

One Wug, Two Wug+s: Transformer Inflection Models Hallucinate Affixes

Farhan Samir
University of British Columbia
fsamir@mail.ubc.ca

Miikka Silfverberg
University of British Columbia
msilfver@mail.ubc.ca

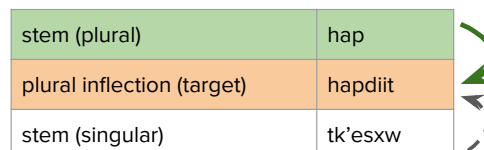
Abstract

Data augmentation strategies are increasingly important in NLP pipelines for low-resourced and endangered languages, and in neural morphological inflection, augmentation by so called data hallucination is a popular technique. This paper presents a detailed analysis of inflection models trained with and without data hallucination for the low-resourced Canadian Indigenous language Gitksan. Our analysis reveals evidence for a concatenative inductive bias in augmented models—in contrast to models trained without hallucination, they strongly prefer affixing inflection patterns over suppletive ones. We find that preference for affixation in general improves inflection performance in “wug test” like settings, where the model is asked to inflect lexemes missing from the training set. However, data hallucination dramatically reduces prediction accuracy for reduplicative forms due to a misanalysis of reduplication as affixation. While the overall impact of data hallucination for unseen lexemes remains positive, our findings call for greater qualitative analysis and more varied evaluation conditions in testing automatic inflection systems. Our results indicate that further innovations in data augmentation for computational morphology are desirable.

1 Introduction

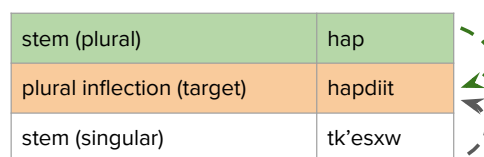
Data augmentation strategies, for instance, back-translation (Sennrich et al., 2016) and mixed sample data augmentation (Zhang et al., 2018; Guo et al., 2020), are increasingly important components of NLP pipelines (Feng et al., 2021). These strategies often form the cornerstone of modern NLP models for lower-resourced and endangered languages and dialects in particular (e.g., Kumar et al., 2021; Hauer et al., 2020; Zhao et al., 2020; Ryan and Hulden, 2020), where models can otherwise badly overfit due to the paucity of training data.

stem (plural)	hap
plural inflection (target)	hapdiit
stem (singular)	tk'esxw



(a) augmented model

stem (plural)	hap
plural inflection (target)	hapdiit
stem (singular)	tk'esxw



(b) standard model

Figure 1: Predicting a plural inflection for a lexeme using two possible source forms (singular stem and plural stem). **(a)** A Transformer model trained with data hallucination prefers the plural form as the source (depicted by a thicker arrow, representing model confidence). **(b)** The same model trained without hallucination exhibits no preference.

Consider the task of low-resource morphological inflection: high-capacity neural models trained without data augmentation are prone to collapsing at test time, achieving as little as 0% accuracy (Silfverberg et al., 2017). Conversely, those very same models trained on artificially augmented data can generalize respectably. Unfortunately, there is little research on understanding why these augmentation strategies work. We know little about the changes they cause in the model – are they simply a form of weight regularization? Do they alleviate class imbalance? Or do they provide a task-specific inductive bias?

In this paper, we investigate the data hallucination strategy, a relatively commonplace strategy (Anastasopoulos and Neubig, 2019; Silfverberg et al., 2017) for increasing the size of small morphological datasets. We conduct our study in the context of developing a Paradigm Cell-Filling (PCFP; Ackerman et al., 2009; Silfverberg and Hulden, 2018) system for the Gitksan language – a critically

endangered language with an estimated 300-850 speakers (Dunlop et al., 2018) – that can be used for applications such as developing pedagogical noun and verb conjugation exercises and further computer-assisted language learning applications.

