IPS-WASEDA system at CoNLL–SIGMORPHON 2018 Shared Task on morphological inflection

Rashel Fam **Yves Lepage**
Graduate School of Information, Production, and Systems
Waseda University
2-7 Hibikino, Wakamatsu-ku, Kitakyushu-shi, 808-0135 Fukuoka-ken, Japan
{fam.rashel@fuji., yves.lepage@}waseda.jp

Abstract

This paper presents the system submitted by IPS-WASEDA University for CoNLL–SIGMORPHON 2018 Shared Task 1: Type level inflection. We develop a system based on a holistic approach which considers whole-word form as a unit, instead of breaking them into smaller pieces (e,g. morphemes) like the baseline systems does. We also implement an encoder-decoder model which has recently become the new standard in many natural language processing (NLP) tasks. The results show that the neural approach outperforms the baseline and our holistic approach on bigger resources settings. The use of data augmentation helps to improve the performance of the model in lower resources settings, although it still cannot beat the other systems. In the end, for the low resources setting, our holistic approach performs best in comparison to the baseline and the neural approach (even with data augmentation).

1 Introduction

Lemma:	*illustrate*
Target MSDs:	*V;V.PTCP;PRS*
Target form:	*illustrating*

Figure 1: An example of inflection task in English: given the lemma *illustrate*, we are asked to generate the present participle form *illustrating*.

We address the problem of inflection task: given a **lemma** (e.g. the dictionary form of a word) and the target form's **morphosyntactic descriptions (MSD)**, generate a target inflected form. Figure 1 shows an example of inflection task in English.

Many NLP tasks, like machine translation, require analysis and generation of morphological word forms, even previously unseen ones. Different languages exhibit different richness of morphology. This makes the task an interesting problem. Dreyer and Eisner (2011) show that data sparsity is a common issue for language with rich morphology which usually leads to poor generalisations in machine learning.

There are three main approaches at the problem:

- **The hand-engineered rule-based approach** offers a high accuracy but costs time during construction. It usually faces the world coverage problem and is language-dependent.
- **The supervised approach** automatically induces the rules from a given training data and applies the best rules to generate the target forms by using some classification techniques (Ahlberg et al., 2015). It is practically language independent and relatively easier to build. However, the data sparsity is an issue.
- **The neural approach** is the model which triumphed in the task recently, especially the RNN encoder-decoder model (Kann and Schütze, 2016; Makarov et al., 2017). Some drawbacks of this approach are very long training times and the need for a large amount of training data.

This paper describes the systems we developed for the CoNLL–SIGMORPHON 2018 Shared Task 1 (Cotterell et al., 2018). The recent success of neural approach encouraged us to implement a sequence-to-sequence (seq2seq) model to solve the task. Knowing that the neural approach tends to need a large amount of training data, we also consider another approach as a back-off, which is a holistic approach. We treat the task of generating target forms as the task of solving analogical equations between words.

2 Languages and data

Task 1 consists of 93 different languages. 10 additional surprise languages are given in the middle of

Proceedings of the CoNLL–SIGMORPHON 2018 Shared Task: Universal Morphological Reinflection, pages 33–42,
Brussels, Belgium, October 31, 2018.

Feature	low			medium			high		
	Avg	Min	Max	Avg	Min	Max	Avg	Min	Max
# of characters (train)	29	14	51	33	14	63	40	19	86
# of unseen characters (dev)	4	0	21	1	0	8	0	0	4
# of lemmata (train)	77	5	100	487	5	989	2,308	15	8,643
# of unseen lemmata (dev)	414	0	984	295	0	960	98	0	743
# of MSDs (train)	22	5	43	23	5	48	25	7	48
# of unseen MSDs (dev)	1	0	8	0	0	2	0	0	1
# of MSD patterns (train)	45	4	95	94	4	726	126	5	1,649
# of unseen MSD patterns (dev)	44	0	695	8	0	414	0	0	6
# of rules (train)	98	26	100	838	147	1,000	5,642	815	9,842
# of unseen rules (dev)	561	12	997	504	30	995	398	22	971

Table 1: Statistics on the dataset given. Number of rules and unseen rules are based on rule extraction method explained in Section 5.3.1.

the development phase. The languages vary from Germanic, Celtic and Slavic languages, which are mainly used in Europe, to Indo-Aryan, Iranian, etc. The dataset consists of lines of triplet. A triplet consists of a lemma, a target form, and a target MSD pattern separated by tabulation characters. The MSDs are morphological tags presented in Unimorph Schema (Kirov et al., 2018).

The provided resources are categorized into:

- **train**: this dataset is the dataset which can be manipulated by the participant to solve the task. It consists of three different sizes:

  low : 100 word forms
  medium : 1,000 word forms
  high : 10,000 word forms

  Telugu has only the *low* training dataset. Some languages have only *low* and *medium* training datasets: Cornish, Greenlandic, Inggrian, Karelian, Kashubian, Kazakh, Khakas, Mapudungun, Middle-Low-German, Middle-High-German, Murrinhpatha, Norman, Old-Irish, Scottish-Gaelic, Tibetan, Turkmen.

- **dev**: this dataset is given to evaluate the performance of our system during the development phase. It consists of 1,000 word forms.

- **test**: this dataset is given at the test phase. This dataset does not contain the target forms. It consists of 1,000 word forms, similar to *dev* dataset.

For some languages, the size of the dataset is smaller than the one mentioned above.

Let us now look at some statistics on the given dataset shown in Table 1. Overall, we can observe a non-decreasing phenomenon from *low* to *high* for all of the number of pieces of information (features) found in the training dataset. On the opposite, we found a non-increasing pattern for the unseen information contained in the dev dataset relatively to training dataset. This shows that bigger resources gradually cover the unseen data encountered in the smaller ones.

