The Geometry of Statistical Machine Translation

Aurelien Waite, William Byrne
Department of Engineering, University of Cambridge, UK
aaw35@cam.ac.uk, wjb31@cam.ac.uk

Abstract

Most modern statistical machine translation systems are based on linear statistical models. One extremely effective method for estimating the model parameters is minimum error rate training (MERT), which is an efficient form of line optimisation adapted to the highly non-linear objective functions used in machine translation. We describe a polynomial-time generalisation of line optimisation that computes the error surface over a plane embedded in parameter space. The description of this algorithm relies on convex geometry, which is the mathematics of polytopes and their faces.

Using this geometric representation of MERT we investigate whether the optimisation of linear models is tractable in general. Previous work on finding optimal solutions in MERT (Galley and Quirk, 2011) established a worst-case complexity that was exponential in the number of sentences, in contrast we show that exponential dependence in the worst-case complexity is mainly in the number of features.

Although our work is framed with respect to MERT, the convex geometric description is also applicable to other error-based training methods for linear models. We believe our analysis has important ramifications because it suggests that the current trend in building statistical machine translation systems by introducing a very large number of sparse features is inherently not robust.

1 Introduction

The linear model of Statistical Machine Translation (SMT) (Och and Ney, 2002) casts translation as a search for translation hypotheses under a linear combination of weighted features: a source language sentence $\mathbf{f}$ is translated as

$$\hat{\mathbf{e}}(\mathbf{f};\mathbf{w}) = \underset{\mathbf{e}}{\operatorname{argmax}}\{\mathbf{w}\mathbf{h}(\mathbf{e},\mathbf{f})\} \quad (1)$$

where translation scores are a linear combination of the $D \times 1$ feature vector $\mathbf{h}(\mathbf{e},\mathbf{f}) \in \mathbb{R}^D$ under the $1 \times D$ model parameter vector $\mathbf{w}$.

Convex geometry (Ziegler, 1995) is the mathematics of such linear equations presented as the study of convex polytopes. We use convex geometry to show that the behaviour of training methods such as MERT (Och, 2003; Macherey et al., 2008), MIRA (Crammer et al., 2006), PRO (Hopkins and May, 2011), and others converge with a high feature dimension. In particular we analyse how robustness decreases in linear models as feature dimension increases. We believe that severe overtraining is a problem in many current linear model formulations due to this lack of robustness.

In the process of building this geometric representation of linear models we discuss algorithms such as the Minkowski sum algorithm (Fukuda, 2004) and projected MERT (Section 4.2) that could be useful for designing new and more robust training algorithms for SMT and other natural language processing problems.

2 Training Linear Models

Let $\mathbf{f}_1 \ldots \mathbf{f}_S$ be a set of S source language sentences with reference translations $\mathbf{r}_1 \ldots \mathbf{r}_S$. The goal is to estimate the model parameter vector $\mathbf{w}$ so as to minimize an error count based on an automated metric, such as BLEU (Papineni et al., 2002), assumed to be

Human Language Technologies: The 2015 Annual Conference of the North American Chapter of the ACL, pages 376–386,
Denver, Colorado, May 31 – June 5, 2015.

additive over sentences:

$$\hat{\mathbf{w}} = \underset{\mathbf{w}}{\operatorname{argmin}} \sum_{s=1}^{S} E(\hat{\mathbf{e}}(\mathbf{f}_s; \mathbf{w}), \mathbf{r}_s) \quad (2)$$

Optimisation can be made tractable by restricting the search to rescoring of K-best lists of translation hypotheses, $\{\mathbf{e}_{s,i}, 1 \leq i \leq K\}_{s=1}^{S}$. For $\mathbf{f}_s$, let $\mathbf{h}_{s,i} = \mathbf{h}(\mathbf{e}_{s,i}, \mathbf{f}_s)$ be the feature vector associated with hypothesis $\mathbf{e}_{s,i}$. Restricted to these lists, the general decoder of Eqn. 1 becomes

$$\hat{\mathbf{e}}(\mathbf{f_s}; \mathbf{w}) = \underset{\mathbf{e}_{s,i}}{\operatorname{argmax}} \{\mathbf{w}\mathbf{h}(\mathbf{e}_{s,i}, \mathbf{f}_s)\} \quad (3)$$

Although the objective function in Eqn. (2) cannot be solved analytically, MERT as described by Och (2003) can be performed over the K-best lists. The line optimisation procedure considers a subset of parameters defined by the line $\mathbf{w}^{(0)} + \gamma\mathbf{d}$, where $\mathbf{w}^{(0)}$ corresponds to an initial point in parameter space and $\mathbf{d}$ is the direction along which to optimise. Eqn. (3) can be rewritten as:

$$\hat{\mathbf{e}}(\mathbf{f_s}; \gamma) = \underset{\mathbf{e_{s,i}}}{\operatorname{argmax}} \{\mathbf{w}^{(0)}\mathbf{h}_{s,i} + \gamma\mathbf{d}\mathbf{h}_{s,i})\} \quad (4)$$

Line optimisation reduces the D-dimensional procedure in Eqn. (2) to a 1-Dimensional problem that can be easily solved using a geometric algorithm for many source sentences (Macherey et al., 2008).

More recently, Galley and Quirk (2011) have introduced linear programming MERT (LP-MERT) as an exact search algorithm that reaches the global optimum of the training criterion. A hypothesis $\mathbf{e}_{s,i}$ from the sth K-best list can be selected by the decoder only if

$$\mathbf{w}(\mathbf{h}_{s,j} - \mathbf{h}_{s,i}) \leq 0 \text{ for } 1 \leq j \leq K \quad (5)$$

for some parameter vector $\mathbf{w} \neq \mathbf{0}$. If such a solution exists then the system of inequalities is *feasible*, and defines a convex region in parameter space within which any parameter $\mathbf{w}$ will yield $\mathbf{e}_{s,i}$. Testing the system of inequalities in (5) and finding a parameter vector can be cast as a linear programming feasibility problem (Galley and Quirk, 2011), and this can be extended to find a parameter vector that optimizes Eqn. 2 over a collection of K-best lists. We discuss the complexity of this operation in Section 4.1.

