Proceedings of Machine Learning Research 217:80–95, 2023 ICGI regular paper

String Extension Learning Despite Noisy Intrusions

Katherine Wu KATHERINE.WU@STONYBROOK.EDU
Department of Computer Science
Stony Brook University

Jeffrey Heinz JEFFREY.HEINZ@STONYBROOK.EDU
Department of Linguistics
Institute for Advanced Computational Science
Stony Brook University

Editors: François Coste, Faissal Ouardi and Guillaume Rabusseau

Abstract

We examine the conditions in which string extension learning algorithms are able to identify classes of formal languages in the limit from noisy data presentations in polynomial time. A data presentation for a formal language L is noisy if it contains words belonging to the complement of L. In the general case, string extensions learners cannot distinguish noise from true examples and are led astray. The main result is that relative frequencies can be used to distinguish noisy examples from true examples provided the data presentations are constrained to those in which relative frequencies are uniformly present and exceed the rate at which noise is introduced.

Keywords: identification in the limit, noisy data, intrusions, string extension learning

1. Introduction

The identification in the limit from positive data paradigm (Gold, 1967) examines the asymptotic behavior of learning algorithms on data streams which only contain positive examples of the target language. This choice allowed researchers to ask whether successful generalization was even possible in the absence of noise. Later research has shown concretely how this kind of learning problem becomes more difficult when inaccurate information is added to the data streams (Osherson et al., 1986; Fulk and Jain, 1996; Jain, 1996; Stephan, 1997; Jain et al., 1999; Tantini et al., 2006). This is also the case in other learning frameworks (Angluin and Laird, 1988; Vapnik, 1998).

Jain et al. (1999, chap. 8) describe three ways in which a data presentation for a formal language L can be inaccurate.

1. Noisy data. A data presentation for L includes intrusions from the complement of L.
2. Incomplete data. A data presentation for L omits examples from L.
3. Imperfect data. Data presentations for L includes intrusions from the complement of L in addition to omitting examples from L.

This paper studies the identification in the limit from noisy data and imperfect data.

We also narrow the focus to string extension learning algorithms (Heinz, 2010). Classes of languages identifiable by string extension learning algorithms are structured as a lattice and learners "climb" the lattice as positive examples are "extended" to reveal the elements of the underlying grammars (Heinz et al., 2012).

String extension learnable classes are of interest for several reasons. First, several other classes also have this underlying lattice structure including the reversible languages (Angluin, 1982), Locally k-Testable languages (Rogers and Pullum, 2011), Piecewise k-testable languages (Simon, 1975), function-distinguishable languages (Fernau, 2003), and closed-set systems (de Brecht et al., 2006). Heinz et al. (2012) also show that these classes have learners with several desirable properties: they are incremental, consistent, conservative, set-driven, and strongly monotonic. Finally, these classes are important for understanding patterns in natural languages (Lambert et al., 2021).

When noise is introduced into the data presentation of a language belonging to a string extension learnable class, the aforementioned learning algorithms will overgeneralize. The central question this paper addresses is how string extension learning algorithms can be modified to identify noise in a noisy data presentation so that it can be ignored and learners can generalize successfully.

Our contributions are as follows. First, we define classes of data presentations corrupted by noise according to whether the underlying data presentation contains each positive example at least once (plain text) or infinitely many times (fat text) and according to whether the noise replaces words in these texts (replacement noise) or is added into them (insertion noise) with some probability. The first result is negative: without further restrictions identification in the limit is not possible.

Following Osherson et al. (1986, Chapter 5), we ask whether for a given string extension learnable class, there is an algorithm which can identify it in the limit on a suitably constrained class of noisy data presentations. We show that modifying string extension learners to keep track of the relative frequencies of the words in the target languages which support elements of the grammar can be used to distinguish noisy examples from true ones provided the rate of noise is known. In particular, we present two algorithms, SENTIA1 and SENTIA2, which identify noise in similar but distinct ways. In the conclusion, we discuss how the noise rate could be estimated in standard machine learning scenarios, provided it is unknown.

The present work makes a different contribution to understanding identification in the limit in the presence of noise than prior work. The work of Tantini et al. (2006) is perhaps the most closely related. However, they only consider noisy words which are within k edit distance of some word in the target language L, where k is bounded above by some constant, whereas we consider noise to be any word in the complement of L. On the other hand, the algorithms SENTIA1 and SENTIA2 can be thought of as particular algorithms which "denoise on the fly" in their sense. Whereas Stephan (1997) investigates data streams where each correct element appears infinitely often and each incorrect element appears at most finitely often, this paper concerns itself with data streams where each correct and incorrect element both appear infinitely often. Additionally, the works of Fulk and Jain (1996) and Jain (1996) established general hierarchies of classes that can be identified in the limit with noise of a particular density, but cannot be identified when the noise density increases. In this work, we show what a specific strategy of learning (the use of relative frequencies) can offer a specific class of algorithms (string extension learning algorithms) when learning from noisy examples.

2. Preliminaries

This section introduces the notational conventions, identification in the limit, string extension learning, and other key concepts.

2.1. Notation

Multisets which are sets that allow duplicates. Let $\uplus$ denote the union operation permitting duplicates, which produces a multiset of elements. For example, given a set $A = \{a, b, c\}$, $A \uplus A = \{a, a, b, b, c, c\}$. For the sake of brevity, instead of listing out each element in a multiset S, we will henceforth express them as a set of $(x, m(x))$ pairs, where $m(x)$ is the multiplicity of x in S. Thus, in the example above, $A \uplus A = \{(a, 2), (b, 2), (c, 2)\}$. If A and B are both multisets, then $A - B$ is the multiset difference $\{(x, y) : x \in A, y = \max(0, m_A(x) - m_B(x))\}$. For example, if $A = \{(a, 3), (b, 3), (c, 1)\}$ and $B = \{(a, 2), (b, 1), (c, 2)\}$, then $A - B = \{(a, 1), (b, 2)\}$. Note that multiset differences will not produce negative multiplicities. The size of a multiset S is denoted $|S|$ and is equal to $\sum_{(x,y) \in S} y$.

