Morphological Inflection Generation with Multi-space Variational Encoder-Decoders

Chunting Zhou and Graham Neubig
Language Technologies Institute
Carnegie Mellon University
ctzhou,gneubig@cs.cmu.edu

Abstract

This paper describes the CMU submission to shared task 1 of SIGMORPHON 2017. The system is based on the multi-space variational encoder-decoder (MSVED) method of Zhou and Neubig (2017), which employs both continuous and discrete latent variables for the variational encoder-decoder and is trained in a semi-supervised fashion. We discuss some language-specific errors and present result analysis.

1 Introduction

In morphologically rich languages, different affixes (i.e. prefixes, infixes, suffixes) can be combined with the lemma to reflect various syntactic and semantic features of a word. In many areas of natural language processing (NLP) it is important that systems are able to correctly analyze and generate different morphological forms, including previously unseen forms. The ability to accurately analyze and generate morphological forms is crucial to creating applications such as machine translation (Chahuneau et al., 2013) and information retrieval (Darwish and Oard, 2007). Accordingly, learning morphological reinflection patterns from labeled data is an important challenge.

The Universal Morphological Reinflection task at SIGMORPHON 2017 (Cotterell and Schütze, 2017) is an evaluation campaign aimed at systems that tackle the task of morphological inflection. It extends the SIGMORPHON 2016 Morphological Reinflection by conducting tasks in 52 languages instead of 10 Cotterell et al. (2016).

In our system submission, we utilize multi-space variational encoder-decoders (MSVEDs), which are a varitional encoder-decoder with both continuous and discrete latent variables (Zhou and Neubig, 2017). The continuous latent variable is expected to reflect the lemma form of a word and the discrete variables are used to induce the desired labels of the inflected word. The whole model is trained in a semi-supervised fashion. For the supervised part we are reducing the reconstruction error of generating the inflected word given the lemma and corresponding tags. For the unsupervised part, we introduce the discrete latent variables representing the morphological tags, and train an auto-encoder over unlabeled corpora. Thus, the training objective includes both the variational lower bound on the marginal log likelihood of the observed parallel training data and the monolingual data.

There are two tasks in SIGMORPHON 2017, which are morphology inflection (task 1) and paradigm completion (task 2) respectively. We participated in task 1, inflection generation, in which the goal is to output the inflected form of a lemma given a set of desired morphological tags.[1] Experimental results found that our model works relatively well on the shared task 1 without extensive tuning of hyper-parameters and language-specific features.

2 Methods

In this section we will detail the multi-space variational encoder-decoder model.

Notation: In morphological reinflection, the source sequence $\mathbf{x}^{(s)}$ consists of the characters in an inflected word (e.g., "`played`"), while the associated labels $\mathbf{y}^{(t)}$ describe some linguistic features (e.g., $y^{(t)}_{\text{pos}} = \texttt{Verb}$, $y^{(t)}_{\text{tense}} = \texttt{Past}$) that we

[1] We considered participation in task 2, but while the training data in the second task provides all inflection forms for each lemma, the number of different lemmas is rather smaller, which resulted in our model quickly overfitting to the training data when training the neural model. Therefore, we only took part in the first task this time.

Proceedings of the CoNLL SIGMORPHON 2017 Shared Task: Universal Morphological Reinflection, pages 58–65,
Vancouver, Canada, August 3–4, 2017.

hope to realize in the target. The target sequence $\mathbf{x}^{(t)}$ is therefore the characters of the re-inflected form of the source word (e.g., "played") that satisfy the linguistic features specified by $\mathbf{y}^{(t)}$. For this task, each discrete variable $y_k^{(t)}$ has a set of possible labels (e.g. pos=V, pos=ADJ, etc) and follows a multinomial distribution.

