Uniquely Decodable n-gram Embeddings

Leonid Kontorovich
April 28, 2003
CMU-CS-03-134

School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213

Abstract

We define the family of n-gram embeddings from strings over a finite alphabet into the semimodule $\mathbb{N}^K$. We classify all $\xi \in \mathbb{N}^K$ that are valid images of strings under such embeddings, as well as all ξ whose inverse image consists of exactly 1 string (we call such ξ uniquely decodable). We prove that for a fixed alphabet, the set of all strings whose image is uniquely decodable is a regular language.

Part of this research was done at the Hebrew University of Jerusalem. The work at CMU was supported by NSF grant EIA-0205456.

Keywords: n-gram, embedding, finite state automata, strings

1 Introduction

Consider the problem of learning string transformations from examples. We have two alphabets, Σ_1 and Σ_2, and a mapping $f : \Sigma_1^* \to \Sigma_2^*$. A teacher has provided us with numerous examples of the form (u, v) where $u \in \Sigma_1^*$ and $v = f(u)$. The objective is to learn the mapping f.

Among the simplest types of string transformations are those realized by finite state transducers. Yet learning even these is computationally quite hard. There are hardness results for learning FSAs due to Angluin [1], and for approximately learning FSAs due to Pitt and Warmuth [3]. Since FSAs are degenerate cases of FSTs, the hardness carries over to learning FSTs.

We propose the following approach to this problem. Embed the strings $\{u_i\}$ in a vector space $\mathcal{X}$ via $\phi_{\mathcal{X}} : \Sigma_1^* \to \mathcal{X}$ and $\{v_i\}$ in a vector space $\mathcal{Y}$ via $\phi_{\mathcal{Y}} : \Sigma_2^* \to \mathcal{Y}$ (or, more generally, take $\mathcal{X}$ and $\mathcal{Y}$ to be semimodules over semirings) and look for "nice" operators T (for example linear ones) such that $T(\phi_{\mathcal{X}}(u_i)) = \phi_{\mathcal{Y}}(v_i)$.

We point out two specific aspects of this problem. The first is that the mapping f might only make sense on a subset L_1 of Σ_1^* and map it into a subset L_2 of Σ_2^*; if L_1 and L_2 are well-behaved, this could ease our search of good embeddings $\phi_{\mathcal{X}}(\cdot), \phi_{\mathcal{Y}}(\cdot)$ and operators T. The other aspect is that since the ultimate goal is to obtain a mapping on strings, the embedding ϕ should have good invertibility properties.

In this work we study a particular kind of embedding $\phi : \Sigma^* \to \mathbb{N}^K$, namely the bigram one, which generalizes immediately to n-grams. We characterize all $\xi \in \mathbb{N}^K$ which are in the image of Σ^* under ϕ, as well as all $\xi \in \mathbb{N}^K$ such that $\phi^{-1}(\xi)$ consists of a single string.

2 The embedding

Let Σ be a finite alphabet. We will use an additional special character \$ not in Σ, and define

$$\Sigma' = \Sigma \cup \{\$\} = \{\sigma_1 = \$, \sigma_2, \sigma_3, \ldots, \sigma_s\}.$$

We are interested in embedding strings in $\$\Sigma^*\$$ (that is, arbitrary strings in Σ^* padded on the left and right with \$) into a semimodule. We define the **bigram embedding** $\phi : \$\Sigma^*\$ \to \mathbb{N}^{s\times s}$ (where $\mathbb{N} = \{0, 1, 2, \ldots\}$) by

$$[\phi(w)]_{ij} = [\text{the number of times } \sigma_i \text{ occurs immediately before } \sigma_j \text{ in } w] \tag{1}$$

and refer to $\xi = \phi(w)$ as the **(bigram) encoding** of w.

For example, let $\Sigma = \{\mathsf{a}, \mathsf{b}\}$ and consider $w = \$\mathsf{abbbab}\$$. Then

$\phi(\$\mathsf{abbbab}\$) =$

	\$	a	b
\$	0	1	0
a	0	0	2
b	1	1	2

.