Norman, Telugu, Cornish, and Uzbek are languages with a smaller number of lemmata in the training dataset. However, these languages tend to have less, even zero for some languages, unseen lemmata relatively to the dev dataset. They also have a smaller number of unseen characters. On the other hand, languages like Finnish, Russian, English, French, and German have the biggest number of unseen lemmata despite having the biggest number of lemmata in the training dataset compared to other languages.

Let us now turn to the number of MSDs and MSD patterns. These numbers can be interpreted as how large or complex the paradigm for that particular language is. Basque, Quechua, Turkish, Zulu are languages with a higher variety of unique MSD patterns. Basque, in particular, has astonishingly more than 1,600 patterns in comparison to the average of around 126 patterns per language in *high* datasets. The same thing can be seen for *low* and *medium* data. Almost all of the lines are associated with different MSD patterns in the *low* training dataset. Furthermore, Basque also topped as the language with the highest number of unseen MSD patterns for all dataset sizes.

(*substring,*	*replacement,*	*#_of_occurrences*)
'-ε'	'-ing'	1,121
'-e'	'-ing'	832
'-ize'	'-izing'	162
⋮	⋮	⋮
'show'	'showing'	1
⋮	⋮	⋮

Figure 2: Excerpt of affixes remembered by the baseline system from the training data. It memorizes all changes from lemma into target form in various character length.

We also count the number of rules found in the dataset (see the last two rows in Table 1). These rules are not the morphological rules defined by linguists but the one extracted from the method explained in Section 5.3.1. For all languages and all datasets, we count how many unique rules can be extracted and relatively unseen to the respective dev dataset. Telugu, Tatar, and Swahili are the languages with the lowest number of unseen rules. We expect to have good performance in these languages because it means that most of the transformations from lemma into the target form are present in the training data.

3 Baseline system: morpheme-based

The CoNLL–SIGMORPHON 2018 organizers provide a baseline system which is a morpheme-based approach. For each language, it determines whether the language is biased towards prefixing or suffixing. The string will be reversed if the language is biased to prefixing.

For each instance in the training data, it aligns the lemma and target form using Levenshtein distance to cut the word into three categories of candidate: prefix, stem, and suffix. Prefixing and suffixing rules are then extracted and grouped according to the given MSD pattern. The rules are stored as a knowledge in a list of triplets: substring to replace, string replacement, and the number of occurrences. Figure 2 illustrates how the baseline system stores the suffixing rules for English present participle.

In the generation step, it filters the candidate rules by the given target MSD pattern. First, the longest common suffixing rule with the highest number of occurrences is applied. Then the most frequent prefixing rule is applied in the succession to generate the predicted target form.

4 Holistic approach

Another view on the problem is to see that word forms are connected with other word forms systematically. Based on this observation, we can treat the inflection task as the task of solving analogical equation on words[1]:

$$\text{lemma}_t : \text{form}_t :: \text{lemma}_q : x \quad \Rightarrow \quad x = \text{form}_q$$

4.1 Proportional analogy

Analogy is a relationship between four objects: A, B, C, and D usually noted as $A : B :: C : D$. It states that A is to B as C is to D where the ratio between A and B is the same as the ratio between C and D. Here, we consider analogy as a possible way to explain derivation between words as it is already used from the ancient Greek and Latin grammatical tradition up to recent works on computational linguistics, like (Hathout, 2008, 2009).

Various formalisations of analogy have been proposed in (Yvon, 2003; Lepage, 2004; Stroppa and Yvon, 2005). In this work, we select the following definition[2].

$$A : B :: C : D \Rightarrow \begin{cases} d(A,B) = d(C,D) \\ d(A,C) = d(B,D) \\ |A|_a + |D|_a = |B|_a + |C|_a, \\ \forall_a \end{cases} \tag{1}$$

We can construct analogical grids (Fam and Lepage, 2017b, 2018) to give a compact view of different analogies that emerge from a set of words contained in a corpus. An analogical grid is a MxN matrix of words. The special property of this matrix is that any four words from two rows and two columns is an analogy (see Formula 2).

$$\begin{matrix} P_1^1 : P_1^2 : \cdots : P_1^m \\ P_2^1 : P_2^2 : \cdots : P_2^m \\ \vdots \quad \vdots \quad\quad \vdots \\ P_n^1 : P_n^2 : \cdots : P_n^m \end{matrix} \overset{\Delta}{\iff} \begin{matrix} \forall (i,k) \in \{1,\ldots,n\}^2, \\ \forall (j,l) \in \{1,\ldots,m\}^2, \\ P_i^j : P_i^l :: P_k^j : P_k^l \end{matrix} \tag{2}$$

4.2 Solving analogical equation to generate word form

In contrast to the baseline system which uses a morpheme-based approach, our holistic approach

[1] Both lemma_t and form_t are a pair of lemma and target form found in the training data; lemma_q is the lemma given in the question; and form_q is the predicted target form.

[2] $d(A,B)$ stands for the value of the LCS edit distance between two strings A and B that uses only insertions and deletions with cost of 1. $|A|_c$ is the number of occurrences of character c in string A.