2.1 Preliminaries: Variational Autoencoder

The variational autoencoder (Kingma and Welling, 2014) is an efficient way to handle (continuous) latent variables in neural models. We describe it briefly here, and interested readers can refer to Doersch (2016) for details. The VAE learns a generative model of the probability $p(\mathbf{x})$ of observed data $\mathbf{x}$. The generative process consists of first generating a continuous latent variable $\mathbf{z}$ conditioned on the observed data $\mathbf{x}$, which is termed as the recognition model $q(\mathbf{z}|\mathbf{x})$ (encoder) and then use this latent variable to reconstruct the observation $\mathbf{x}$ known as the reconstruction (decoder) model $p(\mathbf{x}|\mathbf{z})$. VAE uses the variational inference to approximate the intractable posterior by learning a parametric posterior distribution for all observations.Th learning objective function is the variational lower bound on the marginal log likelihood of data:

$$\log p_\theta(\mathbf{x}) \geq \mathbb{E}_{\mathbf{z}\sim q_\phi(\mathbf{z}|\mathbf{x})}[\log p_\theta(\mathbf{x}|\mathbf{z})] - \mathrm{KL}(q_\phi(\mathbf{z}|\mathbf{x})||p(\mathbf{z})) \quad (1)$$

To optimize the parameters with gradient descent, Kingma and Welling (2014) introduce a reparameterization trick that allows for training using simple backpropagation w.r.t. the Gaussian latent variables $\mathbf{z}$.

2.2 Multi-space Variational Encoder-Decoders

There are two cases to discuss when employing the variational encoder-decoder framework for labeled sequence transduction. First, when the labels of the inflected words are known as is the format of the training data in the shared task, we don't need to bother introduction the discrete latent variables for the inflected labels. We maximize the variational lower bound on the conditional log likelihood of observing $\mathbf{x}^{(t)}$ and $\mathbf{y}^{(t)}$ as follows:

$$\begin{aligned}
&\log p_\theta(\mathbf{x}^{(t)}, \mathbf{y}^{(t)}|\mathbf{x}^{(s)}) \\
&\geq \mathbb{E}_{\mathbf{z}\sim q_\phi(\mathbf{z}|\mathbf{x}^{(s)})} \log \frac{p_\theta(\mathbf{x}^{(t)}, \mathbf{y}^{(t)}, \mathbf{z}|\mathbf{x}^{(s)})}{q_\phi(\mathbf{z}|\mathbf{x}^{(s)})} \\
&= \mathbb{E}_{\mathbf{z}\sim q_\phi(\mathbf{z}|\mathbf{x}^{(s)})}[\log p_\theta(\mathbf{x}^{(t)}|\mathbf{y}^{(t)}, \mathbf{z}) + \log p_\pi(\mathbf{y}^{(t)})] - \\
&\mathrm{KL}(q_\phi(\mathbf{z}|\mathbf{x}^{(s)})||p(\mathbf{z})) = \mathcal{L}_l(\mathbf{x}^{(t)}, \mathbf{y}^{(t)}|\mathbf{x}^{(s)}) \quad (2)
\end{aligned}$$

which is a simple extension to the vanilla variational auto-enocders.

Second, in the case of unsupervised learning or when the labels of the inflected word is not observed, we only observe a word or a pair of words and we would like to maximize the log likelihood of the observed data by marginalizing over possible morphological labels, which is consisted to the supervised case above. In this scenario, we can introduce the discrete latent variables for the inflected labels which are used to infer the labels for the target word. Then when decoding the word, we condition both on the continuous and discrete latent variables. For the variational encoder-decoder (MSVED), the variational lower bound on the conditional log likelihood is affected by the recognition model, and thus is computed as:

$$\begin{aligned}
&\log p_\theta(\mathbf{x}^{(t)}|\mathbf{x}^{(s)}) \\
&\geq \mathbb{E}_{(\mathbf{y}^{(t)},\mathbf{z})\sim q_\phi(\mathbf{y}^{(t)},\mathbf{z}|\mathbf{x}^{(s)},\mathbf{x}^{(t)})} \log \frac{p_\theta(\mathbf{x}^{(t)}, \mathbf{y}^{(t)}, \mathbf{z}|\mathbf{x}^{(s)})}{q_\phi(\mathbf{y}^{(t)}, \mathbf{z}|\mathbf{x}^{(s)}, \mathbf{x}^{(t)})} \\
&= \mathbb{E}_{\mathbf{y}^{(t)}\sim q_\phi(\mathbf{y}^{(t)}|\mathbf{x}^{(t)})}[\mathbb{E}_{\mathbf{z}\sim q_\phi(\mathbf{z}|\mathbf{x}^{(s)})}[\log p_\theta(\mathbf{x}^{(t)}|\mathbf{y}^{(t)}, \mathbf{z})] \\
&\quad - \mathrm{KL}(q_\phi(\mathbf{z}|\mathbf{x}^{(s)})||p(\mathbf{z})) + \log p_\pi(\mathbf{y}^{(t)}) \\
&\quad - \log q_\phi(\mathbf{y}^{(t)}|\mathbf{x}^{(t)})] = \mathcal{L}_u(\mathbf{x}^{(t)}|\mathbf{x}^{(s)}) \quad (3)
\end{aligned}$$