Now suppose we are given a bigram encoding ξ and want to recover the original string w. In other words, we are given a $\xi \in \mathbb{N}^{s\times s}$ and are asked to compute $\phi^{-1}(\xi)$. First there is the question of whether $\phi^{-1}(\xi) = \emptyset$, that is, whether ξ is the encoding of *any* $w \in \$\Sigma^*\$$; when the answer is affirmative we call ξ a **valid** encoding. Then there is the question of whether $\phi^{-1}(\phi(w)) = \{w\}$. Note that this is not the case in the above example, where $\phi^{-1}(\phi(w)) = \{\$\mathsf{abbbab}\$, \$\mathsf{abbabb}\$, \$\mathsf{ababbb}\$\}$. When the condition does hold, we say that w (and equivalently, its encoding $\xi = \phi(w)$) is **uniquely decodable**.

We will give necessary and sufficient conditions for resolving both of these questions, and prove that the set of all uniquely decodable w is a regular language $L \subset \$\Sigma^*\$$.

3 Some results

3.1 Basic definitions

Before we can state and prove the main theorems, we need to define some terms and constructs. First we observe that there is a one-to-one correspondence between $\mathbb{N}^{s\times s}$ and the set $\mathcal{G}$ of directed graphs on up to s nodes with positive integer weights on the edges. We denote such graphs by $\mathcal{G}(\xi)$ (or sometimes, abusing notation, $\mathcal{G}(w)$), and set $\mathcal{G}(w) = G = (V, E)$, with $V = \Sigma'$ and $E = \{e(i,j)\}$ where $e(i,j) = \xi_{ij}$ is the weight of the edge from σ_i to σ_j (see Fig. 1). It will occasionally be convenient to interpret G as an unweighted directed multigraph $\tilde{G}$, where the number of multi-edges from σ_i to σ_j is $e(i,j)$. We will freely switch between the weighted-edge and multigraph interpretations, indicating the latter by a tilde. When a graph consists of the single node \$, we will call it the **trivial graph** $G_\$ = \mathcal{G}(\$\$)$.

A **traversal** w of G is a chain of nodes $w = [v_0 = \$, v_1, \ldots, v_T = \$]$;[1] w is a **valid** traversal of G if each edge is traversed a number of times equal to its weight. Define the **self-flow** of a node v to be $e(v,v)$. The **inflow** of v is the sum of the weights of all the edges pointing into v minus v's self-flow; the **outflow** of v is the analogous sum for the edges pointing out of v. We denote the relation $e(u,v) = k > 0$ by $u \xrightarrow{k} v$ (or $u \to v$ if we wish to leave the nonzero k unspecified); if $u \to v$ and $u \neq v$ we say that u is a parent of v and v is a child of u.

If w is a valid traversal of G, we will call it a **decoding** of G. We will abuse notation slightly and use $\phi^{-1}(G)$ and $\phi^{-1}(w)$ to denote the set of strings $\phi^{-1}(\phi(w))$; we will use $\sharp\phi^{-1}(\cdot)$ to denote the cardinality of this set. We say that G is **uniquely decodable** if it has a single decoding, i.e., $\sharp\phi^{-1}(G) = 1$. Call a transformation $T : \mathcal{G} \to \mathcal{G}$ **traversal preserving** on G if $\sharp\phi^{-1}(G) = \sharp\phi^{-1}(T(G))$.

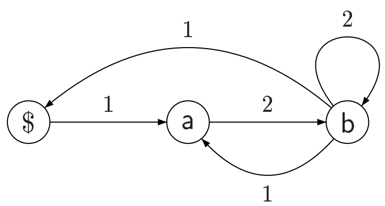

Figure 1: The bigram graph for $w = \$\mathsf{abbbab}\$$

3.2 Valid encodings

Theorem 3.1. *Let $\xi \in \mathbb{N}^{s\times s}$ and let $G = \mathcal{G}(\xi)$. Then ξ is a valid encoding if and only if the following conditions hold:*

(a) *G is a connected graph*

(b) *for all $v \in G$,* $\mathrm{inflow}(v) = \mathrm{outflow}(v)$
(we call graphs with this property ***flow-conserving****)*

(c) *for nontrivial G,* $\mathrm{inflow}(\$) = \mathrm{outflow}(\$) = 1$ *and* $\text{self-flow}(\$) = 0$

[1]As the notation suggests, we blur the distinction between chains of nodes and strings of letters.

