Boosting N-gram Coverage for Unsegmented Languages Using Multiple Text Segmentation Approach

Solomon Teferra Abate
LIG Laboratory,
CNRS/UMR-5217
solomon.abate@imag.fr

Laurent Besacier
LIG Laboratory,
CNRS/UMR-5217
laurent.besacier@imag.fr

Sopheap Seng
LIG Laboratory,
CNRS/UMR-5217
MICA Center, CNRS/UMI-2954
sopheap.seng@imag.fr

Abstract

Automatic word segmentation errors, for languages having a writing system without word boundaries, negatively affect the performance of language models. As a solution, the use of multiple, instead of unique, segmentation has recently been proposed. This approach boosts N-gram counts and generates new N-grams. However, it also produces bad N-grams that affect the language models' performance. In this paper, we study more deeply the contribution of our multiple segmentation approach and experiment on an efficient solution to minimize the effect of adding bad N-grams.

1 Introduction

A language model is a probability assignment over all possible word sequences in a natural language. It assigns a relatively large probability to meaningful, grammatical, or frequent word sequences and a low probability or a zero probability to nonsensical, ungrammatical or rare ones. The statistical approach used in N-gram language modeling requires a large amount of text data in order to make an accurate estimation of probabilities. These data are not available in large quantities for under-resourced languages and the lack of text data has a direct impact on the performance of language models. While the word is usually a basic unit in statistical language modeling, word identification is not a simple task even for languages that separate words by a special character (a white space in general). For unsegmented languages, which have a writing system without obvious word delimiters, the N-grams of words are usually estimated from the text corpus segmented into words employing automatic methods. Automatic segmentation of text is not a trivial task and introduces errors due to the ambiguities in natural language and the presence of out of vocabulary words in the text.

While the lack of text resources has a negative impact on the performance of language models, the errors produced by the word segmentation make those data even less usable. The word N-grams not found in the training corpus could be due not only to the errors introduced by the automatic segmentation but also to the fact that a sequence of characters could have more than one correct segmentation.

In previous article (Seng et al., 2009), we have proposed a method to estimate an N-gram language model from the training corpus on which each sentence is segmented into multiple ways instead of a unique segmentation. The objective of multiple segmentation is to generate more N-grams from the training corpus to use in language modeling. It was possible to show that this approach generates more N-grams (compared to the classical dictionary-based unique segmentation method) that are potentially useful and relevant in language modeling. The application of multiple segmentation in language modeling for Khmer and Vietnamese showed improvement in terms of tri-gram hits and recognition error rate in Automatic Speech Recognition (ASR) systems.

This work is a continuation of our previous work on the use of multiple segmentation. It is conducted on Vietnamese only. A close analysis of N-gram counts shows that the approach has in fact two contributions: boosting the N-gram

Proceedings of the 1st Workshop on South and Southeast Asian Natural Language Processing (WSSANLP), pages 1–7,
the 23rd International Conference on Computational Linguistics (COLING), Beijing, August 2010

counts that are generated with first best segmentation and generating new N-grams. We have also identified that there are N-grams that negatively affect the performance of the language models. In this paper, we study the contribution of boosting N-gram counts and of new N-grams to the performance of the language models and consequently to the recognition performance. We also present experiments where rare or bad N-grams are cut off in order to minimize their negative effect on the performance of the language models.

The paper is organized as follows: section 2 presents the theoretical background of our multiple segmentation approach; in section 3 we point out the set up of our experiment; in section 4 we present the results of our detailed statistical analysis of N-grams generated by multiple segmentation systems. Section 5 presents the evaluation results of our language models for ASR and finally, we give concluding remarks.

2 Multiple Text Segmentation

Text segmentation is a fundamental task in natural language processing (NLP). Many NLP applications require the input text segmented into words before making further progress because the word is considered the basic semantic unit in natural languages. For unsegmented languages segmenting text into words is not a trivial task. Because of ambiguities in human languages, a sequence of characters may be segmented in more than one way to produce a sequence of valid words. This is due to the fact that there are different segmentation conventions and the definition of word in a language is often ambiguous.

