Developing an Adaptive Language Model for Bahasa Indonesia

Satria Nur Hidayatullah[1], Suyanto[2]
School of Computing, Telkom University
Jl. Telekomunikasi No. 01, Terusan Buah Batu
Bandung, West Java, Indonesia 40257

***Abstract*—A language model is one of the important components in a speech recognition system. It is commonly developed using a statistical method called *n*-gram. However, a standard *n*-gram cannot be used for general domains with so many ambiguous semantics of sentences. This paper focuses on developing an adaptive *n*-gram language model for Bahasa Indonesia. First, a text corpus of ten million distinct sentences is crawled from hundreds of websites of news, magazines, personal blogs, and writing forums. The text corpus is then used to construct an adaptive language model using Latent Dirichlet Allocation (LDA) with Collapsed Gibbs Sampling (CGS) training method. Compare to the standard *n*-gram, the adaptive language model gives a better performance in the word selection to produce the best sentence.**

Keywords*—Adaptive Language Model; Bahasa Indonesia; Collapsed Gibbs Sampling; Latent Dirichlet Allocation; text corpus

I. Introduction

A language model is a basic, fundamental task in the field of natural language processing [1]. It plays an important role in many applications, e.g. automatic speech recognition (ASR) [2], [3], spoken dialog systems [4], and statistical machine translation [5] since it provides some apriori probabilities of word sequences.

There are two approaches to develop a language model, i.e. count-based *n*-gram models [6], [7], [8], [9] and neural language models [10], [11]. In [12], the researchers state that the *n*-gram models are faster as well as more flexible and scalable, but the neural language models are usually better in accuracy.

The recent modern language model is developed using neural methods, especially recurrent neural networks (RNN), that gives a high accuracy but a high complexity of computation [13]. To address such problems, the researchers propose many methods of optimization as well as regularization as described in [3], [14], and [15]. It can be said that the neural-based language model is not mature. Hence, the other researchers use an adaptive count-based approach in practices as described in [16] or a combination of both approaches as proposed in [12] that has two advantages, where it learns faster and gives high accuracy.

The adaptive count-based approach can be implemented using some different methods, such as a minimum discriminant estimation [1], a maximum entropy principle [17], a dynamic marginal [9], a semantic clustering [8], etc. All methods are simply implemented using a statistical computation. This paper focuses on developing an adaptive language model for Bahasa Indonesia using a count-based approach. It is implemented using an LDA that is trained by a CGS method.

A language model is generally developed using a text corpus of millions or even billions of sentences. However, the topic domain of a word or a term affects the meaning of a sentence containing the word or term. In Bahasa Indonesia, the same speech intonation may give some different words depends on the topic domain of the word or term. For instance, a fluent Indonesian utterance $\langle$*kemeja*$\rangle$ with the same speech intonation can be written as "*ke meja*" (go to the table) or "*kemeja*" (a dress). Therefore, an adaptation of the language model probabilities to the current topic domain is commonly used to improve the language model [1]. Using an adaptive *n*-gram language model, a sentence "*Terdakwa diseret ke meja hijau*" (the defendant is brought to the trial) should have a higher probability than another similar sentence "*Terdakwa diseret kemeja hijau*" (the defendant is brought green dress) since "*meja hijau*" is an Indonesian idiom that means "trial" while "*kemeja hijau*" (green dress) is not strongly related to the topic domain.

Crawling sources of online text data is a simple way to develop a very large text corpus. In this research, a large Indonesia text corpus of 10 M sentences is obtained from hundreds of websites of news, magazines, personal blogs, and writing forums. The text corpus is used to create an adaptive language model using an LDA, a generative probabilistic model described in [18]. This adaptive language model is expected to give higher performance than the standard *n*-gram language model.

II. Language Model Development

A set of documents is collected by crawling some Indonesian websites. The raw collected set of documents is then preprocessed to become a text corpus. This process keeps going until the target of total unique sentences of a minimum 10 million is reached. The constructed text corpus is then used to construct both adaptive and non-adaptive language models using the LDA and the standard *n*-gram model respectively.

