Neural Morphological Segmentation Model for Mongolian

Weihua Wang[1,3], Rashel Fam[2], Feilong Bao[1,3,*] , Yves Lepage[2], Guanglai Gao[1,3]
1. *College of Computer Science, Inner Mongolia University,* Hohhot, China
2. *Graduate School of Information, Production and Systems, Waseda University* Kitakyshu, Japan
3. *Inner Mongolia Key Laboratory of Mongolian Information Processing Technology,* Hohhot, China
wangwh@imu.edu.cn, fam.rashel@fuji.waseda.jp

***Abstract*—Morphological segmentation is useful for processing Mongolian. In this paper, we manually build a morphological segmentation data set for Mongolian. We then present a character-based encoder-decoder model with attention mechanism to perform the morphological segmentation task. We further investigate the influence of analogy features extracted from scratch and improve the performance of our model using multi languages setting. Experimental results show that our encoder-decoder model with attention mechanism provides a strong baseline for Mongolian morphological segmentation. The analogy features provide useful information to the model and improve the performance of the system. The use of multi languages data set shows the capability of our model to acquire knowledge through different languages and delivers the best result.**

***Index Terms*—Morphological Segmentation, Mongolian, Encoder-Decoder model**

I. Introduction

Morphological segmentation is the task of dividing a given word into morphemes. Since the words of Mongolian have rich morphological structure, this task is considered as an important issue in many NLP applications, e.g., automatic speech recognition [1], machine translation [2], information retrieval [3] and named entity recognition [4], [5].

There are two kinds of segmentation tasks. The first one is *surface segmentation*. In this case, the task is to simply segment a word as a sequence of substrings: given a word, decide where to split into morphemes. For example, a Mongolian word written in classical Mongolian script, "ᠰᠢᠭᠦᠪᠡ" (sigube), means "judged", will be segmented as two morphemes, they are "sigu" and "be".

The second one is referred as *canonical segmentation* [6], it involves further processing to reconstruct the orthographic transformations. In classical Mongolian, edit operations on strings, like deletion, insertion and substitution, are necessary when performing the morphological segmentation. If we ignore the string operation, the morphemes are wrong. For instance, a Mongolian word "ᠪᠠᠲᠤᠯᠠᠭᠠᠨ ᠤ (batvlagan-v)" will be segmented as "ᠪᠠᠲᠤᠯᠠᠭ᠎ᠠ (batvlag_a)", "ᠨ (n)" and "ᠤ (-v)". It does not simply cut the words into morphemes, but also insert the character "_" to transform "ᠠ (a)" into the final form of "᠎ᠠ (_a)". Similarly in English, "inexhaustivity" will be segmented as "in", "exhaustive" and "ity". Figure 1 indicates the difference between the two kinds of segmentation.

word:	*inexhaustivity*
surface segmentation:	*in + exhaustiv + ity*
canonical segmentation:	*in + exhaustiv**e** + ity*

Fig. 1. Example of surface and canonical segmentation for English word "inexhaustivity".

To the best of our knowledge, there is no public data on both surface and canonical segmentation in Mongolian. Previous work in [4] only segments the non-breaking space suffixes for nouns. But we also include the suffixes for verbs. Therefore, we address the issue of surface segmentation and canonical segmentation with a unified model for both Mongolian nouns and verbs.

Inspired by the recent success of encoder-decoder model with attention mechanism in neural machine translation [7], [8], we design a character-based encoder and decoder model for Mongolian morphological segmentation. The model encodes every character using bidirectional long-short term memory network (LSTM) [9] to form the context vector and decodes with attention mechanism to generate the characters at each time step. More importantly, we study the capability of our model to share the character level features learned from different languages in a multi-source setting. We also investigate the effect of using word form features. These additional information would benefit the segmentation task performance, especially for the low resource languages.

The main contributions of this work are as follows:

- Build a data set for Mongolian morphological segmentation;
- Provide a baseline system for Mongolian morphological segmentation based on encoder-decoder model with attention mechanism;
- Augment the model with analogy features and four languages multi-source input.

The paper is organized as follows. Section II presents some related work. Section III introduces the issues in morphological segmentation for Mongolian. Section IV explains our model to perform morphological segmentation. Section V presents the experiments and analyses the results obtained. Section VI gives the conclusion of our work.

*Corresponding author.

II. Related work

There are several approaches to address morphological segmentation. The first one is the unsupervised approach. It aims to extract the morphemes directly from a list of unlabelled data. [10] proposed a method based on Minimum Description Length (MDL) to extract the morphemes. The method do not need annotated data to train the model.

