Estimating Strictly Piecewise Distributions

Jeffrey Heinz
University of Delaware
Newark, Delaware, USA
heinz@udel.edu

James Rogers
Earlham College
Richmond, Indiana, USA
jrogers@quark.cs.earlham.edu

Abstract

Strictly Piecewise (SP) languages are a subclass of regular languages which encode certain kinds of long-distance dependencies that are found in natural languages. Like the classes in the Chomsky and Subregular hierarchies, there are many independently converging characterizations of the SP class (Rogers et al., to appear). Here we define SP distributions and show that they can be efficiently estimated from positive data.

1 Introduction

Long-distance dependencies in natural language are of considerable interest. Although much attention has focused on long-distance dependencies which are beyond the expressive power of models with finitely many states (Chomsky, 1956; Joshi, 1985; Shieber, 1985; Kobele, 2006), there are some long-distance dependencies in natural language which permit finite-state characterizations. For example, although it is well-known that vowel and consonantal harmony applies across any arbitrary number of intervening segments (Ringen, 1988; Baković, 2000; Hansson, 2001; Rose and Walker, 2004) and that phonological patterns are regular (Johnson, 1972; Kaplan and Kay, 1994), it is less well-known that harmony patterns are largely characterizable by the Strictly Piecewise languages, a subregular class of languages with independently-motivated, converging characterizations (see Heinz (2007, to appear) and especially Rogers et al. (2009)).

As shown by Rogers et al. (to appear), the Strictly Piecewise (SP) languages, which make distinctions on the basis of (potentially) discontiguous subsequences, are precisely analogous to the Strictly Local (SL) languages (McNaughton and Papert, 1971; Rogers and Pullum, to appear), which make distinctions on the basis of contiguous subsequences. The Strictly Local languages are the formal-language theoretic foundation for n-gram models (Garcia et al., 1990), which are widely used in natural language processing (NLP) in part because such distributions can be estimated from positive data (i.e. a corpus) (Jurafsky and Martin, 2008). N-gram models describe probability distributions over all strings on the basis of the Markov assumption (Markov, 1913): that the probability of the next symbol only depends on the previous contiguous sequence of length $n-1$. From the perspective of formal language theory, these distributions are perhaps properly called Strictly k-Local distributions (SL_k) where $k = n$. It is well-known that one limitation of the Markov assumption is its inability to express any kind of long-distance dependency.

This paper defines Strictly k-Piecewise (SP_k) distributions and shows how they too can be efficiently estimated from positive data. In contrast with the Markov assumption, our assumption is that the probability of the next symbol is conditioned on the previous set of discontiguous subsequences of length $k-1$ in the string. While this suggests the model has too many parameters (one for each subset of all possible subsequences), in fact the model has on the order of $|\Sigma|^{k+1}$ parameters because of an independence assumption: there is no interaction between different subsequences. As a result, SP distributions are efficiently computable even though they condition the probability of the next symbol on the occurrences of earlier (possibly very distant) discontiguous subsequences. Essentially, these SP distributions reflect a kind of long-term memory.

On the other hand, SP models have no short-term memory and are unable to make distinctions on the basis of contiguous subsequences. We do not intend SP models to replace n-gram models, but instead expect them to be used alongside of

Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics, pages 886–896,
Uppsala, Sweden, 11-16 July 2010.

them. Exactly how this is to be done is beyond the scope of this paper and is left for future research.

Since SP languages are the analogue of SL languages, which are the formal-language theoretical foundation for n-gram models, which are widely used in NLP, it is expected that SP distributions and their estimation will also find wide application. Apart from their interest to problems in theoretical phonology such as phonotactic learning (Coleman and Pierrehumbert, 1997; Hayes and Wilson, 2008; Heinz, to appear), it is expected that their use will have application, in conjunction with n-gram models, in areas that currently use them; e.g. augmentative communication (Newell et al., 1998), part of speech tagging (Brill, 1995), and speech recognition (Jelenik, 1997).

§2 provides basic mathematical notation. §3 provides relevant background on the subregular hierarchy. §4 describes automata-theoretic characterizations of SP languages. §5 defines SP distributions. §6 shows how these distributions can be efficiently estimated from positive data and provides a demonstration. §7 concludes the paper.

2 Preliminaries

We start with some mostly standard notation. Σ denotes a finite set of symbols and a string over Σ is a finite sequence of symbols drawn from that set. Σ^k, $\Sigma^{\leq k}$, $\Sigma^{\geq k}$, and Σ^* denote all strings over this alphabet of length k, of length less than or equal to k, of length greater than or equal to k, and of any finite length, respectively. ϵ denotes the empty string. $|w|$ denotes the length of string w. The prefixes of a string w are $\text{Pfx}(w) = \{v : \exists u \in \Sigma^* \text{ such that } vu = w\}$. When discussing partial functions, the notation $\uparrow$ and $\downarrow$ indicates that the function is undefined, respectively is defined, for particular arguments.

A *language* L is a subset of Σ^*. A *stochastic language* $\mathcal{D}$ is a probability distribution over Σ^*. The probability p of word w with respect to $\mathcal{D}$ is written $Pr_{\mathcal{D}}(w) = p$. Recall that all distributions $\mathcal{D}$ must satisfy $\sum_{w \in \Sigma^*} Pr_{\mathcal{D}}(w) = 1$. If L is language then $Pr_{\mathcal{D}}(L) = \sum_{w \in L} Pr_{\mathcal{D}}(w)$.

A *Deterministic Finite-state Automaton* (DFA) is a tuple $\mathcal{M} = \langle Q, \Sigma, q_0, \delta, F \rangle$ where Q is the state set, Σ is the alphabet, q_0 is the start state, δ is a deterministic transition function with domain $Q \times \Sigma$ and codomain Q, F is the set of accepting states. Let $\hat{d} : Q \times \Sigma^* \rightarrow Q$ be the (partial) path function of $\mathcal{M}$, i.e., $\hat{d}(q, w)$ is the (unique) state reachable from state q via the sequence w, if any, or $\hat{d}(q, w)\uparrow$ otherwise. The language recognized by a DFA $\mathcal{M}$ is $L(\mathcal{M}) \stackrel{\text{def}}{=} \{w \in \Sigma^* \mid \hat{d}(q_0, w)\downarrow \in F\}$.

A state is *useful* iff for all $q \in Q$, there exists $w \in \Sigma^*$ such that $\delta(q_0, w) = q$ and there exists $w \in \Sigma^*$ such that $\delta(q, w) \in F$. *Useless* states are not useful. DFAs without useless states are *trimmed*.