To reduce a multiset to a set of distinct elements, let $\mathcal{R}$ be the function that maps each element (x, y) in a multiset to x. For example, $\mathcal{R}(\{(a, 3), (b, 4), (c, 1)\}) = \{a, b, c\}$.

For a set S, let $\mathcal{P}(S)$ denote the powerset of S and let $\mathcal{P}_{fin}(S)$ denote the set of all subsets of S with finite cardinality.

The alphabet Σ denotes a finite set of symbols. A string over Σ is a finite sequence of symbols from Σ concatenated together. The length of a string w is denoted as $|w|$, and λ denotes the empty string. Thus $|\lambda| = 0$. A string u is said to be a substring of another string w if there exists two other strings v and v' such that $w = vuv'$. Σ^* represents all strings over the alphabet Σ. A language L is any subset of Σ^*. $\overline{L}$ denotes the complement of L, namely the set of all strings not in L. A grammar, denoted as G, is a finite representation of a language and $L(G)$ denotes the language that G represents. We denote the class of all computably enumerable languages with $\mathfrak{L}$. We often use $\mathcal{L}$ to denote some subset of $\mathfrak{L}$.

Finally, we use the symbols $\rtimes$ and $\ltimes$ to mark the beginning and end of strings. $\rtimes$ denotes a left word boundary; $\ltimes$ denotes a right word boundary.

2.2. Learning Paradigm

Gold (1967) introduced identification in the limit. A text t is an infinite sequence of strings drawn from Σ^*. Let $t(i)$ denote the ith string of t and $t[i]$ the finite sequence of strings up to and including $t(i)$. In other words, $t[i] = t(1)$, $t(2)$, ... $t(i)$. Let $\textsc{content}(t)$ equal the set of words in text t: $\{t(i) : i \in \mathbb{N}\}$. A text is *fat* iff for all $w \in \textsc{content}(t)$, the cardinality of the set $\{n : t(n) = w\}$ is infinite.

Denote the set of all texts with $\mathfrak{T}$. Following Osherson et al. (1986, chapter 5), an *evidential relation* is a subset of $\mathfrak{T} \times \mathfrak{L}$. As they explain, “An evidential relation is a means of specifying the environments that count as ‘for’ a given language.” If $\mathcal{E}$ is an evidential relation and $L \in \mathfrak{L}$ then $\{t : (t, L) \in \mathcal{E}\}$ is the set of data presentations for L, relative to $\mathcal{E}$, that we expect learning algorithms to succeed on.

A learning algorithm, denoted ϕ, maps initial finite portions of text $t[i]$ to grammars. A learner ϕ converges to some G on a text t if and only if there exists some $i \in \mathbb{N}$ such that for all $j > i$ it is the case that $\phi(t[j]) = \phi(t[i]) = G$. Given an evidential relation $\mathcal{E}$ and a language L, ϕ *identifies* L *on* $\mathcal{E}$ if and only if for all $t \in \{t : (t, L) \in \mathcal{E}\}$, ϕ converges to G on t and $L(G) = L$. We say ϕ identifies $\mathcal{L}$ on $\mathcal{E}$ (and that $\mathcal{L}$ is identifiable on $\mathcal{E}$) if and only if for all $L \in \mathcal{L}$, ϕ identifies L on $\mathcal{E}$.

As an example, the evidential relation $\mathcal{E}_0 = \{(t, L) : \text{CONTENT}(t) = L\}$ is one of the first studied by Gold (1967). Specifically, his theorem that the finite languages are identifiable in the limit from positive data is translated to "the finite languages are identifiable on $\mathcal{E}_0$." Similarly his theorem that no superfinite class is identifiable can be expressed as "no superfinite class is identifiable on $\mathcal{E}_0$.[1]

2.3. String Extension Learning

Heinz (2010) describes general algorithms for classes of languages identifiable in the limit from positive data (i.e. they are identifiable on $\mathcal{E}_0$). Consider any alphabet Σ and any set A. Elements of A will be called factors. Consider functions of the form $f : \Sigma^* \to \mathcal{P}_{fin}(A)$. Let a grammar G be a nonempty finite subset of A. Elements in G are called permissible factors, and elements in $A - G$ are called forbidden factors. Let $L(G)$ be defined as $\{w : f(w) \subseteq G\}$. Thus, a word belongs to $L(G)$ if and only if all of its factors are permissible. The class of languages $\mathcal{L}_f$ is defined as $\{L(G) : G \subseteq A, |G| \text{ finite}\}$.[2] Heinz (2010) shows that these classes are necessarily closed under intersection, but not necessarily under union or complement.

Each $\mathcal{L}_f$ admits a learning algorithm which identifies it in the limit from positive data. This algorithm maps each string in the text $t[i]$ to the set of factors via a string extension function f. Since $t[i]$ is a positive text for L these factors must be elements of the grammar G for L. Consequently, a learner φ_f which begins hypothesizing an empty grammar and updates this hypothesis at each point i in the text t by adding all factors in the current string $t(i)$ will eventually see all the factors of G. This is essentially because G is finite and t is a positive text for L. Formally, a string extension learner is defined as follows:

$$\varphi_f(t[i]) = \begin{cases} \emptyset & \text{if } i = 0 \\ \varphi_f(t[i-1]) & \text{if } t(i) = \# \\ \varphi_f(t[i-1]) \cup f(t(i)) & \text{otherwise} \end{cases} \tag{1}$$

Heinz (2010) observes that many well-studied subclasses of the regular languages are string extension learnable including the Strictly k-Local, Locally k-Testable, and Strictly k-Piecewise classes in addition to nonregular classes and classes containing infinitely many languages. For additional details, see Heinz (2010) and Heinz et al. (2012).

It is easy to see that φ_f will fail to identify a target language L on a text that contains noise. If the text contains noise then it contains at least one word w from $\overline{L}$. Since $w \notin L$, $f(w)$ includes a forbidden factor x and therefore φ_f will add x to its hypothesized grammar. Since φ_f never removes factors, it will fail to converge to a grammar for L.

