Two Birds with One Stone: Investigating Invertible Neural Networks for Inverse Problems in Morphology

Gözde Gül Şahin, Iryna Gurevych
Research Training Group AIPHES and UKP Lab,
Department of Computer Science, Technische Universität Darmstadt
www.ukp.tu-darmstadt.de
www.aiphes.tu-darmstadt.de

Abstract

Most problems in natural language processing can be approximated as inverse problems such as analysis and generation at variety of levels from morphological *(e.g., cat+Plural↔cats)* to semantic *(e.g., (call + 1 2)↔"Calculate one plus two.")*. Although the tasks in both directions are closely related, general approach in the field has been to design separate models specific for each task. However, having one shared model for both tasks, would help the researchers exploit the common knowledge among these problems with reduced time and memory requirements. We investigate a specific class of neural networks, called Invertible Neural Networks (INNs) (Ardizzone et al. 2019) that enable simultaneous optimization in both directions, hence allow addressing of inverse problems via a single model. In this study, we investigate INNs on morphological problems casted as inverse problems. We apply INNs to various morphological tasks with varying ambiguity and show that they provide competitive performance in both directions. We show that they are able to recover the morphological input parameters, i.e., predicting the lemma (e.g., cat) or the morphological tags (e.g., Plural) when run in the reverse direction, without any significant performance drop in the forward direction, i.e., predicting the surface form (e.g., cats).

Introduction

Inverse problem is a general term used in natural sciences and mathematics to describe the process of recovering the hidden model parameters, $\mathbf{x}$, from a set of observations, $\mathbf{y}$. In general, the forward problem, i.e., generating observations/outputs from parameters, is well-defined; while the inverse problem is generally ill-posed, i.e., no (unique) solution exists. For instance, inferring seismic properties of the Earth's interior from surface observations is a typical inverse problem in geophysics (Snieder and Trampert 1999); since forward problem is well-defined and can be simulated by a forward model, recovering the seismic properties that lead to a specific surface value is ill-posed. Although inverse problems have been tackled in many fields such as imaging (Bertero and Boccacci 1998; Adler et al. 2019), astronomy (Osborne, Armstrong, and Fletcher 2019; Ardizzone et al. 2019) and geophysics (Snieder and Trampert 1999), natural language processing (NLP) has not yet witnessed an explicit exploration, mostly due to the discrete nature of human language. However NLP contains many tasks that resemble inverse problems such as semantic parsing ↔ text generation, morphological analysis ↔ morphological generation, text-to-data (e.g., database records) ↔ data-to-text generation and many more. Recently, the NLP field has replaced traditional discrete representations of text with dense and low-dimensional continuous ones, referred to as word/sentence vectors. This has enabled us to reformulate some of the classical morphological tasks as inverse problems within an existing framework that has been previously employed for such problems in other fields.

Inverse problems always exist together with their forward problem. Even so, traditional methods in NLP, optimize the inverse problems in an isolated fashion via a supervised loss for direct posterior probability learning, $p(\mathbf{x}\,|\,\mathbf{y})$, causing challenges for ill-posed problems, i.e., multiple possible $\mathbf{x}$ values in $\mathbf{y} \rightarrow \mathbf{x}$ [1]. In addition, the direct formulation ignores the connection to its forward problem, losing the potential to exploit the shared knowledge between those two. Furthermore, direct optimization of those problems requires two dedicated models to be trained separately, increasing the computational requirements.

Recently, a new class of neural networks, called Invertible Neural Networks (INNs), have been introduced by Ardizzone et al. (2019) following the research line of Dinh, Krueger, and Bengio; Dinh, Sohl-Dickstein, and Bengio; Kingma and Dhariwal (2014; 2017; 2018). INNs offer three key functionalities: (i) modeling inverse problems within a single network (shared parameters), (ii) having an invertible architecture that allows getting the inverse mapping $\mathbf{y} \rightarrow \mathbf{x}$ for free during prediction; and enables efficient bidirectional training *(i.e., training the network at both ends)*; and (iii) addressing the ill-posed problems via an additional latent output variable $\mathbf{z}$ to have a one-to-one input and output mapping as $\mathbf{x} \leftrightarrow [\mathbf{y}, \mathbf{z}]$.

In this work, we model two well-known morphologi-

[1] In case of a unique solution, we refer to it as a well-posed inverse problem, otherwise it is named as ill-posed following Kabanikhin (2008)

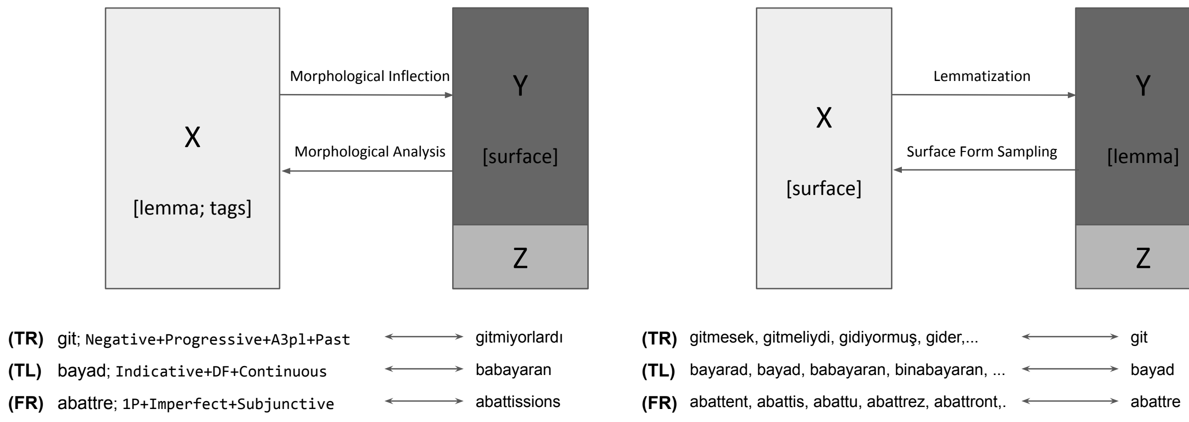
X
[lemma; tags]
Morphological Inflection
Morphological Analysis
Y
[surface]
Z
X
[surface]
Lemmatization
Surface Form Sampling
Y
[lemma]
Z
(TR) git; Negative+Progressive+A3pl+Past ⟷ gitmiyorlardı
(TL) bayad; Indicative+DF+Continuous ⟷ babayaran
(FR) abattre; 1P+Imperfect+Subjunctive ⟷ abattissions
(TR) gitmesek, gitmeliydi, gidiyormuş, gider,... ⟷ git
(TL) bayarad, bayad, babayaran, binabayaran, ... ⟷ bayad
(FR) abattent, abattis, abattu, abattrez, abattront,. ⟷ abattre

