Hybrid N-gram Probability Estimation in Morphologically Rich Languages

Hyopil Shin and Hyunjo You

Department of Linguistics, Seoul National University,
Sillim-dong, Gwanak-gu, Seoul, 151-745, Korea
{hpshin, youhyunjo}@snu.ac.kr

Abstract. N-gram language modeling is essential in natural language processing and speech processing. In morphologically rich languages such as Korean, a word usually consists of at least one lemma (content morpheme) and functional morphemes which represent various grammatical. Most word forms in Korean, however, have problems of sparse data and zero probability, because of quite complex morpheme combinations. Thus morpheme-based N-gram modeling is widely used instead of a word sequence modeling. In this paper, we contend that a morpheme-based N-gram is inefficient language modeling in that it inevitably approximates the probability of unnecessary morpheme sequences, so the longer sequences we have, the lower probability estimates we get. We suggest a hybrid method that joins word-based and morpheme-based language modeling. The new method can also be regarded as an extension of a class-based measurement. Our experimental results show that the method produces better probability estimation than the morpheme-based measurement.

Keywords: hybrid N-gram estimation, Language Modeling, Sparse Data, Smoothing

1 Introduction

An N-gram model is usually n-token sequence of words and is essential in natural language processing and speech processing. In morphologically simple language, like English, a bigram is a two-word sequence like *machine translation*, and *machine learning*. N-gram probabilities are computed in terms of the maximum likelihood estimation (MLE). One usually gets the MLE for the parameters of an N-gram model by normalizing counts from a corpus. The MLE, however, confronts major problems of sparse data and eventually zero probability. The longer n-grams one has, the higher chances of sparse data one has. Thus smoothing techniques have been developed to get better estimates for zero or low frequency sequences.

In morphologically rich languages or in agglutinative languages like Korean, sparse data in language modeling are even more serious. Since words are formed by combining lemmas and various affixes together. For example, the bigram *haksaying-i ka-nta* ‘student-subject marker, go-sentence final ending’ has different count from another bigram *haksaying-i ka-ss-ta* ‘student-subject marker, go-past-sentence’ [1], even though there is only one morpheme difference. In typical N-gram modeling, bigrams with the same lemmas but with slightly different affixes do not get the same counts.

Due to the agglutinative characteristics in Korean, most N-gram modeling is based on morphemes not words. Thus, instead of the bigram, *haksayng-i kata* ‘student go’, three bigram sequences such as ‘*haksaying i*’, ‘*i ka*’, and ‘*ka ta*’ are considered. Since the word forms were broken into morphemes, the sequence of morphemes would have higher counts than in the word bigram. Morpheme-based bigrams, however, cannot directly compute the MLE between two

[1] A hyphen is used to separate a stem from affixes for a visual purpose.

lemmas, except when two words have no affixes, like *haksayng yokeum* ‘student fare’. If one uses a trigram in *haksayng-i ka-nta*, instead of a bigram, the ‘*haksayng i ka*’ sequence is considered. But the sequence would have a lower frequency than that of bigrams. Another problem originates from that morpheme-based N-grams inevitably generates an unnecessary morpheme sequence like ‘*i ka*’ in the example above. The sequence may assign an improper amount of probability to unseen N-grams.

Instead of the prevalent N-gram modeling used in Korean, we suggest a new hybrid method of word and morpheme-based N-grams. The method takes advantage of the agglutinative nature of the Korean language and utilizes class-based N-gram modeling. We make use of a variable-length N-gram model in accordance with the structure of word sequences. We focus on lemmas in word sequences and get probability estimates from lemma bigrams or functional morpheme-lemma combinations. This method also works well with unknown words, since probabilities of unseen words are also approximated by variable-length N-grams.

2 The Korean Morphology

Korean is an agglutinative language whose words are formed by joining morphemes together. Two broad classes of morphemes are usually distinguished: stems and affixes. The stem is the main morpheme of the word, providing the main meaning, while the affixes add additional grammatical or lexical meanings. Stems or content morphemes in Korean usually represent major parts-of-speech such as nouns, verbs, and adverbs. Affixes or functional morphemes contribute to various inflections and derivations. In morphologically rich languages, functional morphemes are notorious for their multiple combinations. For example, the following sentence,

(1)
sal + a + ci + eo + o + ass + um + ey + to
live+connective+auxverb+connective+auxverb+past+nominalized+adverbialparticle+adverbial particle
“even though (somebody)has been living”

consists of one content morpheme, two auxiliary verbs and six functional morphemes. Two auxiliary verbs increase the size of the morpheme sequences and expand the meaning of the main verb, *sal* ‘live’, to passive and progressive mood.