Two strings w and v over Σ are *distinguished* by a DFA $\mathcal{M}$ iff $\hat{d}(q_0, w) \neq \hat{d}(q_0, v)$. They are *Nerode equivalent* with respect to a language L if and only if $wu \in L \iff vu \in L$ for all $u \in \Sigma^*$. All DFAs which recognize L must distinguish strings which are inequivalent in this sense, but no DFA recognizing L necessarily distinguishes any strings which are equivalent. Hence the number of equivalence classes of strings over Σ modulo Nerode equivalence with respect to L gives a (tight) lower bound on the number of states required to recognize L.

A DFA is *minimal* if the size of its state set is minimal among DFAs accepting the same language. The *product* of n DFAs $\mathcal{M}_1 \ldots \mathcal{M}_n$ is given by the standard construction over the state space $Q_1 \times \ldots \times Q_n$ (Hopcroft et al., 2001).

A *Probabilistic Deterministic Finite-state Automaton* (PDFA) is a tuple $\mathcal{M} = \langle Q, \Sigma, q_0, \delta, F, T \rangle$ where Q is the state set, Σ is the alphabet, q_0 is the start state, δ is a deterministic transition function, F and T are the final-state and transition probabilities. In particular, $T : Q \times \Sigma \rightarrow \mathbb{R}^+$ and $F : Q \rightarrow \mathbb{R}^+$ such that

$$\text{for all } q \in Q,\ F(q) + \sum_{a \in \Sigma} T(q, a) = 1. \quad (1)$$

Like DFAs, for all $w \in \Sigma^*$, there is at most one state reachable from q_0. PDFAs are typically represented as labeled directed graphs as in Figure 1.

A PDFA $\mathcal{M}$ generates a stochastic language $\mathcal{D}_{\mathcal{M}}$. If it exists, the (unique) *path* for a word $w = a_0 \ldots a_k$ belonging to Σ^* through a PDFA is a sequence $\langle (q_0, a_0), (q_1, a_1), \ldots, (q_k, a_k) \rangle$, where $q_{i+1} = \delta(q_i, a_i)$. The probability a PDFA assigns to w is obtained by multiplying the transition probabilities with the final probability along w's path if

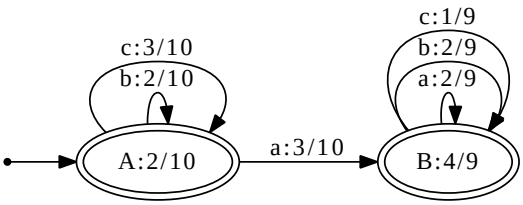

Figure 1: A picture of a PDFA with states labeled A and B. The probabilities of T and F are located to the right of the colon.

it exists, and zero otherwise.

$$Pr_{\mathcal{D}_\mathcal{M}}(w) = \left(\prod_{i=1}^{k} T(q_{i-1}, a_{i-1})\right) \cdot F(q_{k+1}) \quad (2)$$

if $\hat{d}(q_0, w)\downarrow$ and 0 otherwise

A probability distribution is *regular deterministic* iff there is a PDFA which generates it.

The *structural components* of a PDFA $\mathcal{M}$ are its states Q, its alphabet Σ, its transitions δ, and its initial state q_0. By *structure* of a PDFA, we mean its structural components. Each PDFA $\mathcal{M}$ defines a family of distributions given by the possible instantiations of T and F satisfying Equation 1. These distributions have $|Q| \cdot (|\Sigma| + 1)$ independent parameters (since for each state there are $|\Sigma|$ possible transitions plus the possibility of finality.)

We define the product of PDFA in terms of *co-emission probabilities* (Vidal et al., 2005a).

Definition 1 *Let $\mathcal{A}$ be a vector of PDFAs and let $|\mathcal{A}| = n$. For each $1 \leq i \leq n$ let $\mathcal{M}_i = \langle Q_i, \Sigma, q_{0i}, \delta_i, F_i, T_i \rangle$ be the ith PDFA in $\mathcal{A}$. The probability that σ is co-emitted from $q_1, \ldots, q_n$ in $Q_1, \ldots, Q_n$, respectively, is*

$$CT(\langle \sigma, q_1 \ \ldots \ q_n \rangle) = \prod_{i=1}^{n} T_i(q_i, \sigma).$$

Similarly, the probability that a word simultaneously ends at $q_1 \in Q_1 \ \ldots \ q_n \in Q_n$ is

$$CF(\langle q_1 \ \ldots \ q_n \rangle) = \prod_{i=1}^{n} F_i(q_i).$$

Then $\bigotimes \mathcal{A} = \langle Q, \Sigma, q_0, \delta, F, T \rangle$ where

1. *Q, q_0, and δ are defined as with DFA product.*
2. *For all $\langle q_1 \ \ldots \ q_n \rangle \in Q$, let $Z(\langle q_1 \ \ldots \ q_n \rangle) =$*

$$CF(\langle q_1 \ \ldots \ q_n \rangle) + \sum_{\sigma \in \Sigma} CT(\langle \sigma, q_1 \ \ldots \ q_n \rangle)$$

be the normalization term*; and*

   (a) *let $F(\langle q_1 \ \ldots \ q_n \rangle) = \frac{CF(\langle q_1 \ \ldots \ q_n \rangle)}{Z(\langle q_1 \ \ldots \ q_n \rangle)}$; and*

   (b) *for all $\sigma \in \Sigma$, let $T(\langle q_1 \ \ldots \ q_n \rangle, \sigma) = \frac{CT(\langle \sigma, q_1 \ \ldots \ q_n \rangle)}{Z(\langle q_1 \ \ldots \ q_n \rangle)}$*

In other words, the numerators of T and F are defined to be the co-emission probabilities (Vidal et al., 2005a), and division by Z ensures that $\mathcal{M}$ defines a well-formed probability distribution. Statistically speaking, the co-emission product makes an independence assumption: the probability of σ being co-emitted from $q_1, \ldots, q_n$ is exactly what one expects if there is no interaction between the individual factors; that is, between the probabilities of σ being emitted from any q_i. Also note order of product is irrelevant up to renaming of the states, and so therefore we also speak of taking the product of a set of PDFAs (as opposed to an ordered vector).

Estimating regular deterministic distributions is well-studied problem (Vidal et al., 2005a; Vidal et al., 2005b; de la Higuera, in press). We limit discussion to cases when the structure of the PDFA is known. Let S be a finite sample of words drawn from a regular deterministic distribution $\mathcal{D}$. The problem is to estimate parameters T and F of $\mathcal{M}$ so that $\mathcal{D}_\mathcal{M}$ approaches $\mathcal{D}$. We employ the widely-adopted maximum likelihood (ML) criterion for this estimation.

$$(\hat{T}, \hat{F}) = \underset{T,F}{\operatorname{argmax}} \left(\prod_{w \in S} Pr_\mathcal{M}(w)\right) \quad (3)$$

It is well-known that if $\mathcal{D}$ is generated by some PDFA $\mathcal{M}'$ with the same structural components as $\mathcal{M}$, then optimizing the ML estimate guarantees that $\mathcal{D}_\mathcal{M}$ approaches $\mathcal{D}$ as the size of S goes to infinity (Vidal et al., 2005a; Vidal et al., 2005b; de la Higuera, in press).

