Rich Character-Level Information for Korean Morphological Analysis and Part-of-Speech Tagging

Andrew Matteson
Korea University
amatteson@korea.ac.kr

Chanhee Lee
Korea University
chanhee0222@korea.ac.kr

Heuiseok Lim[*]
Korea University
limhseok@korea.ac.kr

Young-Bum Kim
Amazon Alexa
youngbum@amazon.com

Abstract

Due to the fact that Korean is a highly agglutinative, character-rich language, previous work on Korean morphological analysis typically employs the use of sub-character features known as graphemes or otherwise utilizes comprehensive prior linguistic knowledge (i.e., a dictionary of known morphological transformation forms, or actions). These models have been created with the assumption that character-level, dictionary-less morphological analysis was intractable due to the number of actions required. We present, in this study, a multi-stage action-based model that can perform morphological transformation and part-of-speech tagging using arbitrary units of input and apply it to the case of character-level Korean morphological analysis. Among models that do not employ prior linguistic knowledge, we achieve state-of-the-art word and sentence-level tagging accuracy with the Sejong Korean corpus using our proposed data-driven Bi-LSTM model.

Title and Abstract in Korean

한국어 형태소 분석 및 품사 부착에 효과적인 문자 단위 정보

한국어는 대표적인 교착어 중 하나이며, 문자 단위 정보의 중요도가 높다. 이에 따라 기존의 한국어 형태소 분석 연구는 자소 정보와 같은 문자 단위 자질을 사용하거나, 형태소 사전 및 변형 규칙과 같은 다량의 언어학적 지식을 활용하였다. 이러한 접근 방법은 사전을 사용하지 않은 문자 단위 형태소 분석은 변형 규칙의 복잡성으로 인해 불가능에 가깝다는 것을 전제로 하고 있다. 이러한 한계를 극복하고자, 본 연구에서는 어떠한 언어 단위도 입력으로 사용할 수 있으며 다단계 변형을 기반으로 형태소 분석 및 품사 부착을 수행하는 방법을 제안한다. 제안된 방법을 적용하여 구현된 데이터 기반 양방향 LSTM 모델의 성능을 세종 말뭉치를 이용하여 정량적으로 평가한 결과, 언어학적 지식을 활용하지 않은 접근 방법들 중 가장 높은 단어 및 문장 단위 부착 정확도를 보임을 확인하였다.

1 Introduction

Korean has traditionally posed a challenge for word segmentation and morphological analysis. In addition to virtually unbounded vocabulary sizes, out-of-vocabulary (OOV) rates for models can be high. Korean is an agglutinative, phonetic language with a SOV (Subject-Object-Verb) syntax and a flexible word order, although certain word orders are considered to be "canonical". Honorifics, conditionals, imperatives, and other forms are all signified using agglutinative endings which sometimes involve transformation of the stem to which they attach. Some endings can be further combined or fused to other endings in a defined order, and furthermore, morphological transformation rules also apply during this process. Transformation rules are mostly consistent at the grapheme level and can be represented by a handful of spelling rules, but many irregular forms do exist.

---

[*] Corresponding author

#	Morphemes
1	나(na)/VV + 는(neun)/ETM
2	날(nal)/VV + 는(neun)/ETM
3	나(na)/NP + 는(neun)/JX
4	나(na)/NNP + 는(neun)/JX
5	나(na)/VX + 는(neun)/ETM

Table 1: Ambiguous parses of Eojeol "na-neun"

#	Eojeol	Morphemes
(3)	나는 na-neun	나(na)/NP + 는(neun)/JX
	하늘에 ha-neul-e	하늘(ha-neul)/NNG + 에(e)/JKB
(2)	나는 na-neun	날(nal)/VV + 는(neun)/ETM
	새를 sae-leul	새(sae)/NNG + 를(leul)/JKO
	보았다 bo-ass-da.	보(bo)/VV + 았(ass)/EP + 다(da)/EF + ./SF

Table 2: Correct transformation and tag sequence for sample sentence containing ambiguous Eojeol "na-neun". # corresponds to correct parse sequence in Table 1, only labeled here for "na-neun".