1. Our formulation of texts means there is no text for the empty language. Following Gold, many researchers allow texts to include another symbol $\#$, which represents a moment of no information. Doing so allows the empty language exactly one text: $\{\#, \#, \ldots\}$. We omit this because it simplifies the presentation and analysis of our algorithms. The price we pay is we must exclude the empty language from our analysis.
2. The original formulation permits grammars to be empty sets. However. the languages of such grammars is empty and we are excluding those from our analysis.

2.4. Support and Relative Frequencies

It will be useful to relate a factor $x \in A$ to the words in the text which f maps to x.

Definition 1 (Support) *Let f be any string extension function and consider any text t and any $i \in \mathbb{N}$. Then* $\textsc{support}(x, t[i])$ *is the multiset* $\{w \in t[i] : x \in f(w)\}$.

In other words, $\textsc{support}(x, t[i])$ is the multiset of strings in $t[i]$ that 'contain' the factor x. The proportion of words in a text that support particular factors are important.

Definition 2 (Relative Frequency) *Let f be any string extension function and consider any text t and any $i \in \mathbb{N}$. The relative frequency for x in $t[i]$, denoted* $\textsc{rel}(x, t[i])$, *equals* $|\textsc{support}(x, t[i])|/i$.

Finally, we observe that replacing the union operator ($\cup$) in Equation 1 with multiset union ($\uplus$) produces an algorithm which accumulates a multiset of factors from a text.

Definition 3 (Multiset String Extension Learning) *For all string extension functions f define ϕ_f as follows:*

$$\phi_f(t[i]) = \begin{cases} \emptyset & \textit{if } i = 0 \\ \phi_f(t[i-1]) & \textit{if } t(i) = \# \\ \phi_f(t[i-1]) \uplus f(t(i)) & \textit{otherwise} \end{cases}$$

In multiset string extension learning, $f(t(i))$ still returns a set; hence, each factor in $t(i)$ has a multiplicity of one in $f(t(i))$ no matter how many times it occurs in $t(i)$. An example is given later in Table 2.

3. SENTIA1: Denoising Algorithm

If noise is relatively rare, then relative frequencies can be used to distinguish true positive examples from noisy examples. The first String Extension Noise-Tolerant Inference Algorithm (SENTIA1) uses relative frequencies at each point i in the text to determine which factors are forbidden and which are permissible. It assumes that noise is introduced into the data presentation at a fixed rate η which is between 0 and 1 inclusive. SENTIA1 takes as inputs this noise rate η, an initial portion of text $t[i]$, and a current multiset of factors $\phi_f(t[i])$, and it outputs a hypothesized grammar G. At each i, the algorithm calculates $\textsc{rel}(x, t[i])$ for each factor $x \in \mathcal{R}(\phi_f(t[i]))$, only including x in G if $\textsc{rel}(x, t[i]) > \eta$. The helper function named `counts`, taking in a multiset of factors, maps each unique factor in $\phi_f(t[i])$ to its multiplicity in $\phi_f(t[i])$. SENTIA1 is presented in Algorithm 1.

The idea is that SENTIA1 uses η as a dividing line in $\phi_f(t[i])$ to separate the permissible factors from the forbidden ones. While SENTIA1 may err in the beginning especially when only some strings have been presented in t, the intuition is that the relative frequencies will eventually stabilize and remain fixed after a certain point in t, and the separation of factors will become more apparent. In fact the following lemma is straightforward to show.

Lemma 4 *Given a factor $x \in A$ and any initial text segment $t[i]$, SENTIA1 adds x to its hypothesized grammar G if and only if* $\textsc{rel}(x, t[i]) > \eta$.

```
**Data:** $\eta$, $\phi_f(t[i])$, $i$
**Result:** A grammar $G$
function counts($S$):
    Initialize $M = \{\}$
    foreach $factor$ in $S$ do
        if $factor$ not in $M$ then
            $M$.add[$factor$]
        end
        $M[factor]$ += 1
    end
return $M$
function main($\eta$, $\phi_f(t[i])$, $i$):
    Initialize $G = \{\}$
    $M$ = counts($\phi_f(t[i])$)
    foreach $factor$ in $M$ do
        $rel = M[factor]/i$
        if $rel > \eta$ then
            $G$.add($factor$)
        end
    end
return $G$
```

Algorithm 1: String Extension Noise-Tolerant Inference Algorithm (SENTIA1)

Proof Follows directly from the algorithm. If x is a factor with $\text{REL}(x, t[i]) \leq \eta$, the statement $G.\text{add}(factor)$ is never executed for the factor x, so x is never added to G. On the contrary, if x is a factor with $\text{REL}(x, t[i]) > \eta$, $G.\text{add}(factor)$ is executed for the factor x, so x is added to G. ■

Here are remarks regarding whether SENTIA1 over- or under- generalizes.

- If the relative frequencies of all the permissible factors of a language are greater than η and the relative frequencies of all the forbidden factors are less than η, then SENTIA1 hypothesizes the correct grammar G for L.
- If the relative frequencies of some permissible factors and of all forbidden factors are below η, the learner hypothesizes a proper subset of the target G.
- If the relative frequencies of some forbidden factors and of all permissible factors are above η, the learner hypothesizes a superset of the target G, which includes some forbidden factors.
- If some forbidden factors of a language rise above the line and some permissible factors sink below the line, the learner hypothesizes a grammar incomparable with G.
- If all forbidden factors of a language rise above the line and all permissible factors sink below the line, the learner hypothesizes $\overline{G}$, which generates the language $\overline{L}$.