The optimization problem (3) is simple for deterministic automata with known structural components. Informally, the corpus is passed through the PDFA, and the paths of each word through the corpus are tracked to obtain counts, which are then normalized by state. Let $\mathcal{M} = \langle Q, \Sigma, \delta, q_0, F, T \rangle$ be the PDFA whose parameters F and T are to be estimated. For all states $q \in Q$ and symbols $a \in \Sigma$, The ML estimation of the probability of $T(q, a)$ is obtained by dividing the number of times this transition is used in parsing the sample S by the

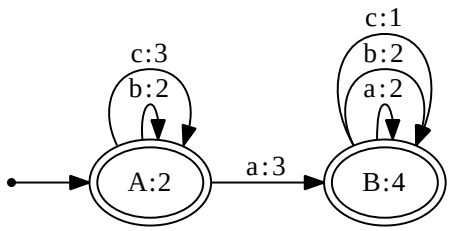

Figure 2: The automata shows the counts obtained by parsing $\mathcal{M}$ with sample $S = \{ab, bba, \epsilon, cab, acb, cc\}$.

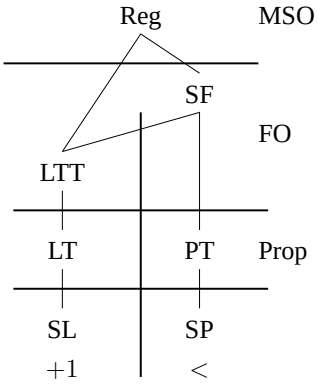

Figure 3: Parallel Sub-regular Hierarchies.

number of times state q is encountered in the parsing of S. Similarly, the ML estimation of $F(q)$ is obtained by calculating the relative frequency of state q being final with state q being encountered in the parsing of S. For both cases, the division is *normalizing*; i.e. it guarantees that there is a well-formed probability distribution at each state. Figure 2 illustrates the counts obtained for a machine $\mathcal{M}$ with sample $S = \{ab, bba, \epsilon, cab, acb, cc\}$.[1] Figure 1 shows the PDFA obtained after normalizing these counts.

3 Subregular Hierarchies

Within the class of regular languages there are dual hierarchies of language classes (Figure 3), one in which languages are defined in terms of their *contiguous substrings* (up to some length k, known as k*-factors*), starting with the languages that are *Locally Testable in the Strict Sense* (SL), and one in which languages are defined in terms of their not necessarily contiguous *subsequences*, starting with the languages that are *Piecewise Testable in the Strict Sense* (SP). Each language class in these hierarchies has independently motivated, converging characterizations and each has been claimed to correspond to specific, fundamental cognitive capabilities (McNaughton and Papert, 1971; Brzozowski and Simon, 1973; Simon, 1975; Thomas, 1982; Perrin and Pin, 1986; García and Ruiz, 1990; Beauquier and Pin, 1991; Straubing, 1994; García and Ruiz, 1996; Rogers and Pullum, to appear; Kontorovich et al., 2008; Rogers et al., to appear).

Languages in the weakest of these classes are defined only in terms of the set of factors (SL) or subsequences (SP) which are licensed to occur in the string (equivalently the complement of that set with respect to $\Sigma^{\leq k}$, the *forbidden factors* or *forbidden subsequences*). For example, the set containing the forbidden 2-factors $\{ab, ba\}$ defines a Strictly 2-Local language which includes all strings except those with contiguous substrings $\{ab, ba\}$. Similarly since the parameters of n-gram models (Jurafsky and Martin, 2008) assign probabilities to symbols given the preceding contiguous substrings up to length $n-1$, we say they describe Strictly n-Local distributions.

These hierarchies have a very attractive model-theoretic characterization. The *Locally Testable* (LT) and *Piecewise Testable* languages are exactly those that are definable by propositional formulae in which the atomic formulae are blocks of symbols interpreted factors (LT) or subsequences (PT) of the string. The languages that are testable in the strict sense (SL and SP) are exactly those that are definable by formulae of this sort restricted to conjunctions of negative literals. Going the other way, the languages that are definable by First-Order formulae with adjacency (successor) but not precedence (less-than) are exactly the *Locally Threshold Testable* (LTT) languages. The *Star-Free* languages are those that are First-Order definable with precedence alone (adjacency being FO definable from precedence). Finally, by extending to Monadic Second-Order formulae (with either signature, since they are MSO definable from each other), one obtains the full class of Regular languages (McNaughton and Papert, 1971; Thomas, 1982; Rogers and Pullum, to appear; Rogers et al., to appear).

The relation between strings which is fundamental along the Piecewise branch is the *subse-*

[1] Technically, this acceptor is neither a simple DFA or PDFA; rather, it has been called a Frequency DFA. We do not formally define them here, see (de la Higuera, in press).

quence relation, which is a partial order on Σ^*:

$$w \sqsubseteq v \overset{\text{def}}{\iff} w = \varepsilon \text{ or } w = \sigma_1 \cdots \sigma_n \text{ and}$$
$$(\exists w_0, \ldots, w_n \in \Sigma^*)[v = w_0\sigma_1 w_1 \cdots \sigma_n w_n].$$

in which case we say w is a *subsequence* of v.

For $w \in \Sigma^*$, let

$$\mathrm{P}_k(w) \overset{\text{def}}{=} \{v \in \Sigma^k \mid v \sqsubseteq w\} \text{ and}$$
$$\mathrm{P}_{\leq k}(w) \overset{\text{def}}{=} \{v \in \Sigma^{\leq k} \mid v \sqsubseteq w\},$$

the set of subsequences of length k, respectively length no greater than k, of w. Let $\mathrm{P}_k(L)$ and $\mathrm{P}_{\leq k}(L)$ be the natural extensions of these to sets of strings. Note that $\mathrm{P}_0(w) = \{\varepsilon\}$, for all $w \in \Sigma^*$, that $\mathrm{P}_1(w)$ is the set of symbols occurring in w and that $\mathrm{P}_{\leq k}(L)$ is finite, for all $L \subseteq \Sigma^*$.

Similar to the Strictly Local languages, Strictly Piecewise languages are defined only in terms of the set of subsequences (up to some length k) which are licensed to occur in the string.