A. Text Corpus Development

Constructing a text corpus has two main steps, i.e. collecting and preprocessing. The step by step of constructing corpus in this paper is described as follows:

1) Collecting millions of sentences by crawling many websites of news, magazines, personal blogs, and writing forums;
2) Cleansing the mistyped text using Regex, where the characteristics of mistyped words are: three or more vowels in a row, more than one punctuation, the word frequency is less than 10;
3) Cleansing the foreign words;
4) Removing the stop-words using a dictionary of Indonesian stop-words described in [19]; and
5) Counting the number of sentences, where a sentence contains maximum ten words and ended by a period ".", a question mark "?", or an exclamation point "!".

Those steps are repeated until the text corpus contains at least 10 million unique sentences.

B. Latent Dirichlet Allocation

In LDA, each document is assumed to contain various topics and the words occurred in the document are supposed to be generated from the topics [18]. The pseudo-code adapted from [18] can be described as:

```
1) Assign $\theta$ using dirichlet distribution ($\alpha$);
2) Assign $\omega$ using dirichlet distribution ($\beta$);
3) For each word in the document do:
    a) Choose a topic $\mathbf{z}$ using a multinomial distribution ($\theta$);
    b) Choose a word distribution using the topic $\mathbf{z}$;
    c) Choose a word $\mathbf{w}_n$ using a multinomial distribution ($\omega_z$);
    d) Loop step a to c for every word in the document.
```

C. Collapsed Gibbs Sampling

The CGS is a method to train an LDA model to construct an adaptive model language. The purpose of this method is to improve a multinomial distribution on θ parameter for every document and ω for every topic [18]. The initial step of this method is counting distribution using LDA, either the topic distribution in every document or the word distribution in every topic. Each iteration of the training is described as follows:

```
1) Iteration for every document $D$
    a) zm = topic distribution for every word in document $D$
    b) nm = total topic distribution in document $D$
2) Iteration for every word $k$ in document $D$
    a) Do decrement for every old topic that has been assigned to the word $k$ and decrement the total of the document that old topic has
    b) Assign new topic using multinomial sampling
    c) For every new topic do increment for the word $k$ and for the total of the document that new topic has [18]
```

III. Research Methodology

In this research, a motherset of 11,011,771 sentences from 339,128 documents is collected by crawling some websites. The motherset represented as a set of documents is then used to develop an adaptive language model using the LDA. Meanwhile, the motherset that is represented as a set of sentences is used to construct a non-adaptive language model using an n-gram.

The developed language model is measured using a perplexity score since it will be used in an ASR system. As described in [20], [21], perplexity is a commonly used metric to measure the performance of a word-based language model applied in an ASR model. This metric has two advantages. Firstly, it is calculated independently with no real ASR. It is categorized as an intrinsic evaluation that is much simpler than an extrinsic one by evaluating the language model on the real ASR model [20]. Secondly, it has a high correlation with word error rate (WER) in an ASR, especially when the models are trained using the same training set of data. But, the perplexity score has a disadvantage where it cannot take into account an important issue in ASR related to the difficulty of acoustic. However, this disadvantage does not significantly reduce the correlation of perplexity with the ASR.

The smaller score of perplexity the better adaptive language model generated by the LDA. The perplexity score of a test set $\boldsymbol{w}$ calculated using

$$\text{perplexity}(\boldsymbol{w}) = \exp\left\{-\frac{\mathcal{L}(\boldsymbol{w})}{T}\right\}, \tag{1}$$

where T is the number of tokens and $\mathcal{L}$ represents a likelihood that is computed using

$$\mathcal{L}(\boldsymbol{w}) = \log p(\boldsymbol{w}|\boldsymbol{\Phi}, \alpha) = \sum_{d} \log p(\boldsymbol{w}_d|\boldsymbol{\Phi}, \alpha). \tag{2}$$

In this research, an adaptive language model is developed gradually using the LDA. First, it is generated using the LDA for only 5 clusters with some iterations to verify its quality. Next, it is then developed using some bigger clusters to get a more realistic language model. Finally, the produced language model is compared to the non-adaptive standard n-gram language model in term of the ratio of selecting the correct sentence to the incorrect one.

IV. Result and Discussion

An adaptive language model is first developed using the LDA. The result is then compared to the non-adaptive standard n-gram language model based on a ratio of selecting the correct sentence to the incorrect one.