Proof. It is clear that for any $w \in \$\Sigma^*\$$, $\mathcal{G}(w)$ satisfies (a)-(c). What remains to be shown is that any ξ that satisfies (a)-(c) is a valid encoding.

First observe that ξ is a valid encoding iff G has a valid traversal w (one simply reads off the nodes in w to obtain a $w \in \phi^{-1}(\xi)$). Call a chain of $t > 1$ nodes $\pi = [v_1, \ldots, v_t]$ valid if the number of transitions from σ_i to σ_j does not exceed $e(i,j)$; we denote this relationship by $v_1 \to^+ v_t$.

We claim that for any node $v \in G$ there is a valid chain from \$ to v. We proceed by induction. If \$ is a parent of v, there is nothing to prove. If u is a parent of v and $\$ \to^+ u$ then clearly $\$ \to^+ v$ (since $e(u,v) \geq 1$). So if there is no valid chain from \$ to v, v must have no parents, which violates (a) or (b).

Thus G must admit a valid chain π starting and ending with \$. Let us translate G into the multigraph $\tilde{G}$. Then only way that π is not a valid traversal is if it fails to traverse some multi-edges of $\tilde{G}$; define the **deficit** of π to be the number of untraversed multi-edges in $\tilde{G}$. Let $\tilde{e}$ be an untraversed multi-edge of $\tilde{G}$ pointing out of a node σ_i. We can assume without loss of generality that all the self-flow multi-edges of σ_i have been traversed (it is trivial to obtain such a valid chain from π). Consider the subgraph of $\tilde{G}$ induced by the nodes that have untraversed multi-edges pointing to/from them; let $\tilde{G}'$ be the connected component of this subgraph that contains σ_i, with the traversed multi-edges deleted from $\tilde{G}'$. By construction, $\tilde{G}'$ is connected; since G is flow-conserving and π, being a valid chain from \$ to \$, traverses an incoming multi-edge of each node as many times as an outgoing multi-edge, when we delete the traversed edges we still have a flow-conserving graph. Now let σ_i play the role of \$ in $\tilde{G}'$ and apply the argument above to obtain a valid chain π' on $\tilde{G}'$, starting and ending with σ_i. The chain π' can now be "spliced" into π (by replacing the first occurrence of σ_i in π by π') to obtain a new valid chain π'' on $\tilde{G}$ of strictly lower deficit. This process can be iterated until the deficit becomes zero, at which point we will have a valid traversal of G. ☐

Corollary 3.2. *A string w is uniquely decodable iff $\mathcal{G}(w)$ has a single valid traversal.*

Let us henceforth call a graph G **valid** if it satisfies (a)-(c) in 3.1.

3.3 Unique decodings

The task of characterizing the uniquely decodable $\{\mathcal{G}(w)\}$ is going to involve a somewhat detailed analysis, but the following simple intuition may be helpful. If every node v of G has only 1 child then clearly G will only have 1 traversal (which will have to be valid since G is connected and flow-conserving). Potential ambiguities in decoding G can only arise when a node of G branches into two or more children. Thus one might conjecture that multiple children lead to multiple decodings. But that is not necessarily the case; consider the example of $w = \$\mathsf{acab}\$$ in Fig. 2. Here, the node a has 2 children (b and c), yet $\mathcal{G}(w)$ admits only one decoding. The node c is "obligatory" in the sense that the valid traversal must visit it immediately after a, for if we go to b first, we have no way of ever visiting c. The idea of the proof is rather simple: we prune such obligatory nodes; if eventually we are left only with the \$ node (which we never prune), G is uniquely decodable, otherwise, it is not.

In all subsequent discussion, the graphs are assumed to have a valid traversal (that is, to satisfy (a)-(c) in 3.1).

In order to characterize the "obligatory" nodes described above, we will need some more definitions.

Let us define the **pruning** procedure for nodes in G with only 1 child. Consider the multigraph $\tilde{G}$, draw a multi-edge from every parent of x to the child of x, then delete x and all the multi-edges

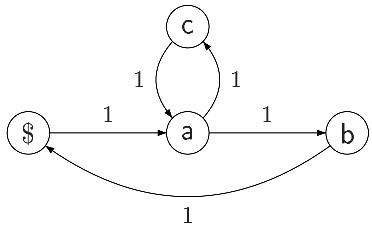

Figure 2: The bigram graph for $w = \$\mathsf{acab}\$$

pointing to/from it. Call the weighted-edge interpretation of the resulting graph x-pruned. Let P_x denote the operator that prunes the node x. Define the action of P_x on strings $w \in \$\Sigma^*\$$ as deleting all occurrences of the letter x from w.