Definition 2 (SP_k **Grammar,** SP**)** *A* SP_k *grammar is a pair* $\mathcal{G} = \langle \Sigma, G \rangle$ *where* $G \subseteq \Sigma^k$. *The language licensed by a* SP_k *grammar is*

$$L(\mathcal{G}) \overset{\text{def}}{=} \{w \in \Sigma^* \mid P_{\leq k}(w) \subseteq P_{\leq k}(G)\}.$$

A language is SP_k iff it is $L(\mathcal{G})$ for some SP_k grammar $\mathcal{G}$. It is SP iff it is SP_k for some k.

This paper is primarily concerned with estimating Strictly Piecewise distributions, but first we examine in greater detail properties of SP languages, in particular DFA representations.

4 DFA representations of SP Languages

Following Sakarovitch and Simon (1983), Lothaire (1997) and Kontorovich, et al. (2008), we call the set of strings that contain w as a subsequence the *principal shuffle ideal*[2] of w:

$$\mathrm{SI}(w) = \{v \in \Sigma^* \mid w \sqsubseteq v\}.$$

The *shuffle ideal* of a set of strings is defined as

$$\mathrm{SI}(S) = \cup_{w \in S} \mathrm{SI}(w)$$

Rogers et al. (to appear) establish that the SP languages have a variety of characteristic properties.

Theorem 1 *The following are equivalent:*[3]

[2]Properly $\mathrm{SI}(w)$ is the principal ideal generated by $\{w\}$ wrt the inverse of $\sqsubseteq$.

[3]For a complete proof, see Rogers et al. (to appear). We only note that 5 implies 1 by DeMorgan's theorem and the fact that every shuffle ideal is finitely generated (see also Lothaire (1997)).

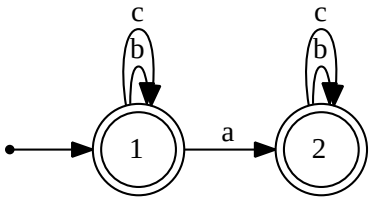

Figure 4: The DFA representation of $\overline{\mathrm{SI}(aa)}$.

1. $L = \bigcap_{w \in S} [\overline{\mathrm{SI}(w)}],\ S$ *finite,*
2. $L \in$ SP
3. $(\exists k)[P_{\leq k}(w) \subseteq P_{\leq k}(L) \Rightarrow w \in L]$,
4. $w \in L$ *and* $v \sqsubseteq w \Rightarrow v \in L$ *(*L *is* subsequence closed*),*
5. $L = \overline{\mathrm{SI}(X)},\ X \subseteq \Sigma^*$ *(*L *is the complement of a shuffle ideal).*

The DFA representation of the complement of a shuffle ideal is especially important.

Lemma 1 *Let* $w \in \Sigma^k$, $w = \sigma_1 \cdots \sigma_k$, *and* $\mathcal{M}_{\overline{\mathrm{SI}(w)}} = \langle Q, \Sigma, q_0, \delta, F \rangle$, *where* $Q = \{i \mid 1 \leq i \leq k\}$, $q_0 = 1$, $F = Q$ *and for all* $q_i \in Q, \sigma \in \Sigma$*:*

$$\delta(q_i, \sigma) = \begin{cases} q_{i+1} & \text{if } \sigma = \sigma_i \text{ and } i < k, \\ \uparrow & \text{if } \sigma = \sigma_i \text{ and } i = k, \\ q_i & \text{otherwise.} \end{cases}$$

Then $\mathcal{M}_{\overline{\mathrm{SI}(w)}}$ *is a minimal, trimmed DFA that recognizes the complement of* $\mathrm{SI}(w)$*, i.e.,* $\overline{\mathrm{SI}(w)} = L(\mathcal{M}_{\overline{\mathrm{SI}(w)}})$.

Figure 4 illustrates the DFA representation of the complement of $\mathrm{SI}(aa)$ with $\Sigma = \{a, b, c\}$. It is easy to verify that the machine in Figure 4 accepts all and only those words which do not contain an aa subsequence.

For any SP_k language $L = L(\langle \Sigma, G \rangle) \neq \Sigma^*$, the first characterization (1) in Theorem 1 above yields a non-deterministic finite-state representation of L, which is a set $\mathcal{A}$ of DFA representations of complements of principal shuffle ideals of the elements of G. The trimmed automata product of this set yields a DFA, with the properties below (Rogers et al., to appear).

Lemma 2 *Let* $\mathcal{M}$ *be a trimmed DFA recognizing a* SP_k *language constructed as described above. Then:*

1. *All states of* $\mathcal{M}$ *are accepting states:* $F = Q$.

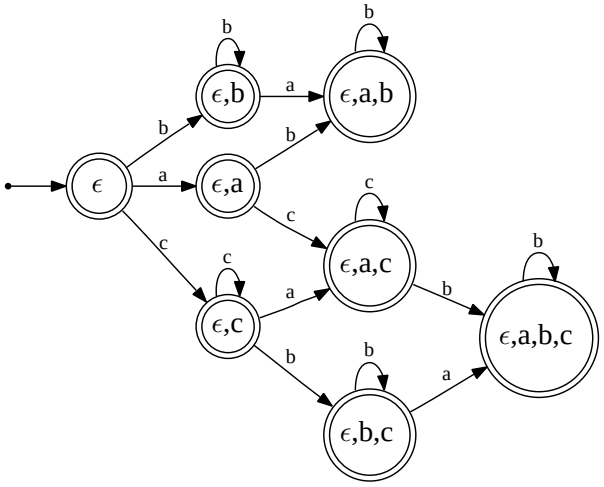

Figure 5: The DFA representation of the of the SP language given by $\mathcal{G} = \langle\{a, b, c\}, \overline{\{aa, bc\}}\rangle$. Names of the states reflect subsets of subsequences up to length 1 of prefixes of the language. Note this DFA is trimmed, but not minimal.

2. *For all $q_1, q_2 \in Q$ and $\sigma \in \Sigma$, if $\hat{d}(q_1, \sigma)\uparrow$ and $\hat{d}(q_1, w) = q_2$ for some $w \in \Sigma^*$ then $\hat{d}(q_2, \sigma)\uparrow$. (Missing edges propagate down.)*

Figure 5 illustrates with the DFA representation of the of the SP_2 language given by $\mathcal{G} = \langle\{a, b, c\}, \overline{\{aa, bc\}}\rangle$. It is straightforward to verify that this DFA is identical (modulo relabeling of state names) to one obtained by the trimmed product of the DFA representations of the complement of the principal shuffle ideals of aa and bc, which are the prohibited subsequences.