The second one is the semi-supervised approach. It utilizes both labeled and unlabeled data, like [11], [12]. The third one is the supervised approach. [13] treated the morphological segmentation task as a sequence labeling problem of assigning a predefined label to each character. They showed that high performance can be achieved with the use of hand-crafted features and classifier, like Conditional Random Field (CRF) [14].

Recently, recurrent neural networks have been used on character level problem, like language model [15], morphological tagging [16]. A windowed LSTM approach was presented to learn extra information about the context input words to segment Hebrew and Arabic words [17]. But this LSTM approach can not address the problems in canonical segmentation.

III. Morphological Segmentation in Mongolian

As an agglutinative language, Mongolian has a complex morphological structure. The words are constructed by successively concatenating suffixes to the root. These suffixes can be categorized into two groups: derivational suffixes and inflectional suffixes. Derivational suffixes, or word-building suffixes, are concatenated to the end of the root. They change the meaning of the root. After one or more derivational suffixes are concatenated to a word, the word will become to a stem. On the other hand, inflectional suffixes, or word-changing suffixes, are concatenated to the end of the stem. They can be categorized into two groups based on the part-of-speech of the word, nouns and verbs:

- **nominal suffixes**: plural, reflexive and case suffixes;
- **verbal suffixes**: voice, aspect and mood suffixes.

Figure 2 shows the order of suffixes inside a word in Mongolian. There is no prefix in Mongolian. A noun word must have a case suffix, in the meantime, a verb must have a mood suffixes. The case suffixes and the mood suffixes are at the end of the word.

To the best of our knowledge, There is still no freely available data for morphological segmentation in Mongolian. To remedy to that, we prepare a data set for Mongolian morphological segmentation.

We crawled around 5,000 sentences from Mongolian news websites. We sampled 10,000 words using uniform distribution. All kinds of suffixes are include in the data.

IV. Proposed Model

In this section, we present the framework used in our experiments. It is an encoder and decoder model with attention mechanism [8].

Stem (noun)	+	[Plural suffix]	+	[Reflexive suffix]	+	Case suffix
Stem (verb)	+	[Voice suffix]	+	[Aspect suffix]	+	Mood suffix

Fig. 2. Inflectional suffixes order inside a word for nouns and verbs in Mongolian. The "[]" means optional suffix: 0-1 occurrence.

A. Encoder-decoder model with attention mechanism

A sequence of characters $\boldsymbol{x} = x_1, \ldots, x_m$ is read by a bidirectional recurrent encoder. The encoder outputs a sequence of hidden states $\boldsymbol{h}_i$, where the h_i is the concatenation of forward $\overrightarrow{h_i}$ and backward $\overleftarrow{h_i}$ as shown in Formula 1.

$$\begin{aligned} \overrightarrow{h_i} &= f(\overrightarrow{h}_{i-1}, s) \\ \overleftarrow{h_i} &= f(\overleftarrow{h}_{i-1}, s) \\ s &= E_s x_i \end{aligned} \tag{1}$$

E_s is the character embedding matrix, which is shared across different languages in the multi-source setting. Function f is the function to compute the current hidden state based on the previous one.

Similar to the encoder, the decoder is also a recurrent network however it is a uni-directional. Formula 2 shows how to compute a new hidden state $\boldsymbol{z}_j$ in the decoder from the encoder output $\boldsymbol{h}_i$. It computes the new hidden state using a $tanh$ activation function g based on previous hidden state $\boldsymbol{z}_{j-1}$, an embedding $\boldsymbol{e}_{j-1}$ and a conditional input vector $\boldsymbol{c}_j$ derived from the encoder output $\boldsymbol{h}_i$. We uses LSTMs [9] for all recurrent cells.

$$\boldsymbol{z}_j = g(\boldsymbol{z}_{j-1}, \boldsymbol{e}_{j-1}, \boldsymbol{c}_j) \tag{2}$$

The conditional input vector $\boldsymbol{c}_j$ is a weighted sum of the attention score a_i and the encoder output $\boldsymbol{h}_i$. There exist several ways to calculate the attention scores. Here we describe the method proposed in [8]. It is a feed-forward neural network architecture with learnable layer which consists of $\boldsymbol{v}_a$, $\boldsymbol{W}_a$ and $\boldsymbol{U}_a$. They are shown in Formula 3.

