Lattice-Based Minimum Error Rate Training using Weighted Finite-State Transducers with Tropical Polynomial Weights

Aurelien Waite[‡] **Graeme Blackwood**[*] **William Byrne**[‡]
[‡] Department of Engineering, University of Cambridge, Trumpington Street, CB2 1PZ, U.K.
{aaw35|wjb31}@cam.ac.uk
[*] IBM T.J. Watson Research, Yorktown Heights, NY-10598
blackwood@us.ibm.com

Abstract

Minimum Error Rate Training (MERT) is a method for training the parameters of a log-linear model. One advantage of this method of training is that it can use the large number of hypotheses encoded in a translation lattice as training data. We demonstrate that the MERT line optimisation can be modelled as computing the shortest distance in a weighted finite-state transducer using a tropical polynomial semiring.

1 Introduction

Minimum Error Rate Training (MERT) (Och, 2003) is an iterative procedure for training a log-linear statistical machine translation (SMT) model (Och and Ney, 2002). MERT optimises model parameters directly against a criterion based on an automated translation quality metric, such as BLEU (Papineni et al., 2002). Koehn (2010) provides a full description of the SMT task and MERT.

MERT uses a line optimisation procedure (Press et al., 2002) to identify a range of points along a line in parameter space that maximise an objective function based on the BLEU score. A key property of the line optimisation is that it can consider a large set of hypotheses encoded as a weighted directed acyclic graph (Macherey et al., 2008), which is called a lattice. The line optimisation procedure can also be applied to a hypergraph representation of the hypotheses (Kumar et al., 2009).

It has been noted that line optimisation over a lattice can be implemented as a semiring of sets of linear functions (Dyer et al., 2010). Sokolov and Yvon (2011) provide a formal description of such a semiring, which they denote the MERT semiring. The difference between the various algorithms derives from the differences in their formulation and implementation, but not in the objective they attempt to optimise.

Instead of an algebra defined in terms of transformations of sets of linear functions, we propose an alternative formulation using the tropical polynomial semiring (Speyer and Sturmfels, 2009). This semiring provides a concise formalism for describing line optimisation, an intuitive explanation of the MERT shortest distance, and draws on techniques in the currently active field of Tropical Geometry (Richter-Gebert et al., 2005) [1].

We begin with a review of the line optimisation procedure, lattice-based MERT, and the weighted finite-state transducer formulation in Section 2. In Section 3, we introduce our novel formulation of lattice-based MERT using tropical polynomial weights. Section 4 compares the performance of our approach with k-best and lattice-based MERT.

2 Minimum Error Rate Training

Following Och and Ney (2002), we assume that we are given a tuning set of parallel sentences $\{(\mathbf{r}_1, \mathbf{f}_1), ..., (\mathbf{r}_S, \mathbf{f}_S)\}$, where $\mathbf{r}_s$ is the reference translation of the source sentence $\mathbf{f}_s$. We also assume that sets of hypotheses $\mathbf{C}_s = \{\mathbf{e}_{s,1}, ..., \mathbf{e}_{s,K}\}$

[*]The work reported in this paper was carried out while the author was at the University of Cambridge.

[1]An associated technical report contains an extended discussion of our approach (Waite et al., 2011)

Proceedings of the 10th International Workshop on Finite State Methods and Natural Language Processing, Donostia - San Sebastian (Spain), July 23-25, 2012.

are available for each source sentence $\mathbf{f}_s$.

Under the log-linear model formulation with feature functions h_1^M and model parameters λ_1^M, the most probable translation in a set $\mathbf{C}_s$ is selected as

$$\hat{\mathbf{e}}(\mathbf{f}_s; \lambda_1^M) = \underset{\mathbf{e} \in \mathbf{C}_s}{\operatorname{argmax}} \left\{ \sum_{m=1}^{M} \lambda_m h_m(\mathbf{e}, \mathbf{f_s}) \right\}. \quad (1)$$

With an error function of the form $E(\mathbf{r}_1^S, \mathbf{e}_1^S) = \sum_{s=1}^{S} E(\mathbf{r}_s, \mathbf{e}_s)$, MERT attempts to find model parameters to minimise the following objective:

$$\hat{\lambda}_1^M = \underset{\lambda_1^M}{\operatorname{argmin}} \left\{ \sum_{s=1}^{S} E(\mathbf{r}_s, \hat{\mathbf{e}}(\mathbf{f}_s; \lambda_1^M)) \right\}. \quad (2)$$

Note that for MERT the hypotheses set $\mathbf{C}_s$ is a k-best list of explicitly enumerated hypotheses, whereas lattice-based MERT uses a larger space.

2.1 Line Optimisation

Although the objective function in Eq. (2) cannot be solved analytically, the line optimisation procedure of Och (2003) can be used to find an approximation of the optimal model parameters. Rather than evaluating the decision rule in Eq. (1) over all possible points in parameter space, the line optimisation considers a subset of points defined by the line $\lambda_1^M + \gamma d_1^M$, where λ_1^M corresponds to an initial point in parameter space and d_1^M is the direction along which to optimise. Eq. (1) can be rewritten as:

$$\begin{aligned}
\hat{\mathbf{e}}(\mathbf{f}_s; \gamma) &= \underset{\mathbf{e} \in \mathbf{C}_s}{\operatorname{argmax}} \left\{ (\lambda_1^M + \gamma d_1^M)^T h_1^M(\mathbf{e}, \mathbf{f}_s) \right\} \\
&= \underset{\mathbf{e} \in \mathbf{C}_s}{\operatorname{argmax}} \left\{ \underbrace{\sum_m \lambda_m h_m(\mathbf{e}, \mathbf{f}_s)}_{a(\mathbf{e}, \mathbf{f}_s)} + \gamma \underbrace{\sum_m d_m h_m(\mathbf{e}, \mathbf{f}_s)}_{b(\mathbf{e}, \mathbf{f}_s)} \right\} \\
&= \underset{\mathbf{e} \in \mathbf{C}_s}{\operatorname{argmax}} \{ \underbrace{a(\mathbf{e}, \mathbf{f}_s) + \gamma b(\mathbf{e}, \mathbf{f}_s)}_{\ell_{\mathbf{e}}(\gamma)} \} \quad (3)
\end{aligned}$$

