Tuning as Ranking

Pairwise Ranking Optimisation (PRO)

HOPKINS, M. & MAY, J.

2011

Statistical Machine Translation (MT)

- An SMT system translates from one human language to another
- Such systems typically have a lot of parameters that need to be tuned

Current Tuning Solutions

- MERT
  - Well-understood, easy to implement, and runs quickly
  - Does not scale beyond a handful of features
- MIRA
  - Shown to perform well on large-scale tasks
  - Complex and architecturally different from MERT

Pairwise Ranking Optimisation (PRO)

- Adapts the MERT system
- Provides comparable performance to both
- Scales comparably to MIRA but is much simpler
- Should take about 2 hours to implement (supposedly)

Set-up (Definitions!)

Candidate Space ⟨Δ, I, J, e, x⟩

- Δ, the space's **dimensionality** (a positive integer)
- I, **sentence indices** (a set of positive integers)
- J maps
  - Each sentence index
  - To a set of **candidate indices** (positive integers)

Candidate Space $\langle \Delta, I, J, e, x \rangle$

- $e(i, j)$ maps
  - Each pair $\langle i, j \rangle \in I \times J(i)$
  - To the j^{th} target-language **candidate translation** of source sentence i
- $x(i, j)$ maps
  - Each pair $\langle i, j \rangle \in I \times J(i)$
  - To a Δ-dimension **feature vector** representation of $e(i, j)$

Policy p(i)

- A function corresponding to a candidate space
- It maps
  - Each source sentence index ($i \in I$)
  - To a candidate sentence index ($\in J(i)$)

Scoring Function, $h_{\mathbf{w}}(i, j) = \mathbf{w} \cdot \mathbf{x}(i, j)$

- Indicates how good candidate j is for source sentence i
- **w** is a weight vector that must be learnt
- Typically returns positive real numbers (higher ⇒ better)
- Can extend this idea to policy p by summing the costs of each candidate translation

$$H_{\mathbf{w}}(p) = \sum_{i \in I} h_{\mathbf{w}}(i, p(i))$$

A Gold Scoring Function, G

- An idealised equivalent of $H_w(p)$
- Maps
  - Each policy
  - To a real-valued score
- Typically calculated by a library, such as IBM Bleu

Goal of Tuning

- Goal is to find a weight vector **w**
- For space s, we want a **w** that, equivalently
  - Gives an $H_{\mathbf{w}}$ which behaves "similarly" to G on s
  - Minimises a **loss function** $l_s(H_{\mathbf{w}}, G)$

MERT

Two-Stage Feedback Loop

- **Candidate Generation**
  - Candidate translations are selected from a base candidate space s
  - Translations are added the **candidate pool**, s'
- **Optimisation**
  - The weight vector **w** is optimised to minimise a loss function $l_{s'}(H_{\mathbf{w}}, G)$
  - Loss defined to prefer weight vectors such that the gold function G scores $H_{\mathbf{w}}$'s best policy as highly as possible (0 loss if equal to G's best)
  - Implemented by line optimisation

Issues

- Does not scale well with dimensionality
- MERT optimisation focuses on H_w's best policy, and not on its overall ability to rank policies

Pairwise Ranking Optimisation (PRO)

Local Scoring Function, g

- Assume the gold scoring function G decomposes to:

$$G(p) = \sum_{i \in I} g(i, p(i))$$

- Here, g is a local scoring function
  - It is equivalent to h_w for H_w
  - It can be used to rank candidate translations for each source sentence

Example

Source Sentence		Candidate Translations				
i	Sentence string	j	$e(i,j)$	$\mathbf{x}(i,j)$	$h_{\mathbf{w}}(i,j)$	$g(i,j)$
1	"il ne va pas"	1	"he goes not"	[2 4]	0	0.28
		2	"he does not go"	[3 8]	2	0.42
		3	"she not go"	[6 1]	-11	0.12
2	"je ne vais pas"	1	"I go not"	[-3 -3]	3	0.15
		2	"we do not go"	[1 -5]	-7	0.18
		3	"I do not go"	[-5 -3]	7	0.34

Reframing the Learning Task with g

- The task is to classify candidate pairs, ⟨e(i, j), e(i, j’)⟩, into two categories
  - Correctly ordered (the first is better than the second)
  - Incorrectly ordered (the second is better than the first)

Reframing the Learning Task with g

- Thus, for a translations e(i, j) and e(i, j'), we want **w** such that

$$g(i, j) > g(i, j') \Leftrightarrow h_{\mathbf{w}}(i, j) > h_{\mathbf{w}}(i, j')$$

- We can algebraically turn this into a binary classification problem!

