Journal of Theoretical and Applied Information Technology
31st October 2014. Vol. 68 No.3

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

COMPARATIVE STUDY OF SMOOTHING TECHNIQUES ON INDONESIAN AND ENGLISH LANGUAGE MODELS

ISMAIL

Computer Engineering Department
School of Applied Science, Telkom University
Bandung, Indonesia
E-mail: ismailrusli@telkomuniversity.ac.id

ABSTRACT

Indonesian language is one of Austronesia languages. It differs from English language, which is one of isolating languages. For Indonesian language, there has been no study of smoothing effect in its language model. Although from mathematical point of view, language model has no direct dependency to specific language, Whittaker [1] showed that, for Russian and English, there are differences in smoothing effect for those languages. In this paper, we studied various smoothing techniques in language model for Indonesian language and compared it to that of English language. Our experiments showed that smoothing effects for statistical Indonesian language model have better perplexity reduction than that of English language. We showed our results in terms of cross-entropy differences among various techniques relative to Katz smoothing.

Keywords: *Smoothing Techniques, Language Model, English and Indonesian Languages, SRILM*

1. INTRODUCTION

Statistical language model estimates probability distribution of sentences in language from a corpus. From mathematical point of view, smoothing techniques used to smooth this probability distribution do not depend on the language of the corpus. However, structure of a language might affect performance of these smoothing techniques. (See [1] for the case of English and Russian). In this paper, we examined statistical language model for Indonesian and English language.

Indonesian language is one of the Austronesia languages [2]. One characteristic of Austronesia languages is that the language is agglutinative. It differs from that of English language that is an isolating language.

For Indonesian language, there has been no study in the effect of various smoothing techniques on its language model. Therefore, our contribution here is to report study of the effects of various smoothing techniques for Indonesian language model and its comparison to that of English language.

We used SRILM [3] as our tool to build language models and experimented with five smoothing techniques, i.e. Absolute Discounting, Katz (Good-Turing), Kneser-Ney, Modified Kneser-Ney, and Witten-Bell. As SRILM provides back off and interpolated version of these techniques (except for Katz smoothing) we provide results for both versions. We used difference in cross-entropy as performance measure of smoothing techniques.

The paper continues as follows. In section 2, we mention several relevant works in smoothing techniques. In Section 3, we detail mathematical description of statistical language model including various smoothing techniques used in this paper. Our experiments and results will be presented in Section 4 which then followed by conclusion in Section 5.

2. PREVIOUS WORKS

Chen and Goodman [4] had reported extensive study on various smoothing techniques for English language. They also proposed a novel technique called Modified Kneser-Ney that is slightly better from the original algorithm.

Another important work is by Goodman [5]. Here, Goodman reported various techniques in language modeling, including higher order *n*-gram, caching, clustering, skipping, and sentence-mixture

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

models. Goodman combined those techniques to get best result for language model. His experiment showed the superiority of Interpolated Kneser-Ney. He also showed that there is no improvement to be gained past 7-gram language model.

Both [4] and [5] work in English language. For different language, Whittaker [1] showed that characteristic difference in English and Russian had resulted in performance different of smoothing techniques for those languages. He built class model for Russian corpus and showed that it provided better relative improvements in perplexity than that of English language.

Teh [6] proposed hierarchical Bayesian model to build language model and reported that its performance is comparable to that of Interpolated Modified Kneser-Ney.

Recent report on the superiority of Modified Kneser-Ney is made by Chelba [7], who also experimented with neural network technique in language modeling.

Advanced techniques in language models studied by Mikolov [8]. The authors used several techniques such as class based model, cache model, maximum entropy model, structured language model, random forest language model, and several types of neural network based language models. The paper showed that combination of models resulted in perplexity reduction of 40% against Modified Kneser-Ney 5-gram language model.

