Morphology is not just a naive Bayes – UniMelb Submission to SIGMORPHON 2022 ST on Morphological Inflection

Andreas Scherbakov **Ekaterina Vylomova**
The University of Melbourne
andreas@softwareengineer.pro vylomovae@unimelb.edu.au

Abstract

The paper describes the Flexica team's submission to the SIGMORPHON 2022 Shared Task 1 Part 1: Typologically Diverse Morphological Inflection. Our team submitted a non-neural system that extracted transformation patterns from alignments between a lemma and inflected forms. For each inflection category, we chose a pattern based on its abstractness score. The system outperformed the non-neural baseline, the extracted patterns covered a substantial part of possible inflections. However, we discovered that such score that does not account for all possible combinations of string segments as well as morphosyntactic features is not sufficient for a certain proportion of inflection cases.

1 Introduction

Previous years' shared tasks on morphological reinflection demonstrated superior performance across a variety of typologically diverse languages, especially in high-resource setting (Cotterell et al., 2016, 2017, 2018; McCarthy et al., 2019; Vylomova et al., 2020; Pimentel et al., 2021). Still, in low-resource setting and languages with limited resources in which paradigms were only partially represented the accuracy numbers were much less optimistic (Vylomova et al., 2020; Pimentel et al., 2021). Recently, Goldman et al. (2022) experimented with the 2020 shared task data splitting it by lemmas and demonstrated the 30% accuracy drop on average among top-3 top ranked systems in that year's shared task. This motivated organizers of this year's shared task to focus on various aspects of morphological generalisation and conduct controlled experiments to evaluate systems' ability to predict inflected forms for unseen lemmas and morphosyntactic feature combinations.

In this paper, we describe a modification of our earlier model, Flexica (Scherbakov, 2020), that has been participated in the 2020 shared task (Vylomova et al., 2020).[1] We provide a summary of its modified version where we attempted to improve its pattern-based generalization ability. We added ability to reuse word forms observed at different combinations of grammatical tags. Also, we improved scoring mechanism to enable better fitting to rule-and-exception hierarchy which typically presents in a language, and to reduce noise in pattern selection.

2 Task Description

This year's shared task setting substantially differed from previous years in controlling the lemma and feature sets. More specifically, the training, development, and tests sets for the shared task were designed to assess various kinds of generalization. The shared task organizers considered four scenarios of overlap between the training and test sets : 1) both test lemma and feature set are observed in the training (but separately); 2) a test lemma is observed in the training set whereas the feature combination is entirely novel; 3) a feature combination is observed in the training set but the lemma is novel; 4) both a test pair's lemma and feature set are entirely novel and were not presented in the training set.

In addition, the training data sizes vary from 700 training instances in the small (low-resource) setting to 7,000 instance in the large (high-resource) setting. For some under-resourced languages the large setting contained fewer samples.

3 Data

3.1 Data Format

All shared task data are in UTF-8 and follow UniMorph annotation schema (Sylak-Glassman, 2016). Training and developments samples consist of a

[1] https://github.com/andreas-softwareengineer-pro/flexica

19th SIGMORPHON Workshop on Computational Research in Phonetics, Phonology, and Morphology, pages 240 - 246

lemma, an inflected (target) form, and its morphosyntactic features (tags). Test samples omit the target form.

3.2 Languages

The shared task covered morphological paradigms for 33 typologically diverse languages representing 11 language families: Arabic (Modern Standard), Assamese, Braj, Chukchi, Eastern Armenian, Evenki, Georgian, Gothic, Gujarati, Hebrew, Hungarian, Itelmen, Karelian, Kazakh, Ket, Khalkha Mongolian, Kholosi, Korean, Lamahalot, Low German, Ludic, Magahi, Middle Low German, Old English, Old High German, Old Norse, Polish, Pomak, Slovak, Turkish, Upper Sorbian, Veps, and Xibe.

4 Baseline Systems

As in previous years' shared tasks, two types of baseline systems were provided: neural and non-neural. The **non-neural** baseline aligns extracts suffixes and prefixes based on lemma–form alignments, later associating them with corresponding morpholosyntatic features (Cotterell et al., 2017, 2018). As the **neural** baseline, organizers provided a character-level adaptation of transformer (Wu et al., 2021).

5 Evaluation

The systems submitted to the shared task were evaluated in terms of test accuracy between predicted and gold forms. Besides the overall accuracy, four categories were distinguished in the analytic data provided by organizers. Depending on whether a test sample lemma has been seen in the training set, and whether an exact tag combination ("feature") has been seen in the training set, a test sample might fall into one of the following four categories: "Lemma Overlap", "Feature Overlap", "Neither Overlap", or "Both Overlap".