$$\begin{aligned} score(z_{j-1}, h_i) &= \boldsymbol{v}_a^T \tanh(\boldsymbol{W}_a z_{j-1} + \boldsymbol{U}_a h_i) \\ a_{ij} &= \frac{exp(score(z_{j-1}, h_i))}{\sum_{j-1}^{m} exp(score(z_{j-1}, \boldsymbol{h}_i))} \\ c_j &= \sum_{j}^{m} a_{ij} \boldsymbol{h}_i \end{aligned} \tag{3}$$

The last step is the generator. It outputs the target sequence characters $\boldsymbol{y} = y_1, \ldots, y_n$ by transforming the LSTM output $\boldsymbol{z}_j$ via a linear layer.

$$p(y_j | y_1, \ldots, y_n, \boldsymbol{x}) = softmax(W_o z_j + b_o) \tag{4}$$

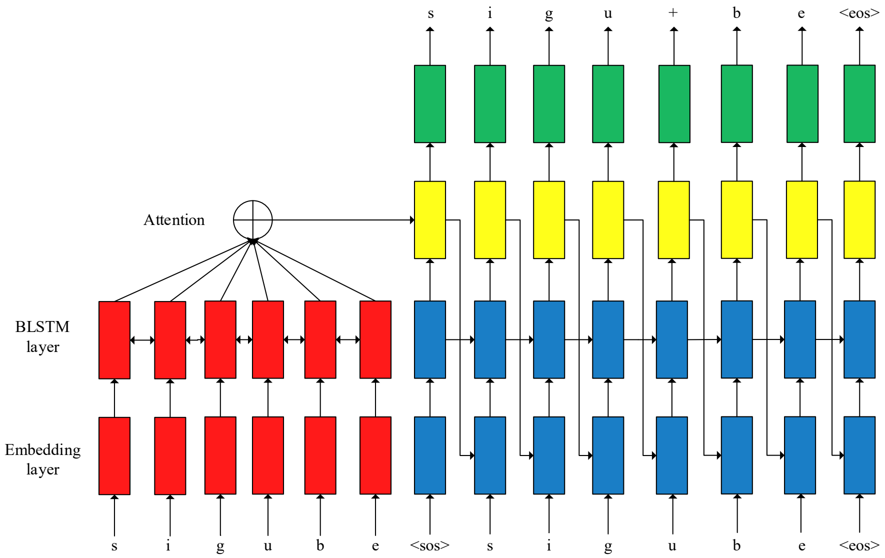

Fig. 3. The architecture of the encoder-decoder model with attention mechanism. A Mongolian word "ᠰᠢᠭᠦᠪᠡ" (sigube) illustrates in this figure, it means "judged". Dash lines between the embedding and BiLSTM layer mean that there is a dropout between the link. A special character "+" denotes the segmentation boundaries.

B. Multi-source model

The multi-source model is a way to train a single model for several languages at once. [18] showed that training several languages at the same time allows the model to learn fewer parameters in the training step. Bear this in mind, we design our model to use multiple codes of languages. The language code of an input string is added to its beginning and end. For example, the Mongolian word "ᠰᠢᠭᠦᠪᠡ" (sigube) will be input as "<mn> s i g u b e <mn>".

C. Analogical grids model

In this model, we add the identifier of the analogical grids, which contain the respective word form, in front of the input string of the system. We build analogical grids from the set of words and its segmented form contained in the Mongolian data set. The analogical grids are filtered using several saturation threshold to study the influence of the saturation of the grid in order to help the system to solve the task. If the word is not existed in any of analogical grid, we just leave the input string as it is.

An analogical grid is a matrix of words. Figure 4 displays an example of analogical grids in both English and Mongolian. It is automatically extracted from a set of words. Any four words taken from any two rows and two columns is a proportional analogy. Previous works, like [19]–[22], use it to study the productivity of a given language because it captures the organization of the lexicon in a language up to a certain extent. Each word form is represented as a feature vector and then grouped as a list of pairs of words by equal ratio before constructed into analogical grids [23]. Formula 5 shows the definition of analogical grids.

$$\begin{array}{c} G_1^1 : G_1^2 : \cdots : G_1^m \\ G_2^1 : G_2^2 : \cdots : G_2^m \\ \vdots \quad \vdots \qquad\quad \vdots \\ G_n^1 : G_n^2 : \cdots : G_n^m \end{array} \quad \overset{\Delta}{\iff} \quad \begin{array}{c} \forall (i,k) \in \{1,\ldots,n\}^2, \\ \forall (j,l) \in \{1,\ldots,m\}^2, \\ G_i^j : G_i^l :: G_k^j : G_k^l \end{array} \tag{5}$$

