

WORD/SUB-WORD LATTICES DECOMPOSITION AND COMBINATION FOR SPEECH RECOGNITION

Viet-Bac Le, Sopheap Seng, Laurent Besacier, Brigitte Bigi

LIG Laboratory, UMR 5217
BP 53, 38041 Grenoble Cedex 9, FRANCE

email: {viet-bac.le, sopheap.seng, laurent.besacier, brigitte.bigie}@imag.fr

ABSTRACT

This paper presents the benefit of using multiple lexical units in the post-processing stage of an ASR system. Since the use of sub-word units can reduce the high out-of-vocabulary rate and improve the lack of text resources in statistical language modeling, we propose several methods to decompose, normalize and combine word and sub-word lattices generated from different ASR systems. By using a *sub-word information table*, every word in a lattice can be decomposed into sub-word units. These decomposed lattices can be combined into a common lattice in order to generate a confusion network. This lattices combination scheme results in an absolute syllable error rate reduction of about 1.4% over the sentence MAP baseline method for a Vietnamese ASR task. By comparing with the *N*-best lists combination and voting method, the proposed method works better.

Index Terms - ASR, lattice decomposition, lattices combination, confusion network.

1. INTRODUCTION

An important problem in ASR is to accurately estimate statistical language models from insufficient amount of data, particularly for languages which have a very rich morphology where prefixes and suffixes augment word stems to form words. The problem is that a word is often defined as a string of characters separated by space. Hence, this word definition is not aware of morphological relationships between different words. In practice this leads to a high out-of-vocabulary (OOV) rate. The above problem is then even more pronounced for dialects, due to the fact that additional prefixes, and sometimes suffixes, are informally introduced during the everyday use of language. Additionally, the amount of text data available for these dialects is usually much smaller than for standard languages, which will lead to poor estimates of the language model probabilities, and hence may hurt ASR performance. In the mean time, some languages like Chinese and Vietnamese, for instance, lack word separators. Then, word language models must be estimated from an error-prone word segmentation or they have to be estimated at a sub-word level (syllables, characters) with potentially bad consequences on the word coverage of the *n*-gram models.

What is common between these two types of languages (*rich morphology* or *without word separators*)? One answer is *the use of sub-word units for language modeling*. The aim of this paper is to investigate how these two views of the data (word and sub-word) can be advantageously combined in an ASR system. We propose to work both at the model level (by proposing hybrid vocabularies

with both word and sub-word) as well as at the ASR output level (by proposing a word/sub-word lattices combination). Combining word graphs with sub-word graphs implies an **elegant and correct** way to decompose a word graph into its sub-word version, which is also proposed in this paper. The experiments presented here are made in the context of the Vietnamese language where the space separates syllables instead of words. However, the lattice decomposition method proposed here can be also applied to languages with rich morphology, as recently done by some authors of this paper for Arabic-to-English speech-to-text translation [1].

Some previous works using these sub-word units for language modeling have recently been published for Arabic and Turkish (morphological analysis). Data-driven or fully unsupervised [2] word decomposition algorithms were used like in [3, 4] as well as working on the character level for unsegmented languages like in [5]. **In this paper, we try to make benefit of the multiple units (word, sub-word) for an ASR system.**

This paper is organized as follows: firstly, we present in section 2 the word decomposition problem in the ASR lattice, which is necessary to be able to combine word and sub-word lattices. In section 3, we describe the lattices combination scheme. The experimental framework and results are presented in section 4. Section 5 concludes the work and gives some future perspectives.

2. WORD LATTICE DECOMPOSITION

2.1. From lattices to confusion networks (CN)

Almost all ASR systems aim at maximizing the posterior probability of the word sequence according to given language and acoustic model. This standard approach is called sentence MAP approach. In evaluation step, we commonly used however the WER as a performance metric. Some previous works have shown the advantage of explicit WER minimization approach in an *N*-best list [6] or in a word lattice [7]. In fact, by using confusion network (a specified form of lattice), L. Mangu concluded that word lattice approach outperforms *N*-best list approach because it works in a more accurate representation of the hypothesis space [7].

