Were We Already There? Applying Minimal Generalization to the SIGMORPHON-UniMorph Shared Task on Cognitively Plausible Morphological Inflection

Colin Wilson[1]
[1]Johns Hopkins University
colin.wilson@jhu.edu

Jane S.Y. Li[1,2]
[2]Simon Fraser University
sli213@jhu.edu

Abstract

xxx TBA

1 Introduction

In a landmark paper, Albright and Hayes (2003) proposed a model that learns morphological rules by recursive **minimal generalization** from lexeme-specific examples (*e.g.*, ɪ → ʌ / st _ ŋ for *sting* ∼ *stung* and ɪ → ʌ / fl _ ŋ for *fling* ∼ *flung* generalized to ɪ → ʌ / X [−syllabic, +coronal, +anterior, . . .] _ ŋ).[1] The model was presented more formally in Albright and Hayes (2002), along with evidence that the rules it learns for the English past tense give a good account of native speakers' productions and ratings in wug-test experiments (*e.g.* judgments that *splung* is quite acceptable as the past tense of the novel verb *spling*). In addition to providing further analysis of the experimental data, Albright and Hayes (2003) compared their proposal with early connectionist models of morphology (*e.g.,* Plunkett and Juola, 1999) and an analogical or 'family resemblance' model inspired by research on psychological categories (Nakisa et al., 2001).

Along with Albright (2002), which presents a parallel treatment of Italian inflection, Albright & Hayes's study of the English past tense is a paradigm example of theory-driven, multiple-methodology, open and reproducible research in cognitive science.[2] Their model has enduring significance for the study of morphological learning and productivity in English (*e.g.,* Rácz et al., 2014, 2020; Corkery et al., 2019) and many other languages (*e.g.,* Hijazi Arabic: Ahyad 2019; Japanese: Oseki et al. 2019; Korean: Albright and Kang 2009; Navajo: Albright and Hayes 2006; Portuguese: Veríssimo and Clahsen 2014; Russian: Kapatsinski 2010; Tgdaya Seediq: Kuo 2020; Spanish: Albright and Hayes 2003; Swedish: Strik 2014).

In this study, we apply a partial reimplementation of the Albright and Hayes (2002, 2003) model to wug-test rating data from three languages (German, English, and Dutch) collected for the SIGMORPHON-UniMorph 2021 Shared Task. Our version of the model is based purely on minimal generalization of morphological rules, as described in §3.1 of Albright and Hayes (2002) and reviewed below. It does not include additional mechanisms for learning phonological rules, and expanding or reigning in morphological rules, that were part of the original model (see Albright and Hayes, 2002, §3.3 - §3.7). We think it is worthwhile to consider minimal generalization on its own, with the other mechanisms ablated, as borne out by our competitive results on the shared task.

1.1 Outline

xxx TBA

2 Minimal Generalization

2.1 Inputs

The model takes as input a set of wordform pairs, one per lexeme, that instantiate the same morphological relationship in a language. In simulations of English past tense formation, these are pairs of bare verb stems and past tense forms such as ⟨⋊wɔk⋉, ⋊wɔkt⋉⟩, ⟨⋊tɔk⋉, ⋊tɔkt⋉⟩, ⟨⋊stɪŋ⋉, ⋊stʌŋ⋉⟩, ⟨⋊flɪŋ⋉, ⋊flʌŋ⋉⟩, and ⟨⋊kʌt⋉, ⋊kʌt⋉⟩. Word-forms consist of phonological segments (here, in broad IPA transcription) delimited by special beginning and end of string symbols. The set Σ of

[1]The square brackets contain the shared phonological feature specifications of /t/ and /l/, which in the feature system used here are xxx.

[2]Albright & Hayes released both the results of their wug-test experiments and an implementation of their model (visit http://www.mit.edu/~albright/mgl/ and https://linguistics.ucla.edu/people/hayes/RulesVsAnalogy/index.html). An impediment to large-scale simulation with the model is that it runs from a GUI interface only. As part of the present project, we have added a command line interface to the original source code (available on request).

phonological segments for the language, and the set $\Sigma_{\#} = \Sigma \cup \{\rtimes, \ltimes\}$, are provided to the model.

The model also requires a phonological feature specification for each of the symbols that appears in wordforms. We used a well-known feature chart, augmenting it with orthogonal and distinct feature specifications for the delimiters $\rtimes$ and $\ltimes$.[3] Φ is the set of all (partial) specifications of the features and $\phi(x)$ gives the specifications of $x \in \Sigma_{\#}$.

