A* shortest string decoding for non-idempotent semirings

Anonymous ACL submission

Abstract

The single shortest path algorithm is undefined for weighted finite-state automata over non-idempotent semirings because such semirings do not guarantee the existence of a shortest path. However, in non-idempotent semirings admitting an order satisfying a monotonicity condition (such as the plus-times or log semirings), the shortest string is well-defined. We describe an algorithm which finds the shortest string for a weighted non-deterministic automaton over such semirings using the backwards shortest distance of an equivalent deterministic automaton (DFA) as a heuristic for A* search performed over a companion idempotent semiring, which is proven to return the shortest string. There may be exponentially more states in the DFA, but the proposed algorithm needs to visit only a small fraction of them if determinization is performed "on the fly".

1 Introduction

Weighted finite-state automata provide a compact representation of hypotheses in various speech recognition and text processing applications (e.g., Mohri, 1997; Mohri et al., 2002; Roark and Sproat, 2007; Gorman and Sproat, 2021). Under a wide range of assumptions, weighted finite-state lattices allow for efficient polynomial-time decoding via shortest-path algorithms (Mohri, 2002).

The shortest path—and the algorithms that compute it—are well-defined when the weights of a lattice are *idempotent* and exhibit the *path property*. These properties formalized below, but informally these they hold that the distance between any two states corresponds to a single path between those states, so that the shortest-path algorithm—having identified this path—does not need to consider the weights of competing paths between those states. However, when the weights of a lattice lack these two properties, there is no guarantee that a shortest path between any two states exists. This situation arises in many speech and language technologies. For instance, generative models for speech recognition and machine translation—and in many unsupervised settings—many require one to learn alignments between sequences using *expectation maximization* (EM; Dempster et al., 1977). EM inference may require one to consider multiple competing paths between pairs of states, and this is incompatible with these two properties. Thus, to efficiently decode a lattice constructed using EM, heuristics are required; one can decode approximately by interpreting the lattice weights as if they were idempotent and had the path property, or can construct the lattice itself using the Viterbi approximation to EM.[1]

In non-idempotent semirings admitting an order satisfying a monotonicity condition, the shortest string is undefined but the closely related notion of *shortest string* is well-defined. We show below that it is still possible to efficiently determine the shortest string for lattices defined over non-idempotent monotonic negative semirings such as the plus-times and log semirings, both used for expectation maximization. We propose a simple algorithm for decoding the shortest string over such semirings which combines shortest-path search with the A* queue discipline (Hart et al., 1968) and "on the fly" determinization (Mohri, 1997). After providing definitions and the algorithm, we describe an implementation and evaluate it using word lattices produced by a speech recognizer. The algorithm is found to scale well as a function of lattice size.

2 Definitions

Before we introduce the proposed decoding algorithm we provide definitions of key notions.

[1] Both of these strategies are discussed in Brown et al. 1993; see §4.3 and §6.2, respectively.

2.1 Semirings

Weighted automata algorithms operate with respect to an algebraic system known as a *semiring*, defined by the combination of two *monoids*.

Definition 2.1. A *monoid* is a pair $(\mathbb{K}, \bullet)$ where $\mathbb{K}$ is a set and $\bullet$ is a binary operator over $\mathbb{K}$ with the following properties:

1. *closure*: $\forall a, b \in \mathbb{K} : a \bullet b \in \mathbb{K}$.
2. *associativity*: $\forall a, b, c \in \mathbb{K} : (a \bullet b) \bullet c = a \bullet (b \bullet c)$.
3. *identity*: $\exists e \in \mathbb{K} : e \bullet a = a \bullet e = a$.

Definition 2.2. A monoid is *commutative* in the case that $\forall a, b \in \mathbb{K} : a \bullet b = b \bullet a$.