This decision rule shows that each hypothesis $\mathbf{e} \in \mathbf{C}_s$ is associated with a linear function of γ: $\ell_{\mathbf{e}}(\gamma) = a(\mathbf{e}, \mathbf{f}_s) + \gamma b(\mathbf{e}, \mathbf{f}_s)$, where $a(\mathbf{e}, \mathbf{f}_s)$ is the y-intercept and $b(\mathbf{e}, \mathbf{f}_s)$ is the gradient. The optimisation problem is further simplified by defining a subspace over which optimisation is performed. The subspace is found by considering a form of the function in Eq. (3) defined with a range of real numbers (Macherey et al., 2008; Och, 2003):

$$\mathrm{Env}(\mathbf{f}) = \max_{\mathbf{e} \in \mathbf{C}} \{ \underbrace{a(\mathbf{e}, \mathbf{f}) + \gamma b(\mathbf{e}, \mathbf{f})}_{\ell_{\mathbf{e}}(\gamma)} : \gamma \in \mathbb{R} \} \quad (4)$$

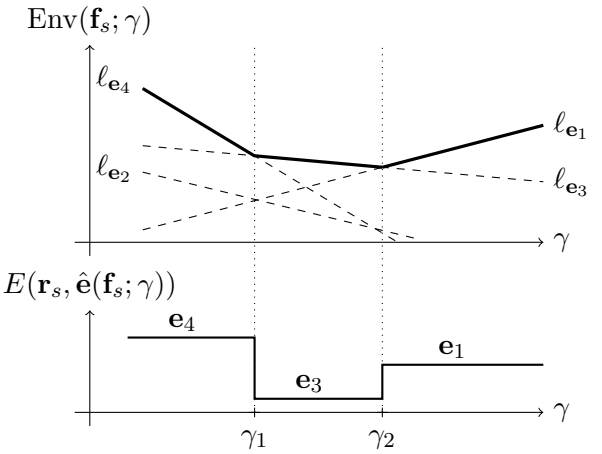

Figure 1: An upper envelope and projected error. Note that the upper envelope is completely defined by hypotheses $\mathbf{e}_4$, $\mathbf{e}_3$, and $\mathbf{e}_1$, together with the intersection points γ_1 and γ_2 (after Macherey et al. (2008), Fig. 1).

For any value of γ the linear functions $\ell_{\mathbf{e}}(\gamma)$ associated with $\mathbf{C}_s$ take (up to) K values. The function in Eq. (4) defines the 'upper envelope' of these values over all γ. The upper envelope has the form of a continuous piecewise linear function in γ. The piecewise linear function can be compactly described by the linear functions which form line segments and the values of γ at which they intersect. The example in the upper part of Figure 1 shows how the upper envelope associated with a set of four hypotheses can be represented by three associated linear functions and two values of γ. The first step of line optimisation is to compute this compact representation of the upper envelope.

Macherey et al. (2008) use methods from computational geometry to compute the upper envelope. The SweepLine algorithm (Bentley and Ottmann, 1979) computes the upper envelope from a set of linear functions with a complexity of $\mathcal{O}(K \log(K))$.

Computing the upper envelope reduces the runtime cost of line optimisation as the error function need only be evaluated for the subset of hypotheses in $\mathbf{C}_s$ that contribute to the upper envelope. These errors are projected onto intervals of γ, as shown in the lower part of Figure 1, so that Eq. (2) can be readily solved.

2.2 Incorporation of Line Optimisation into MERT

The previous algorithm finds the upper envelope along a particular direction in parameter space over

a hypothesis set $\mathbf{C_s}$. The line optimisation algorithm is then embedded within a general optimisation procedure. A common approach to MERT is to select the directions using Powell's method (Press et al., 2002). A line optimisation is performed on each coordinate axis. The axis giving the largest decrease in error is replaced with a vector between the initial parameters and the optimised parameters. Powell's method halts when there is no decrease in error.

Instead of using Powell's method, the Downhill Simplex algorithm (Press et al., 2002) can be used to explore the criterion in Eq. (2). This is done by defining a simplex in parameter space. Directions where the error count decreases can be identified by considering the change in error count at the points of the simplex. This has been applied to parameter searching over k-best lists (Zens et al., 2007).

Both Powell's method and the Downhill Simplex algorithms are approaches based on heuristics to select lines $\lambda_1^M + \gamma d_1^M$. It is difficult to find theoretically sound reasons why one approach is superior. Therefore Cer et al. (2008) instead choose the direction vectors d_1^M at random. They report that this method can find parameters that are as good as the parameters produced by more complex algorithms.

