Detecting Annotation Errors in Morphological Data with the Transformer

Ling Liu and Mans Hulden University of Colorado first.last@colorado.edu

Abstract

Annotation errors that stem from various sources are usually unavoidable when performing large-scale annotation of linguistic data. In this paper, we evaluate the feasibility of using the Transformer model to detect various types of annotator errors in type-based morphological datasets that contain inflected word forms. We evaluate our error detection model on four languages by injecting three different types of artificial errors into the data: (1) typographic errors, where single characters in the data are inserted, replaced, or deleted; (2) linguistic confusion errors where two inflected forms are systematically swapped; and (3) self-adversarial errors where the Transformer model itself is used to generate plausible-looking, but erroneous forms by retrieving high-scoring predictions from a Transformer search beam. Results show that the model can with perfect, or nearperfect recall detect errors in all three scenarios, even when significant amounts of the annotated data (5%-30%) are corrupted on all languages tested. Precision varies across the languages and types of errors, but is high enough that the model can reliably be used to flag suspicious entries in large datasets for further scrutiny by human annotators.

1 Introduction

Deep learning models have been responsible for state-of-the-art performance in many tasks involving morphological generation and analysis (Devlin et al., 2019; Raffel et al., 2019; Cotterell et al., 2016; Vylomova et al., 2020). However, to reach adequate performance, large amounts of labeled examples are usually required for training (Cotterell et al., 2017; Silfverberg et al., 2017; Liu and Hulden, 2021b). Annotation of morphological data is particularly expensive since it requires both domain and language expertise (McCarthy et al., 2020). Manual correction and quality control of annotated data adds to the cost (van Halteren, 2000). In light of this, we evaluate the feasibility of using a deep learning model to automatically detect annotation errors with the goal of reducing the cost of annotation correction and quality control.

Earlier work on annotation error detection has largely been non-neural and focused on other types of annotation, such as part-of-speech (POS) tagging (van Halteren, 2000; Květoň and Oliva, 2002; Dickinson and Meurers, 2003; Loftsson, 2009), syntactic parsing (Eskin, 2000; Ambati et al., 2011), or semantic labeling (Dickinson and Lee, 2008). A neural model error detector— an LSTM-based tagger—has been used by Rehbein and Ruppenhofer (2017) to detect POS tagging errors.

In this paper, we propose a method to apply a Transformer model (Vaswani et al., 2017) to detect annotation errors in morphological data. In order to evaluate the method, we simulate errors by introducing artificial perturbations to our annotated data, which are generated in three different ways to simulate different types of annotation errors. Experimental results show that the Transformer model can detect annotation errors in morphological data very effectively, even when the datasets contain a high percentage of erroneous forms.

2 **Experiments**

2.1 Data

We use data from four languages in the UniMorph project (Kirov et al., 2018) for experiments. The data has been vetted and used in multiple SIG-MORPHON shared tasks (Cotterell et al., 2016, 2017, 2018; McCarthy et al., 2019; Vylomova et al., 2020). Therefore, we expect very few erroneous entries in this dataset. The data is organized into inflection tables where each slot in an inflection table is given as a tab-separated (*lemma, inflected form, morphosyntactic tag*) triple, as shown in the left chart in Figure 1.

Language choice The four languages—Finnish, German, Russian and Spanish—represent differ-

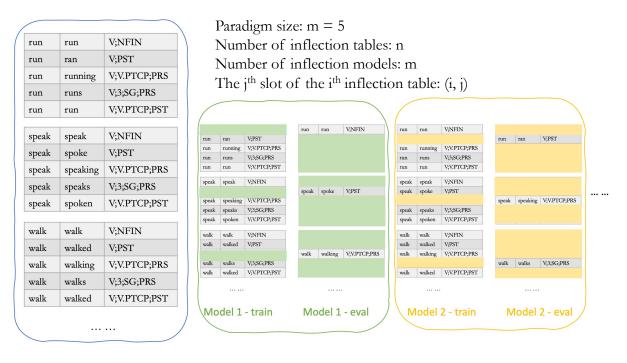


Figure 1: Illustration of the *leave-n-out* training and evaluation data split setup. We systematically leave out one slot in each inflection table for evaluation, and use the remaining slots to train one particular inflection model. For each inflection model, we rotate which slot is left out. The number of models we train is the same as the corresponding paradigm size.

