# Searching for Search Errors in Neural Morphological Inflection

**Martina Forster**[1] **Clara Meister**[1] **Ryan Cotterell**[1,2]
[1]ETH Zürich [2]University of Cambridge
martfors@ethz.ch {clara.meister, ryan.cotterell}@inf.ethz.ch

## Abstract

Neural sequence-to-sequence models are currently the predominant choice for language generation tasks. Yet, on word-level tasks, exact inference of these models reveals the empty string is often the global optimum. Prior works have speculated this phenomenon is a result of the inadequacy of neural models for language generation. However, in the case of morphological inflection, we find that the empty string is almost never the most probable solution under the model. Further, greedy search often finds the global optimum. These observations suggest that the poor calibration of many neural models may stem from characteristics of a specific subset of tasks rather than general ill-suitedness of such models for language generation.

## 1 Introduction

Neural sequence-to-sequence models are omnipresent in the field of natural language processing due to their impressive performance. They hold state of the art on a myriad of tasks, e.g., neural machine translation (NMT; Ott et al., 2018b) and abstractive summarization (AS; Lewis et al., 2019). Yet, an undesirable property of these models has been repeatedly observed in word-level tasks: When using beam search as the decoding strategy, increasing the beam width beyond a size of $k = 5$ often leads to a drop in the quality of solutions (Murray and Chiang, 2018; Yang et al., 2018; Cohen and Beck, 2019). Further, in the context of NMT, it has been shown that the empty string is frequently the most-probable solution under the model (Stahlberg and Byrne, 2019). Some suggest this is a manifestation of the general inadequacy of neural models for language generation tasks (Koehn and Knowles, 2017; Kumar and Sarawagi, 2019; Holtzman et al., 2020; Stahlberg, 2020); in this work, we find evidence demonstrating otherwise.

| | $k = 1$ | $k = 10$ | $k = 100$ | $k = 500$ |
|---|---|---|---|---|
| **NMT** | 63.1% | 46.1% | 44.3% | 6.4% |
| **MI** | 0.8% | 0.0% | 0.0% | 0.0% |

Table 1: Percentage of search errors—which we define as instances where the search strategy does not find the global optimum under the model—for Transformers trained on IWSLT'14 De-En (NMT) and SIGMORPHON 2020 (Morphological Inflection; MI) when decoding with beam search for varying beam widths ($k$). MI results are averaged across languages.

Sequence-to-sequence transducers for character-level tasks often follow the architectures of their word-level counterparts (Faruqui et al., 2016; Lee et al., 2017), and have likewise achieved state-of-the-art performance on e.g., morphological inflection generation (Wu et al., 2020) and grapheme-to-phoneme conversion (Yolchuyeva et al., 2019). Given prior findings, we might expect to see the same degenerate behavior in these models—however, we do not. We run a series of experiments on morphological inflection (MI) generators to explore whether neural transducers for this task are similarly poorly calibrated, i.e. are far from the true distribution $p(\mathbf{y} \mid \mathbf{x})$. We evaluate the performance of two character-level sequence-to-sequence transducers using different decoding strategies; our results, previewed in Tab. 1, show that evaluation metrics do not degrade with larger beam sizes as in NMT or AS. Additionally, only in extreme circumstances, e.g., low-resource settings with less than 100 training samples, is the empty string ever the global optimum under the model.

Our findings directly refute the claim that neural architectures are inherently inadequate for modeling language generation tasks. Instead, our results admit two potential causes of the degenerate behavior observed in tasks such as NMT and

AS: (1) lack of a deterministic mapping between input and output and (2) a (perhaps irreparable) discrepancy between sample complexity and training resources. Our results alone are not sufficient to accept or reject either hypothesis, and thus we leave these as future research directions.

## 2 Neural Transducers

Sequence-to-sequence transduction is the transformation of an input sequence into an output sequence. Tasks involving this type of transformation are often framed probabilistically, i.e., we model the *probability* of mapping one sequence to another. On many tasks of this nature, neural sequence-to-sequence models (Sutskever et al., 2014; Bahdanau et al., 2015) hold state of the art.