2.3 Lattice Line Optimisation

Macherey et al. (2008) describe a procedure for conducting line optimisation directly over a word lattice encoding the hypotheses in $\mathbf{C_s}$. Each lattice edge is labelled with a word e and has a weight defined by the vector of word specific feature function values $h_1^M(e, \mathbf{f})$ so that the weight of a path in the lattice is found by summing over the word specific feature function values on that path. Given a line through parameter space, the goal is to extract from a lattice its upper envelope and the associated hypotheses.

Their algorithm proceeds node by node through the lattice. Suppose that for a state q the upper envelope is known for all the partial hypotheses on all paths leading to q. The upper envelope defines a set of functions $\{\ell_{\tilde{\mathbf{e}}_1}(\gamma), ..., \ell_{\tilde{\mathbf{e}}_N}(\gamma)\}$ over the partial hypotheses $\tilde{\mathbf{e}}_n$. Two operations propagate the upper envelope to other lattice nodes.

We refer to the first operation as the 'extend' operation. Consider a single edge from state q to state q'. This edge defines a linear function associated with a single word $\ell_e(\gamma)$. A path following this edge transforms all the partial hypotheses leading to q by concatenating the word e. The upper envelope associated with the edge from q to q' is changed by adding $\ell_e(\gamma)$ to the set of linear functions. The intersection points are not changed by this operation.

The second operation is a union. Suppose q' has another incoming edge from a state q'' where $q \neq q''$. There are now two upper envelopes representing two sets of linear functions. The first upper envelope is associated with the paths from the initial state to state q' via the state q. Similarly the second upper envelope is associated with paths from the initial state to state q' via the state q''. The upper envelope that is associated with all paths from the initial state to state q' via both q and q'' is the union of the two sets of linear functions. This union is no longer a compact representation of the upper envelope as there may be functions which never achieve a maximum for any value of γ. The SweepLine algorithm (Bentley and Ottmann, 1979) is applied to the union to discard redundant linear functions and their associated hypotheses (Macherey et al., 2008).

The union and extend operations are applied to states in topological order until the final state is reached. The upper envelope computed at the final state compactly encodes all the hypotheses that maximise Eq. (1) along the line $\lambda_1^M + \gamma d_1^M$. Macherey's theorem (Macherey et al., 2008) states that an upper bound for the number of linear functions in the upper envelope at the final state is equal to the number of edges in the lattice.

2.4 Line Optimisation using WFSTs

Formally, a weighted finite-state transducer (WFST) $T = (\Sigma, \Delta, Q, I, F, E, \lambda, \rho)$ over a semiring $(\mathbb{K}, \oplus, \otimes, \bar{0}, \bar{1})$ is defined by an input alphabet Σ, an output alphabet Δ, a set of states Q, a set of initial states $I \subseteq Q$, a set of final states $F \subseteq Q$, a set of weighted transitions E, an initial state weight assignment $\lambda : I \rightarrow \mathbb{K}$, and a final state weight assignment $\rho : F \rightarrow \mathbb{K}$ (Mohri et al., 2008). The weighted transitions of T form the set $E \subseteq Q \times \Sigma \times \Delta \times \mathbb{K} \times Q$, where each transition includes a source state from Q, input symbol from Σ, output symbol from Δ, cost from the weight set $\mathbb{K}$, and target state from Q.

For each state $q \in Q$, let $E[q]$ denote the set of edges leaving state q. For each transition $e \in E[q]$, let $p[e]$ denote its source state, $n[e]$ its target state,

and $w[e]$ its weight. Let $\pi = e_1 \cdots e_K$ denote a path in T from state $p[e_1]$ to state $n[e_K]$, so that $n[e_{k-1}] = p[e_k]$ for $k = 2, \ldots, K$. The weight associated by T to path π is the generalised product $\otimes$ of the weights of the individual transitions:

$$w[\pi] = \bigotimes_{k=1}^{K} w[e_k] = w[e_1] \otimes \cdots \otimes w[e_K] \quad (5)$$

If $\mathcal{P}(q)$ denotes the set of all paths in T starting from an initial state in I and ending in state q, then the shortest distance $d[q]$ is defined as the generalised sum $\oplus$ of the weights of all paths leading to q (Mohri, 2002):

$$d[q] = \oplus_{\pi \in \mathcal{P}(q)} w[\pi] \quad (6)$$

For some semirings, such as the tropical semiring, the shortest distance is the weight of the shortest path. For other semirings, the shortest distance is associated with multiple paths (Mohri, 2002); for these semirings there are shortest distances but need not any be shortest paths. That will be the case in what follows. However, the shortest distance algorithms rely only on general properties of semirings, and once the semiring is specified, the general shortest distance algorithms can be directly employed.

Sokolov and Yvon (2011) define the MERT semiring based on operations described in the previous section. The extend operation is used for the generalised product $\otimes$. The union operation followed by an application of the SweepLine algorithm becomes the generalised sum $\oplus$. The word lattice is then transformed for an initial parameter λ_1^M and direction d_1^M. The weight of edge is mapped from a word specific feature function $h_1^M(e, \mathbf{f})$ to a word specific linear function $\ell_e(\gamma)$. The weight of each path is the generalised product $\otimes$ of the word specific feature linear functions. The upper envelope is the shortest distance of all the paths in the WFST.

3 The Tropical Polynomial Semiring

In this section we introduce the tropical polynomial semiring (Speyer and Sturmfels, 2009) as a replacement for the MERT semiring (Sokolov and Yvon, 2011). We then provide a full description and a worked example of our MERT algorithm.