ent morphological complexities and challenges. German and Russian nouns have relatively small paradigm sizes, while Spanish and Finnish verbs have large paradigms; the paradigm size of Finnish nouns and German verbs is somewhere in between. Finnish has an agglutinative inflectional system with a large paradigm size, especially for verbs. Though German inflection tables are not particularly large, characteristic of the language are the many cases of syncretism in each inflection table. Spanish verbs have a large paradigm size, but the inflection is quite regular. Russian has a fusional morphological system and is written in Cyrillic script whereas the other three languages use Latin script. An additional reason for our particular choice of languages has been to provide a range of difficulty for neural models-German has consistently been among the most difficult languages to inflect in the SIGMORPHON shared tasks; Finnish and Russian have been of intermediate difficulty, and Spanish has been consistently 'easy'. Further, by limiting ourselves to languages that have been used in multiple shared tasks, we assure-importantly-that the gold data for our experiments is itself largely error-free, something which is not obviously the case for many other languages in UniMorph.

Language	POS	Paradigm Size	Table Count	Total Examples	Accuracy	
		m	n	x		
German	Ν	8	160	1,280	0.9664	
Russian	Ν	12	240	2,880	0.9625	
Finnish	Ν	28	140	3,920	0.9959	
German	V	29	145	4,205	0.9919	
Finnish	V	141	141	19,881	0.9896	
Spanish	V	70	70	4,900	0.9980	

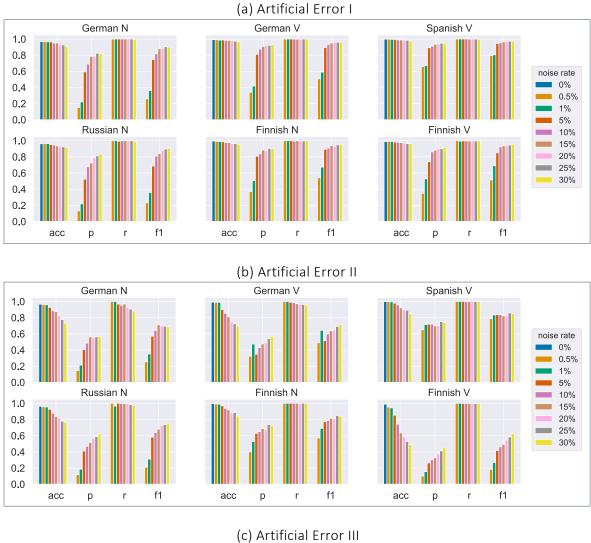
Table 1: Basic data information. The last column presents the Transformer inflection model performance (average accuracy) when no artificial error is inserted.

2.2 Experiment setup

Inflection model The Transformer (Vaswani et al., 2017) is the current state-of-the-art model architecture for morphological inflection generation, even when the amount of training data is limited (Vylomova et al., 2020; Liu and Hulden, 2020a,b, 2021a,b; Moeller et al., 2020, 2021; Wu et al., 2021; Liu, 2021); we therefore adopt this architecure in all experiments.¹

Applying the Transformer to detect morphological data errors The core intuition behind our error detection model is that we *train inflection generation models* on a subset of the inflected forms in our total dataset, and then *apply these models*

¹We implement all models in FAIRSEQ (Ott et al., 2019) and the hyperparameter setting follow Liu and Hulden (2020a) exactly.



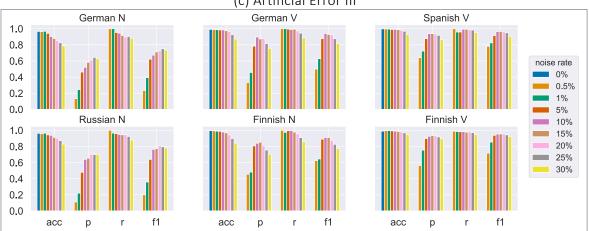


Figure 2: Model performance on adding different types of artificial errors. In each group, the bars from left to right show results for introducing an increasingly larger amount of artificial errors. Accuracy (*acc*) is the inflection model performance. Precision (*p*), recall (*r*) and F1-score (f_1) evaluate the effectiveness of error detection with the inflection model. *p*, *r* and f_1 are not applicable when no artificial error, i.e. 0%, is introduced.

to generate precisely those inflected forms that the inflection models have <u>not</u> been trained on. If a model's prediction for these forms disagrees with

the corresponding held-out annotated form, we flag that particular annotated form as a potential error.