A. Adaptive Language Model

The construction of the adaptive language model using LDA needs some experiments to see how the parameters work. The experiment of adaptive language model construction using LDA is divided into two experiments. First, Experiment 1 examines the training iteration and the total cluster to see if the perplexity really shows the quality of the word separation using human judgment. Next, Experiment 2 tests the parameters on a bigger cluster to check whether it needs more training iterations or not.

The perplexities produced by the model using 5 topics and three different iterations are illustrated by Table I. It shows that the more iteration the lower perplexity. Each perplexity is then investigated to check if a lower perplexity gives a clearer word distribution by the topic or not.

TABLE I. PERPLEXITY ON EXPERIMENT 1

Model/State	Initial	Final
Topic = 5, Iteration = 10	5,089.37	4,418.41
Topic = 5, Iteration = 15	5,545.97	4,358.31
Topic = 5, Iteration = 30	7,125.30	4,296.86

Table II illustrates the word distribution of the model with 5 topics and 10 iterations. Each cluster consists of five words sorted by their rank and produces a unique topic. Cluster 1 that contains "*laku* (behavior)", "*hadap* (face up)", "*hasil* (result)", "*itu,* (that,)", and "*Indonesia*" produces a topic about "Indonesian people" (the behavior of Indonesian people in facing up that result). In this case, "*hasil* (result)" is a vague word since it is not clear what is result. Cluster 2 with five words of "*partai* (party)", "*ketua* (leader)", "*presiden* (president)", "*Jakarta,*", and "*Jakarta*" forms a topic about "Politic" (the leader of political party). Cluster 3 with five words of "*warga* (citizen)", "*jalan* (road)", "*korban* (victim)", "*kabupaten* (district)", and "*rumah* (home)" comes to a topic about "Disaster" (the citizen of district those to be the victim). Cluster 4 with five words "*lihat* (watch)", "*musim* (season)", "*tampil* (compete)", "*ini.* (this.)", and "*liga* (league)" forms a topic about "Sports" (to watch a competition in this season of league). Cluster 5 with five words of "*laku* (behavior)", "*milik* (belongs to)", "*Indonesia*", "*Rp* (Rupiah, the Indonesian currency)", and "*kerja* (work)" forms a topic about "Economy" (the behavior of Indonesian currency).

In Table II, where the model is developed using 5 cluster topics with 10 iterations, the word distribution shows the vague cluster when it is seen from the topic. Meanwhile, in Table III, where the model is developed using 5 cluster topics with 15 iterations, the word "*hasil* (result)" that is one of the vague words on the previous model starting to be clustered clearly. Cluster 3 that contains "*latih* (train)", "*menang* (win)", "*tanding* (compete)", "*hasil* (result)", and "*laga* (fight)" give a clearer assumption that the cluster belongs to the topic of "Sport". Finally, in the last model using 5 topics and 30 iterations illustrated by Table IV, the created word order that is previously in a clear cluster still remains in order. It means that a smaller perplexity brings a clearer word distribution by the topic.

Next, Experiment 2 uses the total cluster of both 15 and 20 as well as the training iteration of 30. It produces some results illustrated by Fig. 1 those show the changes of perplexity scores. It shows that the perplexity score goes up in the early training, but it keeps decreasing in the end until less than the initial perplexity. It means that a bigger cluster needs more training iterations.

TABLE II. WORD DISTRIBUTION USING 5 TOPICS AND 10 ITERATIONS

Cluster/Rank	1	2	3	4	5
1	*laku*	*hadap*	*hasil*	*itu,*	*Indonesia*
2	*partai*	*ketua*	*presiden*	*Jakarta,*	*Jakarta*
3	*warga*	*jalan*	*korban*	*kabupaten*	*rumah*
4	*lihat*	*musim*	*tampil*	*ini.*	*liga*
5	*laku*	*milik*	*Indonesia*	*rp*	*kerja*