3.1 Tropical Polynomials

A polynomial is a linear combination of a finite number of non-zero monomials. A monomial consists of a real valued coefficient multiplied by one or more variables, and these variables may have exponents that are non-negative integers. In this section we limit ourselves to a description of a polynomial in a single variable. A polynomial function is defined by evaluating a polynomial:

$$f(\gamma) = a_n\gamma^n + a_{n-1}\gamma^{n-1} + \cdots + a_2\gamma^2 + a_1\gamma + a_0$$

A useful property of these polynomials is that they form a ring[2] (Cox et al., 2007) and therefore are candidates for use as weights in WFSTs.

Speyer and Sturmfels (2009) apply the definition of a classical polynomial to the formulation of a tropical polynomial. The tropical semiring uses summation for the generalised product $\otimes$ and a $\min$ operation for the generalised sum $\oplus$. In this form, let γ be a variable that represents an element in the tropical semiring weight set $\mathbb{R} \cup \{-\infty, +\infty\}$. We can write a monomial of γ raised to an integer exponent as

$$\gamma^i = \underbrace{\gamma \otimes \cdots \otimes \gamma}_{i}$$

where i is a non-negative integer. The monomial can also have a constant coefficient: $a \otimes \gamma^i$, $a \in \mathbb{R}$. We can define a function that evaluates a tropical monomial for a particular value of γ. For example, the tropical monomial $a \otimes \gamma^i$ is evaluated as:

$$f(\gamma) = a \otimes \gamma^i = a + i\gamma$$

This shows that a tropical monomial is a linear function with the coefficient a as its y-intercept and the integer exponent i as its gradient. A tropical polynomial is the generalised sum of tropical monomials where the generalised sum is evaluated using the min operation. For example:

$$f(\gamma) = (a \otimes \gamma^i) \oplus (b \otimes \gamma^j) = \min(a + i\gamma, b + j\gamma)$$

Evaluating tropical polynomials in classical arithmetic gives the minimum of a finite collection of linear functions.

Tropical polynomials can also be multiplied by a monomial to form another tropical polynomial. For example:

$$\begin{aligned} f(\gamma) &= [(a \otimes \gamma^i) \oplus (b \otimes \gamma^j)] \otimes (c \otimes \gamma^k) \\ &= [(a + c) \otimes \gamma^{i+k}] \oplus [(b + c) \otimes \gamma^{j+k}] \\ &= \min((a + c) + (i + k)\gamma, (b + c) + (j + k)\gamma) \end{aligned}$$

Our re-formulation of Eq. (4) negates the feature

[2] A ring is a semiring that includes negation.

function weights and replaces the argmax by an argmin. This allows us to keep the usual formulation of tropical polynomials in terms of the min operation when converting Eq. (4) to a tropical representation. What remains to be addressed is the role of integer exponents in the tropical polynomial.

3.2 Integer Realisations for Tropical Monomials

In the previous section we noted that the function defined by the upper envelope in Eq. (4) is similar to the function represented by a tropical polynomial. A significant difference is that the formal definition of a polynomial only allows integer exponents, whereas the gradients in Eq. (4) are real numbers. The upper envelope therefore encodes a larger set of model parameters than a tropical polynomial.

To create an equivalence between the upper envelope and tropical polynomials we can approximate the linear functions $\{\ell_{\mathbf{e}}(\gamma) = a(\mathbf{e}, \mathbf{f}_s) + \gamma \cdot b(\mathbf{e}, \mathbf{f}_s)\}$ that compose segments of the upper envelope. We define $\tilde{a}(\mathbf{e}, \mathbf{f}_s) = [a(\mathbf{e}, \mathbf{f}_s) \cdot 10^n]_{\text{int}}$ and $\tilde{b}(\mathbf{e}, \mathbf{f}_s) = [b(\mathbf{e}, \mathbf{f}_s) \cdot 10^n]_{\text{int}}$ where $[x]_{\text{int}}$ denotes the integer part of x. The approximation to $\ell_{\mathbf{e}}(\gamma)$ is:

$$\ell_{\mathbf{e}}(\gamma) \approx \tilde{\ell}_{\mathbf{e}}(\gamma) = \frac{\tilde{a}(\mathbf{e}, \mathbf{f}_s)}{10^n} + \gamma \cdot \frac{\tilde{b}(\mathbf{e}, \mathbf{f}_s)}{10^n} \quad (7)$$

The result of this operation is to approximate the y-intercept and gradient of $\ell_{\mathbf{e}}(\gamma)$ to n decimal places. We can now represent the linear function $\tilde{\ell}_{\mathbf{e}}(\gamma)$ as the tropical monomial $-\tilde{a}(\mathbf{e}, \mathbf{f}_s) \otimes \gamma^{-\tilde{b}(\mathbf{e}, \mathbf{f}_s)}$. Note that $\tilde{a}(\mathbf{e}, \mathbf{f}_s)$ and $\tilde{b}(\mathbf{e}, \mathbf{f}_s)$ are negated since tropical polynomials define the lower envelope as opposed to the upper envelope defined by Eq. (4).

The linear function represented by the tropical monomial is a scaled version of $\ell_{\mathbf{e}}(\gamma)$, but the upper envelope is unchanged (to the accuracy allowed by n). If for a particular value of γ, $\ell_{\mathbf{e}_i}(\gamma) > \ell_{\mathbf{e}_j}(\gamma)$, then $\tilde{\ell}_{\mathbf{e}_i}(\gamma) > \tilde{\ell}_{\mathbf{e}_j}(\gamma)$. Similarly, the boundary points are unchanged: if $\ell_{\mathbf{e}_i}(\gamma) = \ell_{\mathbf{e}_j}(\gamma)$, then $\tilde{\ell}_{\mathbf{e}_i}(\gamma) = \tilde{\ell}_{\mathbf{e}_j}(\gamma)$. Setting n to a very large value removes numerical differences between the upper envelope and the tropical polynomial representation, as shown by the identical results in Table 1.