In this paper, we only studied 5 algorithms, i.e., Katz, Kneser-Ney, Modified Kneser-Ney, Absolute Discounting, and Witten-Bell and did not combine them as experiments conducted by Goodman [5]. We also experimented, despite of the results by Goodman [5], with 9-gram language models for both English and Indonesian languages. Experiment with advance techniques is planned for our future research.

3. LANGUAGE MODEL

A language model gives an answer to question "what is the probability of sentence W appears in a natural language?". Sentence like "I like fruits" is naturally more probable to appear in a text or everyday conversation than grammatically similar sentence "I like table".

Formally, a language model approximates probability of a sequence of words $W = w_1w_2w_3 \ldots w_i = w_1^i$.

$$P(W) = P(w_1 w_2 w_3 \ldots w_i) = P(w_1^i) \quad (1)$$

Equation (1) is equal to find probability of the last word of the sentence given the previous words in the sentence times the probability of previous words in the sentence.

$$P(W) = P(w_1^{i-1}) \times P(w_i|w_1^{i-1}) \quad (2)$$

By chain rule we have

$$P(W) = \prod_{j=1}^{i} P(w_j|w_1^{i-1}) \quad (3)$$

Using Markov property, we can assume that probability of a word depends only on previous n-1 words. This is n-gram language model and we get

$$P_n(W) = \prod_{j=1}^{i} P(w_j|w_{j-n+1}^{j-1}) \quad (4)$$

when it is understood that in w_k^l, if $l < k$, the word is discarded.

For convenience, we will use $n = 3$ in our mathematical descriptions.

To calculate the conditional probability in (4), we use maximum likelihood estimation. That is,

$$\begin{aligned} P_3(w_i|w_{i-2}^{i-1}) &= \frac{c_3(w_{i-2}^i)}{c_3(w_{i-2}^{i-1})} \\ &= \frac{c_3(w_{i-2}^i)}{\sum_{w_i} c_3(w_{i-2}^i)} \end{aligned} \quad (5)$$

with c_3 is function that counts sentences in its argument in training set.

The most common metric for evaluation a language model is perplexity. Perplexity is defined by 2^H, where H is cross-entropy of the test set.

$$H(T) = -\frac{1}{W_T}\log_2 P(T) \quad (6)$$

where W_T is number of words in test set T.

A model is relatively better when it has lower perplexity compared to that of other models. Instead of perplexity, we will use difference in cross-entropy relative to Katz smoothing to show the result of our experiments.

Journal of Theoretical and Applied Information Technology
31st October 2014. Vol. 68 No.3

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

3.1 Smoothing Techniques

Previous model suffers from one problem. It gives zero probability to sentences that do not appear in the corpus. To overcome this, we discount probability from sentences appear in the corpus and distribute it to sentences that have zero probability. This technique is called smoothing.

Generally, those smoothing techniques fall into two categories, back off and interpolated techniques.

In back off technique, probabilities of sentences that do not appear in corpus are estimated using that of its lower *n*-gram.

$$P_3(w_i|w_{i-2}^{i-1}) = \begin{cases} f_3(w_i|w_{i-2}^{i-1}) & c_3(w_{i-2}^{i}) > \\ b_3(w_{i-2}^{i-1})P_2(w_i|w_{i-1}) & \text{otherwise} \end{cases} \quad (7)$$

where f_3 is modified probability for sentences appear in corpus and b_3 is scaling factor chosen to make probability sums to one.

Instead of using back off probability, interpolated technique combines probability of a sentence with that of its lower order, e.g. combined probability of 3-gram, 2-gram, and 1-gram.

$$P_3(w_i|w_{i-2}^{i-1}) = g_3(w_i|w_{i-2}^{i-1}) + b_3(w_{i-2}^{i-1})P_2(w_i|w_{i-1}) \quad (8)$$

All smoothing techniques differ in the way they calculate function f_n, g_n, and therefore also b_n.

