View metadata, citation and similar papers at core.ac.uk

brought to you by CORE

provided by Helsingin yliopiston digitaalinen arkisto

An Encoder-Decoder Approach to the Paradigm Cell Filling Problem

Miikka Silfverberg
Department of Linguistics
University of Colorado / University of Helsinki
first.last@colorado.edu

Mans Hulden
Department of Linguistics
University of Colorado
first.last@colorado.edu

Abstract

The Paradigm Cell Filling Problem in morphology asks to complete word inflection tables from partial ones. We implement novel neural models for this task, evaluating them on 18 data sets in 8 languages, showing performance that is comparable with previous work with far less training data. We also publish a new dataset for this task and code implementing the system described in this paper.[1]

1 Introduction

An important learning question in morphology—both for NLP and models of language acquisition—is the so-called Paradigm Cell Filling Problem (PCFP). So dubbed by Ackerman et al. (2009), this problem asks how it is that speakers of a language can reliably produce inflectional forms of most lexemes without ever witnessing those forms before. For example, a Finnish noun or adjective can be inflected in 2,263 ways if one includes case forms, number, and clitics (Karlsson, 2008). However, it is unlikely that a Finnish speaker would have heard all forms for even a single, highly frequent lexical item. It is also unlikely that all 2,263 forms are found in the aggregate of all the witnessed inflected forms over different lexemes and speakers must be able to assess the felicity of, and possibly produce such inflectional combinations they have never witnessed for any noun or adjective. Figure 1 illustrates the PCFP.

This paper investigates PCFP in three different settings: (1) when we know $n > 1$ randomly selected forms in each of a number of inflection tables, (2) when we know a set of frequent word forms in each table (this most closely resembles an L1 language learning setting), and finally (3) when we know exactly $n = 1$ word form from each table.

We treat settings (1) and (2) as traditional morphological reinflection tasks (Cotterell et al., 2016) as explained in Section 2. In contrast, setting (3) is substantially more challenging because it cannot be handled using a traditional reinflection approach. To overcome this problem, we utilize an adaptive dropout mechanism which will be discussed in Section 2. This allows us to train the reinflection system in a manner reminiscent of denoising autoencoders (Vincent et al., 2008).

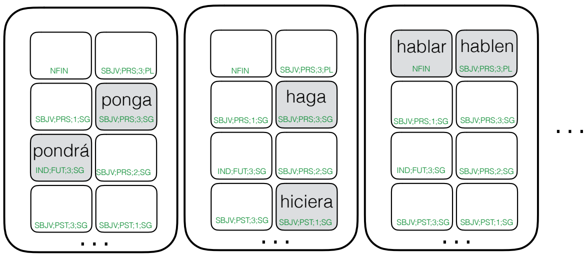

Figure 1: Illustration of the PCFP using a fraction of Spanish verb tables: given such partially filled paradigms, the task is to fill in all the missing forms.

Related Work Neural models have recently been shown to be highly competitive in many different tasks of learning supervised morphological inflection (Faruqui et al., 2016; Kann and Schütze, 2016; Makarov et al., 2017; Aharoni and Goldberg, 2017) and derivation (Cotterell et al., 2017b). Most current architectures are based on encoder-decoder models (Sutskever et al., 2014), and usually contain an attention component (Bahdanau et al., 2015).

The SIGMORPHON (Cotterell et al., 2016) and CoNLL-SIGMORPHON (Cotterell et al., 2017a, 2018) shared tasks in recent years have explored morphological inflection but not explicitly the PCFP. In the 2017 task, participants were given full paradigms—i.e. a listing of all forms—of

[1] https://github.com/mpsilfve/pcfp-data

Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, pages 2883–2889
Brussels, Belgium, October 31 - November 4, 2018. ©2018 Association for Computational Linguistics

lexemes during training after which they were given incomplete paradigms which had to be completed at test time. This is a slightly unrealistic setting in an L1-style learning scenario (Blevins and Blevins, 2009) where arguably very few full paradigms are ever witnessed and where generalization has to proceed on a number of very gappy paradigms. Of course, such gaps form a distribution where frequently used lexemes have fewer gaps than infrequent ones, which we will attempt to model in this work.