Using a scaled version of $\ell_{\mathbf{e}}(\gamma)$ as the basis for a tropical monomial may cause negative exponents to be created. Following Speyer and Sturmfels (2009), we widen the definition of a tropical polynomial to allow for these negative exponents.

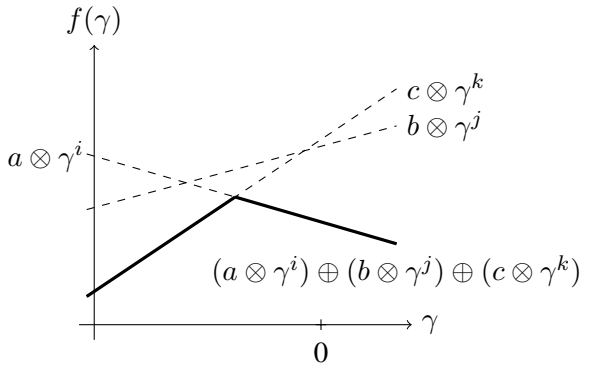

Figure 2: Redundant terms in a tropical polynomial. In this case $(a \otimes \gamma^i) \oplus (b \otimes \gamma^j) \oplus (c \otimes \gamma^k) = (a \otimes \gamma^i) \oplus (c \otimes \gamma^k)$.

3.3 Canonical Form of a Tropical Polynomial

We noted in Section 2.1 that linear functions induced by some hypotheses do not contribute to the upper envelope and can be discarded. Terms in a tropical polynomial can have similar behaviour. Figure 2 plots the lines associated with the three terms of the example polynomial function $f(\gamma) = (a \otimes \gamma^i) \oplus (b \otimes \gamma^j) \oplus (c \otimes \gamma^k)$. We note that the piecewise linear function can also be described with the polynomial $f(\gamma) = (a \otimes \gamma^i) \oplus (c \otimes \gamma^k)$. The latter representation is simpler but equivalent.

Having multiple representations of the same polynomial causes problems when implementing the shortest distance algorithm defined by Mohri (2002). This algorithm performs an equality test between values in the semiring used to weight the WFST. The behaviour of the equality test is ambiguous when there are multiple polynomial representations of the same piecewise linear function. We therefore require a canonical form of a tropical polynomial so that a single polynomial represents a single function. We define the canonical form of a tropical polynomial to be the tropical polynomial that contains only the monomial terms necessary to describe the piecewise linear function it represents.

We remove redundant terms from a tropical polynomial after computing the generalised sum. For a tropical polynomial of one variable we can take advantage of the equivalence with Lattice MERT and compute the canonical form using the SweepLine algorithm (Bentley and Ottmann, 1979). Each term corresponds to a linear function; linear functions

that do not contribute to the upper envelope are discarded. Only monomials which correspond to the remaining linear functions are kept in the canonical form. The canonical form of a tropical polynomial thus corresponds to a unique and minimal representation of the upper envelope.

3.4 Relationship to the Tropical Semiring

Tropical monomial weights can be transformed into regular tropical weights by evaluating the tropical monomial for a specific value of γ. For example, a tropical polynomial evaluated at $\gamma = 1$ corresponds to the tropical weight:

$$f(1) = -\tilde{a}(e, \mathbf{f}_s) \otimes 1^{-\tilde{b}(e,\mathbf{f}_s)} = -\tilde{a}(e, \mathbf{f}_s) - \tilde{b}(e, \mathbf{f}_s)$$

Each monomial term in the tropical polynomial shortest distance represents a linear function. The intersection points of these linear functions define intervals of γ (as in Fig. 1). This suggests an alternate explanation for what the shortest distance computed using the tropical polynomial semiring represents. Conceptually, there is a continuum of lattices which have identical edges and vertices but with varying, real-valued edge weights determined by values of $\gamma \in \mathbb{R}$, so that each lattice in the continuum is indexed by γ. The tropical polynomial shortest distance agrees with the shortest distance through each lattice in the continuum.

Our alternate explanation is consistent with the Theorem of Macherey (Section 2.3), as there could never be more paths than edges in the lattice. Therefore the upper bound for the number of monomial terms in the tropical polynomial shortest distance is the number of edges in the input lattice.

We can use the mapping to the tropical semiring to compute the error surface. Let us assume we have $n + 1$ intervals separated by n interval boundaries. We use the midpoint of each interval to transform the lattice of tropical monomial weights into a lattice of tropical weights. The sequence of words that label the shortest path through the transformed lattice is the MAP hypothesis for the interval. The shortest path can be extracted using the WFST shortest path algorithm (Mohri and Riley, 2002). As a technical matter, the midpoints of the first interval $[-\infty, \gamma_1)$ and last interval $[\gamma_n, \infty)$ are not defined. We therefore evaluate the tropical polynomial at $\gamma = \gamma_1 - 1$ and $\gamma = \gamma_n + 1$ to find the MAP hypothesis in the first and last intervals, respectively.

3.5 The TGMERT Algorithm

We now describe an alternative algorithm to Lattice MERT that is formulated using the tropical polynomial shortest distance in one variable. We call the algorithm TGMERT, for Tropical Geometry MERT. As input to this procedure we use a word lattice weighted with word specific feature functions $h_1^M(e, \mathbf{f})$, a starting point λ_1^M, and a direction d_1^M in parameter space.