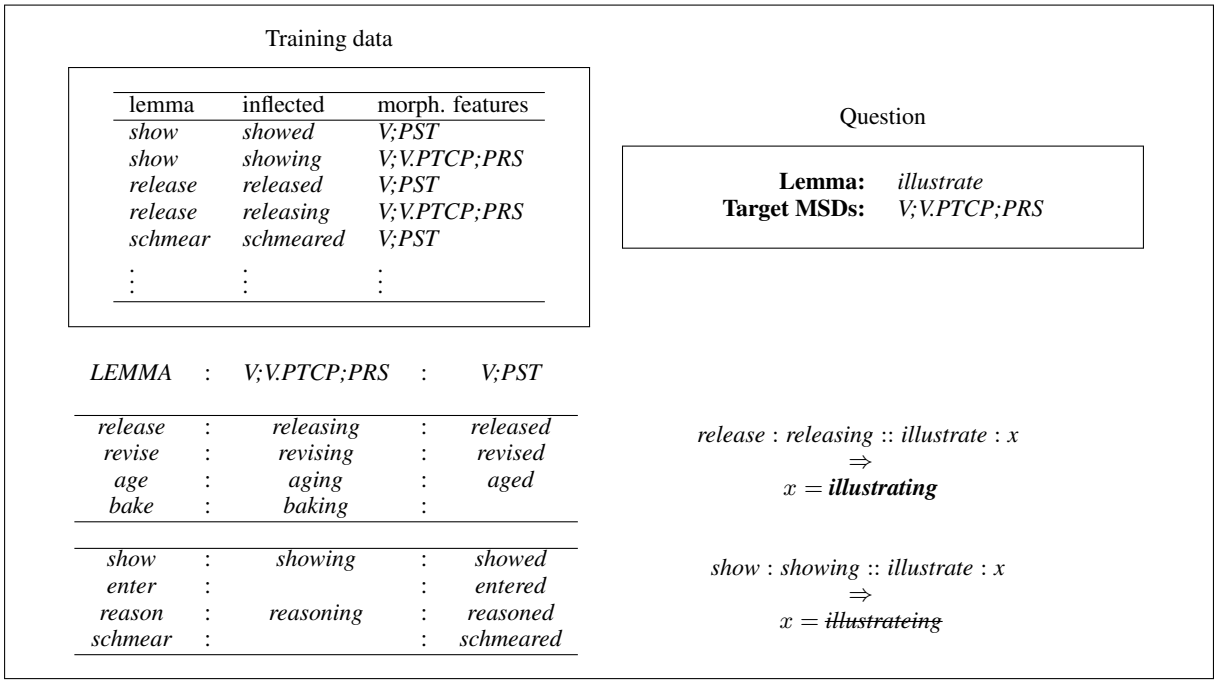

Figure 3: How to generate target form (present participle) of the given lemma *illustrate* as solving analogical equation. Different analogical grids may generate different target forms.

does not break words in pieces (Singh, 2000; Singh and Ford, 2000; Neuvel and Singh, 2001). We generate the target form by solving analogical equation based on the evidence observed in the given training data.

First, the relevant analogical grid is selected according to the given target MSD pattern. If several candidates of analogical equation exist, we use some heuristic features to select the analogical equation. These heuristics are edit distance, longest common subsequence, longest common suffix, and longest common prefix, between the given lemma and lemmata existed in the training dataset. If there are still several candidates after using heuristic features, we solve all of the possible analogical equations to generate all the possible predicted target form. The most frequent answer is chosen as the predicted target form.

For example, Figure 3 illustrates how to generate the target form for the example given in Figure 1. Let us say that we are able to get two analogical grids according to the given MSD pattern. We construct the analogical equation as follows:

$$\text{lemma}_t : \text{form}_t :: \textit{illustrate} : \text{form}_q$$

taken from the first and second column of the analogical grids according to the given MSD pattern. Based on longest common suffix, we choose to use the one in the *top* which produces the word form *illustrating* instead of the *bottom* one which produces *illustrateing*.

5 Neural approach

Following the recent success of neural approach in previous evaluation campaign, we implement a common architecture of seq2seq model. We treat the inflection task as the problem of translating the given target MSDs and lemma into target form. Thus, the input string for the example given in Figure 1 will be as follows.

V V.PTCP PRS i l l u s t r a t e

5.1 Seq2seq model

Our model is a standard seq2seq model with attention mechanism inspired from the one which is used for machine translation (Luong et al., 2015). The difference is that we consider a character or MSD as one token, instead of a word. Each token (character) is represented by a continuous vector representation learned in the embedding layer.

We use a bi-directional Gated Recurrent Unit (GRU) cell (Cho et al., 2014) which is a variation of Long Short-Term Memory (LSTM) cell (Hochreiter and Schmidhuber, 1997) that tries to solve the vanishing gradient problem. Our decoder is two layers of uni-directional GRU cell with attention mechanism. There are various im-

plementations of attention mechanism like (Bahdanau et al., 2015; Luong et al., 2015). In this work, we use the one that has the weight normalization (Salimans and Kingma, 2016) to help the model converges faster.

To handle the unseen tokens, we remember them in a First-In-First-Out (FIFO) list and replace them with a special token <*UNK*> before feeding them into our model. These special tokens are reverted back to the character contained in the list after the decoding phase.

5.2 Hyperparameters

We fixed our hyperparameters for all languages and amounts of resources after doing some preliminary experiments. The number of hidden units is fixed to 100 for each layer in the encoder and decoder. The size of the embedding is 300. We optimize the model using ADAM (Kingma and Ba, 2015) with learning rate of 5×10^4 during training. To make the training process faster, we use mini-batch size of 20.

We trained the model using early-stop mechanism of 30 epochs without improvement on validation data which is a set of lines randomly selected from the original training data.

5.3 Simple data augmentation

Preliminary results show that the neural approach suffers from the lack of data. To tackle this problem, we perform a simple data augmentation which artificially creates additional training data from evidences seen in the original training data. Additional training data is expected to bring improvement to the performance of our model, especially on *low* data situation (Kann and Schütze, 2017; Bergmanis et al., 2017; Silfverberg et al., 2017; Zhou and Neubig, 2017; Nicolai et al., 2017).