While the unsupervised objective is trained by maximizing the following variational lower bound $\mathcal{U}(\mathbf{x})$ on the objective for unlabeled data:

$$\begin{aligned}
\log p_\theta(\mathbf{x}) &\geq \mathbb{E}_{(\mathbf{y},\mathbf{z})\sim q_\phi(\mathbf{y},\mathbf{z}|\mathbf{x})} \log \frac{p_\theta(\mathbf{x}, \mathbf{y}, \mathbf{z})}{q_\phi(\mathbf{y}, \mathbf{z}|\mathbf{x})} \\
&= \mathbb{E}_{\mathbf{y}\sim q_\phi(\mathbf{y}|\mathbf{x})}[\mathbb{E}_{\mathbf{z}\sim q_\phi(\mathbf{z}|\mathbf{x})}[\log p_\theta(\mathbf{x}|\mathbf{z}, \mathbf{y})] \\
&\quad - \mathrm{KL}(q_\phi(\mathbf{z}|\mathbf{x})||p(\mathbf{z})) + \log p_\pi(\mathbf{y}) \\
&\quad - \log q_\phi(\mathbf{y}|\mathbf{x})] = \mathcal{U}(\mathbf{x}) \quad (4)
\end{aligned}$$

Note that when labels are not observed, the inference model $q_\phi(\mathbf{y}|\mathbf{x})$ has the form of a discriminative classifier, thus we can use observed labels as the supervision signal to learn a better classifier. In this case we also minimize the following cross entropy as the classification loss:

$$\mathcal{D}(\mathbf{x}, \mathbf{y}) = \mathbb{E}_{(\mathbf{x},\mathbf{y})\sim p_l(\mathbf{x},\mathbf{y})}[-\log q_\phi(\mathbf{y}|\mathbf{x})] \quad (5)$$

where $p_l(\mathbf{x}, \mathbf{y})$ is the distribution of labeled data.

To sum up, the semi-supervised model (**Semi-sup**) is trained to maximize the variational lower bounds and minimize the classification cross-entropy error of 5.

$$\mathcal{L}(\mathbf{x^{(s)}}, \mathbf{x^{(t)}}, \mathbf{y^{(t)}}, \mathbf{x}) = \alpha \cdot \mathcal{U}(\mathbf{x}) + \mathcal{L}_u(\mathbf{x}^{(s)}|\mathbf{x}^{(t)}) + \mathcal{L}_l(\mathbf{x}^{(t)}, \mathbf{y}^{(t)}|\mathbf{x}^{(s)}) - \mathcal{D}(\mathbf{x}^{(t)}, \mathbf{y}^{(t)}) \quad (6)$$

The weight α controls the relative weight between the loss from unlabeled data and labeled data.

3 Learning MSVED

3.1 Learning Discrete Latent Variables

One challenge in training our model is that discrete random variables in a stochastic computation graph prevent the gradient from being back-propagated due to their non-differentiability, and marginalizing over all label combinations is also infeasible in our case.

To alleviate this problem, we use the recently proposed Gumbel-Softmax trick (Maddison et al., 2014; Gumbel and Lieblein, 1954) to create a differentiable estimator for categorical variables. In experiments, we start with a relatively large temperature and decrease it gradually.