We can characterize analogical grid is by its size and saturation. Size is simply the total number of cells inside an analogical grid. It is calculated by simply multiplying the number of lines and the number of columns (see Formula 6). Thus, the analogical grids in Figure 4 have the size of $5 \times 4 = 20$ (*English*) and $5 \times 4 = 20$ (*Mongolian*).

$$\text{Size} = \text{Number of lines} \times \text{Number of rows} \tag{6}$$

On the other hand, saturation is the ratio between the number of non-empty cells and the total number of cells (size) of an analogical grid. Using Formula 7, we got saturation of $(16/20) \times 100\% = 80\%$ (*left*) and $(16/20) \times 100\% = 80\%$ (*right*) for analogical grids in the Figure 4.

$$\text{Saturation} = \frac{\text{Number of non-empty cells}}{\text{Total number of cells}} \times 100\% \tag{7}$$

V. Experiments

In our experiments, we first compare the performance of our model with two baselines system in three languages: English, German and Indonesian, to verify the effectiveness of our model. The same model will be used to perform the segmentation task in Mongolian. We expect some improvements from our model when using a multi-source setting, where we train on these four languages to perform segmentation on

show	:	*show**s***	:	*show**ing***	:	*show**ed***
walk	:	*walk**s***	:	*walk**ing***	:	*walk**ed***
open	:	*open**s***	:	*open**ing***	:	
study	:		:	*study**ing***	:	
read	:	*read**s***	:	*read**ing***	:	

(yabv)	:	yabv**hv**	:	yabv**n_a**	:	yabv**jai**
(qNsi)	:	qNsi**hv**	:	qNsi**n_a**	:	qNsi**jai**
(vnta)	:	vnta**hv**	:	vnta**n_a**	:	vnta**jai**
(sal)	:	sal**hv**	:		:	sal**jai**
(han)	:		:		:	han**jai**

Fig. 4. Analogical grids in English (*top*) and Mongolian (*bottom*). In the Mongolian example, the second column is present tense, the third column is future tense; the fourth column is past tense.

TABLE I
STATISTICS OF THE TRAINING DATA FOR ALL OF THE LANGUAGES. SEE FIGURE 1 FOR THE DIFFERENCE BETWEEN CONCATENATED AND NON-CONCATENATED SEGMENTATION.

Languages	Average length	Segmentation type (%)		# Unique morphemes
		non-concatenated	concatenated	
English	8.20	22.88	77.12	6799
German	12.49	53.69	46.31	6983
Indonesian	8.65	23.60	76.40	3359
Mongolian	10.74	18.00	82.00	3987

Mongolian data set. Finally, we evaluate the influence of using analogical grid identifier and attention mechanism.

A. Data used

We used the data created by [6]. It consists of three languages with different language families: English, German and Indonesian. The English data was extracted from segmentation derived from CELEX [24]. The German data was generated by rules from DerivBase [25]. The Indonesian data was created from the output of a rule-based morphological analyser for Indonesian, MorphInd [26]. For each of the language, the data is already in the format of 10-fold cross-validation experiment. For each fold, we will have 8,000 words as the training set; 1,000 words as the development set; and another 1,000 words as the test set.

For Mongolian, we use the data that described in Section III. We also prepare the data for 10-fold cross-validation experiment, in a similar way to the other three languages.

The statistics of the training data for each language are shown in Table I. Table I exhibits that the average length of German words is the longest among these four languages, followed by Mongolian. English has the shortest words length in average. Additionally, German data has the largest proportion of non-concatenated segmentation. On the contrary, concatenated segmentation is dominant in Mongolian. The average numbers of unique morphemes in German and English are around two times higher than for Mongolian and Indonesian.

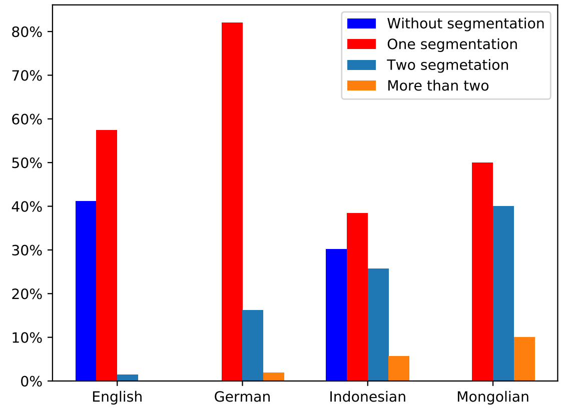

Fig. 5. Average percentage of number of segmentation points to perform inside a word.

