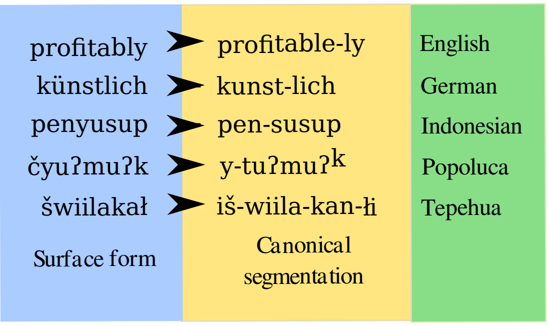
Tackling the Low-resource Challenge for Canonical Segmentation

Manuel Mager[1], **Özlem Çetinoğlu**[1] and **Katharina Kann**[2]
[1]Institute for Natural Language Processing,
University of Stuttgart, Germany
[2]University of Colorado Boulder, USA
{manuel.mager, ozlem}@ims.uni-stuttgart.de,
katharina.kann@colorado.edu

Abstract

Canonical morphological segmentation consists of dividing words into their standardized morphemes. Here, we are interested in approaches for the task when training data is limited. We compare model performance in a simulated low-resource setting for the high-resource languages German, English, and Indonesian to experiments on new datasets for the truly low-resource languages Popoluca and Tepehua. We explore two new models for the task, borrowing from the closely related area of morphological generation: an LSTM pointer-generator and a sequence-to-sequence model with hard monotonic attention trained with imitation learning. We find that, in the low-resource setting, the novel approaches outperform existing ones on all languages by up to 11.4% accuracy. However, while accuracy in *emulated* low-resource scenarios is over 50% for all languages, for the *truly* low-resource languages Popoluca and Tepehua, our best model only obtains 37.4% and 28.4% accuracy, respectively. Thus, we conclude that canonical segmentation is still a challenging task for low-resource languages.



Surface form		Canonical segmentation	
profitably	➤	profitable-ly	English
künstlich	➤	kunst-lich	German
penyusup	➤	pen-susup	Indonesian
čyuʔmuʔk	➤	y-tuʔmuʔ^{k}	Popoluca
šwiilakał	➤	iš-wiila-kan-łi	Tepehua

Figure 1: Canonical segmentation examples for all languages in our experiments.

1 Introduction

Morphological segmentation denotes the task of dividing words into their constituting morphemes, i.e., their smallest meaning-bearing units, and has been studied extensively in natural language processing (NLP) (Ruokolainen et al., 2016). The most common form of segmentation consists of separating morphemes at the surface level. However, this is not always well suited: in fusional languages, morphemes are merged during word formation and, thereby, change their surface forms. Thus, in this paper, we tackle the task of canonical segmentation (Cotterell et al., 2016b), which consists of segmenting a word while restoring the original forms of its morphemes. Considering, e.g., the English word *collision*, its surface segmentation is *collis+ion*, while its canonical segmentation is *collide+ion*. Figure 1 provides examples for all five languages we experiment on.

Neural models have shown to perform well on this task when large amounts of training data are available (Kann et al., 2016; Ruzsics and Samardzic, 2017). Nevertheless, datasets with morphological annotations are difficult to obtain, since they require expert annotators. Furthermore, many languages with complex morphology are spoken by a limited number of people or are listed as endangered languages (Mager et al., 2018), which reduces the possible annotator pool even more. However, morphological segmentation is important for downstream tasks like machine translation (Conforti et al., 2018; Vania and Lopez, 2017), dependency parsing (Seeker and Çetinoğlu, 2015; Vania et al., 2018), or semantic role labeling (Sahin and Steedman, 2018). Moreover, high performance on these tasks can yield more language independent NLP models (Gerz et al., 2018).

Here, we focus on low-resource canonical segmentation. We propose two new models for the task, which have recently been successfully applied to a related morphological generation task called *morphological inflection*. The approaches we in-

Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, pages 5237–5250,
November 16–20, 2020. ©2020 Association for Computational Linguistics

vestigate are (i) an LSTM pointer-generator model (Sharma et al., 2018a), and (ii) a neural transducer trained with imitation learning (IL; Makarov and Clematide, 2018a). Since both canonical segmentation and morphological inflection are character-level string transduction tasks, we hypothesize that models which can learn one from limited data, will also be able to do so for the other.

We experiment on three benchmark datasets in German, English, and Indonesian, but simulate a low-resource scenario by reducing the number of training examples. We further evaluate our models on datasets for two *truly* low-resource languages: Popoluca and Tepehua. We find that our new models indeed outperform previous approaches on all languages. For additional insight, we also evaluate the performance of all models for varying amounts of training data from the high-resource languages and find that the neural-transducer with imitation learning outperforms all other models in all but one setting with up to 600 training examples. Using the entire training set for English, German, and Indonesian, the state-of-the-art LSTM sequence-to-sequence model performs best. However, the difference to our proposed models is below 3.3% accuracy for all languages and models.

Contributions. (i) Inspired by recent advances in the area of morphological generation, we propose two new models for the task of low-resource canonical segmentation, which outperform all baselines. (ii) We introduce two canonical segmentation datasets for the *truly* low-resource languages Popoluca and Tepehua. (iii) We compare all models under multiple different conditions, highlighting their strengths and shortcomings, and conduct an analysis of the errors made by all neural models.

2 Related Work

The task of morphological segmentation was introduced by Harris (1951). Most work has considered the surface segmentation task, for which unsupervised methods like LINGUISTICA (Goldsmith, 2001) and MORFESSOR (Creutz and Lagus, 2002, 2007; Poon et al., 2009) played an important role. The latter was further extended to a semi-supervised version (Kohonen et al., 2010; Grönroos et al., 2014).

Over the last years, supervised methods have attracted more attention: Ruokolainen et al. (2013) cast the task as a sequence labeling problem using conditional random fields (CRFs; Lafferty et al., 2001). A similar approach was suggested by Wang et al. (2016), who employed a long short-term memory network (LSTM; Hochreiter and Schmidhuber, 1997) for tagging. Semi-Markov CRFs were also proposed (Cotterell et al., 2015). Kann et al. (2018) modeled the task as a sequence-to-sequence problem. Supervised methods for surface segmentation were shown to perform acceptably even in the low-resource setting (Grönroos et al., 2019). Recent work also included context to improve morphological disambiguation (Can and Manandhar, 2018; Sakakini et al., 2017). Yang et al. (2019) proposed a pointer network to find surface segmentation boundaries.

