


Unsupervised Paradigm Clustering Using Transformation Rules

Changbing Yang 

 University of Colorado Boulder

changbing.yang@colorado.edu

Garrett Nicolai 

 University of British Columbia

first.last@ubc.ca

Miikka Silfverberg 

Abstract

This paper describes the submission of the CU-UBC team for the SIGMORPHON 2021 Shared Task 2: Unsupervised morphological paradigm clustering. Our system generates paradigms using morphological transformation rules which are discovered from raw data. We experiment with two methods for discovering rules. Our first approach generates prefix and suffix transformations between similar strings. Secondly, we experiment with more general rules which can apply transformations inside the input strings in addition to prefix and suffix transformations. We find that the best overall performance is delivered by prefix and suffix rules but more general transformation rules perform better for languages with templatic morphology and very high morpheme-to-word ratios.

1 Introduction

Supervised sequence-to-sequence models for word inflection have delivered impressive results in the past few years and a number of shared tasks on supervised learning of morphology have helped to raise the state of the art of this task (Cotterell et al., 2016, 2017, 2018; McCarthy et al., 2019; Vylomova et al., 2020). In contrast, unsupervised approaches to morphology have received far less attention in recent years. Nevertheless, the question of whether the morphological system of a language can be discovered from raw text data alone is certainly an interesting one.

This paper describes the submission of the CU-UBC team for the SIGMORPHON 2021 Shared Task 2: Unsupervised morphological paradigm clustering (Wiemerslage et al., 2021).¹ The objective of this task is to group

¹github.com/changbingY/Sigmorph-2021-task2

the distinct inflected forms of lexemes occurring in a corpus into morphological paradigms. Figure 1 illustrates the task.

Our system generates paradigms using *morphological transformation rules* which are discovered from raw data. As an example, consider the rule **ed** → **ing**, which maps an English past tense verb form like **walked** into the present participle **walking**. In this paper, we use *regular expressions of symbol-pairs* (that is, regular relations) in the well-known Xerox formalism (Beesley and Karttunen, 2003) to denote rules: for example, **?+ e:i d:n 0:g**. These rule can be applied using composition of regular relations:

[w a l k e d] .o. [?+ e:i d:n 0:g]

will result in an output form **w a l k i n g**. We cluster forms into the same paradigm if we can find morphological transformation rules which map one of the forms into the other. Our approach is illustrated in Figure 2.

We experiment with two methods for discovering rules, described in Section 3.3. Our first approach is inspired by work on morphology discovery by Soricut and Och (2015), who generate *prefix and suffix transformations* between similar strings. This idea closely parallels our approach for extracting rules. Unlike Soricut and Och (2015), however, we do not utilize word embeddings when extracting rules due to the very small size of the shared task datasets. In addition to prefix and suffix rules, we also experiment with more general *discontinuous transformation rules* which can apply transformations to infixes as well as prefixes and suffixes. For example, the rule

?+ i:0 ?+ e:i ?+ 0:t

would transform the input form **gidem** (‘to bite’ in Maltese) to **gdimt**. Our results

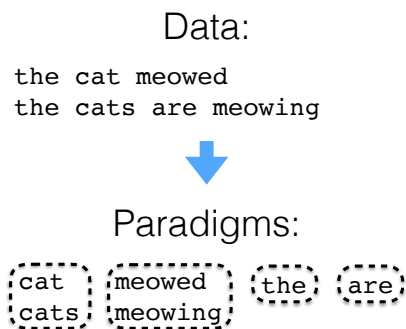


Figure 1: The unsupervised paradigm clustering task.

demonstrate that prefix and suffix rules deliver stronger performance for most languages in the shared task dataset but our more general transformations rules are beneficial for templatic languages like Maltese and languages with a high morpheme-to-word ratio like Basque.

2 Related Work

The unsupervised paradigm clustering task is closely related to the 2020 SIGMORPHON shared task on unsupervised morphological paradigm completion (Kann et al., 2020). However, paradigm clustering systems do not infer missing forms in paradigms. Our system resembles the baseline system for the paradigm completion task (Jin et al., 2020) which also extracts transformation rules, however, in the form of edit trees (Chrupala et al., 2008).