Unlike English, grammatical relations in Korean are realized by various affixes, so word order of Korean is relatively free. The subject in an English sentence is determined by the position in a sentence and a subject comes before a main verb. While in Korean, a subject is realized as a stem with a subject marker, -i/-ka, thus the position is quite flexible. Verbs and nouns in Korean can have several hundreds of forms counting inflections and derivations.

Functional morphemes show certain orders especially in verbal inflections. Pre-wordfinal morphemes take precedence over word-final morphemes. Inside a sequence of pre-wordfinal morphemes, an honorific morpheme precedes tense morphemes, and tense morphemes precede some modal morphemes. Functional morphemes can be combined, but there are certain constraints on morpheme combinations in accordance with grammatical functions and categories.

3 Motivations

Most Korean N-gram probabilities have been estimated not by word sequences but by morpheme sequences because word forms usually have multiple morphemes, so unseen N-grams drastically increase. Let’s consider an example, *haksayng-man cwu-nuntey* “student-only give-connective ending”. The bigram probability of *haksayng-man cwu-nuntey* ‘give to only student’ could not be estimated by counting the numbers of the word sequence. Instead, morpheme sequences from a morphological analysis are used for the estimation based on the Markov assumptions.

(2)

$$\begin{aligned}
&\mathrm{P}(\mathrm{haksayng}\ man\ \mathrm{cwu}\ nuntey) \\
&\quad = \mathrm{P}(\mathrm{haksayng}) \times \mathrm{P}(man|\mathrm{haksayng}) \times \\
&\quad \mathrm{P}(\mathrm{cwu} \mid \mathrm{haksayng}\ man) \times \mathrm{P}(nuntey|\mathrm{haksayng}\ man\ \mathrm{cwu}) \\
&\approx \mathrm{P}(\mathrm{haksayng}) \times (man|\mathrm{haksayng}) \times \mathbf{P}(\mathbf{cwu}|\boldsymbol{man}) \times \mathrm{P}(nuntey|\mathrm{cwu})
\end{aligned}$$

The word form, *haksaying-man cwu-nuntey,* would have very low frequency counts or zero counts, but morpheme sequences, on the contrary, would have more counts because the word form splits into several morphemes.

The widely used morpheme-based estimation, however, has several problems. Firstly, it introduces linguistically meaningless morpheme sequences. In (2), a conditional probability *P*(cwu 'give' $|man$ 'only') is computed for a whole probability, but it lacks linguistic significance. Also the morpheme-based probability may not be a real probability, because it would be adding extra probability mass into the equation. The above, *P*(cwu 'give' $|man$ 'only') is one case. Secondly, the morpheme-based estimation requires linguistic knowledge such as parts-of-speech, grammatical relations and so on. Thus a morphological parsing is essential for N-gram estimation. The results from the morphological analysis, however, may vary. The same sequence of morphemes can be differently segmented off according to the morphological parsers. In this case, the probabilities of the same bigram would be different. Another problem related to a morphological analysis is that as stated in section 2, morphological parsers generally restore an original morpheme of the morphologically contracted and transformed forms, so the probabilities of hidden morpheme sequences in the surface form could be estimated. Lastly, the longer affixes we have, the lower the probability we have, since a long sequence of morphemes needs a longer chain rule of probability and more multiplications inevitably have lower probabilities.

On the contrary, we can consider another extreme; lemmatization. A lemma is a set of lexical forms having the same stem. We strip off affixes to have a lemma sequence such as *haksaying cwu* 'student give'. This method is an extension of word-based estimation. However, it oversimplifies bigram sequences. Thus, whenever two lemmas are identical, we will have the same probabilities regardless of the fact that totally different affixes were attached to the sequences.

As preliminary work, we measured perplexities of the two language models; word-based and morpheme-based estimation. We took the Sejong morphologically and semantically tagged corpus consisting of about 800K running words[2]. We chose 10 sentences as a test set and measured bigram perplexities of the two models using the SRILM Toolkit[3]. We concluded that the number of unseen bigrams resulting from the word-based estimation grew constantly, thus we couldn't possibly choose the estimation, so we would like a new model to do something reasonable with unseen bigrams.