For fusional languages, surface segmentation is not very effective. Therefore, restoring morphemes to their canonical form was previously discussed in linguistics (Kay, 1977) as well as in the NLP literature. Previous approaches include unsupervised (Naradowsky and Goldwater, 2009), as well as joint models for segmentation and transduction (Cotterell et al., 2016b) and neural encoder-decoder models (Kann et al., 2016; Ruzsics and Samardzic, 2017). However, up to now, supervised models have only been explored in the high-resource setting. We aim at closing this gap.

For low-resource morphological segmentation, rule-based approaches have been used frequently, since they do not need large amounts of data. They have been developed, e.g., with finite state transducer (FST) tools like FOMA (Hulden, 2009) or HFST (Lindén et al., 2011). However, this kind of system requires both time and linguistic knowledge. Our aim is to explore data-driven approaches for the low-resource setting in order to overcome this limitation.

In recent years, the area of morphological generation has experienced substantial progress, with a variety of methods that can be used for the canonical segmentation task. Kann et al. (2016) used a sequence-to-sequence model to inflect a word given a set of morphological tags. Sharma et al. (2018a) proposed a pointer-generator model, which was more suitable for the low-resource setting. Aharoni and Goldberg (2017) proposed a neural transducer with hard monotonic attention. Makarov et al. (2017) extended this approach and added a copy operation, and Makarov and Clematide (2018a) proposed imitation learning (Daumé et al., 2009) for training it. Here, we explore the applicability of the models by Sharma et al. (2018a) and Makarov

ENG		DEU		IND		POQ		TTP	
ly	7.53	er	15.66	men	8.65	y	6.08	ya	8.58
ness	3.41	in	10.38	nya	8.29	∅	6.08	łi	6.27
er	2.99	ung	8.14	an	7.18	n	3.98	ka	4.46
ion	1.87	lich	4.37	kan	6.61	ny	3.56	ta	4.29
y	1.50	keit	3.96	di	5.31	k	3.35	ti	3.80
ity	1.24	ig	3.78	pen	4.14	p	2.94	ik	2.81
ation	0.99	los	1.23	ber	2.81	t+k	2.52	ni	2.64
un	0.88	chen	1.16	i	2.45	ky	2.31	ča	2.31
ic	0.85	bar	1.13	ter	1.91	wat	2.10	la	1.82
al	0.81	ver	0.81	per	1.25	aʔ	2.10	maa	1.82
ist	0.76	un	0.77	se	0.72	taʔ	1.89	kin	1.82
able	0.74	e	0.49	ke	0.71	ʔeš	1.26	waa	1.82

Table 1: Relative frequencies of the 12 most common morphemes for each language; ENG=English; DEU=German; IND=Indonesian; POQ=Popoluca; TTP=Tepehua.

	>3Morph.	Surf.	Canon.	NoSeg.	M./W.	Ch./W.
ENG	00.01	36.40	22.83	41.37	01.60	08.18
DEU	01.86	46.07	53.86	00.00	02.20	12.48
IND	05.57	46.21	23.66	30.14	02.07	08.65
POQ	12.12	23.74	56.57	19.70	02.41	06.78
TTP	32.00	21.50	63.00	15.50	03.03	08.62

Table 2: Statistics for all five canonical segmentation datasets. Percentages of words with more than 3 morphemes (>3 Morph.), surface segmentation (Surf.), canonical segmentation (Canon.), and without segmentation (NoSeg.), as well as the average number of morphemes per word (M./W.) and characters per word (Ch./W.).

and Clematide (2018a) to low-resource canonical segmentation.

3 Datasets for Popoluca and Tepehua

We release two new datasets for low-resource canonical segmentation in Popoluca and Tepehua[1]. In this section, we briefly introduce the languages, before describing our datasets. We use these two languages to shed light on polysynthetic languages that also exhibit fusional phenomena. The high-resource datasets introduced by (Cotterell et al., 2016a) cover fusional (German), analytic (English), and agglutinative (Indonesian) languages.

3.1 Languages

In addition to experimenting on high-resource datasets for English, German and Indonesian (Cotterell et al., 2016b), we introduce datasets for two low-resource languages from Mexico: Popoluca and Tepehua. This enables us to evaluate our models in real low-resource settings.

Popoluca. Popoluca of Texistepec (language code: POQ[2]) is part of the Mixe-Zoquean family. Its morphology is classified as polysynthetic, and it mostly follows a verb, subject, object (VSO) word order (Dryer and Haspelmath, 2013). This language is almost extinct with only one native speaker alive reported in 2005 (Gordon Jr, 2005). However, attempts for language revival have been reported (INEGI, 2008). Efforts made for language revitalization can benefit from advances in NLP. Thus, the creation and development of accurate models for those languages is of high importance.

Here we show an example of canonical segmentation in Popoluca, together with its English gloss. The plus symbol is part of the alphabet of the language. We use a '-' as morpheme delimiter.

kki:mba: → ky-k+:m-ba:
You are small

Tepehua. Tepehua (language code: TPP) belongs to the Totonacan language family. It is spoken in three Mexican regions: in the northeastern part of the state of Hidalgo (around 3000 speakers), in the villages of Pisaflores (around 4000 speakers), and in Tlachichilco in the state of Veracruz (around 3000 speakers) (Gordon Jr, 2005). It is also polysynthetic. Tepehua permits free word order, but has a preference for a subject, verb, object (SVO) configuration (Dryer and Haspelmath, 2013).

An example for canonical segmentation is

iklakadíkdi → ik-laka-tikti
I am small

The variant of the language used in our dataset is the one spoken in Pisaflores, Veracruz.