Hopkins and May (2011) note that for the sth source sentence, the parameter $\mathbf{w}$ that correctly ranks its K-best list must satisfy the following set of constraints for $1 \leq i, j \leq K$:

$$\mathbf{w}(\mathbf{h}_{s,j} - \mathbf{h}_{s,i}) \leq 0 \text{ if } \Delta(\mathbf{e}_{s,i}, \mathbf{e}_{s,j}) \geq 0 \quad (6)$$

where Δ computes the difference in error between two hypotheses. The difference vectors $(\mathbf{h}_{s,j} - \mathbf{h}_{s,i})$ associated with each constraint can be used as input vectors for a binary classification problem in which the aim is to predict whether the the difference in error $\Delta(\mathbf{e}_{s,i}, \mathbf{e}_{s,j})$ is positive or negative. Hopkins and May (2011) call this algorithm Pairwise Ranking Optimisation (PRO). Because there are SK^2 difference vectors across all source sentences, a subset of constraints is sampled in the original formulation; with effcient calculation of rankings, sampling can be avoided (Dreyer and Dong, 2015).

The online error based training algorithm MIRA (Crammer et al., 2006) is also used for SMT (Watanabe et al., 2007; Chiang et al., 2008; Chiang, 2012). Using a sentence-level error function, a set of S oracle hypotheses are indexed with the vector $\hat{\mathbf{i}}$:

$$\hat{i}_s = \underset{i}{\operatorname{argmin}}\, E(\mathbf{e}_{s,i}, \mathbf{r}_s) \text{ for } 1 \leq s \leq S$$

For a given s the objective at iteration $n+1$ is :

$$\underset{\mathbf{w}^{(n+1)}}{\text{minimise}} \quad \frac{1}{2}\|\mathbf{w}^{(n+1)} - \mathbf{w}^{(n)}\|^2 + C\sum_{j=1}^{K}\xi_j \quad (7)$$

$$\text{subject to } \xi_j \geq 0 \text{ and for } 1 \leq j \leq K, \hat{i}_s \neq j :$$

$$\mathbf{w}^{(n+1)}(\mathbf{h}_{s,j} - \mathbf{h}_{s,\hat{i}_s}) + \Delta(\mathbf{e}_{s,\hat{i}_s}, \mathbf{e}_{s,j}) - \xi_j \leq 0$$

where $\{\xi\}$ are slack variables added to allow infeasible solutions, and C controls the trade-off between error minimisation and margin maximisation. The online nature of the optimiser results in complex implementations, therefore batch versions of MIRA have been proposed (Cherry and Foster, 2012; Gimpel and Smith, 2012).

Although MERT, LP-MERT, PRO, and MIRA carry out their search in very different ways, we can compare them in terms of the constraints they are attempting to satisfy. A feasible solution for LP-MERT is also an optimal solution for MERT, and vice versa. The constraints (Eqn. (5)) that define LP-MERT are a subset of the constraints (Eqn. (6))

that define PRO and so a feasible solution for PRO will also be feasible for LP-MERT; however the converse is not necessarily true. The constraints that define MIRA (Eqn. (7)) are similar to the LP-MERT constraints (5), although with the addition of slack variables and the Δ function to handle infeasible solutions. However, if a feasible solution is available for MIRA, then these extra quantities are unnecessary. With these quantities removed, then we recover a 'hard-margin' optimiser, which utilises the same constraint set as in LP-MERT. In the feasible case, the solution found by MIRA is also a solution for LP-MERT.

2.1 Survey of Recent Work

One avenue of SMT research has been to add as many features as possible to the linear model, especially in the form of sparse features (Chiang et al., 2009; Hopkins and May, 2011; Cherry and Foster, 2012; Gimpel and Smith, 2012; Flanigan et al., 2013; Galley et al., 2013; Green et al., 2013). The assumption is that the addition of new features will improve translation performance. It is interesting to read the justification for many of these works as stated in their abstracts. For example Hopkins and May (2011) state that:

> We establish PRO's scalability and effectiveness by comparing it to MERT and MIRA and demonstrate parity on both phrase-based and syntax-based systems

Cherry and Foster (2012) state:

> Among other results, we find that a simple and efficient batch version of MIRA performs at least as well as training online.

Along similar lines Gimpel and Smith (2012) state:

> [We] present a training algorithm that is easy to implement and that performs comparable to others.

In defence of MERT, Galley et al. (2013) state:

> Experiments with up to 3600 features show that these extensions of MERT yield results comparable to PRO, a learner often used with large feature sets.

Green et al. (2013) also note that feature-rich models are rarely used in annual MT evaluations, an observation they use to motivate an investigation into adaptive learning rate algorithms.

Why do such different methods give such remarkably 'comparable' performance in research settings? And why is it so difficult to get general and unambiguous improvements through the use of high dimensional, sparse features? We believe that the explanation is in *feasibility*. If the oracle index vector $\hat{\mathbf{i}}$ is feasible then all training methods will find very similar solutions. Our belief is that as the feature dimension increases, the chance of an oracle index vector being feasible also increases.

3 Convex Geometry

We now build on the description of LP-MERT to give a geometric interpretation to training linear models. We first give a concise summary of the fundamentals of convex geometry as presented by (Ziegler, 1995) after which we work through the example in Cer et al. (2008) to provide an intuition behind these concepts.

3.1 Convex Geometry Fundamentals

In this section we reference definitions from convex geometry (Ziegler, 1995) in a form that allows us to describe SMT model parameter optimisation.

Vector Space The real valued vector space $\mathbb{R}^D$ represents the space of all finite D-dimensional feature vectors.

Dual Vector Space The dual vector space $(\mathbb{R}^D)^*$ are the real linear functions $\mathbb{R}^D \rightarrow \mathbb{R}$.

Polytope The polytope $H_s \subseteq \mathbb{R}^D$ is the convex hull of the finite set of feature vectors associated with the K hypotheses for the sth sentence, i.e. $H_s = \text{conv}(\mathbf{h}_{s,1}, \ldots, \mathbf{h}_{s,K})$.

