

Phonological Grammar Induction by Genetic Search

Abstract

1 Introduction

1.1 Computational Phonology and FSAs

Finite-state techniques have always been central to computational phonology. Definite FSAs are commonly used e.g. to encode phonotactics, and are applied to obtain well-formedness judgments on linguistic entities in a variety of NLP contexts (e.g. to reduce the search space for lexical hypotheses (J.Carson-Berndsen, 1993), and for the detection of unknown words (A.Jusek et al., 1997) in speech recognition). Such FSAs are usually constructed manually, and the research presented here aims at providing a practicable automatic method for FSAs construction that helps decrease the cost of manual construction and increases descriptive efficiency.

1.2 Terminology and Notation

Following common notational conventions, a finite-state automaton is a system (S, I, δ, s_0, F) , in which S is a finite nonempty set of states, I is a finite nonempty alphabet, δ is the state transition function, s_0 is the initial state, and F is a nonempty set of final states in S . For every state $s \in S$ and every symbol $a \in I$, $\delta(s, a)$ is either a state in S or ϕ , where ϕ means parse failure. The language L accepted by the finite automaton A , denoted $L(A)$, is $\{x | \delta(s_0, x) \in F\}$.

1.3 Induction of FSAs from Positive Data

The problem of inferring the finite-state automaton A that precisely generates the regular language L from an arbitrary strict subset of L is NP-hard (e.g. E.M.Gold (1978), D.Angluin (1978)). For practical purposes, research has tended to drop the constraint $L(A) = L$, concentrating instead on approximate lan-

guage identification, the heuristic approach of applying prior knowledge to explicitly limit the class of languages of interest, or stochastic grammatical inference.

2 Task Description

Given a known finite alphabet of symbols I , a target finite-state language L , and a data sample $S \subseteq L \subseteq I^*$ (L is finite), genetic search attempts to find an FSA A , such that $L(A) \cap L$ approximates $L(A) = L$, and the size of A approximates the size of the minimal consistent automaton.

Where the target language is known in advance, the degree of approximation can be measured, and its adequacy relative to training set size, or to a given task, can be described. The target language is generally not known in the case of inference of automata that encode (part of) a phonological grammar. Here, approximation and its degree of adequacy can be described relative to a set of theoretical linguistic assumptions that describes a target grammar.

3 Method

3.1 Genetic Search

By direct analogy with natural evolution, GAs work with a population of individuals each of which represents a candidate solution to the given problem. These individuals are assigned a fitness score and on its basis selected to 'mate', and produce the next generation. This process is typically iterated until the population has converged, i.e. when individuals have reached a certain degree of similarity, beyond which further improvement becomes impossible. GAs work because characteristics that form part of good solutions are passed on through the generations and begin to combine in the offspring to approach global optima, an effect

that is known as the *building block hypothesis* (D.E.Goldberg (1989), J.Holland (1975)).

3.2 Representation of Automata as Genotypes

The representation used here is based on the state-transition matrices of automata. It has the advantages that it can be directly implemented as a chromosome, resulting encodings are far more efficient, and search spaces are much smaller (as compared to the more commonly used production rule based representation). This is achieved at the price of loss of detail, the potential disadvantage being that what may be useful minimal building blocks cannot be accessed. Results (described below) indicate that for the purposes of definite FSA induction for phonotactic description, a state-transition based representation scheme is finely grained enough.

3.3 Definition of Fitness

Two fitness (goodness) criteria follow directly from the task description: small number of states (1) and ability to parse strings in the data set (2), where ability to partially parse strings is also rewarded. Used on their own, however, these criteria will lead search toward universal automata that produce all strings $x \in I^*$ up to the length of the longest string in the data set. To avoid this, we add an overgeneration criterion (3) that requires automata to achieve a given degree of overgeneration, such that the size of $L(A)$ is equal to the size of the target language (where the target language is not known, this figure is estimated). Fitness criteria 1-3 are weighted (reflecting their relative importance to fitness evaluation).

