Were We There Already? 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

Morphological rules with various levels of specificity can be learned from example lexemes by recursive application of minimal generalization (**??**). A model that learns rules solely through minimal generalization was used to predict average human wug-test ratings from German, English, and Dutch in the SIGMORPHON-UniMorph 2021 Shared Task, with competitive results. Some formal properties of the minimal generalization operation were proved, justifying the model's learning algorithm and allowing the resulting rule sets to be substantially pruned. An automatic method was developed to create wug-test stimuli for future experiments that investigate whether the model's morphological generalizations are too minimal.

1 Introduction

In a landmark paper, **?** 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 **?**, 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 behavioral data, **?** compared their proposal with early connectionist models of morphology (e.g., **?**) and an analogical or 'family resemblance' model inspired by research on psychological categories (**?**).

Along with **?**, which presents a parallel treatment of Italian inflection, Albright & Hayes's study of the English past tense is an ideal 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., **???**) and many other languages (e.g., Hijazi Arabic: **?**; Japanese: **?**; Korean: **?**; Navajo: **?**; Portuguese: **?**; Russian: **?**; Tgdaya Seediq: **?**; Spanish: **?**; Swedish: **?**).

In this study, we applied a partial reimplementation of the **??** 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 **?** and reviewed below. It does not include additional mechanisms for learning phonological rules, and further expanding or reigning in morphological rules, that were part of the original model (see **?**, §3.3 - §3.7). We think it is worthwhile to consider minimal generalization on its own, with other mechanisms ablated, as borne out by our competitive results on the shared task.

1.1 Outline

In §2 we review the definition of minimal generalization proposed by Albright & Hayes and prove

[1]The square brackets contain all of the the shared phonological feature specifications of /t/ and /l/, which in the system used here are [−syllabic, +consonantal, −nasal, −spread.gl, −labial, −round, −labiodental, +coronal, +anterior, −distributed, −strident, −dorsal].

[2]Albright & Hayes released both the results of their wug-test experiments and an implementation of their model (see 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 and converted the English input files to a more user-friendly format (available on request). We are aware of one other implementation of the minimal generalization model, due to João Veríssimo, but this was unavailable at the time of our study (https://www.jverissimo.net/resources).

a number of original results about the operation and its recursive application in learning rules. We also define a generality relation that can be used to prune insufficiently broad rules without affecting the model's predictions. In §3 we describe how we preprocessed the shared task training data and generated predicted wug-test ratings, and report our results on the task. We briefly summarize our findings in §4 and conclude by discussing a novel method for constructing wug items that can be used in future empirical tests of minimal generalization and other approaches to morphological learning.

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⋉⟩ for the lexemes *walk*, *talk*, *sting*, *fling*, and *cut*. Wordforms consist of phonological segments (here, in broad IPA transcription) delimited by special beginning and end of string symbols. The set Σ of phonological segments for the language, and the set $\Sigma_{\#} = \Sigma \cup \{⋊, ⋉\}$, are provided to the model.

The model also requires a phonological feature specification for each of the symbols that appears in wordforms. We adopted a well-known feature system, augmenting it with orthogonal and distinct feature specifications for the delimiters ⋊ and ⋉.[3] The set Φ contains all possible (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 ⋉ (i.e., the left-hand rule 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 ⟨⋊wɔk⋉, ⋊wɔkt⋉⟩ has the components C = ⋊wɔk, D = ⋉, A = ∅ and B = t (i.e., ∅ → t / ⋊wɔk _ ⋉). The rule for ⟨⋊kʌt⋉, ⋊kʌt⋉⟩ is ∅ → ∅ / ⋊kʌt _ ⋉.

2.3 Minimal Generalization

Given any two base rules R_1 and R_2 that make the same change (A → 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 remainder 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, so that 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 processes the remainders. If both remainders are empty then $D = \sigma_{1\wedge 2}\phi_{1\cap 2}$, otherwise $D = \sigma_{1\wedge 2}\phi_{1\cap 2}X$.

To summarize, 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

[3] The phonological features are available from Bruce Hayes's website (https://linguistics.ucla.edu/people/hayes/120a/Index.htm#features). These features are all binary, with the possibility of underspecification, while Albright & Hayes's original simulations made use of some multi-valued scalar features. Alternative sources of binary feature systems that are compatible with our implementation include PHOIBLE (**?**) and PanPhon (**?**).