Lemma 5 *If the string extension function $f : \Sigma^* \rightarrow A$ runs in polynomial time in the size of $|w|$ where w is a string, then given an input sample I of size n, SENTIA1 outputs a hypothesized grammar G in time polynomial in the size of n.*

Proof Before the algorithm begins, the multiset of factors $\phi_f(I)$ of the current portion of text I must be obtained. We are given that f runs in polynomial time on inputs of size $|w|$, so formally, the time complexity of f is $O(|w|^k)$ for some $k > 0$. Since f must be performed on each string in I, the time complexity for this is $O(n \cdot |w|^k)$. Then during the algorithm, the helper function `counts` is called, making one pass through $\phi_f(I)$, this time creating a map between each unique factor in $\phi_f(I)$ and its multiplicity in the multiset. The time complexity for `counts` is $O(n \cdot |w|^k)$. The algorithm then iterates through each mapping that was constructed from `counts`, adding to G any factors with relative frequencies greater than η. In the worst case, this takes time $O(n \cdot |w|^k)$. Putting it altogether, an upper bound on the time complexity of SENTIA1 is given by $O(3n \cdot |w|^k)$, which is polynomial. ■

We are interested in the general behavior of SENTIA1. What computational learning problems does it solve? One way to answer this question is to identify an evidential relation $\mathcal{E}$ and show that the algorithm identifies any string extension class $\mathcal{L}_f$ on $\mathcal{E}$.

4. Text with Noise

A positive text for a language can be corrupted by noise in several ways. In this paper, we discuss two methods which introduce noise into the text. These two methods combined with plain and fat texts yield four different noisy learning frameworks.

A noisy text for a positive language L includes intrusions from $\overline{L}$. We construct a noisy text d for L by weaving together a positive text t for L and a positive text t' for $\overline{L}$. The basic idea is that for each $i \in \mathbb{N}$, the teacher flips a coin with heads occurring with probability η and tails occurring with probability $1 - \eta$. If the coin lands on heads, $d(i)$ comes from t and if the coin lands on tails, $d(i)$ comes from t'.

There are at least two ways to do this. One way is via replacements. In this case, when the coin lands on heads, $d(i)$ is set to $t(i)$. When it lands on tails, $d(i)$ is set to $t'(i)$. Essentially, for each i, $t(i)$ is replaced with a string from $\overline{L}$ with probability η. Replacement noise may lead to incomplete data because some strings may be omitted in the text: it is possible that if a string $w \in L$ is replaced, it may never occur in the text again.

Another way is via insertions. Set a counter $j = 1$. If the coin lands on heads, $d(i)$ is set to $t(j)$ and j is incremented by one. If the coin lands on tails, $d(i)$ is set to $t'(i)$. Unlike replacement noise, insertion noise does not lead to any omissions of correct strings. Instead, the idea is that with probability η, the teacher simply inserts noisy strings into the text without changing the underlying positive text. Table 1 illustrates these two processes of generating noisy texts.

In terms of evidential relations, we can define these two as follows. Let $\textsc{replace}(t, t', \eta)$ and $\textsc{insert}(t, t', \eta)$ denote processes that generate new texts by weaving together texts t and t' with noise parameter η using the replacement and insertion procedures described above, respectively. Then we can define four evidential relations as follows.

- **Plain text with replacement noise**: Each string in L is guaranteed to appear at least once in the text, but may be replaced by a noisy string with probability η.

$$\{(d, L) : d \in \textsc{replace}(t, t', \eta), \textsc{content}(t) = L, \textsc{content}(t') = \overline{L}\}$$

Text t for L	Text t' for $\overline{L}$	d_I	d_R	coin flip (η)
$t(1)$	$t'(1)$	$t(1)$	$t(1)$	heads
$t(2)$	$t'(2)$	$t(2)$	$t(2)$	heads
$t(3)$	$t'(3)$	$t'(3)$	$t'(1)$	tails
$t(4)$	$t'(4)$	$t(4)$	$t(3)$	heads
$t(5)$	$t'(5)$	$t(5)$	$t(4)$	heads
$t(6)$	$t'(6)$	$t'(6)$	$t'(2)$	tails
...	...	...	...	...

Table 1: An illustration of noisy text construction. Text d_I is the text constructed with insertion noise and text d_R is the text constructed with replacement noise.

- **Plain text with insertion noise**: Each string in L is guaranteed to appear at least once in the text, but intrusions occur with probability η.

$$\{(d, L) : d \in \textsc{insert}(t, t', \eta), \textsc{content}(t) = L, \textsc{content}(t') = \overline{L}\}$$

- **Fat text with replacement noise**: Each string in L is guaranteed to appear infinitely many times in the text, but may be replaced by a noisy string with probability η.

$$\{(d, L) : d \in \textsc{replace}(t, t', \eta), \textsc{content}(t) = L, \textsc{content}(t') = \overline{L}, t \text{ is fat}\}$$

- **Fat text with insertion noise**: Each string in L is guaranteed to appear infinitely many times in the text, but intrusions in the text occur with probability η.

$$\{(d, L) : d \in \textsc{insert}(t, t', \eta), \textsc{content}(t) = L, \textsc{content}(t') = \overline{L}, t \text{ is fat}\}$$

Interestingly, given any $\eta > 0$, SENTIA1 cannot identify any string extension class $\mathcal{L}_f$ on these evidential relations.

Theorem 6 *SENTIA1 fails to identify any $\mathcal{L}_f$ which contains at least one language with at least two strings on plain or fat text with replacement or insertion noise.*

Proof Consider first plain text with replacement or insertion noise and consider any $\mathcal{L}_f$. Consider $L \in \mathcal{L}_f$ with a cardinality of at least 2. Let a, b be distinct strings in L such there is at least one factor x in $f(a)$ that is not in $f(b)$. Consider the text $t = a, b, b, b, b, \ldots$. The target grammar contains $f(a) \cup f(b)$. Consequently, as i increases, $\textsc{rel}(x, t[i])$ decreases. So there is some j such that for all $i > j$, $\textsc{rel}(x, t[i]) < \eta$. By Lemma 4, SENTIA1 will not add x to G and it will fail to learn an exact grammar for L.