TABLE III. WORD DISTRIBUTION USING 5 TOPICS AND 15 ITERATIONS

Cluster/Rank	1	2	3	4	5
1	*laku*	*Jakarta,*	*presiden*	*Jakarta*	*partai*
2	*laku*	*milik*	*Indonesia*	*rp*	*itu,*
3	*latih*	*menang*	*tanding*	*hasil*	*laga*
4	*warga*	*jalan*	*korban*	*polisi*	*laku*
5	*lihat*	*anak*	*itu.*	*jalan*	*buah*

TABLE IV. WORD DISTRIBUTION USING 5 TOPICS AND 30 ITERATIONS

Cluster/Rank	1	2	3	4	5
1	*laku*	*presiden*	*Jakarta,*	*partai*	*ketua*
2	*anak*	*lihat*	*buah*	*rumah*	*itu.*
3	*latih*	*menang*	*tanding*	*hasil*	*laga*
4	*warga*	*korban*	*polisi*	*rumah*	*jalan*
5	*laku*	*Indonesia*	*rp*	*milik*	*kerja*

To construct the final model of LDA, the total cluster of 90 and the training iteration of 300 are used. The parameters come from the combination of results from Experiment 1 and Experiment 2. The final perplexity score of the model is 22,108.03 as illustrated by Fig. 2. The perplexity score in the initial state is less than that in the final state. But, the initial state has lower credibility since it is randomly constructed.

B. Comparison of the Adaptive and Non-adaptive Language Models

The non-adaptive language model is created using a normal *n*-gram with a back-off smoothing method. The ratio of probability for the correct and incorrect sentences generated by the adaptive and non-adaptive models are then compared using ten pairs of the correct and incorrect sentences listed in Table V.

The results in Table VI show that 8 of 10 sentences produced by the adaptive language model have a bigger ratio to separate the correct sentences from the incorrect ones. Evaluating the 10th sentence shows that both language models make incorrect decisions with ratios less than 1. The non-adaptive language model is better than the adaptive model only on the 9th sentence. These facts show that the adaptive language model is more capable of building the best sentence since it carefully selects a word with a fit ratio for either correct or incorrect sentence.

V. CONCLUSION

An adaptive language model for Bahasa Indonesia has been successfully developed using an LDA. The LDA is capable of constructing a good cluster of an adaptive language model by constantly fixing the cluster of words shown by some top clusters on each topic. Compare to the standard *n*-gram, the developed adaptive language model gives a more accurate computation of the probability of word selection shown by some higher ratios of choosing correct sentences. In the future, more clusters can be generated from a bigger text corpus in order to produce a much bigger adaptive language model used in a real-world application of speech technology.

ACKNOWLEDGMENT

We would like to thank Telkom University and colleagues for providing a high-speed internet connection to develop a large mother sentence set used in this research.

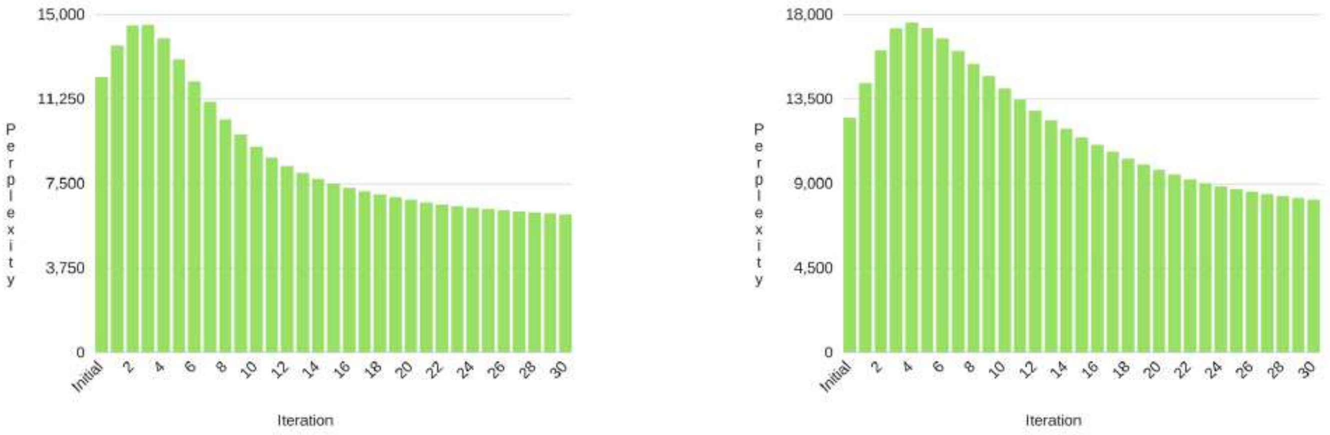

Fig. 1. Perplexity curves of 15 clusters (left) and 20 clusters (right)