Faces in $\mathbb{R}^D$ Suppose for $\mathbf{w} \in (\mathbb{R}^D)^*$ that $\mathbf{wh} \leq \max_{\mathbf{h}' \in H_s} \mathbf{wh}'$, $\forall\, \mathbf{h} \in H_s$. A *face* is defined as

$$F = \{\mathbf{h} \in H_s : \mathbf{wh} = \max_{\mathbf{h}' \in H_s} \mathbf{wh}'\} \qquad (8)$$

Vertex A face consisting of a single point is called a *vertex*. The set of vertices of a polytope is denoted $\text{vert}(H_s)$.

Edge An *edge* is a face in the form of a line segment between two vertices $\mathbf{h}_{s,i}$ and $\mathbf{h}_{s,j}$ in the polytope H_s. The edge can be written as $[\mathbf{h}_{s,i}, \mathbf{h}_{s,j}] = \text{conv}(\mathbf{h}_{s,i}, \mathbf{h}_{s,j})$. If an edge exists then the following

	h_{LM} : $\log(P_{LM}(\mathbf{e}))$	h_{TM} : $\log(P_{TM}(\mathbf{f}\|\mathbf{e}))$
$\mathbf{e}_1$	-0.1	-1.2
$\mathbf{e}_2$	-1.2	-0.2
$\mathbf{e}_3$	-0.9	-1.6
$\mathbf{e}_4$	-0.9	-0.1
$\mathbf{e}_5$	-0.8	-0.9

Table 1: An example set of two dimensional feature vectors (after Cer et al. (2008), Table 1) with language model (h_{LM}) and translation model (h_{TM}) components. A fifth feature vector has been added to illustrate redundancy.

modified system from (5) is feasible

$$\begin{aligned} &\mathbf{w}(\mathbf{h}_j - \mathbf{h}_i) = 0 \\ &\mathbf{w}(\mathbf{h}_k - \mathbf{h}_i) < 0,\ 1 \leq k \leq K, k \neq i, k \neq j \\ &\mathbf{w}(\mathbf{h}_l - \mathbf{h}_j) < 0,\ 1 \leq l \leq K, l \neq i, l \neq j \end{aligned} \tag{9}$$

which implies that $[\mathbf{h}_{s,i}, \mathbf{h}_{s,j}]$ defines a decision boundary in $(\mathbb{R}^D)^*$ between the parameters that maximise $\mathbf{h}_{s,i}$ and those that maximise $\mathbf{h}_{s,j}$.

Normal Cone For the face F in polytope H_s the normal cone N_F takes the form.

$$\begin{aligned} N_F = \{\mathbf{w} : \mathbf{w}(\mathbf{h}_{s,j} - \mathbf{h}_{s,i}) \leq 0, \forall \mathbf{h}_{s,i} \in \text{vert}(F), \\ \forall \mathbf{h}_{s,j} \in \text{vert}(H_s)\} \end{aligned} \tag{10}$$

If the face is a vertex $F = \{\mathbf{h}_{s,i}\}$ then its normal cone $N_{\{\mathbf{h}_{s,i}\}}$ is the set of feasible parameters that satisfy the system in (5).

Normal Fan The set of all normal cones associated with the faces of H_s is called the *normal fan* $\mathcal{N}(H_s)$.

3.2 Drawing a Normal Fan

Following the example in Cer et al. (2008) we analyze a system based on two features: the translation $P_{TM}(\mathbf{f}|\mathbf{e})$ and language $P_{LM}(\mathbf{e})$ models. For brevity we omit the common sentence index, so that $\mathbf{h}_i = \mathbf{h}_{s,i}$. The system produces a set of four hypotheses which yield four feature vectors $\{\mathbf{h}_1, \mathbf{h}_2, \mathbf{h}_3, \mathbf{h}_4\}$ (Table 1). To this set of four hypotheses, we add a fifth hypothesis and feature vector $\mathbf{h}_5$ to illustrate an infeasible solution. These feature vectors are plotted in Figure 1.

The feature vectors form a polytope H shaded in light blue. From Figure 1 we see that $\mathbf{h}_4$ satisfies the

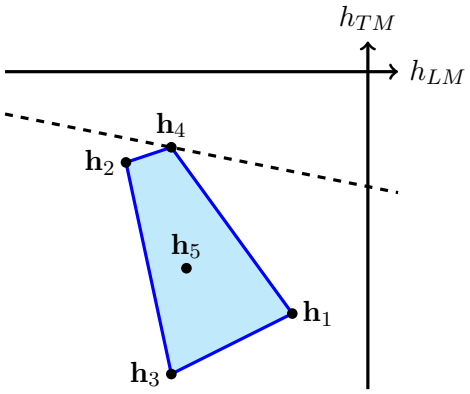

Figure 1: A geometric interpretation of LP-MERT (after Cer et al. (2008) and Galley and Quirk (2011)). The decision boundary represented by the dashed line intersects the polytope at only $\mathbf{h}_4$, making it a vertex. No decision boundary intersects $\mathbf{h}_5$ without intersecting other points in the polytope, making $\mathbf{h}_5$ redundant.

conditions for a vertex in Eqn. (8), because we can draw a decision boundary that interests the vertex and no other $\mathbf{h} \in H$. We also note $\mathbf{h}_5$ is not a vertex, and is *redundant* to the description of H.