Figure 1: Modeling of morphological tasks within the INN framework. **TR:** Turkish, **TL:** Tagalog, **FR:** French

cal tasks: *morphological inflection* and *lemmatization* together with their approximated inverses via modified invertible neural networks. *Morphological inflection* is defined as inflecting a lemma with a set of inflectional morphemes to generate a surface form. Some examples of morphological inflection for Turkish, Tagalog and French are given in the left side of Fig. 1. It is considered as a well-defined problem, since (almost always) only one surface form can be generated given the lemma and the tag combination. Its inverse problem is *morphological analysis*, that aims to extract the lemma and the tags given the surface form. Unlike inflection, it may be ambiguous, i.e. there may be multiple possible analysis of a single surface form, depending on the linguistic properties of the language. Note that the examples in Fig. 1-left have only one possible analysis. *Lemmatization* is the task of finding the lemma of a given surface form. Although it is possible to have multiple lemmas per surface form, it is treated as a well-defined task in literature [2]. Its inverse problem is surface form generation, similar to morphological inflection, however without the guidance of morphological tags. We name it as surface form sampling, since one can sample a z variable and generate surface forms given the lemma as shown in Fig. 1-right.

In this work, we recognize, for the first time, that NLP contains problems similar to "inverse problems" that are found in many other fields such as imaging, geophysics, medical and astronomy. This recognition would provide new ways to approach traditional NLP problems by adapting the existing solutions developed for inverse problems in other fields. We experiment with several languages from diverse language families and morphological typologies and show that:

- INNs have the ability to optimize for both problems simultaneously providing strong results for both, however generally 2%-6% less then a strong method designed specifically for the forward task under our experimental settings,
- Bi-directional training is *crucial* to provide competitive scores for both sides of the network, e.g., adding reverse training to morphological inflection, boosts the lemma prediction performance (inverse problem), while causing only a slight drop in morphological inflection (forward process),
- Introduction of additional categorical latent variables provide improvements in lemmatization for *all languages*, even surpassing a strong model for some languages,
- INNs implicitly learn simple morphological tag distributions even without a dedicated loss function, however words needed direct supervision.

We believe that this initial exploration of INNs for inverse problems in morphology would encourage research in that direction for more complex NLP tasks.

Background: Invertible Neural Networks (INNs)

INNs are originally proposed to address ill-posed inverse problems that have a well-defined forward process, i.e., a unique $x \to y$, whereas the inverse problem is ambiguous, i.e., multiple $y \to x$ mappings. It is addressed by introducing an additional latent variable, as illustrated in Fig. 2, that turns $\mathbf{x} \leftrightarrow [\mathbf{y}, \mathbf{z}]$ into a bijective mapping. In other words, it enables the reparametrization of $p(\mathbf{x} \,|\, \mathbf{y})$ into a deterministic function $\mathbf{x} = f(\mathbf{y}, \mathbf{z})$. Therefore we can define an inverse function $x = f^{-1}(\mathbf{y}, \mathbf{z}) = g(\mathbf{y}, \mathbf{z})$ to estimate the full posterior distribution, $p(\mathbf{x} \,|\, \mathbf{y})$. Recently proposed neural network components that are invertible such as coupling layers can now be employed as described in more details later in this section. Such networks are unique since they can be run backwards to get the inverse $\mathbf{y} \to \mathbf{x}$ without any additional cost at prediction time. Furthmermore, INNs offer efficient training procedure that can optimize both ends of the network simultaneously, which is referred to as bi-directional training.

[2]For instance, the Turkish word "dolar" can have the lemma "dol" (to fill), "dola" (to wrap) or "dolar" (dollar), which can only be determined when the surface form "dolar" is given within a sentence. However this task is considered different and named contextual lemmatization. Therefore such cases are not handled in scope of the lemmatization task.

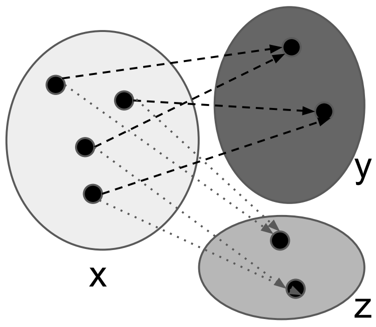

Figure 2: Additional latent variable $\mathbf{z}$ for one-to-one mapping

Invertible Component

INNs consist of layers of "reversible block"s shown with Fig. 3 that contain complemantary affine coupling layers proposed by Dinh, Sohl-Dickstein, and Bengio (2017). First, the input vector u is split into two halves $\mathbf{u}_1$ and $\mathbf{u}_2$ (in a fixed way), which are then transformed by the learned affine functions s_i and t_i, where $i \in \{1, 2\}$ using the following equations:

$$\mathbf{v}_1 = \mathbf{u}_1 \odot \exp\big(s_2(\mathbf{u}_2)\big) + t_2(\mathbf{u}_2)$$
$$\mathbf{v}_2 = \mathbf{u}_2 \odot \exp\big(s_1(\mathbf{v}_1)\big) + t_1(\mathbf{v}_1),$$

where $\odot$ denotes element-wise multiplication. The output $\mathbf{v}$ is then simply calculated by concatenating $\mathbf{v}_1; \mathbf{v}_2$. The input $\mathbf{u}$ can then be recovered with:

$$\mathbf{u}_2 = (\mathbf{v}_2 - t_1(\mathbf{v}_1)) \odot \exp\big(-s_1(\mathbf{v}_1)\big)$$
$$\mathbf{u}_1 = (\mathbf{v}_1 - t_2(\mathbf{u}_2)) \odot \exp\big(-s_2(\mathbf{u}_2)\big).$$

s_i and t_i need not be invertible and generally represented by multi-layered feed-forward neural networks with nonlinear activations, which can then be trained by standard backpropogation algorithm. Similar to previous methods, in order to vary the splits $[\mathbf{u}_1, \mathbf{u}_2]$ among different layers, we employ a permutation layer between the reversible blocks which shuffle the elements in a randomized but a fixed way.

Bi-directional Training

As discussed by Grover, Dhar, and Ermon (2018), networks that are invertible offer a unique oppurtunity to optimize for both the input and output domains *simultaneously*, i.e., apply the loss functions $\mathcal{L}_x$, $\mathcal{L}_y$ and $\mathcal{L}_z$ given in Fig. 3 for the forward and the inverse passes at the same time. This is achieved by performing forward and inverse iterations in an alternating fashion, and then accumulating the gradients to update the network parameters. This training procedure has been shown to be highly beneficial in auto-encoders (Teng and Choromanska 2019) and our own experiments.

Method

Different applications of INNs (Osborne, Armstrong, and Fletcher 2019; Ardizzone et al. 2019; Adler et al. 2019) have commonly used continuous latent variables with Gaussian

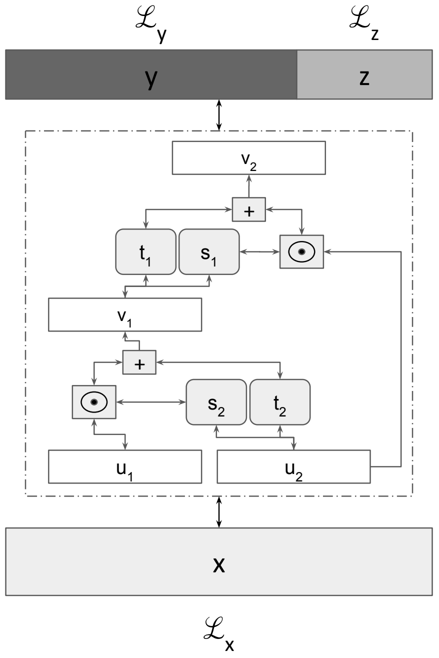

Figure 3: Demonstration of one layer INN. Revertible block is shown inside the dashed box. The operations (e.g., $+, \odot$) are only shown for the forward direction for convenience. They are inverted to subtraction and element-wise division in reverse direction. $\mathcal{L}$ denotes loss function.

priors as $\mathbf{z} \sim \mathcal{N}(\mu, \sigma^2)$. Unlike in previously addressed ill-posed inverse problems, most NLP tasks contain categorical and discrete values. For instance in lemmatization the $\mathbf{z}$ variable is expected to capture the morphological tags which are categorical by definition. Therefore, we have employed categorical latent variables (Jang, Gu, and Poole 2017) that is based on the idea of approximating a categorical distribution via a differentiable distribution called *Gumbel-Softmax (GS)*. This distribution can be smoothly annealed into a categorical distribution via a temperature parameter, τ. As $\tau \to 0$, the GS distributions are identical to samples from a categorical distribution (one-hot), where as $\tau \to \infty$ GS samples become uniform.

Then the method can be formally defined as: Given the input vector $\mathbf{x} \in \mathbb{R}^n$, the output vector $\mathbf{y} \in \mathbb{R}^m$, we assume that the forward process $\mathbf{y} = \mathbf{s}(\mathbf{x})$, *i.e., morphological inflection and lemmatization*, is well-defined by an arbitrary transformation function $\mathbf{s}$. Our goal is to approximate the posterior $p(\mathbf{x} \mid \mathbf{y})$ by a tractable distribution $q(\mathbf{x} \mid \mathbf{y})$. This is then reparameterized by a deterministic function g, represented by the neural network with parameters θ. Categorical output latent variable $\mathbf{z} \in \mathbb{R}^{d \times cat}$ is drawn from a *Gumbel-Softmax (GS)* distribution, where d is latent variable dimension and cat is the number of categories for each dimension. Then g is defined as:

$$\mathbf{x} = g(\mathbf{y}, \mathbf{z}; \theta) \quad \text{where} \quad \mathbf{z} \approx p(z) = GS(z; \tau)$$

We learn the inverse model $g(\mathbf{y}, \mathbf{z}; \theta)$ jointly with the forward model $f(\mathbf{x};\theta)$ that approximates the $s(\mathbf{x})$ function:

$$[\mathbf{y},\mathbf{z}] = f(\mathbf{x};\theta) = [f_y(\mathbf{x};\theta), f_z(\mathbf{x};\theta)] = g^{-1}(\mathbf{x};\theta) \quad \text{and} \quad f_y(\mathbf{x};\theta) \approx s(\mathbf{x})$$