Given a partial inflectional paradigm with n filled slots and a number of empty slots, the task is to complete the paradigm by predicting all the missing slots from the given ones. Following previous work on PCFP (Silfverberg and Hulden, 2018; Liu and Hulden, 2020), we leverage morphological reinflection models to complete PCFP. Specifically, we employ the one-source model of Liu and Hulden (2020): We use each of the n given forms in turn to predict the form in an empty target slot, giving n output forms (see Fig. 1, where $n = 2$). We then select one of the output forms as our prediction for the empty slot: We pick the predictions that the model makes with the highest confidence, a decision strategy we denote MAX.¹

Given the relatively small size of our paradigm dataset, further described in Section 2, we investigate whether data hallucination is an effective strategy for mitigating overfitting. In accordance with recent results (Liu and Hulden, 2021), we find that data hallucination improves performance in “wug test” (Berko, 1958) like conditions: where no inflectional variant of a lexeme was witnessed during training. Surprisingly, however, we also find that data hallucination significantly worsens performance for lexemes which were partially observed during training; that is at least one of the inflectional variants of the lexeme was present in the training data.

These findings motivated a controlled error analysis of our PCFP system to discover why data hallucination generalizes to the unobserved test setting but seemingly slashes performance in the observed test setting. This analysis yields two major insights. First, we find that the model trained without hallucination is “often right for the wrong reason” (McCoy et al., 2019): our error analysis reveals that a unaugmented Transformer model exhibits undesirable memorization to a significant degree, even when incorporating recently prescribed parameter settings for inflection (Wu et al., 2021; Liu and Hulden, 2020). This allows the model to memorize lexeme-specific inflection patterns, rather than

¹Note that other decision strategies such as randomly selecting an output form or taking the majority vote are also possible. These alternative strategies consistently underperform MAX, so we exclude them from the main text.

MSD	Form
ROOT	we / wa
ROOT-1PL.II	wa'm
ROOT-3.II	wet / wat

Table 1: A partial paradigm for the word meaning “name” in Gitksan. The paradigm has two entries (ROOT and ROOT-3.II) that each have two dialectal variants attested in the data. Four different one-to-one (MSD to Form) realizations of the paradigm are possible.

learning the morphophonological structure of the language. That is, we find that the model trained without hallucination relies on a brittle memorization strategy.

Second, we find evidence that data hallucination introduces an inductive bias towards concatenative morphology: where inflection is accomplished by appending affixes to a word stem. We find that the MAX strategy combined with data hallucination selects a simpler transformation: In Fig. 1, the augmented model prefers the simple transformation of appending *diit* to *hap* to predict the target *hapdiit* over the unpredictable transformation from *tk'esxw* to *hapdiit*. Conversely, the model trained without hallucination exhibits no strong preference over either transformation. Since concatenative morphology is the dominant inflection process in Gitksan, this inductive bias serves the hallucination model well in inflecting unfamiliar lexemes during testing.

Data hallucination, however, can be damaging depending on the morphophonological phenomena at hand. We find, for instance, that it dramatically reduces performance in inflections involving reduplication, a transformation that requires copying of phonological material rather than a simple concatenation (Haspelmath and Sims, 2013). While the overall effect of data augmentation in inflection has been reported as overwhelmingly positive (e.g., Lane and Bird, 2020; Anastasopoulos and Neubig, 2019; Liu and Hulden, 2021), our detailed analysis reveals that it carries both benefits and drawbacks and should therefore be applied with caution. Furthermore, our findings call for greater qualitative analysis and more varied evaluation conditions in testing automatic inflection systems.

2 Data

Our dataset comprises paradigms that were programmatically extracted from an interlinear-glossed dataset of 18,000 tokens (Forbes et al., 2017). De-

tails of the gloss to paradigm conversion procedure can be found in Appendix B. The interlinear glosses were collected during still-active language documentation efforts with Gitksan speakers.

The Gitksan-speaking community recognizes two dialects: Eastern (Upriver) and Western (Downriver), and our dataset comprises forms from both dialects. Although the two dialects are largely mutually intelligible, some lexical and phonological differences manifest, with the most prominent being a vowel shift. Consider the Gitksan translation for the word “name” in Table 1. The dialectal variation manifests as several entries for a given morphosyntactic description (henceforth MSD) in the paradigm: *we* (Western) vs. *wa* (Eastern).

