Syllable-level Neural Language Model for Agglutinative Language

Seunghak Yu* **Nilesh Kulkarni*** **Haejun Lee** **Jihie Kim**
Samsung Electronics Co. Ltd., South Korea
{seunghak.yu, n93.kulkarni, haejun82.lee, jihie.kim}@samsung.com

Abstract

Language models for agglutinative languages have always been hindered in past due to myriad of agglutinations possible to any given word through various affixes. We propose a method to diminish the problem of out-of-vocabulary words by introducing an embedding derived from syllables and morphemes which leverages the agglutinative property. Our model outperforms character-level embedding in perplexity by 16.87 with 9.50M parameters. Proposed method achieves state of the art performance over existing input prediction methods in terms of Key Stroke Saving and has been commercialized.

1 Introduction

Recurrent neural networks (RNNs) exhibit dynamic temporal behavior which makes them ideal architectures to model sequential data. In recent times, RNNs have shown state of the art performance on tasks of language modeling (RNN-LM), beating the statistical modeling techniques by a huge margin (Mikolov et al., 2010; Lin et al., 2015; Kim et al., 2016; Miyamoto and Cho, 2016). RNN-LMs model the probability distribution over the words in vocabulary conditioned on a given input context. The sizes of these networks are primarily dependent on their vocabulary size.

Since agglutinative languages, such as Korean, Japanese, and Turkish, have a huge number of words in the vocabulary, it is considerably hard to train word-level RNN-LM. Korean is agglutinative in its morphology; words mainly contain different morphemes to determine the meaning of the word hence increasing the vocabulary size for language model training. A given word in Korean could have similar meaning with more than 10 variations in the suffix as shown in Table 1.

Various language modeling methods that rely on character or morpheme like segmentation of words have been developed (Ciloglu et al., 2004; Cui et al., 2014; Kim et al., 2016; Mikolov et al., 2012; Zheng et al., 2013; Ling et al., 2015). (Chen et al., 2015b) explored the idea of joint training for character and word embedding. Morpheme based segmentation has been explored in both Large Vocabulary Continuous Speech Recognition (LVCSR) tasks for Egyptian Arabic (Mousa et al., 2013) and German newspaper corpus (Cotterell and Schütze, 2015). (Sennrich et al., 2015) used subword units to perform machine translation for rare words.

Morpheme distribution has a relatively smaller frequency tail as compared to the word distribution from vocabulary, hence avoids over-fitting for tail units. However, even with morpheme segmentation the percentage of out-of-vocabulary (OOV) words is significantly high in Korean. Character embedding in Korean is unfeasible as the context of the word is not sufficiently captured by the long sequence which composes the word. We select as features syllable-level embedding which has shorter sequence length and morpheme-level embedding to capture the semantics of the word.

We deploy our model for input word prediction on mobile devices. To achieve desirable performance we are required to create a model that has as small as possible memory and CPU footprint without compromising its performance. We use differentiated softmax (Chen et al., 2015a) for the output layer. This method uses more parameters for the words that are frequent and less for the ones that occur rarely. We achieve better performance than existing approaches in terms of Key Stroke Savings (KSS) (Fowler et al., 2015) and our approach has been commercialized.

* Equal contribution

Word	Morpheme	English
그가	그 + 가	he
그는	그 + 는	he
그에게	그 + 에게	to him
그도	그 + 도	him(he) also
그를	그 + 를	him
그의	그 + 의	his

Table 1: Example of variation of a base word '그 (He)'. It can have more than 10 variation forms according to its postposition.

2 Proposed Method

Following sections propose a model for agglutinative language. In Section 2.1 we discuss the basic architecture of the model as detailed in Figure 1, followed by Section 2.2 that describes our embeddings. In Section 2.3 we propose an adaptation of differentiated softmax to reduce the number of model parameters and improve computation speed.

2.1 Language Model

Overall architecture of our language model consists of a) embedding layer, b) hidden layer, c) softmax layer. Embedding comprises of syllable-level and morpheme-level embedding as described in Section 2.2. We combine both embedding features and pass them through a highway network (Srivastava et al., 2015) which act as an input to the hidden layers. We use a single layer of LSTM as hidden units with architecture similar to the non-regularized LSTM model by (Zaremba et al., 2014). The hidden state of the LSTM unit is affine-transformed by the softmax function, which is a probability distribution over all the words in the output vocabulary.