Figure 1 illustrates an example (in English) of a word lattice outputted by an ASR system and its corresponding CN. In this example, 'CANNOT' and 'CAN' are merged in an alignment in the CN although their durations could be different. This alignment creates a deletion (labeled by 'ε') in the next alignment.

To deal with a language with a rich morphology or without explicit word separators, the use of classical word units in ASR and MT can be replaced by sub-word units like morphemes (case of Arabic) [3] or syllables (case of Vietnamese). Such decomposition can reduce the high OOV rate and improve the lack

of text resources in statistical language modeling. If a sub-word segmenter is already available, applying such decomposition is obvious on word strings (text corpora, N -best list). It is however more problematic when such decomposition must be applied to a word lattice at the output of an ASR system. The problem can be formulated as following: *how the word lattice should be modified when words are segmented into sub-word units?*

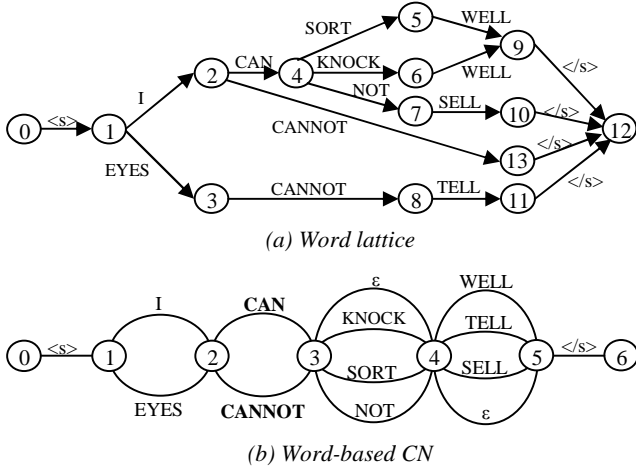


Figure 1: Word lattice and word-based CN.

2.2. Word decomposition in the lattice

A word lattice can be decomposed using the *lattice-tool* (v.1.5.2) of the SRILM toolkit [8]. But with this tool, all the scores of the original word are retained on the first sub-word and the remaining sub-words get 0 scores and 0 duration (the total scores and the sentence posterior probability along the path are thus unchanged). Since the used lattice-to-CN algorithm [7] takes into account the duration of each word, this method might cause some wrong alignments during the converting process. Figure 2 illustrates a sub-word lattice converted by the *lattice-tool* from the word lattice presented in figure 1. Two new nodes 14 and 15 are inserted in the lattice and they are assigned with the same timestamps of nodes 8 and 13, respectively. This decomposition causes a wrong alignment in the CN: ‘NOT’ in the link 13-15 is aligned with ‘WELL’, ‘SELL’ and ‘TELL’ (figure 2.b).

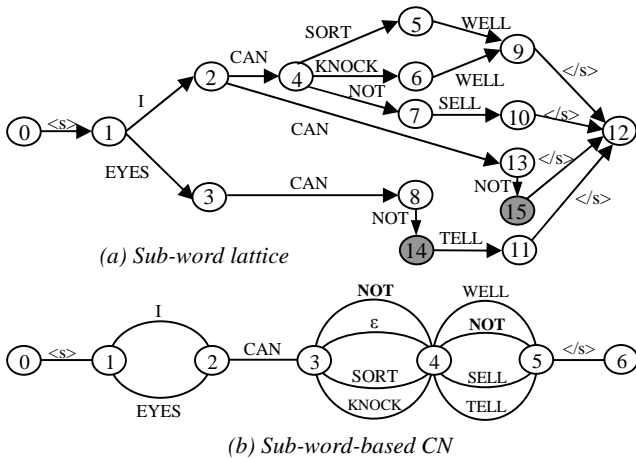


Figure 2: Sub-word lattice converted from word lattice by SRILM *lattice-tool* (-split-multiwords option).