6 System Description

6.1 Training

We implemented a non-neural system (**`Flexica`**) where an inflected form is inferred from string-to-string transformation patterns observed in training samples. We produce multiple transformation patterns per each training sample. Those patterns differ in their level of abstractness and also depend on string-to-string alignments between a lemma and an inflected form. Later on, we also distinguish two types of patterns, namely a *string* pattern abd a *transformation* pattern. A *string* pattern is a string which may consist of concrete characters and wildcards, e.g. `"u①nd"` pattern for the word "understand". A *transformation* pattern is a triple $(lemma\_pattern, tag \rightarrow form\_pattern)$ which is produced from $(lemma, tag \rightarrow form)$ training samples by replacing certain character subsequences with wildcards. $lemma\_pattern$ and $form\_pattern$ share the same wildcards within a transformation pattern.

In order to produce *transformation patterns* for a given training sample we follow the stages:

1. Find the longest common substring for a lemma and its form. Introduce a wildcard (character subsequence) ① and replace the matching part by the wildcard symbol. For example, an inflection `("observe", V;3;SG` $\rightarrow$ `"observes")` produces a pattern `("①", V;3;SG` $\rightarrow$ `"①s")`. If there are multiple longest matches, we produce as many transformation pattern variants. For example, `("bring", V;PST` $\rightarrow$ `"brang")` will result in two patterns at this stage, `("①ing", V;PST` $\rightarrow$ `"①ang")` and `("bri①", V;PST` $\rightarrow$ `"bra①")`. We recursively apply the above procedure to the remaining concrete subsequences, finding longest matches and adding new wildcards until no more matching fragments are available.[2] While doing so, we never nest wildcards into each other. We also reject lemma patterns where two or more wildcards would be immediately adjacent, because it would lead to excessive ambiguity in further matching.

   Note: although the process described above may seem to be proliferating, just a single pattern is produced for a vast majority of inflection samples, as they usually have a single longest match. A notable exception are languages with templatic morphology.

2. Produce patterns with various character refinements. At this stage, we partially "surrender" longest matches found at the alignment stage. We replace some characters

[2] We apply an upper threshold for the number of wildcards specifying its as a hyperparameter (usually 2 or 3), which does not affect prediction accuracy.

in wildcard groups back to their concrete values that were observed in a training sample. Once a character is reverted to its concrete value, a wildcard that contained it may be split into two wildcard groups or even disappear. The latter happens whenever a wildcard standing for an empty substring is produced. We do such for $0, 1..CCL$ characters selected in all possible combinations, where CCL is a limit of the concrete characters.[3] Transformation patterns such as (`"(1)e"`, `V;3;SG` → `"(1)es"`), (`"(1)v(2)"`, `V;3;SG` → `"(1)v(2)s"`), (`"o(1)v(2)e"`, `V;3;SG` → `"o(1)v(2)es"`) constitute a non-exhaustive list of refinements for the pattern (`"(1)"`, `V;3;SG` → `"(1)s"`) produced for an (`"observe"`, `V;3;SG` → `"observe"`) sample.

We collect all unique patterns produced over a training corpus, finally constructing a trie database model in which data records are as follows: $l \rightarrow \{s \rightarrow \{t, c, d\}\}$ where l is a lemma pattern; s is an inflected form pattern; t is a grammatical tag combination; c is a number of training samples matching the transformation $(l, t\} \rightarrow s)$; d is a number of samples where lemma and tags match l and t, respectively, but the inflected form doesn't match s.

6.2 Inference

In order to predict an infected form for a $(lemma, tag)$ pair, our system finds all the transformation patterns that match the lemma (given any non-empty substring substitution for each wildcard group). Then it picks the transformation that yields the highest score. The score is hierarchical which means that a less significant score factor is considered if and only if all the factors of greater significance are in a tie. Here are the list of score factors, ordered by decreasing significance:

1. Penalty for the pattern abstractness, measured as count of characters substituted into wildcard groups. We include an extra "pad" character per group while calculating that sum;

2. Penalty for tag sets' mismatch (which is fixed per each mismatching tag) plus (optionally) a fixed "lump" amount for any two mismatching tag sets;[4]

3. *Representative* premium (optional), which is a fixed bonus assigned to transformations that are the most abstract while being correct representations of at least one training sample. This score component serves as a counterweight to the pattern abstractness score component described above. It may be seen as an adaptation of the idea of the most general paradigm (Hulden et al., 2014);

4. A (squashed) frequency f of transformation pattern occurrence in a training set for the given tag combination, minus double (squashed) frequency observed for alternative transformations for the same lemma pattern and tag combination.