Lemma 3.3. *Suppose $\$ \neq x$ in G only has* 1 *child and let $G' = P_x(G)$. If G is a valid graph, in the sense of satisfying* (a)-(c) *of thm. 3.1, then so is G'.*

Proof. (a) *Connectedness.* Suppose there is a chain (directed path) in G from $y \neq x$ to $z \neq x$, passing through x. Then there is a chain (possibly of length 0) from y to a parent of x. Likewise, there is a chain from the child of x to z. Since by construction, in G' every former parent of x points to the former child of x, there is a chain from y to z in G'.
(b) *Conservation of flow.* Every parent of x loses a multi-edge pointing to x but gains one pointing to the child of x. Suppose there were k such multi-edges. This means that the inflow and outflow of x in G is k, so there are k multi-edges pointing from x to its only child (since G is flow-conserving). The child of x loses these k edges from x but gains them from the parents of x. It follows that for $z \neq x$, the inflow and outflow of z do not change after x is pruned (the self-flow of z will increase if it is both a parent and a child of x in G).
(c) *Boundary conditions on* \$. \$ has exactly 1 parent $a \xrightarrow{1} \$$ and 1 child $b \xrightarrow{1} \$$ in G, and no self-flow. It now follows from (b) that after $x \neq \$$ is pruned, a nontrivial G' will also satisfy these conditions. □

Definition 1. We call a node $x \neq \$$ in G **removable** (with child b) if:

1. x has a single child b
2. none of the parents of x point to b
3. if $x \to x$ then $x \xrightarrow{1} b$
   (if x points to itself, its outflow must be 1)

Let us call such x **type-I removable** if x outflow$(x) = 1$ and **type-II removable** otherwise.

Let us introduce another bit of notation: $\Sigma_{\overline{x}} = \Sigma' \setminus \{x\}$; if $L \subset \Sigma^*$, then $\overline{L} \equiv \Sigma^* \setminus L$. Our first result is that pruning a type-I removable node does not alter the unique-decodability of a graph:

Theorem 3.4. *Let $x \in G$ be type-I removable with child b and let $G' = P_x(G)$. Then $\sharp\phi^{-1}(G) = \sharp\phi^{-1}(G')$ (i.e., P_x is traversal-preserving).*

Proof. Since x has a outflow of 1, it can only have one parent, say a.

Let w be a valid traversal on G. We view w as a string in $\$\Sigma^*\$$. Note that since $x \in G$ is type-I removable, w is contained in the regular language

$$L^{(1)}_{a,x,b} = L(\Sigma^*_{\overline{x}} a x^+ b \Sigma^*_{\overline{x}}) \cap \overline{L(\Sigma'^* ab \Sigma'^*)} \cap L(\$\Sigma^*\$). \tag{2}$$

Informally, $L^{(1)}_{a,x,b}$ is the set of strings in which x occurs as a single contiguous stretch between a and b, and the substring ab does not occur. So we can write $w = \alpha a x^k b \beta$. When we x-prune G we obtain a valid traversal $w' = \alpha ab\beta$ on G'.

To prove the theorem we must show that $P_x : w \mapsto w'$ is bijective. But this is clear: P_x consists of removing the single contiguous stretch of x's from w, while P_x^{-1} consists of inserting the sequence x^k inside the single occurrence of ab in w' (which is a well-defined operation since G' encodes strings that do not contain x and contain exactly one occurrence of ab).