It should be noted that both functions f and g share the same parameters θ, therefore can be implemented by a **single** model. The invertibility property, $f = g^{-1}$ is ensured by the architecture that consists of invertible affine coupling layers defined in previous background section. One of the restrictions of using an invertible architecture is the equality constraint on the dimensions of the input $\mathbf{x}$ and the output $[\mathbf{y},\mathbf{z}]$. To satisfy this constraint, we pad the lower-dimensional side (i.e., input or output) with zeros [3]. Finally the posterior distribution is calculated as the following as shown in (Ardizzone et al. 2019):

$$q(\mathbf{x} = g(\mathbf{y},\mathbf{z};\theta)|\mathbf{y}) = p(\mathbf{z})|J_x|^{-1}, \text{where}$$
$$J_x = det\left(\frac{\partial g(\mathbf{y},\mathbf{z}|\theta)}{\partial[\mathbf{y},\mathbf{z}]}\bigg|_{\mathbf{y},f_z(\mathbf{x})}\right)$$

where J_x is the Jacobian determinant and can be easily calculated due to its triangular structure. We refer the reader to the original study (Ardizzone et al. 2019) for details of the above derivations.

For all morphological tasks, the words (lemmas and surface forms) are divided into smaller units, i.e., subword units, via the Byte-Pair-Encoding (BPE) algorithm (Sennrich, Haddow, and Birch 2016). We then used the 100 dimensional pretrained multilingual embeddings released by Heinzerling and Strube (2018) that provide vector representations for the BPE subword units. Finally each word is represented as sum of its subword unit vectors that we denote as $\vec{lemma} \in \mathbb{R}^{100}$ and $\vec{surface} \in \mathbb{R}^{100}$. For *morphological inflection*, we define the vector $\vec{t} \in [0,1]^N$ to represent the morphological tagset, where N is the number of distinct morphological tags in the training set. Then t_i is set to 1, if the feature i is among the tagset of the surface form.

Variables: For *morphological inflection*, as given in Fig. 1-left, $\mathbf{x}$ is defined as $\mathbf{x} = [\vec{lemma};\vec{t}]$ where ; denotes concatentation operation and $\mathbf{y}$ is simply equal to $\vec{surface}$. According to Fig. 1-right, $\mathbf{x}$ and $\mathbf{y}$ are equal to $\vec{surface}$ and $\vec{lemma}$ accordingly for *lemmatization*.

Loss Functions: We use a supervised loss, implemented via cosine distance function, to minimize the error between the network prediction $\vec{surface}'$ and the gold value $\vec{surface}$ defined as:

$$\mathcal{L}_{surface} = 1 - \frac{\vec{surface} \cdot \vec{surface}'}{||\vec{surface}||||\vec{surface}'||}$$

[3]One could implement an additional loss to keep padding values closer to zero throughout the training. However we have not seen any significant change in performances when implemented.

for both tasks. The loss between the predicted $\vec{lemma}'$ and the gold $\vec{lemma}$ is again calculated via cosine distance. For morphological inflection where $\vec{t'}$ is the predicted and $\vec{t}$ is the gold tag vector, we minimize the error using binary cross entropy loss defined as below:

$$\mathcal{L}_t = -\frac{1}{N}\sum_{i=0}^{N}(t_i.log(t'_i) + (1-t_i)log(1-t'_i))$$

$\mathcal{L}_x$ in Fig. 3 is defined as $\mathcal{L}_{lemma} + \mathcal{L}_t$ for *morphological inflection*, where it is equal to $\mathcal{L}_{surface}$ for *lemmatization* task. Similarly $\mathcal{L}_y$ is defined as $\mathcal{L}_{surface}$ and $\mathcal{L}_{lemma}$ respectively for *morphological inflection* and *lemmatization*. Finally, we use KL divergence loss for $\mathbf{z}$. The final losses are calculated as following:

$$\mathcal{L}_{inflection} = \alpha_x(\mathcal{L}_{lemma} + \mathcal{L}_t) + \alpha_y\mathcal{L}_y + \alpha_z\mathcal{L}_z$$
$$\mathcal{L}_{lemmatization} = \alpha_x\mathcal{L}_x + \alpha_y\mathcal{L}_y + \alpha_z\mathcal{L}_z$$

The relative weights of the losses denoted with α are adjusted as hyperparameters.

Training Procedure: Bi-directional training is performed iteratively to calculate forward and backward losses. The pseudocode for training *morphological inflection* task is given in Algorithm 1.

Algorithm 1: Training Procedure

```
1: procedure TRAININFLECTION
2:     for each lemma, tagset, surface in train_split do
3:         $\mathbf{x} \leftarrow [\vec{lemma};\vec{t}]$
4:         $\mathbf{y} \leftarrow \vec{surface}$
5:         $\mathbf{y}',\mathbf{z}' \leftarrow INN(\mathbf{x})$
6:         $\mathbf{z} \leftarrow GumbelSoftmax(\mathbf{z}')$
7:         $\mathbf{x}' \leftarrow INN(\mathbf{y},\mathbf{z},\textbf{reverse} = \texttt{True})$
8:         $\mathcal{L}_{total} \leftarrow \alpha_x\mathcal{L}_x + \alpha_y\mathcal{L}_y + \alpha_z\mathcal{L}_z$
9:         $\mathcal{L}_{total}.backward()$
```

Testing Procedure: In both models, we choose the word with the highest cosine similarity to the predicted vector during testing. This nearest neighbor search is very efficiently implemented in the gensim framework (Řehůřek and Sojka 2010). To predict morphological tagset, we use sigmoid activation function on $\vec{t'}$ and choose the features with activations above 0.5.

Experiments

We first describe the dataset used in the paper, then detail our experimental design, training settings and evaluation measures.