3.2 Datasets

We collect words for our datasets from two books belonging to the Archive of Indigenous Languages (ALI-Colmex) of the College of Mexico (*Colegio de México*). For Popoluca we used the book by Wichmann (2007) and for Tepehua that by MacKay and Trechsel (2010). We include segmentable as well as non-segmentable words in order to avoid oversegmentation by our systems. For both languages a set of Spanish sentences are used to elicit the data. This set of sentences is the same across the entire ALI-Colmex collection. For each language

[1] Te dataset is available at `http://turing.iimas.unam.mx/wix/canseg`

[2] We use the languages codes defined in the ISO 639-3 standard.

the authors of the books asked native speakers to translate the sentences into the respective languages (elicited data). Afterwards, they performed a glossing of the translated text. For more details we refer the reader to the original books.

In Table 2, we show statistics for all five datasets used in this paper. Importantly, the German dataset only contains multi-morpheme words. Additionally, we observe that most of the Indonesian words only require surface segmentation, while English is the language with the highest ratio of words that do not require any segmentation. On the other hand, Popoluca and Tepehua have the highest proportion of words that require both splitting and restoration of the canonical forms. Moreover, both languages have a high amount of words that contain more than 3 morphemes per word, and also have the highest morphemes-per-word rate. Adding to these facts, the small amount of data available for these languages makes morphological segmentation even harder. To get a better understanding of the underlying morphemes seen in each language, we extract the 15 most common ones for each dataset. These morphemes, together with their relative frequency in our datasets, are shown in Table 1.

4 Models

Inspired by recent successes of two models for low-resource morphological inflection, we propose to apply these architectures to canonical segmentation with limited training data. In this section, we introduce the models.

4.1 Pointer-Generator Network

Motivation. The first model we apply to low-resource canonical segmentation is a pointer-generator network (See et al., 2017), i.e., a sequence-to-sequence model with a mechanism to copy input elements over to the output. Our intuition is that this should make the learning problem easier and help in settings with limited training data. The pointer-generator network can be considered a hybrid between an attention-based sequence-to-sequence model (Bahdanau et al., 2015) and a pointer network (Vinyals et al., 2015).

Model description. Our pointer-generator network consists of a bidirectional LSTM (Hochreiter and Schmidhuber, 1997) encoder and a unidirectional LSTM decoder with an attention mechanism. We cast the task of canonical segmentation as a character-based sequence-to-sequence problem, with the characters of the original word as the input and the characters of the restored morphemes in combination with segment boundary markers as the output. Both our encoder and decoder operate on the character level.

The pointer-generator network differs from the standard sequence-to-sequence architecture in that the decoder calculates a probability for copying an element from the input over to the output instead of generating. Here, we follow Sharma et al. (2018b) and use two separate encoders: one for the lemma and one for the morphological tags. The decoder then computes the probability distribution of the output at each time step as a weighted sum of the probability distribution over the output vocabulary and the attention distribution over the input characters. The weights can be seen as the probability to generate or copy, respectively, and are computed by a feedforward network. For details, we refer the reader to Sharma et al. (2018b).

Hyperparameters. All encoder and decoder hidden states are 100-dimensional, and our embeddings are of size 100. For training, we use Adam (Kingma and Ba, 2014) with a learning rate of 0.001 and a mini-batch size of 32. To avoid overfitting, we use dropout (Srivastava et al., 2014) with a coefficient of 0.3 for the high-resource setting and 0.5 for the low-resource setting. We train our model for 100 and 300 epochs and use early stopping with a patience of 10 and 100 for the high-resource and the low-resource setting, respectively.

4.2 Neural Transducer with Imitation Learning

Motivation. Hard monotonic attention networks (Aharoni and Goldberg, 2017) have shown to perform well on morphological generation in the low-resource setting. These systems use a nearly-monotonic alignment between the source characters and the output characters. For our second model, we employ the variant proposed by Makarov and Clematide (2018c), which makes use of imitation learning for end-to-end training and, thus, avoids error propagation.

Model description. This model is a sequence-to-sequence model with hard monotonic attention (Aharoni and Goldberg, 2017), which transduces an input sequence of characters into an output sequence by performing edit operations. Following Makarov and Clematide (2018b), it can perform three operations: insertion, deletion and copy. However, instead of using maximum likelihood estimation (MLE), training is done with imitation learning. The idea is to train a model to imitate an expert policy that maps the training configurations to a set of optimal actions. We aim to minimize the sequence-level loss and an action level loss.

The training is composed of two steps: a roll-in and a roll-out stage. In the roll-in stage, the model gather actions by sampling from the expert policy. This process returns a set of decoder outputs called configurations. For the roll-out stage: a sequence-level loss is computed for each valid action per configuration. For that, the action is executed and is compared to the optimal action sequence of the expert. This loss is defined in terms of Levenshtein distance (Levenshtein, 1966) between the prediction and the target and the cost of the actions. The cost function uses the information from a character aligner. After calculating the sentence-level loss, this is fed into an action-level loss. This loss expresses how much a certain action suffers relative to the optimal action under the current policy. This is done by minimizing the negative marginal log-likelihood of all optimal actions (Makarov and Clematide, 2018b).

Hyperparameters. For the encoder and the decoder of this model, we use one layer with a 200-dimensional size, with a dropout of 0.5. For optimization we use ADADELTA (Zeiler, 2012) with a learning rate of 0.1. As the RNN unit, we use an LSTM. We train the model for 30 epochs, with a patience of 10 epochs. For IL training, we use an inverse sigmoid, and a decay rate of 12. For decoding, we employ beam search with a beam of width 4.

5 Experiments

We now describe the experiments we conduct to explore the performance of our models both in the high-resource and in the low-resource setting.

5.1 Data

The canonical segmentation datasets for English (ENG), German (DEU) and Indonesian (IND) by Cotterell et al. (2016b) each consist of 8000 training, 1000 development, and 1000 test examples. We consider the complete training set to be high-resource. The datasets feature a splitting into 10 folds for cross-validation. For our low-resource experiments, we randomly take a subset of words from each training fold, but keep the development and test sets unchanged.

The high-resource datasets cover three languages: English, German, and Indonesian. English is an analytic language from the Indo-European family (Konig and Van der Auwera, 2013), German exhibits fusional typology (Hawkins, 2015), while Indonesian is an agglutinative language whose morphology involves the use of affixation, reduplication and cliticization (Hiroki Nomoto and Bond, 2018).