Consider next fat text with replacement or insertion noise. Since each string w in L appears infinitely many times, each factor $x \in G$ also appears infinitely many times. Consequently the probability of every occurrence of a word supporting x being replaced is effectively zero. However, consider a text t_n of the form which repeats n b's, followed by one a and this sequence repeats indefinitely. Clearly this is a fat text for L. However, for sufficiently large n, there exists a j such that for all $i > j$, $\textsc{rel}(x, t[i]) < \eta$. In other words SENTIA1 cannot distinguish the true string a (and by extension the factor x) from noise in text t_n. By Lemma 4, SENTIA1 will not add x to G and it will fail to learn an exact grammar for L. ■

5. η-Distinguishable Texts

There are, however, infinitely many texts with replacement or insertion noise which SENTIA1 succeeds on. This section characterizes those texts and proves that SENTIA1 identifes any $\mathcal{L}_f$ satisfying that evidential relation.

Recall that a family of sets P is a partition of X if and only if the family P does not contain the empty set, the union of sets in P is equal to X, and the intersection of any two distinct sets in P is empty (i.e. the two sets are pairwise disjoint).

Definition 7 (η-distinguishablility) *Given η, a noisy text t is called η-distinguishable for $L \in \mathcal{L}_f$ if and only if there exists some $j \in \mathbf{N}$ such that for all $i > j$, for all $x \in G$ and $y \notin G$, $\textsc{rel}(x, t[i]) > \eta \geq \textsc{rel}(y, t[i])$ where $L(G) = L$.*

In other words, after a specific point j in the text, for all $i > j$, there is a partition on $\phi_f(t[i])$ into two smaller multi-sets P_1 and P_2, with the stipulation that the $\textsc{rel}$ of any unique element in P_1 is strictly greater than the $\textsc{rel}$ of any unique element in P_2. Furthermore, $\mathcal{R}(P_1) = G$ and $L(G) = L$ and $\mathcal{R}(P_2) = A - G$.

This gives us another evidential relation with noise.

η-distinguishable noisy texts

$$\{(d, L) : d \text{ is } \eta\text{-distinguishable for } L\}$$

Theorem 8 *SENTIA1 identifies any string extension learning class $\mathcal{L}_f$ on η-distinguishable texts in polynomial time.*

Proof Consider any $L \in \mathcal{L}_f$ and any η-distinguishable text t for L. Then there exists j such that for $i > j$ it is the case that for all $x \in G$ and for all $y \notin G$, it holds that $\textsc{rel}(x, t[i]) > \eta \geq \textsc{rel}(y, t[i])$. Consequently SENTIA1 will add every $x \in G$ to its hypothesized grammar by Lemma 4. Polynomial time follows from Lemma 5. ■

Table 2 illustrates an η-distinguishable text at $i = 8$ for a Strictly 2-Local language (2-SL). The SL languages can be defined as follows (Heinz et al., 2011):

Definition 9 (SL languages) *Let $fac_k : \Sigma^* \to \mathcal{P}_{fin}(\Sigma^*)$ be a function mapping strings to their k-factors, where a k-factor of a string w is a substring v of w such that $|v| = k$. A language L is Strictly k-Local (SL_k) if and only if there exists a finite set $S \subseteq fac_k(\rtimes\Sigma^*\ltimes)$ such that $L = \{w \in \Sigma^* | fac_k(\rtimes w \ltimes) \subseteq S\}$.*

In Table 2, the target grammar is $G = \{\rtimes a, aa, ab, a\ltimes, b\ltimes\}$, which generates the target 2-SL language $L = \{aa, ab, aaa, aab, ...\}$. Suppose $\eta = 30\%$. We treat word boundaries the same as we treat factors. Computing $\textsc{rel}(x, t[8])$ for each unique $x \in \phi_f(t[8])$, we obtain the values 6/8, 2/8, 5/8, 4/8, 1/8, 2/8, 3/8, 5/8 for the factors $\rtimes a$, $\rtimes b$, aa, ab, ba, bb, $a\ltimes$, $b\ltimes$, respectively. Such a partition exists: $\{\rtimes a, aa, ab, a\ltimes, b\ltimes\} \in \mathcal{R}(P_1)$ and $\{\rtimes b, ba, bb\} \in \mathcal{R}(P_2)$.

i	$t(i)$	$fac_2(t(i))$	$\phi_f(t[i])$
0		$\emptyset$	$\emptyset$
1	aaa	$\{\rtimes a, aa, a\ltimes\}$	$\{(\rtimes a,1),(aa,1),(a\ltimes,1)\}$
2	$aaaab$	$\{\rtimes a, aa, ab, b\ltimes\}$	$\{(\rtimes a,2),(aa,2),(ab,1),(a\ltimes,1),(b\ltimes,1)\}$
3	**bb**	$\{\rtimes b, bb, b\ltimes\}$	$\{(\rtimes a,2),(\rtimes b,1),(aa,2),(ab,1),(bb,1),(a\ltimes,1),(b\ltimes,2)\}$
4	aa	$\{\rtimes a, aa, a\ltimes\}$	$\{(\rtimes a,3),(\rtimes b,1),(aa,3),(ab,1),(bb,1),(a\ltimes,2),(b\ltimes,2)\}$
5	ab	$\{\rtimes a, ab, b\ltimes\}$	$\{(\rtimes a,4),(\rtimes b,1),(aa,3),(ab,2),(bb,1),(a\ltimes,2),(b\ltimes,3)\}$
6	aab	$\{\rtimes a, aa, ab, b\ltimes\}$	$\{(\rtimes a,5),(\rtimes b,1),(aa,4),(ab,3),(bb,1),(a\ltimes,2),(b\ltimes,4)\}$
7	**baa**	$\{\rtimes b, ba, aa, a\ltimes\}$	$\{(\rtimes a,5),(\rtimes b,2),(aa,5),(ab,3),(ba,1),(bb,1),(a\ltimes,3),(b\ltimes,4)\}$
8	**abb**	$\{\rtimes a, ab, bb, b\ltimes\}$	$\{(\rtimes a,6),(\rtimes b,2),(aa,5),(ab,4),(ba,1),(bb,2),(a\ltimes,3),(b\ltimes,5)\}$

Table 2: An η-distinguishable text for the language in Example 1. Noisy strings are in bold.