1. Convert the word specific feature functions $h_1^M(e, \mathbf{f})$ to a linear function $\ell_e(\gamma)$ using λ_1^M and d_1^M, as in Eq. (3).
2. Convert $\ell_e(\gamma)$ to $\tilde{\ell}_e(\gamma)$ by approximating y-intercepts and gradients to n decimal places, as in Eq. (7).
3. Convert $\tilde{\ell}_e(\gamma)$ in Eq. (7) to the tropical monomial $-\tilde{a}(e, \mathbf{f}_s) \otimes \gamma^{-\tilde{b}(e,\mathbf{f}_s)}$.
4. Compute the WFST shortest distance to the exit states (Mohri, 2002) with generalised sum $\oplus$ and generalised product $\otimes$ defined by the tropical polynomial semiring. The resulting tropical polynomial represents the upper envelope of the lattice.
5. Compute the intersection points of the linear functions corresponding to the monomial terms of the tropical polynomial shortest distance. These intersection points define intervals of γ in which the MAP hypothesis does not change.
6. Using the midpoint of each interval convert the tropical monomial $-\tilde{a}(e, \mathbf{f}_s) \otimes \gamma^{-\tilde{b}(e,\mathbf{f}_s)}$ to a regular tropical weight. Find the MAP hypothesis for this interval by extracting the shortest path using the WFST shortest path algorithm (Mohri and Riley, 2002).

3.6 TGMERT Worked Example

This section presents a worked example showing how we can use the TGMERT algorithm to compute the upper envelope of a lattice. We start with a three state lattice with a two dimensional feature vector shown in the upper part of Figure 3.

We want to optimise the parameters along a line in two-dimensional feature space. Suppose the initial parameters are $\lambda_1^2 = [0.7, 0.4]$ and the direction is $d_1^2 = [0.3, 0.5]$. Step 1 of the TGMERT algorithm

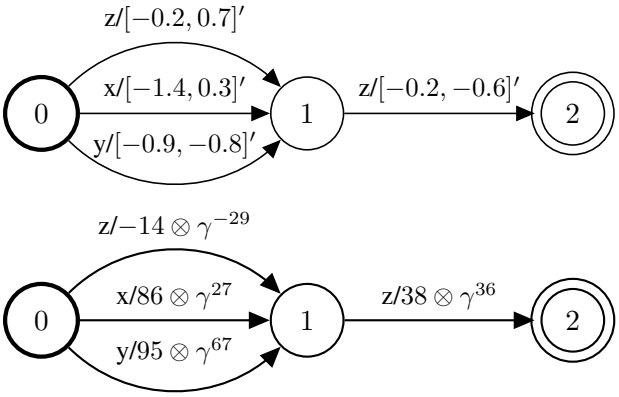

Figure 3: The upper part is a translation lattice with 2-dimensional log feature vector weights $h_1^M(e, \mathbf{f})$ where $M = 2$. The lower part is the lattice from the upper part with weights transformed into tropical monomials.

(Section 3.5) maps each edge weight to a word specific linear function. For example, the weight of the edge labelled "x" between states 0 and 1 is transformed as follows:

$$\ell_e(\gamma) = \underbrace{\sum_{m=1}^{2} \lambda_m h_1^M(e, \mathbf{f})}_{a(e,\mathbf{f})} + \gamma \underbrace{\sum_{m=1}^{2} d_m h_1^M(e, \mathbf{f_s})}_{b(e,\mathbf{f})}$$

$$= \underbrace{0.7 \cdot -1.4 + 0.4 \cdot 0.3}_{a(e,\mathbf{f})} + \gamma \cdot \underbrace{0.3 \cdot -1.4 + 0.5 \cdot 0.3}_{b(e,\mathbf{f})}$$

$$= -0.86 - 0.27\gamma$$

Step 2 of the TGMERT algorithm converts the word specific linear functions into tropical monomial weights. Since all y-intercepts and gradients have a precision of two decimal places, we scale the linear functions $\ell_{\mathbf{e}}(\gamma)$ by 10^2 and negate them to create tropical monomials (Step 3). The edge labelled "x" now has the monomial weight of $86 \otimes \gamma^{27}$. The transformed lattice with weights mapped to the tropical polynomial semiring is shown in the lower part of Figure 3.

We can now compute the shortest distance (Mohri, 2002) from the transformed example lattice with tropical monomial weights. There are three unique paths through the lattice corresponding to three distinct hypotheses. The weights associated with these hypotheses are:

$$-14 \otimes \gamma^{-29} \otimes 38 \otimes \gamma^{36} = 24 \otimes \gamma^{7} \qquad \text{z z}$$

$$86 \otimes \gamma^{27} \otimes 38 \otimes \gamma^{36} = 122 \otimes \gamma^{63} \qquad \text{x z}$$

$$95 \otimes \gamma^{67} \otimes 38 \otimes \gamma^{36} = 133 \otimes \gamma^{103} \qquad \text{y z}$$

The shortest distance from initial to final state is the generalised sum of the path weights: $(24 \otimes \gamma^7) \oplus (133 \otimes \gamma^{103})$. The monomial term $122 \otimes \gamma^{63}$ corresponding to "x z" can be dropped because it is not part of the canonical form of the polynomial (Section 3.3). The shortest distance to the exit state can be represented as the minimum of two linear functions: $\min(24 + 7\gamma, 133 + 103\gamma)$.