Silfverberg et al. (2018) evaluate an extension to a linguistically informed symbolic paradigm model based on stem extraction from the longest common subsequence (LCS) shared among related forms (Ahlberg et al., 2014, 2015). While the original LCS paradigm extraction method was intended to learn from complete inflection tables (Hulden, 2014), Silfverberg et al. (2018) present modifications to allow learning from incomplete paradigms as well, and apply it to the PCFP. Comparing against their results, shows that our neural model consistently outperforms such a subsequence-based learning model.

Kann et al. (2017) report results on so-called multi-source reinflection in which several input forms are used to generate one output form. This task is related to the PCFP; however, Kann et al. (2017) use full inflection tables for training. Moreover, their approach is applicable for PCFP only when 3 or more forms are given in the input tables. Since this mostly excludes our experimental settings, we do not compare to their system. Malouf (2016, 2017) documents an experiment with a generator LSTM in completing inflection tables in up to seven languages with either 10% or 40% of table entries missing. Our work differs from this in that Malouf gives as input a two-hot encoding of both the lexeme and the desired slot during training and testing for which an inflection table is to be completed, which means the system cannot complete paradigms which it has not seen examples of in the training data. By contrast, our system has no notion of lexeme and we simply work from the symbol strings which are collections of inflected forms of a lexeme given in the test data which may in principle be completely disjoint from training data lexemes. We use the Malouf system as a baseline to compare against.

2 Encoder-Decoder Models for PCFP

We explore two different models for paradigm filling. The first model is applicable when $n > 1$ forms are given in each inflection table. When exactly one ($n = 1$) form is given, we use another model.

Case n>1 When more than one form is given in training tables, PCFP can be treated as a morphological reinflection task (Cotterell et al., 2016), where the aim is to translate inflected word forms and their tags into target word forms. For example, a model would translate **tried+PAST** into the present participle (**PRES,PCPLE**) form **trying**. We adopt a common approach employed by Kann and Schütze (2016) and many others: we build a model which translates an input word form, its tag and a target tag, for example **tried+PAST+PRES,PCPLE**, into the target word form **trying**.

Our model closely follows the formulation of the encoder-decoder LSTM model for morphological reinflection proposed by Kann and Schütze (2016). We use a 1-layer bidirectional LSTM encoder for encoding the input word form into a sequence of state vectors and a 1-layer LSTM decoder with an attention mechanism over encoder states for generating the output word form.

We form training pairs by using the given forms in each table, i.e. take the cross-product of the given forms and learn to reinflect each given form in a table to another given form in the same table as demonstrated in Figure 2.[2] During test time, we predict forms for missing slots based on each of the given forms in the table and take a majority vote of the results.[3]

Case n=1 When only one form is given in each inflection table, we cannot train the model as a traditional reinflection model. The best we can do is to train a model to reinflect forms into the same form **walked+PAST+PAST** $\mapsto$ **walked** and then try to apply this model for reinflection to fill in missing forms **walked+PAST+PRES,PCPLE** $\mapsto$ **walking**. According to preliminary experiments, this however leads to massive over-fitting and the model simply learns to only copy input forms.

[2] Note that the CoNLL-SIGMORPHON data provides a 'citation form' that identifies each table; we do not use this form and the model has no knowledge of it.

[3] When only two forms are given in the partial inflection table, we randomly choose one of the resulting output forms since the vote is always tied.

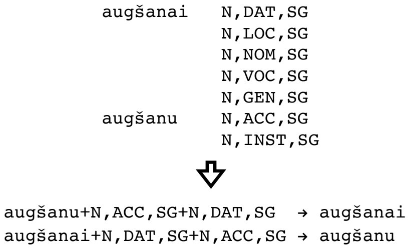

Figure 2: Partial inflection table for the Latvian noun **augana** 'growth'. From a partial inflection table with two given forms, we get two training examples. With n given forms in a table, we hence produce $n(n-1)$ training examples.

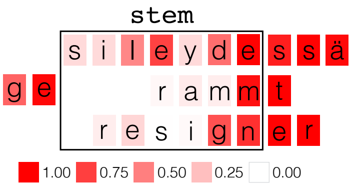

Figure 3: Language model confidences for a Finnish noun (singular inessive of **sileys** 'smoothness'), a German past participle (of **rammen** 'to ram') and a French verb (infinitive form of **resigner** 'to resign'). The figure demonstrates that confidence is higher in the inflectional affixes than in the stem in general. It is also high at the stem-affix boundary.