Definition 2.3. A semiring is a five-tuple $(\mathbb{K}, \oplus, \otimes, \bar{0}, \bar{1})$ such that:

1. $(\mathbb{K}, \oplus)$ is a commutative monoid with the identity element $\bar{0}$.
2. $(\mathbb{K}, \otimes)$ is a monoid with the identity element $\bar{1}$.
3. $\forall a \in \mathbb{K} : a \otimes \bar{0} = \bar{0} \otimes a = \bar{0}$.
4. $\forall a, b, c \in \mathbb{K} : a \otimes (b \oplus c) = (a \otimes b) \oplus (a \otimes c)$.

Definition 2.4. A semiring is *zero-sum-free* if non-$\bar{0}$ elements cannot sum to $\bar{0}$; that is, $\forall a, b \in \mathbb{K} : a \oplus b \implies a = b = \bar{0}$.

Definition 2.5. A semiring is *idempotent* if $\oplus$ is idempotent; that is, $\forall a \in \mathbb{K} : a \oplus a = a$.

Definition 2.6. A semiring has the *path property* if $\forall a, b \in \mathbb{K} : a \oplus b \in \{a, b\}$.

Remark 2.1. If a semiring has the path property it is also idempotent.

Definition 2.7. The *natural order* of an idempotent semiring is a boolean operator $\preceq$ such that $\forall a, b \in \mathbb{K} : a \preceq b$ if and only if $a \oplus b = a$.

Remark 2.2. In a semiring with the path property, the natural order is a *total* order. That is, $\forall a, b \in \mathbb{K}$, either $a \preceq b$ or $b \preceq a$.

Definition 2.8. A semiring is *monotonic* if $\forall a, b, c \in \mathbb{K}$, $a \preceq b$ implies:

1. $a \oplus c \preceq b \oplus c$.
2. $a \otimes c \preceq b \otimes c$.
3. $c \otimes a \preceq c \otimes b$.

Definition 2.9. A semiring is *negative* if and only if $\bar{1} \preceq \bar{0}$.

Remark 2.3. In a monotonic negative semiring, $\forall a, b \in \mathbb{K} : a \preceq \bar{0}$ and $a \oplus b \preceq b$.

Some examples of monotonic negative semirings are given in Table 1.

Definition 2.10. The *companion semiring* of a monotonic negative semiring $(\mathbb{K}, \oplus, \otimes, \bar{0}, \bar{1})$ with total order $\preceq$ is the semiring $(\mathbb{K}, \widehat{\oplus}, \otimes, \bar{0}, \bar{1})$ where $\widehat{\oplus}$ is the minimum binary operator for $\preceq$:

$$a \mathbin{\widehat{\oplus}} b = \begin{cases} a & \text{if } a \preceq b \\ b & \text{otherwise} \end{cases}$$

Remark 2.4. The max-times and tropical semirings are companion semirings to the plus-times and log semirings, respectively.

Remark 2.5. By construction a companion semiring has the path property and natural order $\preceq$.

2.2 Weighted finite-state acceptors

Without loss of generality, we consider single-source ϵ-free weighted finite-state acceptors.[2]

Definition 2.11. A *weighted finite-state acceptor* (WFSA) is defined by a five-tuple $(Q, s, \Sigma, \omega, \delta)$ and a semiring $(\mathbb{K}, \oplus, \otimes, \bar{0}, \bar{1})$ where:

1. Q is a finite set of states.
2. $s \in Q$ is the *initial state*.
3. Σ is the *alphabet*.
4. $\omega \subseteq Q \times \mathbb{K}$ is the *final weight function*.
5. $\delta \subseteq Q \times \Sigma \times \mathbb{K} \times Q$ is the *transition relation*.

Definition 2.12. An WFSA is *acyclic* if there exists a *topological ordering*, an ordering of the states such that if there is a transition from state q to r where $q, r \in Q$, then q is ordered before r. Otherwise, the WFSA is *cyclic*.

2.3 Shortest distance

Definition 2.13. A state $q \in Q$ is *final* if $\omega(q) \neq \bar{0}$.

Definition 2.14. Let $F = \{q \mid \omega(q) \neq \bar{0}\}$ denote the set of final states.

