Data augmentation on spontaneous Indonesian automatic speech recognition using statistical machine translation

P N Hadiwinoto and D P Lestari
School of Electrical Engineering and Informatics, Bandung Institute of Technology
Bandung, Indonesia
E-mail: patrick.nugroho64@gmail.com

Abstract. Language model plays an important role in decoding process of the automatic speech recognition. The accuracy of spontaneous speech recognition is still very low compared to dictated speech of the Indonesian automatic speech recognition. It is due to the lack of the number of spontaneous data. Collecting spontaneous data is also difficult to do, so one of the candidate solutions is to augment data from existing spontaneous data. In this research, experiments are conducted on language models to improve the accuracy of spontaneous Indonesian speech recognition by conducting data augmentation. Data augmentation in this research is done by using statistical machine translation named ‘Moses’. Language modeling technique used here is n-gram. GMM-HMM is used for acoustic modeling. First, spontaneous text corpus is added to the text corpus, then the data augmentation is conducted. When the language model is formed from the addition of a spontaneous text corpus, there is an increase in accuracy of 3.59% relative to the baseline. When data augmentation is done on language model there is an increase in accuracy of 2.74% relative to the baseline. However, this decrease is considered not significant compared to the effort required in collecting spontaneous data manually.

1. Introduction
Language model plays an important role in increasing the accuracy of automatic speech recognition. Language model maps the probability of the occurrence of a word. Language model is focusing on the grammatical aspect of a language rather than the condition of the speech signal. Spontaneous speech has caught the attention of many automatic speech recognition (ASR) researchers recently. It is because of the recognition accuracy of spontaneous speech is extremely poor compared to dictated speech. Word Error Rate (WER) shows how much error occurred in recognizing speech. It is counted based on how many words are wrongly recognized by the system compared to the number of words in the testing data. Indonesian automatic speech recognition (ASR) researches have been focusing on improving the accuracy of spontaneous speech recognition [1, 2]. Nowadays, Indonesian ASR used n-gram to build language model [1, 2]. The accuracy of recognizing dictated speech is already high in Indonesian ASR. In other hand, when recognizing spontaneous speech, the accuracy drops significantly. Hypothetically, the language model of spontaneous speech is not built well in the current Indonesian automatic speech recognitions. It is due to the lack of number of spontaneous speech data. Current Indonesian ASR’s training data mostly consist of dictated speech. It causes problems when recognizing spontaneous speech because of the differences in the characteristic between dictated and spontaneous speech [3].

Spontaneous speech has its own characteristics, some of them are the filled pause, repetition, and revision [4]. In this research, only filled pause will be the focus of the experiments. There are 8 kinds of filled pause that are handled in this research. Those filled pauses are “eu”, “euh”, “eum”, “gitu”, “gitu
yah”, “kan”, “tuh”, and “yah”. We do not handle other kinds of filled pause, such as ‘napas’ (‘breath’), ‘decak’ (‘click’), or ‘tawa’ (‘laughter’). We also do not handle repetition and revision.

The acoustic modeling technique used in this research is the Gaussian Mixture Model (GMM)-Hidden Markov Model (HMM). We prefer GMM-HMM over Deep Neural Network (DNN)-HMM technique due to the number of data. The number of data required for DNN-HMM technique is quite lot. Unfortunately, the number of data available for this research is limited. So, the GMM-HMM is the best choice.

N-gram technique is chosen over deep learning-based technique for language modeling due to the acoustic modeling technique chosen. For GMM-HMM acoustic modeling technique, the state-of-the-art language modeling is n-gram. N-gram is effective on many speech recognition domains. The experiments conducted in this research revolve in language models. There are three experiment scenarios conducted in this research. The acoustic model of these three are all the same, which is triphone model of GMM-HMM. The variables here are the language models used. The first scenario, known as the ‘baseline’ (considered as the standard scenario of the research) consists of the basic text corpus, which are ‘Tala’ text corpus plus the transcript of speech training data. We add spontaneous text corpus in the second scenario. Text corpus for language model used in the third scenario, known as the ‘proposed method’ (which is the core of the research) consist of Tala + transcript of speech training data + spontaneous text corpus used in second scenario + the result of data augmentation of all. The following sections are organized as follows. Section 2 contains brief explanations about n-gram and also n-gram as language modeling technique. The next section explains about statistical machine translation used for data augmentation in this research. The experiments are described in the next section. Later, the final section discusses about result and analysis based on the experiment results