4 Related Work

Most work on N-gram modeling in Korean has been done with morpheme-based methods. We will briefly review two related efforts.

Kwon (2000) compared morpheme-based recognition units with syllable-based recognition units for the performance of large vocabulary continuous speech recognition (LVCSR). For the morpheme-based method, Kwon (2000) merged morpheme units to reduce recognition errors. Thus, short morphemes were merged with one consonant, a stem and endings of auxiliary verbs were merged, and a suffix and the following particle were also merged. For the syllable-based

[2] The Sejong Corpus is one of 8 sub-projects of the 21st Sejong Project funded by the Korean government and developed for 10 years. The Corpus collected a wide range of unconstrained materials and endeavored at annotating the data with part-of-speech, morphological, syntactic, and semantic tags.

[3] http://www.speech.sri.com/projects/srilm/

unit, only text corpus and its pronunciation were used. Any linguistic knowledge such as parts-of-speech information and language-specific rules in the morpheme-based unit was not required. Kwon (2000) concluded that the statistical merging method with appropriate linguistic constraints yielded the best recognition accuracy, and the syllable-based approach did not show comparable performance. The syllable-based method is the same as the word-based method.

Park et al. (2007) suggested a method that adjusted the improperly assigned probabilities of unseen-N grams by taking advantage of the agglutinative characteristics of the Korean language. They argued that the grammatically proper class of a morpheme could be predicted by knowing the previous morpheme. By using this characteristic, they tried to prevent grammatically improper N-grams from achieving relatively high probability and to assign more probability mass to proper N-grams. That is, the model reduced the probabilities of unseen N-grams that violate grammatical constraints while distributing more probabilities to grammatically correct unseen N-grams. They used a part-of-speech tagged morpheme as the N-gram model unit because some morphemes were ambiguous when morphemes were classified into content morphemes and functional morphemes.

The method is similar to our hybrid estimation in that it utilized a morpheme sequence type such as content morpheme and functional morpheme occurrences after a specific POS. But they focused only on the probabilities of POS and morpheme types, and tried to redistribute more probabilities to grammatically correct unseen N-grams. This was due to the fact that pure morpheme based approach inevitably overestimates probabilities of morpheme sequences in the case of unseen N-grams.

5 The Hybrid N-gram Probability Estimation

5.1 Lemma-Morpheme-based Probability Measurement

We reviewed two models of N-gram probability estimation in section 4. The word-based estimation is a typical measurement in N-gram modeling, but in morphologically complex languages, it may not turn out to be true, because of large numbers of unseen N-grams. Also the word-based method has a spacing problem in Korean. A word, *eojeol* in a Korean term, is separated from another word by a space. But the spacing is not strictly observed. The morpheme-based measurement, on the contrary, is common but introduces unnecessary morpheme sequences which may overestimate the whole probability.

As a new measurement, we suggest a lemma-morpheme–based N-gram probability. The new method stands mainly on the word or lemma-based estimation and incorporates benefits from morpheme-based estimation. We can illustrate the method as follows.

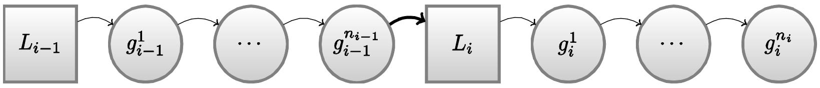

Figure 1-a: Morpheme-based estimation

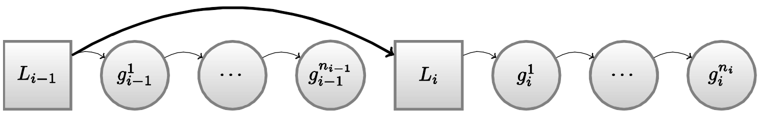

Figure 1-b: Hybrid estimation

Let's assume that a word W_i consists of a leading lexical morpheme L_i and a following cluster G_i of grammatical morphemes. The new method is basically word-based, but morpheme-based modeling is also used inside a word. As seen in figure 1-a, morpheme sequences form bigrams. We can now generalize inside-a-word estimation as follows. $\mathrm{g}_1, \mathrm{g}_2, \dots, \mathrm{g}_\mathrm{n}$ is a cluster of G_i.