Several approaches to unsupervised or minimally supervised morphology learning, which share characteristics with our system, have been proposed. Our rules are essentially identical to the FST rules used by Beemer et al. (2020) for the task of supervised morphological inflection. Likewise, Durrett and DeNero (2013) and Ahlberg et al. (2015) both extract inflectional rules after aligning forms from known paradigms. Yarowsky and Wicentowski (2000) also generate rules for morphological transformations but their system for minimally supervised morphological analysis requires additional information in the form of a list of morphemes as input.

Erdmann et al. (2020) present a task called *the paradigm discovery problem* which is quite similar to the unsupervised paradigm clustering task. In their formulation of the task, inflected forms are clustered into paradigms and corre-

sponding forms in distinct paradigms (like all plural forms of English nouns) are clustered into cells. Their benchmark system is based on splitting every form into a (potentially discontinuous) base and exponent, where the base is the longest common subsequence of the forms in a paradigm and the exponent is the residual of the form. They then maximize the base in each paradigm while minimizing the exponents of individual forms.

3 Methods

This section describes how we extract rules from the dataset and apply them to paradigm clustering. We also describe methods for filtering out extraneous forms from generated paradigms.

3.1 Baseline

As a baseline, we use the character n-gram clustering method provided by the shared task organizers (Wiemerslage et al., 2021). Here all forms sharing a given substring of length n are clustered into a paradigm. Duplicate paradigms are removed. The hyperparameter n can be tuned on validation data if such data is available (we use $n = 5$ in all our experiments).

3.2 Transformation rules

Our approach builds on the baseline paradigms discovered in the previous step. We start by extracting transformation rules between all word forms in a single baseline paradigm. For each pair of strings like **dog** and **dogs** belonging to a paradigm, we generate a rule like $?+ 0:s$ which translates the first form into the second one. From a paradigm of size n , we can therefore extract $n^2 - n$ rules—one for each ordered pair of distinct word forms. Preliminary experiments showed that large baseline paradigms tended to generate many incorrect rules which did not represent genuine morphological transformations. We, therefore, limited rule-discovery to paradigms spanning maximally 20 forms.

After generating transformation rules, we compute rule-frequency over all baseline paradigms and discard rare rules which are unlikely to represent genuine morphological transformations (the minimum threshold for rule frequency is a hyperparameter). The remaining rules are then applied iteratively to

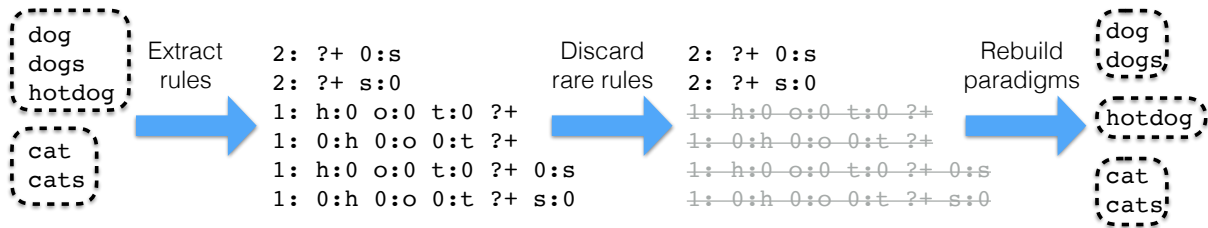


Figure 2: A schematic representation of our approach. We start by generating preliminary paradigms using the baseline method. We then extract transformation rules for each word pair in our paradigms noting how many times each unique rule occurred. For example, here both (**dog**, **dogs**) and (**cat**, **cats**) result in a rule $?* 0:s$ which therefore has count 2. Subsequently, we discard rare rules like $h:0 o:0 t:0 ?*$ which are unlikely to represent genuine morphological transformations. We then use the remaining rules to reconstruct our morphological paradigms as explained in Section 3.3.

our datasets to construct paradigms. We experiment with two rule types which are described below.