Preliminary experiment and data split Throughout our experiments, we use complete inflection tables for our labeled data. Moreover, the dataset is a small subset of the UniMorph tables, ranging from 70 tables (Spanish verbs) to 240 (Russian nouns). The reason for limiting the data is twofold. First, we want to ensure that error detection is feasible with datasets significantly smaller than large projects such as UniMorph. Secondly, before our actual error detection experiment, we want to verify that the Transformer model is powerful enough to reconstruct, with high accuracy, single unseen (or potentially erroneous) forms in the data.

We use a *leave-n-out* cross-validation setup to split the data for training and evaluating the model before attempting to perform error detection. Specifically, as illustrated in Figure 1, we systematically leave one slot out in each inflection table for evaluation and use the remaining slots to train one particular inflection model. For each such model, we rotate which slot is left out. The number of models we train for each POS of a language is thus the same as the corresponding paradigm size, m. The evaluation data size for each model is n, the same as the number inflection tables in the data, and the training data size for each model is $m \times n - n$. Each model is thus trained to make predictions for slots it has not witnessed-one missing slot per table-and the union of all models' predictions cover all the slots. Table 1 shows the accuracy when using the m models to perform an artificial reconstruction of "unseen forms". For example, we train m = 8 inflection models for German nouns, each model is trained on 1,120 $(8 \times 160 - 160)$ slots and evaluated on n = 160 slots.

Generating artificial errors We now simulate noisy annotation data by injecting artificial errors into the above dataset in three different ways before training models. The first method generates artificial errors (Artificial Error I) to mimic typographic errors by *inserting, replacing or deleting* a single character in an inflected word form. The second error model simulates annotator confusion by *swapping two randomly sampled slots* with different inflected forms in a randomly chosen inflection table, denoted as Artificial Error II. The third type of artificial error, Artificial Error III, is self-adversarial to generate plausible-looking noise: we first train a single Transformer inflection model with the complete data for each POS of a language, then apply

it to predict inflected forms for slots it *has* been trained on. We use beam search at decoding time and *pick out the second best (but erroneous) prediction* to represent a noisy inflected form. This self-adversarial approach gives us incorrect word forms which are however very close to the ground truth inflected word forms. We hypothesize that such errors are more difficult to identify than the others.

Erroneous inflected forms of each type are introduced to the original data at different error rates: 0.5%, 1%, 5%, 15%, 20%, 25% and 30% (of all forms).

Evaluation metrics We evaluate the error detection model w.r.t. *accuracy*, i.e. the ratio of correctly predicted forms vs. all predicted forms and also *precision*, *recall*, and *F1-score*.

3 Results and Discussion

Figure 2 provides a summary of the experiment results, plotting the accuracy, precision, recall, and F1-score for each POS of each language, averaged across the m models after adding Artificial Errors I, II, III at different amounts, respectively. Detailed numbers are provided in Table 2 in the appendix.

We observe that the accuracy of the model decreases as more word erroneous forms are added, but is still high overall. This indicates that the *leaven-out* training strategy is robust to noise in the data. For every type of artificial error, the recall is 1.0 or very close to 1.0 after varying amounts of noise is injected. In other words, the model can identify all, or nearly all the artificial errors we introduce, even when a large amount of noise is mixed into the gold data. The precision increases (from a low of 0.11 to a high of 0.95) as more errors are added, indicating that a reasonably small amount of false positives would be produced by the model. (See Table 3 in the appendix for detailed counts.)

As such, if an annotator were to manually correct the forms flagged by the model, all erroneous annotations would be corrected and the annotator should not be frustrated by vetting a large number of already-correct annotations. To illustrate this, consider the average precision (0.43) for all six datasets with Artificial Error type I (typos) where 1% of the forms are corrupted—a plausible scenario in an annotation project. Under such assumptions, our model would present flagged forms in a dataset for vetting to an annotator, and, indeed, nearly half of these flagged forms would be true

errors, and no errors would be undetected (since the recall is 1.0).

However, we observe that the worst case (e.g. lowest F1 scores on average) where the annotation error detection model performs is the second type of artificial error. In this type of error, we consistently switched a portion of slots. The worst error detection model performance on this type of error points to the limitation of the annotation error detection method we propose: it cannot detect consistent errors if the errors in question are present in a large portion of the data; for example, in the extreme case that all the forms in the paradigm carry the same error, it is impossible for the inflection model to learn the ground-truth inflection. Another shortcoming of our proposed approach is that it requires relatively complete inflection tables, which are expensive to annotate as to expertise and effort. Future work is needed to evaluate whether the method works when there are slots missing in most inflection tables.