3.2 Learning Continuous Latent Variables

We observe that with the vanilla implementation the KL cost quickly decreases to near zero, setting $q_\phi(\mathbf{z}|\mathbf{x})$ equal to standard normal distribution. In this case, the RNN decoder can easily degenerate into an RNN language model. Hence, the latent variables are ignored by the decoder and cannot encode any useful information. The latent variable $\mathbf{z}$ learns an undesirable distribution that coincides with the imposed prior distribution but has no contribution to the decoder. To force the decoder to use the latent variables, we take the following two approaches which are similar to Bowman et al. (2016).

KL-Divergence Annealing: We add a coefficient λ to the KL cost and gradually anneal it from zero to a predefined threshold $\lambda_{\mathbf{m}}$. At the early stage of training, we set λ to be zero and let the model first figure out how to project the representation of the source sequence to a roughly right point in the space and then regularize it with the KL cost. This technique can also be seen in (Kočiskỳ et al., 2016; Miao and Blunsom, 2016).

Input Dropout in the Decoder: Besides annealing the KL cost, we also randomly drop out the input token with a probability of β at each time step of the decoder during learning. The previous ground-truth token embedding is replaced with a zero vector when dropped. In this way, the RNN decoder could not fully rely on the ground-truth previous token, which ensures that the decoder uses information encoded in the latent variables.

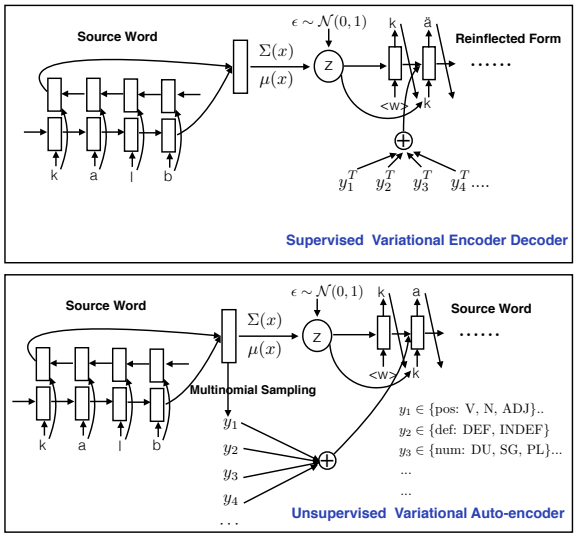

Figure 1: Model architecture for labeled and unlabeled data. For the encoder-decoder model, only one direction from the source to target is given. The classification model is not illustrated in the diagram.

4 Architecture for Morphological Reinflection

The overall model architecture is shown in Fig. 1. Each character and each label is associated with a continuous vector. We employ Gated Recurrent Units (GRUs) for the encoder and decoder. We use only single directional GRUs as the encoder for the input word $\mathbf{x}^{(s)}$. $\mathbf{u}$ is the hidden representation of $\mathbf{x}^{(s)}$ which is the last hidden state of GRUs. and is used as the input for the inference model on $\mathbf{z}$. We represent $\mu(\mathbf{u})$ and $\sigma^2(\mathbf{u})$ as MLPs and sample $\mathbf{z}$ from $\mathcal{N}(\mu(\mathbf{u}), \text{diag}(\sigma^2(\mathbf{u})))$, using $\mathbf{z} = \mu + \sigma \circ \epsilon$, where $\epsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$. Similarly, we can obtain the hidden representation of $\mathbf{x}^{(t)}$ and use this as input to the inference model on each label $\mathbf{y}_i^{(t)}$, which is also an MLP following a softmax layer to generate the categorical probabilities of target labels.

Other experimental setups: We apply temperature annealing in the Gumble-Softmax with the scheme $\max(0.5, \exp(-3e - 5 \cdot t))$ every 2000 updates where t is the update steps. We observe