Figure 1 of Cer et al. (2008) actually shows a normal fan, although it is not described as such. We now describe how this geometric object is constructed step by step in Figure 2. In Part (a) we identify the edge $[\mathbf{h}_4, \mathbf{h}_1]$ in $\mathbb{R}^2$ with a decision boundary represented by a dashed line. We have also drawn a vector $\mathbf{w}$ normal to the decision boundary that satisfies Eqn. (8). This parameter would result in a tied model score such that $\mathbf{w}\mathbf{h}_4 = \mathbf{w}\mathbf{h}_1$. When moving to $(\mathbb{R}^2)^*$ we see that the normal cone $N_{[\mathbf{h}_4,\mathbf{h}_1]}$ is a ray parallel to $\mathbf{w}$. This ray can be considered as the set of parameter vectors that yield the edge $[\mathbf{h}_4, \mathbf{h}_1]$. The ray is also a decision boundary in $(\mathbb{R}^2)^*$, with parameters on either side of the decision boundary maximising either $\mathbf{h}_4$ or $\mathbf{h}_1$. Any vector parallel to the edge $[\mathbf{h}_4, \mathbf{h}_1]$, such as $(\mathbf{h}_1 - \mathbf{h}_4)$, can be used to define this decision boundary in $(\mathbb{R}^2)^*$.

Next in Part (b), with the same procedure we define the normal cone for the edge $[\mathbf{h}_3, \mathbf{h}_1]$. Now both the edges from parts (a) and (b) share the the vertex $\mathbf{h}_1$. This implies that any parameter vector that lies between the two decision boundaries (i.e. between the two rays $N_{[\mathbf{h}_3,\mathbf{h}_1]}$ and $N_{[\mathbf{h}_4,\mathbf{h}_1]}$) would maximise the vertex $\mathbf{h}_1$: this is the set of vectors that comprise

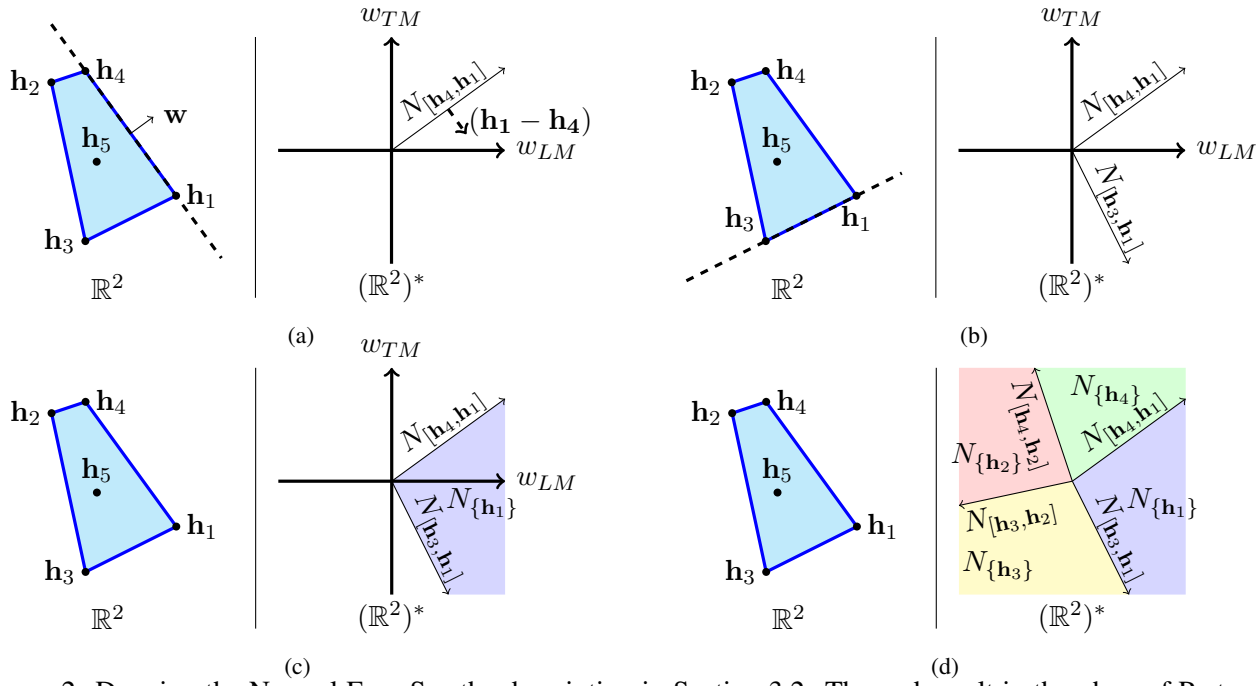

Figure 2: Drawing the Normal Fan. See the description in Section 3.2. The end result in the r.h.s. of Part (d) reproduces Figure 1 from Cer et al. (2008), identifying the normal cones for all vertices.

normal cone of the vertex $N_{\{\mathbf{h}_1\}}$.

In Part (c) we have shaded and labelled $N_{\{\mathbf{h}_1\}}$. Note that no other edges are needed to define this normal cone; these other edges are redundant to the normal cone's description.

Finally in Part (d) we draw the full fan. We have omitted the axes in $(\mathbb{R}^2)^*$ for clarity. The normal cones for all 4 vertices have been identified.

4 Training Set Geometry

The previous discussion treated only a single sentence. For a training set of S input sentences, let $\mathbf{i}$ be an index vector that contains S elements. Each element is an index i_s to a hypothesis and a feature vector for the sth sentence. A particular $\mathbf{i}$ specifies a set of hypotheses drawn from each of the K-best lists. LP-MERT builds a set of K^S feature vectors associated with S dimensional index vectors $\mathbf{i}$ of the form $\mathbf{h_i} = \mathbf{h}_{1,i_1} + \ldots + \mathbf{h}_{S,i_S}$. The polytope of these feature vectors is then constructed.

In convex geometry this operation is called the *Minkowski sum* and for the polytopes H_s and H_t, is defined as (Ziegler, 1995)

$$H_s + H_t := \{\mathbf{h} + \mathbf{h}' : \mathbf{h} \in H_s, \mathbf{h}' \in H_t\} \quad (11)$$

We illustrate this operation in the top part of Figure 3. The Minkowski sum is commutative and associative and generalises to more than two polytopes (Gritzmann and Sturmfels, 1992).

For the polytopes H_s and H_t the *common refinement* (Ziegler, 1995) is

$$\mathcal{N}(H_s) \wedge \mathcal{N}(H_t) := \{N \cap N' : N \in \mathcal{N}(H_s), N' \in \mathcal{N}(H_t)\} \quad (12)$$

Each cone in the common refinement is the set of parameter vectors that maximise two faces in H_s and H_t. This operation is shown in the bottom part of Figure 3.