4 Conclusion

In this work, we propose a method to leverage the Transformer model architecture for annotation error detection in morphological data. We propose to systematically leave out one slot in each morphological inflection table as the data to be detected and use such subsets of annotated data to train individual Transformer inflection models—one for each group of missing slots—and then apply the inflection models to make predictions for the heldout slots. If the predicted form disagrees with the actual annotation (a form the predicting model has not seen), the model flags that form as erroneous.

To check efficiency, we evaluate the model under three different scenarios where we inject artificial errors into gold data, simulating noisy data resulting from an annotation process: typographic errors generated by inserting, replacing or deleting a single character in an inflected word form; errors resulting from annotator confusion where two slots in an inflection table are swapped; and selfadversarial errors where erroneous but plausible predictions generated by the Transformer inflection model are introduced. Our experiments on four languages with different morphological characteristics and levels of irregularity indicate that the proposed method can detect every type of error in morphological datasets very effectively. Even when large portions of the data (5% to 30%) have

been replaced with corrupted forms, our model retains perfect, or near-perfect, recall and also shows increasingly higher precision as more erroneous forms are present.

The results show that the Transformer model can detect various kinds of errors without producing excessive false positive predictions. We believe such a model can directly be incorporated into the correction and quality control process of morphological data annotation projects, specifically for low-resource language where datasets are in the early stages of development and few annotators are available. Further research should investigate how well this basic method of error detection works in other linguistic annotation domains.

References

- Bharat Ram Ambati, Rahul Agarwal, Mridul Gupta, Samar Husain, and Dipti Misra Sharma. 2011. Error detection for treebank validation. In *Proceedings* of the 9th Workshop on Asian Language Resources, pages 23–30, Chiang Mai, Thailand. Asian Federation of Natural Language Processing.
- Ryan Cotterell, Christo Kirov, John Sylak-Glassman, Géraldine Walther, Ekaterina Vylomova, Arya D. Mc-Carthy, Katharina Kann, Sabrina J. 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, pages 1–27, Brussels. 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. 2017. 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.
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In Proceedings of the 2019 Conference of

the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

- Markus Dickinson and Chong Min Lee. 2008. Detecting errors in semantic annotation. In *Proceedings* of the Sixth International Conference on Language Resources and Evaluation (LREC'08), Marrakech, Morocco. European Language Resources Association (ELRA).
- Markus Dickinson and W. Detmar Meurers. 2003. Detecting errors in part-of-speech annotation. In 10th Conference of the European Chapter of the Association for Computational Linguistics, Budapest, Hungary. Association for Computational Linguistics.
- Eleazar Eskin. 2000. Detecting errors within a corpus using anomaly detection. In 1st Meeting of the North American Chapter of the Association for Computational Linguistics.
- 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).
- Pavel Květoň and Karel Oliva. 2002. (semi-)automatic detection of errors in PoS-tagged corpora. In COL-ING 2002: The 19th International Conference on Computational Linguistics.
- Ling Liu. 2021. Computational morphology with neural network approaches. *arXiv preprint arXiv:2105.09404*.
- Ling Liu and Mans Hulden. 2020a. Analogy models for neural word inflection. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 2861–2878, Barcelona, Spain (Online). International Committee on Computational Linguistics.
- Ling Liu and Mans Hulden. 2020b. Leveraging principal parts for morphological inflection. In *Proceedings of the 17th SIGMORPHON Workshop on Computational Research in Phonetics, Phonology, and Morphology*, pages 153–161, Online. Association for Computational Linguistics.
- Ling Liu and Mans Hulden. 2021a. Backtranslation in neural morphological inflection. In *Proceedings* of the Second Workshop on Insights from Negative Results in NLP, pages 81–88, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