Definition 2.15. A *path* through an acceptor p is a triple consisting of:

[2] The definition provided here can easily be generalized to automata with multiple initial states, with initial or final weights, or with ϵ-transitions (e.g., Roark and Sproat, 2007, ch. 1, Mohri, 2009, Gorman and Sproat, 2021, ch. 1).

	$\mathbb{K}$	$\oplus$	$\otimes$	$\bar{0}$	$\bar{1}$	$\preceq$
Plus-times	$\mathbb{R}_+$	$+$	$\times$	0	1	$\geq$
Max-times	$\mathbb{R}_+$	max	$\times$	0	1	$\geq$
Log	$\mathbb{R} \cup \{-\infty, +\infty\}$	$\oplus_{\log}$	$+$	$+\infty$	0	$\leq$
Tropical	$\mathbb{R} \cup \{-\infty, +\infty\}$	min	$+$	$+\infty$	0	$\leq$

Table 1: Common monotonic negative semirings; $a \oplus_{\log} b = -\ln(e^{-a} + e^{-b})$.

1. a state sequence $q[p] = q_1, q_2, \ldots, q_n \in Q^n$,
2. a weight sequence $k[p] = k_1, k_2, \ldots, k_n \in \mathbb{K}^n$, and
3. a string $z[p] = z_1, z_2 \ldots, z_n \in \Sigma^n$

subject to the constraint that $\forall i \in [1, n] : (q_i, z_i, k_i, q_{i+1}) \in \delta$; that is, each transition from q_i to q_{i+1} must have label z_i and weight k_i.

Definition 2.16. Let $P_{q \to r}$ be the set of all paths from q to r where $q, r \in Q$.

Definition 2.17. The *forward shortest distance* $\alpha \subseteq Q \times \mathbb{K}$ is a partial function from a state $q \in Q$ that gives the $\oplus$-sum of the $\otimes$-product of the weights of all paths from the initial state s to q:

$$\alpha(q) = \bigoplus_{p \in P_{s \to q}} \bigotimes_{k_i \in k[p]} k_i.$$

Definition 2.18. The *backwards shortest distance* $\beta \subseteq Q \times \mathbb{K}$ is a partial function from a state $q \in Q$ that gives the $\oplus$-sum of the $\otimes$-product of the weights of all paths from q to a final state, including the final weight of that final state:

$$\beta(q) = \bigoplus_{f \in F} \left(\bigoplus_{p \in P_{q \to f}} \bigotimes_{k_i \in k[p]} k_i \otimes \omega(f) \right).$$

Remark 2.6. For a state q, $\alpha(q)$ and $\beta(q)$ are defined if and only if q is accessible and coaccessible, respectively.

Definition 2.19. The *total shortest distance* through an automaton is given by $\beta(s)$.

2.4 Shortest path

Definition 2.20. A path is *complete* if

1. $(s, z_1, k_1, q_1) \in \delta$.
2. $q_n \in F$.

That is, a complete path must also begin with an arc from the initial state s to q_1 with label z_1 and weight k_1, and halt in a final state.

Definition 2.21. The weight of a complete path is given by the $\otimes$-product of its weight sequence and its final weight:

$$\bar{k} = \left(\bigotimes_{k_i \in k[p]} k_i \right) \otimes \omega(q_n).$$

Definition 2.22. A *shortest path* through an automaton is a complete path whose weight is equal to the total shortest distance $\beta(s)$.

Remark 2.7. Automata over non-idempotent semirings need not have a shortest path (Mohri, 2002, 322). Consider for example the NFA shown in the left side of Figure 1. Let us assume that $k \oplus k \preceq k < k'$. Then, the total shortest distance is $k \oplus k$ but the shortest path is k. Definitionally, a non-idempotent semiring does not guarantee that these two weights will be equal. In that case, there is no complete path whose weight is that of the total shortest distance, and thus there is no shortest path.

Remark 2.8. It is generally impossible to find the shortest path efficiently over non-monotonic semirings.[3]

2.5 Determinization

Definition 2.23. A WFSA is *deterministic* if, for each state $q \in Q$, there is at most one transition with a given label $z \in \Sigma$ from that state, and *non-deterministic* otherwise.