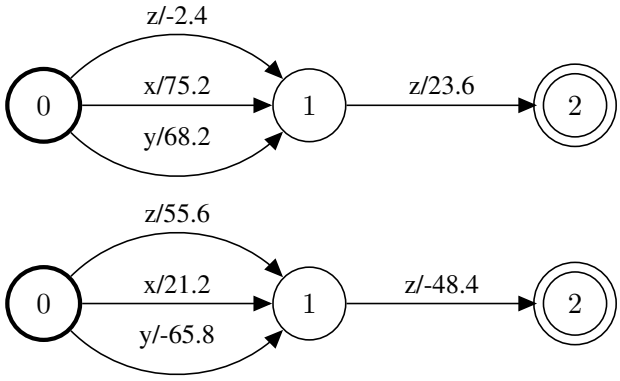

Figure 4: The lattice in the lower part of Figure 3 transformed to regular tropical weights: $\gamma = -0.4$ (top) and $\gamma = -1.4$ (bottom).

We now wish to find the hypotheses that define the error surface by performing Steps 5 and 6 of the TGMERT algorithm. These two linear functions define two intervals of γ. The linear functions intersect at $\gamma \approx -1.4$; at this value of γ the MAP hypothesis changes. Two lattices with regular tropical weights are created using $\gamma = -0.4$ and $\gamma = -2.4$. These are shown in Figure 4. For the lattice shown in the upper part the value for the edge labelled "x" is computed as $86 \otimes -0.4^{27} = 86 + 0.4 \cdot 27 = 75.2$.

When $\gamma = -0.4$ the lattice in the upper part in Figure 4 shows that the shortest path is associated with the hypothesis "z z", which is the MAP hypothesis for the range $\gamma < 1.4$. The lattice in the lower part of Figure 4 shows that when $\gamma = -2.4$ the shortest path is associated with the hypothesis "y z", which is the MAP hypothesis when $\gamma > 1.4$.

3.7 TGMERT Implementation

TGMERT is implemented using the OpenFst Toolkit (Allauzen et al., 2007). A weight class is added for tropical polynomials which maintains them in canonical form. The $\otimes$ and $\oplus$ operations are implemented for piece-wise linear functions, with the SweepLine algorithm included as discussed.

Iteration	Arabic-to-English					
	MERT		LMERT		TGMERT	
	Tune	Test	Tune	Test	Tune	Test
1	36.2 42.1	40.9	36.2 39.7	38.9	36.2 39.7	38.9
2	42.0 45.1	43.2	44.5 45.8	44.3	44.5 45.8	44.3
3	44.5 45.5	44.1				
4	45.6 45.7	44.0				

Iteration	Chinese-to-English					
	MERT		LMERT		TGMERT	
	Tune	Test	Tune	Test	Tune	Test
1	19.5 25.3	16.7	19.5 29.3	22.6	19.5 29.3	22.6
2	16.4 18.9	23.9	22.5 31.4	32.1	22.5 31.4	32.1
3	23.6 28.2	29.1	31.6 32.2	32.5	31.6 32.2	32.5
4	29.2 31.3	31.5	32.2 32.2	32.5	32.2 32.2	32.5
5	31.3 31.8	32.1				
6	32.1 32.4	32.3				
7	32.4 32.4	32.3				

Table 1: GALE AR→EN and ZH→EN BLEU scores by MERT iteration. BLEU scores at the initial and final points of each iteration are shown for the Tune sets.

4 Experiments

We compare feature weight optimisation using k-best MERT (Och, 2003), lattice MERT (Macherey et al., 2008), and tropical geometry MERT. We refer to these as MERT, LMERT, and TGMERT, resp.

We investigate MERT performance in the context of the Arabic-to-English GALE P4 and Chinese-to-English GALE P3 evaluations[3]. For Arabic-to-English translation, word alignments are generated over around 9M sentences of GALE P4 parallel text. Following de Gispert et al. (2010b), word alignments for Chinese-to-English translation are trained from a subset of 2M sentences of GALE P3 parallel text. Hierarchical rules are extracted from alignments using the constraints described in (Chiang, 2007) with additional count and pattern filters (Iglesias et al., 2009b). We use a hierarchical phrase-based decoder (Iglesias et al., 2009a; de Gispert et al., 2010a) which directly generates word lattices from recursive translation networks without any intermediate hypergraph representation (Iglesias et al., 2011). The LMERT and TGMERT optimisation algorithms are particularly suitable for this realisation of hiero in that the lattice representation avoids the need to use the hypergraph formulation of MERT given by Kumar et al. (2009).

MERT optimises the weights of the following features: target language model, source-to-target and target-to-source translation models, word and rule penalties, number of usages of the glue rule, word deletion scale factor, source-to-target and target-to-source lexical models, and three count-based features that track the frequency of rules in the parallel data (Bender et al., 2007). In both Arabic-to-English and Chinese-to-English experiments all MERT implementations start from a flat feature weight initialization. At each iteration new lattices and k-best lists are generated from the best parameters at the previous iteration, and each subsequent iteration includes 100 hypotheses from the previous iteration. For Arabic-to-English we consider an additional twenty random starting parameters at every iteration. All translation scores are reported for the IBM implementation of BLEU using case-insensitive matching. We report BLEU scores for the Tune set at the start and end of each iteration.

The results for Arabic-to-English and Chinese-to-English are shown in Table 1. Both TGMERT and LMERT converge to a small gain over MERT in fewer iterations, consistent with previous reports (Macherey et al., 2008).