Language	Dev	Test	Language	Dev	Test	Language	Dev	Test
Latin	66.2	66.2	Navajo	84.9	84.2	English	93.3	94.6
Icelandic	71.7	68.1	French	84.9	82.4	Lower-Sorbian	93.9	91.3
Irish	72.7	71.9	Armenian	85.3	82.3	Italian	94.2	92.6
Finnish	73.4	74.9	Latvian	85.6	87.5	Basque	95.0	97.0
Hungarian	74.5	73.6	Scottish-Gaelic	86.0	68.0	Estonian	95.1	93.7
Faroese	74.8	74.5	Bulgarian	86.3	86.7	Quechua	95.5	95.5
Russian	75.8	76.4	Macedonian	86.6	86.1	Khaling	96.2	94.8
Norwegian-Nynorsk	77.8	73.8	Northern-Sami	86.7	85.8	Hebrew	96.3	97.5
Polish	78.3	78.1	Slovene	87.3	87.8	Portuguese	96.4	96.4
German	79.3	78.7	Danish	88.1	85.4	Catalan	96.9	96.5
Swedish	80.2	80.6	Arabic	88.6	85.9	Urdu	98.4	97.9
Romanian	80.3	78.6	Sorani	89.6	87.8	Persian	98.6	98.7
Lithuanian	80.6	81.6	Slovak	89.6	87.9	Bengali	99.0	99.0
Serbo-Croatian	81.1	79.6	Turkish	90.4	90.3	Welsh	99.0	99.0
Norwegian-Bokmal	81.2	82	Dutch	91.2	88.9	Haida	99.0	97.0
Czech	83.1	81.9	Albanian	91.9	91.3	Hindi	99.9	99.6
Kurmanji	83.4	83.8	Georgian	92.5	92.3			
Ukrainian	84.5	84.0	Spanish	92.5	92.8	Average	87.18	86.21

Table 1: Results of the ensemble system on the development ang test sets of 52 languages.

Language	Src Word	Tgt Labels	Gold Tgt	Ours
Latin	trygon	Pos=N;Case=ABL;Num=PL	trȳgōnibus	trygōnibus
	largio	Mood=SBJV;Num=PL;Per=2;Tense=PST;Asp=PRF;Pos=V	largīvissētis	largissētis
	compenso	Mood=SBJV;Num=SG;Per=3;Tense=PST;Asp=PFV;Pos=V	compensāverit	compenserit
Icelandic	háspil	Pos=N;Def=DEF;Case=GEN;Num=PL	háspilanna	hásplanna
	gallabuxur	Pos=N;Def=INDF;Case=GEN;Num=SG	gallabuxna	gallabölur
	lest	Pos=N;Def=DEF;Case=GEN;Num=SG L	lestarinnar	lestsins

Table 2: Examples of incorrect inflection generation words on the dev data.

Language	Settings	Dev Acc. (Single Model.)
Icelandic	vanilla Encoder-Decoder + attention, w/o data augmentation	81.0
	our model w/o data augmentation and Wiki	78.6
	our model (full)	71.7
Latin	vanilla Encoder-Decoder + attention, w/o data augmentation	74.6
	our model w/o data augmentation and Wiki	66.6
	our model (full)	66.2
Persian	vanilla Encoder-Decoder + attention, w/o data augmentation	99.6
	our model w/o data augmentation and Wiki	99.6
	our model (full)	98.6
Arabic	vanilla Encoder-Decoder + attention, w/o data augmentation	90.7
	our model w/o data augmentation and Wiki	91.3
	our model (full)	88.6

Table 3: Ablation experiments on the effects of data augmentation and WikiData.

that our model is not sensitive to the temperature in this task. All hyperparameters are tuned on the validation set, and include the following: For KL cost annealing, λ_m is set to be 0.2 for all language settings. For character drop-out at the decoder, we empirically set β to be 0.4 for all languages. We set the dimension of character embeddings to be 300, tag label embeddings to be 200, RNN hidden state to be 256, and latent variable $\mathbf{z}$ to be 150 or 100. We set α the weight for the unsupervised

loss to be 0.8. We train the model with Adadelta (Zeiler, 2012) and use early-stop with a patience of 5. Our system is an ensemble of five models and the probability vector at each time step is obtained by averaging the output probabilities from each model

5 Experiments

5.1 Data pre-processing

Creating morphosyntactic tag maps: In our model, we treat the inference model on discrete labels in the form of discriminator, thus we need to know which label belongs to which morphosyntactic dimension. For example, *V* is a label of *Part-of-speech-tagging*. To obtain such mapping from a specific label to the morphosyntactic dimension, we leverage the Universal Morphological Feature Schema (Sylak-Glassman, 2016) and also add the missing schema from the training data to create the key-value pairs of morphosysntactic dimension and label. Then we reformat the labels provided in the data set into the key-value pairs to train a classifier for each morphosyntactic dimension.