Figure 5 presents the statistics on the number of segmentation points to perform inside a word. We can observe that, for both German and Mongolian, all of the words in the training set need to be segmented. For English and Indonesian, about 40 % and 30 % words do not need any segmentation. 80 % of the words in German need to be segmented one time, whereas it is more than 40 % in English and Mongolian. It is obvious that Mongolian has the largest proportion of performing two or more segmentation points inside a word. In contrast, English and German have a very small percentage of number of segmentations with more than one.

B. Evaluation metrics

We have four metrics to evaluate the performance of the systems. When compare with the work in [6], we use the same evaluation metrics. We also use another metric for Mongolian. They are as follows:

- **Error rate**: one minus the proportion of guesses that are completely correct.
- **Edit distance**: average Levenshtein distance [27] between all guesses and their corresponding gold reference.
- **Morphemes** F_1: compares the morphemes in guess and gold reference. Precision is the proportion of morphemes in guess that occur in gold. Recall is the proportion of morphemes in gold reference that occur in guess. F_1 is the harmonic mean of precision and recall.
- **Accuracy**: the percentage of guesses that are completely correct.

C. Experiment protocol

We perform a 10-fold cross validation experiments for each of the four languages. We trained our model using *adadelta* [28] as optimizer under 20 epochs. The model has one layer to encode and another layer to decode. The number of hidden states in the encoder and decoder are 128. We also use a dropout layer after learning the character representation with a dropout rate of 0.2. The model is adapted from OpenNMT [29], which is an open-source framework for neural machine translation based on pytorch. The model is then tested against the test set and evaluated on the three metrics.

TABLE II
English performances with our model and multi-source setting.

Models	Error rate	Edit distance	F_1
Joint model	0.27(±0.02)	0.98(±0.34)	0.76(±0.02)
ED model	0.25(±0.01)	0.47(±0.02)	0.78(±0.01)
Our base model	0.22(±0.01)	0.41(±0.03)	**0.79**(±0.03)
Multi-source model	**0.21**(±.01)	**0.40** (±0.00)	0.78(±0.01)

TABLE III
German performance with our model and multi-source setting.

Models	Error rate	Edit distance	F_1
Joint model	0.41(±0.03)	1.01(±0.07)	0.76(±0.02)
ED model	0.26(±0.02)	0.51(±0.03)	**0.86**(±0.01)
Our base model	**0.23**(±0.02)	0.45(±0.04)	0.79(±0.02)
Multi-source model	0.23(±0.02)	**0.40** (±0.02)	0.78(±0.01)

TABLE IV
Indonesian performance with our model and multi-source setting.

Models	Error rate	Edit distance	F_1
Joint model	0.10(±0.01)	0.15(±0.02)	0.93(±0.01)
ED model	0.09(±0.01)	0.12(±0.01)	**0.93**(±0.01)
Our base model	0.08(±0.01)	0.11(±0.02)	0.92(±0.01)
Multi-source model	**0.07**(±0.00)	**0.10**(±0.01)	0.92(±0.01)

D. Baselines

We take two different systems as our baselines in this experiments to compare the performance of our model. The first one is based on a log-linear probability model and the second one is based on a neural network model.

Joint model: a feature-rich model that jointly learned to perform segmentation and reconstruct the orthographic transformation. It employs an importance sampling algorithm to perform inference in the model [6]. This model achieved the state of the art performance among non-neural network model.

Encoder-Decoder (ED) model: a character-based encoder-decoder model explained in [30]. They used gated RNN (GRU) [31] to encode one-hot character representation into a fixed representation. The decoder map the representation into a variable length sequence. The model can be extended to use a Re-Ranker model to further boost the performance. However, since our model has no separate steps for re-ranking, we only choose the model without Re-Ranker for a fair comparison.

E. Experiment results

First, we compare our encoder-decoder model with three different languages. Table II, III and IV show the experiment results for each language respectively. Our model outperforms the two baseline systems in terms of error rate and edit distance in all the three languages. Although, we get a better morpheme F_1 score in English, it is not the case for German and Indonesian. The highest error rate is observed for German. This might be caused by the complexity of orthographic transformation in this language. We get the best performance in Indonesian. In addition, our models are also stable in all 10 folds.