2.2 Base rules

For each wordform pair, the model constructs a lexeme-specific morphological rule by first identifying the longest common prefix (lcp) of the wordforms excluding $\ltimes$ (*i.e.*, the left-hand context C), then the longest common suffix from the remainder (the right-hand context D), and finally identifying the remaining symbols in the first (A) and second (B) wordform. The resulting rule is $A \rightarrow B / C\_D$. The symbol $\varnothing \notin \Sigma_{\#}$ denotes the empty string in A or B.[4] To illustrate, the rule formed from $\langle \rtimes$wɔk$\ltimes$, $\rtimes$wɔkt$\ltimes \rangle$ has the components C = $\rtimes$wɔk, D = $\ltimes$, A = $\varnothing$ and B = t (*i.e.*, $\varnothing \rightarrow$ t / $\rtimes$wɔk _ $\ltimes$). The rule for $\langle \rtimes$kʌt$\ltimes$, $\rtimes$kʌt$\ltimes \rangle$ is $\varnothing \rightarrow \varnothing$ $\rtimes$kʌt _ $\ltimes$.

2.3 Minimal Generalization

Given any two base rules R_1 and R_2 that make the same change (A $\rightarrow$ B), the model forms a possibly more general rule by aligning and comparing their contexts. The minimal generalization operation, $R = R_1 \sqcap R_2$, carries over the common change of the two base rules and applies independently to their left-hand (C_1, C_2) and right-hand (D_1, D_2) contexts. For convenience, we define minimal generalization of the right-hand contexts. Minimal generalization of the left-hand contexts can be performed by reversing C_1 and C_2, applying the definition for right-hand contexts, and reversing the result.

The minimal generalization $D = D_1 \sqcap D_2$ is defined precedurally by first extracting the lcp $\sigma_{1\wedge 2}$ of the two contexts and then operating on the remainders (D'_1, D'_2). If both D'_1 and D'_2 are empty then $D = \sigma_{1\wedge 2}$. If one but not both of them are empty then $D = \sigma_{1\wedge 2}X$, where $X \notin \Sigma_{\#}$ is a variable over symbol sequences (*i.e.*, X stands for $\Sigma^*_{\#}$). If neither is empty, then the operation determines whether their initial symbols have any shared features; for this purpose it is useful to consider $\phi(x)$ as a function from symbols to sets of feature-value pairs, in which case the common features are found by set intersection.

If there are no common features, $\phi_{1\cap 2} = \emptyset$, then as before $D = \sigma_{1\wedge 2}X$. Otherwise, the set of common features $\phi_{1\cap 2} \neq \emptyset$ is appended to $\sigma_{1\wedge 2}$, the first symbol is removed from D'_1 and D'_2, and the operation applies to the remainders. If both remainders are empty then $D = \sigma_{1\wedge 2}\phi_{1\cap 2}$, otherwise $D = \sigma_{1\wedge 2}\phi_{1\cap 2}X$.

In summary, the generalized right-hand context D consists of the longest common prefix shared by D_1 and D_2, followed by a single set of shared features (if any), followed by X in case there are no shared features or one context is longer than the other. With the change and generalized left-hand context C determined as already described, the result of applying minimal generalization to the two base rules is $R = A \rightarrow B / C \_ D$.[5]

2.4 Recursive Minimal Generalization

Let $\mathcal{R}_1$ be the set of base rules (one per wordform pair in the input data) and $\mathcal{R}_2$ be the set containing all of the base rules and the result of applying minimal generalization to each eligible pair of base rules. While the rules of $\mathcal{R}_2$ have greater collective scope, they are nevertheless unlikely to account for the level of morphological productivity shown by native speakers. For example, English speakers can systematically rate and produce past tense forms of novel verbs that contain unusual segment sequences, such as *ploamf* /ploʊmf/ (*e.g.*, Prasada and Pinker, 1993). Albright & Hayes propose to apply minimal generalization recursively and demonstrate that this can yield rules of great generality (*e.g.*, in our notation, $\varnothing \rightarrow$ t / X [-voice] _ $\ltimes$).

In the original proposal, recursive minimal generalization was defined only for pairs that include

[3] The phonological feature chart is available from Bruce Hayes's website, https://linguistics.ucla.edu/people/hayes/120a/Index.htm#features. xxx binary with 0s xxx original features distribued with the model included scalar features xxx for example. Alternative binary (with underspecification) feature sets are xxx phoible (Moran et al., 2014) xxx panphon (Mortensen et al., 2016).

[4] In other common notations, the empty string is denoted by λ. xxx notation for phonological segment strings generally follows (**?**) and works cited there.