- Ling Liu and Mans Hulden. 2021b. Can a transformer pass the wug test? Tuning copying bias in neural morphological inflection models. *arXiv preprint arXiv:2104.06483*.
- Hrafn Loftsson. 2009. Correcting a POS-tagged corpus using three complementary methods. In Proceedings of the 12th Conference of the European Chapter of the ACL (EACL 2009), pages 523–531, Athens, Greece. Association for Computational Linguistics.
- Arya D. McCarthy, Christo Kirov, Matteo Grella, Amrit Nidhi, Patrick Xia, Kyle Gorman, Ekaterina Vylomova, Sabrina J. Mielke, Garrett Nicolai, Miikka Silfverberg, Timofey Arkhangelskiy, Nataly Krizhanovsky, Andrew Krizhanovsky, Elena Klyachko, Alexey Sorokin, John Mansfield, Valts Ernštreits, Yuval Pinter, Cassandra L. Jacobs, Ryan Cotterell, Mans Hulden, and David Yarowsky. 2020. UniMorph 3.0: Universal Morphology. In Proceedings of the 12th Language Resources and Evaluation Conference, pages 3922–3931, Marseille, France. European Language Resources Association.
- Arya D. McCarthy, Ekaterina Vylomova, Shijie Wu, Chaitanya Malaviya, Lawrence Wolf-Sonkin, Garrett Nicolai, Christo Kirov, Miikka Silfverberg, Sabrina J. Mielke, Jeffrey Heinz, Ryan Cotterell, and Mans Hulden. 2019. The SIGMORPHON 2019 shared task: Morphological analysis in context and crosslingual transfer for inflection. In Proceedings of the 16th Workshop on Computational Research in Phonetics, Phonology, and Morphology, pages 229–244, Florence, Italy. Association for Computational Linguistics.
- Sarah Moeller, Ling Liu, and Mans Hulden. 2021. To POS tag or not to POS tag: The impact of POS tags on morphological learning in low-resource settings. In Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers), pages 966–978, Online. Association for Computational Linguistics.
- Sarah Moeller, Ling Liu, Changbing Yang, Katharina Kann, and Mans Hulden. 2020. IGT2P: From interlinear glossed texts to paradigms. In Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), pages 5251–5262, Online. Association for Computational Linguistics.
- 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 *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations)*, pages 48–53, Minneapolis, Minnesota. Association for Computational Linguistics.
- Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2019. Exploring the limits

of transfer learning with a unified text-to-text transformer. *arXiv preprint arXiv:1910.10683*.

- Ines Rehbein and Josef Ruppenhofer. 2017. Detecting annotation noise in automatically labelled data. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1160–1170, Vancouver, Canada. Association for Computational Linguistics.
- Miikka Silfverberg, Adam Wiemerslage, Ling Liu, and Lingshuang Jack Mao. 2017. Data augmentation for morphological reinflection. In *Proceedings of the CoNLL SIGMORPHON 2017 Shared Task: Universal Morphological Reinflection*, pages 90–99, Vancouver. Association for Computational Linguistics.
- Hans van Halteren. 2000. The detection of inconsistency in manually tagged text. In *Proceedings of the COLING-2000 Workshop on Linguistically Interpreted Corpora*, pages 48–55, Centre Universitaire, Luxembourg. International Committee on Computational Linguistics.
- Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *arXiv preprint arXiv:1706.03762*.
- Ekaterina Vylomova, Jennifer White, Elizabeth Salesky, Sabrina J. Mielke, Shijie Wu, Edoardo Maria Ponti, Rowan Hall Maudslay, Ran Zmigrod, Josef Valvoda, Svetlana Toldova, Francis Tyers, Elena Klyachko, Ilya Yegorov, Natalia Krizhanovsky, Paula Czarnowska, Irene Nikkarinen, Andrew 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. SIGMORPHON 2020 shared task 0: Typologically diverse morphological inflection. In Proceedings of the 17th SIGMORPHON Workshop on Computational Research in Phonetics, Phonology, and Morphology, pages 1-39, Online. Association for Computational Linguistics.
- Shijie Wu, Ryan Cotterell, and Mans Hulden. 2021. Applying the transformer to character-level transduction. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 1901–1907, Online. Association for Computational Linguistics.