3.2.1 Prefix and Suffix Rules

Our first approach to rule-discovery is based on identifying a contiguous word stem shared by both forms. The stem is defined as the longest common substring of the forms. We split both forms into a prefix, stem and suffix. The morphological transformation is then defined as a joint substitution of a prefix and suffix. For example, given the German forms **acker+n** and **ge+acker+t** (German ‘to plow’), we would generate a rule:

$$0:g 0:e ?+ n:t$$

As mentioned above, these rules are extracted from paradigms generated by the baseline system.

We also experiment with a more restricted form of these rules in which only suffix transformations are allowed. While this limits the possible transformations, it will also result in fewer incorrect rules and may, therefore, deliver better performance for languages which are predominantly suffixing.

3.2.2 Discontinuous rules

Even though prefix and suffix transformations are adequate for representing morphological transformations in many languages, they fail to derive the appropriate generalizations for languages with templatic morphology like Maltese (which was included among the development languages). For example, it is impossible to identify a contiguous stem-like unit spanning more than a single character for the Maltese forms **gidem** ‘to bite’ and **gdimt**. We need

a rule which can apply transformations inside the input string:

$$?+ i:0 ?+ e:i ?+ 0:t$$

Like prefix and suffix rules, discontinuous rules are generated from baseline paradigms. Unlike prefix and suffix rules, however, discontinuous rules require a character-level alignment between the input and output string. To this end, we start by generating a dataset consisting of all string pairs like (**dog**, **dogs**) and (**hotdog**, **dog**), where both strings belong to the the same paradigm. We then apply a character-level aligner based on the iterative Markov chain Monte Carlo method to this dataset.² Using this method, we can jointly align all string pairs in the baseline paradigms. This is beneficial because the MCMC aligner will prefer common substitutions, deletions and insertions over rare ones.³ which enforces consistency of the alignment over the entire dataset. This in turn can help us find linguistically motivated transformation rules.

Character-level alignment results in pairs:

$$\begin{array}{r} \text{INPUT:} \quad d \ o \ g \ 0 \\ \text{OUTPUT:} \quad d \ o \ g \ s \\ \hline \text{INPUT:} \quad h \ o \ t \ d \ o \ g \\ \text{OUTPUT:} \quad 0 \ 0 \ 0 \ d \ o \ g \end{array}$$

Each symbol pair in the alignment represents one of the following types: (1) an identity pair $x:x$, (2) an insertion $0:x$, (3) a deletion $x:0$, or (4) a substitution $x:y$. In order to convert a pair of aligned strings into a transformation

²This aligner was initially used for the baseline system in the 2016 iteration of the SIGMORPHON shared task (Cotterell et al., 2016).

³This is a consequence of the fact that the algorithm iteratively maximizes the likelihood of the alignment for each example given all other examples in the dataset.

rule, we simply replace all contiguous sequences of identity pairs with $?+$. For the alignments above, we get the rules: $?+ 0:s$ and $h:0 o:0 t:0 ?+$.

3.3 Iterative Application of Rules

After extracting a set of rules from baseline paradigms, we discard the baseline paradigms. We then construct new paradigms using our rules. We start by picking a random word form w from the dataset. We then form the paradigm P for w as the set of all forms in our dataset which can be derived from w by applying our rules iteratively. For example, given the form **eats** and the rules:

$$?+ s:0 \text{ and } ?+ 0:i 0:n 0:g$$

the paradigm of **eats** would contain both **eat** (generated by the first rule) and **eating** (generated by the second rule from **eats**) provided that both of these forms are present in our original dataset. All forms in P are removed from the dataset and we then repeat the process for another randomly sampled form in the remaining dataset. This continues until the dataset is exhausted. The procedure is sensitive to the order in which we sample forms from the dataset but exploring the optimal way to sample forms falls beyond the scope of the present work.

For prefix and suffix rules, we limit rule application to a single iteration because this delivered better results in practice. Applying rules iteratively tended to result in very large paradigms. For discontinuous rules, we do apply rules iteratively.