Formally, a neural sequence-to-sequence model defines a probability distribution $p_{\boldsymbol{\theta}}(\mathbf{y} \mid \mathbf{x})$ parameterized by a neural network with a set of learned weights $\boldsymbol{\theta}$ for an input sequence $\mathbf{x} = \langle x_1, x_2, \dots \rangle$ and output sequence $\mathbf{y} = \langle y_1, y_2, \dots \rangle$. Morphological inflection and NMT are two such tasks, wherein our outputs are both strings. Neural sequence-to-sequence models are typically locally normalized, i.e. $p_{\boldsymbol{\theta}}$ factorizes as follows:

$$p_{\boldsymbol{\theta}}(\mathbf{y} \mid \mathbf{x}) = \prod_{t=1}^{|\mathbf{y}|} p_{\boldsymbol{\theta}}(y_t \mid \mathbf{x}, \mathbf{y}_{<t}) \tag{1}$$

Given a vocabulary $\mathcal{V}$, each conditional $p_{\boldsymbol{\theta}}$ is a distribution over $\mathcal{V} \cup \{\text{EOS}\}$ and $y_0 := \text{BOS}$. We consider $p_{\boldsymbol{\theta}}(\mathbf{y} \mid \mathbf{x})$ to be **well-calibrated** if its probability estimates are representative of the true likelihood that a solution $\mathbf{y}$ is correct.

**Morphological Inflection.** In the task of morphological inflection, $\mathbf{x}$ is an encoding of the lemma concatenated with a flattened morphosyntactic description (MSD) and $\mathbf{y}$ is the target inflection. As a concrete example, consider inflecting the German word *Bruder* into the genitive plural, as shown in Tab. 2. Then, $\mathbf{x}$ is the string ⟨B r u d e r GEN PL⟩ and $\mathbf{y}$ is the string ⟨B r ü d e r⟩. As this demonstrates, morphological inflection generation is, by its nature, modeled at the character level (Faruqui et al., 2016; Wu and Cotterell, 2019), i.e., our target vocabulary $\mathcal{V}$ is a set of characters in the language. Note that $\mathbf{y} \in \mathcal{V}^*$, but $\mathbf{x} \notin \mathcal{V}^*$ due to the additional encoding of the MSD. This stands in contrast to NMT, which is typically performed on a (sub)word level, making the vocabulary size orders of magnitude larger.

| | Singular | Plural |
|---|---|---|
| Nominativ | Bruder | Brüder |
| Genitiv | Bruders | Brüder |
| Dativ | Bruder | Brüdern |
| Akkusativ | Bruder | Brüder |

Table 2: Inflection table for the German word *Bruder*

Another important differentiating factor of morphological inflection generation in comparison to many other generation tasks in NLP is the one-to-one mapping between source and target.[1] In contrast, there are almost always many correct ways to translate a sentence into another language or to summarize a large piece of text; this characteristic manifests itself in training data where a single phrase has instances of different mappings, making tasks such as translation and summarization inherently ambiguous.

## 3 Decoding

In the case of probabilistic models, the decoding problem is the search for the most-probable sequence among valid sequences $\mathcal{V}^*$ under the model $p_{\boldsymbol{\theta}}$:

$$\mathbf{y}^\star = \operatorname*{argmax}_{\mathbf{y} \in \mathcal{V}^*} \log p_{\boldsymbol{\theta}}(\mathbf{y} \mid \mathbf{x}) \tag{2}$$

This problem is also known as maximum-a-posteriori (MAP) inference. Decoding is often performed with a heuristic search method such as greedy or beam search (Reddy, 1977), since performing exact search can be computationally expensive, if not impossible.[2] While for a deterministic task, greedy search is optimal under a Bayes optimal model,[3] most text generation tasks benefit from using beam search. However, text quality almost invariably decreases for beam sizes larger than $k = 5$. This phenomenon is sometimes referred to as the **beam search curse**, and has been investigated in detail by a number of scholarly works (Koehn and Knowles, 2017; Murray and Chiang, 2018; Yang et al., 2018; Stahlberg and Byrne, 2019; Cohen and Beck, 2019; Eikema and Aziz, 2020).