A Detailed experiment results

rtificial	icial Artificial Error I				Artificial Error II				Artificial Error III			
or Rate	acc	р	r	f1	acc	р	r	f1	acc	р	r	f1
0%	0.9664	N/A	N/A	N/A	0.9664	N/A	N/A	N/A	0.9664	N/A	N/A	N/A
0.5%	0.9688	0.1489	1.0	0.2592	0.9625	0.1455	1.0	0.254	0.9641	0.1321	1.0	0.2334
1%	0.9641	0.2203	1.0	0.3611	0.9586	0.2121	1.0	0.35	0.968	0.2453	1.0	0.394
5%	0.9641	0.5926	1.0	0.7442	0.9258	0.4052	1.0	0.5714	0.9422	0.4621	0.9531	0.6224
10%	0.9508	0.6882	1.0	0.8153	0.8852	0.4841	1.0	0.6421	0.9031	0.5216	0.9453	0.672
15%	0.95	0.7837	1.0	0.8787	0.8742	0.5619	1.0	0.7113	0.8789	0.5828	0.9167	0.712
20%	0.9344	0.7853	1.0	0.8797	0.8281	0.5574	1.0	0.6969	0.8531	0.6117	0.8984	0.727
25%	0.9266	0.8226	1.0	0.9027	0.7742	0.562	1.0	0.6938	0.8258	0.6437	0.9031	0.751
30%	0.9094	0.8219	0.9974	0.9012	0.732	0.5643	0.9974	0.6877	0.7883	0.6301	0.8828	0.735
0%	0.9625	N/A	N/A	N/A	0.9625	N/A	N/A	N/A	0.9625	N/A	N/A	N/A
0.5%	0.9653	0.1304	1.0	0.2307	0.958	0.1168	1.0	0.2092	0.958	0.1111	1.0	0.2
1%	0.9632	0.218	1.0	0.358	0.9549	0.1847	1.0	0.3101	0.9649	0.2205	0.9655	0.359
5%	0.9524	0.5238	0.9931	0.6859	0.9253	0.4103	0.9931	0.5819	0.9438	0.4792	0.9583	0.638
10%	0.9483	0.6776	1.0	0.8078	0.9255	0.469	1.0	0.6378	0.9378	0.6393	0.9479	0.763
15%	0.9358	0.7248	1.0	0.8404	0.8378	0.5162	1.0	0.6798	0.9128	0.6549	0.9444	0.773
20%	0.9302	0.7888	0.9983	0.8813	0.8201	0.5706	0.9983	0.7257	0.9028	0.705	0.9462	0.808
25%	0.9302	0.8133	0.9986	0.8965	0.0201	0.5884	0.9986	0.7366	0.9020	0.6993	0.9402	0.795
20%	0.9194	0.8367	0.9965	0.9096	0.766	0.622	0.9965	0.759	0.8302	0.7031	0.9250	0.795
<u> </u>	0.9959	N/A	N/A	N/A	0.9959	N/A	N/A	N/A	0.8362	N/A	N/A	N/A
0%	0.9939	0.3704	1.0	0.5406	0.9939	0.4	1.0	0.5714	0.9939	0.4545	1.0	0.625
1%	0.9913	0.5063	1.0	0.5400	0.9923	0.4	1.0	0.6896	0.9939	0.4343	0.975	0.644
1% 5%	0.9901		1.0	0.8929	0.9908	0.5203	1.0	0.0890	0.989	0.4813	0.975	0.890
		0.8066	0.9949	0.8929	0.9702	0.6282	0.9949	0.7718		0.8038		0.890
10%	0.977	0.8369							0.9788		0.9974	
15%	0.976	0.8855	1.0	0.9393	0.9184	0.6861	1.0	0.8138	0.9681	0.8531	0.9779	0.911
20%	0.9643	0.8737	0.9974	0.9315	0.8804	0.678	0.9974	0.8069	0.9429	0.8075	0.9579	0.876
25%	0.9638	0.9047	0.998	0.9491	0.8852	0.7368	0.998	0.8485	0.8982	0.7557	0.9092	0.825
30%	0.9571	0.9052	0.9991	0.9498	0.8434	0.7273	0.9991	0.8421	0.8418	0.7042	0.8605	0.774
0%	0.9919	N/A	N/A	N/A	0.9919	N/A	N/A	N/A	0.9919	N/A	N/A	N/A
0.5%	0.9895	0.3385	1.0	0.5058	0.9891	0.3235	1.0	0.4889	0.9895	0.3333	1.0	0.5
1%	0.9857	0.4175	1.0	0.5891	0.9883	0.4731	1.0	0.6423	0.9879	0.4574	1.0	0.627
5%	0.9874	0.8084	1.0	0.894	0.9006	0.3471	1.0	0.5141	0.985	0.7836	0.9953	0.876
10%	0.9843	0.875	0.9976	0.9323	0.8528	0.4293	0.9976	0.5981	0.986	0.8968	0.9905	0.941
15%	0.9826	0.9078	0.9984	0.9509	0.8098	0.4749	0.9984	0.6379	0.9753	0.8733	0.9937	0.929