We additionally experiment with two polysynthetic low-resource languages: Tepehua and Popoluca (cf. Section 2). As those datasets are small (900 words for each language), we divide the datasets into 9 folds, each containing 100 training, 100 development, and 700 test examples.

5.2 Baselines

We compare the neural-transducer with imitation-learning (`IL`) and the pointer-generator network (`PGNet`) to three strong baselines, including the current state of the art for the canonical segmentation task.

Encoder-Decoder (`s2s`). Our first baseline is a character-based encoder-decoder recurrent neural network (RNN) architecture with attention as proposed by Kann et al. (2016). It defines (in combination with a reranker which we omit here since it is orthogonal to our work) the state of the art on the high-resource datasets. To perform experiments in the low-resource setting, we re-implement this model using OpenNMT (Klein et al., 2017). The hyperparameters suggested by Kann et al. (2016) are as follows: the RNNs of the encoder and decoder have 100 hidden units each; the embedding size is 300. For optimization we use ADADELTA (Zeiler, 2012) with a minibatch size of 20.

Semi-Markov CRF (`semiCRF`). Our first non-neural baseline is the ChipMunk (Cotterell et al., 2015) implementation of a semi-Markov CRF (Sarawagi and Cohen, 2005). Although the system is able to make use of additional complementary information like morphological tags or dictionaries, we decide to not include those, in order to make our results comparable across all languages and systems.

Joint log-linear model (`joint`) As a second non-neural system we use a log-linear model which

	English			German			Indonesian		
	Acc.	ED	F1	Acc.	ED	F1	Acc.	ED	F1
SemiCRF	64.7	64.3	76.6	41.9	108.3	74.1	70.4	46.3	84.3
joint	72.0	98.0	76.0	59.0	101.0	76.0	90.0	15.0	80.0
s2s	♦**78.0**	**41.2**	88.4	♦**77.1**	**47.8**	**89.3**	♦**94.3**	**7.6**	**97.9**
PGNet	77.5	42.4	**88.5**	74.8	52.1	88.2	92.9	10.0	97.5
IL	76.7	42.9	87.2	73.8	52.3	87.2	93.4	8.4	97.6

Table 3: Results for semiCRF, joint, s2s, PGNet, and IL for the high-resource setting of English, German and Indonesian. Lower scores in the ED columns are better. For accuracy, ♦ indicates statistical significance at $p < .01$.

jointly segments and generates underlying representations of the input words (Cotterell et al., 2016b). For segmentation it uses the semiCRF previously described, and for transduction of the underlying forms it uses a probabilistic final state transducer (Cotterell et al., 2014).

5.3 Training Details

We choose the hyperparameters for all models following the mentioned previous work. All neural models and the semiCRF were trained on a server with 2 Intel(R) Xeon(R) CPU v4@ 2.20GHz, with 4 Nvidia GTX 1080ti graphic cards. To train the joint log-linear model a MacBook Pro 2009 laptop was used. Links to the repositories we use are listed in the complementary material.

5.4 Metrics

For evaluation, we use three metrics. The first one is **accuracy**, i.e., the proportion of entirely correctly segmented words, to get a better understanding of partially right segmentation. To get more information about subword-level errors, we also employ **edit distance** on the character level. This is particularly useful to penalize big mistakes in a single word. We also use **F_1 score** on the morpheme level, to measure the overlap between morphemes. Precision corresponds to the proportion of morphemes in the prediction that occur in the gold standard, and recall is the proportion of morphemes in gold that appear in the system's prediction. This will ensure that morphemes that are predicted without appearing in the gold standard are penalized, as well morphemes that are in the gold standard but are omitted in the prediction.

5.5 Results

Low-resource *simulation*. Figure 2 shows the accuracy of all systems for different low-resource training set sizes (100, 200, 300, 400, 500 and 600 examples) for English, German, and Indonesian. To ensure statistical significance we use McNemar's test (McNemar, 1947) for all accuracy results (Tables 3 and 4, Figure 2) comparing the best and the second best systems. All results are significant at $p < 0.01$. The scores of all systems vary across languages. However, IL consistently is among the two best systems in terms of accuracy in all settings. For 100 training examples IL is the second-best performing system with 50.99% for English, just behind the semiCRF with 52.87%. For German 51.49% IL slightly outperforms Joint (51.33%) and obtains the best score for Indonesian with 61.14%, where the second best system is semiCRF (58.82%). Moreover, from 300 examples up to 600, IL strongly outperforms all other systems, including non-neural ones.

If we compare the performance of our two proposed systems with s2s, PGNet strongly outperforms s2s with improvements of 22.27%, 17.84%, and 25.66% absolute accuracy for English, German, and Indonesian, respectively, in the setting with 100 training examples; while IL have even bigger gains with improvements of 30.65%, 32.1% and 35.73% of accuracy respectively.

Looking on the learning curves for each model for increasing training set sizes, we can see that both proposed systems show monotonically increasing performance: they take advantage of more data well, but still achieve decent performance in the low-resource setting, even outperforming all non-neural systems in some settings. On the contrary, the non-neural models joint and semiCRF have in many cases a good start, but only benefit to a limited extends from additional data. A table listing all individual results for this experiment is included in the supplementary material.

	Tepehua			Popoluca		
Model	Acc.	ED	F1	Acc.	ED	F1
`SemiCRF`	21.9	285.3	35.9	26.0	215.0	41.4
`joint`	11.2	335.4	29.5	14.6	393.6	36.8
`s2s`	4.1	532.4	7.7	13.2	309.4	23.3
`PGNet`	17.2	321.7	29.3	27.0	211.0	42.5
`IL`	♦**28.4**	**242.6**	**44.0**	♦**37.4**	**158.8**	**54.7**

Table 4: Results for the low-resource languages Popoluca and Tepehua. For accuracy, ♦ indicates statistical significance at $p < .01$.

