Character level language modeling through knowledge distillation

Haruna Tanaka Keio University
Nicholas Mehr Keio University
Toshinori Kitamura Keio University
Xiaobai Sun Keio University
Masayuki Takeda Keio University
Kai Ru Keio University

Abstract

Language models (LMs) are a crucial element in Natural Language Processing (NLP). A good LM can describe the relationship between words and help machines to understand the meaning of words or sentences. Currently, research has favored word-embeddings over character-embeddings (the basic elements of words). In this paper, we focus on character-level word embeddings. The vocabulary size of a character is small compared to that of a word. Therefore, we employ a knowledge distillation method to train our models. We attempt to have the model learn from a bi-directional embedding representation from transformer (BERT) character-aware convolutional neural network. We then use this model to examine several examples of similar words.

1 Introduction

LMs learn the probability of a sentence or word occurrence based on examples from texts. They are applied to sentence generators, question answering, sentiment analysis, machine translation, etc. Originally, statistical approaches such as N-grams were used as LMs. More recently, neural language models, such as the neural network based LM (NNLM) have far surpassed N-grams and achieved significantly better results[1]. Convolutional and Recurrent Neural Network (CNNs[2] and RNNs, respectively) based approaches have also been developed. In order to model a word, one needs to understand the context. A context is built from several words and for this a concept of attention has been developed where focus is directed to specific words in order to generate the context of a sentence. Based on this attention mechanism, the transformer model was developed where multiple layers are used to analyze context and focus words[3]. Since the introduction of the BERT model, nowandays transformers use bidirectional layers to analyze context[4].

2 Related Work

Traditional LMs and word-embeddings. Traditional LMs have used N-th sequence Markov models where smoothing and counting are used in calculating the N-gram probabilities [5]. Conducting training for these count-based techniques is simple yet, despite these smoothing techniques, the estimation of rare N-gram probabilities may be reduced due to the sparsity of data. NNLMs, in contrast, employ word-embeddings: the parameters are learned during the training process, and the word-embeddings of semantically close words display similarities in vector space [5]. For instance, they are unable to establish priors with word-embeddings across the vector spaces of uneventful, uneventfully, eventfully, and eventful.

Interpretability of NLP Models. NNLM's produce high accuracy but suffer from the same interpretability issues as other deep neural network (DNN) models. New approaches in computer vision have assisted in evaluating DNNs[6] using latent representation accompanied by interpretable input values, such as words or image pixels, which facilitate the evaluation of salience and contribution of inputs. Existing groups fall into three categories; gradient-based, methods that involve layer-wise, and inversion-based propagation. The methods have shown that a quantitative evaluation of intermediate layers improve ones understanding of the DNNs inner layers [6].

LMs with Deeper Self-Attention. Character-level LMs have been thus far dominated by RNNs. RNN variants such as LSTMs have provided good results for character-level LMs[7]. As it concerns our model, Al-Rfou et. al.'s work illustrated how LMs at the byte-level performed better than word LMs using a network of 64 stacked transformer layers. Importantly, they showed that a deep transformer model could provide better results than a typical RNN with the addition of loss functions at the intermediate sequence and outer layers.

Morphological Segmentation Cottrell et. al. [8], presented a context-free-grammar (CFG) approach where i) the intuition of words themselves is captured and may belong to different constituent components and ii) the order in which affixes are attached is represented in a hierarchical structure. This CFG-based approach is novel and it is argued that this hierarchical approach is more appropriate than the canonical flat segmentation in Markov models. The takeaway from their research is that the hierarchical structures can provide greater segmentation of morphemes and lead to greater accuracy in general.

3 Model Architecture

We used a modified Character-Aware CNN model[5], adding a character embedding gate shown in Figure 1 1. The basic idea is to make the model output match a pre-trained Character-level LM. We do this by giving the Character-Aware CNN the role of a "teacher" model. The Character-Aware CNN is composed of a character embedding layer, a CNN layer (with several kernel-sized filters), pooling in the character-sequential direction, a highway architecture (Gate) and a LSTM. We also proposed adding a bi-directional LSTM and Softmax gate between the character embedding layer and the CNN. The Softmax gate is expected to regulate information from each character: the mean squared error (MSE) between the output of the model (525-dim vector output from highway layer) and that of the pre-trained character aware CNN. If the model's output can match that of the pre-trained model, the bi-directional LSTM gate can avoid unnecessary character embeddings entering the CNN and highway layers, allowing the CNN to obtain segmented character-level embeddings. We expect that the bi-directional LSTM gate can somehow segment words into morphs.