3.1.1 Absolute Discounting

Absolute discounting [2] uses interpolated method. Function g for absolute discounting is defined as follow.

$$g_3(w_i|w_{i-2}^{i-1}) = \frac{\max\{c(w_{i-2}^{i}) - D, 0\}}{\sum_{w_i} c(w_{i-2}^{i})} \quad (9)$$

with D is discounting value.

To make the distribution sum to 1, we take

$$b_3(w_{i-2}^{i-1}) = \frac{D}{\sum_{w_i} c(w_{i-2}^{i})} N_{1+}(w_{i-2}^{i-1}\bullet) \quad (10)$$

with N_{1+} is

$$N_{1+}(w_{i-2}^{i-1}\bullet) = |\{w_i : c(w_{i-2}^{i-1}w_i) > 0\}| \quad (11)$$

N_{1+} in (11) is the number of unique words following $w_{i-2}w_{i-1}$.

The suggested value for D is

$$D = \frac{n_1}{n_1 + 2n_2} \quad (12)$$

where n_1 and n_2 are the total number of *n*-grams with exactly one and two counts, respectively.

3.1.2 Katz Smoothing

In Good-Turing, any *n*-gram occurs r times should be thought as occur r^* times

$$r^* = (r+1)\frac{n_{r+1}}{n_r} \quad (13)$$

where n_r is the number of *n*-gram that occur exactly r times in training data.

Katz smoothing uses Good-Turing to estimate probability of nonzero *n*-grams that occur less or equal k times. That is

$$f_3(w_i|w_{i-2}^{i-1}) = \frac{c_{\text{katz}}(w_{i-2}^{i})}{\sum_{w_i} c_{\text{katz}}(w_{i-2}^{i})} \quad (14)$$

with c_{katz} defined as

$$c_{\text{katz}}(w_{i-2}^{i}) = d_r r \quad \text{for } 0 < r < k \quad (15)$$

and d_r equals

$$d_r = \frac{\frac{r^*}{r} - \frac{(k+1)n_{k+1}}{n_1}}{1 - \frac{(k+1)n_{k+1}}{n_1}} \quad (16)$$

For $r > k$, probability is calculated using the maximum likelihood estimation. That is

$$f_3(w_i|w_{i-2}^{i-1}) = \frac{c_3(w_{i-2}^{i})}{\sum_{w_i} c_3(w_{i-2}^{i})} \quad (17)$$

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

For *n*-grams that do not occur in training data, the function *b* is defined as follow

$$b_3(w_{i-2}^{i-1}) = \frac{1 - \sum_{w_i:c(w_{i-2}^i)>0} f_3(w_i|w_{i-2}^{i-1})}{\sum_{w_i:c(w_{i-2}^i)=0} P_2(w_i|w_{i-1})} = \frac{1 - \sum_{w_i:c(w_{i-2}^i)>0} f_3(w_i|w_{i-2}^{i-1})}{1 - \sum_{w_i:c(w_{i-2}^i)>0} P_2(w_i|w_{i-1})} \quad (18)$$

For this technique, Katz suggests $k = 5$.

Note that in SRILM, there is no interpolated version of Katz smoothing.

3.1.3 Kneser-Ney and Modified Kneser-Ney

Kneser-Ney is extension of Absolute Discounting, i.e. it uses a discounting value *D* to subtract a portion of probability from probability of nonzero *n*-gram. Therefore, the *f* function equals to that of Absolute Discounting.

What makes Kneser-Ney different from Absolute Discounting is that Kneser-Ney uses modified probability estimate for lower order *n*-grams used for back off. This modified probability is taken to be proportional to the number of unique words that precede it in training data[1].

$$P_2(w_i|w_{i-1}) = \frac{N_{1+}(\bullet w_{i-1}^i)}{N_{1+}(\bullet w_{i-1}\bullet)} \quad (19)$$

where

$$N_{1+}(\bullet w_{i-1}^i) = |\{w_{i-2} : c(w_{i-2}^i) > 0\}| \quad (20)$$

$$N_{1+}(\bullet w_{i-1}\bullet) = |\{(w_{i-2}, w_i) : c(w_{i-2}^i) > 0\}| = \sum_{w_i} N_{1+}(\bullet w_{i-1}) \quad (21)$$

Instead of using back off as from the original paper, Chen [2] using interpolated method of Kneser-Ney and he showed that this interpolated Kneser-Ney is better than the original one.