The idea for our approach in case $n = 1$ is to first learn to segment word forms into a stem and an affix, for example **walk+ed**. We then hide the affix in the input form and learn to inflect. In other words, we map the word form **walked** into **walk$$** and then learn a mapping **walk$$+PAST** $\mapsto$ **walked**. This model suffers less from overfitting and we can use it to find missing forms in partial inflection tables.

Since we do not have access to segmented training data, we cannot directly train a segmentation model. Instead, we use the forms in the training data to train an LSTM language model conditioned on morphological tags. We then use the language model for identifying which characters belong to stems and which characters belong to affixes.

As shown in Figure 3, the language model in general gives higher confidence for predictions of characters in the affix than in the word stem. Nevertheless, it only gives a probabilistic segmentation into a stem and affix(es). Therefore, we do not perform a deterministic segmentation. Instead we use the language model to guide a character dropout mechanism in our word inflection model. When the language model is very confident, as in the case of affix characters, we frequently drop characters. In contrast, when the language model is less confident, as in the case of stem characters, we typically keep the character. Apart from this adaptive dropout applied during training, our inflection system in case $n = 1$ is exactly the same as in case $n > 1$.

	Our baseline	Malouf (2017)
FINNISH NOUNS	99.50	99.27 ±0.09
FRENCH VERBS	99.88	99.92 ± 0.02
IRISH NOUNS	85.11	85.69 ±1.71
KHALING VERBS	99.66	99.29 ±0.08
MALTESE VERBS	98.65	98.93 ±0.32
P. CHINANTEC VERBS	91.16	91.20 ±0.97
RUSSIAN NOUNS	95.90	96.34 ±0.96

Table 1: We reproduce experiments in Malouf (2017) using our own implementation of the model. In contrast to Malouf (2017), who used cross-validation, we train one system for each language. Therefore, we only report standard deviation for the results in Column 2.

More precisely, given an input word form, which is a sequence of characters $x = x_1, ..., x_T$, the LSTM language model emits a probability $p(x_{t+1}, \mathbf{h_t}, \mathrm{E}_{x_t}, \mathrm{E}_y)$ for the next character x_{t+1} based on the entire previous input sequence $x_1, ..., x_t$. Here $\mathbf{h_t}$ is the hidden state vector of the language model at position t, E a joint tag and character embedding and y the morphological tag of the input word form. The embedding vector E_y is in fact a sum of sub-tag embeddings. For example, $\mathrm{E}_{\text{PAST+PCPLE}}$ denotes $\mathrm{E}_{\text{PAST}} + \mathrm{E}_{\text{PCPLE}}$. This allows us to handle combinations of subtags which we have not seen in the training data. Guided by the language model, we replace input characters x_{t+1} during training of the reinflection system with a dropout character $ with probability equal to language model confidence $p(x_{t+1}, \mathbf{h_t}, \mathrm{E}_{x_t}, \mathrm{E}_y)$.[4]

Baseline Model As a baseline model, we use the neural system presented by Malouf (2016, 2017) for solving PCFP. It is an LSTM generator which is conditioned on the table number of the partial inflection tables and the morphological tag index. The model is trained to generate training word forms in inflection tables. During testing, it can then generate missing forms by conditioning on morphological tags for the missing forms.

In order to assure fair comparison, we perform the paradigm completion experiment described in Malouf (2017), where 90% of the word forms in the data set is used for training and the remaining 10% for testing. [5] As the results in Table 1 show,

[4] In practice, we pad input forms with end-of-sequence characters in order to be able to drop x_1 if needed.

[5] We perform the the experiments on the original data sets,

our results very closely replicate those reported by Malouf (2017).

Implementation details We use 1-layer bidirectional LSTM encoders, decoders and generators with embeddings and hidden states of size 100. We train the language model for case $n > 1$ for 20 epochs and all other models for 60 epochs without batching. We train 10 models for every language and part-of-speech and apply majority voting to get the final output forms. All models were implemented using DyNet (Neubig et al., 2017).