As suggested by Figure 3 the Minkowski sum and common refinement are linked by the following

Proposition 1. $\mathcal{N}(H_s + H_t) = \mathcal{N}(H_s) \wedge \mathcal{N}(H_t)$

Proof. See Gritzmann and Sturmfels (1992) □

This implies that, with $\mathbf{h_i}$ defined for the index vector $\mathbf{i}$, the Minkowski sum defines the parameter vectors that satisfy the following (Tsochantaridis et al., 2005, Eqn. 3)

$$\mathbf{w}(\mathbf{h}_{s,j} - \mathbf{h}_{s,i_s}) \leq 0, \ 1 \leq s \leq S, 1 \leq j \leq K \quad (13)$$

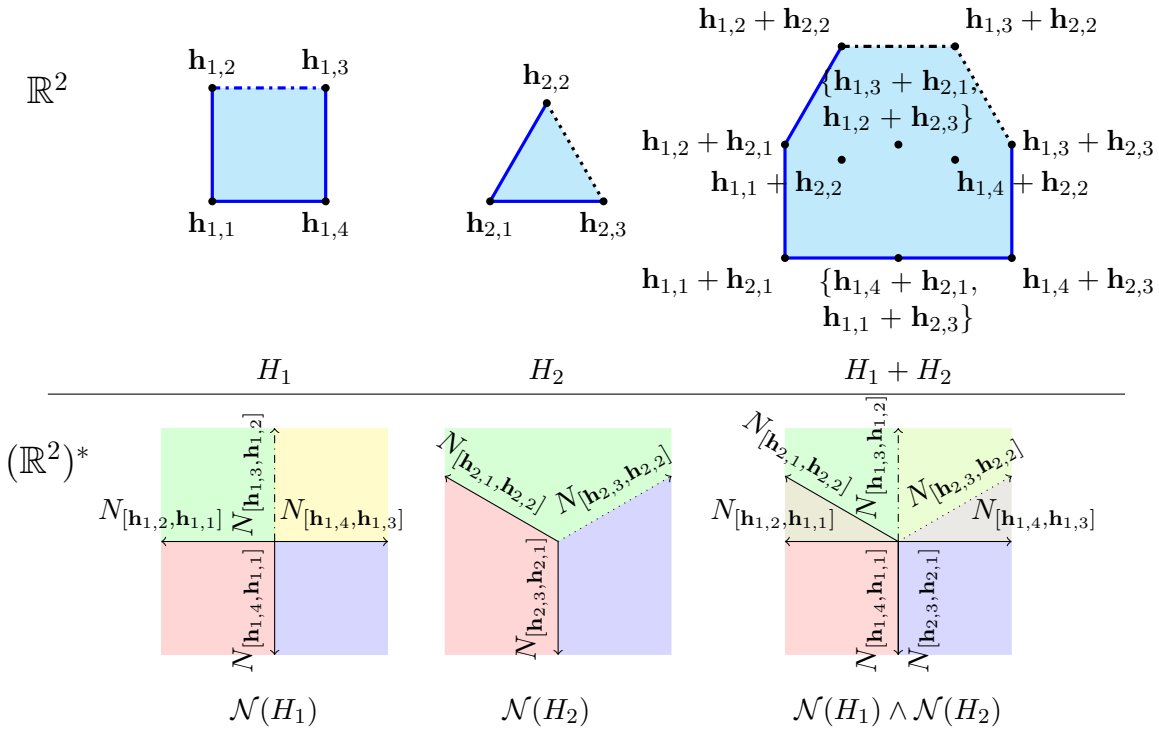

Figure 3: An example of the equivalence between the Minkowski sum and the common refinement.

4.1 Computing the Minkowski Sum

In the top part of the Figure 3 we see that computing the Minkowski sum directly gives 12 feature vectors, 10 of which are unique. Each feature vector would have to be tested under LP-MERT. In general there are K^S such feature vectors and exhaustive testing is impractical. LP-MERT performs a lazy enumeration of feature vectors as managed through a divide and conquer algorithm. We believe that in the worst case the complexity of this algorithm could be $O(K^S)$.

The lower part of Figure 3 shows the computation of the common refinement. The common refinement appears as if one normal fan was superimposed on the other. We can see there are six decision boundaries associated with the six edges of the Minkowski sum. Even in this simple example, we can see that the common refinement is an easier quantity to compute than the Minkowski sum.

We now briefly describe the algorithm of Fukuda (2004) that computes the common refinement. Consider the example in Figure 3. For H_1 and H_2 we have drawn an edge in each polytope with a dashed line. The corresponding decision boundaries in their normal fans have also been drawn with dashed lines.

Now consider the vertex $\mathbf{h}_{1,3} + \mathbf{h}_{2,2}$ in $H = H_1 + H_2$ and note it has two incident edges. These edges are parallel to edges in the summand polytopes and correspond to decision boundaries in the normal cone $N_{\{\mathbf{h}_{1,3}+\mathbf{h}_{2,2}\}}$.

We can find the redundant edges in the Minkowski sum by testing the edges suggested by the summand polytopes. If a decision boundary in $(\mathbb{R}^D)^*$ is redundant, then we can ignore the feature vector that shares the decision boundary. For example $\mathbf{h}_{1,4} + \mathbf{h}_{2,2}$ is redundant and the decision boundary $N_{[\mathbf{h}_{1,3},\mathbf{h}_{1,4}]}$ is also redundant to the description of the normal cone $N_{\{\mathbf{h}_{1,3}+\mathbf{h}_{2,2}\}}$. The test for redundant edges can be performed by a linear program.

Given a Minkowski sum H we can define an undirected cyclic graph $G(H) = (\text{vert}(H), E)$ where E is the set of edges. The degree of a vertex in $G(H)$ is the number of edges incident to a vertex; δ is denoted as the maximum degree of the vertices.