In Unicode, Hangul (Korean alphabet) characters are allocated 11,140 codepoints. Each character contains an initial consonant, vowel, and final consonant represented in C+V+C, C+V, or V+C form. In Unicode, the form is always assumed to be C+V+C and the initial or final consonants are set to null according to the desired target form. Consonants and vowels are considered to be sub-character units called graphemes. Each character is represented using a combination of 19 initial consonants (including null), 21 vowels, and 27 final consonants (including null), and there is a mathematical formula that can be used to combine graphemes to generate the codepoint of a Hangul character. The character "김" (gim) can be represented in C+V+C form as follows.

Initial Consonant	Vowel	Final Consonant
ㄱ (g)	ㅣ(i)	ㅁ (m)

The Korean language has "fusion" spelling rules that apply across character boundaries (within an agglutinative unit), which implies that morphological transformation may occur among adjacent graphemes. When the final consonant of one character meets the initial consonant of the next character during verb inflection, there may be a change in the resulting combined character. This presents character-level embeddings with a unique challenge that is not present in most other languages.

In order to avoid confusion of terminology, we must define the precise meaning of morphological analysis in the context of Korean. For most languages, morphological analysis refers to a word-level tag that describes the aspect, tense, plurality, and other features of the word, whereas part-of-speech (POS) tagging serves to classify the word as a noun, verb, etc. The POS tag is sometimes concatenated to the morphological tag string as in the POSMORPH annotation employed by Heigold, et al (2016a).

In Korean, morphological analysis refers to the segmentation and restoration of morphemes within a "word" unit called an Eojeol and the POS tagging of each constituent morpheme. An Eojeol encodes not only lexical information but also grammatical information due to the agglutinative nature of the Korean language. The recovered morpheme segments often include a stem and other morphemes which indicate tense or other linguistic features. Traditional Korean morphological analysis algorithms operate at the Eojeol level and yield all ambiguous parses (Table 1) that lead to that particular Eojeol, including the morpheme transformations and tags. However, the model[1] proposed in this paper receives input at the sentence level and attempts to

---

[1]Model source code is made available at https://github.com/xtknight/rich-morphological-tagger

produce the one correct sequence of transformations and tags for all Eojeol within the sentence according to the context (Table 2).

2   Related Work

Morphological analysis of the Korean language has traditionally been performed in several ways, including separation of Korean characters into graphemes by using linguistic knowledge, lattice tree lookup (Park et al., 2010), application of regular and irregular inflection rules (Kang and Kim, 1992), morphosyntactic rule sets, and by using a pre-computed dictionary (Shim and Yang, 2004). However, we investigate whether morphological analysis of Korean is feasible without the use of any of these techniques and without a dictionary by making the assumption that common transformations and their underlying grapheme modifications can be easily recognized and learned with a Bi-LSTM model.

Bi-LSTM-CRFs have been used for sequential tagging with BIO annotation (Sang and Veenstra, 1999). Huang, et al (2015) show their effectiveness for POS tagging, chunking, named entity recognition (NER). These models show state-of-the-art accuracy at several tasks.

Similar models have also been proposed in universal morphological analysis. Heigold, et al (2016b) show how a nested LSTM architecture can be applied to word-level morphological tagging for a wide variety of languages. At the lower level, an LSTM network is used for character-level embedding to reduce OOV errors. However, this work does not investigate how such a model would operate for the most widely used Korean Sejong Corpus.

Sub-character tagging has also been attempted. Dong, et al (2016) demonstrate how radical-level features incorporated at the character-level for named entity recognition achieve state-of-the-art accuracy for Chinese. The most convincing attempt to tag Korean at the morpheme level is by Choi, et al (2016) who achieve state-of-the-art (dictionary-less) performance by using a multi-stage Bi-LSTM-CRF model that involves the splitting of Korean character input into constituent graphemes. However, the implicit assumption that Korean characters must first be split into graphemes to achieve optimal performance for morphological analysis is not well supported, and we should consider the splitting of characters into graphemes to be employing linguistic knowledge specific to Korean.