TABLE V
Mongolian performance with different models

Models	Accuracy (%)	Edit distance	F_1 (%)
Our base	90.49 (±1.04)	0.14(±0.02)	91.48 (±1.03)
Multi-source	**91.31** (±0.93)	**0.13** (±0.02)	**92.03** (±0.92)
Analogical grids	90.55 (±0.83)	0.15 (±0.02)	91.06 (±0.86)

TABLE VI
Results of Mongolian character biLSTM model with different attention mechanisms

Attention	Accuracy (%)	Edit distance	F_1 (%)
Gen	88.24 (±1.21)	0.17 (±0.03)	89.62 (±1.26)
Gen + feed	89.71 (±1.55)	0.15 (±0.03)	90.66 (±1.34)
MLP	88.11 (±1.30)	0.18 (±0.01)	90.14 (±1.51)
MLP + feed	90.46 (±0.98)	0.15 (±0.02)	91.46 (±0.76)
Dot	89.21 (±1.13)	0.16 (±0.02)	90.38 (±1.00)
Dot + feed	**90.49** (±1.04)	**0.14** (±0.02)	**91.48** (±1.03)

Table V shows the results obtained in Mongolian. Despite the fact that Mongolian words tend to have more segmentation points to perform inside a given word in comparison to the other languages, the results are very promising. One can think that it is caused by having less canonical segmentation in Mongolian. Another reason is that Mongolian has a lower number of morphemes. This means a lower number of suffixes, which are more frequent in comparison to than other languages.

The use of multi-source languages input setting improves the performance of the system and outperforms base model in Mongolian. Similar behaviour can be observed in English and Indonesian, but not in German. By analysing the stem errors in Mongolian results, we think that the use of multi-source input setting provides more knowledge when learning the context of Mongolian characters. Here, we can see that our model is able to capture common features among different languages although they are not very close to each other.

Table VI shows the experiment results with different kind of attention mechanisms:

- MLP attention [8]
- General (Gen) and Dot attention in [7].

We found that the **Dot + feed** attention performs slightly better in terms of error rate and edit distance. It is a similar observation on neural machine translation described in [32]. Due to the number of parameters are less, the Dot attention mechanism is faster than other two attention mechanisms. The use of input-feeding layer described in [7] gives a small improvement for all type of attention mechanisms. This is because that in this field the alignment is simpler than machine translation.

We also evaluate our Mongolian grid augmented model using different saturation threshold($\geq 10\%$, $\geq 50\%$, and $\geq 90\%$) to study the influence of saturation of analogical grids. Results are shown in Table VII. When the saturation threshold increases, we filtered out more grids. Thus, we have less words inside the analogical grids. We compare the performance with cluster identifier learned using K-means provided by word2vec module [33] . We set the parameter to extract 200 clusters with vector dimension is 300. Using only the highest analogical

TABLE VII
MONGOLIAN PERFORMANCE WITH DIFFERENT GRID IDS

Model	Sat.	Acc.	Edit distance	F_1
Word2Vec	–	89.95 (±1.22)	0.16 (±0.04)	91.00 (±1.21)
Grid (all IDs)	≥10	89.50 (±1.39)	0.16 (±0.03)	90.67 (±1.39)
	≥50	89.63 (±0.78)	0.16 (±0.02)	90.73 (±0.87)
	≥90	86.29 (±4.11)	0.22 (±0.08)	87.33 (±4.04)
Grid (highest ID)	≥10	89.95 (±0.68)	0.15 (±0.01)	91.01 (±0.85)
	≥50	90.30 (±1.17)	0.15 (±0.02)	91.03 (±1.23)
	≥90	**90.55**(±0.83)	**0.15**(±0.02)	**91.06**(±0.86)

grid ID, the system outperform the performance of cluster by word2vec. The reason is that analogical grid learn more about the orthographic information than word2vec. Analogical grids with higher identifier (ID) usually have higher saturation. However, using all of the analogical grid IDs leads to lower results.

F. Error analyses

By analyzing the results in Mongolian, we observed that our model can predict the right suffix even when it fails at reconstructing the orthographic transformations. For example, the correct answer for " ᠬᠠᠭᠠᠳ ᠤᠨ (hagad-vn)" is: "hag_a+d+vn". Our model is able to distinguish the suffix "-vn" although it produces the wrong solution, "hagad+vn". The same phenomenon also exists in the other languages. In English, the correct segmentation for the word "runner" is: "run+er". However, our model outputs the guess of "runn+er". It successes to segment on the right position but fails to delete "n" at the end of the first morpheme. Another example is in German for the word "verschwendung", which is supposed to be segmented into "ver+schwende+ung". In this case, our model is also able to correctly segment the word into "ver+schwend+ung", but it is unable to insert the character "e" at the end of the second morpheme.