Dataset

Wicentowski is the most commonly used dataset for lemmatization, which is one of the most interesting ill-posed inverse problems investigated in this paper. Previous studies (Dreyer 2011; Rastogi, Cotterell, and Eisner 2016) use a subset of the dataset created by Wicentowski (2003) [4]. However, since previous works evaluate on the lemmatization task only, this subset does not contain any morphological tags but only the unique lemma-surface pairs. Therefore we have used the original dataset that additionally contains morphological tags. In this dataset, each language has its own set of morphological tags. Furthermore, they are arbitrarily written (e.g., 1P and Past instead of person="1P" and tense="Past") without providing the morphological category like *person* and *tense*. The statistics for the dataset is given in Table 1.

Dataset	Number of tokens			#Tag
	train	dev	test	
FR	81K/14K	10K/2K	10K/2K	20
FIN	70K/46K	9K/6K	9K/6K	35
RO	41K/5K	5K/1K	5K/1K	23
TR	6.3K/1.6K	782/201	783/202	27
TL	1.4K/1.3K	184/165	184/165	25
GA	864/550	108/69	108/69	17

Table 1: Dataset Statistics. First half: Celex data, Second half: Wicentowski dara. *FR*: French, *FIN*: Finnish, *RO*: Romanian, *TR*: Turkish, *GA*: Irish. Numbers after "/" show the number of unique lemma-surface pairs. #Tag: Number of unique morphological tags in the training data.

Experimental Design

We perform morphological experiments with the proposed framework on 6 different languages from diverse language families and morphology types, namely Turkish (Turkic, agglutinative), Romanian (Romance, fusional), Finnish (Uralic, agglutinative), French (Romance, fusional), Irish (Celtic, fusional) and Tagalog (Austronesian, agglutinative). Agglutinative and fusional languages have different morphological properties. Typical properties of agglutinative languages include (1) the ability to generate/derive words by attaching morphemes like beads on a string; and (2) associating each morpheme with one certain semantic unit only, e.g., using the Turkish "lar" morpheme only for plurality. This leads to having a high ratio of out-of-vocabulary words, however the meaning of words are generally predictable. Unlike agglutinative languages, fusional languages typically have smaller ratio of morphemes per word, i.e., not as productive as agglutinative languages. However one morpheme is generally associated with multiple meaning units, e.g., tense and person information conveyed with one morpheme. This generally results in lower out-of-vocabulary ratios with more ambiguous words, especially when the context is unknown.

Training and Evaluation

For all models, we have used 100-dim vectors extracted from pretrained Byte-Pair-Encoding (BPE) model with vocabulary size of 1K, provided by Heinzerling and Strube (2018).

[4]We thank the author for sharing the dataset with us.

		Model	L (EM%)	Tag (F1%)	S (EM%)
Finnish	LEM	baseline	94.0	-	-
		(Aharoni and Goldberg 2017)	**99.6**	-	-
		INN ($\mathcal{L}_y$+$\mathcal{L}_x$)	97.68	-	
		INN (+$\mathcal{L}_z$, dim_z=2, dim_{cat}=3)	97.54	-	
		INN (+$\mathcal{L}_z$, dim_z=6, dim_{cat}=4)	*98.18*	-	
	INF	baseline	-	-	85.15
		INN ($\mathcal{L}_y$)	0.01	11.84	**94.07**
		INN ($\mathcal{L}_y$+$\mathcal{L}_x$)	**95.56**	12.42	92.23
		INN ($\mathcal{L}_y$+$\mathcal{L}_x$+$\mathcal{L}_t$)	93.33	**49.99**	91.26
French	LEM	baseline	95.91	-	-
		(Aharoni and Goldberg 2017)	**98.24**	-	-
		INN ($\mathcal{L}_y$+$\mathcal{L}_x$)	95.91	-	
		INN (+$\mathcal{L}_z$, dim_z=2, dim_{cat}=3)	96.03	-	
		INN (+$\mathcal{L}_z$, dim_z=6, dim_{cat}=4)	*96.14*	-	
	INF	baseline	-	-	88.94
		INN ($\mathcal{L}_y$)	0.0	23.5	**98.18**
		INN ($\mathcal{L}_y$+$\mathcal{L}_x$)	**98.56**	13.32	97.55
		INN ($\mathcal{L}_y$+$\mathcal{L}_x$+$\mathcal{L}_t$)	98.27	**26.01**	97.49
Irish	LEM	baseline	*95.35*	-	-
		(Aharoni and Goldberg 2017)	94.2	-	-
		INN ($\mathcal{L}_y$+$\mathcal{L}_x$)	94.2	-	
		INN (+$\mathcal{L}_z$, dim_z=2, dim_{cat}=3)	94.2	-	
		INN (+$\mathcal{L}_z$, dim_z=6, dim_{cat}=4)	**95.65**	-	
	INF	baseline	-	-	61.11
		INN ($\mathcal{L}_y$)	0.0	25.98	76.85
		INN ($\mathcal{L}_y$+$\mathcal{L}_x$)	97.22	20.55	**79.63**
		INN ($\mathcal{L}_y$+$\mathcal{L}_x$+$\mathcal{L}_t$)	**100**	**67.33**	77.78
Romanian	LEM	baseline	82.39	-	-
		(Aharoni and Goldberg 2017)	**92.89**	-	-
		INN ($\mathcal{L}_y$+$\mathcal{L}_x$)	89.5	-	
		INN (+$\mathcal{L}_z$, dim_z=2, dim_{cat}=3)	88.85	-	
		INN (+$\mathcal{L}_z$, dim_z=6, dim_{cat}=4)	*90.95*	-	
	INF	baseline	-	-	89.45
		INN ($\mathcal{L}_y$)	0.0	**16.5**	**97.87**
		INN ($\mathcal{L}_y$+$\mathcal{L}_x$)	**99.26**	14.24	97.67
		INN ($\mathcal{L}_y$+$\mathcal{L}_x$+$\mathcal{L}_t$)	98.88	0.02	97.34
Tagalog	LEM	baseline	88.48	-	-
		(Aharoni and Goldberg 2017)	**92.72**	-	-
		INN ($\mathcal{L}_y$+$\mathcal{L}_x$)	88.06	-	
		INN (+$\mathcal{L}_z$, dim_z=2, dim_{cat}=3)	90.03	-	
		INN (+$\mathcal{L}_z$, dim_z=6, dim_{cat}=4)	*91.52*	-	
	INF	baseline	-	-	30.05
		INN ($\mathcal{L}_y$)	0.0	15.24	**37.7**
		INN ($\mathcal{L}_y$+$\mathcal{L}_x$)	**85.41**	10.13	33.33
		INN ($\mathcal{L}_y$+$\mathcal{L}_x$+$\mathcal{L}_t$)	69.73	**43.14**	33.88
Turkish	LEM	baseline	95.28	-	-
		(Aharoni and Goldberg 2017)	96.53	-	-
		INN ($\mathcal{L}_y$+$\mathcal{L}_x$)	98.51	-	
		INN (+$\mathcal{L}_z$, dim_z=2, dim_{cat}=3)	99.01	-	
		INN (+$\mathcal{L}_z$, dim_z=6, dim_{cat}=4)	**99.54**	-	
	INF	baseline	-	-	88.89
		INN ($\mathcal{L}_y$)	0.0	36.29	**97.57**
		INN ($\mathcal{L}_y$+$\mathcal{L}_x$)	99.62	24.37	97.45
		INN ($\mathcal{L}_y$+$\mathcal{L}_x$+$\mathcal{L}_t$)	**99.74**	**77.34**	97.45