5.3.1 Rule extraction

We find the longest common substring between lemma and target form. The left part is assumed as prefix candidate, while the right part is assumed as suffix candidate. Figure 4 shows several examples of rules extracted from the training data in three different languages.

To capture situational affixing where the next or previous character influences the changes, we added the first character from the longest common substring to the extracted prefix candidate and the last character for the suffix candidate. This, for example, happens for regular past form in English where you add only *-d* as suffix for lemmata ended with *e*, instead of adding *-ed*

- Insertion

Language: Irish
Lemma: *fótaidhé-óid*
Target MSDs: *N;NOM;PL;DEF*
Target form: *na fótaidhé-óidí*

	prefix	root	suffix
lemma		*fótaidhé-óid*	
target form	*na*	*fótaidhé-óid*	*í*

- Substitution

Language: French
Lemma: *amoindrir*
Target MSDs: *V;SBJV;PST;3;SG*
Target form: *amoindrît*

	prefix	root	suffix
lemma		*amoindr*	*ir*
target form		*amoindr*	*ît*

- Insertion and substitution at the same time

Language: German
Lemma: *einschließen*
Target MSDs: *V;SBJV;PST;2;SG*
Target form: *schlössest ein*

	prefix	root	suffix
lemma	*ein*	*schl*	*ießen*
target form		*schl*	*össest ein*

Figure 4: Illustrations of rules extraction for data augmentation: simple insertion (Irish); substitution (French); insertion and substitution at the same time (German).

At a glance, it looks similar to how the baseline system extracts the affix rules. However, we only memorize the left (prefix candidate) and right part (suffix candidate), not all of the possible affix combinations with the stem as the baseline system does. It simplifies the rules extraction, and thus, gives us a smaller number of extracted rules in comparison to the baseline system.

5.3.2 Creating additional training data

For each rule which appears less than 10 times in the training data, we artificially create 5 instances of additional training data. The additional training

Method	Accuracy		
	low	medium	high
Baseline	39.3	63.4	77.1
Holistic	**39.6**	64.5	77.3
Seq2seq	13.1	71.3	**90.9**
Seq2seq+Aug	36.9	**78.5**	89.1

Table 2: Average accuracy scores on *dev* dataset.

data is constructed by using a random string with the length of random integer between 1 to 4. Here, we do not employ any language model to asses the probability of the character sequence like the one described in (Silfverberg et al., 2017). For example, we can create the following additional training instance for the examples given in Figure 4. Characters written in boldface are patterns from the extracted rules.

Irish: ***f*bsó*d*** $\Longrightarrow$ ***na f*bsód*í***
French: ***ai*frir** $\Longrightarrow$ ***ai*frît**
German: ***eins***raft***ließen*** $\Longrightarrow$ ***s***raft***lössest ein***

6 Experiment Protocol

We evaluate the performance of the systems using average on accuracy. Accuracy is the ratio of correctly predicted target forms by the total number of questions. Please refer to Formula 3 for the exact definition[3].

$$\text{Accuracy} = \frac{\sum_{i=1}^{N} \delta(\text{predicted}_i = \text{correct}_i)}{N} \times 100 \quad (3)$$

We carry experiments using training dataset and measure the accuracies on dev dataset for all the languages for all training dataset sizes. The system which has the highest score will be picked as our representative system in the test phase for that particular language and dataset size.

7 Results

Table 2 shows the average accuracy in all languages for each of the systems. Our holistic approach is able to perform as good as the baseline system, even slightly better under all of the three dataset sizes. This is the same observation found in (Fam and Lepage, 2017a) on previous year dataset.

The results show that the neural approach using seq2seq model left behind both the baseline system and the holistic approach on *medium* and *high* data situation. The gap is around 15 accuracy points. However, the lack of training data exhibits the drawback of the neural approach as it performs poorly under *low* data situation. Furthermore, the use of data augmentation improves the performance in most cases. We can see an improvement of around 3 times better accuracy on *low* dataset although it still cannot overcome the performance of either the baseline nor the holistic approach.

The baseline system and the holistic approach shine over the neural approach particularly for languages like Albanian, Czech, Haida, Neapolitan, Norwegian-Bokmaal, and Uzbek. Our seq2seq model seems to struggle even on *high* data situation for some of these languages. On the other hand, our seq2seq model gets better accuracy than the baseline system or holistic approach even on *low* data situation in some languages like Azeri, Basque, Breton, Cornish, Greenlandic, Hindi, Karelian, Khaling, Maltese, Middle-Low-German, Middle-High-German, Murrinhpatha, Norman, North-Frisian, Persian, Swahili, Turkish, Turkmen, Welsh, Zulu.

The same trend can be seen on the results for similar languages, like Romance (Catalan, Galician, Portuguese, and Spanish), Semitic (Arabic and Hebrew), and Baltic (Latvian and Lithuanian) languages. The baseline system leads the score on *low* dataset size before started to be outperformed by our seq2seq model on the dataset with bigger sizes. For other language families like Indo-Aryan (Bengali, Hindi, Urdu), Finnic (Estonian and Finnish), and Turkic (Turkish and Turkmen) languages, our seq2seq model steadily leads the score for all dataset sizes. Please refer to Table 3 for detail results per language.

8 Discussion

The results for the baseline system and our holistic approach show the absence of necessity to break down the words into morpheme. The derivation between lemma and target form can also be acquired through analogy. However, selecting the candidates for constructing the analogical equation is a crucial thing. Thus, we need to improve our selection method or use better heuristic features. To handle the problem of unseen MSD patterns, the use of formal concept analysis (Ganter and Wille, 1999) is worth to consider.

[3] N is the total number of questions. $\delta(A = B)$ equals to 1 if the two strings A and B are same, or else it is 0.