3 Data

We use UniMorph morphological paradigm data in our experiments (Kirov et al., 2018). Unimorph data sets are crowd-sourced collections of morphological inflection tables based on Wiktionary. We conduct experiments on noun and verb paradigms from eight languages.[6] Not all languages have 1,000 noun and verb tables. Hence, our selection is not complete as seen in Table 3.

We conduct experiments on two different sets of tables: (1) we randomly sample 1,000 tables for each language and part-of-speech, and (2) we select Unimorph tables including some of the 10,000 most common word forms according to Wikipedia frequency. The Wikipedia word frequencies are based on plain Wikipedia text dumps from the Polyglot project (Al-Rfou et al., 2013). Georgian and Latin did not have a Polyglot Wikipedia so we excluded those. Moreover, we excluded Latvian verbs because there was very little overlap between the most frequent Wikipedia word forms and Unimorph table entries ($<$ 200 forms occurred in both). Details for both types of data sets are given in Tables 3 and 2.

	# Tables	Table Size
Finnish Nouns	1,335	27.3
Finnish Verbs	513	38.9
French Verbs	1,131	47.8
German Verbs	657	24.9
Latvian Nouns	802	12.8
Spanish Verbs	1,067	62.8
Turkish Nouns	884	78.5

Table 2: Details for inflection tables chosen according to Wikipedia word frequency.

	Table Size	Unique Forms per Table
Fin N	27.7	25.7
Fin V	39.0	37.6
Fre V	47.5	36.1
Geo N	19.0	16.9
Ger V	28.9	12.3
Lat N	11.9	7.2
Lat V	99.8	94.8
Lav N	11.6	7.6
Spa V	62.5	52.1
Tur N	74.4	54.8

Table 3: Details for randomly sampled inflection tables. The data for each language and part-of-speech consist of 1,000 tables.

	Our system	Baseline
Finnish Nouns	**63.64** ± 3.24	25.63 ± 1.63
Finnish Verbs	**24.82** ± 1.13	16.14 ± 1.14
French Verbs	**31.34** ± 1.18	14.34 ± 0.87
German Nouns	18.73 ± 1.26	**67.16** ± 3.20
German Verbs	**61.21** ± 1.85	50.18 ± 2.58
Latvian Nouns	**76.90** ± 5.30	57.28 ± 2.05
Spanish Verbs	**27.27** ± 0.72	16.61 ± 0.70
Turkish Nouns	**33.87** ± 2.03	25.00 ± 2.52

Table 4: Overall results for filling in missing forms when the 10,000 most frequent forms are given in the inflection tables. We give the 0.99 confidence intervals as given by a one-sided t-test. Figures where one system significantly outperforms the other one are in boldface.

4 Experiments and Results

We perform two experiments. In the first one, we take the set of 1,000 randomly sampled inflection tables for each language and part-of-speech and then randomly select n=1, 2 or 3 training forms from each table. We then train a reinflection system on these forms and use the resulting system to predict the missing forms. We report accuracy on correctly predicted missing forms and on reconstructing the entire paradigm correctly. In our second experiment, we consider Unimorph tables which contain entries from a list of 10,000 most common word tokens compiled using a Wikipedia dump of the language as explained above. We take the forms in the top-10,000 list as given and train a model which is used to reconstruct the remaining forms in each table. We train an identical model as in the case $n > 1$ on tables with more than one given form. As in the first task, we evaluate with regard to accuracy for reconstructed forms and full tables. Results are presented in Tables 4 and 5, and Figure 4.

5 Discussion and Conclusions

Table 4 shows results for completing tables for common lexemes. Our system significantly out-

however, we did not have access to the exact splits into training and test data used by Malouf (2017). This may influence results.

[6] Finnish (fin), French (fre), Georgian (geo), German (ger), Latin (lat), Latvian (lav), Spanish (spa) and Turkish (tur).