Table 2: Lemmatization and inflection results for different languages and experimental settings. LEM: Lemmatization, INF: Inflection. L (EM%): Lemma exact match score, S (EM%): Surface exact match score. Best scores are given in **bold** for each score. Second best scores for lemmatization is shown in *italics*. EM: Exact Match, dim_z, dim_{cat}: dimensions of $\mathbf{z}$ variable and number of categories for each.

We have initialized the weight parameters orthogonally. For all INN models, we have used 3 invertible blocks, with 3 fully connected linear layers with hidden dimension of 128 (with ReLU activations in the intermediate layers) as affine coefficient function. We used gradient clipping and early stopping to prevent overfitting. Models are optimized with Adam optimizer with the initial learning rate of 0.001, decreased by 0.3 if scores on development set do not improve for 5 epochs. Unless otherwise stated, we used the weights $\alpha_x = 20$, $\alpha_t = 10$, $\alpha_y = 80$ and $\alpha_z = 1$ respectively for $\mathcal{L}_x$, $\mathcal{L}_t$, $\mathcal{L}_y$ and $\mathcal{L}_z$. These weights are intuitively chosen following the previous studies that has employed INNs. We have trained all models for 30 epochs, and did not perform a comprehensive hyperparameter search since the goal of this study is not delivering state-of-the-art results, rather providing and investigate a different perspective.

For the baseline models, we used 3 fully connected linear layers with hidden dimension of 128, with ReLU activations in intermediate layers. We kept the other settings the same as the INNs (except from zero-padding, since it is not necessary for feed forward networks). Our method is not directly comparable to previous methods for three main reasons: (a) none of the previous morphological inflection or works report two (three when tag scores are included) scores with the same model, (b) none of the previous lemmatization studies offer sampling surface forms with the given lemma, (c) most of the previous works are able to generate unseen words, while we treat the generation as "a selection from a large vocabulary" problem that simplifies the problem space. Nevertheless, we have chosen one of the recent state-of-the-art models by (Aharoni and Goldberg 2017) as a reference for comparison, to provide an insight about the proposed model's performance. For evaluation we use the percentage of exact match score for lemma and surface form predictions, and F1 score for morphological tag predictions due to having multiple labels per form.

Since a nearest neighbor search is performed between the network output and the existing word embedding space using cosine similarity, words that have not been encountered in the training data, would not be predicted. In order to address this, we have extended the vector space with 500K most common words, which can be considered quite many, along with the words encountered in the test data during prediction time.

As discussed previously, *morphological analysis*, i.e., inverse of the *morphological inflection* task, may be ill-posed depending on the linguistic properties of the language. For instance, agglutinative languages have one-to-one morpheme-to-tag mapping, while for fusional languages one morpheme may stand for multiple tags. This means that, *morphological analysis* of fusional languages is ill-defined while (mostly) the opposite is true for agglutinative ones. In case of ill-posedness, continuous sampling from $\mathbf{z}$ is necessary to find distributions of the morphological tags, which are hard to score. To simplify scoring of predicted tags, we ignore $\mathbf{z}$ and only consider the most likely tagset in a current setting. Since the ambiguity greatly varies with the language families, we focus on the tag scores of languages with less ambiguous input.

Results and Analysis

We present the results of lemmatization and the morphological inflection tasks for 6 languages in Table 2. First, we observe that all models in all experimental settings outperform the baseline.

Lemmatization: Inroducing latent variables help increasing the performance of lemmatization *in all languages*, surpassing the (Aharoni and Goldberg 2017) by a small margin for Turkish and Irish, and providing similar results for other languages. More specifically, relative performance to (Aharoni and Goldberg 2017) ranges between [-2%, +3%]. In addition, we observe that larger the z dimension gets, better the scores become in all languages. It may be due to simply providing the network with a larger representation space, allowing for a more flexible learning process. We have then used the trained model to sample surface forms for the given lemma. We have observed that, the model almost always generates a valid surface form when run backwards. However we have also noticed the diversity of the generated surface forms being quite low. This may be due to the statistical properties of the dataset, i.e., observing one dominant morphological tag combination throughout the trainig set. Second reason, is the problem similar to the one described in Bowman et al. (2016). This study uses Variational Autoencoders (VAE) to generate diverse sentences via the help of latent variable, z, that aims to encode useful global information hence enable generation of diverse sentences. However, they observe *KL-divergence* loss of zero that causes the model to ignore z.