In our paper, we seek to answer the question of whether Korean morphemes can be tagged without grapheme-level splitting, rules specific to the language, or a dictionary. Although we initially considered Bi-LSTM-CRF for our model architecture, we show that the performance benefit by adding CRF is minimal and practically unnecessary compared to a standard Bi-LSTM model. Furthermore, CRF adds training and inference computational complexity due to the Viterbi algorithm.

3   Lemma and Form Alignment

고(go)	통(tong)	스(seu)	런(reon)
B-KEEP	I-KEEP	B-KEEP	I-MOD-럽, B-MOD-ㄴ
고통(go-tong)		스럽(seu-reob)	ㄴ (n)

Figure 1: Gold morphological transformation actions given by alignment oracle, including the resulting morphemes after running the BIO actions (Sejong corpus)

고통(go-tong)	스럽(seu-reob)	ㄴ (n)
NNG	XSA	ETM

Figure 2: Gold tagging actions (Sejong corpus)

In the Sejong corpus, Eojeol are annotated with their corresponding POS-tagged morpheme constituents, exactly as shown in Table 2. As mentioned earlier, morpheme spelling transformations may occur, and therefore the morpheme constituents may have slightly different graphemes than what is present in the original Eojeol form. To generate our training data, we must align the Eojeol form and its constituent morphemes at the character level, as we forbid using linguistic knowledge such as sub-character elements (graphemes) in our model. Like most agglutinative

languages, the Eojeol form and lemmas (morphemic elements in the Sejong corpus) often share overlapping characters at the beginning or end, and we utilize this assumption in our algorithm.

Gold Action	Count
B-KEEP	23,725,534
I-KEEP	8,650,166
B-MOD-하(ha), B-MOD-ㄴ (n)	153,130
NOOP	131,016
B-MOD:하(ha), B-MOD-았(ass)	61,592
B-MOD:이(i), B-MOD-ㄴ (n)	58,093
B-MOD:하(ha), B-MOD-아(a)	57,515
I-MOD:하(ha), B-MOD-아(a)	48,987
B-MOD:되 (doe), B-MOD-ㄴ (n)	41,335
B-MOD:하(ha), B-MOD-ㄹ (l)	36,419

Table 3: Top 10 morphing actions

We present an action-based algorithm (called an "alignment oracle") to align two arbitrary strings. Our oracle attempts to generate a 1:1 character-level mapping between the morphological form and the lemmas of an Eojeol by searching for a prefix, suffix, and modified inner string portion. Three primary actions are defined: KEEP (no modification to character), NOOP (drop character), and MOD (modify character). In the case of Korean, morphological transformations happen at the end of a form, so there is rarely a common suffix unless no transformation occurs at all. These primary actions are then augmented with B- and I- actions to facilitate morpheme segmentation. It is important to note that our algorithm is not specific in any way to the Sejong corpus or Korean itself.

The full process is demonstrated in Figure 1 starting from the source form. The gold untagged segmented lemma form is shown in the bottom row, and the actions generated by the oracle to generate the lemma are given in the middle row. The first three characters (go-tong-seu) are preserved with KEEP actions and the last character is considered the "modified inner string". In this case, the number of actions (5) exceeds the number of full input characters (4), and therefore two actions are assigned to the last character which split the "reon" syllable into "reob" and "n". The output after morpheme segmentation can be seen in the bottom row. In Figure 2, these output morphemes are then placed through a standard sequential tagger to assign part-of-speech tags.
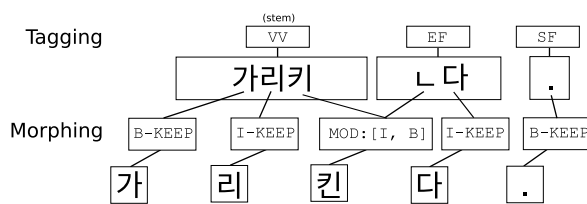