We propose in our work a new algorithm for splitting a word into a sequence of sub-words. Depending on the number of decomposed sub-words, some new nodes with sub-word labels are also inserted to the lattice. The main difference of our algorithm is that the duration and the acoustic score of each new sub-word can be looked up in a *sub-word information table*. If this kind of table is unavailable, the duration and the acoustic score may be approximately distributed as a function of the number of graphemes in each sub-word.

More precisely, the word lattice decomposition algorithm can be described with the following steps:

1. Based on a word/sub-word dictionary or a morphological analyzer, all decomposable words in the word lattice are identified.
2. Each of these words is decomposed into a sequence of sub-words that depends on the number of sub-words in the word. Some new nodes and links are thus inserted in the word lattice.
3. By using a sub-word-based speech recognizer, a sub-word lattice is built for the same utterance. From this lattice, all sub-words with different timestamps, durations and acoustic scores are stored in a *sub-word information table*. For each new decomposed sub-word in the current word lattice, the new acoustic score and the duration is modified according to the appropriate values found in the *sub-word information table*. If such a sub-word recognizer is unavailable or the decomposed sub-words are not found in the *sub-word information table*, the duration and the acoustic score of the initial word are proportionally divided into sub-words as a function of the number of graphemes in the sub-words.
4. An approximation is made for the LM score: the LM score corresponding to the first sub-word of the decomposed word is equal to the LM score of the initial word, while we assume that after the first sub-word, there is only one path to the last sub-word of the word (so the following LM scores are made equal to 0).

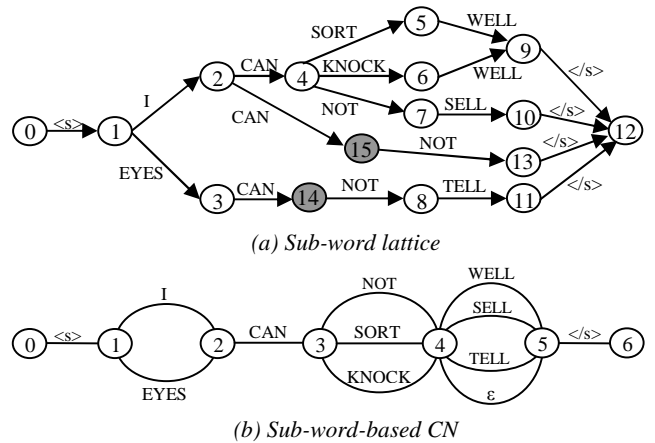


Figure 3: Sub-word lattice obtained with our decomposition algorithm and the associated sub-word-based CN.

Figure 3 presents a new sub-word lattice and the resulting converted CN. The words ‘CANNOT’ in the link 2-13 and 3-8 are decomposed into two pairs of syllables ‘CAN’ and ‘NOT’ by inserting two new nodes in the lattice (node 14 and node 15). If no *sub-word information table* is available, the duration of ‘CAN’ in the new link 2-15 and ‘NOT’ in the new link 15-13 are equal due to the same number of graphemes. The new obtained CN seems more reasonable than the ones shown in figure 1 and figure 2.

3. WORD AND SUB-WORD LATTICES COMBINATION

3.1. Lattices combination

In this section, the use of multiple levels of lexical units (word, morpheme, syllable ...) during the ASR decoding process is proposed. By using different word and sub-word units in the lexicon, different LMs are built and different word and sub-word lattices are thus outputted by different speech recognizers. The question is *what the benefit is, if we merge these different lattices in a common lattice.*

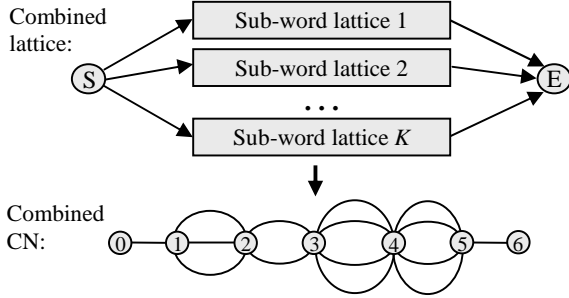


Figure 4: Combined sub-word lattice and the corresponding CN.