[1] While there are cases where there exist multiple inflected forms of a lemma, e.g., in English the past tense of *dream* can be realized as either *dreamed* or *dreamt*, these cases (termed "overabundance") are rare (Thornton, 2019).

[2] The search space is exponential in the sequence length and due to the non-Markov nature of (typical) neural transducers, dynamic-programming techniques are not helpful.

[3] Under such a model, the correct token $y_i$ at time step $i$ will be assigned all probability mass.

| | Transformer | | | | HMM | | | |
|---|---|---|---|---|---|---|---|---|
| | $k=1$ | $k=10$ | $k=100$ | Dijkstra | $k=1$ | $k=10$ | $k=100$ | Dijkstra |
| Overall | 90.34% | 90.37% | 90.37% | 90.37% | 86.03% | 85.62% | 85.60% | 85.60% |
| Low-resource | 84.10% | 84.12% | 84.12% | 84.12% | 70.99% | 69.37% | 69.31% | 69.31% |
| High-resource | 94.05% | 94.08% | 94.08% | 94.08% | 93.60% | 93.72% | 93.72% | 93.72% |

Table 3: Prediction accuracy (averaged across languages) by decoding strategy for Transformer and HMM. We include breakdown for low-resource and high-resource trained models. $k$ indicates beam width.

Exact decoding can be seen as the case of beam search where the beam size is effectively stretched to infinity.[4] By considering the complete search space, it finds the globally best solution under the model $p_{\boldsymbol{\theta}}$. While, as previously mentioned, exact search can be computationally expensive, we can employ efficient search strategies due to some properties of $p_{\boldsymbol{\theta}}$. Specifically, from Eq. (1), we can see that the scoring function for sequences $\mathbf{y}$ is monotonically decreasing in $t$. We can therefore find the provably optimal solution with Dijkstra's algorithm (Dijkstra, 1959), which terminates and returns the global optimum the first time it encounters an EOS. Additionally, to prevent a large memory footprint, we can lower-bound the search using any complete hypothesis, e.g., the empty string or a solution found by beam search (Stahlberg and Byrne, 2019; Meister et al., 2020). That is, we can prematurely stop exploring solutions whose scores become less than these hypotheses at any point in time. Although exact search is an exponential-time method in this setting, we see that, in practice, it terminates quickly due to the peakiness of $p_{\boldsymbol{\theta}}$ (see App. A). While the effects of exact decoding and beam search decoding with large beam widths have been explored for a number of word-level tasks (Stahlberg and Byrne, 2019; Cohen and Beck, 2019; Eikema and Aziz, 2020), to the best of our knowledge, they have not yet been explored for any character-level sequence-to-sequence tasks.

## 4 Experiments

We run a series of experiments using different decoding strategies to generate predictions from morphological inflection generators. We report results for two near-state-of-the-art models: a multilingual Transformer (Wu et al., 2020) and a (neuralized) hidden Markov model (HMM; Wu and Cotterell, 2019). For reproducibility, we mimic their proposed architectures and exactly follow their data pre-processing steps, training strategies and hyperparameter settings.[5]

[4] This interpretation is useful when comparing with beam search with increasing beam widths.

| | Beam $k=1$ | Beam $k=10$ | Optimum | Empty String |
|---|---|---|---|---|
| Transformer | -0.619 | -0.617 | -0.617 | -6.56 |
| HMM | -1.08 | -0.89 | -0.80 | -20.15 |

Table 4: Average log probability of inflections generated with various decoding strategies and the empty string (averaged across all languages).