We have established a one-to-one correspondence between the valid traversals on such G and on G'; this proves the theorem. □

We now state a similar result for type-II removable nodes:

Theorem 3.5. *Let $x \in G$ be type-II removable with child b and let $G' = P_x(G)$. Then $\sharp\phi^{-1}(G) = \sharp\phi^{-1}(G')$.*

Proof. As in the proof of Thm. 3.4, we use the existence of the type-II removable node x with child b and parents $A = \{a_1, a_2, \ldots, a_p\}$ to describe the set of all valid traversals w on G by the regular language

$$L^{(2)}_{A,x,b} \;=\; L((\Sigma^*_{\overline{x}} A x b \Sigma^*_{\overline{x}})^+) \cap \overline{L(\Sigma'^* a_1 b \Sigma'^*)} \cap \overline{L(\Sigma'^* a_2 b \Sigma'^*)} \cap \ldots \cap \overline{L(\Sigma'^* a_p b \Sigma'^*)} \cap L(\$\Sigma^*\$). \tag{3}$$

[2] Informally, $L^{(2)}_{A,x,b}$ is the set of strings where x occurs in stretches of 1, only when preceded by a member of A and followed by b, and no $a \in A$ can precede b. If w is a valid traversal on G, $w' = P_x(w)$ is a valid traversal on G'. The bijectivity of $P_x : w \mapsto w'$ is easily seen: P_x replaces every occurrence of $a_i x b$ in w by $a_i b$; P_x^{-1} replaces every occurrence of $a_i b$ in w' by $a_i x b$. Thus type-II pruning is a traversal-preserving operation, so the theorem is proved. □

Remark 3.6. Note that in general, P_x is not invertible; e.g., we can compute $P_{\mathsf{c}}(\$\mathsf{abcacb}\$) = \$\mathsf{abab}\$$ but $P_{\mathsf{c}}^{-1}(\$\mathsf{abab}\$)$ is not unique (it is not clear where to insert the c's, and how many). However, when restricted to the domain $\mathcal{G}^{(1)}_{a,x,k,b}$ of graphs where x is type-I removable with parent a, child b, and self-flow k, P_x is invertible, both as an operator on graphs $G \in \mathcal{G}^{(1)}_{a,x,k,b}$ and strings $w \in \phi^{-1}(G)$. A similar remark holds for type-II removable nodes. In the sequel, when we talk about the invertibility of P_x for removable x and write P_x^{-1}, we shall always have in mind the appropriately restricted domain.

Lemma 3.7. *Let G be a valid graph such that every $x \neq \$$ in G has* 2 *children and* self-flow$(x) = 0$. *Then G has multiple valid traversals.*

[2] Here, A denotes both an unordered set and the regular expression $A = (a_1 + a_2 + \ldots + a_p)$. Note that (3) does not adequately handle the case where x's child b is a member of A. This is fixed by letting

$$L^{(2)}_{A,x,b} = L((\Sigma^*_{\overline{x}} A x b(\varepsilon + (xb)^*)\Sigma^*_{\overline{x}})^+) \cap \overline{L(\Sigma'^* a_1 b \Sigma'^*)} \cap \overline{L(\Sigma'^* a_2 b \Sigma'^*)} \cap \ldots \cap \overline{L(\Sigma'^* a_p b \Sigma'^*)} \cap L(\$\Sigma^*\$)$$

when $b \in A$.

Proof. Let G be as stated and assume (to get a contradiction) that G is uniquely decodable. Let x be the child of \$ in G. Let b_0 and c be the two children of x. Then the valid traversal w of G must be of the form
(i) $w = \$xb_0\beta xc\gamma\$$
or
(ii) $w = \$xc\gamma xb_0\beta\$$,
where $\beta = b_1b_2\ldots b_K$ and γ are strings over Σ. Now (i) and (ii) cannot both be possible, for then G would have more than one decoding. So suppose (i) is the only type of decoding possible for G. Since (ii) is not a possible decoding of G, there must be no way to return to x after the $x \to c$ edge is traversed; this means that γ cannot contain any of $\{b_i\}_{i=0}^{K}$. But each node in G has two children, so each letter in w must appear at least twice. This means that there is a smallest k_0, $0 < k_0 \leq K$ such that $b_{k_0} = b_0$, so we can write $w = \$xb_0b_1b_2\ldots b_{k_0-1}b_0b_{k_0+1}\ldots b_K\beta xc\gamma\$$.