Instead of attempting to model one-to-many (MSD to form) paradigms, we adhere to the simplifying constraint that each paradigm have a single realization per morphosyntactic description. In order to convert a one-to-many paradigm to a one-to-one paradigm, we aim to select a single form for each MSD so that, taken together, the inflected forms are maximally similar to each other. In the partial paradigm for for Table 1, the inflection from *wa* to *wa'm* is a simpler transformation than *we* to *wa'm*, making it simpler for a neural inflection model to acquire generalizable inflection rules. Thus, in Table 1 we would select a one-to-one paradigm with the forms *wa*, *wa'm*, and *wat*.

To obtain maximally similar inflected forms, we apply the following algorithm to a one-to-many paradigm. First, we generate all possible one-to-one realizations of the paradigm. For instance, for Table 1 one paradigm could comprise the MSD-to-form mappings: *ROOT:wa*, *ROOT:-1PL.II:wa'm*, *ROOT-3.II:wet*; there would be four possible one-to-one paradigms in total. Next, given a candidate one-to-one paradigm, we construct a fully-connected graph where each inflectional form is a vertex and every (undirected) edge is weighted by the Levenshtein distance. We then compute the weight of the minimum spanning-tree of the graph. Finally, we return the one-to-one paradigm that has the minimum-spanning tree with the lowest weight.²

We divide the resulting paradigms into four disjoint subsets. (1) A dataset for training a morpho-

²Note that the resulting paradigms are not necessarily free of dialectal variation. For instance, a paradigm where only the Western dialect form was observed for the *ROOT* and the Eastern dialect was observed for *ROOT-3.II* would still contain forms from both dialects.

logical reinflexion model Π_{train} that will be used for the PCFP task; (2) A test set containing partial paradigms Π_{obs} so that **some** of the lexemes’ inflectional variants were seen during training while the other inflectional variants are used only for testing; A validation set Π_{dev} constructed in the same manner as Π_{obs} ; (4) A test set simulating the conditions of a “wug test” (Liu and Hulden, 2021; Berko, 1958) containing complete paradigms (Π_{wug}) so that **none** of the lexemes’ inflectional variants were observed during training.

In order to train or evaluate a reinflexion system for PCFP, we first need all the paradigms to have at least two entries. This is necessary since a reinflexion datapoint is of the form *src_form:src_msd;tgt_form:tgt_msd*. Thus, our first step is to drop all paradigms that only have a single entry, providing us with 459 paradigms. Next, we randomly sample paradigms (without replacement) and add them to Π_{wug} until Π_{wug} contains 10% of the 1303 forms in our dataset.³ This procedure guarantees that no forms in paradigms belonging to Π_{wug} are ever observed during training.

For the remaining paradigms π , we split them into two disjoint sets: π_{train} and $\pi_{hold-out}$. The forms in π_{train} are added to the training set Π_{train} . The forms in $\pi_{hold-out}$ are added either to the development set Π_{dev} or partially observed test set Π_{obs} . This way, the model is allowed to observe some of the forms belonging to the (partial) paradigms in Π_{dev} and Π_{obs} during training. However, it is guaranteed not to have observed the particular forms in Π_{dev} and Π_{obs} during training.⁴

More concretely, for a paradigm of size n , between 2 and $n - 1$ forms (inclusive) are placed into train and the remaining forms are all placed into test (or all placed into dev). We obtain the following number of inflectional variants in each disjoint subset: $|\Pi_{train}| = 927$, $|\Pi_{dev}| = 124$, $|\Pi_{obs}| = 125$, $|\Pi_{wug}| = 131$. In the next section, we describe our procedure for employing these four sets of (partial) paradigms for training and evaluating a PCFP system.

³Strictly speaking, it will contain slightly more than 10%, since the last sampled paradigm may have more forms than the desired amount.

⁴More specifically, it has never seen the *MSD:form* pairs occurring in the training set.

3 Experiments and Results

Having split our paradigm dataset into the desired disjoint subsets Π_{train} , Π_{obs} , Π_{dev} , Π_{wug} , we can train Transformers in morphological inflection that can, in turn, be used for the PCFP task.⁵

Training. We form inflection training pairs by using the given forms in each paradigm in Π_{train} . Concretely, for every $\pi \in \Pi_{train}$, we take the cross product of the entries in π and learn to inflect each given form in the paradigm to another form in the same paradigm as demonstrated in Fig. 2.⁶ Counting inflection datapoints over all paradigms, we obtain 1365 datapoints in the training set for the inflection system.