2. N-gram and GMM-HMM Technique

2.1. Description of n-gram

N-gram is a statistical language modeling technique that assigns probability to word sequences given some prior probabilities. N-gram is considered the state-of-the-art language modeling technique, especially in Indonesian automatic speech recognitions [1, 2]. Equation (1) shows the formula of the Maximum Likelihood Estimation of n-gram.

\[
P(w_n \mid w_{n-N+1}^{n-1}) = \frac{C(w_{n-N+1}^{n-1}w_n)}{C(w_{n-N+1}^{n-1})}
\]

In this equation, \(P(w_n \mid w_{n-1})\) is the probability of word \(w_n\) following word \(w_{n-1}\); \(N\) is the value of the gram: 2 for bigram, 3 for trigram, and so on; \(C(w_{n-N+1}^{n-1}w_n)\) is the number of occurrence of word \(w_{n-1}\) followed by word \(w_n\); \(C(w_{n-1})\) is the number of occurrence of word \(C(w_n)\) in the training corpus.

2.2. GMM-HMM technique

Gaussian Mixture Model (GMM)-Hidden Markov Model (HMM) is one of the most well-known acoustic modeling technique used in any automatic speech recognition. HMM is actually probabilistic model that consists of transition probability and observation probability. Transition probability is the value of probability of moving from one state to another state. Observation model is the value of the probability in each state. Fig 1 shows the HMM with 5 states.
Every emitting state (state 2, 3, and 4 in the Figure 1) has its own GMM. It means every word can consist of one or more phoneme. Every phoneme has its own phone HMM just like Fig. 1. Every phone HMM has emitting states and non-emitting states (state 1 and 5 in Figure 1). Every emitting state has its own GMM.

3. Data augmentation
We use statistical machine translation to augment data. The purpose of data augmentation is to increase the number of spontaneous data to enrich the characteristic of spontaneous speech in language model used for training. We use ‘Moses’ statistical machine translation to do the data augmentation. Figure 2 from [6] shows the general architecture of statistical machine translation.

There are two main process in machine translation: training and decoding (testing). Training process needs parallel corpus, which is a pair of source and target text with its respective language. For example, if you make a machine translation of French-English, then the source language is French and the target language is English. But in this research, the source language is dictated Indonesian language (Bahasa) and the target language is spontaneous Bahasa. Before used, every text corpus must be pre-processed first which includes tokenization, truecasing, and cleaning process.

![Figure 1. HMM of a word with 5 states](image1)

![Figure 2. General architecture of statistical machine translation [6]](image2)
Training process uses count-based probability calculations on the dataset to train two statistical models: translation model and language model. Translation model is obtained by training parallel corpus of the source and target datasets. Language model is extracted from training on monolingual corpus, which is the target dataset.

Decoding process is actually the process of searching for possible translations and score them, then maximize the obtained score to get the most probable one given both the translation and language models. Algorithms used for decoding are varied, such as depth-first search (DFS), breadth-first search (BFS), or greedy search. The state-of-the art of decoding algorithm in statistical machine translation is beam search. It can prune some candidates with higher cost and instead following path with lower cost. It can speed up the decoding process.

4. Experiment

4.1. Data acquisition
First, data from prior researches are collected. Training data for this research made use of data from [5]. The testing data for this research comes from [1]. The testing data consists of two kinds of data: dictated and spontaneous. The complete text training corpus summary for this research is on the Table 1. The TR_DIKT is the transcript of the speech data from [5]. Table 2 shows the speech training corpus used in this research.

| Name      | Size (words) | Sentence Style |
|-----------|--------------|----------------|
| TR_DIKT   | 78,107       | Dictated       |
| TEXT_TALA | 10,299,778   | Dictated       |
| TEXT_TAMB_SP | 44,213 | Spontaneous   |

Table 2. Speech training corpus

| Name         | Duration                                      |
|--------------|-----------------------------------------------|
| AUDIO_TR_DIKT| 14 hours 25 minutes (20 speaker, 6,679 utterances) |

The complete text testing corpus summary for this research is on the Table 3. Table 4 shows the speech testing corpus used in this research. Both dictated and spontaneous data are from [1]. The text data is the transcript of the speech data.

| Kind of testing data | Size (words) |
|----------------------|--------------|
| Dictated             | 5,446        |
| Spontaneous          | 8,159        |

Table 3. Text testing corpus

| Name         | Duration                                      |
|--------------|-----------------------------------------------|
| AUDIO_TS_DIKT| 53 minutes 45 seconds (20 speaker, 600 utterances) |
| AUDIO_TS_SPON| 1 hour 14 minutes (20 speaker, 595 utterances)  |

Table 4. Speech testing corpus
4.2. Language model training
In this research, we conduct three experiment scenarios with its own language model each. Table 5 shows the scenario of language model experiment.