	Our System			Baseline		
	1 form	2 forms	3 forms	1 form	2 forms	3 forms
FIN N	**18.87** ± 0.41 (0.00 ± 0.00)	**81.72** ± 0.78 (**16.50** ± 3.76)	**93.07** ± 0.71 (**54.80** ± 4.27)	6.07 ± 0.29 (0.00 ± 0.00)	46.64 ± 0.97 (0.00 ± 0.00)	65.60 ± 1.25 (0.80 ± 0.65)
FIN V	**19.88** ± 0.69 (0.00 ± 0.00)	**87.73** ± 0.36 (**59.20** ± 5.73)	**94.63** ± 0.41 (**75.50** ± 3.82)	12.35 ± 0.58 (0.00 ± 0.00)	63.56 ± 0.89 (0.90 ± 0.90)	81.49 ± 0.42 (17.20 ± 2.21)
FRE V	**15.66** ± 0.65 (0.00 ± 0.00)	**78.30** ± 0.66 (**23.50** ± 4.06)	**83.64** ± 0.72 (**35.60** ± 4.63)	11.46 ± 0.33 (0.00 ± 0.00)	61.01 ± 0.79 (0.40 ± 0.53)	74.07 ± 1.00 (7.60 ± 2.53)
GEO N	**28.66** ± 1.12 (0.00 ± 0.00)	**90.53** ± 0.48 (**53.20** ± 6.03)	**96.02** ± 0.48 (**84.80** ± 3.28)	21.14 ± 0.84 (0.00 ± 0.00)	78.91 ± 0.56 (23.50 ± 4.03)	90.61 ± 0.76 (51.30 ± 6.19)
GER N	39.46 ± 2.18 (2.50 ± 1.83)	**84.65** ± 2.00 (**61.30** ± 4.33)	**93.38** ± 0.86 (**78.30** ± 3.14)	40.25 ± 2.09 (4.40 ± 1.83)	72.26 ± 1.70 (32.70 ± 4.70)	86.49 ± 2.09 (57.30 ± 5.11)
GER V	**43.38** ± 0.68 (0.00 ± 0.00)	**92.73** ± 0.41 (**54.70** ± 3.75)	**95.83** ± 0.38 (**70.00** ± 4.99)	33.97 ± 1.13 (0.00 ± 0.00)	83.32 ± 0.48 (17.10 ± 3.27)	90.51 ± 0.62 (34.90 ± 3.92)
LAT N	16.89 ± 1.20 (0.00 ± 0.00)	**83.59** ± 1.20 (**49.50** ± 5.13)	**91.02** ± 0.76 (**68.70** ± 4.47)	**23.62** ± 1.19 (0.10 ± 0.32)	63.27 ± 1.25 (17.40 ± 4.09)	77.96 ± 1.34 (32.90 ± 5.60)
LAT V	**17.34** ± 0.37 (0.00 ± 0.00)	**83.01** ± 0.30 (**27.00** ± 2.65)	**89.66** ± 0.44 (**2.80** ± 1.35)	5.96 ± 0.21 (0.00 ± 0.00)	52.68 ± 0.43 (0.00 ± 0.00)	68.95 ± 0.47 (0.00 ± 0.00)
LAV N	**30.11** ± 1.27 (2.00 ± 1.37)	**85.41** ± 1.07 (**48.50** ± 4.00)	**94.83** ± 0.53 (**83.40** ± 4.37)	22.35 ± 0.88 (2.60 ± 1.39)	64.76 ± 1.28 (22.20 ± 3.17)	79.21 ± 1.13 (40.80 ± 4.41)
SPA V	**27.78** ± 0.69 (0.00 ± 0.00)	**87.44** ± 0.34 (**32.20** ± 5.71)	**94.81** ± 0.25 (**59.00** ± 6.71)	10.88 ± 0.35 (0.00 ± 0.00)	70.67 ± 0.27 (0.40 ± 0.53)	84.08 ± 0.38 (11.60 ± 2.91)
TUR N	**15.70** ± 0.44 (0.00 ± 0.00)	**88.90** ± 0.60 (**19.20** ± 4.89)	**92.07** ± 0.37 (**22.80** ± 4.22)	7.94 ± 0.39 (0.00 ± 0.00)	61.95 ± 0.72 (5.60 ± 2.95)	77.02 ± 0.49 (11.40 ± 3.82)

Table 5: Accuracy for filling in missing forms when n=1,2 or 3 forms are given in the inflection table (accuracy for complete paradigms in parentheses). We give the 0.99 confidence intervals as given by a one-sided t-test. Figures where one system significantly outperforms the other one are in boldface.