Looking further at the output of Mongolian system, our model can even predict stems better than the gold standard. For example, the gold standard for " ᠤᠯᠤᠯᠬᠠᠳᠠᠭ (wlwlqadag)" is "wlwlqa+dag". The guess given by our model is "wlw+lqa+dag". We can see that our model is capable to recognize the two stems of "wlw" and "lqa". This is possibly a better answer. In English, the gold standard for "placed" is "placed", our model could predict the right answer "place+ed". In Indonesian, the gold standard is: terjemah+an is better guessed as "ter+jemah+an" by our model.

VI. CONCLUSION

We built a data set for Mongolian morphological segmentation. From that data, we found that Mongolian tends to have a larger number of segmentation points inside one word in comparison to the three other languages: English, German and Indonesian.

We adopted the encoder-decoder model with attention mechanism to perform morphological segmentation in Mongolian. We performed experiments in four different languages. Experimental results show that we achieved a better performance in all the three languages. Our model can be considered as a strong baseline for Mongolian with the accuracy of 91.31% under the multi-source setting.

We concluded that the use of multi-source setting can give an additional boost to the model. With multi-source setting, our model could learn knowledge across languages although they are not very close to each other. Adding analogical grid identifier into the input string provide the orthographic features about the surface form of the words. Integrating the use of analogical grid identifier with multi-source setting may further improve the system.

ACKNOWLEDGE

This work was funded in part by National Natural Science Foundation of China (No.61773224 and No.61563040), Natural Science Foundation of Inner Mongolia (No. 2016ZD06). We would also like to thank the anonymous reviewers for their useful comments.

REFERENCES

[1] F. Bao, G. Gao, X. Yan, and W. Wang, "Segmentation-based mongolian LVCSR approach," in *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2013, pp. 8136–8139.

[2] X. Su, G. Gao, Y. Jiang, J. Wu, and F. Bao, "Mongolian inflection suffix processing in NLP: A case study," in *Proceedings of the Sixth Conference on Natural Language Processing and Chinese Computing (NLPCC 2017)*. Springer, 2015, pp. 347–352.

[3] G. Gao, W. Jin, F. Long, and H. Hou, "A first investigation on mongolian information retrieval." in *EVIA@ NTCIR*, 2008.

[4] W. Wang, F. Bao, and G. Gao, "Mongolian named entity recognition system with rich features," in *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*. Osaka, Japan: The COLING 2016 Organizing Committee, December 2016, pp. 505–512. [Online]. Available: http://www.aclweb.org/anthology/C16-1049

[5] ——, "Mongolian named entity recognition with bidirectional recurrent neural networks," in *Proceedings of 28th IEEE International Conference on Tools with Artificial Intelligence (ICTAI)*. IEEE, 2016, pp. 495–500.

[6] R. Cotterell, T. Vieira, and H. Schütze, "A joint model of orthography and morphological segmentation," in *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Association for Computational Linguistics, 2016, pp. 664–669. [Online]. Available: http://www.aclweb.org/anthology/N16-1080

[7] T. Luong, H. Pham, and C. D. Manning, "Effective approaches to attention-based neural machine translation," in *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 2015, pp. 1412–1421. [Online]. Available: http://www.aclweb.org/anthology/D15-1166

[8] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," *arXiv preprint arXiv:1409.0473*, 2014.

[9] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997. [Online]. Available: http://dx.doi.org/10.1162/neco.1997.9.8.1735

[10] M. Creutz and K. Lagus, "Unsupervised discovery of morphemes," in *Proceedings of the ACL-02 workshop on Morphological and phonological learning*, vol. 6. Association for Computational Linguistics, 2002, pp. 21–30.

[11] T. Ruokolainen, O. Kohonen, S. Virpioja, and M. Kurimo, "Painless semi-supervised morphological segmentation using conditional random fields," in *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics (Volume 2: Short Papers)*. Gothenburg, Sweden: Association for Computational Linguistics, April 2014, pp. 84–89. [Online]. Available: http://www.aclweb.org/anthology/E14-4017

[12] S.-A. Grönroos, S. Virpioja, P. Smit, and M. Kurimo, "Morfessor flatcat: An hmm-based method for unsupervised and semi-supervised learning of morphology," in *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*. Dublin, Ireland: Dublin City University and Association for Computational Linguistics, August 2014, pp. 1177–1185. [Online]. Available: http://www.aclweb.org/anthology/C14-1111

[13] T. Ruokolainen, O. Kohonen, K. Sirts, S.-A. Grönroos, M. Kurimo, and S. Virpioja, "A comparative study of minimally supervised morphological segmentation," *Computational Linguistics*, 2016.