[5] There could be a slight difference between our definition of context generalization and that in Albright and Hayes (2002), hinging on whether the empty feature set is allowed in rules. In our definition, $\phi_{1\cap 2} = \emptyset$ is replaced by the variable X. It is possible that the original proposal intended for empty and non-empty feature sets to be treated alike. The definitions can diverge when applied to contexts that are of identical length and share all but the last (resp. first) segments, in which case our version would result in a broader rule.

one base rule; it was conjectured that no additional generalizations could result from dropping this restriction. Here we define the operation for any two right-hand contexts $D_1, D_2 \in \Sigma^*_\#(\Phi)(X)$. As before, only rules that make the same change are eligible for generalization and the operation applies to left-hand contexts under reversal.

The revised definition of $D = D_1 \sqcap D_2$ is identical to that given above except that we must consider input contexts that contain feature sets and X (which previously could occur only in outputs). As before, we first identify the lcp of symbols from $\Sigma_\#$ in the two contexts, $\sigma_{1\wedge 2}$, and then operate on the remainders (D'_1, D'_2). If both D'_1 and D'_2 are empty then $D = \sigma_{1\wedge 2}$. If one but not both of them are empty then $D = \sigma_{1\wedge 2}X$. If both are non-empty then their initial elements are either symbols in $\Sigma_\#$, feature sets in Φ, or X. Replace any initial symbol $x \in \Sigma_\#$ with its feature set $\phi(x)$, extend the function ϕ so that $\phi(X) = \emptyset$, and compute the unification $\phi_{1\cap 2}$ of the initial elements. The rest of the definition is unchanged (see the end of §2.3).

By construction, the contexts that result from this operation are also in $\Sigma^*_\#(\Phi)(X)$ (*i.e.*, no ordinary symbol can occur after a feature set, there is at most one feature set, X can occur only at the end of the context, etc.). Therefore, the revised definition supports the application of minimal generalization to its own products. Let $\mathcal{R}_k$ be the set of rules containing every member of $\mathcal{R}_{k-1}$ and the result of applying minimal generalization to each eligible pair of rules in $\mathcal{R}_{k-1}$ (for $k > 1$). In principle, there is an infinite sequence of rules set related by inclusion $\mathcal{R}_1 \subseteq \mathcal{R}_2 \subseteq \mathcal{R}_3 \cdots$. In practice, the equality becomes strict after a small number of iterations of minimal generalization (typically 6-7), at which point there are no more rules to be found.

2.5 Completeness

Having defined minimal generalization for arbitrary contexts (as allowed by the model), we can revisit the conjecture that nothing is lost by restricting the operation to pairs at least one of which is a base rule. This is a practical concern, as the number of base rules is a constant determined by the input data while the number of generalized rules can increase exponentially.

Conceptually, each rule learned by unrestricted minimal generalization has a (possibly non-unique) 'history' of base rules from which it originated. A base rule $R \in \mathcal{R}_1$ has the history $\{R\}$. A rule in $R \in \mathcal{R}_2$ has the history $\{R_1, R_2\}$ containing the two base rules from which it derived. In general, the history of each rule in $\mathcal{R}_k$ is the union of the histories of two rules in $\mathcal{R}_{k-1}$ $(k > 1)$.

Because all rules are learned 'bottom-up' in this sense, the conjecture can be proved by showing that the minimal generalization operation is associative; we also show that it is commutative — both properties inherited from equality, lcp, set intersection, and other more primitive ingredients. As before, we explicitly consider right-hand contexts, from which parallel results for left-hand contexts and entire rules follow immediately. It follows that any rule R can be replaced, for the purposes of minimal generalization, with the base rules in its history (in any order).

Commutative. Let $D = D_1 \sqcap D_2$ for any $D_1, D_2 \in \Sigma^*_\#(\Phi)(X)$. We prove by construction that D is also equal to $D_2 \sqcap D_1$. The lcp of elements from $\Sigma_\#$ is the same regardless of the order of the contexts ($\sigma_{1\wedge 2} = \sigma_{2\wedge 1}$) as are the remainders (D'_1 and D'_2). If both remainders are empty, then the result of minimal generalization is $\sigma_{1\wedge 2} = \sigma_{2\wedge 1}$. If one but not both of them are empty then the result is $\sigma_{1\wedge 2}X = \sigma_{2\wedge 1}X$; note that X appears regardless of which context is longer. If both are non-empty then we ensure that their initial elements are (possibly empty) feature sets and take their intersection, which is order independent: $\phi_{1\cap 2} = \phi_{2\cap 1}$. If $\phi_{1\cap 2} = \emptyset$ then the result is $\sigma_{1\wedge 2}X = \sigma_{2\wedge 1}X$. Otherwise, the initial elements are removed and the operation continues to the remainders. If both remainders are empty the result is $\sigma_{1\wedge 2}\phi_{1\cap 2} = \sigma_{2\wedge 1}\phi_{2\cap 1}$, otherwise it is the same expressions terminated by X.