Automata can satisfy the third criterion by generating any (ungrammatical) strings to make up the required number. Overgeneration must be constrained so that it becomes meaningful, hence generalisation over the data set. We achieve this by a set of heuristics: (i) sparse labelling (i.e. the elements of I should appear on as few arcs as possible), (ii) similar length paths (paths through the automaton that are shorter than the shortest member of the data set should

be avoided), (iii) functional economy (arcs that are not used by any member of the data set should be avoided). None of these heuristics are absolute requirements. No explicit linguistic knowledge is used.

3.4 Related Research

Inference of regular and context-free grammars with evolutionary techniques is a small but growing field (e.g. Zhou and Grefenstette (1986), Kammeyer and Belew (1996), Lucas (1994), Dupont (1994), Schwehm and Ost (1995), Wyard (1989) and (1991)). Most of this research bases inference on both negative and positive examples, and no real linguistic data sets have been used. Genotype representation is usually production-rule based, and the target grammar is almost always known.

4 Results

4.1 Russian Nouns

The data used in the tests described here were bisyllabic feminine Russian nouns ending in *-a*. The alphabet consisted of 36 phonemic symbols. The data sets comprised 200 strings of which 100 were used during search, and the remainder as a test set.

Results for the first data set are shown in Table 1. The target degree of overgeneration was set to 100. Tests were carried out for different weights assigned to the overgeneration criterion. The first row gives results for the best automaton found in 10 runs with the best weight setting, and the second row gives the corresponding average values for all 10 runs. States refers to the number of states in automata, while Links means the total number of labels on arcs.

As the target automaton was not known here, results are evaluated relative to phonological theory (M.Halle (1971)). Figure 1 shows the fittest automaton from Table 2. Phonemes¹ are grouped together (as label sets on arcs) in

¹The set of phonemic symbols used here is based on Halle (1971). Capital symbols represent sharpened versions of non-capitalised counterparts. Most letters represent a phoneme similar to the sound that the letter would imply to an English speaker. \$ represents /sh/, % its voiced equivalent, c is /ts/, and @ the null symbol.

	Training set	Test set	Actual overgeneration	States	Links
Best automaton	94%	61%	101	7	75
Average (over 10 runs)	87%	52%	118	7.3	75

Table 1: Results for Russian data set 1.

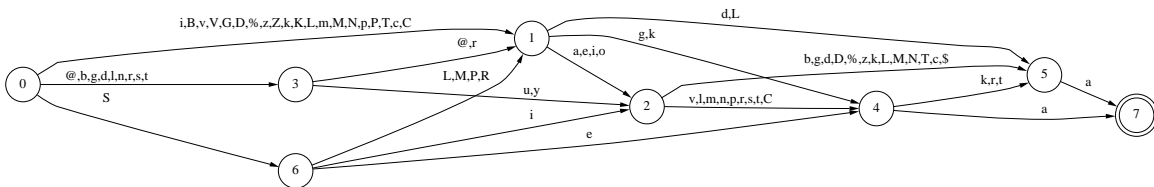


Figure 1: Best automaton for Russian data set 1.

several linguistically useful ways. Vowels and consonants are separated completely. Vowels are separated into the set of those that can be preceded by sharp consonants (capital letters) and those that cannot. Correspondingly, sharp consonants tend to be separated from nonsharped equivalents. The phonemes **k**, **r**, **t** are singled out (arc $4 \rightarrow 5$) because (in the present data set) they combine only with nonsharped consonants to form stem-final clusters. The groupings **S** ($0 \rightarrow 6$) and **L,M,P,R** ($6 \rightarrow 1$) reflect stem-initial consonant clusters.