Data Augmentation: We augment the data set in the similar way as Kann and Schütze (2016). By doing so, the training data is not limited to the form of lemma to inflected word but can also be any word pairs that share the same lemma. This helps our model generalize better and learn the latent continuous representations more effectively. The size of training data set after augmentation scales with a factor of 2 to 20 times compared with the original one.

Monolingual WikiData: We process the Wikipedia corpus provided by the shared task organizer as our unsupervised training data together with words in the training data. For each language, we first get the character vocabulary of the corresponding training data and only keep words in the Wiki corpus for which characters are all in the character set we obtained. All words that occur less than 20 times are eliminated. We also limit the number of words used during training to be the 50000 most frequent words.

5.2 Results and Analysis

The results on the dev and test data of the 52 languages are presented in 1. We obtain a generation accuracy above 80% over more than 25% languages and an average of 87.2% for both dev and test data. The generation accuracy is almost consistent on the dev and test data except that the test data accuracy of Scottish-Gaelic drops by near 21%. We find that only a medium volume of training data is provided for Scottish-Gaelic. This may be the reason why the model trained for Scottish-Gaelic can not generalize as well as other languages.

We do not tune the hyper-parameters for each language manually. However, we test on different dimensions for the continuous latent variables. The dimension size we have used included 100 and 150. And we observe significant improvement by using a larger dimension size of latent variables over a portion of languages including Faroese, Lithuanian, Navajo, Scottish-gaelic, Northern-sami, Slovene, Sorani, Slovak. However, we also observe that for some languages including Finnish, German, French, etc, the performance drops signficantly after increasing the size of continuous latent variable dimension. This indicates that for different languages, the continuous space required to encode the lemma and inflected information varies from language to language. We will further investigate this in the future work.

5.3 Effect of Data Augmentation and Using Wiki Data

While our performance was reasonable, it was not as good as that presented in our previous work (Zhou and Neubig, 2017), nor was it competitive with the highest-scoring models on the shared task. In order to examine the reason for this, we performed several ablations, the results of which are presented in Tab. 3

First, we first examined the effects of data augmentation and Wiki Data for semi-supervised learning on the performance of our model. By removing the augmented data from the training set, we observe a large gain in the generation accuracy. Besides, we find that Wiki Data for semi-supervised learning doesn't help much to increase the model's performance. The reasons for this will be examined further in the following section.

We additionally reimplemented a vanilla encoder-decoder model with attention that concatenates the input characters and target word tags together with a special token in the middle as the new input sequence to the encoder (Kann and Schütze, 2016). The results show that the vanilla encoder-decoder works better than our

Dimension	Label	Train Data	WikiData	Difference
Case	None	0.58	0.35	-0.22
	ACC	0.14	0.51	0.38
	NOM	0.14	0.12	-0.02
	GEN	0.14	0.01	-0.13
Possession	None	0.86	0.31	-0.55
	PSSD	0.14	0.69	0.55
Language-Specific-Features	None	0.90	0.42	-0.48
	LGSPEC1	0.10	0.58	0.48
Mood	None	0.68	0.10	-0.58
	IND	0.20	0.62	0.42
	IMP	0.02	0.03	0.01
	SBJV	0.10	0.25	0.15
Definiteness	None	0.57	0.60	0.03
	DEF	0.22	0.34	0.12
	NDEF	0.21	0.06	-0.15
Gender	None	0.53	0.52	-0.01
	FEM	0.23	0.27	0.04
	MASC	0.23	0.20	-0.03
Politeness	None	0.85	0.58	-0.28
	INFM	0.14	0.42	0.28
Number	None	0.01	0.16	0.15
	DU	0.22	0.34	0.12
	SG	0.47	0.31	-0.15
	PL	0.30	0.18	-0.11
Person	None	0.58	0.74	0.15
	1	0.06	0.02	-0.05
	3	0.18	0.17	-0.01
	2	0.17	0.08	-0.09
Tense	None	0.90	0.51	-0.40
	PST	0.10	0.49	0.40
Aspect	None	0.80	0.21	-0.59
	PRF	0.10	0.41	0.31
	IPFV	0.10	0.38	0.28
Part-of-Speech	None	0.00	0.03	0.03
	V+V.PTCP	0.01	0.29	0.28
	V+V.MSDR	0.00	0.15	0.14
	N	0.43	0.36	-0.07
	ADJ	0.14	0.14	-0.00
	V	0.42	0.03	-0.39
Voice	None	0.57	0.40	-0.18
	PASS	0.16	0.39	0.22
	ACT	0.27	0.22	-0.05