(3)

$$\begin{aligned} P(W) &= P(Lg_1g_2 \dots g_n) \\ &\approx P(L)P(g_1|L)P(g_2|g_1) \dots P(g_n|g_{n-1}) \end{aligned}$$

In the new method, however as shown in figure 1-b, a lemma plays a barrier to the functional morphemes-lemma sequence. Instead of joining a functional morpheme and a lemma, a L_{i-1}-L_i lemma sequence is formed. In the previous example, *haksayng-man cwu-nuntey* ‘give to only student’, the combination *man* ‘only’ with *cwu* ‘give’ is blocked, and the lemma sequence, *haksayng cwu* ‘student give’ is formed instead. Then the *P*(*cwu|haksayng*) is approximated. This lemma combination is a kind of ‘variable-length N-gram modeling’ because functional morphemes between the two lemmas can be any size, so the lemma combination may be any N-grams.

We can further generalize the lemma sequences. First of all, let’s consider the following definition of class-based N-grams (Brown et al., 1992)

(4)

$$\begin{aligned} &P(w_i|w_{i-1}) \approx P(c_i|c_{i-1})P(w_i|c_i) \\ &P(w|c) = \frac{C(w)}{C(c)} \\ &P(c_i|c_{i-1}) = \frac{C(c_{i-1}c_i)}{\sum_c C(c_{i-1}c)} \end{aligned}$$

We can apply the class-based N-gram to the hybrid method. We extend the *P*(*cwu|haksayng*) to *P*(*cwu|haksayng**). Here *haksayng** is a class which includes all the word forms of *haksayng*, such as *haksayng* ‘student’, *haksayng-i* ‘student-subject marker’, *haksayng-eykey* ‘to student’, haksayng-*ul* ‘student-object marker’, and *haksayng-man-ul* ‘only student-object-marker’. We can redefine the class-based N-gram equations in example (4) as follows. Here L^* is any word W with the lemma L. L^* is the class of W, and G is a sequence of grammatical morphemes.

(5)

$$\begin{aligned} &P(W_i|W_{i-1}) \approx P(L_i^*|L_{i-1}^*)P(W_i|L_i^*) \\ &P(W|L^*) = \frac{C(W)}{C(L^*)} = \frac{C(LG)}{C(L)} = P(G|L) \\ &P(L_i^*|L_{i-1}^*) = \frac{C(L_{i-1}^*L_i^*)}{\sum_{L^*} C\left(L_{i-1}^*L^*\right)} = \frac{\sum_G C(L_{i-1}GL_i)}{C(L_{i-1})} \end{aligned}$$

Now, our example *haksayng-man cwu-nuntey* ‘give to only student’ is computed as follows.

(6)

$$\begin{aligned} \mathrm{P(cwu - nuntey|haksayng - man)} &\approx \mathrm{P(cwu^*|haksayng^*)P(cwu - nuntey|cwu^*)} \\ &= \mathrm{P(cwu^*|haksayng^*)P(nuntey|cwu)} \\ \mathrm{P(cwununtey|cwu^*)} &= \mathrm{P(nuntey|cwu)} \end{aligned}$$

$$\mathrm{P(cwu^*|haksayng^*)} = \frac{\mathrm{C(haksayng^*cwu^*)}}{\sum_{\mathrm{L^*}} \mathrm{C(haksayng^*L^*)}} = \frac{\sum_{\mathrm{G}} \mathrm{C(haksayng\ G\ cwu)}}{\mathrm{C(haksayng)}}$$

Finally we can reach the following generalization of the hybrid estimation.

(7)

$$\begin{aligned} &P(W_i|W_{i-1}) = P\left(L_i g_{i1} g_{i2} \cdots g_{in_i} | L_{i-1} g_{(i-1)1} g_{(i-1)2} \cdots g_{(i-1)n_{i-1}}\right) \\ &\approx P(L_i^*|L_{i-1}^*)P(W_i|L_i^*) \\ &\approx P(L_i^*|L_{i-1}^*)P(g_{i1}|L_i)P(g_{i2}|g_{i1}) \dots P\left(g_{in}|g_{i(n-1)}\right) \end{aligned}$$

We would like to point out that this approach not only admits an importance of morpheme sequences in N-gram estimation, but also tries to minimize overestimation of the morpheme sequence probabilities. This means that a chaining of morpheme sequences has only local effects within a word, thus the method prevents unnecessary morpheme sequences from being computed and expanded. And, by putting two lemmas together, we can take into account word sequences as well.