In addition to interpolated Kneser-Ney, Chen [2] also uses 3 values of constant *D*, i.e., D_1, D_2, D_{3+} which is discounting constant for *n*-grams with count one, two, and three or more respectively in training data. This is called Modified Kneser-Ney.

3.1.4 Witten-Bell

Witten-Bell originally uses interpolated method. The idea behind Witten-Bell is to consider the number of unique words following $w_{i-2}w_{i-1}$ (example for 3-grams). This number is defined

$$N_{1+}(w_{i-2}^{i-1}\bullet) = |\{w_i : c(w_{i-2}^i) > 0\}| \quad (22)$$

With this number, *b* function is defined as

$$b_3(w_{i-2}^{i-1}) = \frac{N_{1+}(w_{i-2}^{i-1}\bullet)}{N_{1+}(w_{i-2}^{i-1}\bullet) + \sum_{w_i} c(w_{i-2}^i)} \quad (23)$$

and higher order distribution is defined as follow.

$$P_2(w_i|w_{i-1}) = \frac{c(w_{i-1}^i)}{N_{1+}(w_{i-1}\bullet) + \sum w_i c(w_{i-1}^i)} \quad (24)$$

4. EXPERIMENTS AND RESULTS

We used English and Indonesia version of Wikipedia as sources for our corpora. Although the English version was a lot larger than that of Indonesian, we made them equal in size.

Wikipedia text is freely available in their website. Before we use it as our corpus, we did several preprocessing in order to make it suitable for building language model. The steps in preprocessing are listed in the followings.

1. First, we removed all unwanted lines such as lines with xml tags and lines with programming languages. We also removed all the lines with garbage characters, e.g., lines contain repetition of dash character.
2. We then removed punctuations.
3. Next, we split paragraphs into sentences.
4. Then, we shuffled and removed duplicate sentences.
5. Finally, we removed sentences contain less than two tokens.

The number of words and sentences in resulting texts are shown in *Table 1*.

Table 1: Number of words and sentences in Indonesian and English Corpora

TEXT	INDONESIAN	ENGLISH
Words	1,513,390	1,986,062
Sentences	23,083,390	33,694,678
Avg. Words/Sent	15.25	16.97

After we preprocessed the text, we did following steps to prepare the data.

[1] Manual of SRILM's ngram-discount

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

We split text into two sets, i.e. training set and test set. We decided to use three different sizes of training set, i.e. set with size of 10K, 100K, and 1M sentences. Total number of words in the training sets is about 17M. For test set, we used four sets each about 10% of the size of training set, i.e. 100K sentences. See *Table 2*.

Table 2: Number of words and sentences in TR (Training Set) and TS (Test Set)

SETS	SENT.	WORDS(ID)	WORDS(EN)
TR1	10K	153,037	170,991
TR2	100K	1,526,113	1,696,746
TR3	1M	15,244,819	16,966,029
TS1	100K	1,528,398	1,695,642
TS2	100K	1,526,624	1,698,129
TS3	100K	1,527,957	1,695,729
TS4	100K	1,524,857	1,697,765

With these training and test sets, we built language models using SRILM. The following are SRILM commands we used to generate language models for different smoothing techniques. Note that example of interpolated method is presented for Absolute discounting only. For other smoothing techniques, the same patterns applied.