We train two Transformer models. First, we train a “standard” Transformer model on the aforementioned 1365 datapoints using the parameter settings described in Wu et al. (2021) and Liu and Hulden (2020); see Appendix A. Next, we train a second “augmented” Transformer model, using the same hyperparameter settings, on the original 1365 datapoints in addition to 10,000 datapoints hallucinated from the original training dataset. We obtain the hallucination method, number of hallucinated examples (10,000), and implementation from Anastasopoulos and Neubig (2019).

Evaluation. We evaluate the models both on paradigms describing lexemes whose inflections were partially observed (Π_{obs}) and lexemes that are entirely unfamiliar (Π_{wug}). Since most of our paradigms are very sparse, containing only contain a few forms, we do a leave-one-out style evaluation procedure where, for every target form in either Π_{wug} or Π_{obs} that belongs to paradigm π , we predict it using every other form that belongs to the same paradigm π .⁷ This gives us $|\pi| - 1$ predictions for a target form, where $|\pi|$ is the total number filled slots in the paradigm.

Finally, we use the MAX strategy to select the form that was predicted with the highest likelihood averaged across output characters. We consider a paradigm π as correctly predicted if all forms for the paradigm that are present in Π_{obs} or Π_{wug} were correctly predicted.

Results and Discussion. We make a number of

⁵All code and results for this paper are available at: <anonymized for review>.

⁶Note that this means that we filter out identity pairs.

⁷We also predict from forms that belong to the training set if forms from paradigm π were included in the training set, but we only evaluate performance on the forms in Π_{wug} and Π_{obs} .

observations regarding the results in Fig. 3. First, we observe that there is a significant reduction in performance for the unfamiliar lexemes (Π_{wug}) relative to the familiar lexemes (Π_{obs}) – replicating observations made in the context of the SIGMORPHON shared tasks (Goldman et al., 2021; Cotterell et al., 2017; Liu and Hulden, 2021). We find that the augmented model reduces the deficit to 10%. That hallucination improves performance on unfamiliar lexemes has been previously observed (Liu and Hulden, 2021).

We also find, however, that hallucination worsens performance on familiar lexemes. In both cases, the aggregate accuracy scores glean little insights into these surprising results. Why does accuracy drop by nearly 50% for the non-hallucination model across the two testing conditions? How does hallucination improve performance on unfamiliar lexemes? And why does hallucination reduce performance on familiar lexeme paradigms? To understand these differences in performance between the two models and testing conditions, we turn to an analysis of the errors.

4 Error analysis

To reveal insights into the behaviour of the two Transformer models, we look into the case of Gitksan pluralization, which is instantiated as suppletion or reduplication depending on the lexeme, enabling us to investigate whether either Transformer can learn two disparate inflectional strategies. This error analysis enables us to systematically characterize the effects hallucination has on the Transformer model in inflection, demonstrating that the effects can be both beneficial and adverse.

Unaugmented Transformers memorize inflection patterns. We begin by analyzing the models’ behaviour on suppletive forms; Gitksan uses suppletion as a productive strategy for pluralization. For instance, the stem for singular forms for “laugh” is *tk’esxw*, but the plural stem is *hap*. The transformation from a singular form to a suppletive plural form is unpredictable (*ts’ehlx* → *hapdiit*); the model must instead rely on other plural source forms (e.g., *hap* → *hapdiit*). Even if the model is unable to produce the correct suppletive plural inflection, it should be able to perform the simpler task of placing higher confidence in the prediction from the plural source form (*hap*) over the singular source form (*ts’ehlx*). Failing to exhibit this preference would indicate that the model is simply