20%	0.9793	0.9231	0.9988	0.9595	0.7477	0.4888	0.9988	0.648	0.9636	0.8797	0.9738	0.924
25%	0.9729	0.922	1.0	0.9594	0.7244	0.5397	1.0	0.6915	0.9272	0.8154	0.9449	0.875
30%	0.9693	0.9292	0.9984	0.9626	0.6923	0.5716	0.9984	0.7153	0.8728	0.7571	0.8867	0.816
0%	0.998	N/A	N/A	N/A	0.998	N/A	N/A	N/A	0.998	N/A	N/A	N/A
0.5%	0.9973	0.6579	1.0	0.7937	0.9971	0.65	1.0	0.7879	0.9971	0.641	1.0	0.781
1%	0.9951	0.6712	1.0	0.8033	0.9959	0.7143	1.0	0.8333	0.9959	0.7231	0.9592	0.824
5%	0.9937	0.8909	1.0	0.9423	0.9794	0.7193	1.0	0.8367	0.9908	0.8769	0.9592	0.916
10%	0.9894	0.9108	1.0	0.9533	0.9573	0.7208	1.0	0.8363	0.992	0.9383	0.9939	0.965
15%	0.9873	0.9327	0.9986	0.9645	0.921	0.698	0.9986	0.8217	0.9884	0.9396	0.9946	0.966
20%	0.9849	0.9441	1.0	0.9712	0.898	0.7033	1.0	0.8255	0.98	0.9353	0.9878	0.960
25%	0.9829	0.9481	0.9992	0.973	0.8924	0.752	0.9992	0.8582	0.9688	0.917	0.9829	0.948
30%	0.9753	0.9453	0.9986	0.9712	0.8484	0.74	0.9986	0.8496	0.93	0.8653	0.9524	0.906
0%	0.9896	N/A	N/A	N/A	0.9896	N/A	N/A	N/A	0.9896	N/A	N/A	N/A
0.5%	0.9905	0.346	1.0	0.5141	0.9545	0.1003	1.0	0.1823	0.9961	0.5625	0.99	0.717
1%	0.991	0.528	0.995	0.6899	0.9442	0.1542	0.995	0.2672	0.9966	0.7538	0.9849	0.854
1% 5%	0.9818	0.7394	0.998	0.8495	0.8527	0.2631	0.998	0.2072	0.9934	0.898	0.9819	0.938
10%	0.9818	0.7594	0.998	0.8495	0.8327	0.2031	0.998	0.4622	0.9902	0.9257	0.9819	0.953
					1							0.955
												0.955
												0.955
												0.943
15% 20% 25% 30%		0.9765 0.971 0.9633 0.9622	0.9765 0.8826 0.971 0.9002 0.9633 0.9006	0.97650.88260.99830.9710.90020.9980.96330.90060.997	0.9765 0.8826 0.9983 0.9369 0.971 0.9002 0.998 0.9466 0.9633 0.9006 0.997 0.9464	0.9765 0.8826 0.9983 0.9369 0.6335 0.971 0.9002 0.998 0.9466 0.5865 0.9633 0.9006 0.997 0.9464 0.5266	0.9765 0.8826 0.9983 0.9369 0.6335 0.3259 0.971 0.9002 0.998 0.9466 0.5865 0.3759 0.9633 0.9006 0.997 0.9464 0.5266 0.4121	0.9765 0.8826 0.9983 0.9369 0.6335 0.3259 0.9983 0.971 0.9002 0.998 0.9466 0.5865 0.3759 0.998 0.9633 0.9006 0.997 0.9464 0.5266 0.4121 0.997	0.9765 0.8826 0.9983 0.9369 0.6335 0.3259 0.9983 0.4914 0.971 0.9002 0.998 0.9466 0.5865 0.3759 0.998 0.5463 0.9633 0.9006 0.997 0.9464 0.5266 0.4121 0.997 0.5832	0.9765 0.8826 0.9983 0.9369 0.6335 0.3259 0.9983 0.4914 0.9855 0.971 0.9002 0.998 0.9466 0.5865 0.3759 0.998 0.5463 0.9805 0.9633 0.9006 0.997 0.9464 0.5266 0.4121 0.997 0.5832 0.9688	0.9765 0.8826 0.9983 0.9369 0.6335 0.3259 0.9983 0.4914 0.9855 0.9349 0.971 0.9002 0.998 0.9466 0.5865 0.3759 0.998 0.5463 0.9805 0.9317 0.9633 0.9006 0.997 0.9464 0.5266 0.4121 0.997 0.5832 0.9688 0.9182	0.9765 0.8826 0.9983 0.9369 0.6335 0.3259 0.9983 0.4914 0.9855 0.9349 0.9769 0.971 0.9002 0.998 0.9466 0.5865 0.3759 0.998 0.5463 0.90805 0.9317 0.9806 0.9633 0.9006 0.997 0.9464 0.5266 0.4121 0.997 0.5832 0.9688 0.9182 0.971