**Data.** We use the data provided by the SIGMORPHON 2020 shared task (Vylomova et al., 2020), which features lemmas, inflections, and corresponding MSDs in the UniMorph schema (Kirov et al., 2018) in 90 languages in total. The set of languages is typologically diverse (spanning 18 language families) and contains both high- and low-resource examples, providing a spectrum over which we can evaluate model performance. The full dataset statistics can be found on the task homepage.[6] When reporting results, we consider languages with $< 1000$ and $\geq 10000$ training samples as low- and high-resource, respectively.

**Decoding Strategies.** We decode morphological inflection generators using exact search and beam search for a range of beam widths. We use the SGNMT library for decoding (Stahlberg et al., 2017) albeit adding Dijkstra's algorithm.

### 4.1 Results

Tab. 3 shows that the accuracy of predictions from neural MI generators generally does not decrease when larger beam sizes are used for decoding; this observation holds for both model architec-

[5] https://github.com/shijie-wu/neural-transducer/tree/sharedtasks

[6] https://sigmorphon.github.io/sharedtasks/2020/task0/

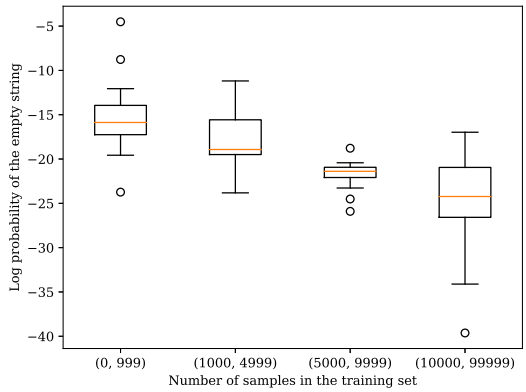


Figure 1: Average (log) probability of the empty string for different training dataset sizes for HMM.

| | HMM | Transformer |
|---|---|---|
| Overall | 2.03% | 0% |
| Low-resource | 8.65% | 0% |
| High-resource | 0.0002% | 0% |

Table 5: Average percentage of empty strings when decoding with exact inference for HMM and Transformer, with resource group breakdown.

| | $k=1$ | $k=10$ | $k=100$ | $k=200$ |
|---|---|---|---|---|
| HMM | 6.20% | 2.33% | 0.001% | 0.0% |
| Transformer | 0.68% | 0.0% | 0.0% | 0.0% |

Table 6: Average percentage of search errors (averaged across languages) for beam search with beam width $k$.

tures. While it may be expected that models for low-resource languages generally perform worse than those for high-resource ones, this disparity is only prominent for HMMs, where the difference between high- and low-resource accuracy is $\approx 24\%$ vs. $\approx 10\%$ for the Transformers. Notably, for the HMM, the global optimum under the model is the empty string far more often for low-resource languages than it is for high-resource ones (see Tab. 5). We can explicitly see the inverse relationship between the log-probability of the empty string and resource size in Fig. 1. In general, across models for all 90 languages, the global optimum is rarely the empty string (Tab. 5). Indeed, under the Transformer-based transducer, the empty string was *never* the global optimum. This is in contrast to the findings of Stahlberg and Byrne (2019), who found for word-level NMT that the empty string was the optimal translation in more than 50% of cases, even under state-of-the-art models. Rather, the average log-probabilities of the empty string (which is quite low) and the chosen inflection lie far apart (Tab. 4).

## 5 Discussion

Our findings admit two potential hypotheses for poor calibration of neural models in certain language generation tasks, a phenomenon we do not observe in morphological inflection. First, the tasks in which we observe this property are ones that lack a deterministic mapping, i.e. tasks for which there may be more than one correct solution for any given input. As a consequence, probability mass may be spread over an arbitrarily large number of hypotheses (Ott et al., 2018a; Eikema and Aziz, 2020). In contrast, the task of morphological inflection has a near-deterministic mapping. We observe this empirically in Tab. 4, which shows that the probability of the global optimum on average covers most of the available probability mass—a phenomenon also observed by Peters and Martins (2019). Further, as shown in Tab. 6, the dearth of search errors even when using greedy search suggests there are rarely competing solutions under the model. We posit it is the lack of ambiguity in morphological inflection that allows for the well-calibrated models we observe.