Table 5. Language model experiment scenario

| No | Language model          | Text corpus used                                |
|----|-------------------------|-------------------------------------------------|
| 1. | NGRAM_LM_BASE           | TEXT_TALA + TR_DIKT                             |
| 2. | NGRAM_LM_TAMB_SP        | TEXT_TALA + TR_DIKT + TEXT_TAMB_SP              |
| 3. | NGRAM_LM_PROPOSED       | TEXT_TALA + TR_DIKT + TEXT_TAMB_SP + result of data augmentation |

These three language models are used for the scenario of developing automatic speech recognition. Table 6 shows the scenario of ASR development.

Table 6. Automatic Speech Recognition (ASR) scenario

| No | Acoustic model | Language model          | Lexicon (words) |
|----|----------------|-------------------------|-----------------|
| 1. | Triphone GMM-  | NGRAM_LM_BASE           | 63,976          |
|    | HMM            |                         |                 |
| 2. |                | NGRAM_LM_TAMB_SP        | 74,531          |
| 3. |                | NGRAM_LM_PROPOSED       | 75,931          |

To build the ASR, we use Kaldi [7]. Kaldi is a well-known automatic speech recognition toolkit. It can accommodate whether acoustic modeling used is n-gram or deep learning-based, such as Long-Short Term Memory (LSTM). In triphone model training phase, we have to input the parameter for the training. There are two of them, ‘num_leaves’ and ‘tot_gauss’. Based on the experiments, the configuration which gives the least WER is ‘1,000’ and ‘tot_gauss’ = 16,000. The result of the first experiment scenario (the baseline) is given in the following Table 7.

Table 7. Testing result of first scenario

| Testing Data | Perplexity | WER (%) |
|--------------|------------|---------|
| Dictated     | 127,412    | 9.93    |
| Spontaneous  | 759,883    | 42.25   |

The result of the second experiment scenario is given in the following Table 8.

Table 8. Testing result of second scenario

| Testing Data | Perplexity | WER (%) |
|--------------|------------|---------|
| Dictated     | 135,901    | 8.80    |
| Spontaneous  | 840,462    | 38.66   |

The result of the third experiment scenario is given in the following Table 9.
Table 9. Testing result of third scenario

| Testing Data   | Perplexity | WER (%) |
|----------------|------------|---------|
| Dictated       | 128,281    | 7.34    |
| Spontaneous    | 852,025    | 39.51   |

The result of comparing the WER score of these three scenario of dictated testing data is given in Figure 3.

![Figure 3. WER of three experiment scenarios of dictated testing data](image)

As seen in Figure 3, the WER score of every scenario keeps decreasing in every step. It means the addition of spontaneous data also improves the recognition of dictated speech. The result of comparing the WER score of these three scenario of spontaneous testing data is given in Figure 4.

![Figure 4. WER of three experiment scenarios of spontaneous testing data](image)

The following Table 10 shows two sentences when recognized with the first, second, and third scenario ASR. ‘Ref’ is the real sentence, while ‘Hyp 1’ is hypothetical text recognized by first scenario, ‘Hyp 2’ is hypothetical text recognized by second scenario, and ‘Hyp 3’ is hypothetical text recognized by third scenario. The ‘***’ in ‘Hyp’ sign means that that word is not recognized by that system (there is a deletion). If it happens in ‘Ref’ it means that there is insertion. That is why the value of WER can be more than 100%, because of the formula is adding all substitution, deletion, and insertion and then dividing all of them with the number of word in test data.
Table 10. Decoding result from three scenarios