Figure 3: Actions for transformation with output segmentation

For Korean, the task is considerably more complicated. Rather than merely character-level transformation, new morpheme boundaries based on the results of those transformations are also required. A segmentation module adds B-/I- (beginning and inside) annotations to the KEEP and MOD actions. These actions allow morpheme segmentation to take place even amidst the modified character output sequence. This is detailed in Figure 1, where the final consonant sub-character unit ("n") of the last character of input ("reon") is transformed to "b" and the resulting fused full character is appended to the previous output morpheme, whereas the "n"
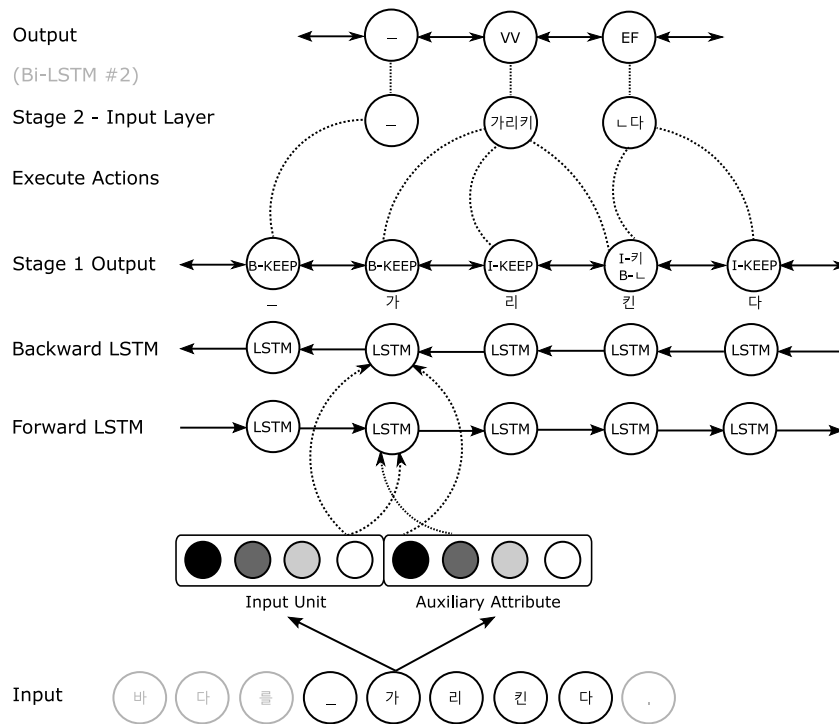
Figure 4: Partial example of two-stage tagging process for Korean phrase "to point out the sea"

sub-character unit becomes separated and represented as an entirely new morpheme itself. The top 10 resulting actions for the Sejong corpus on the form and lemma alignment stage are shown in Table 3.

4 The Model

Our model makes the assumption that in order to support morphological analysis for languages like Korean, two stages are required, which we call morphing and tagging. For tasks such as morphological transformation, word-level morphological analysis, or morpheme segmentation, only one stage is strictly necessary. To obtain tags for morphemes following morphological transformation, as in Korean, both stages are necessary, with tagging following morphing. The stages have no fundamental difference from each other: the second stage simply acts on the result of applying actions output by the first stage. Each stage outputs a single action for a single input unit. A single action could be as simple as a tag or as complicated as information resulting in advanced multi-character transformation along with the specification of the morphemic segments of those resulting characters. Model parameters are trained independently for each stage unless otherwise specified.

The model presented in this paper is inspired by word-level morphological analysis work by Heigold, et al (2016a) with the goal of allowing analysis at arbitrary units of input at each stage. Because we do not specify whether the input unit should be a word, morpheme, character, or even a unit at the sub-character level (such as graphemes in Korean or radicals in Chinese), we theoretically have the flexibility to tag a variety of languages at any level.

We employ a bidirectional Long Short-Term Memory (Hochreiter and Schmidhuber, 1997; Graves and Schmidhuber, 2005) network in our model and also experiment with optimization to a conditional random field (CRF) objective (Lafferty et al., 2001). In theory, CRF allows us to consider the likelihood of neighboring outputs and therefore jointly decode the highest probable chain of output labels for a given set of inputs. Although we posit that CRF is not strictly necessary, the overall architecture of our model is otherwise identical to a standard Bi-LSTM-CRF sequence tagging model (Huang et al., 2015) used for POS tagging and NER.