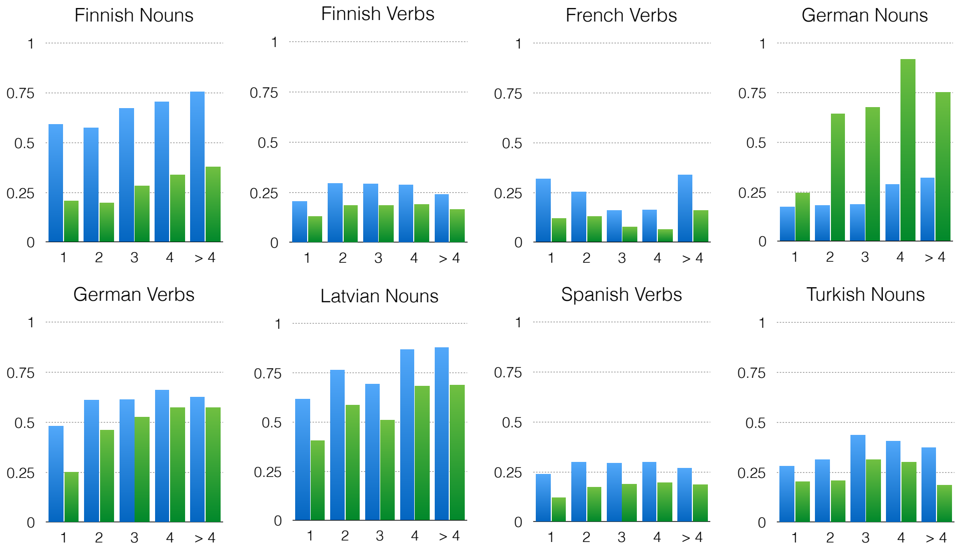

Figure 4: Detailed results for filling in missing forms when the 10,000 most frequent forms are given in the inflection tables. The blue bars (on the left) denote accuracy for our system and green bars (on the right) accuracy for the baseline system. The graphs show accuracy separately for tables where 1, 2, 3, 4, and > 4 forms are given.

performs the baseline on all other datasets apart from German nouns. We believe that the reason for the German outlier is the high degree of syncretism in German noun tables. To see why syncretism is harmful, consider the German noun **Gräben**. Its paradigm consists of eight forms but four of those are identical: **Gräben**. Only this form is observed among the top 10,000 forms in the German Wikipedia. Following Section 2, this gives rise to 12 training examples where both the input and output form are **Gräben**. This strongly biases the system to copying input forms into the output. However, this will never give the correct output because, by design, missing forms cannot be **Gräben**.[7] This can be seen as a problem with our datasets rather than the model itself. Consequently, an important future work in addressing the PCFP from an acquisition perspective is to create realistic and accurate data sets that model learner exposure both in word types and frequencies to enable assessment of the true difficulty of the PCFP.

There is a notable transition from witnessing one form in each inflection table to witnessing two forms. With only two forms given, we already approach accuracies reported in earlier work (Malouf, 2016, 2017) that used almost complete tables to train—only 10% of the forms were missing. Additionally, our encoder-decoder model strongly outperforms that generator model designed for the same task with the same amount of training data on nearly all of our datasets.

Acknowledgements

The first author was supported by The Society of Swedish Literature in Finland (SLS). NVIDIA Corp. donated the Titan Xp GPU used for this research.

[7] If the same word form occurs in multiple slots, all of them are considered known.

References

Farrell Ackerman, James P. Blevins, and Robert Malouf. 2009. Parts and wholes: Implicative patterns in inflectional paradigms. pages 54–82. Oxford University Press.

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.

Malin Ahlberg, Markus Forsberg, and Mans Hulden. 2014. Semi-supervised learning of morphological paradigms and lexicons. In *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics*, pages 569–578, Gothenburg, Sweden. Association for Computational Linguistics.

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

Rami Al-Rfou, Bryan Perozzi, and Steven Skiena. 2013. Polyglot: Distributed word representations for multilingual NLP. In *Proceedings of the Seventeenth Conference on Computational Natural Language Learning*, pages 183–192, Sofia, Bulgaria. Association for Computational Linguistics.

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

James P. Blevins and Juliette Blevins, editors. 2009. *Analogy in Grammar*. Oxford University Press.