3.4 Filtering Paradigms

According to our preliminary experiments, many large paradigms generated by transformation rules contained word forms which were morphologically unrelated to the other forms in the paradigm. To counteract this, we experimented with three strategies for filtering out individual extraneous forms from generated paradigms: the degree test, the rule-frequency test and the embedding-similarity test. Forms which fail all of our three tests are removed from the paradigm.⁴

⁴These filtering strategies are applied to paradigms containing > 20 forms. This threshold was determined based on examining the output clusters for the development languages.

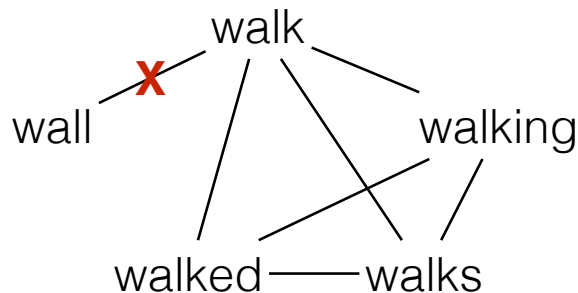


Figure 3: Given the candidate paradigm $\{\mathbf{walk}, \mathbf{wall}, \mathbf{walking}, \mathbf{walked}, \mathbf{walks}\}$, we can form a graph where two word forms are connected if a rule like $?+ 0:e 0:d$ derives one of the forms like **walked** from the other one **walk**. We experiment with filtering out forms which have low degree in this graph since those are more likely to be spurious additions resulting from rules like $?+ 1:k$ in the example, which do not capture genuine morphological regularities. In this example, **wall** might be filtered out because it has low degree one compared to all other forms which have degree three.

If we first generate all paradigms and then filter out extraneous forms, we will be left with a number of forms which have not been assigned to a paradigm. In order to circumvent this problem, we apply filtering immediately after generating each individual paradigm. Forms which are filtered out from the paradigm are placed back into the original dataset. They can then be included in paradigms which are generated later in the process.

Degree test Our morphological transformation rules induce dependencies and therefore a graph structure between the forms in a paradigm as demonstrated in Figure 3. Within each paradigm, we calculate the degree of a word in the following way: For each attested word w in the generated paradigm, its degree is the number of forms w' in the paradigm for which we can find a transformation rule mapping $w \rightarrow w'$. We increment the degree if there is at least one edge between words w and w' in the paradigm (the number of distinct rules mapping form w to w' is irrelevant here as long as there is at least one). *If the degree of a word is less than a third of the paradigm size, the word fails the degree test.*

Rule-Frequency test Some rules like $?+ e:i d:n 0:g$ for English represent genuine inflectional transformations and will therefore

occur often in our datasets. Others, like the rule $?* 1:k$ in Figure 3, instead result from coincidence, and will usually have low frequency. We can, therefore, use rule frequency as a criterion when identifying extraneous forms in generated paradigms. We examine the cumulative frequency of all rules applying to the form in our paradigm. *If this frequency is lower than the median cumulative frequency in the paradigm, the form fails the rule-frequency test.*

Embedding-similarity test If a word fails to pass the degree and the rule frequency tests, we will measure the semantic similarity of the given form with other forms in the paradigm. To this end, we trained FastText embeddings (Bojanowski et al., 2017) and calculated cosine similarity between embedding vectors as a measure of semantic relatedness.⁵ We start by selecting two reference words in the paradigm which have high degree (at least 50% of the maximal degree) and whose cumulative rule frequency is above the paradigm’s median value. We then compute their cosine similarity as a reference point r . For all other words in the paradigm, we then compare their cosine similarity r' to one of the reference forms. *Forms fail the embedding-similarity test if $r' < 0.5$ and $r - r' > 0.3$.*

4 Experiments and Results

In this section, we describe experiments on the shared task development and test languages.

4.1 Data and Resources

The shared task uses two data resources. Corpus data for the four development languages (Maltese, Persian, Russian and Swedish) and nine test languages (Basque Bulgarian, English, Finnish, German, Kannada, Navajo, Spanish and Turkish) are sourced from the Johns Hopkins Bible Corpus (McCarthy et al., 2020b). For most of the languages, complete Bibles were provided but for some of them, we only had access to a subset (see Wiemerslage et al. (2021) for details). Gold standard paradigms were automatically generated using the Uni-morph 3.0 database (McCarthy et al., 2020a).