Table 4: The distribution of morphosyntactic tags for Arabic on Wikipedia and the shared task training data respectively. The linguistic tag classifier has an average accuracy of 93.36% on the Dev data.

model in some cases. We suspect that since task 1 is purely an inflection task and because semi-supervised learning did not provide a particularly large benefit, a simpler model that utilizes attention may be sufficient. This is in contrast to our previous findings, where semi-supervised learning was highly effective, and the proposed model out-performed the simpler attention-based baseline.

5.4 Analysis on the Distribution of Linguistic Tags of Wiki Data and Training Data

One potential reason for the lack of effectiveness of semi-supervised training is that the semi-supervised data that we used for training was not appropriate for the task at hand, or that we were not able to use it in the most effective way. In order to do so, we analyze the distribution of linguistic tags for words from the training data in the shared task and the Wiki Data provided by the organizer, with the hypothesis that if the distribution of tags for the Wiki Data is very different from the training and test data for the shared task, our predictions may be biased away from the testing distribution by incorporating the unsupervised Wiki data. To perform this examination, we use the tag classifier trained in our model to predict the labels for each word in the Wiki Data.

Dimension	Label	Train Data	WikiData	Difference
Mood	None	0.79	0.13	-0.66
	IMP	0.03	0.69	0.66
	SBJV	0.18	0.18	-0.00
Politeness	None	0.52	0.30	-0.22
	COL	0.48	0.70	0.22
Number	None	0.04	0.67	0.62
	SG	0.48	0.19	-0.30
	PL	0.47	0.15	-0.32
Person	None	0.04	0.28	0.24
	1	0.31	0.23	-0.08
	3	0.31	0.22	-0.09
	2	0.34	0.27	-0.07
Finiteness	None	0.98	0.33	-0.66
	NFIN	0.02	0.67	0.66
Tense	None	0.13	0.07	-0.07
	FUT	0.04	0.42	0.38
	PST	0.46	0.14	-0.32
	PRS	0.36	0.37	0.01
Aspect	None	0.39	0.37	-0.01
	PROG	0.18	0.07	-0.11
	PRF	0.17	0.03	-0.14
	IPFV	0.18	0.09	-0.08
	PFV	0.09	0.44	0.35
Part-of-Speech	None	0.00	0.44	0.44
	V+V.PTCP	0.03	0.18	0.15
	V	0.97	0.38	-0.59

Table 5: The distribution of morphosyntactic tags for Persian on Wikipedia and the shared task training data respectively. The linguistic tag classifier has an average accuracy of 95.26% on the Dev data.

The percentages of each label within each morphosyntactic dimension for Arabic and Persian are listed in Tab. 4 and Tab. 5. We found that the distribution of the linguistic tags for the Wiki Data and the training data in the shared task are not always consistent. For example, in Arabic, the distributions of predicted tags with respect to case, possession, part-of-speech, and several other classes differ significantly from the original training data. Such difference suggests that either the words in the unlabeled Wiki Data have very different characteristics than our training set, or our tag classifier is not functioning properly to identify the tags. Either case would be detrimental to semi-supervised learning. The problem is even more stark for Persian: in Persian the only labeled words in the training data are verbs, so all non-verb words in the Wiki Data will receive an incorrect analysis, which is obviously not conducive to learning anything useful. As a recommendation for the future, when performing semi-supervised learning for morphology where the labeled data only represents a subset of the phenomena in the language, it is likely necessary to first identify which of the available unlabeled data is appropriate for semi-supervised learning before applying such methods.