States in the DFA in Figure 5 correspond to the subsequences up to length 1 of the prefixes of the language. With this in mind, it follows that the DFA of $\Sigma^* = L(\Sigma, \Sigma^k)$ has states which correspond to the subsequences up to length $k-1$ of the prefixes of Σ^*. Figure 6 illustrates such a DFA when $k = 2$ and $\Sigma = \{a, b, c\}$.

In fact, these DFAs reveal the differences between SP languages and PT languages: they are exactly those expressed in Lemma 2. Within the state space defined by the subsequences up to length $k-1$ of the prefixes of the language, if the conditions in Lemma 2 are violated, then the DFAs describe languages that are PT but not SP. Pictorially, PT_2 languages are obtained by arbitrarily removing arcs, states, and the finality of states from the DFA in Figure 6, and SP_2 ones are obtained by non-arbitrarily removing them in accordance with Lemma 2. The same applies straightforwardly for any k (see Definition 3 below).

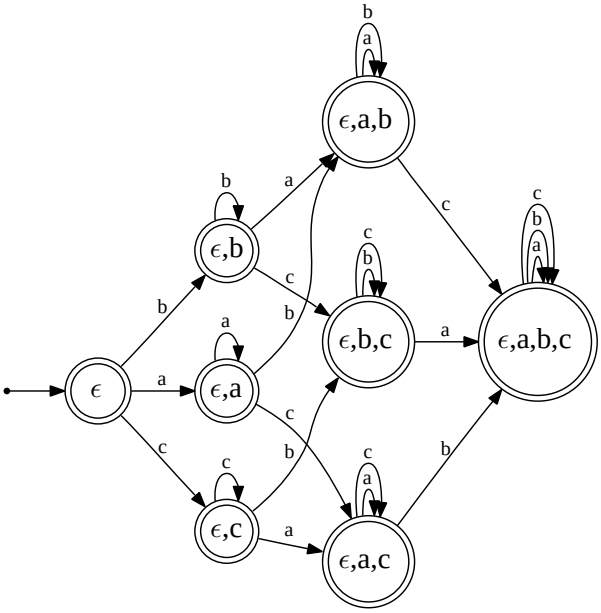

Figure 6: A DFA representation of the of the SP_2 language given by $\mathcal{G} = \langle\{a, b, c\}, \Sigma^2\rangle$. Names of the states reflect subsets of subsequences up to length 1 of prefixes of the language. Note this DFA is trimmed, but not minimal.

5 SP Distributions

In the same way that SL distributions (n-gram models) generalize SL languages, SP distributions generalize SP languages. Recall that SP languages are characterizable by the intersection of the complements of principal shuffle ideals. SP distributions are similarly characterized.

We begin with Piecewise-Testable distributions.

Definition 3 *A distribution $\mathcal{D}$ is k-Piecewise Testable (written $\mathcal{D} \in \text{PTD}_k$) $\overset{def}{\iff}$ $\mathcal{D}$ can be described by a PDFA $\mathcal{M} = \langle Q, \Sigma, q_0, \delta, F, T\rangle$ with*

1. $Q = \{P_{\leq k-1}(w) : w \in \Sigma^*\}$
2. $q_0 = P_{\leq k-1}(\epsilon)$
3. *For all $w \in \Sigma^*$ and all $\sigma \in \Sigma$, $\delta(P_{\leq k-1}(w), a) = P_{\leq k-1}(wa)$*
4. *F and T satisfy Equation 1.*

In other words, a distribution is k-Piecewise Testable provided it can be represented by a PDFA whose structural components are the same (modulo renaming of states) as those of the DFA discussed earlier where states corresponded to the subsequences up to length $k-1$ of the prefixes of the language. The DFA in Figure 6 shows the

structure of a PDFA which describes a PT_2 distribution as long as the assigned probabilities satisfy Equation 1.

The following lemma follows directly from the finite-state representation of PT_k distributions.

Lemma 3 *Let $\mathcal{D}$ belong to PTD_k and let $\mathcal{M} = \langle Q, \Sigma, q_0, \delta, F, T\rangle$ be a PDFA representing $\mathcal{D}$ defined according to Definition 3.*

$$Pr_D(\sigma_1 \dots \sigma_n) = T(P_{\leq k-1}(\epsilon), \sigma_1) \cdot \left(\prod_{2 \leq i \leq n} T(P_{\leq k-1}(\sigma_1 \dots \sigma_{i-1}), \sigma_i) \right) \cdot F(P_{\leq k-1}(w)) \quad (4)$$

PT_k distributions have $2^{|\Sigma|^{k-1}}(|\Sigma|+1)$ parameters (since there are $2^{|\Sigma|^{k-1}}$ states and $|\Sigma|+1$ possible events, i.e. transitions and finality).

Let $Pr(\sigma \mid \#)$ and $Pr(\# \mid P_{\leq k}(w))$ denote the probability (according to some $\mathcal{D} \in \text{PTD}_k$) that a word begins with σ and ends after observing $P_{\leq k}(w)$. Then Equation 4 can be rewritten in terms of conditional probability as

$$Pr_D(\sigma_1 \dots \sigma_n) = Pr(\sigma_1 \mid \#) \cdot \left(\prod_{2 \leq i \leq n} Pr(\sigma_i \mid P_{\leq k-1}(\sigma_1 \dots \sigma_{i-1})) \right) \cdot Pr(\# \mid P_{\leq k-1}(w)) \quad (5)$$

Thus, the probability assigned to a word depends not on the observed contiguous sequences as in a Markov model, but on observed subsequences.

Like SP languages, SP distributions can be defined in terms of the product of machines very similar to the complement of principal shuffle ideals.

Definition 4 *Let $w \in \Sigma^{k-1}$ and $w = \sigma_1 \cdots \sigma_{k-1}$. $\mathcal{M}_w = \langle Q, \Sigma, q_0, \delta, F, T\rangle$ is a* w-subsequence-distinguishing PDFA *(w-SD-PDFA) iff $Q = \text{Pfx}(w)$, $q_0 = \epsilon$, for all $u \in \text{Pfx}(w)$ and each $\sigma \in \Sigma$,*

$$\delta(u, \sigma) = \begin{array}{l} u\sigma \text{ iff } u\sigma \in \text{Pfx}(w) \text{ and} \\ u \text{ otherwise} \end{array}$$

and F and T satisfy Equation 1.