Associative. Let $D = (D_1 \sqcap D_2) \sqcap D_3$ for any $D_1, D_2, D_3 \in \Sigma^*_\#(\Phi)(X)$. We prove by construction that D is equal to $E = D_1 \sqcap (D_2 \sqcap D_3)$. Let σ be the longest prefix of symbols from $\Sigma_\#$ in D. Because σ occurs in D iff it is the lcp of this type in $(D_1 \sqcap D_2)$ and D_3, it must be a prefix of each of D_1, D_2, D_3 and the longest such prefix in at least one of them. It follows that σ is also the lcp of symbols from $\Sigma_\#$ in D_1 and $(D_2 \sqcap D_3)$. Therefore, D and E both begin with σ. We now remove the prefix σ from all of the input contexts and consider the remainders D'_1, D'_2, D'_3.

If all of the remainders are empty, then $D = E = \sigma$. If all but one of them are empty, then $D = E = \sigma X$.[6] If none of the remainder is empty,

[6] If D'_1 or D'_2 is the longest context, assume by com-

let ϕ_1, ϕ_2, ϕ_3 be their (featurized) initial elements. The intersection of those elements is independent of grouping, $\phi = (\phi_1 \cap \phi_2) \cap \phi_3 = \phi_1 \cap (\phi_2 \cap \phi_3)$. If the intersection is empty then again $D = E = \sigma X$. If the intersection is non-empty then D and E both begin $\sigma\phi$. Finally, remove the initial elements of each of D'_1, D'_2, D'_3 and compare the lengths of the remainders to determine whether X appears at the end of D and E; this is independent of grouping along the same lines shown previously.

Complete. We now prove by induction that, for any $R \in \mathcal{R}_k$ and $R_1, R_2 \in \mathcal{R}_{k-1}$ $(k > 1)$ such that $R = R_1 \sqcap R_2$, rule R can also be derived by applying minimal generalization to R_1 and one or more base rules (*i.e.*, the rules in the history of R_2).[7] For $R \in \mathcal{R}_2$ this is true by definition. For $R \in \mathcal{R}_3$, we have $R = R_1 \sqcap R_2 = R_1 \sqcap (R_{21} \sqcap R_{22}) = (R_1 \sqcap R_{21}) \sqcap R_{22}$, where R_{21} and R_{22} are base rules whose minimal generalization results is R_2. In general, suppose that the statement is true for $k - 1 > 0$. Then it is also true for k because $R \in \mathcal{R}_k$ can be derived by $R_1 \sqcap R_2 = R_1 \sqcap (\sqcap_{i=1}^{n} R_{2i}) = (((R_1 \sqcap R_{21}) \sqcap R_{22}) \cdots \sqcap R_{2n})$ where $R_1, R_2 \in \mathcal{R}_{k-1}$ and each R_{2i} is a base rule in the history of R_2.

These results validate the rule learning algorithm proposed by Albright and Hayes (2002) and used in our implementation. Any minimal generalization of two arbitrary rules R_1 and R_2 (as allowed by the model) can also be derived from R_1 (or R_2) by recursive application of minimal generalization with one or more base rules.

2.6 Relative generality

While not required for the minimal generalization operation itself, we define here a (partial) generality relation on rules. The definition uses the same notation as above and is employed in pruning rules after recursive minimal generalization has been applied (see §3.4 below).

Relative generality is defined only for rules R_1 and R_2 that make the same change. It is sufficient to consider the right-hand contexts D_1 and D_2 and then apply the same definition to the reversed left-hand contexts. Conceptually, context D_2 is at least as general as context D_1, $D_1 \sqsubseteq D_2$, iff the set of strings represented by D_1 is a subset of that represented by D_2 when both contexts are considered as regular expressions over $\Sigma^*_{\#}$. The formal definition is complicated somewhat by X, which can appear at the end of either context.