Text segmentation techniques generally use an algorithm which searches in the text the words corresponding to those in a dictionary. In case of ambiguity, the algorithm selects the one that optimizes a parameter dependent on the chosen strategy. The most common optimization strategies consist of maximizing the length of words ("longest matching") or minimizing the number of words in the entire sentence ("maximum matching"). These techniques rely heavily on the availability and the quality of the dictionaries and while it is possible to automatically generate a dictionary from an unsegmented text corpus using unsupervised methods, dictionaries are often created manually. The state-of-the-art methods generally use a combination of hand-crafted, dictionary and statistical techniques to obtain a better result. However, statistical methods need a large corpus segmented manually beforehand. Statistical methods and complex training methods are not appropriate in the context of under-resourced languages as the resources needed to implement these methods do not exist. For an under-resourced language, we seek segmentation methods that allow better exploitation of the limited resources. In our previous paper (Seng et al., 2009) we have indicated the problems of existing text segmentation approaches and introduced a weighted finite state transducer (WFST) based multiple text segmentation algorithm.

Our approach is implemented using the AT & T FSM Toolkit (Mohri et al., 1998). The algorithm is inspired with the work on the segmentation of Arabic words (Lee et al., 2003). The multiple segmentation of a sequence of characters is made using the composition of three controllers. Given a finite list of words we can build a finite state transducer M (or word transducer) that, once composed with an acceptor I of the input string that represent a single character with each arc, generates a lattice of the words that represent all of the possible segmentations. To handle out-of-vocabulary entries, we make a model of any string of characters by a star closure operation over all the possible characters. Thus, the unknown word WFST can parse any sequence of characters and generate a unique *unk* word symbol. The word transducer can, therefore, be described in terms of the WFST operations as $M = (WD \cup UNK)+$ where WD is a WFST that represents the dictionary and UNK represents the unknown word WFST. Here, $\cup$ and + are the union and Kleene "+" closure operations. A language model L is used to score the lattice of all possible segmentations obtained by the composition of our word transducer M and the input string I. A language model of any order can be represented by a WFST. In our case, it is important to note that only a simple uni-gram language model is used. The uni-gram model is estimated from a small training corpus segmented automatically into words using a dictionary based method. The composition of the sequence of input string I

with the word transducer M yields a transducer that represents all possible segmentations. This transducer is then composed with the language model L, resulting in a transducer that represents all possible segmentations for the input string I, scored according to L. The highest scoring paths of the compound transducer is the segmentation m that can be defined as:

$$P(m) = maxP(m_k)$$

The segmentation procedure can then be expressed formally as:

$$m = bestpath(I \circ M \circ L)$$

where ◦ is the composition operator. The N-best segmentations are obtained by decoding the final lattice to output the N-best highest scoring paths and will be used for the N-gram count.

3 Experimental Setup

3.1 Language Modeling

First, it is important to note that Vietnamese texts are naturally segmented into syllables (not words). Each syllable tends to have its own meaning and thus a strong identity. However, the Vietnamese monosyllable is not automatically a word as we would define a word in English. Often, two syllables go together to form a single word, which can be identified by the way it functions grammatically in a sentence. To have a word-based language model, word segmentation would, therefore, be a must in Vietnamese.

A Vietnamese training corpus that contains 3 millions sentences from broadcast news domain has been used in this experiment. A Vietnamese dictionary of 30k words has been used both for the segmentation and counting the N-grams. Therefore, in the experiments, the ASR vocabulary always remains the same and only the language model is changing. The segmentation of the corpus with dictionary based, "longest matching" unique segmentation method gives a corpus of 46 millions words. A development corpus of 1000 sentences, which has been segmented automatically to obtain 44k words, has been used to evaluate the tri-gram hits and the perplexity. The performance of each language model produced will be evaluated in terms of the tri-gram hits and perplexity on the development corpus and in terms of ASR performance on a separate speech test set (different from the development set).