Second, our experiments contrasting high- and low-resource settings indicate insufficient training data may be the main cause of the poor calibration in sequence-to-sequence models for language generation tasks. We observe that models for MI trained on fewer data typically place more probability mass on the empty string. As an extreme example, we consider the case of the Zarma language, whose training set consists of only 56 samples. Under the HMM, the average log-probability of the generated inflection and empty string are very close ($-8.58$ and $-8.77$, respectively). Furthermore, on the test set, the global optimum of the HMM model for Zarma is the empty string 81.25% of the time.

From this example, we can conjecture that lack of sufficient training data may manifest itself as the (relatively) high probability of the empty string or the (relatively) low probability of the optimum. We can extrapolate to models for NMT and other word-level tasks, for which we frequently see the above phenomenon. Specifically, our experiments suggest that when neural lan-

guage generators frequently place high probability on the empty string, there may be a discrepancy between the available training resources and the number of samples needed to successfully learn the target function. While this at first seems an easy problem to fix, we expect the number of resources needed in tasks such as NMT and AS is much larger than that for MI if not due to the size of the output space alone; perhaps so large that they are essentially unattainable. Under this explanation, for certain tasks, there may not be a straightforward fix to the degenerate behavior observed in some neural language generators.

## 6 Conclusion

In this work, we investigate whether the poor calibration often seen in sequence-to-sequence models for word-level tasks also occurs in models for morphological inflection. We find that character-level models for morphological inflection are generally well-calibrated, i.e. the probability of the globally best solution is almost invariably much higher than that of the empty string. This suggests the degenerate behavior observed in neural models for certain word-level tasks is not due to the inherent incompatibility of neural models for language generation. Rather, we find evidence that poor calibration may be linked to specific characteristics of a subset of these task, and suggest directions for future exploration of this phenomenon.

## References


Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *3rd International Conference on Learning Representations*.

Eldan Cohen and Christopher Beck. 2019. Empirical analysis of beam search performance degradation in neural sequence models. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97, pages 1290–1299, Long Beach, California, USA.

Edsger W. Dijkstra. 1959. A note on two problems in connexion with graphs. *Numerische Mathematik*, 1(1):269–271.

Bryan Eikema and Wilker Aziz. 2020. Is MAP decoding all you need? The inadequacy of the mode in neural machine translation. *CoRR*, abs/12005.10283.

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*, pages 634–643, San Diego, California. Association for Computational Linguistics.

Ari Holtzman, Jan Buys, Maxwell Forbes, and Yejin Choi. 2020. The curious case of neural text degeneration. *International Conference on Learning Representations*.

Christo Kirov, Ryan Cotterell, John Sylak-Glassman, Géraldine Walther, Ekaterina Vylomova, Patrick Xia, Manaal Faruqui, Sabrina J. Mielke, Arya McCarthy, Sandra Kübler, 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)*, Miyazaki, Japan. European Language Resources Association (ELRA).

Philipp Koehn and Rebecca Knowles. 2017. Six challenges for neural machine translation. In *Proceedings of the First Workshop on Neural Machine Translation*, pages 28–39, Vancouver. Association for Computational Linguistics.

Aviral Kumar and Sunita Sarawagi. 2019. Calibration of encoder decoder models for neural machine translation. *CoRR*, abs/1903.00802.

Jason Lee, Kyunghyun Cho, and Thomas Hofmann. 2017. Fully character-level neural machine translation without explicit segmentation. *Transactions of the Association for Computational Linguistics*, 5:365–378.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Ves Stoyanov, and Luke Zettlemoyer. 2019. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. *CoRR*, abs/1910.13461.

Clara Meister, Ryan Cotterell, and Tim Vieira. 2020. Best-first beam search. *Transactions of the Association for Computational Linguistics*, 8(0).

Kenton Murray and David Chiang. 2018. Correcting length bias in neural machine translation. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 212–223, Brussels, Belgium. Association for Computational Linguistics.