[14] J. Lafferty, A. McCallum, and F. C. Pereira, "Conditional random fields: Probabilistic models for segmenting and labeling sequence data," 2001.

[15] Y. Kim, Y. Jernite, D. Sontag, and A. M. Rush, "Character-aware neural language models." in *Proceedings of the 13rd AAAI Conference on Artificial Intelligence*, 2016, pp. 2741–2749.

[16] R. Cotterell and G. Heigold, "Cross-lingual character-level neural morphological tagging," in *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 2017, pp. 748–759. [Online]. Available: http://aclweb.org/anthology/D17-1078

[17] L. Wang, Z. Cao, Y. Xia, and G. de Melo, "Morphological segmentation with window LSTM neural networks." in *Proceedings of the 13rd AAAI Conference on Artificial Intelligence*, 2016, pp. 2842–2848.

[18] T. Ha, J. Niehues, and A. H. Waibel, "Toward multilingual neural machine translation with universal encoder and decoder," *CoRR*, vol. abs/1611.04798, 2016. [Online]. Available: http://arxiv.org/abs/1611.04798

[19] R. Singh and A. Ford, "In praise of Sakatayana: some remarks on whole word morphology," in *The Yearbook of South Asian Languages and Linguistics-200*, R. Singh, Ed. Thousand Oaks: Sage, 2000.

[20] S. Neuvel and S. A. Fulop, "Unsupervised learning of morphology without morphemes," in *Proceedings of the ACL-02 Workshop on Morphological and Phonological Learning*, July 2002, pp. 31–40. [Online]. Available: http://www.aclweb.org/anthology/W02-0604

[21] N. Hathout, "Acquistion of the morphological structure of the lexicon based on lexical similarity and formal analogy," in *Proceedings of the 3rd Textgraphs workshop on Graph-based Algorithms for Natural Language Processing*, Manchester, UK, August 2008, pp. 1–8. [Online]. Available: http://www.aclweb.org/anthology/W08-2001

[22] R. Fam and Y. Lepage, "A study of the saturation of analogical grids agnostically extracted from texts," in *Proceedings of the Computational Analogy Workshop at the 25th International Conference on Case-Based Reasoning (ICCBR-CA-17)*, Trondheim, Norway, 2017, pp. 11–20.

[23] ——, "Morphological predictability of unseen words using computational analogy," in *Proceedings of the Computational Analogy Workshop at the 24th International Conference on Case-Based Reasoning (ICCBR-CA-16)*, Atlanta, Georgia, 2016, pp. 51–60.

[24] R. H. Baayen, R. Piepenbrock, and R. van H, "The CELEX lexical data base on CD-ROM," 1993.

[25] B. Zeller, J. Šnajder, and S. Padó, "Derivbase: Inducing and evaluating a derivational morphology resource for german," in *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2013, pp. 1201–1211.

[26] S. D. Larasati, V. Kuboň, and D. Zeman, "Indonesian morphology tool (morphind): Towards an indonesian corpus," in *International Workshop on Systems and Frameworks for Computational Morphology*. Springer, 2011, pp. 119–129.

[27] V. Levenshtein, "Binary codes capable of correcting deletions, insertions and reversals," *Soviet Physics-doklady*, vol. 10, no. 8, pp. 707–710, Feb. 1966.

[28] M. D. Zeiler, "Adadelta: an adaptive learning rate method," *arXiv preprint arXiv:1212.5701*, 2012.

[29] G. Klein, Y. Kim, Y. Deng, J. Senellart, and A. Rush, "Opennmt: Open-source toolkit for neural machine translation," in *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 4: System Demonstrations)*. Vancouver, Canada: Association for Computational Linguistics, July 2017, pp. 67–72. [Online]. Available: http://aclweb.org/anthology/P17-4012

[30] K. Kann, R. Cotterell, and H. Schütze, "Neural morphological analysis: Encoding-decoding canonical segments," in *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 2016, pp. 961–967. [Online]. Available: http://www.aclweb.org/anthology/D16-1097

[31] K. Cho, B. van Merrienboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, "Learning phrase representations using rnn encoder–decoder for statistical machine translation," in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 2014, pp. 1724–1734. [Online]. Available: http://www.aclweb.org/anthology/D14-1179

[32] J. Gehring, M. Auli, D. Grangier, and Y. Dauphin, "A convolutional encoder model for neural machine translation," in *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics, 2017, pp. 123–135. [Online]. Available: http://www.aclweb.org/anthology/P17-1012

[33] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," *arXiv preprint arXiv:1301.3781*, 2013.