First of all, a language model named lm_1 is trained using the SRILM toolkit (Stolcke 2002) from the first best segmentation (Segmul1), which has the highest scoring paths (based on the transducer explained in section 2) of each sentence in the whole corpus. Then, additional language models have been trained using the corpus segmented with N-best segmentation: the number of N-best segmentations to generate for each sentence is fixed to 2, 5, 10, 50, 100 and 1000. The resulting texts are named accordingly as Segmul2, Segmul5, Segmul10, Segmul50, Segmul100, Segmul1000. Using these as training data, we have developed different language models. Note that a tri-gram that appears several times in multiple segmentations of a single sentence has a count set to one.

3.2 ASR System

Our automatic speech recognition systems use the CMU's Sphinx3 decoder. The decoder uses Hidden Markov Models (HMM) with continuous output probability density functions. The model topology is a 3-state, left-to-right HMM with 16 Gaussian mixtures per state. The preprocessing of the system consists of extracting a 39 dimensional features vector of 13 MFCCs, the first and second derivatives. The CMU's SphinxTrain has been used to train the acoustic models used in our experiment.

The Vietnamese acoustic modeling training corpus is made up of 14 hours of transcribed read speech. More details on the automatic speech recognition system for Vietnamese language can be found in (Le et al., 2008). While the evaluation metric WER (Word Error Rate) is generally used to evaluate and compare the performance of the ASR systems, this metric does not fit well for unsegmented languages because the errors introduced during the segmentation of the references and the output hypothesis may prevent a fair comparison of different ASR system outputs. We, therefore, used the Syllable Error Rate (SER) as Vietnamese text is composed of syllables naturally separated by white space. The automatic speech recognition is done on a test corpus of 270 utterances (broadcast news domain).

4 Statistical Analysis of N-grams in Multiple Text Segmentation

The change in the N-gram count that results from multiple segmentation is two fold: first there is a boosting of the counts of the N-grams that are already found with the first best segmentation, and secondly new N-grams are added. As we have made a closed-vocabulary counting, there are no new uni-grams resulting from multiple segmentation. For the counting, the SRILM toolkit (Stolcke 2002) is used setting the -gtnmin option to zero so that all the N-gram counts can be considered.

Figure 1 shows the distribution of tri-gram counts for the unique and multiple segmentation of the training corpus. It can be seen that the majority of the tri-grams have counts in the range of one to three.

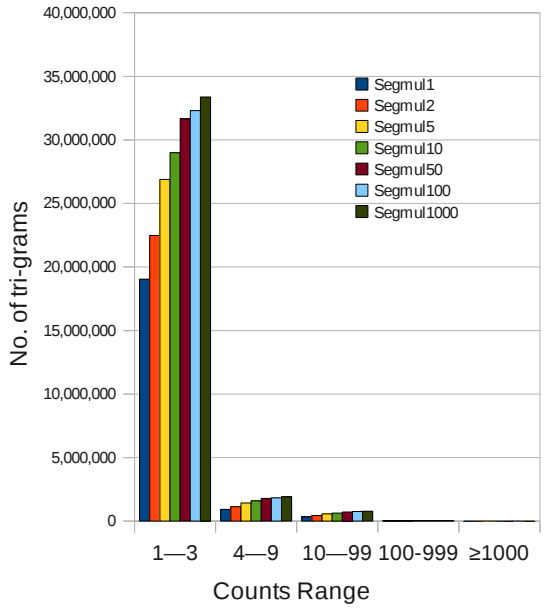

Figure 1: Distribution of tri-gram counts

The boosting (the counts of the tri-grams that are already found with the first best segmentation) effect of the multiple segmentation is indicated in table 1. We can see from the table that Segmul2, for example, reduced the number of rare tri-grams (count range 1-3) from 19.04 to 16.15 million. Consequently, the ratio of rare tri-grams to all tri-grams that are in Segmul1 is reduced from 94% (19.04/20.31*100) of Segmul1 only to 79% (15.96/20.31*100) by the boosting effect of Segmul1000, which increased the number of tri-grams with count range of 4-9 from 0.91M to 3.34M. This implies, in the context of under-resourced languages, that multiple segmentation is boosting the N-gram counts. However, one still has to verify if this boosting is relevant or not for ASR.