Figure 7 shows the structure of $\mathcal{M}_a$ which is almost the same as the complement of the principal shuffle ideal in Figure 4. The only difference is the additional self-loop labeled a on the rightmost state labeled a. $\mathcal{M}_a$ defines a family of distributions over Σ^*, and its states distinguish those

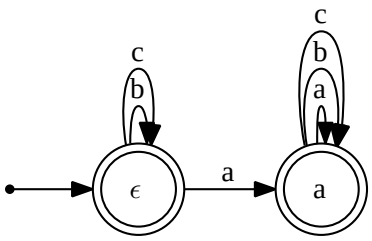

Figure 7: The structure of PDFA $\mathcal{M}_a$. It is the same (modulo state names) as the DFA in Figure 4 except for the self-loop labeled a on state a.

strings which contain a (state a) from those that do not (state ϵ). A set of PDFAs is *a k-set of SD-PDFAs* iff, for each $w \in \Sigma^{\leq k-1}$, it contains exactly one w-SD-PDFA.

In the same way that missing edges propagate down in DFA representations of SP languages (Lemma 2), the final and transitional probabilities must propagate down in PDFA representations of SP_k distributions. In other words, the final and transitional probabilities at states further along paths beginning at the start state must be determined by final and transitional probabilities at earlier states non-increasingly. This is captured by defining SP distributions as a product of k-sets of SD-PDFAs (see Definition 5 below).

While the standard product based on co-emission probability could be used for this purpose, we adopt a modified version of it defined for k-sets of SD-PDFAs: the *positive co-emission probability*. The automata product based on the positive co-emission probability not only ensures that the probabilities propagate as necessary, but also that such probabilities are made on the basis of observed subsequences, and not unobserved ones. This idea is familiar from n-gram models: the probability of σ_n given the immediately preceding sequence $\sigma_1 \dots \sigma_{n-1}$ does not depend on the probability of σ_n given the other $(n-1)$-long sequences which do not immediately precede it, though this is a logical possibility.

Let $\mathcal{A}$ be a k-set of SD-PDFAs. For each $w \in \Sigma^{\leq k-1}$, let $\mathcal{M}_w = \langle Q_w, \Sigma, q_{0w}, \delta_w, F_w, T_w\rangle$ be the w-subsequence-distinguishing PDFA in $\mathcal{A}$. The positive co-emission probability that σ is simultaneously emitted from states $q_\epsilon, \dots, q_u$ from the statesets $Q_\epsilon, \dots Q_u$, respectively, of each SD-

PDFA in $\mathcal{A}$ is

$$PCT(\langle \sigma, q_\epsilon \ \ldots \ q_u \rangle) = \prod_{\substack{q_w \in \langle q_\epsilon \ldots q_u \rangle \\ q_w = w}} T_w(q_w, \sigma) \quad (6)$$

Similarly, the probability that a word simultaneously ends at n states $q_\epsilon \in Q_\epsilon, \ldots, \ q_u \in Q_u$ is

$$PCF(\langle q_\epsilon \ \ldots \ q_u \rangle) = \prod_{\substack{q_w \in \langle q_\epsilon \ldots q_u \rangle \\ q_w = w}} F_w(q_w) \quad (7)$$

In other words, the positive co-emission probability is the product of the probabilities restricted to those assigned to the maximal states in each $\mathcal{M}_w$. For example, consider a 2-set of SD-PDFAs $\mathcal{A}$ with $\Sigma = \{a, b, c\}$. $\mathcal{A}$ contains four PDFAs $\mathcal{M}_\epsilon, \mathcal{M}_a, \mathcal{M}_b, \mathcal{M}_c$. Consider state $q = \langle \epsilon, \epsilon, b, c \rangle \in \bigotimes \mathcal{A}$ (this is the state labeled ϵ, b, c in Figure 6). Then

$$CT(a, q) = T_\epsilon(\epsilon, a) \cdot \ T_a(\epsilon, a) \cdot \ T_b(b, a) \cdot \ T_c(c, a)$$

but

$$PCT(a, q) = T_\epsilon(\epsilon, a) \cdot \ T_b(b, a) \cdot \ T_c(c, a)$$

since in PDFA $\mathcal{M}_a$, the state ϵ is not the maximal state.

The positive co-emission product ($\otimes^+$) is defined just as with co-emission probabilities, substituting PCT and PCF for CT and CF, respectively, in Definition 1. The definition of $\otimes^+$ ensures that the probabilities propagate on the basis of observed subsequences, and not on the basis of unobserved ones.

Lemma 4 *Let $k \geq 1$ and let $\mathcal{A}$ be a k-set of SD-PDFAs. Then $\otimes^+ S$ defines a well-formed probability distribution over Σ^*.*

Proof Since $\mathcal{M}_\epsilon$ belongs to $\mathcal{A}$, it is always the case that PCT and PCF are defined. Well-formedness follows from the normalization term as in Definition 1. ⊣

Definition 5 *A distribution $\mathcal{D}$ is k-Strictly Piecewise (written $\mathcal{D} \in \text{SPD}_k$) $\stackrel{def}{\Longleftrightarrow}$ $\mathcal{D}$ can be described by a PDFA which is the positive co-emission product of a k-set of subsequence-distinguishing PDFAs.*

By Lemma 4, SP distributions are well-formed. Unlike PDFAs for PT distributions, which distinguish $2^{|\Sigma|^{k-1}}$ states, the number of states in a k-set of SD-PDFAs is $\sum_{i<k}(i+1)|\Sigma|^i$, which is $\Theta(|\Sigma|^{k+1})$. Furthermore, since each SD-PDFA only has one state contributing $|\Sigma|+1$ probabilities to the product, and since there are $|\Sigma^{\leq k}| = \frac{|\Sigma|^k - 1}{|\Sigma| - 1}$ many SD-PDFAs in a k-set, there are

$$\frac{|\Sigma|^k - 1}{|\Sigma| - 1} \cdot (|\Sigma| + 1) = \frac{|\Sigma|^{k+1} + |\Sigma|^k - |\Sigma| - 1}{|\Sigma| - 1}$$

parameters, which is $\Theta(|\Sigma|^k)$.