Noisy texts which are η-distinguishable effectively guarantee that all forbidden factors appear sufficiently rare enough so that SENTIA1 can distinguish them from the permissible factors and drop them accordingly. Note however, that while η-distinguishability requires that each forbidden factor eventually occur with $\textsc{rel} \leq \eta$, it gives no restriction on the summation of these relative frequencies. As such, the learner may hypothesize the correct grammar on texts with more than $\eta \cdot i$ noisy strings at a point i, as long as the $\textsc{rel}$ of each forbidden factor is less than or equal to η and each permissible factor greater than η. This is most likely to occur towards the beginning of texts when not many strings have been presented yet. However, in the limit, it is guaranteed that the number of noisy strings in the text will converge to its expected value of $\eta \cdot i$ at each point i, and the difference between the current observed number of noisy strings in the text and the expected number of noisy strings $\eta \cdot i$ in the text will converge to zero.

With that being considered, we may want such a denoising algorithm that simulates more of how the noisy text was constructed in the first place, which is that with probability η, a noisy string was either inserted or a correct string replaced by a noisy string. At each point i in the text, the learner should expect that $\eta \cdot i$ strings are noisy. Instead of learning from individual factors which is what SENTIA1 currently does, the alternative would be to learn from strings, dropping the expected number of noisy strings from the text and extracting information from the remaining strings to obtain a hypothesis at each i.

The next section formalizes this idea with an alternative denoising algorithm we call the second String Extension Noise-Tolerant Inference Algorithm (SENTIA2), and how this notion of learning gives rise to another set of noisy texts that satisfy a different evidential relation, for which learning can succeed.

6. SENTIA2: An Alternative Denoising Algorithm

The second algorithm also takes as input a noise rate η, the current portion of text $t[i]$, the current multiset of factors $\phi_f(t[i])$ obtained from $t[i]$, and the current length of the text i. It finds the factor in the current text with the smallest REL and concludes it is forbidden, then deletes all strings in $t[i]$ that contain this factor. It repeats this process, stopping just before more than $\eta \cdot i$ strings have been deleted. SENTIA2 is presented in Algorithm 2.

Two helper functions are used, `get_min` and `delete_strings`. We omit the implementation of `get_min`, since it is self-explanatory: it takes as input a multiset S and returns the element in S with the least multiplicity or count, along with the multiplicity itself. `delete_strings` modifies both $\phi_f(t[i])$ and $t[i]$, deleting all strings in SUPPORT$(x, t[i])$ from $t[i]$ and all factors in SUPPORT$(x, t[i])$ from $\phi_f(t[i])$, where x is the factor with the current minimum REL.

```
Data: $\eta$, $t[i]$, $\phi_f(t[i])$, $i$
Result: A grammar $G$
function delete_strings($factor$, $t$, $S$):
    $S \leftarrow S - \phi_f(\textsc{support}(factor, t))$
    $t \leftarrow t - \textsc{support}(factor, t)$
return $S, t$
function main($\eta$, $t[i]$, $\phi_f(t[i])$, $i$):
    Initialize $G = \{\}$
    Initialize $\eta' = 0$
    Let $S \leftarrow \phi_f(t[i])$, $t' \leftarrow t[i]$
    while $true$ do
        $min\_fac, count \leftarrow$ get_min($S$)
        $\eta'$ += $count/i$
        if $\eta' > \eta$ then
            break
        end
        $S, t' \leftarrow$ delete_strings($min\_fac, t', S$)
    end
    $G \leftarrow \mathcal{R}(S)$ //get only the unique factors
return $G$
```

Algorithm 2: String Extension Noise-Tolerant Inference Algorithm 2 (SENTIA2)

The purpose of SENTIA2 is to transform noisy texts to noise-free ones by removing approximately $\eta \cdot i$ strings which contain the factors with the lowest relative frequencies, but never removing more than $\eta \cdot i$ strings from the text. Nonetheless, the following negative result holds.

Theorem 10 *SENTIA2 fails to identify any string extension learnable class $\mathcal{L}_f$ which contains at least one language with at least two strings on plain or fat text with replacement or insertion noise.*

Proof Without loss of generality, consider a plain text with replacement or insertion noise as well as any $\mathcal{L}_f$. Even though the number of noisy strings converges to $\eta \cdot i$ in the limit, at some point i in t, there could be exactly $\eta \cdot i$ noisy strings, but at the next point j in the text, $t(j)$ could be noisy. There are now more than $\eta \cdot j$ noisy strings in the text, and because SENTIA2 never overestimates the number of noisy strings in the text, SENTIA2 fails to converge to the correct target grammar G. ■

7. η-Distinguishable-Up-to-Expectation Texts

There are infinitely many texts with replacement or insertion noise that SENTIA2 succeeds on. This section characterizes those texts, called η-distinguishable-up-to-expectation texts, and proves that SENTIA2 identifies any $\mathcal{L}_f$ satisfying that evidential relation.

To properly characterize this evidential relation, we first want to extract an important kind of factor from each string in the current text. We will want to convert the current text $t[i]$ into a new text $t'[i]$ containing just these factors. This factor is called the minimum relative factor, which we define formally below.

Definition 11 (Minimum Relative Factor) *Given a string s, a string extension function f, and a text $t[i]$, the minimum relative factor of s, denoted* $\textsc{min\_rel}(s)$*, is the factor with relative frequency equal to* $\min_{x \in f(s)}\{\textsc{rel}(x, t[i])\}$.

In the case where $f(s)$ is a singleton set, the only factor in that set is trivially the minimum relative factor. In the case where there are two or more potential minimum relative factors for a string, we choose any one. We are now ready to define the evidential relation that SENTIA2 succeeds on.