An overview of the architecture is shown in Figure 4. The input unit is embedded as a multi-dimensional vector. At the input level, an auxiliary attribute may be concatenated with the input unit to include auxiliary information for the unit, such as word break-level information, although empirical findings indicate our model performs best when using only the input unit. Nested embeddings as in (2016a) may also be appended at the input level.

All embeddings are concatenated to form a combined embedding which is then passed to the primary Bi-LSTM-CRF network and trained against a set of output actions. Whitespace delimiting Eojeols is represented as a reserved spacing token in the input unit.

Although each stage is independent and can accept an arbitrary unit of input suitable for any language, the following sections describe how this model pertains to our primary task of morpheme-level morphological analysis for the Korean language.

4.1 Morphing

The first stage of the model operates at the character level and is responsible for morphological transformation of the input form into the desired output lemma(s). During training, each input character is assigned one of three primary types by the alignment oracle as described in Section 3. Morpheme segmentation actions are also generated and augmented to the transformation actions at this stage for proper morpheme boundary identification. During inference, instead of using the alignment oracle, one action (including transformation and B-/I- tags) is predicted for each character based on trained parameters. At this point, tags are not yet assigned to each output morpheme.

4.2 Tagging

After the necessary morphological transformation and segmentation, tagging occurs at the morpheme level and acts on output produced by actions in the first stage of the model. An example of this is shown in Figure 2. In this stage, the action is simply to assign a POS tag to the morpheme, which is the input unit.

5 Experiments

5.1 Datasets

We conduct experiments using the full Sejong Korean Balanced Corpus dataset. The experiments are coded in Python using the TensorFlow library. The Sejong Corpus has been preprocessed to resolve punctuation inconsistencies and other surface-level errors. All datasets are converted at the sentence-level to a simple two-column format with each line containing an input unit and target action.

For all experiments, we follow an 85/10/5 cross-validation split for training, testing, and validation sets respectively. All data is randomly shuffled prior to splitting. Actions are inferred from the dataset by using lemma and form alignment. For evaluation, output from predicted actions in the first stage is used as input to the subsequent stage.

UniTagger represents the model proposed in this paper. The following number (for example, 500) represents the maximum action count for the morphing stage. The tagging stage only has as many actions as possible POS tags (45 in the Sejong corpus, including the reserved space token). Action pruning is performed at the training level, which removes from the training set the least common morphological transformation actions generated by the alignment oracle. For fair evaluation, actions are not removed from validation or test sets. All accuracy figures in this paper are reported based on a held-out test set.

5.2 Training

Optimization is performed using Adam (Kingma and Ba, 2014) with a learning rate of 0.001 and decay of 0.9. In the case of multi-stage models, model parameters are optimized independently for each stage. We use identical hyperparameters for all morphing and tagging models. Input

unit embedding size was set to 300 (for character and morpheme input). The final Bi-LSTM concatenating all embeddings before an optional CRF layer was used with an LSTM unit size of 300. Batch size was set to 64 for all experiments, except for the CRF experiment where it was set to 16. The maximum LSTM input length was set at a per-batch level which yielded optimal performance, and the maximum number of input units (whether characters or morphemes) was limited to 400 in both stages. A dropout of 10% was used for the reported model with best performance. Dropout is only applied at the unit embedding layer. Epoch count was set to 100 with early-stopping after 3 epochs with no improvement in validation set performance. Experiments were performed on GTX 1080 Ti 11GB GPUs. Average total training duration was around 5 hours for the entire Sejong dataset on a GTX 1080 Ti. In TensorFlow, the NVIDIA CuDNN-optimized LSTM was used (Appleyard et al., 2016).

5.3 Results

In Table 4, we show Eojeol-level morphological analysis accuracy for Korean. Note here that an Eojeol is considered correctly tagged only if all its constituent morphemes have been transformed, segmented, and tagged properly. Table 5 measures sentence-level tagging performance, which is the accuracy of all morphemes being transformed and tagged properly.