Morphological Inflection: One of the most important findings for this task is the evidence of network's ability to provide remarkably strong results for both directions when $\mathcal{L}_y$+$\mathcal{L}_x$ are used. This suggests that, optimizing the same network parameters (θ) with loss functions of dual problems, is feasible and necessary for dual strong results. Although majority of the time, adding $\mathcal{L}_t$, slightly decreased the lemma recovery scores, in Irish and Turkish, we observe improvement on both scores, which may be due to relatively small number of training data. We see mixed results for tag F1 scores, due to varying morphological ambiguity in languages. For agglutinative languages, Finnish, Tagalog and Turkish, where each morpheme is associated with a tag, the ambiguity is lower, therefore F1 scores are higher; in contrast to fusional languages, Romanian and French. Interestingly, even when the network is only optimized for $\mathcal{L}_y$, INNs started to implicitly learn the morphological tags *for all languages*, demonstrated by the tag F1 scores ranging between 11.84-36.29. However no improvement had been observed for the lemma, suggesting that the distribution of the lemma is too complicated to learn implicitly, hence an explicit supervision is necessary. Finally, for all agglutinative languages, lemma exact match scores in *lemmatization* task are better than or very similar to the scores in the inverse task of *morphological analysis*; which can be explained by one-to-one morpheme to tag mapping, that is implicitly learned without any guidance.

Related Work

Morphological Tasks Most morphological problems dealt in this paper are also known as string-to-string transduction problems. Dreyer, Smith, and Eisner (2008) introduce a general modeling framework based on weighted finite state transducers (WFST) that employ n-gram features and latent variables. They perform experiments on morphological form generation and lemmatization and show that incorporating latent variables improves the results dramatically. Although they use the same framework for both problems, the models are trained separately for each problem. Furthermore, the framework learns the best alignment between the lemma and the inflected form, *separately* for each task, which may not be known beforehand in a realistic scenario. Schnober et al. (2016) compare more recent encoder-decoder architectures such as seq2seq with hard monotonic attention (Aharoni and Goldberg 2017) with traditional transduction techniques based on conditional random fields and WFSTs on classical string transduction tasks including lemmatization. They find that although traditional models have similar performance to neural models in most cases, their performances fall behind of the neural models for lemmatization task. This suggests that the lemmatization can be considered as a more complex/ambiguous problem compared to other tasks. Ribeiro et al. (2018) introduce a method to reduce the transduction to sequence labeling problem, and show that although neural methods perform on par or worse (e.g., on Finnish OCR) in most cases, it is the opposite for the morphological inflection task.

Invertibility in NLP He, Neubig, and Berg-Kirkpatrick (2018) employ invertible transformations (coupling layers that are very similar to the ones used in this study), to perform unsupervised learning of syntactic structure. They demonstrate the efficiency of the invertibility property on POS tagging and dependency parsing. Recently, Ziegler and Rush (2019) proposed a flow-based model for discrete sequences such as text, and show that their proposed autoregressive model performs on par with traditional sequential models on character-level language modeling task.

Joint models A few decades ago, the researchers have attempted to design a unified framework, i.e., a reversible, single grammar, for parsing and generation (Shieber 1988; Wintner, Gabrilovich, and Francez 2000). However, the grammar development has been replaced by advanced statistical tools; therefore those works have not been explored recently. Another set of models that are conceptually similar to ours, perform paired training (Konstas et al. 2017; Hu et al. 2017; Cao et al. 2019), however still optimize separate models and generally have a more complicated training procedure.

Conclusions

We have proposed modeling several inverse problems in morphology together with their dual problem, such as morphological analysis $\leftrightarrow$ inflection, with recently proposed invertible neural networks that uses a single model for both problems, offer efficient bi-directional training with the help of invertibility layers and provide free inverse mapping. We showed that they are capable of simultaneous optimization of such dual problems providing strong results on both; and lemmatization benefits from additional categorical latent variables. We demonstrated that simple distributions are implicitly learned while complex, multimodal distributions needed supervision. We hope that these initial encouraging results for inverse problems in morphology would inspire the researchers to explore other inverse problems of NLP.

Acknowledgments

This work has been supported by the German Research Foundation through the German-Israeli Project Cooperation (DIP, grant DA 1600/1-1 and grant GU 798/17-1). We would like to thank anonymous reviewers, Jonas Pfeiffer, Mohsen Mesgar, Ilia Kuznetsov and Yevgeniy Puzikov for their constructive feedback. We are grateful to Lynton Ardizzone, Jakob Kruse and Ullrich Köthe for the initial discussions and kindly sharing the INN implementation before the official release with us.

References

Adler, T. J.; Ardizzone, L.; Vemuri, A.; Ayala, L.; Gröhl, J.; Kirchner, T.; Wirkert, S.; Kruse, J.; Rother, C.; Köthe, U.; et al. 2019. Uncertainty-aware performance assessment of optical imaging modalities with invertible neural networks. *International journal of computer assisted radiology and surgery* 1–11.

Aharoni, R., and Goldberg, Y. 2017. Morphological inflection generation with hard monotonic attention. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, ACL 2017, Vancouver, Canada, July 30 - August 4, Volume 1: Long Papers*, 2004–2015.

Ardizzone, L.; Kruse, J.; Rother, C.; and Köthe, U. 2019. Analyzing inverse problems with invertible neural networks. In *International Conference on Learning Representations*.

Bertero, M., and Boccacci, P. 1998. *Introduction to inverse problems in imaging*. CRC press.