4 Results

We trained our models using the Glove dataset.The training loss decreased as shown in Figure 4, where both the model had low loss according to training and test data. We analyzed word embedding output from the highway layers of the model, checking the 7 closest word embeddings among randomly chosen words as shown in Figure 3. The gate result via bidirectional LSTM are shown in Figure 2. The words were divided between a word and a padded area. However, we were unable to observe any morph boundaries.

5 Conclusion

We built a Bi-directional LSTM to implement a Gate for input character-level embeddings. The gate was expected to reduce inputs and decrease training loss. The proposed network was able to find similar character-level words. However, the proposed network could not find the morph gaps via the LSTM outputs, and it could not learn the embedding of the teacher model. Possible points for improvement include: addition of a loss at second or third similar hidden layers, and implementation of mutual information into the loss. The character aware NNs do not appear to be well suited for learning because they did not find similar words well.

Acknowledgements We are grateful to Stephen Fitz for his fruitful comments, corrections and inspiration.

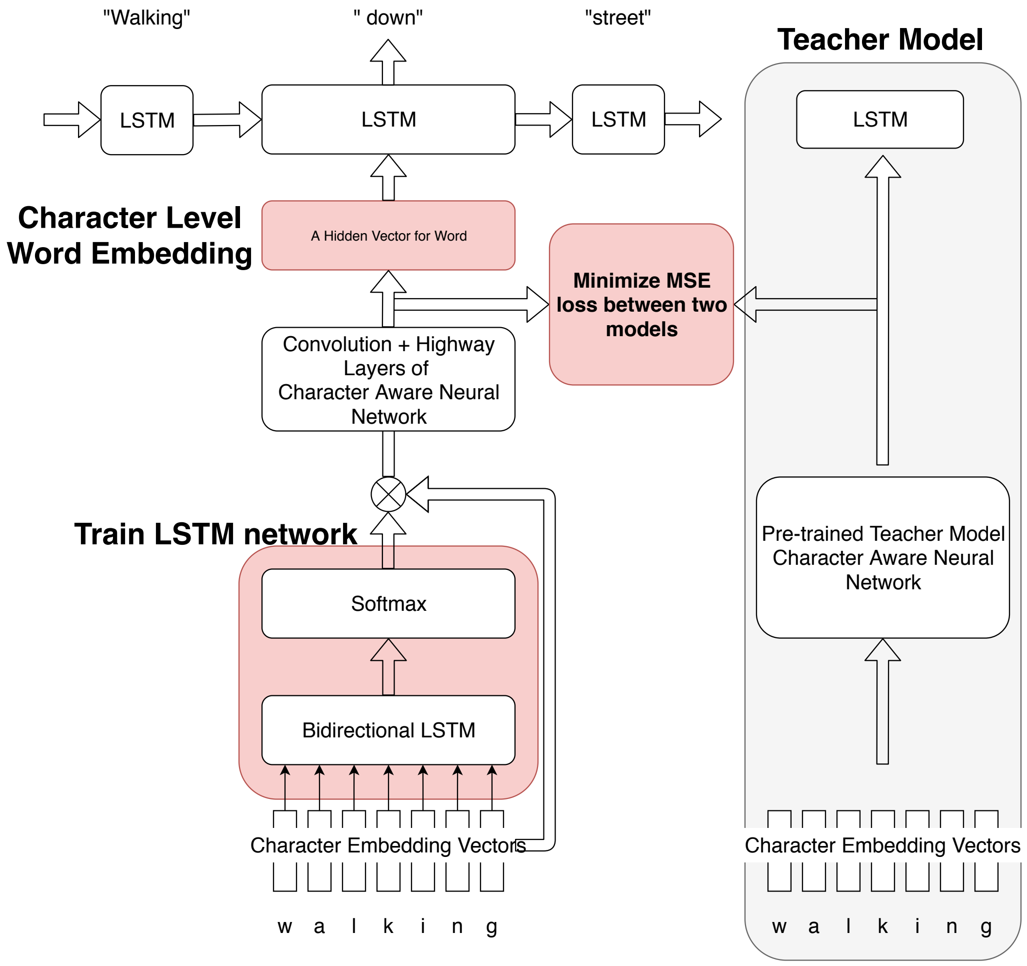

"Walking"
" down"
"street"
Teacher Model
LSTM
LSTM
LSTM
LSTM
Character Level Word Embedding
A Hidden Vector for Word
Minimize MSE loss between two models
Convolution + Highway Layers of Character Aware Neural Network
Train LSTM network
Softmax
Bidirectional LSTM
Pre-trained Teacher Model Character Aware Neural Network
Character Embedding Vectors
Character Embedding Vectors
w a l k i n g
w a l k i n g

Figure 1: Proposed model architecture

References

[1] Yoshua Bengio, Réjean Ducharme, Pascal Vincent, and Christian Jauvin. A neural probabilistic language model. *Journal of machine learning research*, 3(Feb):1137–1155, 2003.

[2] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[3] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008, 2017.

[4] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

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

[6] Chaoyu Guan, Xiting Wang, Quanshi Zhang, Runjin Chen, Di He, and Xing Xie. Towards a deep and unified understanding of deep neural models in nlp. 2019.

[7] Choe et al. Al-Rfou. Character-level language models with depper self-attention. *Google AI Language*, 2018.

[8] Kumar Cottrell and Schutze. Morphological segmentation inside-out. *ENMLP*, 2016.

