

Syntax Repair as Idempotent Tensor Completion

Breandan Considine¹, Jin Guo¹, and Xujie Si²

¹ McGill University, Montréal, QC H2R 2Z4, Canada

² University of Toronto, Toronto, ON, M5S 1A1 Canada

Abstract. We introduce a new technique for correcting syntax errors in arbitrary context-free languages. To do so, we reduce CFL recognition onto a Boolean tensor completion and compare various techniques for introducing the holes, and solving for their inhabitants. Our technique has practical applications for real-time syntax correction in programming languages.

Keywords: Error correction · CFL reachability · Language games.

1 Introduction

Syntax repair, broadly speaking, is the problem of taking a grammar and a malformed string, and modifying the string so it conforms to the grammar. Prior work has been devoted to fixing syntax errors using handcrafted heuristics. We take a first-principles approach that makes no assumptions about the string or grammar and focuses on accuracy and end-to-end latency. The result is a tool that is applicable to any context-free and conjunctive language, and which is provably sound and complete up to a Levenshtein bound.

1.1 Problem

Syntax repair can be framed as a language intersection problem between a context-free language (CFL) and a string with holes.

Definition 1 (Bounded Levenshtein-CFL reachability). *Given a CFL ℓ and an invalid string $\sigma : \ell^{\mathbb{C}}$, the bounded Levenshtein-CFL reachability problem is to find every valid string reachable within d edits of σ , i.e., letting Δ be the Levenshtein metric and $L(\sigma, d) := \{\sigma' \mid \Delta(\sigma, \sigma') \leq d\}$, we seek to find $L(\sigma, d) \cap \ell$.*

To solve this problem, we will instead pose a simpler problem, then provide a few strategies for solving it exactly or approximately.

Definition 2 (Porous completion). *Let $\underline{\Sigma} := \Sigma \cup \{_\}$, where $_$ denotes a hole. We denote $\sqsubseteq : \Sigma^n \times \underline{\Sigma}^n$ as the relation $\{\langle \sigma', \sigma \rangle \mid \sigma_i \in \Sigma \implies \sigma'_i = \sigma_i\}$ and the set of all inhabitants $\{\sigma' \mid \sigma' \sqsubseteq \sigma\}$ as $H(\sigma)$. Given a porous string, $\sigma : \underline{\Sigma}^*$ we seek all syntactically admissible inhabitants, i.e., $A(\sigma) := H(\sigma) \cap \ell$.*

Definition 3 (Ranked repair). *Given a finite language $\ell^\cap := L(\underline{\sigma}, d) \cap \ell$ and a probabilistic language model $P_\theta : \Sigma^* \rightarrow [0, 1] \subset \mathbb{R}$, the ranked repair problem is to find the top- k repairs by likelihood under the language model. That is,*

$$R(\ell^\cap, P_\theta) := \operatorname{argmax}_{\{\sigma \mid \sigma \subseteq \ell^\cap, |\sigma| \leq k\}} \sum_{\sigma \in \sigma} \prod_{i=1}^{|\sigma|} P_\theta(\sigma_i \mid \sigma_{1\dots i})^{\frac{1}{|\sigma|}} \quad (1)$$

Our main innovation and the central problem this paper tackles is, given $\underline{\sigma}, d, P_\theta$, to find $R(\ell^\cap, P_\theta)$ while minimizing latency and maximizing accuracy.

1.2 Solving

There are a number of strategies to tackling this problem. A first approach requires solving for $A(\sigma)$ using a semiring algebra, and propagating the values from the bottom-up as a string to a list of strings. Letting $D = V \rightarrow \mathcal{P}(\Sigma^*)$, we define $\oplus, \otimes : D \times D \rightarrow D$. Initially, we have $p(s : \Sigma) := \{v \mid [v \rightarrow s] \in P\}$ and $p(_) := \bigcup_{s \in \Sigma} p(s)$, then we compute the fixpoint using the following algebra:

$$X \oplus Z := \{v \rightarrow (X(v) \cup Z(v)) \mid v \in V\} \quad (2)$$

$$X \otimes Z := \bigoplus_{w, x, z} \{w \rightarrow (l + r) \mid [w \rightarrow xz] \in P, \langle l, r \rangle \in X(x) \times Z(x)\} \quad (3)$$