⁵We train 300-dimensional embeddings with context window 3 and use character n -grams of size 3-6.

4.2 Experiments on validation languages

Since our transformation rules are generated from paradigms discovered by the baseline system, which contain incorrect items, it is to be expected that some incorrect rules are generated. We filter out infrequent rules, as they are less likely to represent genuine morphological transformations. For prefix and suffix rules (i.e., PS), we experimented with including the top 2000 (PS-2000), 5000 (PS-5000), and all rules (PS-all), as measured by rule-frequency. Additionally, we present experiments using a system which relies exclusively on suffix transformations including all of them regardless of frequency (S-all). For discontinuous rules (D), we used lower thresholds because our preliminary experiments indicated that incorrect generalizations were a more severe problem for this rule type. We selected the 200 (D-200), 300 (D-300), and 500 (D-500) most frequent rules, respectively. Results with regard to best-match F1 score (see Wiemerslage et al. (2021) for details) are shown in Table 1.

According to the results, all of our systems outperform the baseline system by at least 25.53% as measured using the mean best match F1 score. Plain suffix rules (S-all) provide the best performance with a mean F1 score of 65.41%, followed by other affixal systems (PS-2000, PS-5000 and PS-all). On average, discontinuous rules (D-200, D-300 and D-500) are slightly less-successful, but they deliver the best performance for Maltese. Table 1 demonstrates that simply increasing the number of rules does not always contribute to better performance—the optimal threshold varies between languages.

As explained in Section 3.4, we aim to filter out extraneous forms from overly-large paradigms. We applied this approach to discontinuous rules with a 500 threshold. Results are shown in Table 2. As the table shows, a filtering strategy can offer very limited improvements. Most of the languages do not benefit from this approach and even for languages which do, the gain is miniscule. Due to their very limited effect, we did not apply filtering strategies to test languages.

	Maltese	Persian	Portuguese	Russian	Swedish	Mean
Baseline	29.07	30.04	34.15	36.30	43.62	34.64
PS-2000	35.41	50.17	65.53	81.20	81.14	62.69
PS-5000	36.81	50.40	71.33	81.96	79.82	64.06
PS-all	40.67	53.15	76.63	75.39	72.46	63.66
S-all	30.32	52.69	82.67	80.65	80.74	65.41
D-200	42.99	54.65	66.86	70.38	68.76	60.73
D-300	42.99	53.64	69.38	72.33	67.14	61.10
D-500	45.05	51.82	66.37	75.26	62.30	60.16

Table 1: F1 Scores for each of the model types on all development languages. The best F1 scores are in bold.

	Maltese	Persian	Portuguese	Russian	Swedish	Mean
Baseline	29.07	30.04	34.15	36.30	43.62	34.64
D-500	45.05	51.82	66.37	75.26	62.30	60.16
Filter	45.05	51.82	66.45	75.26	62.30	60.18

Table 2: F1 score for Discontinuous rules systems and Filtering systems across five validation languages.

4.3 Experiments on Test Languages

Results for the test languages are presented in Table 3. We find that all of our systems surpassed the baseline results by at least 23.06% in F1 score. The prefix and suffix system using all of the suffix rules displays the best performance with an F1 score of 66.12%. Among the discontinuous systems, the system with a threshold of 500 has the best results. On average, the affixal systems outperform the discontinuous ones. In particular, these methods perform best on languages which are known to be predominantly suffixing, such as English, Spanish, and Finnish. Contrarily, discontinuous rules deliver the best performance for Navajo—a strongly prefixing language. Discontinuous rules also result in the best performance for Basque, which has a very high morpheme-to-word ratio.