The linear program for testing redundancy of decision boundaries has a runtime of $O(D^{3.5}\delta)$ (Fukuda, 2004). Enumerating the vertices of graph $G(H)$ is not trivial due to it being an undirected and cyclic graph. The solution is to use a *reverse search* algorithm (Avis and Fukuda, 1993). Essen-

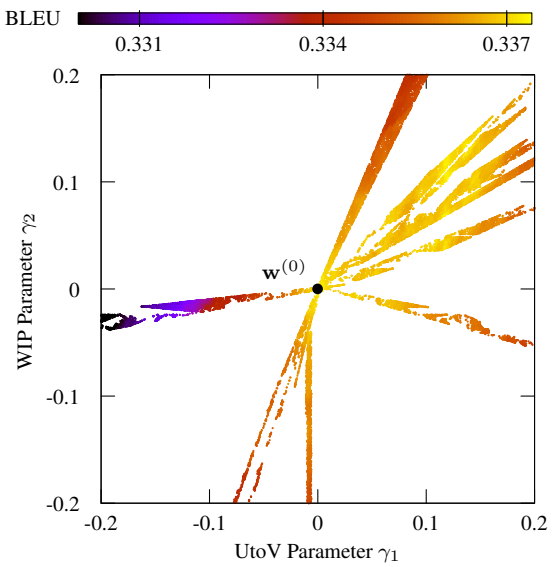

Figure 4: The BLEU score over a 1502 sentence tune set for the CUED Russian-to-English (Pino et al., 2013) system over two parameters. Enumerated vertices of the Minkowski sum are shown in the shaded regions.

tially reverse search transforms the graph into a tree. The vertex associated with $\mathbf{w}^{(0)}$ is denoted as the root of the tree, and from this root vertices are enumerated in reverse order of model score under $\mathbf{w}^{(0)}$. Each branch of the tree can be enumerated independently, which means that the enumeration can be parallelised.

The complexity of the full algorithm is $O(\delta(D^{3.5}\delta)|\operatorname{vert}(H)|)$ (Fukuda, 2004). In comparison with the $O(K^S)$ for LP-MERT the worst case complexity of the reverse search algorithm is linear with respect to the size of $\operatorname{vert}(H)$.

4.2 Two Dimensional Projected MERT

We now explore whether the reverse search algorithm is a practical method for performing MERT using an open source implementation of the algorithm (Weibel, 2010). For reasons discussed in the next section, we wish to reduce the feature dimension. For $M < D$, we can define a projection matrix $A_{M+1,D}$ that maps $\mathbf{h}_i \in \mathbb{R}^D$ into $\mathbb{R}^{M+1}$ as $A_{M+1,D}\mathbf{h}_i = \tilde{\mathbf{h}}_i$, $\tilde{\mathbf{h}}_i \in \mathbb{R}^{M+1}$. There are technical constraints to be observed, discussed in Waite (2014). We note that when $M = 1$ we obtain Eqn. (4).

For our demonstration, we plot the error count over a plane in $(\mathbb{R}^D)^*$. Using the CUED Russian-to-English (Pino et al., 2013) entry to WMT'13 (Bojar et al., 2013) we build a tune set of 1502 sentences. The system uses 12 features which we initially tune with lattice MERT (Macherey et al., 2008) to get a parameter $\mathbf{w}^{(0)}$. Using this parameter we generate 1000-best lists. We then project the feature functions in the 1000-best lists to a 3-dimensional representation that includes the source-to-target phrase probability (UtoV), the word insertion penalty (WIP), and the model score due to $\mathbf{w}^{(0)}$. We use the Minkowski sum algorithm to compute BLEU as $\boldsymbol{\gamma} \in (\mathbb{R}^2)^*$ is applied to the parameters from $\mathbf{w}^{(0)}$.

Figure 4 displays some of the characteristics of the algorithm[1]. This plot can be interpreted as a 3-dimensional version of Figure 3 in Macherey et al. (2008) where we represent the BLEU score as a heatmap instead of a third axis. Execution was on 12 CPU cores, leading to the distinct search regions, demonstrating the parallel nature of the algorithm. Weibel (2010) uses a depth-first enumeration order of $G(H)$, hence the narrow and deep exploration of $(\mathbb{R}^D)^*$. A breadth-first ordering would focus on cones closer to $\mathbf{w}^{(0)}$. To our knowledge, this is the first description of a generalised line optimisation algorithm that can search all the parameters in a plane in polynomial time. Extensions to higher dimensional search are straightforward.

5 Robustness of Linear Models

In the previous section we described the Minkowski sum polytope. Let us consider the following upper bound theorem

Theorem 1. *Let $H_1, \ldots, H_S$ be polytopes in $\mathbb{R}^D$ with at most N vertices each. Then for $D > 2$ the upper bound on number of vertices of $H_1 + \ldots + H_S$ is $O(S^{D-1}K^{2(D-1)})$.*

Proof. See Gritzmann and Sturmfels (1992) □

Each vertex $\mathbf{h_i}$ corresponds to a single index vector $\mathbf{i}$, which itself corresponds to a single set of selected hypotheses. Therefore the number of distinct sets of hypotheses that can be drawn

[1] A replication of this experiment forms part of the UCAM-SMT tutorial at `http://ucam-smt.github.io`

from the S K-best lists in bounded above by $O(\min(K^S, S^{D-1}K^{2(D-1)}))$.

For low dimension features, i.e. for D : $S^{D-1}K^{2(D-1)} \ll K^S$, the optimiser is therefore tightly constrained. It cannot pick arbitrarily from the individual K-best lists to optimise the overall BLEU score. We believe this acts as an *inherent form of regularisation*.

For example, in the system of Section 4.2 (D=12, S=1502, K=1000), only 10^{-4403} percent of the K^S possible index vectors are feasible. However, if the feature dimension D is increased to $D = 493$, then $S^{D-1}K^{2(D-1)} \gg K^S$ and this inherent regularisation is no longer at work: any index vector is feasible, and sentence hypotheses can chosen arbitrarily to optimise the overall BLEU score.

This exponential relationship of feasible solutions with respect to feature dimension can be seen in Figure 6 of Galley and Quirk (2011). At low feature dimension, they find that the LP-MERT algorithm can run to completion for a training set size of hundreds of sentences. As feature dimension increases, the runtime increases exponentially.