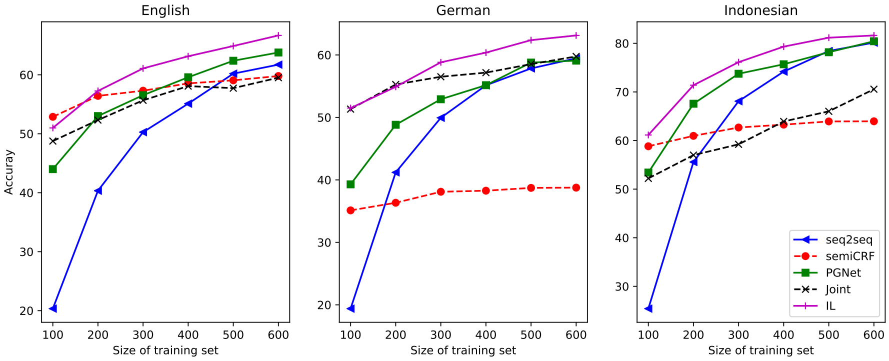


Figure 2: Accuracy for different simulated low-resource settings for our high-resource languages.

Low-resource *languages*. Results for Popoluca and Tepehua are shown in Table 4 and confirm most of the tendencies seen in our low-resource simulation experiment. `s2s` barely predicts any correct segmentation for Tepehua, and only obtains 4.14% absolute accuracy and 13.19 F_1 score. Similarly, for Popoluca, `s2s` reaches only 13.19% accuracy. The performance of `IL` is consistently better on all metrics, with substantial gains for Tepehua of 6.5% accuracy over the closest system (`semiCRF`) and 10.4% accuracy over `PGNet`.

The performance of `PGNet` is consistently better than that of `s2s`, with gains of 13.03% and 13.77% accuracy for Tepehua and Popoluca, respectively. `joint` surprisingly shows a low performance for our two low-resource languages, obtaining a 17.2% lower accuracy than the best model for Tepehua (`IL`), and a 22.8% lower accuracy than the best system for Popoluca (`IL`).

Overall, all systems perform notably worse for Tepehua and Popoluca than for the high-resource languages. This could be due to their high morphological complexity, as shown in Table 2.

High-resource setting. Table 3 shows results for `IL`, `PGNet`, `s2s`, `joint`, and `semiCRF` for the high-resource experiment. The `s2s` model gets the best results in this setting with 78.02%, 77.06%, and 94.30% accuracy for English, German, and Indonesian, respectively. However, it only obtains a slightly higher accuracy than `PGNet` and the differences in F_1 scores are similarly small. Overall, the pointer-generator network achieves results that are comparable with the state of the art in the high-resource setting. In contrast to the good performance for low-resource settings, `IL` under-performs on all metrics compared to `s2s` and `PGNet`. The `joint` model is the best non-neural system and performs clearly worse than both neural systems. Compared to `PGNet`, its accuracy is 5.54% lower for English, 15.80% lower for German, and 2.90% lower for Indonesian. `semiCRF` performs even worse.

6 Error Analysis

To get a better understanding of the results obtained with our neural models, we perform an error analysis on the output for the development sets of all folds. By manual inspection, we identify five not mutually exclusive types of errors: **Oversegmentation (Overseg.)** arises when the number of morpheme boundaries in the prediction is higher than in the gold standard annotation. **Undersegmentation (Underseg.)** occurs when the number of morpheme boundaries is lower than in the gold

Oversegmentation	
Input	internationalisierung
Gold	internationale isier ung
Error	internationale *is i er* ung
Description	The morpheme `isier` is segmented wrongly into three morphemes.
Undersegmentation	
Input	internationalisierung
Gold	internationale isier ung
Error	internationale *isieung*
Description	The morphemes `isier` and `ung` are lacking of a segmentation boundary.
Restoration Error	
Input	internationalisierung
Gold	internationale isier ung
Error	*international* isier ung
Description	The system did not perform the needed restoration for the stem `internationale`.
Overrestoration	
Input	internationalisierung
Gold	internationale isier ung
Error	internationaler *isierer* ung
Description	The systems perfomed a restoration on a morpheme that is not supposed to be restored.
Wrong segmentation	
Input	internationalisierung
Gold	internationale isier ung
Error	internationale *isi erung*
Description	The segmentation was done with the exact number of morphemes as in gold, however, the segmentation points are wrongly placed. In this error count all instances that do not match the exact segmentation boundaries.

Table 5: Examples of error types. Wrong parts are marked in italics.

standard. **Restoration error (Res.)** occurs when the prediction does not match the gold annotation, and the predicted word without boundaries does not match the input. These are errors that occur to words that undergo orthographic changes during word-formation. **Overrestoration (Overres.)** refers to outputs with errors where the correct output needs only segmentation and a copy of the input to the output. **Wrong segmentation (Wrong seg.)** arises when the morpheme boundaries in the prediction are not the same as in gold. From each segmented word, we extract the indices within the word where the segmentation is performed. If the segmentation indices from the gold standard and the prediction are not equal, it counts as this error.

Table 6 shows the percentage of errors in all languages for both experimental settings (100 examples in the low-resource setting). For the high-resource experiments, the results for oversegmentation and undersegmentation errors are mixed: for English, `s2s` avoids to generate too many segmentation boundaries, but this also has the drawback of not segmenting sufficient when it is needed. The opposite happens for German, where `IL` performs better as well, with respect to oversegmentation but fails regarding undersegmentation. `PGNet` shows no strong wins or problems regarding these errors, except for English, where it performs better for undersegmentation. `s2s` performs better for restoration errors with the exception of English, where again `PGNet` improves. With respect to oversegmentation errors, `IL` wins on all languages when compared to the other neural systems. As Indonesian has a relatively regular morphology, all error types are much less frequent for this language. If we only consider the exact segmentation point prediction, `s2s` performs better for all languages. However, the differences between the observed error rates are relatively small between `s2s` and `PGNet` models. Overall, wrong segmentation errors are the most common error type for all languages in the high-resource setting.

In the low-resources experiments, `IL` excels over all other models for oversegmentation and overrestoration, and for all languages with except to Indonesian for wrong segmentation errors. This low error rate explains the important gains that this model shows for low-resource languages. `PGNet` shows, however, better performance avoiding undersegmentation errors in all languages. It also performs better for Popoluca and Tepehua for restoration errors, while `s2s` has the lowest restoration errors for English, German, and Indonesian.