Definition 2.24. A zero-sum-free semiring is *weakly divisible* if

$$\forall a, b \in \mathbb{K} \; \exists c \in \mathbb{K} : a = (a \oplus b) \otimes c.$$

Definition 2.25. A weakly divisible semiring is *cancellative* if c is unique and can thus be denoted by $c = (a \oplus b)^{-1} a$ (Mohri, 2009, 238).

Remark 2.9. All semirings in Table 1 are zero-sum-free, weakly divisible, and cancellative.

[3] See Mohri (2002) for general conditions under which the shortest path can be found in polynomial time.

Remark 2.10. For every non-deterministic, acyclic WFSA (or NFA) over a zero-sum-free, weakly divisible and cancellative semiring, there exists an equivalent deterministic WFSA (or DFA). However, a DFA may be exponentially larger than an equivalent NFA (Hopcroft et al., 2008, §2.3.6).

We now provide a brief presentation of the determinization algorithm for WFSAs. Proofs can be found in Mohri 1997. Given an WFSA $A = (Q, s, \Sigma, \omega, \delta)$ over a zero-sum-free, weakly divisible and cancellative semiring $(\mathbb{K}, \oplus, \otimes, \bar{0}, \bar{1})$, its equivalent DFA can be defined and constructed as the DFA $A_d = (Q_d, s_d, \Sigma, \omega_d, \delta_d)$ where Q_d is a finite set whose elements are subsets of $Q \times \mathbb{K}$, recursively defined as follows:

1. $s_d = \{(s, \bar{1})\} \in Q_d$.
2. $\kappa_d \subseteq Q_d \times \Sigma \times \mathbb{K}$ is the *weight transition function*, defined as
$$\kappa_d(q, z) = \bigoplus_{(q_i,k_i)\in q} k_i \otimes \left(\bigoplus_{(q_i,z,k_j,r_j)\in\delta} k_j \right).$$
3. $\nu_d \subseteq Q_d \times \Sigma \times Q_d$ is the *next-state transition function*, defined as $\nu_d(q, z) =$
$$\bigcup_{\substack{(q_i, k_i) \in q \\ (q_i, z, k_j, r_j) \in \delta}} \{(r_j, \kappa_d(q, z)^{-1} l_j)\}$$
where $l_j = \bigoplus_{(q_i,z,k_j,r_j)\in\delta} k_i \otimes k_j$.
4. $Q_d = \nu_d^*(s_d, \Sigma)$ defines the set of states as the closure of the next-state transition function.

The transition relation is then defined as

$$\delta_d = \{(q, z, \kappa_d(q, z), \nu_q(q, z)) | (q, z) \in Q_d \times \Sigma\}$$

and the final weight function $\omega_d \subseteq Q_d \times \mathbb{K}$ as

$$\omega_d(q) = \bigoplus_{(q_i,k_i)\in q} k_i \otimes \omega(q_i).$$

The intuition underlying this construction is that a state $q \in Q_d$ encodes a set of states in Q that can be reached from s by some common strings. More precisely, let p' be the unique path in $P_{s_d \to q}$ labeled by some $z' \in \Sigma^*$, then for any $(q_i, k_i) \in q$:

$$k[p'] \otimes k_i = \bigoplus_{p\in P_{s\to q_i}:z[p]=z'} k[p].$$

Termination is guaranteed for acyclic WFSAs (Mohri, 1997).

Figure 1 gives an example of an NFA and an equivalent DFA. States 0 and 1 in the DFA correspond respectively to the subsets $(0, \bar{1})$ and $(1, \bar{1})$ and $\kappa_d(0, a) = k \otimes k$.

Remark 2.11. Given a NFA A with backwards shortest distance β, the backwards shortest distance β_d over the equivalent DFA A_d can be computed from β:

$$\beta_d(q) = \bigoplus_{(q_i,k_i)\in q} k_i \otimes \beta(q_i)$$

for any $q \in Q_d$ (Mohri and Riley, 2002).