PRO and other ranking methods are similarly constrained for low dimensional feature vectors.

Theorem 2. *If H is a D-dimensional polytope, then for $D \geq 3$ the following is an upper bound on the number of edges $|E|$*

$$|E| \leq \binom{|\operatorname{vert}(H)|}{2} \tag{14}$$

Proof. This is a special case of the upper bound theorem. See Ziegler (1995, Theorem 8.23). □

Each feasible pairwise ranking of pairs of hypotheses corresponds to an edge in the Minkowski sum polytope. Therefore in low dimension ranking methods also benefit from this inherent regularisation.

For higher dimensional feature vectors, these upper bounds no longer guarantee that this inherent regularisation is at work. The analysis suggests - but does not imply - that index vectors, and their corresponding solutions, can be picked arbitrarily from the K-best lists. For MERT overtraining is clearly a risk.

MIRA and related methods have a regularisation mechanism due to the margin maximisation term in their objective functions. Although this form of regularisation may be helpful in practice, there is no guarantee that it will prevent overtraining due to the exponential increase in feasible solutions. For example the adaptive learning rate method of Green et al. (2013) finds gains of over 13 BLEU points in the training set with the addition of 390,000 features, yet only 2 to 3 BLEU points are found in the test set.

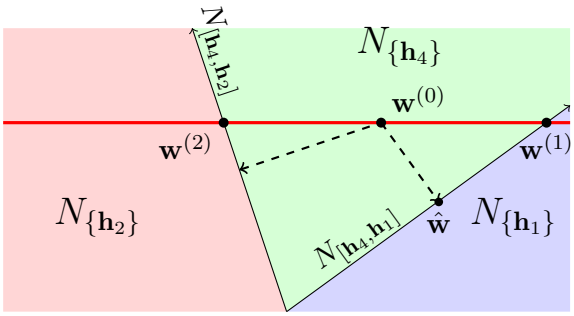

Figure 5: We redraw the normal fan from Figure 2 with potential optimal parameters under the ℓ_2 regularisation scheme of Galley et al. (2013) marked. The thick red line is the subspace of $(\mathbb{R}^2)^*$ optimised. The dashed lines mark the distances between the decision boundaries and $\mathbf{w}^{(0)}$.

5.1 A Note on Regularisation

The above analysis suggest a need for regularisation in training with high dimensional feature vectors. Galley et al. (2013) note that regularisation is hard to apply to linear models due to the magnitude invariance of $\mathbf{w}$ in Eqn. (1). Figure 2 makes the difficulty clear: the normal cones are determined entirely by the feature vectors of the training samples, and within any particular normal cone a parameter vector can be chosen with arbitrary magnitude. This renders schemes such as L1 or L2 normalisation ineffective. To avoid this, Galley et al. (2013) describe a regularisation scheme for line optimisation that encourages the optimal parameter to be found close to $\mathbf{w}^{(0)}$. The motivation is that $\mathbf{w}^{(0)}$ should be a trusted initial point, perhaps taken from a lower-dimensional model. We briefly discuss the challenges of doing this sort of regularisation in MERT.

In Figure 5 we reproduce the normal fan from Figure 2. In this diagram we represent the set of parameters considered by a line optimisation as a thick red line. Let us assume that both $\mathbf{e}_1$ and $\mathbf{e}_2$ have a

similarly low error count. Under the regularisation scheme of Galley et al. (2013) we have a choice of $\mathbf{w}^{(1)}$ or $\mathbf{w}^{(2)}$, which are equidistant from $\mathbf{w}^{(0)}$. In this affine projection of parameter space it is unclear which one is the optimum. However, if we consider the normal fan as a whole we can clearly see that $\hat{\mathbf{w}} \in N_{\{\mathbf{h}_i\}}$ is the optimal point under the regularisation. However, it is not obvious in the projected parameter space that $\hat{\mathbf{w}}$ is the better choice. This analysis suggests that direct intervention, e.g. monitoring BLEU on a held-out set, may be more effective in avoiding overtraining.

6 Discussion

The main contribution of this work is to present a novel geometric description of MERT. We show that it is possible to enumerate all the feasible solutions of a linear model in polynomial time using this description. The immediate conclusion from this work is that the current methods for estimating linear models as done in SMT works best for low dimensional feature vectors.

We can consider the SMT linear model as a member of a family of linear models where the output values are highly structured, and where each input yields a candidate space of possible output values. We have already noted that the constraints in (13) are shared with the structured-SVM (Tsochantaridis et al., 2005), and we can also see the same constraints in Eqn. 3 of Collins (2002). It is our belief that our analysis is applicable to all models in this family and extends far beyond the discussion of SMT here.

We note that the upper bound on feasible solutions increases polynomially in training set size S, whereas the number of possible solutions increases exponentially in S. The result is that the ratio of feasible to possible solutions decreases with S. Our analysis suggests that inherent regularisation should be improved by increasing training set size. This confirms most researchers intuition, with perhaps even larger training sets needed than previously believed.

Another avenue to prevent overtraining would be to project high-dimensional feature sets to low dimensional feature sets using the technique described in Section 4.1. We could then use existing training methods to optimise over the projected feature vectors.

We also note that non-linear models methods, such as neural networks (Schwenk et al., 2006; Kalchbrenner and Blunsom, 2013; Devlin et al., 2014; Cho et al., 2014) and decision forests (Criminisi et al., 2011) are not bound by these analyses. In particular neural networks are non-linear functions of the features, and decision forests actively reduce the number of features for individual trees in the forrest. From the perspective of this paper, the recent improvements in SMT due to neural networks are well motivated.

Acknowledgments

This research was supported by a doctoral training account from the Engineering and Physical Sciences Research Council.

References

David Avis and Komei Fukuda. 1993. Reverse search for enumeration. *Discrete Applied Mathematics*, 65:21–46.