In order to better understand the behavior of our systems, we analyzed the distribution of the size of generated paradigms for prefix and suffix systems as well as discontinuous systems. Results for selected systems are shown in Figure 4. We conducted this experiment for the overall best system (S-all), as well as the best discontinuous system (D-500). Both systems follow the same overall pattern: large paradigms are rarer than smaller ones and the frequency drops very rapidly with increasing paradigm size. The majority of generated paradigms have sizes in the range 1-5. Although the tendency is similar for

suffix rules and discontinuous rules, discontinuous rules tend to generate more paradigms of size 1. In contrast to the paradigms generated by our systems, the frequency of gold standard paradigms drops far slower as the paradigms grow. For example, for Finnish and Kannada, paradigms containing 10 forms are still very common. The only language where the distribution generated by our systems very closely parallels the gold standard is Spanish. For all other languages, our systems very clearly over-generate small paradigms.

5 Discussion and Conclusions

Paradigm construction can suffer from two main difficulties: overgeneralization, and underspecification. In the former, paradigms are too generous when adding new members. Consider, for example, a paradigm headed by “sea”. We would want to include the plural “seas”, but not the unrelated words “seal”, “seals”, “undersea”, etc. Contrarily, a paradigm selection algorithm that is overly selective will result in a large number of small paradigms - less than ideal in a morphologically-dense language.

Considering the results described in the previous section, we note that our two best models skew towards conservatism - they prefer smaller paradigms. This is likely an artifact of our development cycle - we found that the baseline preferred large paradigms, often capturing derivational features, or even circumstantial

	English	Navajo	Spanish	Finnish	Bulgarian	Basque	Kannada	German	Turkish	Mean
Baseline	51.49	33.25	38.83	28.97	38.89	21.48	23.79	38.22	25.23	33.35
PS-2000	83.89	48.69	77.71	52.60	73.50	25.81	42.35	74.49	46.80	58.42
PS-5000	81.16	48.69	79.60	57.88	74.14	29.03	47.47	74.27	51.26	60.39
PS-all	76.41	48.69	76.94	66.03	69.50	29.03	57.71	65.26	60.97	61.17
S-all	88.68	42.48	83.21	73.42	76.96	29.03	59.34	74.18	67.80	66.12
D-200	76.93	58.45	66.05	50.68	70.48	26.19	40.57	70.26	48.05	56.41
D-300	73.23	59.36	69.46	53.66	69.39	26.19	43.71	68.52	51.00	57.17
D-500	69.33	61.66	69.92	56.51	63.23	33.33	46.94	62.54	53.24	57.41

Table 3: F1 Scores for each of the model types on all test languages. The best F1 scores are in bold.