Finally, we also perform an error analysis of `joint` (cf. supplementary material). In our low-resource simulation experiments, we notice a surprisingly good performance of `joint` for German.

		Overseg.			Underseg.			Res.			Overres.			Wrong seg.		
		`IL`	`PGNet`	`s2s`	`IL`	`PGNet`	`s2s`	`IL`	`PGNet`	`s2s`	`IL`	`PGNet`	`s2s`	`IL`	`PGNet`	`s2s`
High	ENG	5.54	05.60	**05.28**	08.13	**06.58**	07.37	08.04	**05.86**	06.28	**02.34**	05.30	04.00	21.67	19.68	**17.01**
	DEU	**4.17**	04.42	04.88	09.30	08.11	**07.02**	09.24	07.42	**06.94**	**06.08**	07.55	06.40	25.46	23.65	**20.49**
	IND	2.26	02.52	**01.91**	01.76	01.67	**01.50**	00.46	00.52	**00.45**	**00.58**	01.22	00.79	05.16	05.29	**03.26**
Low	ENG	**5.84**	07.52	11.97	26.06	**18.82**	21.75	18.94	10.39	**04.96**	**02.56**	20.48	49.43	**46.92**	48.35	70.19
	DEU	**1.40**	04.11	07.79	17.56	**14.83**	15.70	32.01	16.26	**07.81**	**03.94**	21.88	33.93	**41.66**	51.52	71.78
	IND	**10.94**	11.03	15.47	15.24	**10.61**	14.00	4.91	03.19	**01.46**	**02.96**	19.90	50.00	34.64	**34.25**	36.16
	TPP	**15.86**	27.75	34.45	42.43	**07.56**	23.42	32.16	**07.58**	14.10	**03.80**	25.04	44.20	**69.52**	73.39	86.34
	POQ	**15.86**	21.88	26.10	28.43	**10.22**	25.18	22.86	**10.29**	17.68	**07.86**	22.17	49.42	**55.71**	57.54	76.81

Table 6: Error types found in the development set. The high resource configuration includes three languages, while the low-resourced setting refers to model performance using 100 training examples. This error analysis was done for all five languages.

The data for this language is special since all words contained in the set are segmentable. We find that `joint` has no undersegmentation errors at all. Also, it makes very few copy errors (9.5%, compared to 21.7% of `PGNet`). For our new datasets, this model obtains a high rate of wrong segmentation (88.87% for Popoluca and 91.57% for Tepehua). It further seems to not easily be able to decide which words should or should not be segmented. This is shown by the high undersegmentation rate (50.14% for Popoluca and 62.43% for Tepehua). Thus, the low performance of `joint` on those languages can be explained by this error type and the high morphemes-per-word rate of those languages as shown in Table 2.

7 Conclusion

We proposed two new models for the task of canonical segmentation in the low-resource setting: an LSTM pointer-generator model and a neural transducer trained with imitation learning. We evaluated the performance of both models against multiple state-of-the-art baselines on five languages of different morphological typology: English, German, Indonesian, Tepehua, and Popoluca. In emulated low-resource settings with up to 600 training examples, our best proposed model outperformed all baselines in all but one setting. We obtained a similar picture for experiments on the truly low-resource languages Popoluca and Tepehua: our best approach outperformed the best baseline by 11.4% and 6.5% accuracy. For large training sets, our systems performed close to the state of the art. However, we find a large gap between the *emulated* and the *real* low-resource scenarios: while accuracy is above 50% for all high-resource languages even with reduced amounts of training data, for Popoluca and Tepehua, our best model only obtains 37.4% and 28.4% accuracy, respectively.

Acknowledgments

We would like to thank Hilaria Cruz, Jaime Gonzalez and Ekta Sood for their valuable feedback. We also want to thank all the anonymous reviewers for their valuable comments and suggestions. This project has benefited from financial support to Manuel Mager and Özlem Çetinoğlu by DFG via project CE 326/1-1 "Computational Structural Analysis of German-Turkish Code-Switching" and to Manuel Mager by DAAD Doctoral Research Grant.

References

Roee Aharoni and Yoav Goldberg. 2017. Morphological inflection generation with hard monotonic attention. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2004–2015, Vancouver, Canada. Association for Computational Linguistics.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. *ICLR*.

Burcu Can and Suresh Manandhar. 2018. Tree structured dirichlet processes for hierarchical morphological segmentation. *Computational Linguistics*, 44(2):349–374.

Costanza Conforti, Matthias Huck, and Alexander Fraser. 2018. Neural morphological tagging of lemma sequences for machine translation. In *AMTA*, volume 1, pages 39–53.

Ryan Cotterell, Thomas Müller, Alexander Fraser, and Hinrich Schütze. 2015. Labeled morphological segmentation with semi-markov models. In *Proceedings of the Nineteenth Conference on Computational Natural Language Learning*, pages 164–174.

Ryan Cotterell, Nanyun Peng, and Jason Eisner. 2014. Stochastic contextual edit distance and probabilistic fsts. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 625–630.

Ryan Cotterell, Hinrich Schütze, and Jason Eisner. 2016a. Morphological smoothing and extrapolation of word embeddings. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1651–1660, Berlin, Germany. Association for Computational Linguistics.

Ryan Cotterell, Tim Vieira, and Hinrich Schütze. 2016b. A joint model of orthography and morphological segmentation. In *NAACL-HLT*, pages 664–669.

Mathias Creutz and Krista Lagus. 2002. Unsupervised discovery of morphemes. In *CoNLL–SIGMORPHON*.

Mathias Creutz and Krista Lagus. 2007. Unsupervised models for morpheme segmentation and morphology learning. *ACM TSLP*, 4(1):3.

Hal Daumé, John Langford, and Daniel Marcu. 2009. Search-based structured prediction. *Machine learning*, 75(3):297–325.

Matthew S. Dryer and Martin Haspelmath, editors. 2013. *WALS Online*. Max Planck Institute for Evolutionary Anthropology, Leipzig.

Daniela Gerz, Ivan Vulić, Edoardo Maria Ponti, Roi Reichart, and Anna Korhonen. 2018. On the relation between linguistic typology and (limitations of) multilingual language modeling. In *EMNLP*, pages 316–327.