Model	Accuracy
Lee, et al. (2005)	92.96
Ahn, et al. (2007)	93.12
Lee, et al. (2009)	92.95
Choi, et al. (2016)	94.89
UniTagger-500	**96.20**

Table 4: End-to-end Eojeol-level accuracy for morphological analysis of Korean (Sejong Corpus)

Model	Model Type	Acc
Choi, et al. (2016)	Bi-LSTM-CRF	61.00
UniTagger-500	Bi-LSTM	**70.83**

Table 5: End-to-end sentence-level accuracy for morphological analysis of Korean (Sejong Corpus)

Form	Gold Action
났(nass)	B-나 + B-았
샀(sass)	B-사 + B-았
잤(jass)	B-자 + B-았
팠(pass)	B-파 + B-았
됐(daess)	B-되 + B-었
했(haess)	B-하 + B-았
녔(nyeoss)	B-니 + B-었
렸(ryeoss)	B-리 + B-었
셨(syeoss)	B-시 + B-었
졌(jieoss)	B-지 + B-었
겼(gyeoss)	B-기 + B-었
왔(oass)	B-오 + B-았
놨(noass)	B-놓 + B-았
췄(chuweoss)	B-추 + B-었

Table 6: Past tense morphing actions shown in embedding and gold actions from Sejong corpus

Model	Model Type	Acc
Choi, et al. (2016)	Bi-LSTM-CRF	67.25
UniTagger-500	Bi-LSTM	**79.49**

Table 7: Eojeol-level OOV accuracy

5.4 Analysis and Discussion

Our results show that our model can outperform previous state-of-the-art performance for Eojeol and sentence-level morphological analysis of Korean without linguistic knowledge.

When the dropout factor is adjusted, all metrics follow a similar trend as seen in Figure 5. Sentence accuracy is most sensitive to dropout factor adjustment. Best performance is achieved

with a dropout rate of 10%, and increasing the dropout rate further does not increase Eojeol-level OOV accuracy. This is a positive finding, as it indicates the model is not considerably overfitting to the training data beyond approximately the 10% level.

In Figure 6, a 300-dimensional unit embedding layer of the morphing stage is visualized using 2-component t-SNE. The corresponding gold actions are shown in Table 6, where all past tense morphemes end with final consonant "ㅆ" (ss). The model is able to infer that most of the forms shown in the graph represent the past tense and that they share a similar transformation pattern at the final consonant grapheme level. This shows that our model is able to correlate similar sub-character level morphological transformations even when operating at the character level. Furthermore, it is worth noting that Chinese characters still occur rarely in the Korean language in certain contexts. We can see Chinese characters grouped in a cluster, which shows that the model is able to distinguish one character-rich language (Korean) from another (Chinese). Other characters, such as punctuation, are also grouped by type in largely distinct clusters with occasional overlap.

Joint training of both stages was also attempted, though an initial investigation suggests that performance is not significantly different from training each stage's parameters independently.
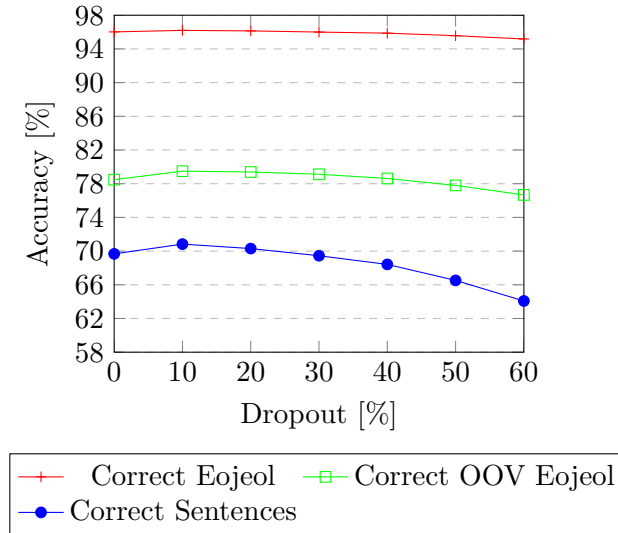