Figure 4 presents our combination scheme which can be described with the followings steps:

1. By applying the lattice decomposition algorithm presented above, all words and sub-words in different lattices are decomposed into a unique sub-word set.
2. Create a new starting node S and a new ending node E for the common lattice. Then, we link the node S with starting nodes of all lattices and link ending nodes of all lattices with E. After this step, all lattices are merged into a common lattice. **This operation can also be seen as a “union” of lattices [9].**
3. The obtained lattice is then converted into CN and the consensus hypothesis can be decoded.

Another lattices combination scheme was also presented in [10] where they used an initial step (similar to step 2 of our scheme) to merge lattices together. Then, merged lattice was edited by merging similar links, building new links among nodes and renormalizing acoustic scores from different lattices. The sentence MAP hypothesis was finally decoded from this merged lattice. The difference of our combination scheme is that we do not edit the nodes and the links of the merged lattice because it is converted into CN in order to decode the consensus hypothesis.

3.2. Normalization of posterior probabilities

Since word and sub-word lattices are generated by different systems, a normalization step is needed. Sentence posteriors can be normalized by the sum of the sentence posteriors in the lattice:

$$P(W^k|A) = \frac{P(W^k)P(A|W^k)}{\sum_{k=1}^N P(W^k)P(A|W^k)} \quad (1)$$

where k ranges over the set of hypotheses outputted by the speech recognizer [7]. In a lattice, the total of the sentence posteriors can be computed by the Forward-Backward algorithm.

This normalization step can be used in the lattices combination scheme presented above. Before combining into a common lattice in step 2, word and sub-word lattices are decomposed and then

normalized by equation (1). In next section, performances of the combination scheme with and without normalization are compared.

4. EXPERIMENTS AND RESULTS

This section presents our experiments of lattice decomposition and combination in the post-processing stage of an ASR system for Vietnamese language.

4.1. Experimental framework

4.1.1. ASR system

All recognition experiments use the IBIS decoder of the JANUS toolkit [11] developed at the ISL Laboratories. The model topology is a 3- state left-to-right HMM with 32 Gaussian mixtures per state. The pre-processing of the system consists of extracting a 43 dimensional feature vector every 16 ms. The features consist of 13 MFCCs, energy, the first and second derivatives, and zero-crossing rate. An LDA transformation is used to reduce the feature vector dimensionality to 32. **The ASR performance is measured with Syllable Error Rate (SLER) since Vietnamese word segmentation is not a trivial task and segmentation errors may prevent a fair comparison of different ASR hypotheses.**

4.1.2. Vietnamese Text and Speech Resources

Since syllable plays an important role in Vietnamese language (it is both morphological and phonological base units), a vocabulary of about 6,500 syllables (called V0 since there is no word in this vocabulary) was extracted from a 35k word vocabulary (called V35k). Then the syllable-based and the word-based pronunciation dictionaries were built by applying our *VNPhoneAnalyzer* [12].

Documents were gathered from Internet and filtered for building a *Broadcast news* text corpus. After the data preparation steps, the text corpus has a size of 317 MB, i.e. 55 million words. A syllable-based and a word-based trigram LMs were trained from this text corpus using the SRILM toolkit [8] with a Good-Turing discounting and Katz backoff for smoothing. It is important to note that with this toolkit, the unknown words are removed in our case, since we are in the framework of closed-vocabulary models.