Replace each symbol $x \in \Sigma_{\#}$ in D_1 or D_2 with its feature set $\phi(x)$, treat X as equivalent to $\emptyset$, and let $|D|$ be the length of context D.Then $D_1 \sqsubseteq D_2$ iff (i) $|D_1| \geq |D_2|$ and $D_1[k] \subseteq D_2[k]$ for all $1 \leq k \leq |D_1|$, except when $|D_1| = |D_2| + 1$ and the last element of D_1 but not D_2 is X, or (ii) $|D_1| = |D_2| - 1$, $D_1[k] \subseteq D_2[k]$ for all $1 \leq k \leq |D_1|$, and the last element of D_2 is X. Context D_2 is strictly more general than D_1, $D_1 \sqsubset D_2$, iff $D_1 \sqsubseteq D_2$ and $D_2 \not\sqsubseteq D_1$. Rule R_2 is at least as general as R_1, $R_1 \sqsubseteq R_2$, iff $C_1 \sqsubseteq C_2$ and $D_1 \sqsubseteq D_2$; it is a strictly more general rule iff either of the context relations is strict.

3 System Description and Results

Our system for the shared task preprocesed the input wordforms, learned rules with recursive minimal generalization, scored the rules in two ways, pruned rule that have no effect on the model's predictions, and applied the remaining rules to wug forms in order to generated predicted ratings.

3.1 Preprocessing

The shared task provided space-separated broad IPA transcriptions of the training and wug wordforms (*e.g.*, s t ɪ ŋ, s t ʌ ŋ, w ɔ k, w ɔ k t). As already mentioned, we added explicit beginning and end of string symbols. Because minimal generalization requires each wordform symbol to have a phonological feature specificiation, but some segments in the data lack entries in our feature system, we further simplified or split the symbols as follows.

For German, we split the diphthongs /ai̯ au̯ oi̯ iːə eːə ɛːə/ into their component vowels and additionally regularized /i̯ u̯/ to /i u/. For English, we split the diphthongs /aɪ aʊ ɔɪ uːɪ/ into their components and /ɝ/ into /ɛ ɹ/, simplified /eɪ əʊ/ to /e o/, and regularized /m̩ n̩ r l̩ ɔ̃/ to /m n ɹ l ɔ/. We also deleted all length marks /ː/ and instances of /ˠ/. For Dutch, we split /ɛɪ ɑʊ ʊɪ/ into their components.

Checking that all wordform symbols appear in a phonological feature chart is also useful for data cleaning. It helped us to identify a few thousand Dutch wordforms containing '+' (indicating a Verb - Preposition combination), which we removed. And it caught an encoding error in which two

mutativity that it is D'_1. The minimal generalizations are $(D'_1 \sqcap D'_2) = X$ and $X \sqcap D'_3 = X$, which gives the same result as $(D'_2 \sqcap D'_3) = \lambda$ and $D'_1 \sqcap \lambda = X$. Similar reasoning applies if D'_3 is the longest context.

[7] We ignore rules that are carried over from $\mathcal{R}_{k-1}$ to $\mathcal{R}_k$.

distinct but perceptually similar Unicode symbols were used for /g/.

Two acknowledged limitations of the original version of the minimal generalization model, and our version, are relevant here. First, the model learns rules for individual morphological relations (*e.g.*, mapping a bare stem to a past tense form), not for entire morphological systems jointly. Therefore, we retained from the preprocessed input data only the wordform pairs that instantiate the relations targeted by the wug tests: formation of past participles in German (Clahsen, 1999) and past tenses in English and Dutch (Booij, 2019).

Second, the model cannot learn sensible rules for circumfixes (xxx). This could be remedied by allowing the model to form rules that simultaneously make changes at both edges of inputs, or by allowing it to apply multiple single-edge rules when mapping inputs to outputs. As a provisional solution, we removed the invariant prefix /gə-/ whenever it occured at the beginning of a German past participle (training or wug wordform).[8]

3.2 Rules

Given the preprocessed and filter input data, a base rule was learned for each lexeme and then minimal generalization was applied recursively as in §2. This results in tens of thousands of morphological rules for each of the three languages (xxx table reference).

A major goal of Albright & Hayes was to learn rules that can construct outputs from inputs (as opposed to merely rating or selecting outputs that are generated by some other source). Their model achieved this goal, and a substantial portion of its original implementation was dedicated to rule application. We instead delegated the application of rules to a general purpose finite-state library (Pynini; Gorman, 2016; Gorman and Sproat, 2021).