Lemma 5 *Let $\mathcal{D} \in \text{SPD}_k$. Then $\mathcal{D} \in \text{PTD}_k$.*

Proof Since $\mathcal{D} \in \text{SPD}_k$, there is a k-set of subsequence-distinguishing PDFAs. The product of this set has the same structure as the PDFA given in Definition 3. ⊣

Theorem 2 *A distribution $\mathcal{D} \in \text{SPD}_k$ if $\mathcal{D}$ can be described by a PDFA $\mathcal{M} = \langle Q, \Sigma, q_0, \delta, F, T \rangle$ satisfying Definition 3 and the following.*

For all $w \in \Sigma^*$ and all $\sigma \in \Sigma$, let

$$Z(w) = \prod_{s \in P_{\leq k-1}(w)} F(P_{\leq k-1}(s)) + \sum_{\sigma' \in \Sigma} \left(\prod_{s \in P_{\leq k-1}(w)} T(P_{\leq k-1}(s), \sigma') \right) \quad (8)$$

(This is the normalization term.) Then T must satisfy: $T(P_{\leq k-1}(w), \sigma) =$

$$\frac{\prod_{s \in P_{\leq k-1}(w)} T(P_{\leq k-1}(s), \sigma)}{Z(w)} \quad (9)$$

and F must satisfy: $F(P_{\leq k-1}(w)) =$

$$\frac{\prod_{s \in P_{\leq k-1}(w)} F(P_{\leq k-1}(s))}{Z(w)} \quad (10)$$

Proof That SPD_k satisfies Definition 3 Follows directly from Lemma 5. Equations 8-10 follow from the definition of positive co-emission probability. ⊣

The way in which final and transitional probabilities propagate down in SP distributions is reflected in the conditional probability as defined by Equations 9 and 10. In terms of conditional probability, Equations 9 and 10 mean that the probability that σ_i follows a sequence $\sigma_1 \ldots \sigma_{i-1}$ is not only a function of $P_{\leq k-1}(\sigma_1 \ldots \sigma_{i-1})$ (Equation 4) but further that it is a function of each subsequence in $\sigma_1 \ldots \sigma_{i-1}$ up to length $k-1$.

In particular, $Pr(\sigma_i \mid P_{\leq k-1}(\sigma_1 \ldots \sigma_{i-1}))$ is obtained by substituting $Pr(\sigma_i \mid P_{\leq\, k-1}(s))$ for $T(P_{\leq\, k-1}(s), \sigma)$ and $Pr(\# \mid P_{\leq\, k-1}(s))$ for $F(P_{\leq k-1}(s))$ in Equations 8, 9 and 10. For example, for a SP_2 distribution, the probability of a given $P_{\leq 1}(bc)$ (state ϵ, b, c in Figure 6) is the normalized product of the probabilities of a given $P_{\leq 1}(\epsilon)$, a given $P_{\leq 1}(b)$, and a given $P_{\leq 1}(c)$.

To summarize, SP and PT distributions are regular deterministic. Unlike PT distributions, however, SP distributions can be modeled with only $\Theta(|\Sigma|^k)$ parameters and $\Theta(|\Sigma|^{k+1})$ states. This is true even though SP distributions distinguish $2^{|\Sigma|^{k-1}}$ states! Since SP distributions can be represented by a single PDFA, computing $Pr(w)$ occurs in only $\Theta(|w|)$ for such PDFA. While such PDFA might be too large to be practical, $Pr(w)$ can also be computed from the k-set of SD-PDFAs in $\Theta(|w|^k)$ (essentially building the path in the product machine on the fly using Equations 4, 8, 9 and 10).

6 Estimating SP Distributions

The problem of ML estimation of SP_k distributions is reduced to estimating the parameters of the SD-PDFAs. Training (counting and normalization) occurs over each of these machines (i.e. each machine parses the entire corpus), which gives the ML estimates of the parameters of the distribution. It trivially follows that this training successfully estimates any $\mathcal{D} \in \text{SPD}_k$.

Theorem 3 *For any $\mathcal{D} \in \text{SPD}_k$, let $\mathcal{D}$ generate sample S. Let $\mathcal{A}$ be the k-set of SD-PDFAs which describes exactly $\mathcal{D}$. Then optimizing the MLE of S with respect to each $\mathcal{M} \in \mathcal{A}$ guarantees that the distribution described by the positive co-emission product of $\bigotimes^+ \mathcal{A}$ approaches $\mathcal{D}$ as $|S|$ increases.*

Proof The MLE estimate of S with respect to SPD_k returns the parameter values that maximize the likelihood of S. The parameters of $\mathcal{D} \in \text{SPD}_k$ are found on the maximal states of each $\mathcal{M} \in \mathcal{A}$. By definition, each $\mathcal{M} \in \mathcal{A}$ describes a probability distribution over Σ^*, and similarly defines a family of distributions. Therefore finding the MLE of S with respect to SPD_k means finding the MLE estimate of S with respect to each of the family of distributions which each $\mathcal{M} \in \mathcal{A}$ defines, respectively.

Optimizing the ML estimate of S for each $\mathcal{M} \in \mathcal{A}$ means that as $|S|$ increases, the estimates $\hat{T}_{\mathcal{M}}$ and $\hat{F}_{\mathcal{M}}$ approach the true values $T_{\mathcal{M}}$ and $F_{\mathcal{M}}$. It follows that as $|S|$ increases, $\hat{T}_{\bigotimes^+ \mathcal{A}}$ and $\hat{F}_{\bigotimes^+ \mathcal{A}}$ approach the true values of $T_{\bigotimes^+ \mathcal{A}}$ and $F_{\bigotimes^+ \mathcal{A}}$ and consequently $\mathcal{D}_{\bigotimes^+ \mathcal{A}}$ approaches $\mathcal{D}$. ⊣

We demonstrate learning long-distance dependencies by estimating SP_2 distributions given a corpus from Samala (Chumash), a language with sibilant harmony.[4] There are two classes of sibilants in Samala: [-anterior] sibilants like [s] and [t͡s] and [+anterior] sibilants like [ʃ] and [t͡ʃ].[5] Samala words are subject to a phonological process wherein the last sibilant requires earlier sibilants to have the same value for the feature [anterior], no matter how many sounds intervene (Applegate, 1972). As a consequence of this rule, there are generally no words in Samala where [-anterior] sibilants follow [+anterior]. E.g. [ʃtojonowonowaʃ] 'it stood upright' (Applegate 1972:72) is licit but not *[ʃtojonowonowas].

The results of estimating $\mathcal{D} \in \text{SPD}_2$ with the corpus is shown in Table 6. The results clearly demonstrate the effectiveness of the model: the probability of a [α anterior] sibilant given $P_{\leq 1}$([-α anterior]) sounds is orders of magnitude less than given $P_{\leq 1}(\alpha$ anterior]) sounds.

$Pr(x \mid P_{\leq 1}(y))$		x			
		s	t͡s	ʃ	t͡ʃ
y	s	0.0335	0.0051	0.0011	0.0002
	t͡s	0.0218	0.0113	0.0009	0.
	ʃ	0.0009	0.	0.0671	0.0353
	t͡ʃ	0.0006	0.	0.0455	0.0313

Table 1: Results of SP_2 estimation on the Samala corpus. Only sibilants are shown.

7 Conclusion

SP distributions are the stochastic version of SP languages, which model long-distance dependencies. Although SP distributions distinguish $2^{|\Sigma|^{k-1}}$ states, they do so with tractably many parameters and states because of an assumption that distinct subsequences do not interact. As shown, these distributions are efficiently estimable from positive data. As previously mentioned, we anticipate these models to find wide application in NLP.