Definition 12 (η-distinguishability-up-to-expectation) *Given η, a noisy text t is called η-distinguishable-up-to-expectation for $L \in \mathcal{L}_f$ if and only if there exists some j such that for all $i > j$, it holds that for all $x \in G$ and for all $y \notin G$,* $\sum_{y \notin G} \textsc{rel}(y, t'[i]) < \eta < \sum_{x \in G} \textsc{rel}(x, t'[i])$ *where $t'[i]$ is the sequence of factors obtained from $t[i]$ by applying* $\textsc{min\_rel}(s)$ *for each s in $t[i]$.*

This gives us the following evidential relation with noise.

η-distinguishable-up-to-expectation noisy texts

$$\{(d, L) : d \text{ is } \eta\text{-distinguishable-up-to-expectation for } L\}$$

For classes that deal with word boundaries such as the SL languages, we treat the word boundaries the same as we treat factors. Using the same language in Example 1 and the same text in Table 2, Table 3 illustrates the notion of minimum relative factors (in bold), and how it is used to construct a new text $t'[i]$ which is then used to identify as many as $\eta \cdot i$ strings as noise. This text is η-distinguishable-up-to-expectation for $\eta = 40\%$ because the $\textsc{rel}$ value with respect to $t'[i]$ of each factor in $t'[i]$ is 0, 0, 0, 3/8, 1/8, 2/8, 2/8, 0 for for the factors $\rtimes a$, $\rtimes b$, aa, ab, ba, bb, $a\ltimes$, $b\ltimes$, respectively; $\sum_{y \notin G} \textsc{rel}(y, t'[i]) = 3/8$ and $\sum_{x \notin G} \textsc{rel}(x, t'[i]) = 5/8$, and $3/8 < 0.4 < 5/8$.

i	$t[i]$	$fac_2(t(i))$ with their REL values	$t'[i]$	REL$(x, t'[i]))$
0		$\emptyset$		
1	aaa	$\{\rtimes a : 6/8, aa : 5/8, \mathbf{a}\ltimes : \mathbf{3/8}\}$	$a\ltimes$	2/8
2	$aaaab$	$\{\rtimes a : 6/8, aa : 5/8, \mathbf{ab} : \mathbf{4/8}, b\ltimes : 5/8\}$	ab	3/8
3	bb	$\{\rtimes b : 2/8, \mathbf{bb} : \mathbf{2/8}, b\ltimes : 5/8\}$	bb	2/8
4	aa	$\{\rtimes a : 6/8, aa : 5/8, \mathbf{a}\ltimes : \mathbf{3/8}\}$	$a\ltimes$	2/8
5	ab	$\{\rtimes a : 6/8, \mathbf{ab} : \mathbf{4/8}, b\ltimes : 5/8\}$	ab	3/8
6	aab	$\{\rtimes a : 6/8, aa : 5/8, \mathbf{ab} : \mathbf{4/8}, b\ltimes : 5/8\}$	ab	3/8
7	baa	$\{\rtimes b : 2/8, \mathbf{ba} : \mathbf{1/8}, aa : 5/8, a\ltimes : 3/8\}$	ba	1/8
8	abb	$\{\rtimes a : 6/8, ab : 4/8, \mathbf{bb} : \mathbf{2/8}, b\ltimes : 5/8\}$	bb	2/8

Table 3: An η-distinguishable-up-to-expectation text for the language in Example 1.

Theorem 13 *SENTIA2 identifies any string extension learning class $\mathcal{L}_f$ on η-distinguishable-up-to-expectation texts in polynomial time.*

Proof Consider any $L \in \mathcal{L}_f$, grammar G with $L(G) = L$, and any η-distinguishable-up-to-expectation text t for L. Now consider any two subsequent iterations in the while loop. Let x denote the min_fac for the first iteration and let y denote the min_fac for the second iteration. We want to show that *count*/i for x and *count*/i for y at their respective iterations is equal to REL$(x, t'[i])$ and REL$(y, t'[i])$, respectively.

Suppose there are no strings in $t[i]$ that contain both x and y. This means that *count*/i during the x iteration is equal to $|$SUPPORT$(x, t[i])|/i$ and that *count*/i during the y iteration is equal to $|$SUPPORT$(y, t[i])|/i$. Moreover, the strings that contain x and the strings that contain y in $t[i]$ have minimum relative factors of x and y, respectively. This means that REL$(x, t'[i])$ equals $|$SUPPORT$(x, t[i])|/i$ and that REL$(y, t'[i])$ equals $|$SUPPORT$(y, t[i])|/i$. Hence, *count*/i = REL$(x, t'[i])$ during the x iteration and *count*/i = REL$(y, t'[i])$ during the y iteration.

Now suppose that there is at least one string in t that contains both factors x and y. This means that *count*/i during the x iteration equals $|$SUPPORT$(x, t[i])|/i$, but that *count*/i during the y iteration equals $|$SUPPORT$(y, t[i]) -$ SUPPORT$(x, t[i])|/i$. Moreover, all strings containing x in $t[i]$ have a minimum relative factor of x, and any strings containing y but not x have a minimum relative factor of y. This means that REL$(x, t'[i])$ equals $|$SUPPORT$(x, t[i])|/i$ and that REL$(y, t'[i])$ equals $|$SUPPORT$(y, t[i]) -$ SUPPORT$(x, t[i])|/i$. Hence, *count*/i = REL$(x, t'[i])$ during the x iteration and *count*/i = REL$(y, t'[i])$ during the y iteration.

■

This section concludes with some observations comparing SENTIA2 with SENTIA1. At each i, SENTIA2 consistently underestimates the number of noisy strings in the text, never dropping more than $\eta \cdot i$ strings. Therefore, there are η-distinguishable texts on which SENTIA1 succeeds but on which SENTIA2 fails. Consider Example 1 (Table 2) illustrating each string in the text and $\eta = 30\%$. η-distinguishability is satisfied, despite there being $3/8 > \eta$ noisy strings. SENTIA2 at $i = 8$ only deletes two strings from the text since $2/8 < 30\% < 3/8$. SENTIA2 returns a different G than what SENTIA1 would return.