Language	Accuracy											
	low				medium				high			
	B	H	S	S+Aug	B	H	S	S+Aug	B	H	S	S+Aug
adyghe	59.8	71.6	35.5	**73.8**	85.5	88.1	88.0	**89.5**	94.8	93.6	**95.6**	95.2
albanian	22.7	**24.5**	0.6	11.6	60.2	**71.5**	44.8	65.2	77.2	**86.4**	81.1	80.5
arabic	22.9	**24.7**	0.1	21.0	37.0	46.2	61.1	**67.9**	42.8	59.1	**93.0**	91.7
armenian	**37.9**	35.7	1.2	34.2	72.4	76.9	76.5	**83.7**	88.6	89.6	**94.1**	90.9
asturian	**59.8**	58.6	19.7	53.1	87.9	88.0	87.4	**89.7**	95.5	95.4	**97.8**	97.2
azeri	21.0	28.0	13.0	**37.0**	48.0	57.0	**69.0**	67.0	69.0	72.0	81.0	**82.0**
bashkir	**38.8**	38.6	11.5	35.9	73.8	71.8	**87.0**	81.0	89.1	86.7	**94.1**	92.6
basque	0.1	0.2	1.9	**8.6**	1.9	2.5	67.0	**79.2**	8.4	9.8	**97.4**	96.9
belarusian	7.2	**10.7**	4.6	5.7	22.5	25.3	44.6	**55.4**	41.4	42.3	**85.3**	80.9
bengali	44.0	43.0	14.0	**49.0**	75.0	74.0	94.0	**96.0**	84.0	84.0	98.0	**99.0**
breton	20.0	17.0	18.0	**61.0**	51.0	59.0	83.0	**88.0**	55.0	61.0	91.0	**92.0**
bulgarian	32.7	33.0	4.3	**49.8**	74.4	76.9	70.6	**82.1**	90.6	89.5	**95.4**	94.3
catalan	**54.3**	51.2	4.6	32.6	82.2	81.2	85.0	**92.3**	94.3	94.2	**98.1**	95.9
classical-syriac	**89.0**	87.0	41.0	72.0	**98.0**	**98.0**	94.0	**98.0**	98.0	97.0	98.0	**100.0**
cornish	2.0	0.0	7.5	**22.5**	4.0	2.0	47.5	**57.5**				
crimean-tatar	53.0	**66.0**	16.0	63.0	74.0	76.0	**95.0**	89.0	93.0	92.0	**99.0**	98.0
czech	38.1	**38.4**	1.6	26.1	78.8	**79.4**	51.1	76.6	89.0	**89.5**	85.5	86.3
danish	57.4	**65.2**	30.2	53.0	78.1	**79.7**	74.3	69.8	88.9	88.8	**91.3**	85.8
estonian	22.6	21.7	0.7	**28.4**	62.4	60.6	60.0	**70.3**	76.2	77.0	**90.6**	88.0
faroese	35.6	**38.3**	3.3	16.6	61.0	**63.0**	51.0	60.6	74.2	74.5	**79.8**	74.5
finnish	15.4	15.4	0.7	**18.7**	43.5	43.3	42.6	**69.9**	79.3	77.2	**84.1**	82.0
friulian	**51.0**	48.0	25.0	49.0	86.0	85.0	89.0	**94.0**	94.0	93.0	98.0	**99.0**
galician	**52.5**	51.9	9.1	30.7	82.3	81.2	77.9	**88.9**	93.7	93.2	**98.4**	97.4
georgian	**71.8**	70.5	17.2	58.9	89.7	90.0	82.9	**92.5**	93.8	94.0	**98.5**	98.4
greek	**27.7**	27.0	2.0	12.0	61.0	**63.0**	44.3	56.6	77.4	77.6	81.7	**83.3**
greenlandic	36.0	42.0	27.5	**57.5**	74.0	60.0	75.0	**85.0**				
haida	**43.0**	28.0	5.0	23.0	**59.0**	**59.0**	50.0	52.0	**71.0**	68.0	53.0	52.0
hebrew	27.9	**29.8**	4.1	13.8	40.0	49.0	**76.3**	**76.3**	55.9	60.7	**98.1**	97.2
hindi	34.9	31.8	23.9	**65.6**	86.1	83.9	94.3	**95.1**	93.6	93.5	**98.6**	97.5
hungarian	14.9	**22.0**	0.9	12.1	39.9	46.7	47.3	**53.1**	68.7	69.7	**77.5**	63.5
icelandic	35.8	**38.1**	6.5	14.9	60.4	**63.6**	52.3	61.3	77.2	77.1	**84.3**	78.7
ingrian	20.0	12.0	**27.5**	20.0	46.0	42.0	**80.0**	75.0				
irish	31.8	**35.7**	3.7	20.9	44.7	49.2	42.6	**57.7**	54.3	58.1	**83.0**	77.2
italian	43.3	**44.4**	3.3	41.3	70.5	83.1	81.3	**91.1**	77.2	93.1	**97.9**	95.4
kabardian	78.0	74.0	51.0	**83.0**	90.0	87.0	**95.0**	**95.0**	90.0	86.0	**96.0**	**96.0**
karelian	40.0	34.0	20.0	**67.5**	48.0	48.0	95.0	**97.5**				
kashubian	56.0	64.0	12.5	**57.5**	74.0	68.0	85.0	**92.5**				
kazakh	44.0	50.0	**52.5**	47.5	64.0	62.0	72.5	**77.5**				
khakas	36.0	48.0	27.5	**65.0**	92.0	92.0	85.0	**92.5**				
khaling	3.9	1.6	4.6	**11.2**	18.4	17.8	77.3	**86.4**	53.8	48.0	**99.6**	98.4
kurmanji	82.1	**85.8**	0.0	58.4	84.7	**88.9**	83.7	88.2	91.9	91.4	**92.8**	91.4
ladin	**59.0**	53.0	30.0	52.0	85.0	86.0	88.0	**95.0**	92.0	91.0	**98.0**	**98.0**
latin	**16.0**	12.6	0.8	5.4	**36.8**	28.5	25.2	36.2	45.6	37.1	**70.1**	55.5
latvian	**53.4**	50.9	4.1	18.3	85.8	**86.6**	60.5	82.4	92.0	91.2	**94.8**	**94.8**
lithuanian	**23.5**	19.4	0.8	5.6	**53.0**	50.3	33.7	51.6	64.7	63.6	**86.2**	84.1
livonian	25.0	**27.0**	1.0	**27.0**	47.0	47.0	69.0	**77.0**	58.0	59.0	**92.0**	**92.0**
lower-sorbian	30.7	**35.8**	2.9	19.3	70.4	79.3	64.1	**81.4**	88.1	87.9	**95.2**	94.8
macedonian	**51.4**	47.4	5.1	37.7	83.8	88.2	75.7	**89.8**	93.2	93.5	**96.4**	95.3
maltese	11.0	19.0	0.0	**23.0**	21.0	29.0	87.0	**93.0**	25.0	29.0	97.0	**98.0**
mapudungun	62.0	60.0	57.5	**95.0**	80.0	88.0	**97.5**	**97.5**				
middle-french	**78.7**	76.1	10.1	67.2	90.8	91.3	89.2	**93.0**	95.8	95.1	**98.8**	96.3
middle-high-german	44.0	48.0	35.0	**67.5**	54.0	60.0	**97.5**	**97.5**				
murrinhpatha	2.0	4.0	25.0	**35.0**	14.0	10.0	**95.0**	90.0				
navajo	14.3	**14.6**	2.0	13.8	31.8	31.2	35.8	**41.5**	40.0	40.5	**82.5**	76.0
neapolitan	**83.0**	81.0	25.0	65.0	94.0	93.0	91.0	**95.0**	**99.0**	98.0	95.0	95.0
norman	38.0	34.0	45.0	**60.0**	60.0	52.0	77.5	**80.0**				
northern-sami	**17.8**	13.1	2.1	11.6	38.8	35.0	43.2	**60.7**	64.5	62.4	**93.4**	88.0
norwegian-bokmaal	69.0	**73.2**	13.8	54.8	79.8	**81.0**	78.0	76.5	**90.6**	90.3	88.9	77.0
norwegian-nynorsk	51.4	**53.7**	11.9	37.6	**61.6**	61.1	52.5	57.0	74.7	75.1	**84.0**	75.8
occitan	**79.0**	77.0	15.0	55.0	87.0	87.0	94.0	**98.0**	94.0	92.0	**100.0**	**100.0**
old-armenian	27.6	**28.8**	1.5	14.8	64.9	68.0	48.9	**69.3**	76.7	79.3	**86.0**	85.1
old-church-slavonic	**34.0**	32.0	11.0	29.0	65.0	65.0	74.0	**78.0**	64.0	57.0	92.0	**96.0**
old-french	30.4	27.6	4.9	**35.4**	61.3	65.2	65.0	**68.9**	79.7	79.5	**87.5**	84.8
old-irish	**12.0**	**12.0**	5.0	5.0	20.0	18.0	20.0	**32.5**				
old-saxon	**25.3**	19.0	2.7	5.2	41.7	35.6	63.0	**68.0**	60.5	56.0	**95.3**	94.6