Multiple Seg.	Counts Range				
	1–3 (M)	4-9 (M)	10-99 (M)	100-999 (M)	≥1000 (M)
Segmul1	19.04	0.91	0.34	0.016	0.00054
Segmul2	16.15	3.23	0.89	0.043	0.0017
Segmul5	16.06	3.28	0.92	0.045	0.0017
Segmul10	16.03	3.30	0.93	0.045	0.0017
Segmul50	15.99	3.33	0.95	0.046	0.0017
Segmul100	15.98	3.33	0.95	0.046	0.0017
Segmul1000	15.96	3.34	0.96	0.046	0.0017

Table 1. boosting tri-gram counts

We have also analyzed the statistical behavior of the newly added tri-grams with regard to their count distribution (see figure 2). As we can see from the figure, the distribution of the new tri-grams is somehow similar to the distribution of the whole tri-grams that is indicated in figure 1.

As shown in table 2, the total number of newly added tri-grams is around 15 millions. We can see from the table that the rate of new tri-gram contribution of each segmentation increases as N increases in the N-best segmentation. However, as it is indicated in figure 2, the major contribution is in the area of rare tri-grams.

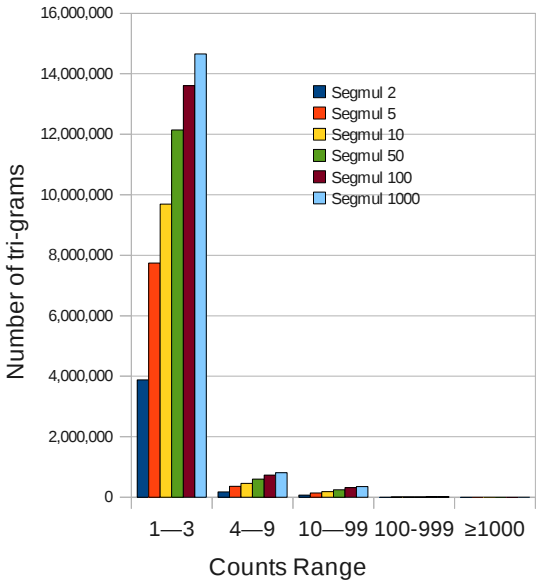

Figure 2: Distribution of new tri-gram counts

Mul. Segmentation	No.	%
Segmul2	4,125,881	26,05
Segmul5	8,249,684	52,09
Segmul10	10,355,433	65,39
Segmul50	13,002,700	82,11
Segmul100	14,672,827	92,65
Segmul1000	15,836,120	100,0

Table 2. tri-gram contribution of multiple segmentation

5 Experimental Results

In this section we present the various language models we have developed and their performance in terms of perplexity, tri-gram hits and ASR performance (syllable error rate).

We use the results obtained with the method presented in (Seng et al., 2009) as baseline. This method consists in re-estimating the N-gram counts using the multiple segmentation of the training data and add one to the count of a tri-gram that appears several times in multiple segmentations of a single sentence. These baseline results are presented in Table 3. The results show an increase of the tri-gram coverage and slight improvements of the ASR performance.

Language Models	3gs(M)	3g hit(%)	Ppl	SER
Lm_1	20.31	46.9	126.6	27
lm_2	24.06	48.6	118.1	**26.2**
Lm_5	28.92	49.2	125.9	27
Lm_10	32.82	49.4	129.0	26.5
Lm_50	34.20	49.7	133.4	26.7
lm_100	34.93	49.7	134.8	26.9
lm_1000	36.11	49.88	137.7	27.3

Table 3. Results of experiments using the baseline method presented in (Seng et al., 2009)

5.1 Separate effect of boosting tri-gram counts

To see the effect of boosting tri-gram counts only, we have updated the counts of the tri-grams obtained from the 1-best segmentation (baseline approach) by the tri-gram counts of different multiple segmentations. Note that no new tri-grams are added here, and we evaluate only the effect and, therefore, the tri-gram hit remains the same as that of lm_1.

We have then developed different language models using the uni-gram and bi-gram counts of the first best segmentation and the updated trigram counts after multiple segmentation. The performance of the language models have been evaluated in terms of perplexity and their contribution to the performance improvement of a speech recognition system. We have observed (detailed results are not reported here) that boosting only the tri-gram counts has not contributed any improvement in the performance of the language models. The reason is probably due to the fact that simply updating tri-gram counts without updating the uni-grams and the bi-grams lead to a biased and inefficient LM.