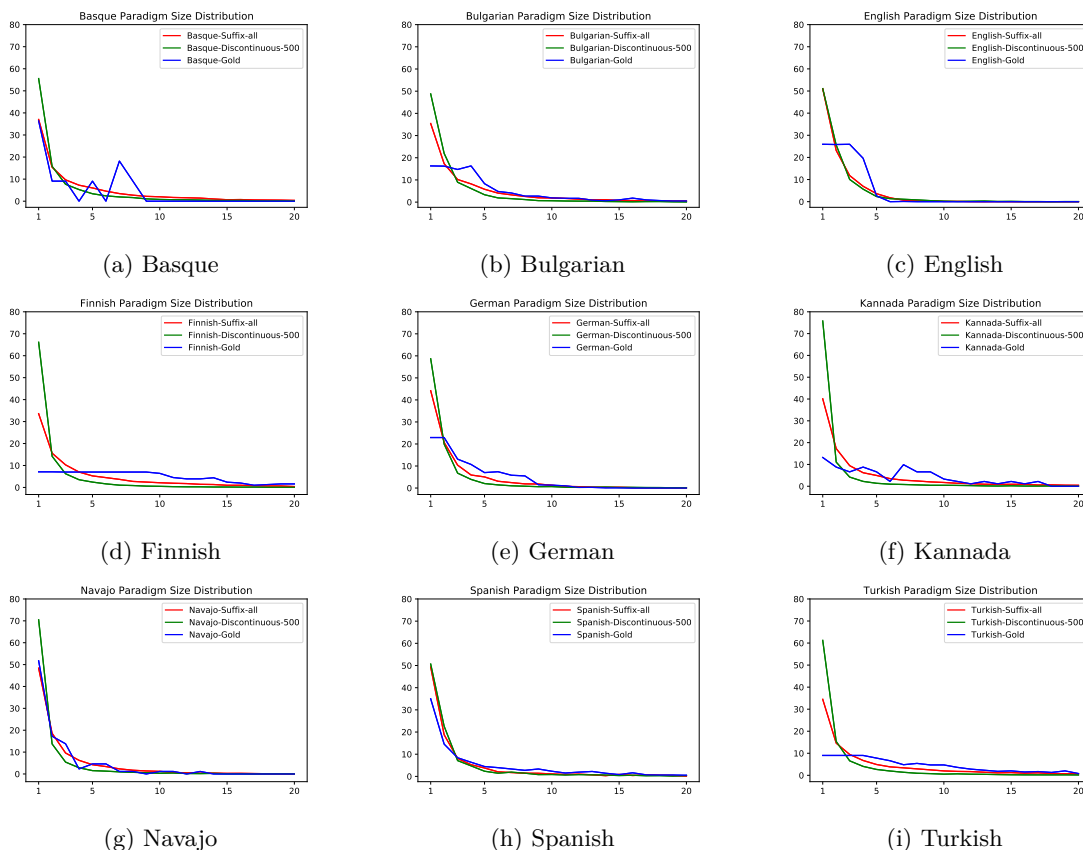


Figure 4: Paradigm size distribution across nine test languages. The x axis stands for paradigm size ranging from 1 to 20. The y axis shows the percentage of each paradigm size accounts for among all paradigms the system generates.

string similarities, when clustering paradigms. Much of our focus was thus on limiting rule application only to those rules we could be certain were genuine. Unfortunately, this means that many words are excluded, residing in singleton paradigms.

Our methods were also affected by the choice of development languages. Of these languages, only one (Persian) is agglutinating, and none of the authors can read the script, so it had a smaller impact on the evolution of our methods. We believe that several languages —namely, Finnish, Turkish, and Basque— could have benefited from iterative rule application; however, the iterative process was not selected after seeing a degradation (due to overgeneralization) on the development languages.

It is also worth discussing two outliers in our system selection. Our suffix-first model performed very well on all of the development languages except Maltese. This is not surprising, given its templatic morphology. Maltese inspired the creation of our discontinuous rule set, and indeed, these rules outperformed the suffixes for Maltese. Switching to the test languages, we see that this model has higher performance for Navajo and Basque —two languages that are rarely described as templatic. We observe, however, that both languages make heavy use of *prefixing*. Note in Table 2 that including prefixes (PS-All) significantly improves Navajo: the only language to see such a benefit. Likewise, Navajo also has significant stem alternation, which may be benefiting from discontinuous rule sets. Basque is trickier - it does not improve simply from including prefixal rules. Upon closer inspection, we observe that much Basque prefixation more closely resembles *circumfixation*: the stem has a prefixal vowel to indicate tense, which is jointly applied with inflectional suffixes. One round of rule application - even if it includes both suffixes and prefixes, appears to be insufficient.

There is still plenty of ground to be covered, with the mean F1 score below 70%. We believe that the next step lies in re-establishing a bottom-up construction for those paradigms that our methods currently separate into small sub-paradigms. Our methods predict roughly twice to 3 times as many singleton paradigms as exist in the gold data, and there is not signifi-

cant rule support to combine them. Possible areas for exploration include iterative rule extraction on successively more correct paradigms, or the incorporation of a machine learning element that can predict missing forms.

In this paper, we have presented a method for automatically building inflectional paradigms from raw data. Starting with an n -gram baseline, we extract intra-paradigmatic rewrite rules. These rules are then re-applied to the corpus in a discovery process that re-establishes known paradigms. Our methods prove very competitive, with our best model finishing within 2% of the best submitted system.