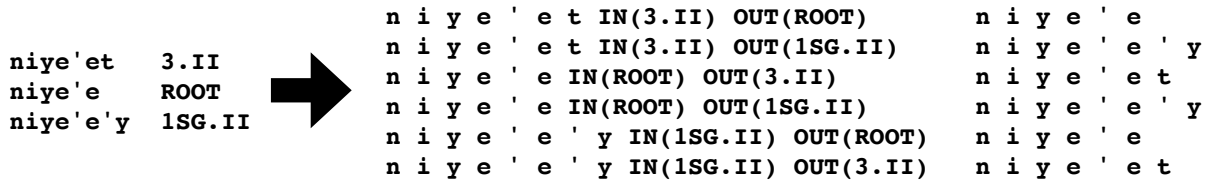


Figure 2: From a paradigm in the training data spanning three forms, we can generate six reinflexion training examples. Forms are tokenized into individual characters. Further, we distinguish tags for the input form from tags for the output form.

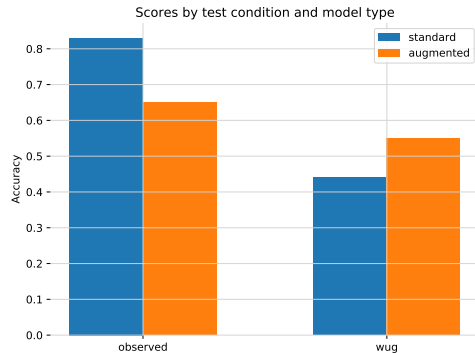


Figure 3: Performances of the augmented and standard models using the MAX decision strategy on Π_{obs} and Π_{wug} test sets.

memorizing target inflectional forms, rather than trying to acquire the morphophonological structure of the language.

Concretely, we acquire all of the 95 suppletive plurals in either Π_{wug} or Π_{obs} . We then follow our leave-one-out procedure, where every other form in the same paradigm π as the target suppletive plural form is used as a source to try to predict the target form. Instead of evaluating whether the target form was correctly predicted, we test whether the model assigns higher likelihoods to the reinflexion examples where the source form is also a suppletive plural (*hap*) over examples where the source form is singular (*tk'esxw*).

This analysis can be interpreted as a binary classification task when we hold the target suppletive form (*hapdiit*) fixed. The task is then to classify the source suppletive plural forms as positive instances and the source singular forms as negative instances. We can then use standard binary classification metrics to quantify performance. We use weighted Average Precision (Murphy, 2012), where the weight is the total number of suppletive forms in the paradigm π . We use the Average Precision implementation from `scikit-learn` (Pedregosa et al., 2011).⁸

⁸<https://scikit-learn.org/stable/>

We find that the augmented model performs significantly better in this task, achieving a weighted Average Precision of .89 while the unaugmented model achieves .52. This analysis provides evidence that the unaugmented model is memorizing the target suppletive plural form (*hapdiit*), rather than attending to and copying the suppletive plural stem (*hap*) and concatenating the appropriate affix (“diit”). This result can explain, in part, the substantial drop in performance of the unaugmented model from Π_{obs} to Π_{wug} : memorization is unlikely to generalize well for inflecting unfamiliar lexemes. Further, it can explain the stronger performance of the hallucination model in predicting forms in Π_{wug} : this inductive bias towards concatenative morphology can generalize well to unfamiliar lexemes given the prevalence of concatenative morphology in the Gitksan dataset.

Augmented Transformers struggle with non-concatenative morphology. Our Gitksan paradigm dataset comprises more than just concatenative morphology, however. Another pluralization strategy in Gitksan, albeit rarer, is reduplication, where number is indicated by copying a part of the word stem. For example, *wat* (“name”) and *hu-wat* (“name+PL”). The copied stem segment frequently undergoes further phonological alternations in the case of partial reduplication (as opposed to full reduplication; Haspelmath and Sims, 2013). While reduplication bears superficial resemblance to affixation, it cannot be analyzed as a concatenation of a stem and affix.

This resemblance, however, is sufficient to confuse prominent data hallucination techniques (Anastasopoulos and Neubig, 2019; Silfverberg et al., 2017). Consider the Gitksan word *dew* (“freeze”) which is pluralized using full reduplication: *dewdew*. The hallucinated form of this data-

`modules/generated/sklearn.metrics.average_precision_score.html`

point would have random characters substituted for the stem: e.g., *txu* -> *dewtxu*. Clearly, this hallucinated datapoint does not preserve the reduplicative structure. Unfortunately, the hallucination strategy could impair the model’s ability to perform reduplication, given that the number of examples of reduplication would become smaller relative to the size of the complete dataset.