1. Absolute discounting back off:
```
ngram-count -text 1M -order 9 -lm 1M.9.ad.bo.lm -cdiscount 0.5
```
2. Absolute discounting interpolated:
```
ngram-count -text 1M -order 9 -lm 1M.9.ad.int.lm -unk -cdiscount 0.5 -interpolate
```
3. Katz smoothing (we used default $k = 7$):
```
ngram-count -text 1M -order 9 -lm 1M.9.gt.lm -unk
```
4. Modified Kneser-Ney back off:
```
ngram-count -text 1M -order 9 -lm 1M.9.kn.bo.lm -unk -kndiscount
```
5. Kneser-Ney back off:
```
ngram-count -text 1M -order 9 -lm 1M.9.ukn.bo.lm -unk -ukndiscount
```
6. Witten-Bell
```
ngram-count -text 1M -order 9 -lm 1M.9.wb.bo.lm -unk -wbdiscount
```
7. Calculating perplexity (example for Modified Kneser-Ney):
```
ngram -ppl test1 -order 9 -unk -lm 1m.9.kn.bo.km
```

With `-unk` parameters, we mapped all unknown words (words that do not appear in training set) to a word `<unk>` and treated it as regular word. Therefore, we have zero Out-of-Vocabulary (OOV) in our language models.

In the following figures, we plot results of our experiments for different *n*-gram language model both for back off and interpolated methods. The results of our experiments confirmed that Modified Kneser-Ney (interpolated) outperforms other smoothing techniques. in term of perplexity both for English and Indonesian languages.

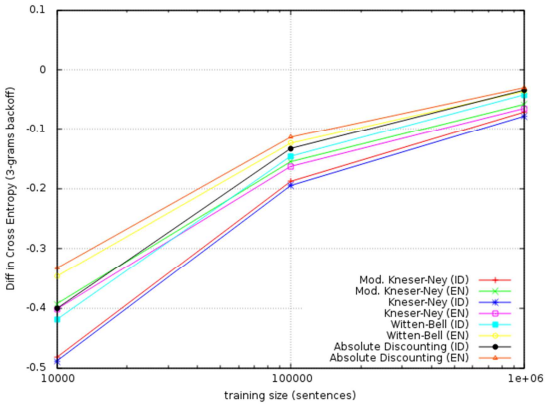


Figure 1: Difference in cross-entropy for 3-grams back off relative to Katz smoothing

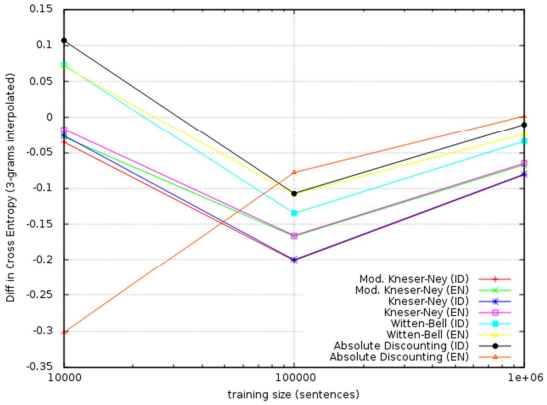


Figure 2: Difference in cross-entropy for 3-grams interpolated relative to Katz smoothing

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

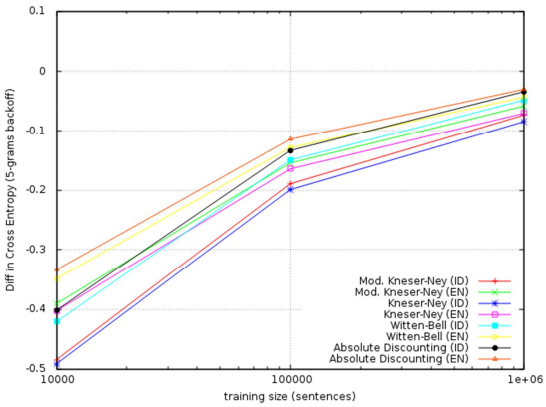


Figure 3: Difference in cross-entropy for 5-grams back off relative to Katz smoothing