After the fixpoint M^* is attained, the solutions can be read off via $M^*[0, |\sigma|](S)$. The issue here is an exponential growth in cardinality when eagerly computing the Cartesian product, which becomes impractical for even small strings. We can make this encoding more compact by propagating an algebraic data type (ADT) \mathbb{T}_2 using the operations $\oplus, \otimes : 2^{\mathbb{T}_2} \times 2^{\mathbb{T}_2} \rightarrow 2^{\mathbb{T}_2}$ as follows:

$$X \oplus Z := \{\mathbb{T}_2(k, Q_x \cup Q_z) \mid (k, Q)_X \bowtie_k (k, Q)_Z\} \quad (4)$$

$$X \otimes Z := \bigoplus_{w, x, z} \{\mathbb{T}_2(w, \{\langle T_x, T_z \rangle\}) \mid [w \rightarrow xz] \in P, x \in \pi_1(X), z \in \pi_1(Z)\} \quad (5)$$

Decoding, then becomes a matter of enumerating binary trees from the ADT using a recursive choice function that emits a sequence of strings satisfying $A(\sigma)$, with the type signature $\mathcal{C} : \mathbb{T}_2 \rightarrow (\mathbb{N} \rightarrow \Sigma^*)$ defined as follows:

$$\mathcal{C}(t : \mathbb{T}_2) := \begin{cases} \pi_1(t) & \text{if } \pi_2(t) = \emptyset, \text{ or} \\ \{x + z \mid \langle X, Z \rangle \in \pi_2(t), x \in \mathcal{C}(X), z \in \mathcal{C}(Z)\} & \text{otherwise.} \end{cases}$$

1.3 Example

Let us consider an example with two holes, $\sigma = 1 _ _$, and the grammar being $G := \{S \rightarrow NON, O \rightarrow + \mid \times, N \rightarrow 0 \mid 1\}$. This can be rewritten into CNF as $G' := \{S \rightarrow NL, N \rightarrow 0 \mid 1, O \rightarrow \times \mid +, L \rightarrow ON\}$. Using the algebra where $\oplus := \cup$, $X \otimes Z := \{w \mid \langle x, z \rangle \in X \times Z, (w \rightarrow xz) \in P\}$, the fixpoint $M' = M + M^2$ can be computed as follows, shown in the leftmost column:

	2^V	$\mathbb{Z}_2^{ V }$	$\mathbb{Z}_2^{ V } \rightarrow \mathbb{Z}_2^{ V }$
M_0	$\begin{pmatrix} \{N\} \\ \{N, O\} \\ \{N, O\} \end{pmatrix}$	$\begin{pmatrix} \square \blacksquare \square \square \\ \square \blacksquare \blacksquare \square \\ \square \blacksquare \blacksquare \square \end{pmatrix}$	$\begin{pmatrix} V_{0,1} \\ V_{1,2} \\ V_{2,3} \end{pmatrix}$
M_1	$\begin{pmatrix} \{N\} & \emptyset \\ \{N, O\} & \{L\} \\ & \{N, O\} \end{pmatrix}$	$\begin{pmatrix} \square \blacksquare \square \square \square \square \square \\ \square \blacksquare \blacksquare \square \blacksquare \square \square \square \\ \square \blacksquare \blacksquare \square \end{pmatrix}$	$\begin{pmatrix} V_{0,1} & V_{0,2} \\ V_{1,2} & V_{1,3} \\ V_{2,3} \end{pmatrix}$
M_∞	$\begin{pmatrix} \{N\} & \emptyset & \{S\} \\ \{N, O\} & \{L\} \\ & \{N, O\} \end{pmatrix}$	$\begin{pmatrix} \square \blacksquare \square \square \square \square \square \square \blacksquare \\ \square \blacksquare \blacksquare \square \blacksquare \square \square \square \\ \square \blacksquare \blacksquare \square \end{pmatrix}$	$\begin{pmatrix} V_{0,1} & V_{0,2} & V_{0,3} \\ V_{1,2} & V_{1,3} \\ V_{2,3} \end{pmatrix}$

The same procedure can be translated, without loss of generality, into the bit domain $(\mathbb{Z}_2^{|V|})$ using a lexicographic ordering, however these both are recognizers. That is to say, $[S \in M_{0,3}] \Leftrightarrow [M_{0,3,3} = 1] \Leftrightarrow [A(\sigma) \neq \emptyset]$. Since we have $V_{0,3} = \{S\}$, we know there exists at least one solution to $1 _ _$, but M_∞ does not tell us what they are.