Indeed, we find strong evidence that the hallucination model is unable to perform reduplication. We find that the standard model is able to predict the 12 instances of reduplication in Π_{wug} and Π_{obs} with .92 accuracy, while the hallucination model slashes this proficiency to a mere .25. Our analysis emphasizes the need for data-augmentation techniques that preserve reduplicative structure, given the phenomenon’s typologically robust prevalence (Haspelmath and Sims, 2013).

Reduplication is pronounced in the Gitksan dataset and causes problems for current data hallucination methods. However, it is by no means the only phenomenon where data hallucination can generate incorrect inflection patterns. Consider the example of lenition in our paradigm dataset where the final consonant undergoes voicing between vowels: *ayook* + *3.II* -> *ayook+’m* -> *ayooqa’m*. Hallucination identifies *ayoo* as the stem here due to the k/g alternation. If a hallucinated stem ending in a consonant like *dap* is used, we get an example *dapk* -> *dapqa’m*, where *k* is no longer surrounded by vowels but is still voiced when the *a’m* affix attaches, contrary to the morphophonology of Gitksan. Thus, it is possible that hallucination’s inability to preserve morphological phenomena like reduplication and lenition explain the drop in performance on the observed paradigms.⁹ Approaches that try to perform data hallucination incorporating the target language’s structure have been explored (Lane and Bird, 2020), but it’s unclear how to generalize this method without expert knowledge of the target language.

5 General Discussion

In this paper, we explore the effect of data hallucination on the Gitksan language that is currently underserved in NLP. Given the low amount of training data for the model, inflection models are likely to encounter many unfamiliar lexemes during test

time. Thus, it is important to assess the model’s ability to make adequate morphological generalizations for such lexemes. To this end, we tested the model’s ability to generalize for lexemes on a cline of familiarity from familiar (Π_{obs}) to unfamiliar (Π_{wug} Section 2).

Under these disparate conditions, we find that a data-augmented model and a standard model exhibit drastically different behaviours. We found that the standard model, a Transformer model trained under recommended parameter settings (Wu et al., 2021), memorizes inflection patterns to a significant degree (Section 3 and Section 4). At the same time, we find that data hallucination alleviates the need for memorization significantly, generalizing well to unfamiliar lexemes (Section 3) with an inductive bias towards concatenative morphology (Section 4). Data hallucination, however, is not universally beneficial: we find it reduces the model’s capacity to recognize common morphophonological phenomena (Section 4), limiting the performance improvements it can bring.

Although our study was conducted on a single language, we note that our characterization of data hallucination could be informative for languages other than Gitksan. As Section 4 demonstrates, data hallucination can encourage the model to apply voicing in incorrect contexts. Such effects are not limited to Gitksan. In English, data hallucination could give rise to erroneously conditioned allomorphy: for instance, hallucination can generate a synthetic past tense inflection example *mar* -> *mard* from a gold standard training example such as *like* -> *liked*. The desired hallucinated past tense form is of course *mared*. Overall, our work suggests common data augmentation strategies for NLP like data hallucination merit closer inspection and that further innovations in data augmentation for computational morphology are desirable.

⁹It could also explain why we don’t see a greater increase in performance on the Π_{wug} test set with the augmented model.