Speech data was extracted from the *VNSpeechCorpus* [12], which was built at LIG and MICA laboratories. In order to train the acoustic models, 13 hours of speech data spoken by 36 speakers were used. The test set contains 277 utterances spoken by 2 speakers different from the training speakers.

4.2. Experimental Results

4.2.1. Word decomposition experiments

In order to test the performance of the word lattice decomposition method, we use the following test protocol: firstly, from the initial syllable vocabulary (V0), we progressively add N most frequent words in the V0. By increasing N from 0 to 35k, we have 10 different hybrid syllable/word vocabularies (called V0, V0.5k, V1k, ...V35k) and 10 different trigram LMs are trained with these vocabularies. Secondly, words in lattices outputted from 10 speech recognizers (called *original lattices*) are decomposed into syllables (called *decomposed lattices*). Finally, these lattices are converted into CNs. Figure 5 shows a comparison of the consensus hypothesis decoded from the *original CN* and the *decomposed CN*. Even if results show that the syllable-based LM is never outperformed by hybrid word/syllable based LMs, the *decomposed CN* works systematically better than *original CN*. It results in an absolute SLER reduction of 0.5% over the *original CN* when the V25k vocabulary is used.

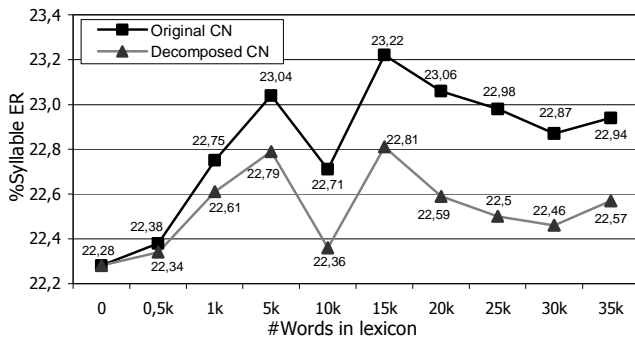


Figure 5: Comparison of the original lattices and the decomposed lattices as a function of the number of words added to the initial syllable vocabulary (V0).

4.2.2. N-best lists and lattices combination experiments

In the first combination experiment, we investigate a simple N -best lists combination method. We decode 20-best hypotheses from the syllable-based (V0) and the word-based (V35k) ASR systems. Every word in these hypotheses is segmented into syllables. Then, we merge both 20-best list from V0 system and 20-best list from V35k system to form a 40-best list. Similar to ROVER [13], we use a voting algorithm based on the number of occurrence of syllables in the N -best list to decode the best hypothesis. Table 1 shows both SLER and Oracle SLER for different hypotheses: sentence MAP baseline, 20-best list and merged 40-best list for both syllable-based and word-based system. We conclude that the merged 40-best list significantly outperforms the MAP hypothesis. Moreover, the same reduction is also obtained in the Oracle SLER.

Method	% Oracle SLER	% SLER
MAP (V0)	-	22.69
MAP (V35k)	-	22.81
20-best Voting (V0)	14.0	22.80
20-best Voting (V35k)	14.8	22.90
40-best Voting (V0+V35k)	11.1	21.40

Table 1: Comparison of sentence MAP baseline hypothesis and N -best voting hypothesis.

In the second combination experiment, the word and sub-word lattices combination scheme presented in section 3 is used. Syllable-based lattice and word-based lattice are first decoded from V0 and V35k system, respectively. Every word in the word-based lattice is then decomposed into syllables. Before converting to CN, both lattices are combined with (called *CN_Norm*) and without (called *CN_NoNorm*) the normalization of the posterior probabilities.

Figure 6 presents an overview of the results: sentence MAP baseline hypotheses for V0 and V35k systems, merged 40-best list presented above, consensus hypotheses decoded from CNs for both systems, consensus hypotheses decoded from *CN_NoNorm* and *CN_Norm*. The results show the benefit of the lattices combination (when done with normalization) compared to the simple voting approach although the difference between these two is not significant. However, both approaches lead to a significant improvement compared to the sentence MAP baseline approach, which shows the interest of using multiple units (word, sub-word) for LM in ASR.