In order to extract the inhabitants, we can translate the bitwise procedure into an equation with free variables. To encode $M = M^2$ we proceed as follows, defining $X \otimes Z := [X_2 \wedge Z_1, -, -, X_1 \wedge Z_0]$ and $X \oplus Z := [X_i \vee Z_i]_{i \in [0, |V|]}$.

$$V_{0,2} = V_{0,1} * V_{1,2} \quad (6)$$

$$= [L \in V_{0,2}, \perp, \perp, S \in V_{0,2}] \quad (7)$$

$$= [O \in V_{0,1} \wedge N \in V_{1,2}, \perp, \perp, N \in V_{0,1} \wedge L \in V_{1,2}] \quad (8)$$

$$= [V_{0,1,2} \wedge V_{1,2,1}, \perp, \perp, V_{0,1,1} \wedge V_{1,2,0}] \quad (9)$$

$$V_{1,3} = V_{1,2} * V_{2,3} \quad (10)$$

$$= [L \in V_{1,3}, \perp, \perp, S \in V_{1,3}] \quad (11)$$

$$= [O \in V_{1,2} \wedge N \in V_{2,3}, \perp, \perp, N \in V_{1,2} \wedge L \in V_{2,3}] \quad (12)$$

$$= [V_{1,2,2} \wedge V_{2,3,1}, \perp, \perp, V_{1,2,1} \wedge V_{2,3,0}] \quad (13)$$

$$V_{0,3} = V_{0,j} \cdot V_{i,3} \quad (14)$$

$$= V_{0,1} * V_{1,3} + V_{0,2} * V_{2,3} \quad (15)$$

$$= [V_{0,1,2} \wedge V_{1,3,1} \vee V_{0,2,2} \wedge V_{2,3,1}, \perp, \perp, V_{0,1,1} \wedge V_{1,3,0} \vee V_{0,2,1} \wedge V_{2,3,0}] \quad (16)$$

Now, we only care about $V_{0,3,3} \Leftrightarrow [S \in V_{0,3}]$, so we can ignore the first three entries and solve for:

$$V_{0,3,3} = V_{0,1,1} \wedge V_{1,3,0} \vee V_{0,2,1} \wedge V_{2,3,0} \quad (17)$$

$$= V_{0,1,1} \wedge (V_{1,2,2} \wedge V_{2,3,1}) \vee V_{0,2,1} \wedge \perp \quad (18)$$

$$= V_{0,1,1} \wedge V_{1,2,2} \wedge V_{2,3,1} \quad (19)$$

$$= N \in V_{0,1} \wedge O \in V_{1,2} \wedge N \in V_{2,3} \quad (20)$$

Now we know $\sigma = 1 \underline{Q} \underline{N}$, which tells us $A(\sigma) = \{1 + 0, 1 + 1, 1 \times 0, 1 \times N\}$.

1.4 Ranking

Since the number of solutions can be very large, we can use a language model to rank the results maximizing likelihood, or minimizing perplexity, subject to the constraints. This ranking can be used to guide the propagation, sample the choice function, sample hole locations or as a post-processing step after all solutions have been found.

References

1. Author, B.: Article title. Journal **2**(5), 99–110 (2016)
2. Author, F., Author, S.: Title of a proceedings paper. In: Editor, F., Editor, S. (eds.) CONFERENCE 2016, LNCS, vol. 9999, pp. 1–13. Springer, Heidelberg (2016). <https://doi.org/10.1007/1234567890>
3. Author, F., Author, S., Author, T.: Book title. 2nd edn. Publisher, Location (1999)
4. Author, A.-B.: Contribution title. In: 9th International Proceedings on Proceedings, pp. 1–2. Publisher, Location (2010)
5. LNCS Homepage, . Last accessed 4 Oct 2017