If β has already been computed, $\beta_d(q)$ can be computed in linear time in $|q| \leq |Q|$ for any $q \in Q_d$. This computation can be computed on-demand ("on-the-fly") as soon as the existence of $q \in Q_d$ is known, without requiring A_d to be fully constructed.

2.6 Shortest string

Definition 2.26. Let P_z be a set of paths with string $z \in \Sigma^*$, and let the weight of P_z be

$$\sigma(z) = \bigoplus_{p\in P_z} \bar{k}[p].$$

Definition 2.27. A *shortest string* z is one such that $\forall z' \in \Sigma^*, \sigma(z) \preceq \sigma(z')$.

Lemma 2.1. In an idempotent semiring, a shortest path's string is also a shortest string.

Proof. Let p be a shortest path. By definition, $\bar{k}[p] \preceq \bar{k}[p']$ for all complete paths p'. It follows that $\forall z' \in \Sigma^*$

$$\sigma(z[p]) = \bigoplus_{p\in P_z} \bar{k}[p] \preceq \sigma(z'[p'])$$
$$= \bigoplus_{p'\in P_z} \bar{k}[p']$$

so $z[p]$ is the shortest string. □

Lemma 2.2. In a DFA over a monotonic semiring, a shortest string is the string of a shortest path in that DFA viewed as an WFSA over the corresponding companion semiring.

Proof. Determinism implies that for all complete path p', $\bar{k}[p'] = \sigma(z[p'])$. Let z be the shortest string in the DFA and p the unique path admitting the string z. Then

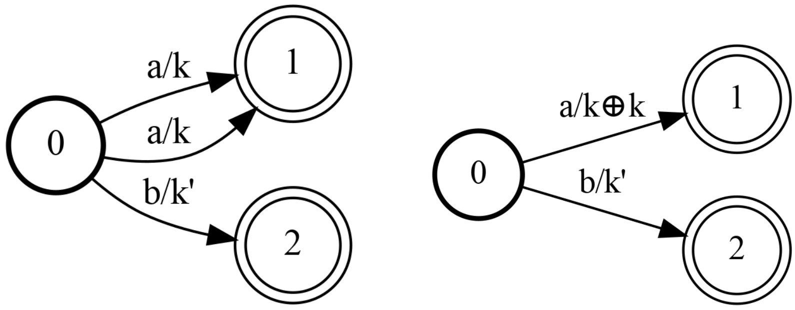

Figure 1: State diagrams showing a weighted NFA (left) and an equivalent DFA (right).

$$\bar{k}[p] = \sigma(z) \preceq \sigma(z[p']) = \bar{k}[p']$$

for any complete path p'. Hence

$$\bar{k}[p] = \widehat{\bigoplus}_{p' \in P_{s \to F}} \bar{k}[p'].$$

Thus p is a shortest path in the DFA viewed over the companion semiring. $\square$

2.7 A* search

A* search (Hart et al., 1968) is a common *shortest-first* search strategy for computing the shortest path in a WFSA over an idempotent semiring. It can be thought of as a variant of Dijkstra's (1959) algorithm, in which exploration is guided by a shortest-first priority queue discipline. At every iteration, the algorithm explores the state q which minimizes $\alpha(q)$, the shortest distance from the initial state s to q, until all states have been visited. In A* search, priority is instead a function of $\digamma \subseteq Q \times \mathbb{K}$, known as the *heuristic*, which gives an estimate of the weight of paths from some state to a final state. At every iteration, A* instead explores the state q which minimizes $\alpha(q) \otimes \digamma(q)$.[4]

Definition 2.28. An A* heuristic is *admissible* if it never overestimates the shortest distance to a state. That is, it is admissible if $\forall q \in Q : \digamma(q) \preceq \beta(q)$.

Definition 2.29. An A* heuristic is *consistent* if it never overestimates the cost of reaching a successor state. That is, it is consistent if $\forall q, r \in Q$ such that $\digamma(q) \preceq k \otimes \digamma(r)$ if $(q, z, k, r) \in \delta$, i.e., if there is a transition from q to r with some label z and weight k.