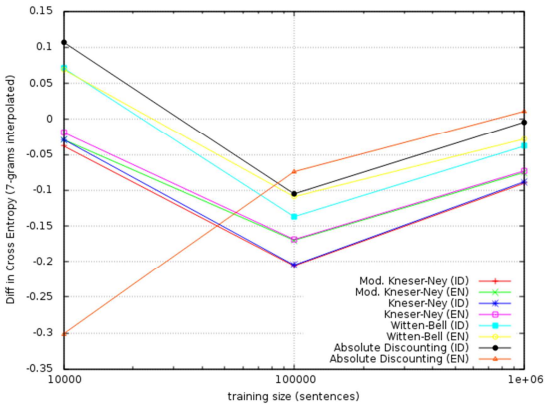


Figure 6: Difference in cross-entropy for 7-grams interpolated relative to Katz smoothing

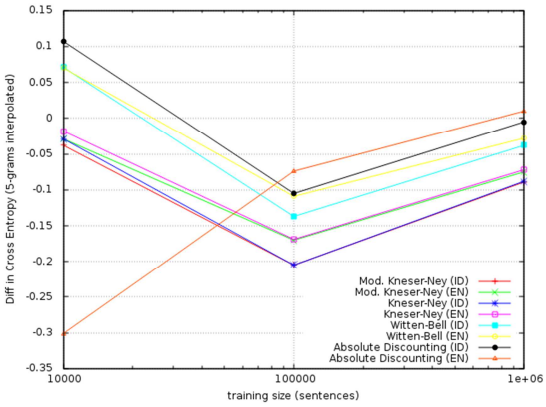


Figure 4: Difference in cross-entropy for 5-grams interpolated relative to Katz smoothing

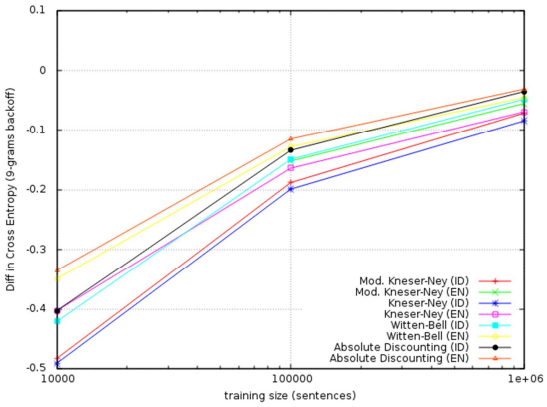


Figure 7: Difference in cross-entropy for 9-grams back off relative to Katz smoothing

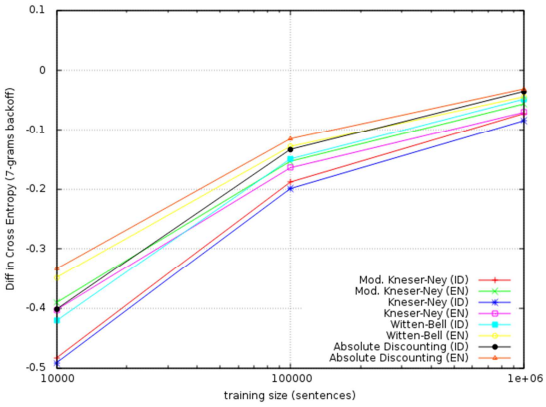


Figure 5: Difference in cross-entropy for 7-grams back off relative to Katz smoothing

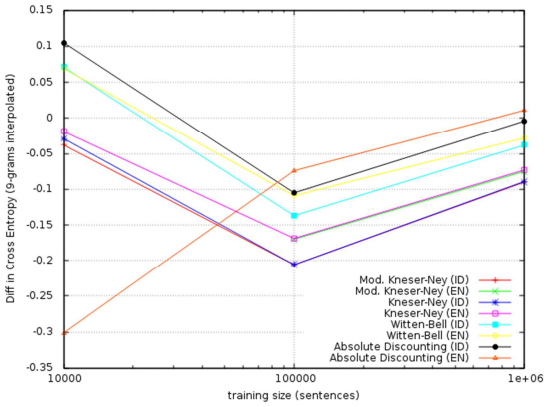


Figure 8: Difference in cross-entropy for 9-grams interpolated relative to Katz smoothing