Figure 5: Impact of dropout on end-to-end tagging performance

The use of using an auxiliary binary break level attribute to represent whitespace was also investigated, but significantly higher accuracy was achieved by using a reserved spacing token instead. Despite the auxiliary break level attribute embedding, both stages of the model have a tendency to learn ambiguous morpheme transformations for adjacent Eojeol. In other words, even though the morpheme transformation and tags are correct, the Eojeol boundaries were incorrectly identified. With the reserved spacing token, this issue was extremely rare.

6 Conclusion and Future Work

In this work, we address the commonly held notion that Korean can not be tagged with competitive performance at the character level without prior linguistic knowledge. Our model architecture is not novel compared to previous work. The novelty of our morphological analyzer is its striking simplicity compared to previous approaches for character-rich languages such as Korean. The alignment oracle does not require any cost value for alignment operations such as in the Needleman–Wunsch algorithm (Needleman and Wunsch, 1970). The Bi-LSTM model is able to learn and utilize alignments that are purely arbitrary and apply them to unseen test data. Even when significantly limiting the number of actions in the training data, we show that by using the most common morphological transformation actions in an agglutinative language, we can
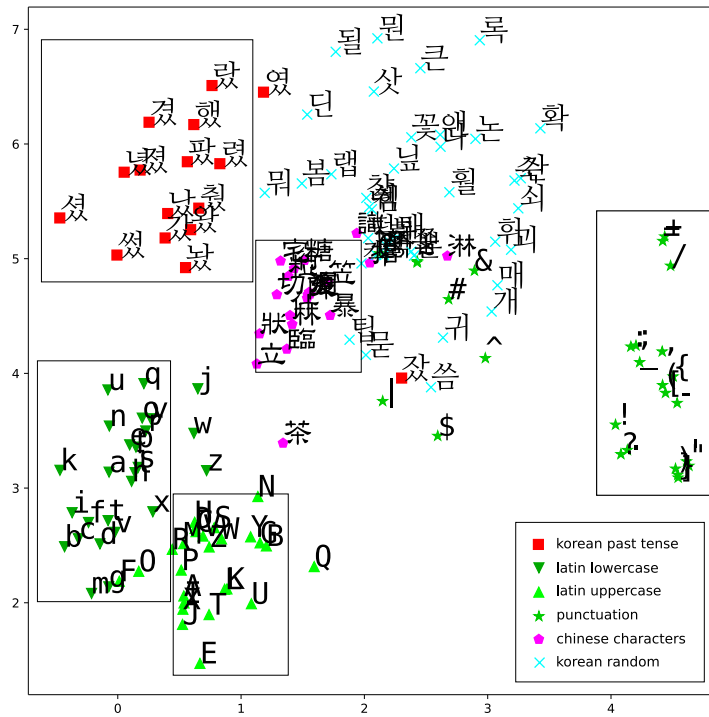
Figure 6: Deep embeddings of characters at the morphing stage (t-SNE)

exceed the performance of a model that uses linguistic knowledge such as sub-character features. We also show that the widely used CRF layer may in fact be unnecessary for high performance and add unnecessary computational complexity. This exceeded our own expectations and raises the possibility that a single architecture can handle tagging universally with only two simple Bi-LSTM stages. We contribute the necessary source code to replicate the experiments and to attempt alignment and training for any other language, assuming a corpus exists. Nevertheless, there are several points that future work should address.

Out-of-vocabulary morphemes generated by the morphing stage can also result in errors at the tagging stage, as the tagging stage was trained on the assumption of gold morphemes. We would like to experiment with including possible morpheme transformation errors at the tagging stage to determine if tagging performance can be improved.

We attempted joint training but found that end-to-end accuracy was marginally lower. We suspect this is because optimization of each individual stage is hindered by attempting to find optimal parameters for both stages. Future work should attempt joint training of an end-to-end model with the preinitialized parameters from optimizing each stage independently, which has been shown to be ideal in sequential models (Tang et al., 2016).