References

- Farrell Ackerman, James P Blevins, and Robert Malouf. 2009. Parts and wholes: Implicative patterns in inflectional paradigms. *Analogy in grammar: Form and acquisition*, pages 54–82.
- Antonios Anastasopoulos and Graham Neubig. 2019. Pushing the limits of low-resource morphological inflection. *arXiv preprint arXiv:1908.05838*.
- Jean Berko. 1958. The child’s learning of english morphology. *WORD*, 14:150–177.
- Ryan Cotterell, Christo Kirov, John Sylak-Glassman, Géraldine Walther, Ekaterina Vylomova, Patrick Xia, Manaal Faruqui, Sandra Kübler, David Yarowsky, Jason Eisner, and Mans Hulden. 2017. Conll-sigmorphon 2017 shared task: Universal morphological reinflection in 52 languages. In *CoNLL Shared Task*.
- Britt Dunlop, Suzanne Gessner, Tracey Herbert, and Aliana Parker. 2018. [Report on the status of BC First Nations languages](#). Report of the First People’s Cultural Council. Retrieved March 24, 2019.
- Steven Y. Feng, Varun Gangal, Jason Wei, Sarath Chandar, Soroush Vosoughi, Teruko Mitamura, and Edward Hovy. 2021. [A survey of data augmentation approaches for NLP](#). In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 968–988, Online. Association for Computational Linguistics.
- Clarissa Forbes, Henry Davis, Michael Schwan, and the UBC Gitksan Research Laboratory. 2017. Three Gitksan texts. In *Papers for the 52nd International Conference on Salish and Neighbouring Languages*, pages 47–89. UBC Working Papers in Linguistics.
- Omer Goldman, David Guriel, and Reut Tsarfaty. 2021. (un) solving morphological inflection: Lemma overlap artificially inflates models’ performance. *arXiv preprint arXiv:2108.05682*.
- Demi Guo, Yoon Kim, and Alexander Rush. 2020. [Sequence-level mixed sample data augmentation](#). In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 5547–5552, Online. Association for Computational Linguistics.
- Martin Haspelmath and Andrea Sims. 2013. *Understanding morphology*. Routledge.
- Bradley Hauer, Amir Ahmad Habibi, Yixing Luan, Arnob Mallik, and Grzegorz Kondrak. 2020. [Low-resource G2P and P2G conversion with synthetic training data](#). In *Proceedings of the 17th SIGMORPHON Workshop on Computational Research in Phonetics, Phonology, and Morphology*, pages 117–122, Online. Association for Computational Linguistics.
- Sachin Kumar, Antonios Anastasopoulos, Shuly Winter, and Yulia Tsvetkov. 2021. Machine translation into low-resource language varieties. In *ACL/IJCNLP*.
- William Lane and Steven Bird. 2020. Bootstrapping techniques for polysynthetic morphological analysis. *arXiv preprint arXiv:2005.00956*.
- L. Liu and Mans Hulden. 2020. Analogy models for neural word inflection. In *COLING*.
- Ling Liu and Mans Hulden. 2021. Can a transformer pass the wug test? tuning copying bias in neural morphological inflection models. *arXiv preprint arXiv:2104.06483*.
- Tom McCoy, Ellie Pavlick, and Tal Linzen. 2019. [Right for the wrong reasons: Diagnosing syntactic heuristics in natural language inference](#). In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3428–3448, Florence, Italy. Association for Computational Linguistics.
- Kevin P Murphy. 2012. *Machine learning: a probabilistic perspective*. MIT press.
- Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael Auli. 2019. fairseq: A fast, extensible toolkit for sequence modeling. In *NAACL*.
- F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. 2011. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.
- Zach Ryan and Mans Hulden. 2020. [Data augmentation for transformer-based G2P](#). In *Proceedings of the 17th SIGMORPHON Workshop on Computational Research in Phonetics, Phonology, and Morphology*, pages 184–188, Online. Association for Computational Linguistics.
- Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. [Improving neural machine translation models with monolingual data](#). In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 86–96, Berlin, Germany. Association for Computational Linguistics.
- Miikka Silfverberg and Mans Hulden. 2018. An encoder-decoder approach to the paradigm cell filling problem. In *EMNLP*.
- Miikka Silfverberg, Adam Wiemerslage, L. Liu, and Lingshuang Jack Mao. 2017. Data augmentation for morphological reinflection. In *CoNLL*.
- Shijie Wu, Ryan Cotterell, and Mans Hulden. 2021. Applying the transformer to character-level transduction. In *EACL*.

Hongyi Zhang, Moustapha Cisse, Yann N. Dauphin, and David Lopez-Paz. 2018. mixup: Beyond empirical risk minimization. In *International Conference on Learning Representations*.