Language	Accuracy											
	low				medium				high			
	B	H	S	S+Aug	B	H	S	S+Aug	B	H	S	S+Aug
pashto	**41.0**	**41.0**	8.0	21.0	71.0	73.0	69.0	**75.0**	77.0	77.0	**100.0**	98.0
persian	27.1	29.3	2.8	**35.7**	67.3	71.7	82.1	**85.7**	81.0	83.7	**96.0**	95.4
portuguese	**65.7**	64.3	6.9	31.0	92.2	91.8	78.2	**92.5**	97.1	97.2	**97.6**	97.5
quechua	17.1	11.7	3.2	**31.2**	**71.5**	52.1	52.0	55.9	**95.2**	89.6	56.3	56.0
romanian	**44.1**	42.8	3.2	30.3	70.2	**73.0**	59.7	72.3	80.4	78.5	**84.6**	83.1
sanskrit	30.0	37.5	4.8	**42.7**	57.9	77.8	67.9	**80.7**	78.7	83.4	88.0	**88.3**
scottish-gaelic	42.0	38.0	25.0	**50.0**	46.0	44.0	80.0	**90.0**				
serbo-croatian	22.8	20.9	1.3	**25.4**	67.3	65.4	52.9	**74.1**	84.0	85.0	85.2	**86.9**
slovak	37.7	**46.3**	3.3	23.8	71.0	**72.9**	61.3	70.6	82.5	82.8	**90.0**	89.9
slovene	35.2	**37.4**	13.7	25.9	73.5	75.2	63.4	**86.0**	87.3	85.7	**95.2**	93.8
sorani	**20.5**	18.8	1.2	15.6	52.8	52.1	60.3	**71.4**	64.3	60.1	**88.0**	87.7
spanish	**62.4**	57.7	4.9	46.7	85.9	84.9	84.3	**90.3**	91.5	93.6	**97.1**	95.8
swahili	29.0	29.0	27.0	**66.0**	71.0	76.0	**94.0**	93.0	72.0	82.0	**100.0**	**100.0**
swedish	55.6	**62.8**	7.8	39.9	75.2	**76.8**	62.2	68.0	85.8	85.6	**86.1**	76.2
tatar	57.0	**68.0**	17.0	53.0	85.0	88.0	**94.0**	87.0	91.0	91.0	**100.0**	99.0
telugu	80.0	80.0	40.0	**82.5**								
tibetan	**54.0**	42.0	32.5	42.5	48.0	50.0	37.5	**52.5**				
turkish	11.8	12.3	1.1	**28.5**	32.1	40.1	**71.4**	68.3	72.3	74.4	**91.8**	87.0
turkmen	30.0	54.0	37.5	**60.0**	70.0	76.0	87.5	**92.5**				
ukrainian	39.4	**44.6**	6.7	23.3	**72.7**	71.8	55.3	71.3	84.8	84.3	**89.9**	87.1
urdu	29.9	27.4	24.9	**57.8**	86.8	85.7	91.5	**95.0**	96.0	95.7	97.4	**97.6**
uzbek	53.0	35.0	47.0	**74.0**	**93.0**	92.0	78.0	78.0	93.0	**94.0**	78.0	78.0
venetian	**69.0**	68.3	16.6	42.3	89.5	89.0	91.6	**93.1**	93.7	92.1	**99.6**	99.0
votic	**15.0**	12.0	11.0	13.0	38.0	39.0	68.0	**76.0**	41.0	39.0	**78.0**	**78.0**
welsh	26.0	23.0	11.0	**30.0**	55.0	56.0	83.0	**88.0**	71.0	70.0	**95.0**	**95.0**
west-frisian	**47.0**	44.0	8.0	40.0	66.0	64.0	86.0	**93.0**	66.0	62.0	91.0	**95.0**
yiddish	**70.0**	68.0	6.0	60.0	80.0	79.0	83.0	**92.0**	88.0	83.0	98.0	**99.0**
zulu	19.2	18.4	11.0	**33.3**	56.5	65.8	81.6	**86.7**	71.0	81.1	**99.2**	97.7
dutch	53.2	**54.2**	7.8	24.1	72.0	72.8	73.5	**79.4**	88.9	87.3	**96.2**	95.1
english	77.2	**81.7**	28.5	56.4	90.8	**91.4**	85.7	88.0	94.9	94.7	**95.6**	93.6
french	**56.8**	54.5	3.9	37.7	**74.1**	73.7	71.9	71.6	81.9	81.0	**83.7**	73.5
german	51.4	**54.2**	10.7	11.5	74.2	**77.8**	66.0	71.1	83.1	85.8	**88.4**	82.0
kannada	31.0	**36.0**	9.0	27.0	58.0	64.0	83.0	**90.0**	66.0	62.0	**95.0**	**95.0**
middle-low-german	20.0	18.0	22.5	**25.0**	34.0	30.0	90.0	**92.5**				
north-frisian	23.0	23.0	11.0	**27.0**	33.0	32.0	**85.0**	82.0	31.0	32.0	94.0	**95.0**
old-english	**16.7**	11.8	4.3	12.7	28.2	22.1	38.3	**53.3**	44.2	35.8	**83.8**	79.5
polish	40.7	**42.8**	1.8	13.9	73.6	**76.9**	60.0	76.1	88.4	88.6	88.1	**89.5**
russian	41.4	**41.6**	1.8	11.5	75.7	**77.8**	54.4	76.5	85.2	85.7	**89.2**	87.7
Average	39.3	**39.6**	13.1	36.9	63.4	64.5	71.3	**78.5**	77.1	77.3	**90.9**	89.1