Each component of a rule $A \rightarrow B / C\_D$ was first converted to a regular expression over symbols in $\Sigma_{\#}$ by mapping any feature set $\phi \in \Phi$ to the |-disjunction of symbols that bear all of the specified features and deleting instances of X. Segments were then encoded as integers using a symbol table. Pynini provides a function `cdrewrite` that compiles rules in this format to finite-state transducers, a function `accep` for converting input strings to linear finite-state acceptors encoded with the same symbol table, a composition function `@` that applies rules to inputs yielding output acceptors, and the means to decode the result back to string form.[9]

3.3 Scoring

The *score* of a rule is a function of its accuracy on the training data. The simplest notion of score would be accuracy: the number of training outputs that are correctly predicted by the rule (*hits*), divided by the number of training inputs that meet the structural description of the rule (*scope*). Albright & Hayes propose instead to discount the scores of rules with smaller scopes, using a formula previously applied to linguistic rules by Mikheev (1997) (see xxx; one free parameter α set to 0.55 as in A&H 2003, p.127). Our implementation includes this way of scoring rules, which Albright & Hayes call *confidence*.

Because confidence imposes only a modest penalty on rules with small scopes, we also considered a score function of the form $score_{\beta} = hits/(scope + \beta)$, where β is a non-negative discount factor (here, $\beta = 10$). A rules that is perfectly accurate and applies to just 5 cases has high confidence $= .90$ but much lower $\text{score}_{10} = .33$; one that applies perfectly to 1000 cases has a near-maximal score ($> .99$) regardless of which function is used. Clearly, these are only two of a wide range of score functions that could be explored.

3.4 Pruning

When applied to training data consisting of thousands of lexemes, recursive minimal generalization can produce tens of thousands of distinct rules. Albright & Hayes mention but do not implement the possibility of pruning the rules on the basis of their generality and scores. We pursued this suggestion by first partitioning the set of all learned rules according to their change and imposing a partial order on each of the resulting subsets.

We ordered rules by generality (§2.6), score, and length when expressed with features (Chomsky and Halle, 1968). Rule R_2 dominated rule R_1 in the order, $R_1 \prec R_2$ iff R_2 was at least as general as R_1 ($R_1 \sqsubseteq R_2$) and (i) R_2 had a higher score or (ii)

[8] xxx prefix constant in wug outputs, occurred both initially and finally in training outputs; removed only if absolutely initial

[9] xxx (Mohri and Sproat, 1996) xxx (Riley et al., 2009) The technique of mapping feature matrices to disjunctions (*i.e.*, natural classes) of segments and beginning/end symbols, and ultimately to disjunctions of integer ids, was also used in the finite-state implementation of Hayes and Wilson (2008). X was deleted because it occurs only at the beginning of left-hand contexts and at the end of right-hand contexts, both positions where Pynini's rule compiler implicitly adds $\Sigma^*_{\#}$.

the rules tied on score and R_2 was either strictly more general ($R_1 \sqsubset R_2$) or shorter. Dominated rules were pruned without affecting the predictions of the model, as we discuss next.

3.5 Prediction

Once rules have been learned by minimal generalization and scored, they can be used for multiple purposes: to generate potential outputs for input wordforms (by finite-state composition), to determine possible inputs for a given output wordform (by composition with the inverted transducer), and to assign scores to input/output mappings. Following Albright & Hayes, we assume that the score of a mapping is taken from the highest-scoring rule(s) that could produce it. Rules neither 'gang up' — multiple rules cannot contribute to the score of a mapping — nor do they compete — rules that prefer different outputs for the same input do not detract from the score. When no rule produces a mapping, we assign it the minimal score of zero.

As for the scoring function itself, many other possibilities could be considered. For example, rule scores could be normalized within or across changes, a type of competition that is inherent to probabilistic models. See Albright and Hayes (2006) for a different kind of competition model in which rules learned by minimal generalization are weighted like conflicting constraints.

3.6 Results

xxx TBA

4 Conclusions and Future Directions

xxx TBA

(Tenenbaum, 1999)

(Plotkin, 1970)

4.1 Near misses

As the organizers of this shared task have emphasized, implemented models can be used not only to predict the results of experiments but also to generate stimuli. Ideally, stimulus items would be designed to test the core tenets of a single model or to probe systematic differences in prediction among models. As part of our implementation, we have developed a method for generating wug items that targets what is likely to be the principal failing of minimal generalization: namely, that by learning rules in a purely bottom-up way it undergeneralizes, predicting sharp contrasts in inflectional behavior on the basis of minor differences in wordforms.

We illustrate our method with the English irregular pattern ɪ → ʌ, which has attracted new members in the history of English and elicited relatively high production rates and acceptability ratings in previous wug tests (*e.g.*, Bybee and Moder, 1983; Albright and Hayes, 2003). We first extracted all of the onsets and rimes that appear in the bare forms of monosyllabic English verbs, and freely combined these to create a large pool of possible stimulus items. We eliminated items that are real verbs, then shrunk the pool to those items that are one (segmental) edit away from some existing irregular verb that undergoes the change ɪ → ʌ. We also required the remaining items in the pool to have the same rime as at least one such irregular verb.[10] All of the nonce items in the pool are highly similar, in this sense, to existing irregulars.