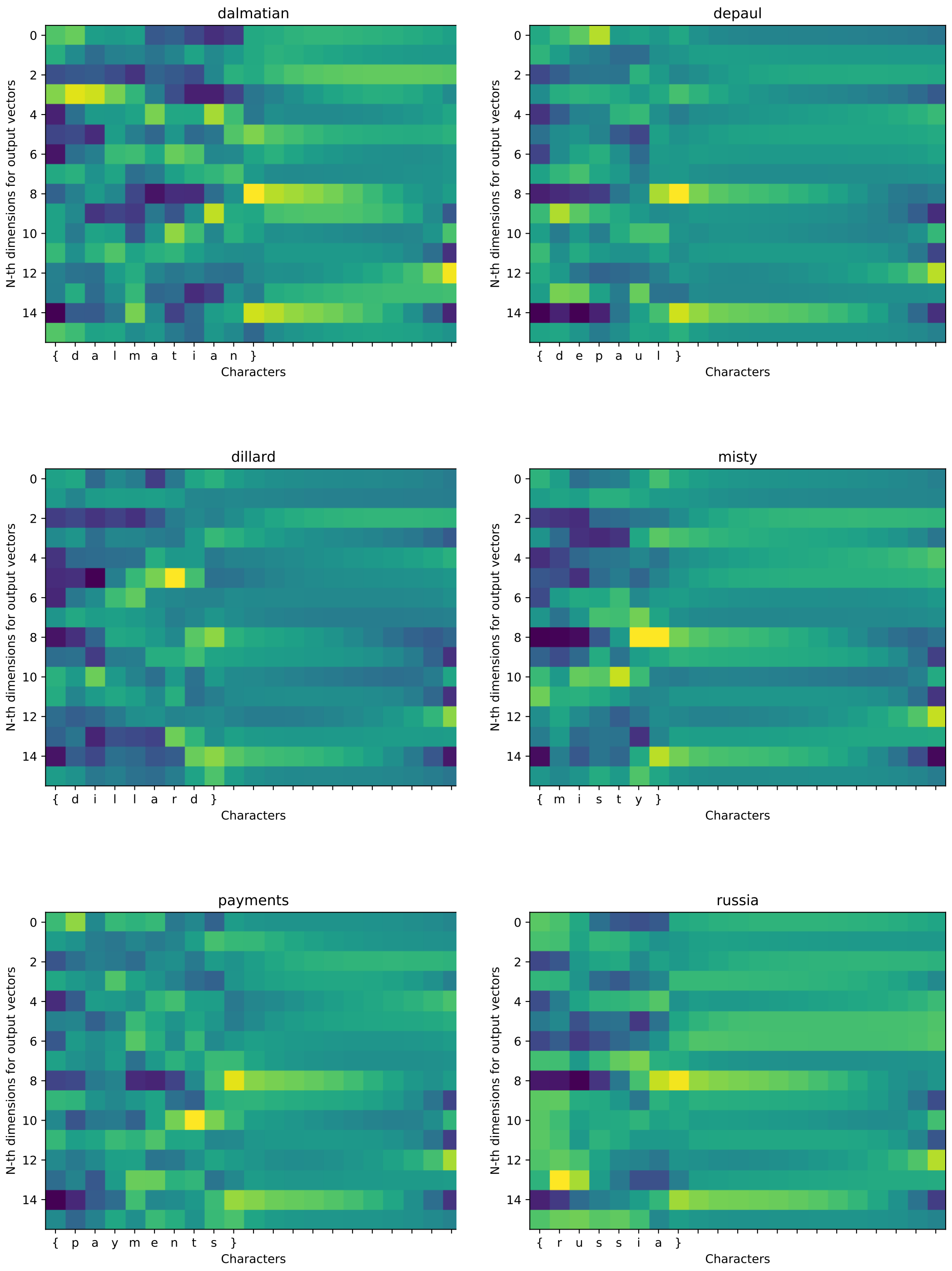