John Goldsmith. 2001. Unsupervised learning of the morphology of a natural language. *Computational Linguistics*, 27(2):153–198.

Raymond G Gordon Jr. 2005. Ethnologue, languages of the world. *http://www. ethnologue. com/*.

Stig-Arne Grönroos, Sami Virpioja, and Mikko Kurimo. 2019. North sámi morphological segmentation with low-resource semi-supervised sequence labeling. In *International Workshop on Computational Linguistics for Uralic Languages*, pages 15–26.

Stig-Arne Grönroos, Sami Virpioja, Peter Smit, and Mikko Kurimo. 2014. Morfessor flatcat: An hmm-based method for unsupervised and semi-supervised learning of morphology. In *COLING*, pages 1177–1185.

Zellig Sabbettai Harris. 1951. *Methods in structural linguistics*. Chicago University Press.

John A Hawkins. 2015. *A comparative typology of English and German: Unifying the contrasts*. Routledge.

David Moeljadi Hiroki Nomoto, Hannah Choi and Francis Bond. 2018. Malindo morph: Morphological dictionary and analyser for malay/indonesian. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Paris, France. ELRA.

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

Mans Hulden. 2009. Foma: a finite-state compiler and library. In *Proceedings of the 12th Conference of the European Chapter of the Association for Computational Linguistics: Demonstrations Session*, pages 29–32. Association for Computational Linguistics.

INEGI. 2008. *Programa de revitalización, fortalecimiento y desarrollo de las lenguas indígenas nacionales, 2008-2012*. Instituto Nacional de Lenguas Indígenas.

Katharina Kann, Ryan Cotterell, and Hinrich Schütze. 2016. Neural morphological analysis: Encoding-decoding canonical segments. In *EMNLP*, pages 961–967.

Katharina Kann, Jesus Manuel Mager Hois, Ivan Vladimir Meza Ruiz, and Hinrich Schütze. 2018. Fortification of neural morphological segmentation models for polysynthetic minimal-resource languages. In *NAACL-HLT*, volume 1, pages 47–57.

Martin Kay. 1977. Morphological and syntactic analysis. *Linguistic structures processing*, 5:131.

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

Guillaume Klein, Yoon Kim, Yuntian Deng, Jean Senellart, and Alexander M Rush. 2017. Opennmt: Open-source toolkit for neural machine translation. In *Proceedings of ACL 2017, System Demonstrations*, pages 67–72.

Oskar Kohonen, Sami Virpioja, and Krista Lagus. 2010. Semi-supervised learning of concatenative morphology. *ACL*, page 78.

Ekkehard Konig and Johan Van der Auwera. 2013. *The germanic languages*. Routledge.

John Lafferty, Andrew McCallum, and Fernando CN Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *ICML*.

Vladimir I Levenshtein. 1966. Binary codes capable of correcting deletions, insertions, and reversals. In *Soviet physics doklady*, volume 10, pages 707–710.

Krister Lindén, Erik Axelson, Sam Hardwick, Tommi A Pirinen, and Miikka Silfverberg. 2011. Hfst—framework for compiling and applying morphologies. In *International Workshop on Systems and Frameworks for Computational Morphology*, pages 67–85. Springer.

Carolyn J MacKay and Frank R Trechsel. 2010. *Tepehua de Pisaflores, Veracruz*. El Colegio de México, Centro de Estudios Lingüísticos y Literarios.

Manuel Mager, Ximena Gutierrez-Vasques, Gerardo Sierra, and Ivan Meza-Ruiz. 2018. Challenges of language technologies for the indigenous languages of the americas. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 55–69. ACL.

Peter Makarov and Simon Clematide. 2018a. Imitation learning for neural morphological string transduction. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2877–2882, Brussels, Belgium. Association for Computational Linguistics.

Peter Makarov and Simon Clematide. 2018b. Imitation learning for neural morphological string transduction. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2877–2882.

Peter Makarov and Simon Clematide. 2018c. Neural transition-based string transduction for limited-resource setting in morphology. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 83–93, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Peter Makarov, Tatyana 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.

Quinn McNemar. 1947. Note on the sampling error of the difference between correlated proportions or percentages. *Psychometrika*, 12(2):153–157.

Jason Naradowsky and Sharon Goldwater. 2009. Improving morphology induction by learning spelling rules. In *Twenty-first International Joint Conference on Artificial Intelligence*.

Hoifung Poon, Colin Cherry, and Kristina Toutanova. 2009. Unsupervised morphological segmentation with log-linear models. In *NAACL-HLT*, pages 209–217. Association for Computational Linguistics.

Teemu Ruokolainen, Oskar Kohonen, Kairit Sirts, Stig-Arne Grönroos, Mikko Kurimo, and Sami Virpioja. 2016. A comparative study of minimally supervised morphological segmentation. *Computational Linguistics*, 42(1):91–120.

Teemu Ruokolainen, Oskar Kohonen, Sami Virpioja, and Mikko Kurimo. 2013. Supervised morphological segmentation in a low-resource learning setting using conditional random fields. In *CoNLL*, pages 29–37.

Tatyana Ruzsics and Tanja Samardzic. 2017. Neural sequence-to-sequence learning of internal word structure. In *Proceedings of the 21st Conference on Computational Natural Language Learning (CoNLL 2017)*, pages 184–194.

Gozde Gul Sahin and Mark Steedman. 2018. Character-level models versus morphology in semantic role labeling. In *ACL*, volume 1, pages 386–396.

Tarek Sakakini, Suma Bhat, and Pramod Viswanath. 2017. Morse: Semantic-ally drive-n morpheme segment-er. In *ACL*, pages 552–561. ACL.

Sunita Sarawagi and William W Cohen. 2005. Semi-markov conditional random fields for information extraction. In *Advances in neural information processing systems*, pages 1185–1192.

Abigail See, Peter J Liu, and Christopher D Manning. 2017. Get to the point: Summarization with pointer-generator networks. In *ACL*.

Wolfgang Seeker and Özlem Çetinoğlu. 2015. A graph-based lattice dependency parser for joint morphological segmentation and syntactic analysis. *TACL*, 3(1):359–373.