On the other hand, SENTIA2 can succeed on texts that SENTIA1 fails on. Consider some $L \in \mathcal{L}_f$ with $L(G) = L$ and suppose $x \in G$, $y, z_1, z_2, ..., z_n \notin G$, some noisy text t such that for all $i > j$, $\textsc{rel}(z_1, t[i]) \leq \textsc{rel}(z_2, t[i]) ... \leq \textsc{rel}(z_n, t[i]) < \eta < \textsc{rel}(x, t[i]) < \textsc{rel}(y, t[i])$. Suppose further that for all $i > j$, $\textsc{support}(z_1, t[i]) \cup \textsc{support}(z_2, t[i]) ... \cup \textsc{support}(z_n, t[i]) = \textsc{support}(y, t[i])$. This means each string in $\textsc{support}(y, t[i])$ must contain one or more of the factors $z_1, z_2, ..., z_n$. Right before it terminates, if SENTIA2 selects each $z_1, ...z_n$ as the factor with the current minimum $\textsc{rel}$ and thus drops the strings supporting those factors, all strings in $\textsc{support}(y, t[i])$ and their factors will have also been deleted from $t[i]$ and $\phi_f(t[i])$ respectively, and SENTIA2 will output $G = \{x\}$ as intended. SENTIA1 will only drop factors z_1, z_2, ..., z_n and output $G = \{x, y\}$.

8. Future Work and Conclusion

Multiset string extension learning which makes use of relative frequencies provides a better understanding of how string extension classes can be learned despite noisy data. SENTIA1 and SENTIA2 are two different approaches, and to investigate further the difference in their behaviors, empirical studies can be carried out and tests can be run on linguistic corpora.

In practice, the noise rate η may be unknown, in which case measures must be taken to estimate the value of η. A common approach used in machine learning is to use hyperparameter estimation on the data corpus of interest. The corpus would be divided into a training set, development set, and test set. A number of models could be trained over a range of η values and their performance on the development set can be compared to select the best one.

Moreover, regarding SENTIA1's efficiency, we have only considered what Pitt (1989) regards as polynomial update time: the inference algorithm is allowed at most $q(n, m_1 + m_2 + ... + m_i)$ steps to produce its i-th hypothesis, where q is any polynomial function of two variables. As he states, polynomial update time is not sufficiently restrictive. In future work, we may want to investigate what happens when we require that the number of changes to the learner's hypothesis be at most $p(n)$ for some polynomial p, in addition to requiring polynomial update time. We may also want to investigate further what happens when we bound the number of implicit prediction errors as well.

We believe that more noise-tolerant algorithms can be devised based on the ideas presented in this paper and future work can identify the evidential relations which characterize the learning problems they solve. For example, Dai (2023) presents an interesting approach of learning phonotactic patterns despite exceptions. His algorithm uses the ratio of the number of observed factors over the number of expected factors. He provides promising empirical results, but no theoretical ones. It would also be of interest to investigate further the types of evidential relations that characterize the learning problem his algorithm solves.

Acknowledgements

We would like to thank the three anonymous reviewers for their helpful comments, which led to an improved version of the paper.

References

Dana Angluin. Inference of reversible languages. *Journal for the Association of Computing Machinery*, 29(3):741–765, 1982.

Dana Angluin and Philip Laird. Learning from noisy examples. *Machine Learning*, 2: 343–370, 1988.

Huteng Dai. A Neo-Trubetzkoyan approach to phonotactic learning in the presence of exceptions. Lingbuzz, 2023. URL `https://ling.auf.net/lingbuzz/007163`.

M. de Brecht, M. Kobayashi, H. Tokunaga, and A. Yamamoto. Inferability of closed set systems from positive data. In *Proc. of the conference of the JSAI*, pages 265–275, 2006.

Henning Fernau. Identification of function distinguishable languages. *Theoretical Computer Science*, 290:1679–1711, 2003.

Mark Fulk and Sanjay Jain. Learning in the presence of inaccurate information. *Theoretical Computer Science*, 161:235–261, 1996.

E.M. Gold. Language identification in the limit. *Information and Control*, 10:447–474, 1967.

Jeffrey Heinz. String extension learning. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 897–906, Uppsala, Sweden, July 2010.

Jeffrey Heinz, Chetan Rawal, and Herbert G. Tanner. Tier-based strictly local constraints for phonology. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics*, pages 58–64, USA, 2011.

Jeffrey Heinz, Anna Kasprzik, and Timo Kötzing. Learning with lattice-structured hypothesis spaces. *Theoretical Computer Science*, 457:111–127, October 2012.

Sanjay Jain. Program synthesis in the presence of infinite number of inaccuracies. *Journal of Computer and System Sciences*, 53:583–591, 1996.

Sanjay Jain, Daniel Osherson, James S. Royer, and Arun Sharma. *Systems That Learn: An Introduction to Learning Theory (Learning, Development and Conceptual Change)*. The MIT Press, 2nd edition, 1999.

Dakotah Lambert, Jonathan Rawski, and Jeffrey Heinz. Typology emerges from simplicity in representations and learning. *Journal of Language Modelling*, 9(1):151–194, 2021.

Daniel Osherson, Scott Weinstein, and Michael Stob. *Systems that Learn.* MIT Press, Cambridge, MA, 1986.

Leonard Pitt. Inductive inference, dfas, and computational complexity. In *Analogical and Inductive Inference*, pages 18–44, Berlin, Heidelberg, 1989.

James Rogers and Geoffrey Pullum. Aural pattern recognition experiments and the subregular hierarchy. *Journal of Logic, Language and Information*, 20:329–342, 2011.

Imre Simon. Piecewise testable events. In *Automata Theory and Formal Languages*, pages 214–222. 1975.

Frank Stephan. Noisy inference and oracles. *Theoretical Computer Science*, 185:129–157, 1997.

Frederic Tantini, Colin de la Higuera, and Jean-Christophe Janodet. Identificaiton in the limit of systematic noisy languages. In *Proceedings of the 8th International Colloquium on Grammatical Inference*, pages 19–31, Tokyo, Japan, September 2006.

Vladimir Vapnik. *Statistical Learning Theory.* Wiley, New York, 1998.