These groupings are typical of the automata discovered in all 10 runs. Occasionally **ka** was singled out as a separate ending, and the stem vowels were frequently grouped together more efficiently. Thus, linguistically useful generalisation was achieved, but search failed to find an automaton that could parse more than 94% of the *training* data set. Analysis of results collected for the 500,000 individuals produced in all generations of 100 different runs showed that 15 strings were parsed by far fewer automata than any of the remaining 85. All of these 15 strings were found to contain stem-initial or stem-final consonant clusters that occur only once in the data set, while the other 85 strings form clusters in phonological space. It appeared that if a data set contains isolated points as well as clusters in phonological space,

then the isolated points are unlikely to be learnt. To test this hypothesis, we ran a second set of experiments after replacing the 15 difficult strings with strings located in one of the clusters already present in the data set. The test set, parameter settings and random initialisation were the same as before. Results (shown in Table 2) confirmed the hypothesis. Out of 1000 automata that were in the final populations of 10 runs, only 2 failed to parse the entire data set. Test set performance did not improve, however, an indication that generalisability was not affected.

It should be pointed out that the members of the second data set do not form a single cluster in phonological space, but several clusters of varying size and density, with varying distances in between. The difference in learnability is between single points and clusters and does not appear to depend on the overall homogeneity of the data set.

5 Conclusion and Further Research

The results presented here indicate that genetic search can successfully be applied to the automatic discovery of definite finite-state automata that encode phonological grammars from subsets of positive data, but that results are better for data sets that contain no isolated points in phonological space. We are currently applying

	Training set	Test set	Actual overgeneration	States	Links
Best automaton	100%	59%	100	7	80
Average (over 10 runs)	99%	54%	158	7.6	74

Table 2: Results for Russian data set 2.

the method to German syllable phonotactics, and plan to extend the approach to feature-based phonotactic description.

References

- A.Jusek, H.Rautenstrauch, G.A.Fink, F.Kummert, G.Sagerer, J.Carson-Berndsen, and D.Gibbon. 1997. Detektion unbekannter woerter mit hilfe phonotaktischer modelle. Technical report, Fakultae fuer Linguistik und Literaturwissenschaft, University of Bielefeld.
- D.Angluin. 1978. On the complexity of minimum inference of regular sets. *Information and Control*, 39:337–350.
- D.E.Goldberg. 1989. *Genetic Algorithms in search, optimization and machine learning*. Addison-Wesley.
- E.M.Gold. 1978. Complexity of automaton identification from given data. *Information and Control*, 37:302–320.
- H.Zhou and J.J.Grefenstette. 1986. Induction of finite automata by genetic algorithms. *Proceedings of the 1986 International Conference on Systems, Man and Cybernetics*, pages 170–174.
- J.Carson-Berndsen. 1993. An event-based phonotactics for german. Technical Report ASL-TR-29-92/UBI, Fakultae fuer Linguistik und Literaturwissenschaft, University of Bielefeld.
- J.Holland. 1975. *Adaptation in Natural and Artificial Systems*. MIT Press.
- M.Halle. 1971. *The Sound Pattern of Russian*. Mouton, The Hague.
- M.Schwehm and A.Ost. 1995. Inference of stochastic regular grammars by massively parallel genetic algorithms. In *Proceedings of the Sixth International Conference on Genetic Algorithms*, pages 520–527. Morgan Kaufmann.
- P.Dupont. 1994. Regular grammatical inference from positive and negative samples by genetic search: the gig method. In *Grammatical Inference and Applications, Second International Colloquium, ICGI-94, Proceedings*, Berlin. Springer.
- P.Wyard. 1989. Representational issues for context free grammar induction using genetic algorithms. Technical report, Natural Language Group, Systems Research Division, BT Laboratories, Ipswich, UK.
- S.Lucas. 1994. Context-free grammar evolution. In *First International Conference on Evolutionary Computing*, pages 130–135.
- T.E.Kammeyer and R.K.Below. 1996. Stochastic context-free grammar induction with a genetic algorithm using local search. Technical Report CS96-476, Cognitive Computer Science Research Group, Computer Science and Engineering Department, University of California at San Diego.
- P. Wyard. 1991. Context free grammar induction using genetic algorithms. In Richard K. Below and Lashon B. Booker, editors, *Proceedings of the Fourth International Conference on Genetic Algorithms*, pages 514–518, San Diego, CA. Morgan Kaufmann.