References

- Malin Ahlberg, Markus Forsberg, and Mans Hulden. 2015. Paradigm classification in supervised learning of morphology. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1024–1029.
- Sarah Beemer, Zak Boston, April Bukoski, Daniel Chen, Princess Dickens, Andrew Gerlach, Torin Hopkins, Parth Anand Jawale, Chris Koski, Akanksha Malhotra, Piyush Mishra, Saliha Muradoglu, Lan Sang, Tyler Short, Sagarika Shreevastava, Elizabeth Spaulding, Testumichi Umada, Beilei Xiang, Changbing Yang, and Mans Hulden. 2020. [Linguist vs. machine: Rapid development of finite-state morphological grammars](#). In *Proceedings of the 17th SIGMORPHON Workshop on Computational Research in Phonetics, Phonology, and Morphology*, pages 162–170, Online. Association for Computational Linguistics.
- Kenneth R Beesley and Lauri Karttunen. 2003. Finite-state morphology: Xerox tools and techniques. *CSLI, Stanford*.
- Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2017. Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, 5:135–146.
- Grzegorz Chrupala, Georgiana Dinu, and Josef van Genabith. 2008. [Learning morphology with Morfette](#). In *Proceedings of the Sixth International Conference on Language Resources and Evaluation (LREC'08)*, Marrakech, Morocco. European Language Resources Association (ELRA).
- Ryan Cotterell, Christo Kirov, John Sylak-Glassman, Géraldine Walther, Ekaterina Vylomova, Arya D McCarthy, Katharina Kann, Se-

- bastian J Mielke, Garrett Nicolai, Miikka Silfverberg, et al. 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.
- Ryan Cotterell, Christo Kirov, John Sylak-Glassman, Géraldine Walther, Ekaterina Vylomova, Patrick Xia, Manaal Faruqui, Sandra Kübler, David Yarowsky, Jason Eisner, et al. 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.
- 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.
- Greg Durrett and John DeNero. 2013. Supervised learning of complete morphological paradigms. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1185–1195.
- Alexander Erdmann, Micha Elsner, Shijie Wu, Ryan Cotterell, and Nizar Habash. 2020. [The paradigm discovery problem](#). In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7778–7790, Online. Association for Computational Linguistics.
- Huiming Jin, Liwei Cai, Yihui Peng, Chen Xia, Arya McCarthy, and Katharina Kann. 2020. [Unsupervised morphological paradigm completion](#). In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6696–6707, Online. Association for Computational Linguistics.
- Katharina Kann, Arya D. McCarthy, Garrett Nicolai, and Mans Hulden. 2020. [The SIGMORPHON 2020 shared task on unsupervised morphological paradigm completion](#). In *Proceedings of the 17th SIGMORPHON Workshop on Computational Research in Phonetics, Phonology, and Morphology*, pages 51–62, Online. 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. 2020a. [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, et al. 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.
- Arya D. McCarthy, Rachel Wicks, Dylan Lewis, Aaron Mueller, Winston Wu, Oliver Adams, Garrett Nicolai, Matt Post, and David Yarowsky. 2020b. [The Johns Hopkins University Bible corpus: 1600+ tongues for typological exploration](#). In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 2884–2892, Marseille, France. European Language Resources Association.
- Radu Soricut and Franz Och. 2015. [Unsupervised morphology induction using word embeddings](#). In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1627–1637, Denver, Colorado. Association for Computational Linguistics.
- Ekaterina Vylomova, Jennifer White, Elizabeth Salesky, Sabrina J Mielke, Shijie Wu, Edoardo Ponti, Rowan Hall Maudslay, Ran Zmigrod, Josef Valvoda, Svetlana Toldova, et al. 2020. SIGMORPHON 2020 shared task 0: Typologically diverse morphological inflection. *SIGMORPHON 2020*.
- Adam Wiemerslage, Arya McCarthy, Alexander Erdmann, Garrett Nicolai, Manex Agirrezabal, Miikka Silfverberg, Mans Hulden, and Katharina Kann. 2021. The SIGMORPHON 2021 shared task on unsupervised morphological paradigm clustering. In *Proceedings of the 18th SIGMORPHON Workshop on Computational Research in Phonetics, Phonology, and Morphology*. Association for Computational Linguistics.
- David Yarowsky and Richard Wicentowski. 2000. [Minimally supervised morphological analysis by multimodal alignment](#). In *Proceedings of the 38th Annual Meeting of the Association for Computational Linguistics*, pages 207–216, Hong Kong. Association for Computational Linguistics.