Now we can apply the same argument to the two children of b_0 to conclude that w must be of the form $w = \$xb_0b_1\ldots b_{k_1-1}b_1b_{k_1+1}\ldots b_{k_0-1}b_0b_{k_0+1}\ldots b_K\beta xc\gamma\$$. We can repeat this process t times, obtaining $w = \$xb_0b_1b_2\ldots b_t\ldots b_{k_t-1}b_{k_t}b_{k_t+1}\ldots b_{k_0-1}b_{k_0}b_{k_0+1}\ldots b_K\beta xc\gamma\$$ and noting that for each t, we have

1. $t < k_t$
2. $k_{t'} < k_t$ for $t' > t$
3. $b_t = b_{k_t}$

This process assigns to each b_t that occurs in w for the first time a location k_t where it occurs for the second time, and distinct t's are assigned distinct k_t's. Suppose there are T distinct letters $\{b_t\}$. By §1 we cannot have $k_T \leq T$. But $k_T > T$ is also impossible: since there are only T distinct b_t's, we have that $b_{T'>T}$ has already occurred as $b_{t'}$ for $t' < T$, and so $k_{t'} = T'$ is already "taken".

We have reached a contradiction, so G must have more than 1 decoding.

□

Theorem 3.8. *If a nontrivial graph G has a single valid traversal then there is a removable $x_0 \in G$.*

Proof. Suppose a nontrivial G has no removable nodes. A node $x \neq \$$ in G is not removable if it violates any item of def. 1:

1. x has multiple children
2. a parent of x points to a child of x
3. self-flow$(x) > 0$ and outflow$(x) > 1$

We observe immediately that if $a \to x \to b$ and $a \to b$ then any decoding w of G must contain the substrings axb and ab, but there are at least two orders in which the two substrings can occur, so G is not uniquely decodable.

Likewise, outflow$(x) > 1$ means that x appears in at least two distinct locations of a decoding w, and self-flow$(x) > 0$ means that at any of these locations two or more consecutive x's may occur; this also precludes G from being uniquely decodable.

We claim that if x has 3 or more children, G cannot be uniquely decodable. Let x have children a, b, and c. It may be that after the $x \to a$ edge is traversed, there is no way back to x, but this still leaves at least two decodings of G: $w = \$\alpha xb\beta xc\gamma xa\alpha'\$$ and $w = \$\alpha xc\gamma xb\beta xa\alpha'\$$.

The only scenario we have left to deal with is one where every $x \neq \$$ in G has two children and no self-loops, and this is handled by lemma 3.7.

This shows that if a nontrivial G has no removable nodes, it cannot be uniquely decodable, and proves the theorem.

□

Corollary 3.9. *If G is uniquely decodable then there is a sequence of nodes $x_0, x_1, \ldots, x_T$ and a sequence of graphs $G = G^{(0)}, G^{(1)}, \ldots, G^{(T+1)} = G_\$$ such that x_t is removable in $G^{(t)}$ and $G^{(t+1)} = P_{x_t}(G^{(t)})$.*

Proof. If $G = G^{(0)}$ is uniquely decodable, then thm. 3.8 furnishes a removable x_0 in $G^{(0)}$. Theorems 3.4 and 3.5 show that pruning a removable node is a traversal-preserving operation, ensuring the unique decodability of $G^{(1)} = P_{x_0}(G^{(0)})$. This process may now be continued until x_T is pruned and $G^{(T+1)} = G_\$$ is what remains. □

Theorem 3.10. *If, for a graph G, there is a sequence of nodes $x_0, x_1, \ldots, x_T$ and a sequence of graphs $G = G^{(0)}, G^{(1)}, \ldots, G^{(T+1)} = G_\$$ such that x_t is removable in $G^{(t)}$ and $G^{(t+1)} = P_{x_t}(G^{(t)})$ then G has a single valid traversal.*

Proof. Let $w_{T+1} = \$\$$ and let $w_t = P_{x_t}^{-1}(w_{t+1})$ for $t = T, T-1, \ldots, 0$ (with the qualification made in remark 3.6). It is straightforward to verify that for $0 \leq t \leq T+1$, w_t is a decoding of $G^{(t)}$. Since $G^{(T+1)} = G_\$$ obviously only has 1 valid traversal and each P_{x_k} is traversal-preserving, w_0 is the unique decoding of $G = G^{(0)}$. □

The preceding results give a simple algorithm for determining whether G is uniquely decodable: iteratively prune the removable nodes of G until we are left with $G_\$$ or a non-trivial graph with no removable nodes. In the former case, the answer is affirmative; in the latter, it is negative. Thm. 3.10 gives a simple way to construct the decoding of G.