5.2 Separate effect of new tri-grams

To explore the contributions of only newly added tri-grams, we have added their counts to the N-gram counts of Segmul1. It is important to note that the model obtained in that case is different from the baseline model whose results are presented in Table 3 (the counts of the tri-grams already found in the unique segmentation are different between models). As it is presented in table 4, including only the newly added tri-grams consistently improved tri-gram hits, while the improvement in perplexity stopped at Segmul10. Moreover, the use of only new tri-grams do not reduce the speech recognition error rate.

Language Models	3gs (M)	3g hit(%)	ppl	SER
lm_1	20.3	46.9	126.6	27
lm_2_new	24.4	48.7	119.1	26.9
lm_5_new	28.6	49.0	122.5	27.8
lm_10_new	30.7	49.2	124.2	27.9
lm_50_new	33.3	49.4	126.8	27.8
lm_100_new	35	49.8	127.8	28
lm_1000_new	36.1	49.9	129.7	27.9

Table 4. Contributions of new tri-grams

5.3 Pooling unique and multiple segmentation models

We have developed language models by pooling unique and multiple segmentation models altogether. For instance, all the N-grams of lm_5 multiple segmentation are pooled with all N-grams of lm_1 unique segmentation before estimating the language model probabilities. In other words, ngram-count command is used with multiple count files. The results are presented in table 5.

As it can be noted from table 5, we have got a significant improvement in all the evaluation criteria as compared with the performance of lm_1 that has perplexity of 126.6, tri-gram hit

of 46.91% and SER of 27. The best result obtained (25.4) shows a 0.8 absolute SER reduction compared to the best result presented in (Seng et al., 2009).

Language Models	3gs (M)	3g hit(%)	ppl	SER
lm_1	20.31	46.9	126.6	27
lm_2+lm_1	24.4	48.7	120.9	**25.4**
lm_5+lm_1	29.12	49.2	123.2	26.2
lm_10+lm_1	31.4	49.4	124.2	26
lm_50+lm_1	34.3	49.7	126	26
lm_100+lm_1	35	49.8	126.5	26.2
lm_1000+lm_1	36.2	49.9	128	26.2

Table 5. Performance with pooling

5.4 Cutting off rare tri-grams

With the assumption that bad N-grams occur rarely, we cut off rare tri-grams from the counts in developing language models. We consider all tri-grams with a count of 1 to be rare. Our hope, here, is that using this cut off we will remove bad N-grams introduced by the multiple segmentation approach, while keeping correct new N-grams in the model. Table 6 shows the performance of the language models developed with or without tri-gram cut off for the baseline method (the results presented on the lines indicating All3gs are the same as the ones presented in Table 3) .

Language models		Evaluation Criteria			
		3gs (M)	3g hit (%)	ppl	SER
lm_1	All 3gs	20.31	46.91	126.6	27
	Cut off	4.17	38.09	129.3	26.6
lm_2	All 3gs	24.06	48.6	118.1	26.2
	Cut off	5.11	39.6	121.0	26.7
lm_5	All 3gs	28.92	49.2	125.9	27
	Cut off	6.4	40.11	129.2	26.6
lm_10	All 3gs	32.82	49.41	129.0	26.5
	Cut off	6.98	40.27	132.4	26.6
lm_50	All 3gs	34.20	49.68	133.4	26.7
	Cut off	7.8	40.51	136.9	26.9
lm_100	All 3gs	34.93	49.74	134.8	26.9
	Cut off	7.98	40.59	138.4	26.8
lm_1000	All 3gs	36.11	49.88	137.7	27.3
	Cut off	8.33	40.71	141.3	26.8

Table 6. Performance with cut off.

The result shows that cutting off reduced the number of tri-grams highly (4 tri-grams over 5 are removed in that case). It, therefore, reduces the size of the language models significantly. Although the results obtained are not conclusive, a reduction of recognition error rate has been observed in four out of the seven cases while the perplexity increased and the tri-gram hits decreased in all cases.