Xingyuan Zhao, Satoru Ozaki, Antonios Anastasopoulos, Graham Neubig, and Lori Levin. 2020. [Automatic interlinear glossing for under-resourced languages leveraging translations](#). In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 5397–5408, Barcelona, Spain (Online). International Committee on Computational Linguistics.

A Transformer training details

We train all models using the implementation of Transformer in the `fairseq` package (Ott et al., 2019). Both the encoder and decoder have 4 layers with 4 attention heads, an embedding size of 256 and hidden layer size of 1024. We train with the Adam optimizer starting of the learning rate at 0.001. We chose the batch size (400) and maximum updates (5000) based on the highest accuracy on the development data. Our model setting resembles the work of Wu et al. (2021) who found that a relatively large batch size is beneficial for morphological inflection. Prediction is performed with the best checkpoint model, according to validation accuracy score, and a beam of width 5.

B Database of Gitksan Inflection Tables

We perform all experiments on a database of Gitksan inflection tables. In total, there are 1055 inflection tables containing 2125 inflected forms. An interlinear-glossed corpus of Gitksan narratives Forbes et al. (2017) forms the basis of our database. The Gitksan corpus is glossed at the root level meaning that word forms are broken down into roots, derivational morphemes and inflectional morphemes. This level of description is too fine-grained for our purposes and we, therefore, combine roots and potential derivational material into word stems. The inflected forms for each noun and verb stem are gathered into inflection tables. In total, there are 33 possible inflected forms and each inflection table will contain a subset of these forms. An example table is shown in Appendix C.

C Sample inflection table

A Gitksan inflection table for *jok* ('to dwell') generated from IGT and displayed in TSV format. Each row in the table contains five cells: (1) a morphosyntactic description, (2) an English translation, (3) a gloss with an English lemma, (3) a canonical segmented output form, (4) the surface word form, and (5) a gloss with a Gitksan lemma. Many cells in the table are empty since they were unattested in the IGT data.

```

ROOT dwell jok jok jok
ROOT-SX dwell-SX jok-it jogat jok-SX
ROOT-SX dwell-SX jok-it jogot jok-SX
ROOT-PL _ _ _ _
ROOT-3PL _ _ _ _
ROOT-ATTR _ _ _ _
ROOT-3.II dwell-3.II jok-t jokt jok-3.II
ROOT-PL-SX PL~dwell-SX CVC~jok-it jaxjogat PL~jok-SX
ROOT-PL-SX PL~dwell-SX CVC~jok-it jaxjogot PL~jok-SX
ROOT-1SG.II dwell-1SG.II jok-'y jogo'y jok-1SG.II
ROOT-2SG.II _ _ _ _
ROOT-2PL.II _ _ _ _
ROOT-3PL.II dwell-3PL.II jok-diit jokdiit jok-3PL.II
ROOT-1PL.II dwell-1PL.II jok-'m jogo'm jok-1PL.II
ROOT-PL-3PL _ _ _ _
ROOT-TR-3.II _ _ _ _
ROOT-PL-3.II PL~dwell-3.II CVC~jok-t jaxjokt PL~jok-3.II
ROOT-PL-ATTR _ _ _ _
ROOT-PL-2SG.II _ _ _ _
ROOT-TR-1SG.II _ _ _ _
ROOT-PL-3PL.II PL~dwell-3PL.II CVC~jok-diit jaxjokdiit PL~jok-3PL.II
ROOT-PL-1SG.II _ _ _ _
ROOT-TR-1PL.II _ _ _ _
ROOT-PL-1PL.II PL~dwell-1PL.II CVC~jok-'m jaxjogo'm PL~jok-1PL.II
ROOT-TR-2PL.II _ _ _ _
ROOT-TR-3PL.II _ _ _ _
ROOT-TR-2SG.II _ _ _ _
ROOT-PL-TR-3.II _ _ _ _
ROOT-PL-TR-2SG.II _ _ _ _
ROOT-PL-TR-3PL.II _ _ _ _
ROOT-PL-TR-1SG.II _ _ _ _
ROOT-PL-TR-1PL.II _ _ _ _
ROOT-PL-TR-2PL.II _ _ _ _

```