TABLE V. TEN PAIRS OF THE CORRECT AND INCORRECT SENTENCES

No	Correct Sentence	Incorrect Sentence
1	*Terdakwa diseret ke meja hijau* (the defendant is dragged to the trial)	Terdakwa diseret kemeja hijau (the defendant is dragged green dress)
2	*Mandi sambil keramas* (Bathe while shampooing)	*Mandi sambil kera mas* (Bathe while monkey brother)
3	*Ketua partai memimpin sidang* (The leader of party leads the session)	*Ke tua partai memimpin sidang* (To old party leads the session)
4	*Kejahatan yang kejam dan sadis* (A crime that is cruel and sadistic)	*Kejahatan yang ke jam dan sadis* (A crime that is to clock and sadistic)
5	*Pemimpin keras kepala* (A stubborn leader)	*Pemimpin ke ras kepala* (A leader to head race)
6	*Nasi kebuli sangat nikmat* (The Kebuli rice is so delicious)	*Nasi ke buli sangat nikmat* (The rice to bladder is so delicious)
7	*Tidur pakai selimut* (Sleep using a blanket)	*Tidur pakai sel imut* (Sleep using a cute cell)
8	*Besok ujian tentang peribahasa* (Tomorrow the test of proverbs)	*Besok ujian tentang peri bahasa* (Tomorrow the test of a language fairy)
9	*Tamasya ke kebun bunga* (A trip to the flower garden)	*Tamasya ke ke bun bunga* (A trip to to the flower bun)
10	*Simpanan dana Bu RT* (Deposits of funds from Mrs. RT)	*Simpanan dan abu RT* (Deposits and RT ashes)

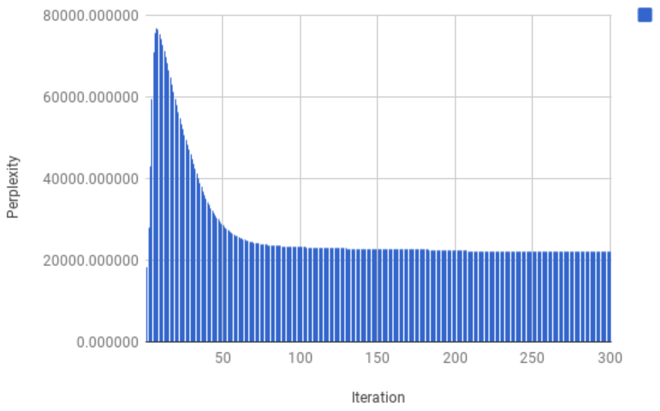

Fig. 2. Perplexity curves of 90 clusters and 300 iterations

TABLE VI. COMPARISON OF PROBABILITY RATIO OF ADAPTIVE AND NON-ADAPTIVE LANGUAGE MODELS

No	Non-adaptive model	Adaptive model
1	1.670465	1.167508
2	1.405522	1.435420
3	1.649173	1.709780
4	1.130223	1.138503
5	1.568608	1.635319
6	1.090512	1.114604
7	1.393444	1.420549
8	1.269316	1.291451
9	1.519365	1.338187
10	0.929215	0.915490

REFERENCES

[1] S. Deila Pietra, V. Deila Pietra, R. L. Mercer, and S. Roukos, “Adaptive language modeling using minimum discriminant estimation,” *EEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, vol. 1, no. Mdi, pp. 633–636, 1992.

[2] T. Matsuoka, R. Hasson, M. Barlow, and S. Furui, “Language model acquisition from a text corpus for speech understanding,” in *IEEE International Conference on Acoustics, Speech, and Signal Processing Conference (ICASSP)*, vol. 1, may 1996, pp. 413–15A vol. 1.