Lastly, although we are unaware of a language more character-rich and more morphologically complex than Korean, we would like to see our model applied to other morphologically complex languages to prove its universality. At the time of writing, we lacked sufficient baseline figures and methodology for generating training data for analyzing other languages at the morpheme level using the Universal Dependencies corpus, and the morphological tags were often conflated with part-of-speech tags. The baselines we found did not specify whether or not morpheme segmentation was taken into account. Without this information, it would be difficult to prove the performance of our model for other languages and we decided to leave training other languages as future work. That being said, our model does not employ any linguistic knowledge specific to Korean, and we therefore have no reason to believe it cannot be trained on any other arbitrary corpus with minor modifications at the preprocessing level.

Acknowledgements

References

Young-Min Ahn and Young-Hoon Seo. 2007. Korean part-of-speech tagging using disambiguation rules for ambiguous word and statistical information. *IEEE International Conference on Convergence Information Technology*, pages 1598–1601.

Jeremy Appleyard, Tomáš Kociský, and Phil Blunsom. 2016. Optimizing performance of recurrent neural networks on gpus. *arXiv*, (1604.01946).

Jihun Choi, Jonghem Youn, and Sang goo Lee. 2016. A grapheme-level approach for constructing a korean morphological analyzer without linguistic knowledge. *IEEE International Conference on Big Data*, pages 3872–3879.

Chuanhai Dong, Jiajun Zhang, Chengqing Zong, Masanori Hattori, and Hui Di. 2016. Character-based lstm-crf with radical-level features for chinese named entity recognition. *International Conference on Computer Processing of Oriental Languages*, pages 239–250.

A. Graves and J. Schmidhuber. 2005. Framewise phoneme classification with bidirectional lstm and other neural network architectures. *Neural Networks*, 18(5–6):602–610.

Georg Heigold, Guenter Neumann, and Josef van Genabith. 2016a. Neural morphological tagging from characters for morphologically rich languages. *arXiv*, (1606.06640).

Georg Heigold, Guenter Neumann, and Josef van Genabith. 2016b. Scaling character-based morphological tagging to fourteen languages. *IEEE International Conference on Big Data*.

Sepp Hochreiter and Jurgen Schmidhuber. 1997. Long short-term memory. *Neural Computation*, 9(8):1735–1780.

Zhiheng Huang, Wei Xu, and Kai Yu. 2015. Bidirectional lstm-crf models for sequence tagging. *arXiv*, (1508.01991).

Seungshik Kang and Yungtaek Kim. 1992. A computational analysis model of irregular verbs in korean morphological analyzer. *Journal of Korea Information Science Society*, 19(2):151–164.

Diederik Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv*, (1412.6980).

J. Lafferty, A. McCallum, and F. Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. *Proceedings of ICML*.

Do-Gil Lee and Hae-Chang Rim. 2005. Probabilistic models for korean morphological analysis. *Companion to the Proceedings of the International Joint Conference on Natural Language Processing*, pages 197–202.

Do-Gil Lee and Hae-Chang Rim. 2009. Probabilistic modeling of korean morphology. *IEEE Transactions on Audio, Speech, and Language Processing*, 17(5):945–955.

Saul B. Needleman and Christian D. Wunsch. 1970. A general method applicable to the search for similarities in the amino acid sequence of two proteins. *Journal of Molecular Biology*.

Sangwon Park, D Choi, E Kim, and K.-S Choi. 2010. A plug-in component-based korean morphological analyzer.

Tjong Kim Sang and Jorn Veenstra. 1999. Representing text chunks. *EACL '99 Proceedings of the ninth conference on European chapter of the Association for Computational Linguistics*, pages 173–179.

Gwang-Seob Shim and Jae-Hyung Yang. 2004. High speed korean morphological analysis based on adjacency condition check. *KIISE: Software and Applications*, 31(1):89–99.

Hao Tang, Weiran Wang, Kevin Gimpel, and Karen Livescu. 2016. End-to-end training approaches for discriminative segmental models. *IEEE Spoken Language Technology Workshop*.