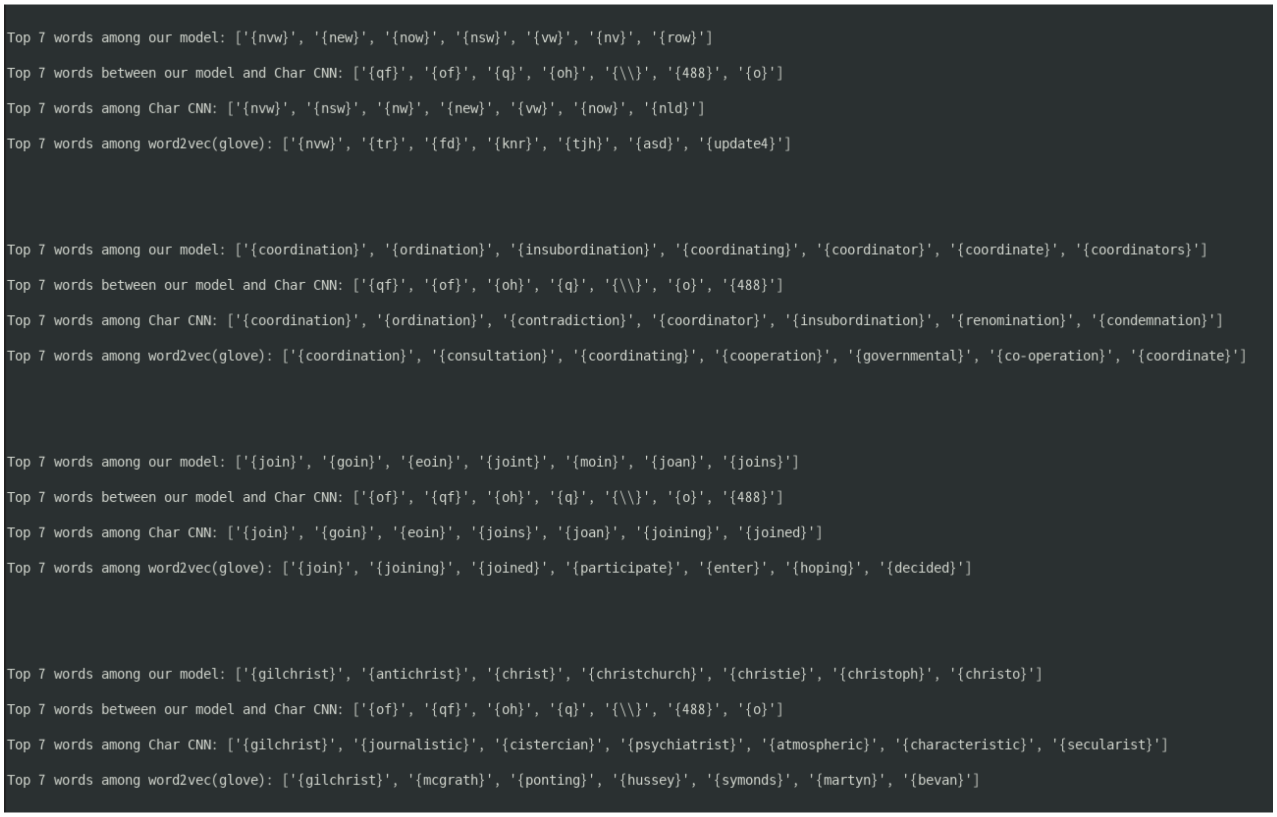