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

Ryan Cotterell, Christo Kirov, John Sylak-Glassman, Géraldine Walther, Ekaterina Vylomova, Patrick Xia, Manaal Faruqui, Sandra Kübler, David Yarowsky, Jason Eisner, and Mans Hulden. 2017a. CoNLL-SIGMORPHON 2017 shared task: Universal morphological reinflection in 52 languages. In *Proceedings of the CoNLL SIGMORPHON 2017 Shared Task: Universal Morphological Reinflection*, pages 1–30, Vancouver. Association for Computational Linguistics.

Ryan Cotterell, Christo Kirov, John Sylak-Glassman, David Yarowsky, Jason Eisner, and Mans Hulden. 2016. The SIGMORPHON 2016 shared task—morphological reinflection. In *Proceedings of the 14th SIGMORPHON Workshop on Computational Research in Phonetics, Phonology, and Morphology*, pages 10–22, Berlin, Germany. Association for Computational Linguistics.

Ryan Cotterell, Ekaterina Vylomova, Huda Khayrallah, Christo Kirov, and David Yarowsky. 2017b. Paradigm completion for derivational morphology. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 714–720, Copenhagen, Denmark. Association for Computational Linguistics.

Manaal Faruqui, Yulia Tsvetkov, Graham Neubig, and Chris Dyer. 2016. Morphological inflection generation using character sequence to sequence learning. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, San Diego, California. Association for Computational Linguistics.

Mans Hulden. 2014. Generalizing inflection tables into paradigms with finite state operations. In *Proceedings of the 2014 Joint Meeting of SIGMORPHON and SIGFSM*, pages 29–36. Association for Computational Linguistics.

Katharina Kann, Ryan Cotterell, and Hinrich Schütze. 2017. Neural multi-source morphological reinflection. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, pages 514–524, Valencia, Spain. Association for Computational Linguistics.

Katharina Kann and Hinrich Schütze. 2016. Single-model encoder-decoder with explicit morphological representation for reinflection. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 555–560, Berlin, Germany. Association for Computational Linguistics.

Fred Karlsson. 2008. *Finnish: An essential grammar*. Routledge.

Christo Kirov, Ryan Cotterell, John Sylak-Glassman, Graldine Walther, Ekaterina Vylomova, Patrick Xia, Manaal Faruqui, Sebastian J. Mielke, Arya McCarthy, Sandra Kb̈ler, David Yarowsky, Jason Eisner, and Mans Hulden. 2018. Unimorph 2.0: Universal morphology. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Paris, France. European Language Resources Association (ELRA).

Peter Makarov, Tatiana Ruzsics, and Simon Clematide. 2017. Align and copy: UZH at SIGMORPHON 2017 shared task for morphological reinflection. In *Proceedings of the CoNLL SIGMORPHON 2017*

Shared Task: Universal Morphological Reinflection, pages 49–57, Vancouver. Association for Computational Linguistics.

Robert Malouf. 2016. Generating morphological paradigms with a recurrent neural network. *San Diego Linguistic Papers*.

Robert Malouf. 2017. Abstractive morphological learning with a recurrent neural network. *Morphology*, 27(4):431–458.

Graham Neubig, Chris Dyer, Yoav Goldberg, Austin Matthews, Waleed Ammar, Antonios Anastasopoulos, Miguel Ballesteros, David Chiang, Daniel Clothiaux, Trevor Cohn, Kevin Duh, Manaal Faruqui, Cynthia Gan, Dan Garrette, Yangfeng Ji, Lingpeng Kong, Adhiguna Kuncoro, Gaurav Kumar, Chaitanya Malaviya, Paul Michel, Yusuke Oda, Matthew Richardson, Naomi Saphra, Swabha Swayamdipta, and Pengcheng Yin. 2017. DyNet: The dynamic neural network toolkit. *arXiv preprint arXiv:1701.03980*.

Miikka Silfverberg, Ling Liu, and Mans Hulden. 2018. A computational model for the linguistic notion of morphological paradigm. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 1615–1626. Association for Computational Linguistics.

Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. 2014. Sequence to sequence learning with neural networks. In *Advances in Neural Information Processing Systems (NIPS)*, pages 3104–3112.

Pascal Vincent, Hugo Larochelle, Yoshua Bengio, and Pierre-Antoine Manzagol. 2008. Extracting and composing robust features with denoising autoencoders. In *Proceedings of the 25th international conference on Machine learning*, pages 1096–1103. ACM.