2.2 Syllable & Morphological Embedding

We propose syllable-level embedding that attenuates OOV problem. (Santos and Zadrozny, 2014; Kim et al., 2016) proposed character aware neural networks using convolution filters to create character embedding for words. We use convolution neural network (CNN) based embedding method to get syllable-level embedding for words. We use 150 filters that consider uni, bi, tri and quad syllable-grams to create a feature representation for the word. This is followed by max-pooling to concatenate the features from each class of filters resulting in a syllable embedding representation

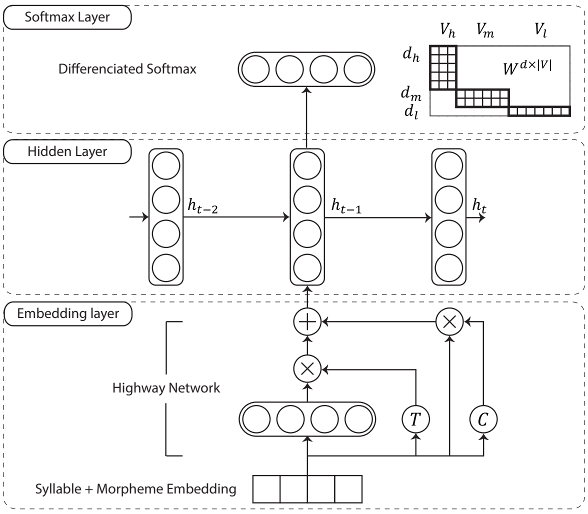

Figure 1: Overview of the proposed method. T and C are the transform gate and carry gate of the highway network respectively

for the word. Figure 2 in the left half shows an example sentence embedded using the syllable-level embedding.

Figure 3 highlights the difference between various embedding and the features they capture. The syllable embedding is used along with a morphological embedding to provide richer features for the word. The majority of words (95%) in Korean has at most three morphological units. Each word can be broken into start, middle, and end unit. We embed each morphological unit by concatenating to create a joint embedding for the word. Advantage of morphological embedding over syllable is all the sub-units have an abstract value in the language and this creates representation for words relying on the usage of these morphemes. Both morphological and syllable embeddings are concatenated and fed through a highway network (Srivastava et al., 2015) to get a refined representation for the word as shown in the embedding layer for Figure 1.

2.3 Differentiated Softmax

The output layer models a probability distribution over words in vocabulary conditioned on the given context. There is a trade-off between required memory and computational cost which determines the level of prediction. To generate a complete word, using morpheme-level predictions requires beam search which is expensive as compared to word-level predictions. Using beam search to predict the word greedily does not adhere to the com-

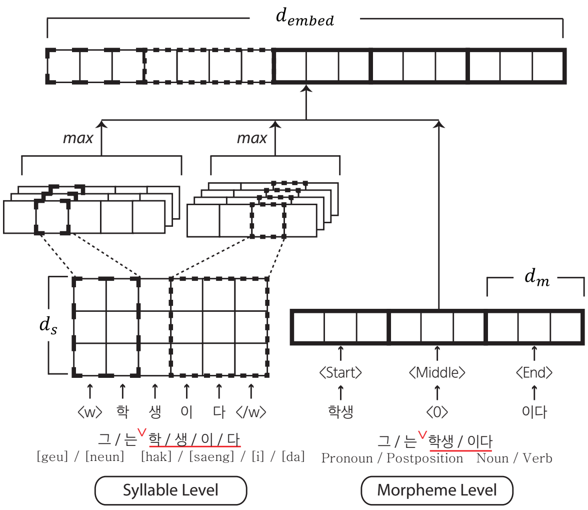

Figure 2: Proposed embedding method for agglutinative languages. We take an input word as syllable and morpheme level, embed them separately and concatenate them to make an entire embedding.

putational requirements set forth for mobile devices. Thus, we have to choose word-level outputs although it requires having a vocabulary of over 0.2M words to cover 95% of the functional word forms. Computing a probability distribution function for 0.2M classes is computational intensive and overshoots the required run-time and the allocated memory to store the model parameters.

Therefore, the softmax weight matrix, $W_{softmax}$, needs to be compressed as it is contributing to huge model parameters. We initially propose to choose an appropriate rank for the $W_{softmax}$ in the following approximation problem; $W_{softmax} = W_A \times W_B$, where W_A and W_B have ranks less than r. We extend the idea of low rank matrix factorization in (Sainath et al., 2013) by further clustering words into groups and allowing a different low rank r' for each cluster. The words with high frequency are given a rank, r_1, such that $r_1 \geq r_2$ where r_2 is the low rank for the words with low frequency. The core idea being, words with higher frequency have much richer representation in higher dimensional space, whereas words with low frequency cannot utilize the higher dimensional space well.