Ondřej Bojar, Christian Buck, Chris Callison-Burch, Christian Federmann, Barry Haddow, Philipp Koehn, Christof Monz, Matt Post, Radu Soricut, and Lucia Specia. 2013. Findings of the 2013 Workshop on Statistical Machine Translation. In *Proceedings of the Eighth Workshop on Statistical Machine Translation*, pages 1–44, Sofia, Bulgaria, August. Association for Computational Linguistics.

Daniel Cer, Dan 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, Columbus, Ohio, June. Association for Computational Linguistics.

Colin Cherry and George Foster. 2012. Batch tuning strategies for statistical machine translation. In *Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 427–436, Montréal, Canada, June. Association for Computational Linguistics.

David Chiang, Yuval Marton, and Philip Resnik. 2008. Online large-margin training of syntactic and structural translation features. In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*, pages 224–233, Honolulu, Hawaii, October. Association for Computational Linguistics.

David Chiang, Kevin Knight, and Wei Wang. 2009. 11,001 new features for statistical machine translation. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 218–226, Boulder, Colorado, June. Association for Computational Linguistics.

David Chiang. 2012. Hope and fear for discriminative training of statistical translation models. *The Journal of Machine Learning Research*, 13(1):1159–1187.

Kyunghyun Cho, Bart van Merrienboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using RNN encoder–decoder for statistical machine translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1724–1734, Doha, Qatar, October. Association for Computational Linguistics.

Michael Collins. 2002. Discriminative training methods for hidden Markov models: Theory and experiments with perceptron algorithms. In *Proceedings of the 2002 Conference on Empirical Methods in Natural Language Processing*, pages 1–8. Association for Computational Linguistics, July.

Koby Crammer, Ofer Dekel, Joseph Keshet, Shai Shalev-Shwartz, and Yoram Singer. 2006. Online passive-aggressive algorithms. *The Journal of Machine Learning Research*, 7:551–585.

A. Criminisi, J. Shotton, and E. Konukoglu. 2011. Decision forests for classification, regression, density estimation, manifold learning and semi-supervised learning. Technical report, Microsoft Research.

Jacob Devlin, Rabih Zbib, Zhongqiang Huang, Thomas Lamar, Richard Schwartz, and John Makhoul. 2014. Fast and robust neural network joint models for statistical machine translation. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1370–1380, Baltimore, Maryland, June. Association for Computational Linguistics.

Markus Dreyer and Yuanzhe Dong. 2015. APRO: All-pairs ranking optimization for MT tuning. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*.

Jeffrey Flanigan, Chris Dyer, and Jaime Carbonell. 2013. Large-scale discriminative training for statistical machine translation using held-out line search. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 248–258, Atlanta, Georgia, June. Association for Computational Linguistics.

Komei Fukuda. 2004. From the zonotope construction to the Minkowski addition of convex polytopes. *Journal of Symbolic Computation*, 38(4):1261–1272.

Michel Galley and Chris Quirk. 2011. Optimal search for minimum error rate training. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 38–49, Edinburgh, Scotland, UK., July. Association for Computational Linguistics.

Michel Galley, Chris Quirk, Colin Cherry, and Kristina Toutanova. 2013. Regularized minimum error rate training. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1948–1959, Seattle, Washington, USA, October. Association for Computational Linguistics.

Kevin Gimpel and Noah A. Smith. 2012. Structured ramp loss minimization for machine translation. In *Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 221–231, Montréal, Canada, June. Association for Computational Linguistics.

Spence Green, Sida Wang, Daniel Cer, and Christopher D. Manning. 2013. Fast and adaptive online training of feature-rich translation models. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 311–321, Sofia, Bulgaria, August. Association for Computational Linguistics.

Peter Gritzmann and Bernd Sturmfels. 1992. Minkowski addition of polytopes: Computational complexity and applications to Gröbner bases. *SIAM Journal on Discrete Mathematics*, 6(2).

Mark Hopkins and Jonathan May. 2011. Tuning as ranking. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 1352–1362, Edinburgh, Scotland, UK., July. Association for Computational Linguistics.

Nal Kalchbrenner and Phil Blunsom. 2013. Recurrent continuous translation models. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1700–1709, Seattle, Washington, USA, October. Association for Computational Linguistics.

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, Honolulu, Hawaii, October. Association for Computational Linguistics.

Franz Josef Och and Hermann Ney. 2002. Discriminative training and maximum entropy models for statistical machine translation. In *Proceedings of 40th*

Annual Meeting of the Association for Computational Linguistics, pages 295–302, Philadelphia, Pennsylvania, USA, July. Association for Computational Linguistics.

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

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA, July. Association for Computational Linguistics.

Juan Pino, Aurelien Waite, Tong Xiao, Adrià de Gispert, Federico Flego, and William Byrne. 2013. The University of Cambridge Russian-English system at WMT13. In *Proceedings of the Eighth Workshop on Statistical Machine Translation*, pages 200–205, Sofia, Bulgaria, August. Association for Computational Linguistics.

Holger Schwenk, Daniel Dechelotte, and Jean-Luc Gauvain. 2006. Continuous space language models for statistical machine translation. In *Proceedings of the COLING/ACL 2006 Main Conference Poster Sessions*, pages 723–730, Sydney, Australia, July. Association for Computational Linguistics.

Ioannis Tsochantaridis, Thorsten Joachims, Thomas Hofmann, and Yasemin Altun. 2005. Large margin methods for structured and interdependent output variables. In *Journal of Machine Learning Research*, pages 1453–1484.

Aurelien Waite. 2014. *The Geometry of Statistical Machine Translation*. Ph.D. thesis, University of Cambridge, Cambridge, United Kingdom.

Taro Watanabe, Jun Suzuki, Hajime Tsukada, and Hideki Isozaki. 2007. Online large-margin training for statistical machine translation. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 764–773.

Christophe Weibel. 2010. Implementation and parallelization of a reverse-search algorithm for minkowski sums. In *Proceedings of the 12th Workshop on Algorithm Engineering and Experiments (ALENEX 2010)*, pages 34–42. SIAM.

G Ziegler. 1995. *Lectures on Polytopes*. Springer-Verlag.