Table 2: Model performance in details on adding artificial errors of different types in different amounts. This is the information used to create Figure 2 in section 3. When no artificial errors, i.e. 0%, are introduced, precision, recall and F1-score are not applicable.

	Artificial	Artificial Error I			Ar	tificial Erro	or II	Artificial Error III		
	Error	True	Detected	Artificial	True	Detected	Artificial	True	Detected	Artificial
	Rate	Positive	Error	Error	Positive	Error	Error	Positive	Error	Error
	0%	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
	0.5%	7	47	7	8	47	8	7	53	7
	1%	13	59	13	14	59	14	13	53	13
	5%	64	108	64	62	108	64	61	132	64
German N	10%	128	186	128	122	186	128	121	232	128
	15%	192	245	192	186	245	192	176	302	192
	20%	256	326	256	238	326	256	230	376	256
	25%	320	389	320	290	389	320	289	449	320
	30%	383	466	384	338	466	384	339	538	384
	0%	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
	0.5%	15	115	15	16	115	16	15	135	15
	1%	29	133	29	29	133	30	28	127	29
	5%	143	273	144	144	273	144	138	288	144
Russian N	10%	288	425	288	287	425	288	273	427	288
	15%	432	596	432	430	596	432	408	623	432
	20%	575	729	576	574	729	576	545	773	576
	25%	719	884	720	709	884	720	665	951	720
	30%	861	1029	864	841	1029	864	760	1081	864
	0%	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
	0.5%	20	54	20	20	54	20	20	44	20
	1%	40	79	40	40	79	40	39	81	40
	5%	196	243	196	196	243	196	195	242	196
Finnish N	10%	390	466	392	392	466	392	391	466	392
	15%	588	664	588	588	664	588	575	674	588
	20%	782	895	784	781	895	784	751	930	784
	25%	978	1081	980	980	1081	980	891	1179	980
	30%	1175	1298	1176	1176	1298	1176	1012	1437	1176
	0%	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
	0.5%	22	65	22	22	65	22	22	66	22
	1%	43	103	43	44	103	44	43	94	43
	5%	211	261	211	210	261	212	210	268	211
German V	10%	420	480	421	416	480	422	417	465	421
	15%	630	694	631	614	694	632	627	718	631
	20%	840	910	841	809	910	842	819	931	841
	25%	1052	1141	1052	1012	1141	1052	994	1219	1052
	30%	1260	1356	1262	1206	1356	1262	1119	1478	1262
	0%	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
	0.5%	25	38	25	26	38	26	25	39	25
	1%	49	73	49	50	73	50	47	65	49
a	5%	245	275	245	246	275	246	235	268	245
Spanish V	10%	490	538	490	488	538	490	487	519	490
	15%	734	787	735	735	787	736	731	778	735
	20%	980	1038	980	979	1038	980	968	1035	980
	25%	1224	1291	1225	1225	1291	1226	1204	1313	1225
	30%	1468	1553	1470	1466	1553	1470	1400	1618	1470
	0%	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
	0.5%	100	289	100	100	289	100	99	176	100
	1%	198	375	199	200	375	200	196	260	199
	5%	993	1343	995	994	1343	996	977	1088	995
Finnish V	10%	1983	2301	1989	1987	2301	1990	1957	2114	1989
	15%	2978	3374	2983	2980	3374	2984	2914	3117	2983
	20%	3969	4409	3977	3975	4409	3978	3900	4186	3977
	25%	4956	5503	4971	4957	5503	4972	4827	5257	4971
	30%	5951	6484	5965	5951	6484	5966	5646	6271	5965

Table 3: Count of errors. "True Positive" column lists the count of errors which are artificial errors we introduce to the data and identified by the model as being erroneous. "Detected Error" column lists the number of inflected forms which the model detects as being erroneous, and the inflection model is trained with corrupted data by adding artificial errors at different amounts. "Artificial Error" column lists the number of artificial errors for each artificial error type we introduce to the original morphological data.