Remark 2.12. If $\digamma$ is *admissible* and *consistent*, A* search is guaranteed to find a shortest path (if one exists) after visiting all states such that $\digamma[q] \preceq \beta[s]$ (Hart et al., 1968, 104f.).

[4] One can view Dijkstra's algorithm as a special case of A* search with the uninformative heuristic $\digamma = \bar{1}$.

3 The algorithm

Consider an acyclic, ϵ-free WFSA over a monotonic negative semiring $(\mathbb{K}, \oplus, \otimes, \bar{0}, \bar{1})$ with total order $\preceq$ for which we wish to find the shortest string. The same WFSA can also be viewed as a WFSA over the corresponding companion semiring $(\mathbb{K}, \widehat{\oplus}, \otimes, \bar{0}, \bar{1})$, and we denote by $\widehat{\beta}$ the backward shortest-distance over this companion semiring. We prove two theorems, and then introduce an algorithm for search.

Theorem 3.1. The backwards shortest distance of an WFSA over a monotonic negative semiring is an admissible heuristic for the A* search over its companion semiring.

Proof. In a monotonic negative semiring, the $\oplus$-sum of any n terms is upper-bounded by each of the n terms and hence by the $\widehat{\oplus}$-sum of these n terms. It follows that

$$\begin{aligned}
\digamma(q) &= \beta(q) \\
&= \bigoplus_{p \in P_{q \to F}} \bar{k}[p] \preceq \widehat{\bigoplus}_{p \in P_{q \to F}} \bar{k}[p] \\
&= \widehat{\beta}(q),
\end{aligned}$$

and this shows that $\digamma = \beta$ is an admissible heuristic for $\widehat{\beta}$. $\square$

Theorem 3.2. The backwards shortest distance of an WFSA over a monotonic negative semiring is a consistent heuristic for the A* search over its companion semiring.

Proof. Let (q, z, k, r) be a transition in δ. Leveraging again the property that an $\oplus$-sum of any n

terms is upper-bounded by any of these terms, we show that

$$
\begin{aligned}
F(q) &= \beta(q) \\
&= \bigoplus_{p \in P_{q \to F}} \bar{k}[p] \\
&= \bigoplus_{(q,z',k',r') \in \delta} k' \otimes \beta(r') \preceq k \otimes \beta(r) \\
&= k \otimes F(r)
\end{aligned}
$$

showing $F = \beta$ is a consistent heuristic. □

Having established that this is an admissible and consistent heuristic for A* search over the companion semiring, a naïve algorithm then suggests itself, following Lemma 2.2 and Remark 2.12. Given a non-deterministic WFSA over the monotonic negative semiring $(\mathbb{K}, \oplus, \otimes, \bar{0}, \bar{1})$, apply determinization to obtain an equivalent DFA, compute β_d, the backwards shortest distance over the resulting DFA over $(\mathbb{K}, \oplus, \otimes, \bar{0}, \bar{1})$ and then perform A* search over the companion semiring using β_d as the heuristic. However, as mentioned in Remark 2.10 above, determinization has an exponential worse-case complexity in time and space and is often prohibitive in practice. Yet determinization—and the computation of elements of β_d—only need to be performed for states actually visited by A* search. Let β_n denote backwards shortest distance over a non-deterministic WFSA over the monotonic negative semiring $(\mathbb{K}, \oplus, \otimes, \bar{0}, \bar{1})$. Then, the algorithm is as follows:

1. Compute β_n over $(\mathbb{K}, \oplus, \otimes, \bar{0}, \bar{1})$.
2. Lazily determinize the WFSA, lazily computing β_d from β_n over $(\mathbb{K}, \oplus, \otimes, \bar{0}, \bar{1})$.
3. Perform A* search for the shortest string over $(\mathbb{K}, \widehat{\oplus}, \otimes, \bar{0}, \bar{1})$ with β_d as the heuristic.

4 Evaluation

We evaluate the proposed algorithm using non-idempotent speech recognition lattices.