We then divided the pool into two sets: items that are within the scope of at least one ɪ → ʌ rule learned by minimal generalization (*close neighbors*), and items that are beyond the scope of all such rules (*near misses*). For the former, we recorded the highest-scoring applicable rule. We wanted to provide the model the opportunity to form rules that were as broad as possible — making it more difficult for us to find near misses — and therefore implemented cross-context base rules as described earlier.[11]

Some of the close neighbors and near misses are minimal pairs. For example, /lɪŋ/ (.67) and /ʃɪŋ/ (.61) could potentially undergo ɪ → ʌ rules with the indicated confidence values. But /fɪŋ/ and /vɪŋ/ are ineligible for the change, according to the model, because no existing irregular verb of this type has a labial fricative immediately before the vowel. Other differences in onsets can also dramatically affect the model's predictions: /θɹɪŋk/ (.88) and /glɪŋ/ (.67) are close neighbors but /smɪŋk/ and /smɪŋ/ are near misses. The latter are phonotactically challenged (Davis, 1989), but is /θɹʌŋk/ a far superior past tense forms than /smʌŋk/?

Our method can be applied to any irregular (or indeed regular) change. For i → ɛpt (as in *sleep* ∼ *slept*), the close neighbors include /gip/ (.85) and /flip/ (.73, one of Albright & Hayes's wug

[10] Studies of English irregular verbs have focused primarily on vowels and codas of monosyllables, though see Bybee and Moder (1983) on the potential role of onsets.

[11] With this change in implementation, which we did not apply elsewhere in the paper, the total number of rules for the English past tense ballooned to 191,874. Even after pruning there were tens of thousands of rules (69,747) and 128 for just ɪ → ʌ. The majority of the rules have very low scores.

Language	Lexemes	Rules (all)	Rules (pruned)	AIC (dev wugs)	AIC (test wugs)
German	3,417	31,562	3,629	-127.6	-135.0
English	5,803	30,728	263	-112.0	-62.2
Dutch	7,823	55,114	1,862	-58.5	-76.5

Table 1: xxx

stems) while among the near misses are /fip/, /vip/, /nip/, and /snip/. Would the nonce past tense /gɛpt/ be far more acceptible than /fɛpt/? We look forward to further empirical tests of the minimal generalization model, along these lines and others, to determine where we are and how much further we have to go in cognitive modeling of inflection.

Acknowledgements

We would like to thank the organizers for all of their work on the shared task. Special thanks to Adam Albright and Bruce Hayes for inspiring this study and for stimulating conversations over many years. The research presented here was partially supported by NSF grant BCS-1844780 to CW.

References

Honaida Yousuf Ahyad. 2019. *Vowel Distribution in the Hijazi Arabic Root*. Ph.D. dissertation, State University of New York at Stony Brook.

Adam Albright. 2002. Islands of reliability for regular morphology: Evidence from Italian. *Language*, 78(4):684–709.

Adam Albright and Bruce Hayes. 2002. Modeling English past tense intuitions with minimal generalization. In *Proceedings of the ACL-02 Workshop on Morphological and Phonological Learning*, pages 58–69. Association for Computational Linguistics.

Adam Albright and Bruce Hayes. 2003. Rules vs. analogy in English past tenses: A computational/experimental study. *Cognition*, 90(2):119–161.

Adam Albright and Bruce Hayes. 2006. Modeling productivity with the Gradual Learning Algorithm: The problem of accidentally exceptionless generalizations. In Gisbert Fanselow, Caroline Fery, Matthias Schlesewsky, and Ralf Vogel, editors, *Gradience in Grammar: Generative Perspectives*, pages 185–204. Oxford University Press, Oxford.

Adam Albright and Yoonjung Kang. 2009. Predicting innovative alternations in Korean verb paradigms. *Current issues in unity and diversity of languages: Collection of the papers selected from the 18th International Conference of Linguistics*, pages 1–20.

Geert Booij. 2019. *The Morphology of Dutch*, second edition. Oxford University Press, New York.

Joan L. Bybee and Carol Lynn Moder. 1983. Morphological classes as natural categories. *Language*, 59(2):251–270.

Noam Chomsky and Morris Halle. 1968. *The Sound Pattern of English*. MIT Press, Cambridge, MA.

Harald Clahsen. 1999. Lexical entries and rules of language: A multidisciplinary study of German inflection. *Behavioral and Brain Sciences*, 22(6):991–1013.