We observe that 87% of the words appear in the tail of the distribution by the frequency of occurrence. We provide a higher rank to the top 2.5% words and much lower rank to the bottom 87%. This different treatment reduces the number of parameters and leads to better modeling.

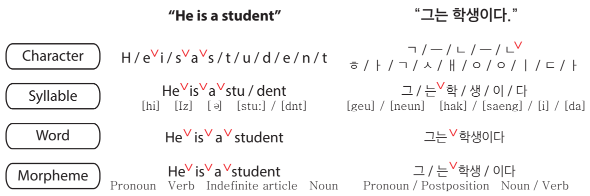

Figure 3: Comparison of various embedding levels. In case of Korean, syllable can be used as a basic unit of sequence to solve OOV with shorter sequence length compare to character level. Also, morpheme level is effective to make the size of vocabulary smaller.

3 Experiment Results

3.1 Setup

We apply our method to web crawled dataset consisting on news, blogs, QA. Our dataset consists of over 100M words and over 10M sentences. For morpheme-level segmentation, we use lexical analyzer and for syallable-level we just syllabify the dataset. We empirically test our model and its input vocabulary size is around 20K morphemes and 3K syllables. The embedding size for morpheme is 52 and that for syllable is 15. We use one highway layer to combine the embeddings from syllable and morpheme. Our hidden layer consists of 500 LSTM units. The differentiated softmax outputs the model's distribution over the 0.2M words in the output vocabulary with top 5K (by frequency) getting a representation dimension (low rank in $W_{softmax}$) of 152, next 20K use a representation dimension of 52 and the rest 175K get a representation dimension of 12. All the compared models have word level outputs and use differentiated softmax.

3.2 Comparison of embedding methods

We randomly select 10% of our crawled data (10M words, 1M sentences) to compare embedding methods as shown in Table 2. We test character, syllable, morpheme and word-level embeddings. The word-level embedding has the highest number of parameters but has the worst performance. As expected breaking words into their subforms improves the language model. However, our experiment reaches its peak performance when we use syllable level embeddings. To improve the performance even further we propose using syllable

Embedding	Param.	Perplexity	Vocab.
Word	15.72M	327.17	200K
Morph	6.61M	283.54	20K
Character	8.66M	235.52	40
Syl	8.71M	231.30	3K
Syl + Morph	**9.50M**	**218.65**	**23K**

Table 2: Results of different embedding methods. Param. : Total model paramerters, Vocab: Input vocabulary size, Syl : Syllable, Morph: Morpheme.

and morpheme which outperforms all the other approaches in terms of perplexity.

3.3 Performance evaluation

Proposed method shows the best performance compared to other solutions in terms of Key Stroke Savings (KSS) as shown in Table 3. KSS is a percentage of key strokes not pressed compared to a vanilla keyboard which does not have any prediction or completion capabilities. Every user typed characters using the predictions of the language model counts as key stroke saving. The dataset[1] used to evaluate KSS was manually curated to mimic user keyboard usage patterns.

The results in Table 3 for other commercialized solutions are manually evaluated due to lack of access to their language model. We use three evaluators from inspection group to cross-validate the results and remove human errors. Each evaluator performed the test independently for all the other solutions to reach a consensus. We try to minimize user personalization in predictions by creating a new user profile while evaluating KSS.

The proposed method shows 37.62% in terms of KSS and outperforms compared solutions. We have achieved more than 13% improvement over the best score among existing solutions which is 33.20% in KSS. If the user inputs a word with our solution, we require on an average 62.38% of the word prefix to recommend the intended word, while other solutions need 66.80% of the same. Figure 4 shows an example of word prediction across different solutions. In this example, the predictions from other solutions are same irrespective of the context, while the proposed method treats them differently with appropriate predictions.

[1]The dataset consists of 67 sentences (825 words, 7,531 characters) which are collection of formal and informal utterances from various sources. It is available at https://github.com/meinwerk/SyllableLevelLanguageModel

Developer	KSS(%)
Proposed	**37.62**
Swiftkey	33.20
Apple	31.90
Samsung	31.40

Table 3: Performance comparison of proposed method and other commercialized keyboard solutions by various developers.

Figure 4: Example of comparison with other commercialized solutions. Predicted words for the Context A (rain heavily) and Context B (too much rice). Other solutions make same prediction regardless of the context (only consider the last two words of context).