Abhishek Sharma, Ganesh Katrapati, and Dipti Misra Sharma. 2018a. IIT(BHU)–IIITH at CoNLL–SIGMORPHON 2018 shared task on universal morphological reinflection. In *Proceedings of the CoNLL–SIGMORPHON 2018 Shared Task: Universal Morphological Reinflection*, pages 105–111, Brussels. Association for Computational Linguistics.

Abhishek Sharma, Ganesh Katrapati, and Dipti Misra Sharma. 2018b. IIT(BHU)–IIITH at CoNLL–SIGMORPHON 2018 shared task on universal morphological reinflection. In *Proceedings of the CoNLL–SIGMORPHON 2018 Shared Task: Universal Morphological Reinflection*, pages 105–111, Brussels. Association for Computational Linguistics.

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: a simple way to prevent neural networks from overfitting. *JMLR*, 15(1):1929–1958.

Clara Vania, Andreas Grivas, and Adam Lopez. 2018. What do character-level models learn about morphology? the case of dependency parsing. In *EMNLP*, pages 2573–2583.

Clara Vania and Adam Lopez. 2017. From characters to words to in between: Do we capture morphology? In *ACL)*, volume 1, pages 2016–2027.

Oriol Vinyals, Meire Fortunato, and Navdeep Jaitly. 2015. Pointer networks. In *NeurIPS*, pages 2692–2700.

Linlin Wang, Zhu Cao, Yu Xia, and Gerard de Melo. 2016. Morphological segmentation with window LSTM neural networks. In *AAAI*.

Søren Wichmann. 2007. *Popoluca de Texistepec*, volume 27. El Colegio de Mexico AC.

Yaofei Yang, Shupin Li, Yangsen Zhang, and Hua-Ping Zhang. 2019. Point the point: Uyghur morphological segmentation using pointernetwork with gru. In *China National Conference on Chinese Computational Linguistics*, pages 371–381. Springer.

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

A Appendices

		semiCRF			joint			s2s			PGNet			IL		
DS	Lang.	Acc.	ED	F1	Acc.	ED	F1	Acc.	ED	F1	Acc.	ED	F1	Acc.	ED	F1
100	EN	**52.87**	**80.80**	61.27	48.76	88.32	**64.45**	20.34	232.07	44.35	44.0	118.53	59.28	50.99	88.21	62.47
	GR	35.12	131.78	61.75	51.33	109.08	69.07	19.39	283.53	60.11	39.29	140.17	69.99	**51.49**	**94.44**	**74.29**
	ID	58.82	68.41	71.44	52.21	95.13	70.03	25.41	207.57	68.7	53.41	86.16	78.58	**61.14**	**56.56**	**79.86**
200	EN	56.42	**75.26**	66.91	52.32	85.37	70.05	40.34	137.98	61.74	53.0	92.57	69.47	**57.26**	79.26	**70.08**
	DE	36.34	124.36	65.08	55.27	102.61	71.22	41.2	155.71	72.35	48.82	109.28	75.39	**54.90**	**88.04**	**76.94**
	ID	60.96	62.16	75.51	57.00	89.34	72.98	55.58	97.97	82.62	67.57	54.97	85.92	**71.38**	**39.89**	**86.05**
300	EN	57.30	73.37	68.50	55.67	81.39	72.23	50.27	107.28	68.35	56.54	85.81	72.87	**61.08**	**71.67**	**73.72**
	DE	38.10	118.48	68.28	56.51	101.18	71.85	49.94	118.36	76.73	52.91	97.17	77.78	**58.82**	**79.78**	**79.16**
	ID	62.68	58.48	77.92	59.22	82.89	75.23	68.09	62.39	87.29	73.74	42.84	88.87	**76.12**	**32.06**	**88.92**
400	EN	58.54	71.78	69.78	58.08	74.79	74.35	55.10	92.78	72.11	59.58	79.63	75.68	**63.14**	**68.32**	**76.35**
	DE	38.27	116.31	69.11	57.17	101.23	65.59	55.14	99.11	79.22	55.16	92.99	78.81	**60.37**	**77.61**	**80.15**
	ID	63.27	56.52	79.39	63.92	56.12	79.67	74.18	48.39	89.73	75.69	39.29	89.88	**79.32**	**27.31**	**90.80**
500	EN	59.06	70.87	69.97	57.73	74.21	73.71	60.19	83.62	76.02	62.38	74.47	77.45	**64.88**	**65.20**	**77.70**
	DE	38.72	113.84	70.08	58.53	99.16	73.00	57.84	90.63	80.66	58.78	85.44	80.20	**62.37**	**73.11**	**80.87**
	ID	63.93	55.36	79.75	66.01	52.42	78.14	78.43	39.58	91.49	78.17	34.76	91.15	**81.16**	**25.16**	**91.80**
600	EN	59.79	69.96	71.05	59.51	68.27	73.71	61.72	77.06	76.73	63.78	71.02	78.62	**66.67**	**61.46**	**79.37**
	DE	38.76	113.71	69.94	59.76	93.21	74.00	59.46	87.29	81.03	59.09	84.32	80.40	**63.12**	**71.11**	**81.41**
	ID	63.96	55.27	79.65	70.56	50.45	81.92	80.14	32.93	92.23	80.43	30.59	**92.36**	**81.62**	**24.18**	92.15

Table 7: Performance of all systems for increasing training set sizes; DS=dataset size.

	Overseg.	Underseg.	Res.	Overres.	Wrong seg.
English	0.4	21.3	12.0	18.8	71.0
German	0.0	21.7	27.9	9.5	54.9
Indonesian	11.3	41.5	11.5	17.0	63.1
Tepehua	0.4	50.1	9.2	31.2	88.8
Popoluca	13.4	62.4	3.8	26.7	91.5

Table 8: Error types found in the development set for the `Joint` model.

System	Link
`semiCRF`	http://cistern.cis.lmu.de/chipmunk/
`Joint`	https://github.com/ryancotterell/treeseg
`s2s`	https://opennmt.net/
`PGNet`	https://github.com/abhishek0318/conll-sigmorphon-2018
`IL`	https://github.com/ZurichNLP/emnlp2018-imitation-learning-for-neural-morphology

Table 9: Links to all system used in this research