[3] M. Ma, M. Nirschl, M. Ma, M. Nirschl, F. Biadsy, and S. Kumar, “Approaches for Neural-Network Language Model Adaptation,” in *INTERSPEECH*, 2017.

[4] R. A. Solsona, E. Fosler-Lussier, H. J. Kuo, A. Potamianos, and I. Zitouni, “Adaptive language models for spoken dialogue systems,” in *2002 IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 1, may 2002, pp. I–37–I–40.

[5] P. Baltescu and P. Blunsom, “Pragmatic Neural Language Modelling in Machine Translation,” in *The 2015 Annual Conference of the North American Chapter of the ACL*, 2015, pp. 820–829.

[6] R. Rosenfeld, “A Hybrid Approach to Adaptive Statistical Language Modeling,” in *Proceedings of the Workshop on Human Language Technology*, ser. HLT ’94. Stroudsburg, PA, USA: Association for Computational Linguistics, 1994, pp. 76–81. [Online]. Available: https://doi.org/10.3115/1075812.1075827

[7] C.-H. Lee and J.-L. Gauvain, “Adaptive Learning in Acoustic and Language Modeling,” in *Speech Recognition and Coding*, A. J. R. Ayuso and J. M. L. Soler, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 1995, pp. 14–31.

[8] R. Kneser and J. Peters, “Semantic clustering for adaptive language modeling,” in *1997 IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 2, apr 1997, pp. 779–782 vol.2.

[9] R. Kneser, J. Peters, and D. Klakow, “Language Model Adaptation Using Dynamic Marginals,” in *EUROSPEECH*, 1997.

[10] S. Merity, B. McCann, and R. Socher, “Revisiting Activation

Regularization for Language RNNs," *CoRR*, vol. abs/1708.0, 2017. [Online]. Available: http://arxiv.org/abs/1708.01009

[11] S. Merity, N. S. Keskar, and R. Socher, "Regularizing and Optimizing LSTM Language Models," *CoRR*, vol. abs/1708.0, 2017. [Online]. Available: http://arxiv.org/abs/1708.02182

[12] G. Neubig and C. Dyer, "Generalizing and Hybridizing Count-based and Neural Language Models," in *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 2016, pp. 1163–1172. [Online]. Available: http://aclweb.org/anthology/D16-1124

[13] A. Baevski and M. Auli, "Adaptive Input Representations for Neural Language Modeling," *CoRR*, pp. 1–12, 2018.

[14] E. Grave and A. Joulin, "Unbounded cache model for online language modeling with open vocabulary," in *The 31st Conference on Neural Information Processing Systems (NIPS)*, 2017.

[15] S. Merity, N. Shirish, and K. Richard, "An Analysis of Neural Language Modeling at Multiple Scales," *CoRR*, 2018.

[16] J. Li, P. Zhang, D. Song, and Y. Hou, "An adaptive contextual quantum language model," *Physica A: Statistical Mechanics and its Applications*, vol. 456, pp. 51–67, 2016. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0378437116002594

[17] R. Lau, R. Rosenfeld, and S. Roukos, "Adaptive Language Modeling Using the Maximum Entropy Principle," in *Proceedings of the Workshop on Human Language Technology*, ser. HLT '93. Stroudsburg, PA, USA: Association for Computational Linguistics, 1993, pp. 108–113. [Online]. Available: https://doi.org/10.3115/1075671.1075695

[18] D. M. Blei, A. Y. Ng, and M. I. Jordan, "Latent Dirichlet Allocation," *Journal of Machine Learning Research*, vol. 3, pp. 993–1022, 2003.

[19] F. Z. Tala, "A Study of Stemming Effects on Information Retrieval in Bahasa Indonesia," Ph.D. dissertation, 2003.

[20] P. Wang, R. Sun, H. Zhao, and K. Yu, "A New Word Language Model Evaluation Metric for Character Based Languages," in *Chinese Computational Linguistics and Natural Language Processing Based on Naturally Annotated Big Data*, M. Sun, M. Zhang, D. Lin, and H. Wang, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013, pp. 315–324.

[21] S. Chen, D. Beeferman, and R. Rosenfeld, "Evaluation metrics for language models," 1998.