Maria Corkery, Yevgen Matusevych, and Sharon Goldwater. 2019. Are we there yet? encoder-decoder neural networks as cognitive models of English past tense inflection. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3868–3877, Florence, Italy. Association for Computational Linguistics.

Stuart Davis. 1989. Cross-vowel phonotactic constraints. *Computational Linguistics*, 15(2):109–110.

Kyle Gorman. 2016. Pynini: A Python library for weighted finite-state grammar compilation. In *Proceedings of the SIGFSM Workshop on Statistical NLP and Weighted Automata*, pages 75–80, Berlin, Germany. Association for Computational Linguistics.

Kyle Gorman and Richard Sproat. 2021. Finite-State Text Processing. *Synthesis Lectures on Human Language Technologies*, 14(2):1–158.

Bruce Hayes and Colin Wilson. 2008. A maximum entropy model of phonotactics and phonotactic learning. *Linguistic Inquiry*, 39(3):379–440.

Vsevolod Kapatsinski. 2010. Velar palatalization in Russian and artificial grammar: Constraints on models of morphophonology. *Laboratory Phonology*, 1(2):361–393.

Jennifer Kuo. 2020. *Evidence for Base-Driven Alternation in Tgdaya Seediq*. Master's Thesis, UCLA.

Andrei Mikheev. 1997. Automatic rule induction for unknown-word guessing. *Computational Linguistics*, 23(3):405–423.

Mehryar Mohri and Richard Sproat. 1996. An efficient compiler for weighted rewrite rules. In *34th Annual Meeting of the Association for Computational Linguistics*, pages 231–238, Santa Cruz, California, USA. Association for Computational Linguistics.

Steven Moran, Daniel McCloy, and Richard Wright, editors. 2014. *PHOIBLE Online*. Max Planck Institute for Evolutionary Anthropology, Leipzig.

David R. Mortensen, Patrick Littell, Akash Bharadwaj, Kartik Goyal, Chris Dyer, and Lori Levin. 2016. PanPhon: A resource for mapping IPA segments to articulatory feature vectors. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 3475–3484, Osaka, Japan. The COLING 2016 Organizing Committee.

Ramin Charles Nakisa, Kim Plunkett, and Ulrika Hahn. 2001. A cross-linguistic comparison of single and dual-route models of inflectional morphology. In Peter Broeder and Jaap Murre, editors, *Models of Language Acquisition: Inductive and Deductive Approaches*, pages 201–222. MIT Press, Cambridge, MA.

Yohei Oseki, Yasutada Sudo, Hiromu Sakai, and Alec Marantz. 2019. Inverting and modeling morphological inflection. In *Proceedings of the 16th Workshop on Computational Research in Phonetics, Phonology, and Morphology*, pages 170–177, Florence, Italy. Association for Computational Linguistics.

Gordon D. Plotkin. 1970. A note on inductive generalization. In *Machine Intelligence*, volume 5, pages 153–163. Edinburgh University Press.

Kim Plunkett and Patrick Juola. 1999. A connectionist model of English past tense and plural morphology. *Cognitive Science*, 23(4):463–490.

Sandeep Prasada and Steven Pinker. 1993. Generalisation of regular and irregular morphological patterns. *Language and Cognitive Processes*, 8(1):1–56.

Péter Rácz, Clay Beckner, Jennifer B. Hay, and Janet B. Pierrehumbert. 2020. Morphological convergence as on-line lexical analogy. *Language*, 96(4):735–770.

Péter Rácz, Clayton Beckner, Jennifer B. Hay, and Janet B. Pierrehumbert. 2014. Rules, analogy, and social factors codetermine past-tense formation patterns in English. In *Proceedings of the 2014 Joint Meeting of SIGMORPHON and SIGFSM*, pages 55–63, Baltimore, Maryland. Association for Computational Linguistics.

Michael Riley, Cyril Allauzen, and Martin Jansche. 2009. OpenFst: An open-source, weighted finite-state transducer library and its applications to speech and language. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics, Companion Volume: Tutorial Abstracts*, pages 9–10, Boulder, Colorado. Association for Computational Linguistics.

Oscar Strik. 2014. Explaining tense marking changes in Swedish verbs: An application of two analogical computer models. *Journal of Historical Linguistics*, 4(2):192–231.

Joshua Tenenbaum. 1999. Bayesian modeling of human concept learning. In *Advances in Neural Information Processing Systems*, volume 11, Cambridge, MA. MIT Press.

João Veríssimo and Harald Clahsen. 2014. Variables and similarity in linguistic generalization: Evidence from inflectional classes in Portuguese. *Journal of Memory and Language*, 76:61–79.

A Example Appendix

This is an appendix.