Figure 2: Result of LSTM gate



```
Top 7 words among our model: ['{nvw}', '{new}', '{now}', '{nsw}', '{vw}', '{nv}', '{row}']

Top 7 words between our model and Char CNN: ['{qf}', '{of}', '{q}', '{oh}', '{\\}', '{488}', '{o}']

Top 7 words among Char CNN: ['{nvw}', '{nsw}', '{nw}', '{new}', '{vw}', '{now}', '{nld}']

Top 7 words among word2vec(glove): ['{nvw}', '{tr}', '{fd}', '{knr}', '{tjh}', '{asd}', '{update4}']



Top 7 words among our model: ['{coordination}', '{ordination}', '{insubordination}', '{coordinating}', '{coordinator}', '{coordinate}', '{coordinators}']

Top 7 words between our model and Char CNN: ['{qf}', '{of}', '{oh}', '{q}', '{\\}', '{o}', '{488}']

Top 7 words among Char CNN: ['{coordination}', '{ordination}', '{contradiction}', '{coordinator}', '{insubordination}', '{renomination}', '{condemnation}']

Top 7 words among word2vec(glove): ['{coordination}', '{consultation}', '{coordinating}', '{cooperation}', '{governmental}', '{co-operation}', '{coordinate}']



Top 7 words among our model: ['{join}', '{goin}', '{eoin}', '{joint}', '{moin}', '{joan}', '{joins}']

Top 7 words between our model and Char CNN: ['{of}', '{qf}', '{oh}', '{q}', '{\\}', '{o}', '{488}']

Top 7 words among Char CNN: ['{join}', '{goin}', '{eoin}', '{joins}', '{joan}', '{joining}', '{joined}']

Top 7 words among word2vec(glove): ['{join}', '{joining}', '{joined}', '{participate}', '{enter}', '{hoping}', '{decided}']



Top 7 words among our model: ['{gilchrist}', '{antichrist}', '{christ}', '{christchurch}', '{christie}', '{christoph}', '{christo}']

Top 7 words between our model and Char CNN: ['{of}', '{qf}', '{oh}', '{q}', '{\\}', '{488}', '{o}']

Top 7 words among Char CNN: ['{gilchrist}', '{journalistic}', '{cistercian}', '{psychiatrist}', '{atmospheric}', '{characteristic}', '{secularist}']

Top 7 words among word2vec(glove): ['{gilchrist}', '{mcgrath}', '{ponting}', '{hussey}', '{symonds}', '{martyn}', '{bevan}']
```

Figure 3: Result of similar words

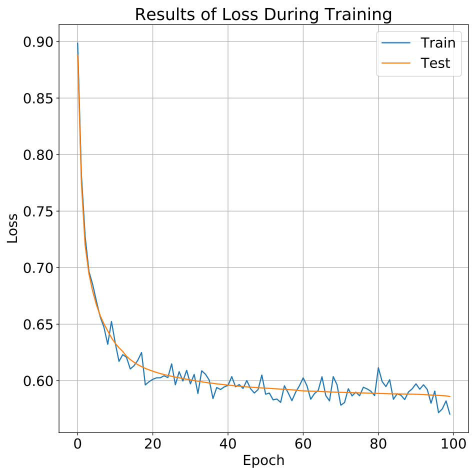


Figure 4: Result of training loss