National Institute of Standards and Technology to compute the significance levels, which is based on two-proportion z-test comparing the difference between the recognition results of our proposed approach and the baseline. All the WER reductions are statistically significant. For our reference, we also compare the performance of using Functional Head Constraint to that of using inversion constraint in (Li and Fung, 2012, 2013) and found that the present model reduces WER by 0.85% on Test 2 but gives no improvement on Test 1. We hypothesize that since

Table 2: *Our proposed system outperforms the baselines in terms of WER on the lecture speech*

	Matrix	Embedded	Overall
MixedLM	34.41%	39.16%	35.17%
InterpolatedLM	34.11%	40.28%	35.10%
AdaptedLM	35.11%	38.41%	34.73%
Sequential coupling	33.17%	36.84%	33.76%
Tight coupling	**33.14%**	**36.65%**	**33.70%**

Table 3: *Our proposed system outperforms the baselines in terms of WER on the lunch conversation*

	Matrix	Embedded	Overall
MixedLM	46.4%	48.55%	46.83%
InterpolatedLM	46.04%	49.04%	46.64%
AdaptedLM	46.64%	48.39%	46.20%
Sequential coupling	43.24%	46.27%	43.89%
Tight coupling	**42.97%**	**46.03%**	**43.58%**

Test 1 has mostly Chinese words, the proposed method is not as advantageous compared to our previous work. Another future direction is for us to improve the lattice parser as we believe it will lead to further improvement on the final result of our proposed method.

6 Conclusion

In this paper, we propose using lattice parsing and partial parsing to incorporate a well-known syntactic constraint for code mixing speech, namely the Functional Head Constraint, into a continuous speech recognition system. Under the Functional Head Constraint, code switch cannot occur between the functional head and its complements. Since code mixing speech data is scarce, we propose to instead learn the code mixing language model from bilingual data with this constraint. The constrained code switching language model is obtained by first expanding the search network with a translation model, and then using parsing to restrict paths to those permissible under the constraint. Lattice parsing enables a sequential coupling of parsing then constraint filtering whereas partial parsing enables a tight coupling between parsing and filtering. A WFST-based decoder then combines a bilingual acoustic model and the proposed code-switch language model in an integrated approach. Lattice-based parsing and partial parsing are used to provide the syntactic structure of the matrix language. Matrix words at the leave nodes of the syntax tree are permitted to switch to the embedded language if the switch does not violate the Functional Head Constraint. This reduces the permissible search paths from those expanded by the bilingual language model. We tested our system on a lecture speech dataset with 16% embedded second language, and on a lunch conversation dataset with 20% embedded second language. Our language models with lattice parsing and partial parsing reduce word error rates from a baseline mixed language model by 3.72% to 3.89% relative in the first task, and by 5.85% to 5.97% in the second task. They are reduced from an interpolated language model by 3.69% to 3.74%, and by 5.46% to 5.77% in the first and second task respectively. WER reductions from an adapted language model are 2.51% to 2.63%, and by 4.47% to 4.74% in the two tasks. The F-measure for code switch point detection is improved from 0.64 by the interpolated model to 0.68, and from 0.67 by the adapted model to 0.70 by our method. Our proposed approach avoids making early decisions on code-switch boundaries and is therefore more robust. Our approach also avoids the bottleneck of code switch data scarcity by using bilingual data with syntactic structure. Moreover, our method reduces word error rates for both the matrix and the embedded language.

Acknowledgments

This work is partially supported by grant number RGF 612211 of the Hong Kong Research Grants Council, by 1314159-0PAFT20F003 of the Ping An Research Institute and by 13140910 of the Huawei Noah's Ark Lab.

References

J.J. Gumperz, “Discourse strategies”, Cambridge University Press, 1, 1982.

Coulmas, F., “The handbook of sociolinguistics”, Wiley-Blackwell, 1998.

Vu, N.T. and Lyu, D.C. and Weiner, J. and Telaar, D. and Schlippe, T. and Blaicher, F. and Chng, E.S. and Schultz, T. and Li, H. *A first speech recognition system for Mandarin-English code-switch conversational speech’*, ICASSP, 2012

J.Y.C. Chan and PC Ching and T. Lee and H.M. Meng “Detection of language boundary in code-switching utterances by bi-phone probabilities” Chinese Spoken Language Processing, 2004 International Symposium on, 293–296.

C.J. Shia and Y.H. Chiu and J.H. Hsieh and C.H. Wu “Language boundary detection and identification of mixed-language speech based on MAP estimation””, ICASSP 2004.

D.C. Lyu and R.Y. Lyu “Language identification on code-switching utterances using multiple cues” Ninth Annual Conference of the International Speech Communication Association, 2008.

Tsai, T.L. and Chiang, C.Y. and Yu, H.M. and Lo, L.S. and Wang, Y.R. and Chen, S.H. “A study on Hakka and mixed Hakka-Mandarin speech recognition” Chinese Spoken Language Processing (ISCSLP), 2010 7th International Symposium on, 199–204

Yeh, C.F. and Huang, C.Y. and Sun, L.C. and Lee, L.S. “An integrated framework for transcribing Mandarin-English code-mixed lectures with improved acoustic and language modeling” Chinese Spoken Language Processing (ISCSLP), 2010 7th International Symposium on, 214–219

K. Bhuvanagiri and S. Kopparapu, “An Approach to Mixed Language Automatic Speech Recognition”, Oriental COCOSDA, Kathmandu, Nepal, 2010

Cao, H. and Ching, PC and Lee, T. and Yeung, Y.T. “Semantics-based language modeling for Cantonese-English code-mixing speech recognition Chinese Spoken Language Processing (ISCSLP), 2010 7th International Symposium on,246–250

Chelba, Ciprian, and Frederick Jelinek. ”Structured language modeling.” Computer Speech & Language 14, no. 4 (2000): 283-332.

Imseng, D. and Bourlard, H. and Magimai-Doss, M. and Dines, J., “Language dependent universal phoneme posterior estimation for mixed language speech recognition”, ICASSP, 2011.

Q. Zhang and J. Pan and Y. Yan, “Mandarin-English bilingual speech recognition for real world music retrieval”, ICASSP, 2008.

Bouselmi, G. and Fohr, D. and Illina, I., “Combined acoustic and pronunciation modelling for non-native speech recognition”, Eighth Annual Conference of the International Speech Communication Association, 2007.

Woolford, E., “Bilingual code-switching and syntactic theory”, in Linguistic Inquiry, 14(3):520–536, JSTOR, 1983.

MacSwan, J., “13 Code-switching and grammatical theory”, in The Handbook of Bilingualism and Multilingualism, 323 Wiley-Blackwell, 2012.

Poplack, S. and Sankoff, D., “A formal grammar for code-switching”, in Papers in Linguistics: International Journal of Human Communication, 3–45, 1980.

Moore, Robert C and Lewis, William, “Intelligent selection of language model training data” Proceedings of the ACL 2010 Conference Short Papers, 220–224.

Belazi, Heidi; Edward Rubin; Almeida Jacqueline Toribio “Code switching and X-Bar theory: The functional head constraint”. Linguistic Inquiry 25 (2): 221-37, 1994.

Bhatt, Rakesh M., “Code-switching and the functional head constraint” In Janet Fuller et al. Proceedings of the Eleventh Eastern States Conference on Linguistics. Ithaca, NY: Department of Modern Languages and Linguistics. pp. 1-12, 1995

Chappelier, Jean-C?dric, et al., “Lattice parsing for speech recognition.” TALN 1999.