4.1 Data

We search for the shortest string in a sample of 700 word lattices derived from Google Voice Search traffic. This data set was previously used by Mohri and Riley (2015) and Gorman and Sproat (2021, ch. 4) for evaluating related WFSA algorithms. Each path in these lattices is a single hypothesis transcription produced by a production-grade automatic speech recognizer, here treated as a black box. The exact size of each input lattice size is determined by a probability threshold, so paths with probabilities below a certain threshold have been pruned. These lattices are acyclic, ϵ-free, non-deterministic WFSAs over the log semiring, a monotonic non-idempotent semiring.

4.2 Implementation

The above algorithm is implemented as part of an open-source C++17 library released under the Apache-2.0 license.[5] This toolkit includes a command-line tool which implements the above algorithm over the log semiring, using the tropical semiring as a companion semiring. This implementation depends in turn on implementations of determinization, shortest distance, and shortest path algorithms provided by OpenFst (Allauzen et al., 2007). This command-line tool, along with various OpenFst command-line utilities, were used to conduct the following experiment.

4.3 Method

We compute the number of states in the non-deterministic (NFA) lattice, the number of states in an equivalent DFA lattice—created by applying weighted determinization as implemented by OpenFst's `fstdeterminize` command—and the number of DFA states visited during A* search.

4.4 Results

Results are shown in Figure 2. The relationship between the size of the NFA and the number of DFA states visited by the proposed algorithm is roughly monomial (i.e., log-log linear).

The experiment was repeated by performing weighted determinization on each of the NFA lattices beforehand, which produces roughly a factor-of-seven increase in the size of the lattice. Since the increase in state size associated with full determinization is itself roughly linear, a monomial relation also holds between the size of the DFA and the number of DFA states visited by A* search. From these results we infer that the heuristic substantially reduces the number of DFA states visited, and thus the degree of determinization required, compared to the naïve algorithm.

[5] https://redacted.org

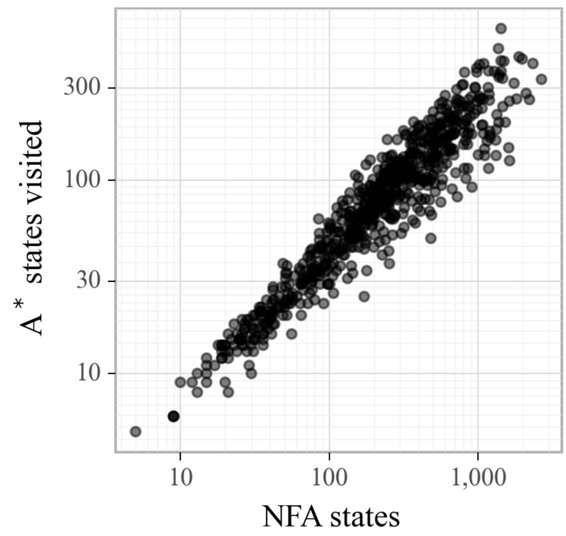

Figure 2: Word lattice decoding with the proposed algorithm. The x-axis shows the number of states in each word lattice NFA; the y-axis shows the number of states visited by A* decoding. Note that logarithmic scale is used for both axes.

5 Related work

Several prior studies use A* search for decoding speech lattices over idempotent semirings. For example, Mohri and Riley (2002) describe a related algorithm for finding the n-best strings over an idempotent WFSA. Like the algorithm proposed here, they use A* search and on-the-fly determinization; however, they do not consider decoding over non-idempotent semirings.

6 Conclusions

We propose an algorithm which allows for efficient shortest string decoding of weighted automata over non-idempotent semirings using A* search and on-the-fly determinization. We find that A* search results in a substantial reduction in the number of DFA states visited during decoding, which in turn minimizes the degree of determinization required to find the shortest path.

We envision several possible applications for the proposed algorithm. It could be used to exactly decode noisy channel "decipherment" models (e.g., Knight et al., 2006) of the form

$$\hat{P}(p \mid c) \propto P(p)P(c \mid p)$$

estimated with expectation maximization, as well as training scenarios which mix ordinary and Viterbi EM (e.g., Spitkovsky et al., 2011).