7 Results

Tables 1 and 2 present accuracy across all the shared task's languages measured for the small and large settings, respectively. For `Flexica`, the column "B" stands for the basic option (without representative bonus), while the column "R" stands for the option with representative bonus. The official submission accuracy numbers are shown in the "Sb." column. Also, accuracy results for the non-neural and neural baselines ("BL") and best results across neural systems submitted to the shared task ("neural"/"max"), are presented for the reference.

We also explored some modifications to pattern scoring, but they did not affect performance much. In particular, we tried the following options:

- Increased penalty for impure patterns where different transformations were learnt for a given lemma pattern. The change resulted in approx. 1% accuracy increase for Middle Low German, although a nearly equal decrease happened in Old High German;

- We added an extra bonus for the exact match of grammatical tag combinations. Surprisingly, due to a notable sparseness of such combinations in the dataset we used, that

[3] In our officially reported results CCL is taken to be 3, because computations are too numerous for greater values. However, our observations suggest that this value is not sufficient, and increasing it enables better performance.

[4] We also considered using a variable tag-to-tag mismatch penalty which was proportional to a negative log-likelihood of tag interchangeability, but our experiments demonstrated lower accuracy for that option.

lang	non-neural					neural	
	Flexica				BL	max	BL
	B	R	Av.	Sb.			
ang	41	41	85	37	49	54	33
ara	31	31	70	32	65	66	22
asm	33	33	47	30	54	57	26
bra	55	56	82	58	55	60	57
ckt	21	21	29	10	6	21	13
evn	3	3	43	3	29	34	25
gml	27	27	92	26	42	56	22
goh	49	50	73	40	56	60	42
got	38	38	68	18	60	61	38
guj	47	47	61	47	39	66	48
heb	19	19	31	19	39	40	14
hsb	13	13	52	13	5	83	10
hsi	16	16	27	13	0	96	20
hun	26	26	58	25	65	61	23
hye	40	40	61	39	64	86	38
itl	30	30	53	31	34	34	28
kat	36	36	63	34	60	59	43
kaz	40	40	52	34	55	65	42
ket	21	21	42	18	10	35	32
khk	24	24	46	22	41	41	28
kor	32	31	57	30	23	50	28
krl	23	23	31	23	16	45	5
lud	88	87	91	88	46	87	88
mag	58	58	79	58	51	64	55
nds	29	29	62	31	25	50	16
non	35	35	71	39	55	52	30
pol	40	40	67	43	59	78	30
poma	29	29	49	29	51	50	22
sjo	55	55	90	65	58	76	67
slk	44	44	81	51	61	84	38
slp	7	7	51	8	15	30	5
tur	18	18	25	18	34	85	16
vep	20	20	41	20	35	42	21

Table 1: Accuracy (in %) measured in the small training condition. `B` - basic options; `R` - with a bonus score for "representative" patters; `Av.` - theoretical limit at a perfect pattern choice; `Sb.` - submitted version; `BL` - baseline; `max` - best among submitted systems

change produced no significant difference, except for a minor accuracy increase for Gothic and Georgian.

- Tag combinations in some UniMorph inflection data files may denote multiple options. For instance, multiple tags corresponding to the same category may be included into a single combination, in which any of them is meant to be equally suitable for producing a given inflected form. In order to meet that an alternative tagging format, we tried a modified tag mismatch penalty. Namely, an absence of a target tag in a learnt tag combination is interpreted as "one unit" of tag mismatch. This option yields approximately the same performance as the previous one described.

lang	non-neural					neural	
	Flexica				BL	max	BL
	B	R	Av.	Sb.			
ang	46	47	91	41	61	64	43
ara	37	37	79	37	78	75	26
asm	35	35	63	34	76	75	31
evn	3	3	70	3	57	57	25
got	44	44	80	21	72	73	46
heb	29	29	45	28	48	51	20
hun	34	34	75	32	77	74	37
hye	43	42	66	42	69	93	44
kat	32	32	75	45	87	83	45
kaz	40	40	52	34	55	65	42
khk	31	31	50	23	49	49	38
kor	33	34	63	33	56	54	32
krl	36	37	53	37	27	64	5
lud	83	78	93	89	52	89	89
non	41	41	86	47	84	87	37
pol	50	50	84	52	69	90	43
poma	34	34	65	33	63	61	24
slk	49	49	87	58	70	93	47
tur	36	36	53	35	39	94	36
vep	30	30	60	30	48	62	32