3.4 Characterization as a regular language

The last section characterizes the uniquely decodable $w \in \$\Sigma^*\$$: $G = \mathcal{G}(w)$ is uniquely decodable iff there is a "pruning" sequence of nodes $x_0, x_1, \ldots, x_T$ and a sequence of graphs $G = G^{(0)}, G^{(1)}, \ldots, G^{(T+1)} = G_\$$ such that x_t is removable in $G^{(t)}$ and $G^{(t+1)} = P_{x_t}(G^{(t)})$. Let $L_{\text{uniq}} \subset \$\Sigma^*\$$ denote the set of uniquely decodable strings. The next result shows that L_{uniq} is a rather well-behaved set:

Theorem 3.11. *L_{uniq} is a regular language.*

Proof. It follows from (2) and (3) that L_x, the set of all $w \in \$\Sigma^*\$$ with a removable node x, is a regular language:

$$L_x = \left(\bigcup_{a,b \in \Sigma_{\overline{x}}} L^{(1)}_{a,x,b} \right) \cup \left(\bigcup_{A \subset \Sigma_{\overline{x}}, b \in \Sigma_{\overline{x}}} L^{(2)}_{A,x,b} \right),$$

since it is a finite union of regular languages. Therefore there is a deterministic finite state automaton M_x that accepts L_x. Now M_x is trivially converted into a finite state transducer T_x, which deletes the letter x from w: every state transition $(q, a \neq x) \to q'$ in M_x becomes $(q, a) \to (q', a)$ in T_x, and $(q, x) \to q'$ in M_x becomes $(q, x) \to (q', \varepsilon)$ in T_x.

Suppose $G = \mathcal{G}(w)$ is uniquely decodable and let a given $x_0, x_1, \dots, x_K$ be a pruning sequence for G, with $w^{(k)}$ as the unique decoding of $G^{(k)}$. Let $T_{\{x_k\}}$ be the concatenation of the transducers T_{x_k}:

$$T_{\{x_k\}}(w) = T_{x_K} T_{x_{K-1}} \dots T_{x_0}(w) = \$\$,$$

which is well defined on w since the application of each T_{x_k} produces a string with a removable node x_{k+1}. It is now straightforward to convert $T_{\{x_k\}}$ into the FSA $M_{\{x_k\}}$ which accepts precisely the strings w with a pruning sequence $x_0, x_1, \dots, x_K$. Observe that since the alphabet size $s = |\Sigma|$ is finite, so is the number of possible pruning sequences (it is bounded by $N = \sum_{t=0}^{s} \frac{s!}{(s-t)!}$). Letting M be the finite union over all the $M_{\{x_k\}}$'s, we obtain an FSA which accepts L_{uniq}.

$\square$

3.5 Extension to higher n-grams

All of our discussion up to now has dealt with the bigram embedding. Now we show that the results readily generalize to higher n-grams.

Define the n-gram embedding $\phi_n : \$^{n-1}\Sigma^*\$ \to \mathbb{N}^{s^n}$ by

$$[\phi_n(w)]_{i_1 i_2 \dots i_n} = [\text{the number of times the substring } \sigma_{i_1}\sigma_{i_2}\dots\sigma_{i_n} \text{ occurs in } w]. \tag{4}$$

Let $\Sigma'_n = (\Sigma')^n$ be the set of all ordered n-tuples of letters in Σ'. It is clear that any string $w \in \$^{n-1}\Sigma^*\$$ can be written as a string Ω over $(\Sigma'_n)^*$, by sliding a window of length n left-to-right across w and letting the contents of the window at time t be the tth "character" of Ω. However, not every string Ω over $(\Sigma'_n)^*$ can be interpreted as a string $w \in \$^{n-1}\Sigma^*\$$: we need adjacent "characters" to overlap by $(n-1)$ letters. Formally, if $\Omega = \omega_1\omega_2\dots\omega_T$, adjacent characters $\omega_t = (\sigma_{i_1}\sigma_{i_2}, \dots, \sigma_{i_n})$ and $\omega_{t+1} = (\sigma_{j_1}\sigma_{j_2}, \dots, \sigma_{j_n})$ must satisfy

$$\sigma_{j_k} = \sigma_{i_{k+1}} \text{ for } 1 \le k \le n-1 \tag{5}$$

(Ω must also satisfy the obvious boundary conditions: $\omega_1 = (\$, \$, \dots, \$, \sigma)$ and $\omega_t = (\sigma_{i_1}, \sigma_{i_2}, \dots, \sigma_{i_{n-1}}, \$)$). We call such Ω valid strings over $(\Sigma'_n)^*$. Now all questions about n-gram embeddings of $w \in \$^{n-1}\Sigma^*\$$ have been reduced to questions about bigram embeddings of valid $\Omega \in (\Sigma'_n)^*$, and all of our previous results apply.