| No. | Decoding Result |
|-----|-----------------|
| 1.  | Ref jadi <euh> kuliahnya yang penting asal datang |
|     | Hyp 1 jadi telah kuliahnya yang penting ada tamu |
|     | Hyp 2 jadi <eu> kuliahnya yang akan datang |
|     | Hyp 3 jadi <eu> kuliahnya yang penting astaga |
| 2.  | Ref <napas> nah di situ kita <napas> menggelar <napas> <euh> kain ceritanya nih jadi dukun tapi <euh> enggak enggak sampai yang harus ruis ruis gitu ya <napas> misalnya kepala dipakein apa gitu enggak <napas> |
|     | Hyp 1 *** *** dari situ kita *** menggelar *** ar kain ceritanya *** jadi dukun *** *** amda asal ayam *** harus berselisih *** *** *** telah bisanya kepala dipakai apalagi *** termak *** |
|     | Hyp 2 *** *** dari situ kita *** menggelar *** h kain berceritanya ini jadi dukun tapi *** *** karena sampeyan mau harus ruis ruis itu ya bisa saja kepala dipakein kapal itu *** *** |
|     | Hyp 3 *** *** dari situ kita *** menggelar *** h kain berceritanya ini jadi dukun tapi kita ga ga sampai yang harus bis bis itu ya bisa saja kepala dipakein kapal itu *** *** |

When comparing the old and new corpus (the one before and after data augmentation), it is seen that there is an addition of 36,929 new words in the new corpus. The total number of the addition of eight filled pauses/fillers “eu”, “euh”, “eum”, “gitu”, “gitu yah”, “kan”, “tuh”, and “yah” is 35,109 words. It means 95% of the addition revolves around these eight words. From these 35,109 new words, the filler “eum” is the most dominant with 9,702 addition or 27.63% of the total. After “eum”, “gitu” comes second with 23.60% and “yah” with 13.41%. The filler “eu” is 12.63% and “euh” is 10.93%. The bottom three are “tuh” with 4.97%, “gitu yah” with 4.01% and the last one is “kan” with 2.82%.

5. Conclusion
In this research, we use n-gram to build an Indonesian language model for Indonesian spontaneous automatic speech recognition. Language model is the subject of the experiment in this research. There are three experiment scenarios in this research. The first one is baseline, the second one is with the addition of spontaneous text corpus which was from [8]. The third one is the proposed method of this research which conducted data augmentation of the language model of the second scenario. There is an increase of accuracy of recognizing spontaneous testing data of 2.74% relative to the baseline for the proposed method. When tested to dictated testing data, there is an increase of 2.59%. Actually the accuracy in recognizing spontaneous data is higher in second scenario compared to the third one. The second scenario gives an increase of 3.59% relative to the baseline in recognizing spontaneous testing data.

If we look back to the original motivation of doing data augmentation, which is to minimize the effort required in collecting spontaneous data manually, the decrease is still considered reasonable. The fact that there is an increase of accuracy by doing data augmentation gives hope in improving spontaneous Indonesian automatic speech recognitions’ performance. For future works, the researchers need to do pre-processing to the spontaneous data used for data augmentation, do more development on lexicon,
and also do post-processing of the corpus after augmented. Last but not least, the size of parallel corpus needs to be increased in order to get a better translation model.

References

[1] Hoesen D, Satriawan C H, Lestari D P and Khodra M L 2016 Towards Robust Indonesian Speech Recognition with Spontaneous-Speech Adapted Acoustic Models Procedia Computer Science vol 81 pp 167-73

[2] Yuwan R and Lestari D P 2018 Deep Neural Network Based Acoustic Modeling for Indonesia Spontaneous Speech Recognition with Active Learning unpublished

[3] Nakamura M, Iwano K and Furui S 2008 Differences between acoustic characteristics of spontaneous and read speech and their effects on speech recognition performance Computer Speech & Language 22(2) pp 171-84

[4] Jurafsky D and Martin H 2008 Natural and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition (New Jersey: Prentice-Hall)

[5] Lestari D P and Furui S 2010 Adaptation to Pronunciation Variations in Indonesian Spoken Query-Based Information Retrieval IEICE Transactions on Information and System E93.D(9) pp 2388-96

[6] Maazouzi Z E, Mohajir B E E and Achhab M A 2017 A systematic reading in statistical translation: From the statistical machine translation to the neural translation models J. of Information and Communication Technology 16(2) pp 408-41

[7] Povey D et al 2011 The Kaldi Speech Recognition Toolkit IEEE 2011 Workshop on Automatic Speech Recognition and Understanding (Hawaii: IEEE Signal Processing Society) pp 1-4

[8] Zakiah I and Lestari D P 2019 Transfer Learning from News Domain to Lecture Domain on Automatic Speech Recognition unpublished

Acknowledgments

Thanks are due to PT. Prosa Solusi Cerdas and other parties who contributed in this research or during the process of working for this paper.