The decoding algorithm could also be used for exact decoding of lattices scored with interpolated language models (e.g., Jelinek and Mercer, 1980) of the form

$$\hat{P}(w \mid h) = \lambda_h \tilde{P}(w \mid h) + (1 - \lambda_h)\hat{P}(w \mid h')$$

where λ_h is estimated using ordinary EM.

7 Limitations

While the evaluation (§4) finds the proposed algorithm to be substantially more efficient than the naïve algorithm on real-world data, it has the same exponential worst-case complexity as determinization of acyclic WFSAs. We conjecture worst cases are unlikely to arise for topologies encountered in speech and language processing applications.

8 Broader impacts

We are aware of no ethical issues raised by the proposed algorithm beyond issues of dual use, bias, etc., which are inherent to all known speech and language technologies.

References

Cyril Allauzen, Michael Riley, Johan. Schalkwyk, Wojciech Skut, and Mehryar Mohri. 2007. OpenFst: a general and efficient weighted finite-state transducer library. In *Implementation and Application of Automata: 12th International Conference (CIAA 2007)*, pages 11–23.

Peter F. Brown, Vincent J. Della Pietra, Stephen A. Della Pietra, and Robert L. Mercer. 1993. The mathematics of statistical machine translation: parameter estimation. *Computational Linguistics*, 19(2):263–312.

Arthur Dempster, Nan Laird, and Donald Rubin. 1977. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society*, 39(1):1–38.

Edsger W. Dijkstra. 1959. A note on two problems in connexion with graphs. *Numerische Mathematik*, 1(1):269–271.

Kyle Gorman and Richard Sproat. 2021. *Finite-State Text Processing*. Morgan & Claypool.

Peter E. Hart, Nils J. Nilsson, and Bertram Raphael. 1968. A formal basis for the heuristic determination of minimal cost paths. *IEEE Transactions on Systems Science and Cybernetics*, 4(2):100–107.

John E. Hopcroft, Rajeev Motwani, and Jeffrey D. Ullman. 2008. *Introduction to Automata Theory, Languages, and Computation*, 3rd edition. Pearson.

Frederick Jelinek and Robert L. Mercer. 1980. Interpolated estimation of Markov source parameters from sparse data. In *Proceedings of the Workshop on Pattern Recognition in Practice*, pages 381–397.

Kevin Knight, Anish Nair, Nishit Rathod, and Kenji Yamada. 2006. Unsupervised analysis for decipherment problems. In *Proceedings of the COLING/ACL 2006 Main Conference Poster Sessions*, pages 499–506.

Mehryar Mohri. 1997. Finite-state transducers in language and speech processing. *Computational Linguistics*, 23(2):269–311.

Mehryar Mohri. 2002. Semiring frameworks and algorithms for shortest-distance problems. *Journal of Automata, Languages and Combinatorics*, 7(3):321–350.

Mehryar Mohri. 2009. Weighted automata algorithms. In Manfred Droste, Werner Kuich, and Heiko Vogler, editors, *Handbook of Weighted Automata*, pages 213–254. Springer.

Mehryar Mohri, Fernando Pereira, and Michael Riley. 2002. Weighted finite-state transducers in speech recognition. *Computer Speech and Language*, 16(1):69–88.

Mehryar Mohri and Michael D. Riley. 2015. On the disambiguation of weighted automata. In *Implementation and Application of Automata 20th International Conference (CIAA 2015)*, pages 263–278.

Mehyar Mohri and Michael Riley. 2002. An efficient algorithm for the n-best-strings problem. In *7th International Conference on Spoken Language Processing*, pages 1313–1316.

Brian Roark and Richard Sproat. 2007. *Computational Approaches to Morphology and Syntax*. Cambridge University Press.

Valentin I. Spitkovsky, Hiyan Alshawi, and Daniel Jurafsky. 2011. Lateen EM: unsupervised training with multiple objectives, applied to dependency grammar induction. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 1269–1280.