[4] We instead use $\lambda \notin \Sigma_{\#}$ to stand for the empty string in left- and right- hand contexts. Our notation for strings of phonological segments generally follows **?** and research cited there.

than the other. With the change and generalized left-hand context C determined as noted above, 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 than those of $\mathcal{R}_1$, 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., **?**). Albright & Hayes propose to apply minimal generalization recursively and demonstrate that this can yield rules that are highly general (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 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 definition of $D = D_1 \sqcap D_2$ needed for recursive application is identical to the one 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 union $\phi_{1 \cap 2}$ of the initial elements. The rest of the definition is unchanged (see 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 only be a terminal element, 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 $\{R_1, R_2\}$ consisting of 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

[5] There could be a small difference between our definition of context generalization and that in **?**, 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 right contexts that are of identical length and share all but the last segment (resp. left contexts that share all but the last segment), in which case our version would result in a broader rule.

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 input 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 that appears in all 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 remainders is empty, 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 in 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 **?** and used in our implementation. Any minimal generalization of two rules R_1 and R_2 allowed by the model can 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 applied (see §3.4 below).

Relative generality is defined only for rules R_1 and R_2 that make the same change. As usual, 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_2 is a superset of that represented by D_1 when both contexts are considered as regular expressions over $\Sigma^*_\#$. The procedural 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.

[6] If D'_1 or D'_2 is the longest context, assume by commutativity 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 $\mathcal{R}_k$.

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 alternative ways, pruned rule that have no effect on the model's predictions, and applied the remaining rules to wug forms to yield predicted ratings.

3.1 Preprocessing

The shared task provided space-separated broad IPA transcriptions of the training and wug wordforms (e.g., /w ɔ k/, /w ɔ k t/, /s t ɪ ŋ/, /s t ʌ ŋ/). 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 chart, 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 useful for data cleaning. It helped us to identify a few thousand Dutch wordforms containing '+' (indicating a Verb + Preposition juncture), which we removed. And it caught an encoding error in which two distinct but perceptually similar Unicode symbols were used for the voiced velar stop /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 shared task: formation of past participles in German (**?**) and past tenses in English and Dutch (**?**).

Second, the model cannot learn sensible rules for circumfixes (**?**, §5.2). This could be remedied by allowing the model to form rules that simultaneously make changes at both wordform edges, or by allowing it to apply multiple rules when mapping inputs to outputs. As a workaround, we simply removed the prefix /gə-/ whenever it occured at the beginning of a German past participle (training or wug wordform).

3.2 Rules

Given the preprocessed and filtered input data, a base rule was learned for each lexeme and then minimal generalization was applied recursively as in §2. This resulted in tens of thousands of morphological rules for each of the three languages (see Table 1).

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; **??**), as follows.

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 output back to strings.[8]

3.3 Scoring

The *score* of a rule is related to its accuracy on the training data. The simplest notion of score would be just 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

[8] 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 **?**. X was deleted here 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 $\Sigma^*_{\#}$. Pynini's implementation of finite-state automata wraps and extends OpenFst (**?**) and its rule compilation algorithm is due to **?**.

rules with smaller scopes, using a formula previously applied to linguistic rules by **?**. Our implementation also includes this way of scoring rules, which Albright & Hayes call *confidence*.[9]

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 score_{10} (.33); one that applies perfectly to 1000 cases has a near-maximal value ($> .99$) regardless of how the score is calculated. 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 (**?**). Rule R_2 dominates rule R_1 in the order, $R_1 \prec R_2$ iff R_2 is at least as general as R_1 ($R_1 \sqsubseteq R_2$) and (i) R_2 has a higher score or (ii) the rules tie on score and R_2 is 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 assigned 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 **?** for a different kind of competition model in which rules learned by minimal generalization are weighted as conflicting constraints.

3.6 Results

Table 1 provides quantitative details of our simulations for the three morphological relations in the shared task. The AIC values were calculated with an evaluation script provided by the organizers, which compares average human ratings of output wordforms with ratings predicted by the model. (Values are not directly comparable across the languages because the number of wug forms differed.)

We used whichever scoring method, confidence or score_{10}, achieved a better AIC value on the development wug data. For German and English, this was confidence; for Dutch it was score_{10}. Upon close inspection of the development data for English, we found it plausible that human participants had down-rated regular past tense forms of bare forms ending in coronal stops /t d/ because these might appear to be 'double past' inflections (e.g., /vaɪndəd/ for the stem /vaɪnd/, which has a rime /aɪnd/ that is rare outside of past tense forms). Therefore, in generating predictions for the English wug test we added a penalty to the model score for such outputs. The magnitude of the penalty was fit by linear regression to the development data. As the development and test wugs were generated by different methods, addition of this factor could have had a detrimental effect on the model's performance. On the contrary, our model had the best AIC for the German and English test data and the best overall AIC (summed over the languages).