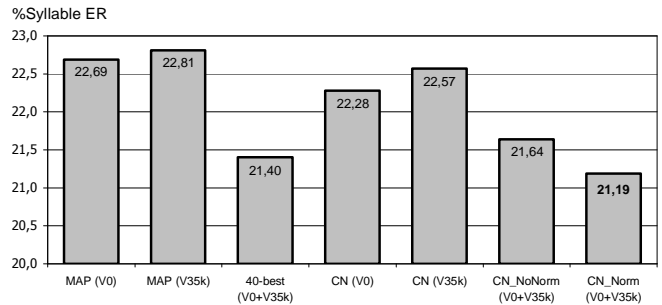


Figure 6: Comparison of the syllable-based lattices, word-based lattices and the combined lattices.

5. CONCLUSIONS

In this paper, a word/sub-word lattices decomposition and combination approach is proposed in order to exploit the use of multiple units in ASR. This approach was tested in an ASR system for Vietnamese. We conclude that our lattices combination method outperformed both sentence MAP baseline and the N -best lists combination methods. Moreover, the lattices decomposition and combination tools are made available by the authors for any person who is interested in. In the future, we plan to apply these methods in Khmer language in which more lexical units (word, syllable, characters cluster and character) can be exploited.

6. REFERENCES

- [1] L. Besacier et al., "The LIG Arabic/English Speech Translation System at IWSLT'07", *IWSLT'07*, Trento, Italy, 2007.
- [2] M. Kurimo et al., "Unsupervised segmentation of words into morphemes - Morpho Challenge 2005: Application to Automatic Speech Recognition", *Interspeech'06*, pp. 1021-1024, Pittsburgh, PA, 2006.
- [3] M. Afify et al., "On the use of morphological analysis for dialectal Arabic Speech Recognition", *Interspeech'06*, pp. 277-280, Pittsburgh, PA, 2006.
- [4] N. Abdillahi et al., "Automatic transcription of Somali language", *Interspeech'06*, pp. 289-292, Pittsburgh, PA, 2006.
- [5] E. Denoual, Y. Lepage, "The character as an appropriate unit of processing for non-segmenting languages", *NLP Annual Meeting*, pp.731-734, Tokyo, Japan, 2006.
- [6] A. Stolcke et al., "Explicit word error minimization in N -best list rescoring", *Eurospeech'97*, pp.163-165, Rhodes, Greece, 1997.
- [7] L. Mangu et al., "Finding Consensus in Speech Recognition: Word Error Minimization and Other Applications of Confusion Networks", *CSL*, vol. 14, no. 4, pp. 373-400, 2000.
- [8] A. Stolcke, "SRILM - An Extensible Language Modeling Toolkit", *ICSLP'02*, vol. 2, pp. 901-904, Denver, CO, 2002.
- [9] M. Mohri, "Finite-State Transducers in Language and Speech Processing", *Computational Linguistics*, vol. 23, no. 2, pp. 269-311, 1997.
- [10] X. Li, R. Singh, R. M. Stern, "Lattice Combination for Improved Speech Recognition", *ICSLP'02*, Denver, CO, 2002.
- [11] H. Soltau et al., "A One Pass-Decoder Based On Polymorphic Linguistic Context", *ASRU'01*, pp. 214-217, Trento, Italy, 2001.
- [12] V. B. Le et al., "Spoken and written language resources for Vietnamese", *LREC'04*, pp. 509-602, Lisbon, Portugal, May 2004.
- [13] J. G. Fiscus, "A Post-Processing System to Yield Reduced Word Error Rates: Recognizer Output Voting Error Reduction (ROVER)", *ASRU'97*, pp. 347-352, Santa Barbara, CA, 1997.