Our results is also in accordance with Goodman's [5] which stated that there is no significant improvement beyond 7-gram. For example, in our experiments, perplexity value for 7-gram language model with 1M sentences of training

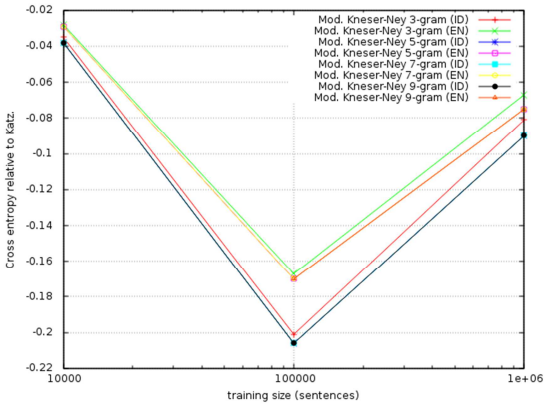
ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

data and Modified Kneser-Ney techniques (interpolated) is 391,39 while for 9-gram is 389,84.

It is important to note that we did not attempt to optimize parameters in language models. We considered parameters optimization is irrelevant for our experiments.

5. CONCLUSION

We saw from graphics in previous section that cross-entropy for different order of *n*-grams does not differ much. Our results showed that smoothing techniques in Indonesian language has greater effect in reducing perplexity (hence increasing cross-entropy difference relative to Katz smoothing) than in English language. *Figures 9* shows comparison of Modified Kneser-Ney (interpolated) for Indonesian and English languages.



Figures 9: Comparison of Modified Kneser-Ney (Interpolated) for Indonesian and English Languages

We need to make point here that we used Wikipedia as our source of training data. In English language, there are standard corpora that has been extensively used to benchmark works on language models or other works in Natural Language Processing. In Indonesia, there is no such thing as standard corpus. Furthermore, there is limited amount of Indonesian corpus available free. During this research, we only found one free corpus, i.e. work by Adriani [9]. This corpus consists of 500K words.

The results of our experiments will be affected by corpus we used. We did not claim our corpus is clean from noise. In the future, we will attempt to make a cleaner corpus from Wikipedia and make it freely available for public use in the hope it will be a standard for Natural Language Processing research in Indonesian language.

REFRENCES:

[1] E. W. Whittaker and P. C. Woodland, "Comparison of Language Modeling Techniques for Russian and English", in *ICLSP*, 1998.

[2] S. D. Larasati, "IDENTIC Corpus: Morphologically Enriched Indonesian-English Parallel Corpus", in *LREC*, 2012, pp. 902-906.

[3] Stolcke, Andreas. "SRILM-an extensible language modeling toolkit." *INTERSPEECH*. 2002.

[4] S. F. Chen and J. Goodman, "An Empirical Study of Smoothing Techniques for Language Modeling", in *Proceedings of the 34th annual meeting on Association for Computational Linguistics*. Association for Computational Linguistics, 1996, pp. 310-318.

[5] Goodman, Joshua T. "A bit of progress in language modeling." *Computer Speech & Language* 15.4 (2001): 403-434.

[6] Y. W. Teh, "A Hierarchical Bayesian Language Model based on Pitman-Yor Processes", in *Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics.* Association for Computational Linguistics, 2006, pp. 985-992.

[7] Chelba, Ciprian, et al. "One Billion Word Benchmark for Measuring Progress in Statistical Language Modeling." arXiv preprint arXiv:1312.3005 (2013).

[8] Mikolov, Tomas, et al. "Empirical Evaluation and Combination of Advanced Language Modeling Techniques." *INTERSPEECH*. 2011.

[9] M. Adriani and H. R. Designation, "*Research Report on Local Language Computing: Development of Indonesian Language Resources and Translation System*."