Table 2: Accuracy (in %) measured in the large training condition. `B` - basic options; `R` - with a bonus score for "representative" patters; `Av.` - theoretical limit at a perfect pattern choice; `Sb.` - submitted version; `BL` - baseline; `max` - best among submitted systems

As exact tag combinations were significantly sparse in training and test sets, the majority of mispredictions can be attributed to failures to inference tag interchangeability. Indeed, in most cases of misprediction a correct transformation was available in the learnt model, but it deemed to be irrelevant due to low "similarity" between the learnt tag combination and the target one. The "Av." column in Tables 1 and 2 shows the percentage of test samples where a correct transformation was available for the model. It tells the upper bound of accuracy that our system would have if the pattern selection mechanism worked perfectly.

8 Discussion

The system we explored in this paper relies on two simple hypotheses. According to the first one, a choice of inflection paradigm in most cases may be associated with some distinctive subsequence of characters in a lemma. The second hypothesis claims existence of a hierarchy of rules and exceptions in most languages, where each exception domain is fenced by a more concrete character pattern than one associated with an embracing general inflection rule. We note that our current approach only admits a very restrictive meaning for such a "concreteness", namely, the number of concrete characters in a template. Due to this substantial limitation, we only consider an *approximate* split of rule-specific domains.

While the analysis of predictions suggests this approach is generally reasonable, the distinguishing of relevant patterns from noise is challenging. Certain information-based criteria such as entropy, cross-entropy and the like did not work, mainly due to specific patterns being sparsely distributed in the dataset (especially small ones), so that majority of highly concrete patterns peaked the distribution of inflection transformations. On the other hand, many relevant generic patterns demonstrate rather disperse distributions due to numerous exceptions. As a result, it is not possible to easily link the entropy to the relevancy. We intentionally avoided imposing extra biases toward "known" common language rules in order to focus our exploration on the system's learning capability itself. Unfortunately, we have not yet found universal enough criteria to assess pattern relevance against inflection rules, so in this aspect the system should be considered as a work in progress. We attempted "promotion" of one maximally abstract pattern per training sample, that match the given sample and does not contradict any other observed samples. The underlying hypothesis was that every inflection paradigm is probably justified by a single "cause", where a "cause" in our restricted context stands for a distinct character pattern for a lemma. Therefore, it should be reasonable to restrict prediction selection to those transformation patterns which were proven to be correctly representing at least one training sample in the most generic way. However, our experiments disproved such an approach, because, as we already said above, relevance criteria based on distribution purity are fundamentally flawed.

Our system operates at character level without considering more generic classes of sub-patterns. However, it did not seem to be a significant issue in most languages. In other words, patterns needed for correct inflection have usually been successfully learnt in most languages (still, non necessarily with the same grammatical tag). However, there are numerous languages where correct patterns cannot be found for a large fraction of examples; this severely jeopardised the respective prediction rates. Besides the "genuinely" high morphological complexity of languages such as Veps, there also occurred some "technical" reasons for the pattern match missing, such as non-standardized scripting of spoken languages (Pomak, Evenki). It is our system's lack of a mechanism for the affix concatenation which was responsible for inferior results observed in agglutinative languages like Turkish of Hungarian, especially in their low-resource settings.

In the 2022 shared task, we faced a new challenge of extreme sparsity of grammatical tag combinations. A separate model per learnt tag combination does not work in such an environment. We allowed using transformation patterns observed at grammatical tag combinations different from a requested one, with a score penalty proportional to the number of different "atomic" tags (morphosyntatic features). From the inflection perspective, many grammatical tags are not as significant for a correct prediction as others are. This inspired us to use variable penalty per tag substitutions, which was proportional to a log-likelihood of observing the same transformation regardless whether a given tag is present, as measured over all learn transformation patterns, without considering other tags. For instance, in Polish, the animacy does not affect inflection paradigms much, and ignoring it would significantly increase the average accuracy of inference. However, to our surprise, according to the likelihood, some case tag substitutions occurred to be better candidates for being ignored. For instance, the dative and the instrumental cases produce same forms for a majority of Polish feminine nouns, therefore our predictor frequently chooses `INS` $\rightarrow$ `DAT` substitution, which is usually incorrect beyond the feminitive (instead of correct `ANIM` $\rightarrow$ `INAN`). Thus, such Bayesian approach, that considers tags independently, even failed to outperform a simplistic technique based on the "edit distance" between tag combinations. We did not yet consider more complex sub-combinations of tags, still the results definitely suggest one to do that way.