Table 3: Accuracy scores on development set (*dev*) in each language for baseline system (**B**), holistic approach(**H**), our seq2seq model without data augmentation (**S**) and with data augmentation (**S+Aug**).

The improvement shown by using data augmentation seems promising. One may think to increase the amount of the artificially created additional training data. However, there is a trade-off between performance and training time. Another thing to consider is how many more additional training data should be created. We can see that the data augmentation seems not to improve the performance on *high* data situation anymore. In addition, the current method to extract the affix rules is very simple. Although it may capture circumfixes, it is still strongly biased to prefixing and suffixing only. A better method is expected to also capture other phenomena, such as parallel infixing (Arabic), repetition (Greek), and reduplication (Malay, Indonesian).

9 Conclusion

We developed several systems for morphological inflection task. The first one is based on a holistic approach. We generate the target forms by solving analogical equations on words. The second one is a seq2seq neural network model. A simple data augmentation is also implemented to help on low data situation. We evaluated their performance on the development dataset and choose the best system on each language and dataset size as our representative system for the submission.

Experimental results show that the neural approach using seq2seq model has the best performance in most cases on *medium* and *high* data situation. However, both baseline and our holistic approach are toe-to-toe on *low* data situation.

References

Malin Ahlberg, Markus Forsberg, and Mans Hulden. 2015. Paradigm classification in supervised learning of morphology. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1024–1029, Denver, Colorado. Association for Computational Linguistics.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *Proceedings of the 3rd International Conference for Learning Representations (ICLR-15)*, San Diego.

Toms Bergmanis, Katharina Kann, Hinrich Schütze, and Sharon Goldwater. 2017. Training data augmentation for low-resource morphological inflection. In *Proceedings of the CoNLL SIGMORPHON 2017 Shared Task: Universal Morphological Reinflection*, pages 31–39, Vancouver. Association for Computational Linguistics.

Kyunghyun Cho, Bart van Merrienboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using rnn encoder–decoder for statistical machine translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1724–1734, Doha, Qatar. Association for Computational Linguistics.