5.2 Probability Estimation for Unknown Words

The hybrid estimation also works well in unseen events where certain bigrams have zero counts. Let's consider a new example, *haksaying-tul-eykey-man cwu-si-ess-nuntey* '*give*-honorific-past *to only students*-plural'. Assuming that each word form, such as *haksaying-tul-eykey-man* and *cwu-si*-ess-*nuntey* is unknown, we can approximate each word's probability using the equation shown in example (3).

(8) a. $\mathrm{P}(\mathrm{haksayng}\mathit{tuleykeyman})$
$\approx \mathrm{P}(\mathrm{haksayng})\mathrm{P}(\mathit{tul}|\mathrm{haksayng})\mathrm{P}(\mathit{eykey}|\mathit{tul})\ \mathrm{P}(\mathit{man}|\mathit{eykey})$
b. $\mathrm{P}(\mathrm{cwu}\mathit{siessnuntey}) \approx \mathrm{P}(\mathrm{cwu})\mathrm{P}(\mathrm{si}|\mathrm{cwu})\mathrm{P}(\mathrm{ess}|\mathrm{si})\mathrm{P}(\mathrm{nuntey}|\mathrm{ess})$

The probability of an unknown word is estimated with the inside-a-word equation which is basically the same as morpheme-based estimation. According to our experiment, all the morpheme sequences have counts in the training corpus.

Now consider the case where only one word is unknown in the bigram, *haksaying-tul-eykey-man cwu-si-ess-nuntey* 'give-*honorific-past* to only students-*plural*'. If *haksaying-tul-eykey-man* is unseen, we can estimate the bigram probability as follows.

(9) $\mathrm{P}(\mathrm{haksayng}\mathit{tuleykeyman}\ \mathrm{cwu}\mathit{siessnuntey})$
$= \mathrm{P}(\mathrm{haksayng}\mathit{tuleykeyman})\ \mathbf{P}(\mathbf{cwu}\boldsymbol{siessnuntey}|\mathbf{haksayng}\boldsymbol{tuleykeyman})$

We estimated the probability of the unknown haksaying-*tul-eykey-man* by equation 8-a, so we need to compute a conditional probability boldfaced in equation (9). Below we can use the chain rule to expand the sequence.

(10) $\mathrm{P}(\mathrm{cwu}\mathit{siessuntey}|\mathrm{haksayng}\mathit{tuleykeyman})$
$= \mathrm{P}(\mathrm{cwu}|\mathrm{haksayngtuleykeyman})\ \mathrm{P}(\mathrm{si}|\mathrm{haksayngtuleykeyman\ cwu})$
$\mathrm{P}(\mathrm{ess}|\mathrm{haksayngtuleykeyman\ cwusi})\ \mathrm{P}(\mathrm{nuntey}|\mathrm{haksayngtuleykeyman\ cwusiess})$
$\approx \mathrm{P}(\mathrm{cwu}|\mathrm{haksayngtuleykeyman})\mathrm{P}(\mathrm{si}|\mathrm{cwu})\ \mathrm{P}(\mathrm{ess}|\mathrm{si})\mathrm{P}(\mathrm{nuntey}|\mathrm{ess})$

The problem here is how we can approximate the $\mathrm{P}(\mathrm{cwu}|\mathrm{haksayngtuleykeyman})$. The morpheme-based method would have the bigram estimation as follows.

(11) $\mathrm{P}(\mathrm{cwu}|\mathrm{haksayngtuleykeyman}) \approx \mathrm{P}(\mathrm{cwu}|\mathrm{man})$

This results in a useless sequence. Instead, we can apply our hybrid method and have the following estimation.

(12) $\mathrm{P}(\mathrm{cwu}|\mathrm{haksayngtuleykeyman}) \approx \mathrm{P}(\mathrm{cwu}|\mathrm{haksayng}\ *)$

6 Experiments

We performed two main experiments. One experiment compared the morpheme-based estimation and the hybrid N-gram estimation according the length of a sentence. The subsequent experiment compared the two methods according to the training size. We used the 21st Sejong morphologically and semantically tagged corpus. For the hybrid N-gram computation, we converted the data into ARPA format of the SRILM Toolkit.

For the first experiment, out of about 90K sentences, we randomly selected 941 test sentences with 0.01probability, and we computed bigram probabilities of the sentences. For the second experiment, we divided the training data into 10 sets whose size ranged from 1K to 512K. Then we tested 1,000 sentences from each set and we got 10,000 results. The following table shows a fragment from the first experiment.