Myle Ott, Michael Auli, David Grangier, and Marc'Aurelio Ranzato. 2018a. Analyzing uncertainty in neural machine translation. In *International Conference on Machine Learning*, pages 3956–3965.

Myle Ott, Sergey Edunov, David Grangier, and Michael Auli. 2018b. Scaling neural machine translation. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 1–9, Brussels, Belgium. Association for Computational Linguistics.

Ben Peters and André F. T. Martins. 2019. IT–IST at the SIGMORPHON 2019 shared task: Sparse two–headed models for inflection. In *Proceedings of the 16th Workshop on Computational Research in Phonetics, Phonology, and Morphology*, pages 50–56, Florence, Italy. Association for Computational Linguistics.

Raj Reddy. 1977. Speech understanding systems: A summary of results of the five-year research effort at carnegie mellon university.

Felix Stahlberg. 2020. *The Roles of Language Models and Hierarchical Models in Neural Sequence-to-Sequence Prediction*. Ph.D. thesis, University of Cambridge.

Felix Stahlberg and Bill Byrne. 2019. On NMT search errors and model errors: Cat got your tongue? In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3356–3362, Hong Kong, China. Association for Computational Linguistics.

Felix Stahlberg, Eva Hasler, Danielle Saunders, and Bill Byrne. 2017. SGNMT – a flexible NMT decoding platform for quick prototyping of new models and search strategies. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 25–30, Copenhagen, Denmark. Association for Computational Linguistics.

Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to sequence learning with neural networks. In Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 27*, pages 3104–3112. Curran Associates, Inc.

Anna M. Thornton. 2019. Overabundance in morphology. *Oxford Research Encyclopedia of Linguistics*.

Ekaterina Vylomova, Jennifer White, Elizabeth Salesky, Sabrina J. Mielke, Shijie Wu, Edoardo Ponti, Rowan Hall Maudslay, Ran Zmigrod, Joseph Valvoda, Svetlana Toldova, Francis Tyers, Elena Klyachko, Ilya Yegorov, Natalia Krizhanovsky, Paula Czarnowska, Irene Nikkarinen, Andrej Krizhanovsky, Tiago Pimentel, Lucas Torroba Hennigen, Christo Kirov, Garrett Nicolai, Adina Williams, Antonios Anastasopoulos, Hilaria Cruz, Eleanor Chodroff, Ryan Cotterell, Miikka Silfverberg, and Mans Hulden. 2020. The SIGMORPHON 2020 Shared Task 0: Typologically diverse morphological inflection. In *Proceedings of the 17th SIGMORPHON Workshop on Computational Research in Phonetics, Phonology, and Morphology*.

Shijie Wu and Ryan Cotterell. 2019. Exact hard monotonic attention for character-level transduction. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1530–1537, Florence, Italy. Association for Computational Linguistics.

Shijie Wu, Ryan Cotterell, and Mans Hulden. 2020. Applying the transformer to character-level transduction. *CoRR*, abs/2005.10213.

Yilin Yang, Liang Huang, and Mingbo Ma. 2018. Breaking the beam search curse: A study of (re-)scoring methods and stopping criteria for neural machine translation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3054–3059, Brussels, Belgium. Association for Computational Linguistics.

Sevinj Yolchuyeva, Géza Németh, and Bálint Gyires-Tóth. 2019. Transformer based grapheme-to–phoneme conversion. In *Interspeech 2019, 20th Annual Conference of the International Speech Communication Association*, pages 2095–2099. ISCA.

# A Timing

| | **Transformer** | | **HMM** | |
|---|---|---|---|---|
| | $k = 1$ | Dijkstra | $k = 1$ | Dijkstra |
| Overall | 0.082 | 0.091 | 0.016 | 0.027 |
| Low-resource | 0.072 | 0.082 | 0.013 | 0.032 |
| High-resource | 0.075 | 0.083 | 0.017 | 0.026 |

Table 7: Average time (s) for inflection generation by decoding strategy. Breakdown by resource group is included.