4 Conclusion

We have proposed a practical method for modeling agglutinative languages, in this case Korean. We use syllable and morpheme embeddings to tackle large portion of OOV problem owing to practical limit of vocabulary size and word-level prediction with differentiated softmax to compress size of model to a form factor making it amenable to running smoothly on mobile device. Our model has 9.50M parameters and achieves better perplexity than character-level embedding by 16.87. Our proposed method outperforms the existing commercialized keyboards in terms of key stroke savings and has been commercialized. Our commercialized solution combines above model with n-gram statistics to model user behavior thus supporting personalization.

References

Welin Chen, David Grangier, and Michael Auli. 2015a. Strategies for training large vocabulary neural language models. *arXiv preprint arXiv:1512.04906* .

Xinxiong Chen, Lei Xu, Zhiyuan Liu, Maosong Sun, and Huanbo Luan. 2015b. Joint learning of character and word embeddings. In *Twenty-Fourth International Joint Conference on Artificial Intelligence*.

T Ciloglu, M Comez, and S Sahin. 2004. Language modelling for turkish as an agglutinative language. In *Signal Processing and Communications Applications Conference, 2004. Proceedings of the IEEE 12th*. IEEE, pages 461–462.

Ryan Cotterell and Hinrich Schütze. 2015. Morphological word-embeddings. In *HLT-NAACL*. pages 1287–1292.

Qing Cui, Bin Gao, Jiang Bian, Siyu Qiu, and Tie-Yan Liu. 2014. Learning effective word embedding using morphological word similarity. *arXiv preprint arXiv:1407.1687* .

Andrew Fowler, Kurt Partridge, Ciprian Chelba, Xiaojun Bi, Tom Ouyang, and Shumin Zhai. 2015. Effects of language modeling and its personalization on touchscreen typing performance. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*. ACM, pages 649–658.

Yoon Kim, Yacine Jernite, David Sontag, and Alexander M Rush. 2016. Character-aware neural language models. In *Thirtieth AAAI Conference on Artificial Intelligence*.

Rui Lin, Shujie Liu, Muyun Yang, Mu Li, Ming Zhou, and Sheng Li. 2015. Hierarchical recurrent neural network for document modeling. In *EMNLP*. pages 899–907.

Wang Ling, Tiago Luís, Luís Marujo, Ramón Fernandez Astudillo, Silvio Amir, Chris Dyer, Alan W Black, and Isabel Trancoso. 2015. Finding function in form: Compositional character models for open vocabulary word representation. *arXiv preprint arXiv:1508.02096* .

Tomas Mikolov, Martin Karafiát, Lukas Burget, Jan Cernockỳ, and Sanjeev Khudanpur. 2010. Recurrent neural network based language model. In *Interspeech*. volume 2, page 3.

Tomáš Mikolov, Ilya Sutskever, Anoop Deoras, Hai-Son Le, Stefan Kombrink, and Jan Cernocky. 2012. Subword language modeling with neural networks. *preprint (http://www. fit. vutbr. cz/imikolov/rnnlm/char. pdf)* .

Yasumasa Miyamoto and Kyunghyun Cho. 2016. Gated word-character recurrent language model. *arXiv preprint arXiv:1606.01700* .

Amr El-Desoky Mousa, Hong-Kwang Jeff Kuo, Lidia Mangu, and Hagen Soltau. 2013. Morpheme-based feature-rich language models using deep neural networks for lvcsr of egyptian arabic. In *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*. IEEE, pages 8435–8439.

Tara N Sainath, Brian Kingsbury, Vikas Sindhwani, Ebru Arisoy, and Bhuvana Ramabhadran. 2013. Low-rank matrix factorization for deep neural network training with high-dimensional output targets. In *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*. IEEE, pages 6655–6659.

Cicero D Santos and Bianca Zadrozny. 2014. Learning character-level representations for part-of-speech tagging. In *Proceedings of the 31st International Conference on Machine Learning (ICML-14)*. pages 1818–1826.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2015. Neural machine translation of rare words with subword units. *arXiv preprint arXiv:1508.07909* .

Rupesh Kumar Srivastava, Klaus Greff, and Jürgen Schmidhuber. 2015. Highway networks. *arXiv preprint arXiv:1505.00387* .

Wojciech Zaremba, Ilya Sutskever, and Oriol Vinyals. 2014. Recurrent neural network regularization. *arXiv preprint arXiv:1409.2329* .

Xiaoqing Zheng, Hanyang Chen, and Tianyu Xu. 2013. Deep learning for chinese word segmentation and pos tagging. In *EMNLP*. pages 647–657.