Table 1: A part of results of the experiment

n	p1	p2	pp1	pp2
8	-19.608	-20.413	282.50	356.17
35	-67.011	-53.534	82.15	33.85
43	-93.001	-79.231	145.49	69.60
22	-41.804	-38.846	79.46	58.30
55	-104.709	-86.221	80.13	36.95
38	-75.332	-62.034	96.03	42.90
46	-83.680	-68.661	65.94	31.09
13	-22.547	-23.737	54.25	66.97
5	-12.024	-12.024	253.98	253.98
29	-53.523	-47.763	70.08	44.36

We only listed 10 sentences out of a total of 941 sentences in Table 1. All probabilities are log probabilities. The first column *n* means the number of morphemes in a sentence. And the second and the third column, p1 and p2 specify morpheme-based estimation and hybrid estimation respectively. Pp1 and pp2 show the perplexities of the p1 and p2. Figure 2 shows the two perplexities.

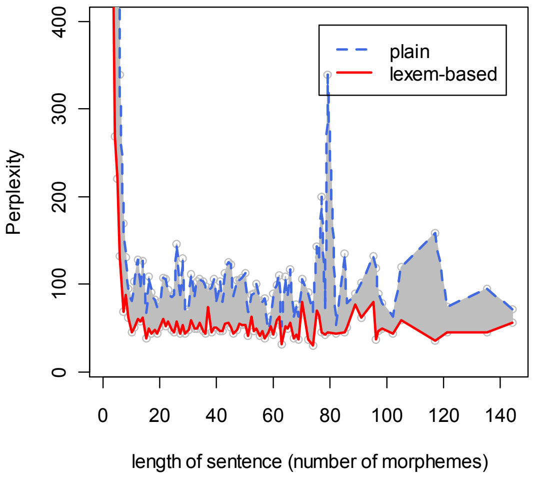

Figure 2: Perplexity by length of a sentence

This shows that the hybrid method gives us better estimation regardless of the length of the sentence. We observed that the longer the sentence on which we estimate the probability, the bigger difference the probability. This shows that the morpheme-based method produces poor estimates when morpheme sequences get longer because unnecessary morpheme sequences occur more in a longer sentence than in a shorter sentence. Table 2 specifies the results according to the size of the training set.

Table 2: The result from the second experiment

training set	pp1	pp2
1K	**26.82270**	**6.831377**
2K	**32.05148**	**8.742955**
4K	**37.98896**	**10.93762**
8K	**45.92444**	**13.99782**
16K	**53.90555**	**17.86562**
32K	**61.72063**	**23.47712**
64K	**69.05248**	**29.36778**
128K	**75.85412**	**36.17477**
256K	**84.05591**	**41.79223**
512K	**94.14079**	**46.99320**

The perplexities of pp1 from the morpheme-based estimation are higher than the perplexities of pp2 from the hybrid estimation. Thus we can conclude that the better model is the hybrid one that has a tighter fit to the test data. Figure 3 shows the perplexities graphically.

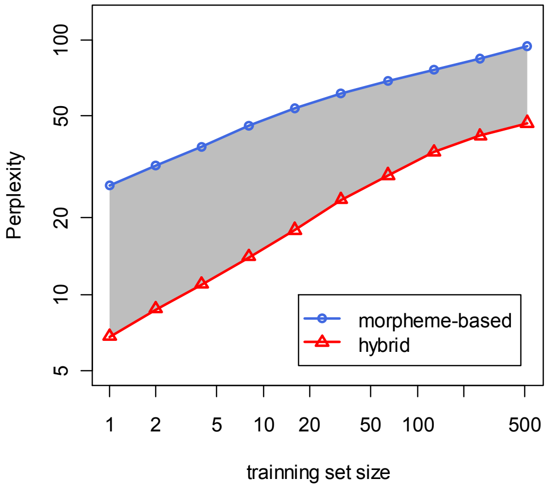

Figure 3: Perplexities by the training set size

Figure 3, however, does not show the probability distributions at the position of each training data size. Thus, we bean-plotted the data as shown in Figure 4. The beanplot shows how 1,000 data from each training set are distributed (Kampstra 2008).