4 Summary and Future Directions

We have described the minimal generalization operation for morphological rules as proposed by Albright & Hayes and presented some new formal results on this operation. We have also described our partial implementation of their model — a pure minimal generalization learner — and applied it to wug-test data from three related languages. We conclude with some remarks on how our imple-

[9] The confidence formula has one free parameter, which we set to $\alpha = .55$ following **?**, p. 127.

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

Table 1: Number of lexemes (wordform pairs) used for training, number of rules learned by minimal generalization (before and after pruning), and evaluation on average human wug-test ratings for each language. Lower AIC values indicate a better match between model predictions and behavioral results.

mentation could be extended and how the central concept of minimal generalization could be empirically tested in future behavioral experiments.

4.1 Extensions

The most obvious extension of the present study would be to compare our stripped-down model with the original one. For some of the additional mechanisms proposed by Albright & Hayes this would be straightforward and we have alreay begun to do so; other modifications would require larger changes to the model and enhancements to the training data. For example, **?**, §3.4 motivate a second generalization mechanism that creates *cross-context* (or more jocularly 'Doppelgänger') rules: for each pair of rules $A \rightarrow B / C \_ D$ and $A \rightarrow B' / C' \_ D'$, their model adds $A \rightarrow B' / C \_ D$ and $A \rightarrow B / C' \_ D'$. This is a simple change to our implementation that, using the results of §2, need only apply to base rules.

Learning phonological rules along with morphology, as in **?**, §3.3, would require the training data to contain lexeme frequencies. This is because the original implementation processes the training lexemes in order of descending frequency, ensuring that a phonological rule learned on the basis of one lexeme is consistent with all previous (i.e., higher frequency) training examples. We have not yet begun to explore this or alternative means of incorporating phonology into the model; this is an important extension because, as Albright & Hayes demonstrate, learning fully general morphological rules requires taking into account the downstream effects of phonology. We have also not explored *impugnment* (**?**, §3.7), which unlike the other components of the model seeks to limit rather than expand upon minimal generalization.

4.2 Near misses

As the organizers of the shared task have emphasized, implemented models can be used not only to predict the results of behavioral 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 an automatic method of selecting wug items to investigate a main concern about minimal generalization: namely, that by learning rules in a strictly bottom-up way it will *undergeneralize*, predicting sharp contrasts in inflectional behavior on the basis of slight differences in form.

We illustrate our method with the English irregular pattern ɪ → ʌ, which attracted new members in the history of English and has elicited relatively high production rates and acceptability ratings in previous wug tests (e.g., **??**). We extracted all of the onsets and rimes that appear in the bare forms of monosyllabic English verbs and freely combined them 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 ɪ → ʌ. We further required each item to share its rime with at least one such irregular verb.[10] All of the wugs in the final 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 (*potential hits*), and items that are outside the scope of all such rules (*near misses*). For the former, we recorded the highest-scoring applicable rule. We wanted to provide the model with 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 potential hits and near misses are

[10] Studies of English irregular verbs have focused primarily on vowels and codas of monosyllables, though see **?** on the potential role of onsets.

[11] With this modification to the implementation, which was not used in previous sections, 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.

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 non-coronal fricative immediately before the vowel). Other differences in the onset can also dramatically affect the model's predictions: /θɹɪŋk/ (.88) and /glɪŋ/ (.67) are potential hits but /smɪŋk/ and /smɪŋ/ are near misses. The second two are phonotactically challenged (**?**), but are /θɹʌŋk/ and /glʌŋ/ far superior to /smʌŋk/ and /smʌŋ/ when the phonotactic acceptability of their bare forms is factored out?

The same procedure can be applied to any irregular (or indeed regular) change. For i → ɛpt (as in *sleep* ∼ *slept*), we find that the potential hits include /gip/ (.85) and /flip/ (.73, one of Albright & Hayes's wug items) while /fip/, /vip/, /nip/, and /snip/ are among the near misses. Would native English speakers rate the novel past form /gɛpt/ much higher than /fɛpt/, as the model predicts?[12]

We look forward to future empirical tests of minimal generalization, along these lines and others, as part of the collective effort to find out 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 our study and for stimulating conversations on related topics over many years. This research was partially supported by NSF grant BCS-1844780 to CW.

[12] If irregular pasts of these potential hits and near misses are all rated low, this could instead suggest that the model has *overgeneralized*, perhaps supporting an alternative that uses something more like detailed exemplars to represent and extend irregular patterns.