Ryan Cotterell, Christo Kirov, John Sylak-Glassman, Géraldine Walther, Ekaterina Vylomova, Arya D. McCarthy, Katharina Kann, Sebastian Mielke, Garrett Nicolai, Miikka Silfverberg, David Yarowsky, Jason Eisner, and Mans Hulden. 2018. The CoNLL–SIGMORPHON 2018 shared task: Universal morphological reinflection. In *Proceedings of the CoNLL–SIGMORPHON 2018 Shared Task: Universal Morphological Reinflection*, Brussels, Belgium. Association for Computational Linguistics.

Markus Dreyer and Jason Eisner. 2011. Discovering morphological paradigms from plain text using a dirichlet process mixture model. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing (EMNLP'2011)*, pages 616–627, Edinburgh, Scotland, UK. Association for Computational Linguistics.

Rashel Fam and Yves Lepage. 2017a. A holistic approach at a morphological inflection task. In *Proceedings of the 8th Language and Technology Conference (LTC-17)*, pages 88–92, Poznań, Poland. Fundacja uniwersytetu im. Adama Mickiewicza.

Rashel Fam and Yves Lepage. 2017b. 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)*, pages 11–20, Trondheim, Norway.

Rashel Fam and Yves Lepage. 2018. Tools for the production of analogical grids and a resource of n-gram analogical grids in 11 languages. In *Proceedings of the 11th edition of the Language Resources and Evaluation Conference (LREC-18)*, pages 1060–1066, Miyazaki, Japan. ELRA.

Bernhard Ganter and Rudolf Wille. 1999. *Formal Concept Analysis: Mathematical Foundations*. Springer-Verlag Berlin Heidelberg.

Nabil Hathout. 2008. 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*, pages 1–8, Manchester, UK. Coling 2008 Organizing Committee.

Nabil Hathout. 2009. Acquisition of morphological families and derivational series from a machine readable dictionary. *CoRR*, abs/0905.1609.

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

Katharina Kann and Hinrich Schütze. 2016. Med: The LMU system for the SIGMORPHON 2016 Shared Task on morphological reinflection. In *Proceedings of the 2016 Meeting of SIGMORPHON*, Berlin, Germany. Association for Computational Linguistics.

Katharina Kann and Hinrich Schütze. 2017. The LMU system for the CoNLL-SIGMORPHON 2017 Shared Task on universal morphological reinflection. In *Proceedings of the CoNLL SIGMORPHON 2017 Shared Task: Universal Morphological Reinflection*, pages 40–48, Vancouver. Association for Computational Linguistics.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *Proceedings of the 3rd International Conference for Learning Representations (ICLR-15)*, San Diego.

Christo Kirov, Ryan Cotterell, John Sylak-Glassman, Géraldine Walther, Ekaterina Vylomova, Patrick Xia, Manaal Faruqui, Sebastian Mielke, Arya D. McCarthy, Sandra Kübler, David Yarowsky, Jason Eisner, and Mans Hulden. 2018. UniMorph 2.0: Universal Morphology. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan. European Language Resources Association (ELRA).

Yves Lepage. 2004. Analogy and formal languages. *Electronic notes in theoretical computer science*, 53:180–191.

Thang Luong, Hieu Pham, and Christopher D. Manning. 2015. Effective approaches to attention-based neural machine translation. In *Proceedings of the*

2015 Conference on Empirical Methods in Natural Language Processing, pages 1412–1421, Lisbon, Portugal. Association for Computational Linguistics.

Peter Makarov, Tatiana Ruzsics, and Simon Clematide. 2017. Align and copy: UZH at SIGMORPHON 2017 Shared Task for morphological reinflection. In *Proceedings of the CoNLL SIGMORPHON 2017 Shared Task: Universal Morphological Reinflection*, pages 49–57, Vancouver. Association for Computational Linguistics.

Sylvain Neuvel and Rajendra Singh. 2001. Vive la différence ! what morphology is about. *Folia Linguistica*, 35(3-4):313–320.

Garrett Nicolai, Bradley Hauer, Mohammad Motallebi, Saeed Najafi, and Grzegorz Kondrak. 2017. If you can't beat them, join them: the University of Alberta system description. In *Proceedings of the CoNLL SIGMORPHON 2017 Shared Task: Universal Morphological Reinflection*, pages 79–84, Vancouver. Association for Computational Linguistics.

Tim Salimans and Diederik P Kingma. 2016. Weight normalization: A simple reparameterization to accelerate training of deep neural networks. In D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett, editors, *Advances in Neural Information Processing Systems 29*, pages 901–909. Curran Associates, Inc.

Miikka Silfverberg, Adam Wiemerslage, Ling Liu, and Lingshuang Jack Mao. 2017. Data augmentation for morphological reinflection. In *Proceedings of the CoNLL SIGMORPHON 2017 Shared Task: Universal Morphological Reinflection*, pages 90–99, Vancouver. Association for Computational Linguistics.

Rajendra Singh, editor. 2000. *The Yearbook of South Asian Languages and Linguistics-200*. Sage, Thousand Oaks.

Rajendra Singh and Alan Ford. 2000. In praise of Sakatayana: some remarks on whole word morphology. In Rajendra Singh, editor, *The Yearbook of South Asian Languages and Linguistics-200*. Sage, Thousand Oaks.

Nicolas Stroppa and François Yvon. 2005. An analogical learner for morphological analysis. In *Proceedings of the Ninth Conference on Computational Natural Language Learning (CoNLL-2005)*, pages 120–127, Ann Arbor, Michigan. Association for Computational Linguistics.

François Yvon. 2003. Finite-state machines solving analogies on words. Technical report, ENST.

Chunting Zhou and Graham Neubig. 2017. Morphological inflection generation with multi-space variational encoder-decoders. In *Proceedings of the CoNLL SIGMORPHON 2017 Shared Task: Universal Morphological Reinflection*, pages 58–65, Vancouver. Association for Computational Linguistics.