Bowman, S. R.; Vilnis, L.; Vinyals, O.; Dai, A. M.; Józefowicz, R.; and Bengio, S. 2016. Generating sentences from a continuous space. In *Proceedings of the 20th SIGNLL Conference on Computational Natural Language Learning, CoNLL 2016, Berlin, Germany, August 11-12, 2016*, 10–21.

Cao, R.; Zhu, S.; Liu, C.; Li, J.; and Yu, K. 2019. Semantic parsing with dual learning. In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, 51–64.

Dinh, L.; Krueger, D.; and Bengio, Y. 2014. Nice: Non-linear independent components estimation. *arXiv preprint arXiv:1410.8516*.

Dinh, L.; Sohl-Dickstein, J.; and Bengio, S. 2017. Density estimation using real NVP. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*.

Dreyer, M.; Smith, J.; and Eisner, J. 2008. Latent-variable modeling of string transductions with finite-state methods. In *2008 Conference on Empirical Methods in Natural Language Processing, EMNLP 2008, Proceedings of the Conference, 25-27 October 2008, Honolulu, Hawaii, USA, A meeting of SIGDAT, a Special Interest Group of the ACL*, 1080–1089.

Dreyer, M. 2011. *A Non-parametric Model for the Discovery of Inflectional Paradigms from Plain Text Using Graphical Models over Strings*. Ph.D. Dissertation, Baltimore, MD, USA. AAI3463374.

Grover, A.; Dhar, M.; and Ermon, S. 2018. Flow-gan: Combining maximum likelihood and adversarial learning in generative models. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), the 30th innovative Applications of Artificial Intelligence (IAAI-18), and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI-18), New Orleans, Louisiana, USA, February 2-7, 2018*, 3069–3076.

He, J.; Neubig, G.; and Berg-Kirkpatrick, T. 2018. Unsupervised learning of syntactic structure with invertible neural projections. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018*, 1292–1302.

Heinzerling, B., and Strube, M. 2018. BPEmb: Tokenization-free Pre-trained Subword Embeddings in 275 Languages. In chair), N. C. C.; Choukri, K.; Cieri, C.; Declerck, T.; Goggi, S.; Hasida, K.; Isahara, H.; Maegaard, B.; Mariani, J.; Mazo, H.; Moreno, A.; Odijk, J.; Piperidis, S.; and Tokunaga, T., eds., *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*. Miyazaki, Japan: European Language Resources Association (ELRA).

Hu, Z.; Yang, Z.; Liang, X.; Salakhutdinov, R.; and Xing, E. P. 2017. Toward controlled generation of text. In *Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017*, 1587–1596.

Jang, E.; Gu, S.; and Poole, B. 2017. Categorical reparameterization with gumbel-softmax. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*.

Kabanikhin, S. I. 2008. Definitions and examples of inverse and ill-posed problems. *Journal of Inverse and Ill-Posed Problems* 16(4):317–357.

Kingma, D. P., and Dhariwal, P. 2018. Glow: Generative flow with invertible 1x1 convolutions. In *Advances in Neural Information Processing Systems*, 10215–10224.

Konstas, I.; Iyer, S.; Yatskar, M.; Choi, Y.; and Zettlemoyer, L. 2017. Neural AMR: sequence-to-sequence models for parsing and generation. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, ACL 2017, Vancouver, Canada, July 30 - August 4, Volume 1: Long Papers*, 146–157.

Osborne, C. M.; Armstrong, J. A.; and Fletcher, L. 2019. Radynversion: Learning to invert a solar flare atmosphere with invertible neural networks. *The Astrophysical Journal* 873(2):128.

Rastogi, P.; Cotterell, R.; and Eisner, J. 2016. Weighting finite-state transductions with neural context. In *Proceedings of NAACL*.

Řehůřek, R., and Sojka, P. 2010. Software Framework for Topic Modelling with Large Corpora. In *Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks*, 45–50. Valletta, Malta: ELRA. http://is.muni.cz/publication/884893/en.

Ribeiro, J.; Narayan, S.; Cohen, S. B.; and Carreras, X. 2018. Local string transduction as sequence labeling. In *Proceedings of the 27th International Conference on Computational Linguistics, COLING 2018, Santa Fe, New Mexico, USA, August 20-26, 2018*, 1360–1371.

Schnober, C.; Eger, S.; Dinh, E. D.; and Gurevych, I. 2016. Still not there? comparing traditional sequence-to-sequence models to encoder-decoder neural networks on monotone string translation tasks. In *COLING 2016, 26th International Conference on Computational Linguistics, Proceedings of the Conference: Technical Papers, December 11-16, 2016, Osaka, Japan*, 1703–1714.

Sennrich, R.; Haddow, B.; and Birch, A. 2016. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, ACL 2016, August 7-12, 2016, Berlin, Germany, Volume 1: Long Papers*.

Shieber, S. M. 1988. A uniform architecture for parsing and generation. In *Proceedings of the 12th International Conference on Computational Linguistics, COLING '88, Budapest, Hungary, August 22-27, 1988*, 614–619.

Snieder, R., and Trampert, J. 1999. Inverse problems in geophysics. In *Wavefield inversion*. Springer. 119–190.

Teng, Y., and Choromanska, A. 2019. Invertible autoencoder for domain adaptation. *Computation* 7(2):20.

Wicentowski, R. H. 2003. *Modeling and Learning Multilingual Inflectional Morphology in a Minimally Supervised Framework*. Ph.D. Dissertation. AAI3068229.

Wintner, S.; Gabrilovich, E.; and Francez, N. 2000. Amalia: A unified platform for parsing and generation. *AMSTERDAM STUDIES IN THE THEORY AND HISTORY OF LINGUISTIC SCIENCE SERIES 4* 257–270.

Ziegler, Z. M., and Rush, A. M. 2019. Latent normalizing flows for discrete sequences. *arXiv preprint arXiv:1901.10548*.