5.5 Case Study on Inflected Words

In Tab. 1, we notice that the performance on Latin is relatively poor compared with other languages. Latin is a highly inflected languages with three distinct genders, seven noun cases, four verb conjugations, four verb principal parts, six tenses, three persons, three moods, two voices, two aspects and two numbers. In addition to this, we found that the data set size after augmentation was only enlarged 2 times. We examine some errors made by our system on two worst performed languages Latin and Icelandic in Tab. 2. As shown in the table, we found that the inflections of Latin and Icelandic have more suffix variations from the lemma. We guess our model still lacks the ability to capture more complicated inflections for such languages. We might consider adding the dependencies between different inflections for multiple target labels in our future work.

6 Conclusion and Future Work

In this work, we further examine the method proposed in (Zhou and Neubig, 2017) for the shared task of SIGMORPHON 2017 on 52 languages and

demonstrate the effectiveness of this approach. We will further improve our model's sophistication by investigating strategies for choosing appropriate semi-supervised data, and examining the model's performance on languages with a high inflection level.

Acknowledgments

This work has been supported in part by an Amazon Academic Research Award. We thank Matthew Honnibal for pointing out that the data distribution of Wikipedia corpus might be biased.

References

Samuel R Bowman, Luke Vilnis, Oriol Vinyals, Andrew M Dai, Rafal Jozefowicz, and Samy Bengio. 2016. Generating sentences from a continuous space. *Proceedings of CoNLL* .

Victor Chahuneau, Eva Schlinger, Noah A Smith, and Chris Dyer. 2013. Translating into morphologically rich languages with synthetic phrases. Association for Computational Linguistics.

Ryan Cotterell, Christo Kirov, John Sylak-Glassman, David Yarowsky, Jason Eisner, and Mans Hulden. 2016. The sigmorphon 2016 shared task—morphological reinflection. In *Proceedings of the 2016 Meeting of SIGMORPHON*. Association for Computational Linguistics, Berlin, Germany.

Ryan Cotterell and Hinrich Schütze. 2017. Joint semantic synthesis and morphological analysis of the derived word. *arXiv preprint arXiv:1701.00946* .

Kareem Darwish and Douglas W Oard. 2007. Adapting morphology for arabic information retrieval. In *Arabic Computational Morphology*, Springer, pages 245–262.

Carl Doersch. 2016. Tutorial on variational autoencoders. *arXiv preprint arXiv:1606.05908* .

Emil Julius Gumbel and Julius Lieblein. 1954. Statistical theory of extreme values and some practical applications: a series of lectures. *US Government Printing Office Washington* .

Katharina Kann and Hinrich Schütze. 2016. Med: The lmu system for the sigmorphon 2016 shared task on morphological reinflection. In *In Proceedings of the 14th SIGMORPHON Workshop on Computational Research in Phonetics, Phonology, and Morphology*. Berlin, Germany.

D.P. Kingma and M. Welling. 2014. Auto-encoding variational bayes. In *The International Conference on Learning Representations*.

Tomáš Kočiskỳ, Gábor Melis, Edward Grefenstette, Chris Dyer, Wang Ling, Phil Blunsom, and Karl Moritz Hermann. 2016. Semantic parsing with semi-supervised sequential autoencoders. *the 2016 Conference on Empirical Methods in Natural Language Processing (EMNLP)* .

Chris J Maddison, Daniel Tarlow, and Tom Minka. 2014. A* sampling. In *Advances in Neural Information Processing Systems*. pages 3086–3094.

Yishu Miao and Phil Blunsom. 2016. Language as a latent variable: Discrete generative models for sentence compression. *the 2016 Conference on Empirical Methods in Natural Language Processing (EMNLP)* .

John Sylak-Glassman. 2016. The composition and use of the universal morphological feature schema (unimorph schema) .

Matthew D Zeiler. 2012. Adadelta: an adaptive learning rate method. *arXiv preprint arXiv:1212.5701* .

Chunting Zhou and Graham Neubig. 2017. Multi-space variational encoder-decoders for semi-supervised labeled sequence transduction. In *The 55th Annual Meeting of the Association for Computational Linguistics (ACL)*. Vancouver, Canada. https://arxiv.org/abs/1704.01691.