Perhaps one of them is worth stating as a theorem:

Theorem 3.12. *Let $L^{(n)}_{\text{uniq}}$ be the set of all uniquely decodable strings $w \in \$^{n-1}\Sigma^*\$$ under the n-gram embedding ϕ_n. Then $L^{(n)}_{\text{uniq}}$ is a regular language.*

Proof. Theorem 3.11 shows that Λ_{uniq}, the set of all valid, uniquely decodable $\Omega \in (\Sigma'_n)^*$ is a regular language. Let M_Λ be a deterministic finite state automaton which accepts Λ_{uniq}. But now it is a simple matter to convert M_Λ into M', an automaton which accepts $L^{(n)}_{\text{uniq}}$. Let M' be a deterministic FSA with the same state space as M_Λ (plus an extra state to consume all the leading \$'s). For every deterministic transition $(q, \omega) \to q'$ in M_Λ, with $\omega = (\sigma_{i_1}\sigma_{i_2}, \dots, \sigma_{i_n})$, let M' have the deterministic transition $(q, \sigma_{i_n}) \to q'$. The requirement that Ω be valid, expressed by (5), ensures that M_Λ and M' perform identical tasks. $\square$

4 Conclusion

We have characterized those strings that are uniquely decodable under the n-gram embedding, showing in particular that for a fixed n and alphabet Σ, they form a regular language. This may be viewed as a step in the direction of tackling the problems outlined in the Introduction.

Several immediate questions are left unanswered:

1. Which finite state transducers leave $L_{\text{uniq}}^{(n)}$ invariant? That is, for which FSTs M, does $M(L_{\text{uniq}}^{(n)}) \subset L_{\text{uniq}}^{(n)}$ hold? What about $M(L) \subset L$ for some general fixed regular language L?

2. Let $\mathcal{M}_{\text{uniq}}$ be the set of FSTs M such that $M(L_{\text{uniq}}^{(n)}) \subset L_{\text{uniq}}^{(n)}$ – i.e., the transducers that leave $L_{\text{uniq}}^{(n)}$ invatiant. Each such M induces an operator T_M on the embedded n-grams $\xi = \phi_n(w)$, defined in (4). What interesting connections can we make between the transducer M and the semimodule operator T_M? We might tentatively conjecture that for a fixed alphabet $\Sigma = \{\sigma_1, \ldots, \sigma_s\}$ and a given transducer there exists an n-gram embedding such that the transduction corresponds to a "nice" (e.g., linear) operator on the embedded elements $\xi \in \mathbb{N}^{s^n}$.

3. Given a regular language L in some description (regular expression, an automaton), what does $\phi(L) = Z \subset \mathbb{N}^K$ look like? Can a result similar in spirit to Parikh's Theorem ([2], p. 127) be obtained? In particular, what does $\phi(L_{\text{uniq}})$ look like[3]? If $\phi(L) = Z \subset \mathbb{N}^K$, then which families of (linear) operators on $\mathbb{N}^K$ leave Z invariant?

We intend to investigate these in the near future.

Acknowledgements

I would like to thank Anupam Gupta, Alex Kontorovich, John Lafferty, and Yoram Singer for helpful and insightful discussions.

References

[1] D. Angluin. On the complexity of minimum inference of regular sets. *Information and Control*, 39:337–350, 1978.

[2] H. Lewis and C. Papadimitriou. *Elements of the Theory of Computation*. Prentice Hall, Inc., 1981.

[3] L. Pitt and M. Warmuth. The minimum consistent dfa problem cannot be approximated within any polynomial. *JACM*, 40(1), 1993.

[3] It seems we have found a simple answer to this question and intend to address it in future work.