An excessive number of generated patterns is another challenge which yet needs to be addressed. Currently, our system unrolls all the combinations of concrete characters in lemma patterns until ultimately discriminative ones are found over a training set. This leads to huge proliferation of noisy patterns of no extra value. In practice, this fact prevents the system from considering longer subsequences of concrete characters where those subsequences could really help to delimit paradigm domains.

Summarizing our impressions from the experiments, we suggest that the system is primarily interesting as it prototypes a simple but efficient approach to the conversion of a sequence-to-sequence task into a "plain" classification task. In this view, further enhancements of the system may be broken into two separate directions. The first one concerns the pattern matching mechanism which would become less consuming, more generalized, based on incrementally collected "cues" (and, in such a way, borrowing features of the "soft attention"). Another direction, which is less specific, would be an exploration of better classification models to be used. Also, the principally optimistic results obtained in our experiments inspire us to attempt expanding the proposed multi-pattern approach into other sequence-to-sequence tasks beyond the re-inflection one.

9 Conclusion

We developed a non-neural system for morphological inflection. We submitted it to the SIGMORPHON 2022 shared task 1, part 1. The system outperformed the non-neural baseline, still we discovered a fundamental insufficiency of simplistic approaches that rely on observed probabilities of particular transformation patterns.

Acknowledgements

We are deeply thankful to all the organizers of SIGMORPHON workshop and its re-inflection shared task, and to all the contributors to the UniMorph database, for the opportunity to participate in this inspirational contest and to carry out insightful experiments on amazingly diverse morphological corpora.

References

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

Omer Goldman, David Guriel, and Reut Tsarfaty. 2022. (un)solving morphological inflection: Lemma overlap artificially inflates models' performance. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 864–870, Dublin, Ireland. Association for Computational Linguistics.

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

Arya D. McCarthy, Ekaterina Vylomova, Shijie Wu, Chaitanya Malaviya, Lawrence Wolf-Sonkin, Garrett Nicolai, Christo Kirov, Miikka Silfverberg, Sebastian J. Mielke, Jeffrey Heinz, Ryan Cotterell, and Mans Hulden. 2019. The SIGMORPHON 2019 shared task: Morphological analysis in context and cross-lingual 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.

Tiago Pimentel, Maria Ryskina, Sabrina J. Mielke, Shijie Wu, Eleanor Chodroff, Brian Leonard, Garrett Nicolai, Yustinus Ghanggo Ate, Salam Khalifa, Nizar Habash, Charbel El-Khaissi, Omer Goldman, Michael Gasser, William Lane, Matt Coler, Arturo Oncevay, Jaime Rafael Montoya Samame, Gema Celeste Silva Villegas, Adam Ek, Jean-Philippe Bernardy, Andrey Shcherbakov, Aziyana Bayyr-ool, Karina Sheifer, Sofya Ganieva, Matvey Plugaryov, Elena Klyachko, Ali Salehi, Andrew Krizhanovsky, Natalia Krizhanovsky, Clara Vania, Sardana Ivanova, Aelita Salchak, Christopher Straughn, Zoey Liu, Jonathan North Washington, Duygu Ataman, Witold Kieraś, Marcin Woliński, Totok Suhardijanto, Niklas Stoehr, Zahroh Nuriah, Shyam Ratan, Francis M. Tyers, Edoardo M. Ponti, Grant Aiton, Richard J. Hatcher, Emily Prud'hommeaux, Ritesh Kumar, Mans Hulden, Botond Barta, Dorina Lakatos, Gábor Szolnok, Judit Ács, Mohit Raj, David Yarowsky, Ryan Cotterell, Ben Ambridge, and Ekaterina Vylomova. 2021. SIGMORPHON 2021 shared task on morphological reinflection: Generalization across languages. In *Proceedings of the 18th SIGMORPHON Workshop on Computational Research in Phonetics, Phonology, and Morphology*, pages 229–259, Online. Association for Computational Linguistics.

Andreas Scherbakov. 2020. The UniMelb submission to 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*, pages 177–183.

John Sylak-Glassman. 2016. The composition and use of the universal morphological feature schema (unimorph schema). *Johns Hopkins University*.

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.