[4] The corpus was kindly provided by Dr. Richard Applegate and drawn from his 2007 dictionary of Samala.

[5] Samala actually contrasts glottalized, aspirated, and plain variants of these sounds (Applegate, 1972). These laryngeal distinctions are collapsed here for easier exposition.

References

R.B. Applegate. 1972. *Ineseño Chumash Grammar*. Ph.D. thesis, University of California, Berkeley.

R.B. Applegate. 2007. *Samala-English dictionary : a guide to the Samala language of the Ineseño Chumash People*. Santa Ynez Band of Chumash Indians.

Eric Baković. 2000. *Harmony, Dominance and Control*. Ph.D. thesis, Rutgers University.

D. Beauquier and Jean-Eric Pin. 1991. Languages and scanners. *Theoretical Computer Science*, 84:3–21.

Eric Brill. 1995. Transformation-based error-driven learning and natural language processing: A case study in part-of-speech tagging. *Computational Linguistics*, 21(4):543–566.

J. A. Brzozowski and Imre Simon. 1973. Characterizations of locally testable events. *Discrete Mathematics*, 4:243–271.

Noam Chomsky. 1956. Three models for the description of language. *IRE Transactions on Information Theory*. IT-2.

J. S. Coleman and J. Pierrehumbert. 1997. Stochastic phonological grammars and acceptability. In *Computational Phonology*, pages 49–56. Somerset, NJ: Association for Computational Linguistics. Third Meeting of the ACL Special Interest Group in Computational Phonology.

Colin de la Higuera. in press. *Grammatical Inference: Learning Automata and Grammars*. Cambridge University Press.

Pedro García and José Ruiz. 1990. Inference of k-testable languages in the strict sense and applications to syntactic pattern recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 9:920–925.

Pedro García and José Ruiz. 1996. Learning k-piecewise testable languages from positive data. In Laurent Miclet and Colin de la Higuera, editors, *Grammatical Interference: Learning Syntax from Sentences*, volume 1147 of *Lecture Notes in Computer Science*, pages 203–210. Springer.

Pedro Garcia, Enrique Vidal, and José Oncina. 1990. Learning locally testable languages in the strict sense. In *Proceedings of the Workshop on Algorithmic Learning Theory*, pages 325–338.

Gunnar Hansson. 2001. *Theoretical and typological issues in consonant harmony*. Ph.D. thesis, University of California, Berkeley.

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

Jeffrey Heinz. 2007. *The Inductive Learning of Phonotactic Patterns*. Ph.D. thesis, University of California, Los Angeles.

Jeffrey Heinz. to appear. Learning long distance phonotactics. *Linguistic Inquiry*.

John Hopcroft, Rajeev Motwani, and Jeffrey Ullman. 2001. *Introduction to Automata Theory, Languages, and Computation*. Addison-Wesley.

Frederick Jelenik. 1997. *Statistical Methods for Speech Recognition*. MIT Press.

C. Douglas Johnson. 1972. *Formal Aspects of Phonological Description*. The Hague: Mouton.

A. K. Joshi. 1985. Tree-adjoining grammars: How much context sensitivity is required to provide reasonable structural descriptions? In D. Dowty, L. Karttunen, and A. Zwicky, editors, *Natural Language Parsing*, pages 206–250. Cambridge University Press.

Daniel Jurafsky and James Martin. 2008. *Speech and Language Processing: An Introduction to Natural Language Processing, Speech Recognition, and Computational Linguistics*. Prentice-Hall, 2nd edition.

Ronald Kaplan and Martin Kay. 1994. Regular models of phonological rule systems. *Computational Linguistics*, 20(3):331–378.

Gregory Kobele. 2006. *Generating Copies: An Investigation into Structural Identity in Language and Grammar*. Ph.D. thesis, University of California, Los Angeles.

Leonid (Aryeh) Kontorovich, Corinna Cortes, and Mehryar Mohri. 2008. Kernel methods for learning languages. *Theoretical Computer Science*, 405(3):223 – 236. Algorithmic Learning Theory.

M. Lothaire, editor. 1997. *Combinatorics on Words*. Cambridge University Press, Cambridge, UK, New York.

A. A. Markov. 1913. An example of statistical study on the text of 'eugene onegin' illustrating the linking of events to a chain.

Robert McNaughton and Simon Papert. 1971. *Counter-Free Automata*. MIT Press.

A. Newell, S. Langer, and M. Hickey. 1998. The rôle of natural language processing in alternative and augmentative communication. *Natural Language Engineering*, 4(1):1–16.

Dominique Perrin and Jean-Eric Pin. 1986. First-Order logic and Star-Free sets. *Journal of Computer and System Sciences*, 32:393–406.

Catherine Ringen. 1988. *Vowel Harmony: Theoretical Implications*. Garland Publishing, Inc.

James Rogers and Geoffrey Pullum. to appear. Aural pattern recognition experiments and the subregular hierarchy. *Journal of Logic, Language and Information*.

James Rogers, Jeffrey Heinz, Matt Edlefsen, Dylan Leeman, Nathan Myers, Nathaniel Smith, Molly Visscher, and David Wellcome. to appear. On languages piecewise testable in the strict sense. In *Proceedings of the 11th Meeting of the Assocation for Mathematics of Language*.

Sharon Rose and Rachel Walker. 2004. A typology of consonant agreement as correspondence. *Language*, 80(3):475–531.

Jacques Sakarovitch and Imre Simon. 1983. Subwords. In M. Lothaire, editor, *Combinatorics on Words*, volume 17 of *Encyclopedia of Mathematics and Its Applications*, chapter 6, pages 105–134. Addison-Wesley, Reading, Massachusetts.

Stuart Shieber. 1985. Evidence against the context-freeness of natural language. *Linguistics and Philosophy*, 8:333–343.

Imre Simon. 1975. Piecewise testable events. In *Automata Theory and Formal Languages: 2nd Grammatical Inference conference*, pages 214–222, Berlin ; New York. Springer-Verlag.

Howard Straubing. 1994. *Finite Automata, Formal Logic and Circuit Complexity*. Birkhäuser.

Wolfgang Thomas. 1982. Classifying regular events in symbolic logic. *Journal of Computer and Systems Sciences*, 25:360–376.

Enrique Vidal, Franck Thollard, Colin de la Higuera, Francisco Casacuberta, and Rafael C. Carrasco. 2005a. Probabilistic finite-state machines-part I. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(7):1013–1025.

Enrique Vidal, Frank Thollard, Colin de la Higuera, Francisco Casacuberta, and Rafael C. Carrasco. 2005b. Probabilistic finite-state machines-part II. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(7):1026–1039.