5 Discussion

We have described a lattice-based line optimisation algorithm which can be incorporated into MERT for parameter tuning of SMT systems and systems based on log-linear models. Our approach recasts the optimisation procedure used in MERT in terms of Tropical Geometry; given this formulation implementation is relatively straightforward using standard WFST operations and algorithms.

[3] See http://projects.ldc.upenn.edu/gale/data/catalog.html

References

C. Allauzen, M. Riley, J. Schalkwyk, W. Skut, and M. Mohri. 2007. OpenFst: A general and efficient weighted finite-state transducer library. In *Proceedings of the Ninth International Conference on Implementation and Application of Automata*, pages 11–23.

O. Bender, E. Matusov, S. Hahn, S. Hasan, S. Khadivi, and H. Ney. 2007. The RWTH Arabic-to-English spoken language translation system. In *Automatic Speech Recognition Understanding*, pages 396 –401.

J.L. Bentley and T.A. Ottmann. 1979. Algorithms for reporting and counting geometric intersections. *Computers, IEEE Transactions on*, C-28(9):643 –647.

Daniel Cer, Daniel Jurafsky, and Christopher D. Manning. 2008. Regularization and search for minimum error rate training. In *Proceedings of the Third Workshop on Statistical Machine Translation*, pages 26–34.

David Chiang. 2007. Hierarchical phrase-based translation. *Computational Linguistics*, 33.

David A. Cox, John Little, and Donal O'Shea. 2007. *Ideals, Varieties, and Algorithms: An Introduction to Computational Algebraic Geometry and Commutative Algebra, 3/e (Undergraduate Texts in Mathematics)*.

Adrià de Gispert, Gonzalo Iglesias, Graeme Blackwood, Eduardo R. Banga, and William Byrne. 2010a. Hierarchical phrase-based translation with weighted finite-state transducers and shallow-n grammars. *Computational Linguistics*, 36(3):505–533.

Adrià de Gispert, Juan Pino, and William Byrne. 2010b. Hierarchical phrase-based translation grammars extracted from alignment posterior probabilities. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, pages 545–554.

Chris Dyer, Adam Lopez, Juri Ganitkevitch, Jonathan Weese, Ferhan Ture, Phil Blunsom, Hendra Setiawan, Vladimir Eidelman, and Philip Resnik. 2010. cdec: A decoder, alignment, and learning framework for finite-state and context-free translation models. In *Proceedings of the ACL 2010 System Demonstrations*, pages 7–12, July.

Gonzalo Iglesias, Adrià de Gispert, Eduardo R. Banga, and William Byrne. 2009a. Hierarchical phrase-based translation with weighted finite state transducers. In *Proceedings of HLT: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 433–441.

Gonzalo Iglesias, Adrià de Gispert, Eduardo R. Banga, and William Byrne. 2009b. Rule filtering by pattern for efficient hierarchical translation. In *Proceedings of the 12th Conference of the European Chapter of the Association for Computational Linguistics*, pages 380–388.

Gonzalo Iglesias, Cyril Allauzen, William Byrne, Adrià de Gispert, and Michael Riley. 2011. Hierarchical phrase-based translation representations. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 1373–1383. Association for Computational Linguistics.

Philipp Koehn. 2010. *Statistical Machine Translation*. Cambridge University Press.

Shankar Kumar, Wolfgang Macherey, Chris Dyer, and Franz Och. 2009. Efficient minimum error rate training and minimum bayes-risk decoding for translation hypergraphs and lattices. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*, pages 163–171.

Wolfgang Macherey, Franz Och, Ignacio Thayer, and Jakob Uszkoreit. 2008. Lattice-based minimum error rate training for statistical machine translation. In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*, pages 725–734.

Mehryar Mohri and Michael Riley. 2002. An efficient algorithm for the n-best-strings problem. In *Proceedings of the International Conference on Spoken Language Processing 2002*.

Mehryar Mohri, Fernando C. N. Pereira, and Michael Riley. 2008. Speech recognition with weighted finite-state transducers. *Handbook on Speech Processing and Speech Communication*.

Mehryar Mohri. 2002. Semiring frameworks and algorithms for shortest-distance problems. *J. Autom. Lang. Comb.*, 7(3):321–350.

Franz Josef Och and Hermann Ney. 2002. Discriminative training and maximum entropy models for statistical machine translation. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, pages 295–302.

Franz Josef Och. 2003. Minimum error rate training in statistical machine translation. In *Proceedings of the 41st Annual Meeting on Association for Computational Linguistics*, pages 160–167.

K. A. Papineni, S. Roukos, T. Ward, and W. Zhu. 2002. BLEU: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, pages 311–318.

W. H. Press, W. T. Vetterling, S. A. Teukolsky, and B. P. Flannery. 2002. *Numerical Recipes in C++: the art of scientific computing*. Cambridge University Press.

J. Richter-Gebert, B. Sturmfels, and T. Theobald. 2005. First steps in tropical geometry. In *Idempotent mathematics and mathematical physics*.

Artem Sokolov and François Yvon. 2011. Minimum error rate training semiring. In *Proceedings of the European Association for Machine Translation*.

David Speyer and Bernd Sturmfels. 2009. Tropical mathematics. *Mathematics Magazine*.

Aurelien Waite, Graeme Blackwood, and William Byrne. 2011. Lattice-based minimum error rate training using weighted finite-state transducers with tropical polynomial weights. Technical report, Department of Engineering, University of Cambridge.

Richard Zens, Sasa Hasan, and Hermann Ney. 2007. A systematic comparison of training criteria for statistical machine translation. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 524–532.