5.5 Hybrid of pooling and cutting off methods

As it has been already indicated, cutting off increased the perplexity of the language models and decreased the tri-gram hits. To reduce the negative effect of cutting off on tri-gram hits and perplexity, we have developed language models using both pooling and cut off methods. We then cut off tri-grams of count 1 from the pooled N-grams. The result, as presented in table 7, shows that we can gain significant reduction in recognition error rate and improvement in tri-gram hits as compared to lm_1 that is developed with cut off, even if no improvement in perplexity is observed.

The best result obtained (25.9) shows a 0.3 absolute SER reduction compared to the best system presented in (Seng et al., 2009).

Language Models	3gs (M)	3g hit (%)	ppl	SE R
lm_1 (no cutoff)	20.3	46.9	126.6	27
lm_1 (cutoff)	4.2	38.1	129.3	26.6
lm_2+lm_1 (cutoff)	5.2	39.7	126.4	26.8
lm_5+lm_1 (cutoff)	6.4	40.2	129.5	**25.9**
lm_10+lm_1 (cutoff)	7.0	40.3	131.1	26.3
lm_50+lm_1 (cutoff)	7.8	40.5	133.5	26.4
lm_100+lm_1 (cutoff)	8.0	40.6	134.3	26.4
lm_1000+lm_1 (cutoff)	8.3	40.7	161.5	26.7

Table 7. Performance with hybrid method

6 Conclusion

The two major contributions of multiple segmentation are generation of new N-grams and boosting N-gram counts of those found in first best segmentation. However, it also produces bad N-grams that affect the performance of language models. In this paper, we studied the contribution of multiple segmentation approach more deeply and conducted experiments on efficient solutions to minimize the effect of adding bad N-grams. Since only boosting the tri-gram counts of first best segmentation and adding only new tri-grams did not reduce recognition error rate, we have proposed to pool all N-grams of N-best segmentations to that of first best segmentation and got a significant improvement in perplexity and tri-gram hits from

which we obtained the maximum (0.8 absolute) reduction in recognition error rate.

To minimize the effect of adding bad N-grams, we have cut off rare tri-grams in language modeling and got reduction in recognition error rate. The significant reduction of tri-grams that resulted from the cut off revealed that the majority of tri-grams generated by multiple segmentation have counts 1. Cutting off such a big portion of the trigrams reduced tri-gram hits and as a solution, we proposed a hybrid of both pooling and cutting off tri-grams from which we obtained a significant reduction in recognition error rate.

It is possible to conclude that our methods make the multiple segmentation approach more useful by minimizing the effect of bad N-grams that it generates and utilizing the contribution of different multiple segmentations.

However, we still see rooms for improvement. A systematic selection of new tri-grams (for example, based on the probabilities of the N-grams and/or application of simple linguistic criteria to evaluate the usefulness of new tri-grams), with the aim of reducing bad tri-grams, might lead to performance improvement. Thus, we will do experiments in this line. We will also apply these methods to other languages, such as Khmer.

References

Lee, Young-Suk, Papineni, Kishore, Roukos, Salim Emam, Ossama and Hassan, Hany. 2003. *Language model based arabic word segmentation.* In Proceedings of the ACL'03, pp. 399–406.

Le, Viet-Bac, Besacier, Laurent, Seng, Sopheap, Bigi, Brigite and Do, Thi-Ngoc-Diep. 2008. *Recent advances in automatic speech recognition for vietnamese.* SLTU'08, Hanoi Vietnam.

Mohri, Mehryar, Fernando C. N. Pereira, and Michael Riley, "A rational design for a weighted finite-state transducer library," in Lecture Notes in Computer Science. Springer, 1998, pp. 144–158.

Seng, Sopheap, Besacier, Laurent, Bigi, Brigitte, Castelli, Eric. 2009. *Multiple Text Segmentation for Statistical Language Modeling.* InterSpeech, Brighton, UK,

Stolcke, Andreas. 2002. SRILM: an extensible language modeling toolkit. Proceedings of International Conference on Spoken Language Processing, volume II, 901–904 . 129.88.65.115