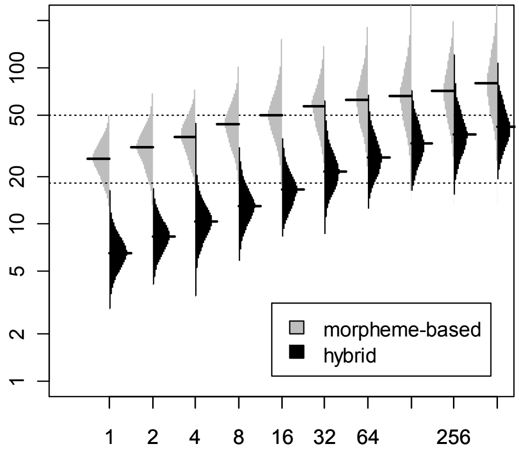

Figure 4: Perplexity distribution by the training data size

As shown in Figure 4, the two methods have largely different distributions, even though there are some overlaps in the bigger size of the training data. This also validates our conclusions.

7 Discussion and Conclusions

Thus far, we explained the new method of N-gram probability estimation in morphologically complex languages. Someone may argue that the hybrid method also has a problem of unnecessary morpheme sequences in that inside a word, the hybrid and the morpheme-based method approximates the same sequence of morphemes, thus there will be no big differences between the two.

The hybrid method, however, can produce a better estimation of morpheme sequences in a word. According to our experiment, the average number of morphemes in a Korean word is 2.5. This means that a word consists of about 2.5 morphemes including lemmas. Example (1) consisting of 9 morphemes is an extreme case. And if we investigate the structure of the sequence, we can figure out that auxiliary verbs are required for longer morpheme sequences. In the case of (1), two auxiliary verbs, *ci* ‘passive morpheme’ and *o*, ‘progressive morpheme’ are combined. Without auxiliary verbs, longer morpheme sequences are not usually permitted. Also spacing is not quite strict in Korean. Someone may separate auxiliary verbs from the main verb, so the sequence, *salacieoassumeyto* ‘even though (somebody)has been living’ can be split into three words like *sala cie oassumeyto* ‘even though (somebody)has been living’.

We take only affixes as functional morphemes. So auxiliary verbs are content morphemes and can be a barrier to the functional morpheme and lemma combination. Even though no spaces appear in the structure, our hybrid method approximates the probabilities of the P(*ci*|*sal**) and the P(*o*|*ci**), instead of the P(*ci*|*a*) and the P(*o*|*a*) respectively. The lemma and class-based estimation can give us a better estimate of the true probability of Korean.

Our next job is to incorporate the language model into a speech recognition system and see how the new method contributes to the overall performance. We strongly believe that the hybrid method can be a better N-gram estimator, both for natural language processing, and for speech processing in morphologically rich languages.

References

Brown, Peter F. and Vincent J. Della Pietra. 1992. Class-Based N-gram Models of Natural Language. *Computational Linguistics*, 18(4):467-479.

Chen, S. 1996. *Building Probabilistic Models for Natural Language*, PhD. Thesis. Harvard University.

Jelinek, Frederick. 1998. *Statistical Methods for Speech Recognition*. The MIT Press.

Jurafsky, Daniel and James H. Martin. 2008. *Speech and Language Processing-An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition*, 2nd edition. Pearson Education.

Justo, Raquel and M. Ines Torres. 2007. Word Segments in Category-Based Language Models from Automatic Speech Recognition, *Lecture Notes in Computer Science* 4477:249-256. Springer-Verlag.

Kampstra, Peter. 2008. Beanplot: A Boxplot Alternative for Visual Comparison of Distributions. *Journal of Statistical Software*, volume 28.

Kwon, Oh-Wook. 2000. Performance of LVCSR with Morpheme-based and Syllable-based Recognition Units. *2000 IEEE International Conference on Acoustics, Speech and Signal Processing*, volume 3:1567-1570.

Manning, Christopher D. and Hinrich Schutze. 1999. *Foundations of Statistical Natural Language Processing*. The MIT Press.

Niesler, T.R. and P.C. Woodland. 1996. A Variable-length Category-based N-gram Language Model. *1996 International Conference on the Acoustics, Speech, and Signal Processing*, volume 3:164-167.

Park, Jae-Hyun, Young-In Song, and Hae-Chang Rim. 2007. Smoothing Algorithm for N-Gram Model Using Agglutinative Characteristic of Korean. *Proceedings of the International Conference on Semantic Computing*:397-404.

Rosenfeld, R. and X. Huang. 1991. Improvements in Stochastic Language Modeling, *HLT '91: Proceedings of the Workshop on Speech and Natural Language*:107-111.