$$\begin{aligned} g(i,j) > g(i,j') &\Leftrightarrow h_{\mathbf{w}}(i,j) > h_{\mathbf{w}}(i,j') \\ &\Leftrightarrow h_{\mathbf{w}}(i,j) - h_{\mathbf{w}}(i,j') > 0 \\ &\Leftrightarrow \mathbf{w} \cdot \mathbf{x}(i,j) - \mathbf{w} \cdot \mathbf{x}(i,j') > 0 \\ &\Leftrightarrow \mathbf{w} \cdot (\mathbf{x}(i,j) - \mathbf{x}(i,j')) > 0 \end{aligned}$$

To Create Training Instances

1. Compute the difference vector $\mathbf{x}(i, j) - \mathbf{x}(i, j')$
2. Label it:
   - 'Positive' if the first vector is superior, according to g
   - 'Negative' if the second vector is superior, according to g

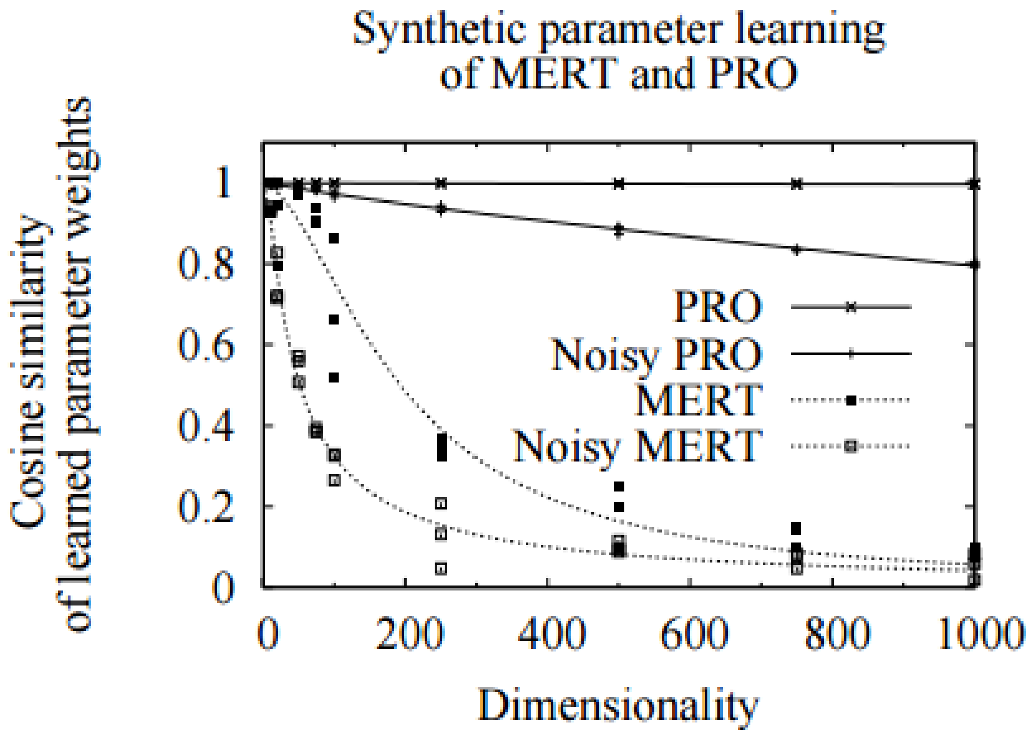
- Consider both difference vectors from a pair
- Randomly sample these vectors to create training data

Dimensional Scalability Evaluation

Set-up

1. Define G = $H_{\mathbf{w}*}$ (p) for some gold weight vector $\mathbf{w}^*$
2. Generate a Δ-dimensionality candidate pool
   - 500 source "sentences", each with 100 candidate "translations"
   - Draw, at random, Δ-dimensional feature vector values
3. Run the tuners
4. Repeat 1-3 with different Δ values
5. Repeat 1-4 with Gaussian noise added to feature vectors

Results


Synthetic parameter learning
of MERT and PRO
Cosine similarity
of learned parameter weights
1
0.8
0.6
0.4
0.2
0
0
200
400
600
800
1000
Dimensionality
PRO
Noisy PRO
MERT
Noisy MERT

Translation Evaluation

SBMT vs PBMT

- Syntax-Based systems (SBMT)
  - Based on the idea of translating syntactic units
  - Rather than single words or sequences of words
- Phrase-Based systems (PBMT)
  - Based on idea of translating whole sequences of words
  - Reduces the restrictions of word-based translation
  - The sequence lengths may differ

Evaluation Feature Sets

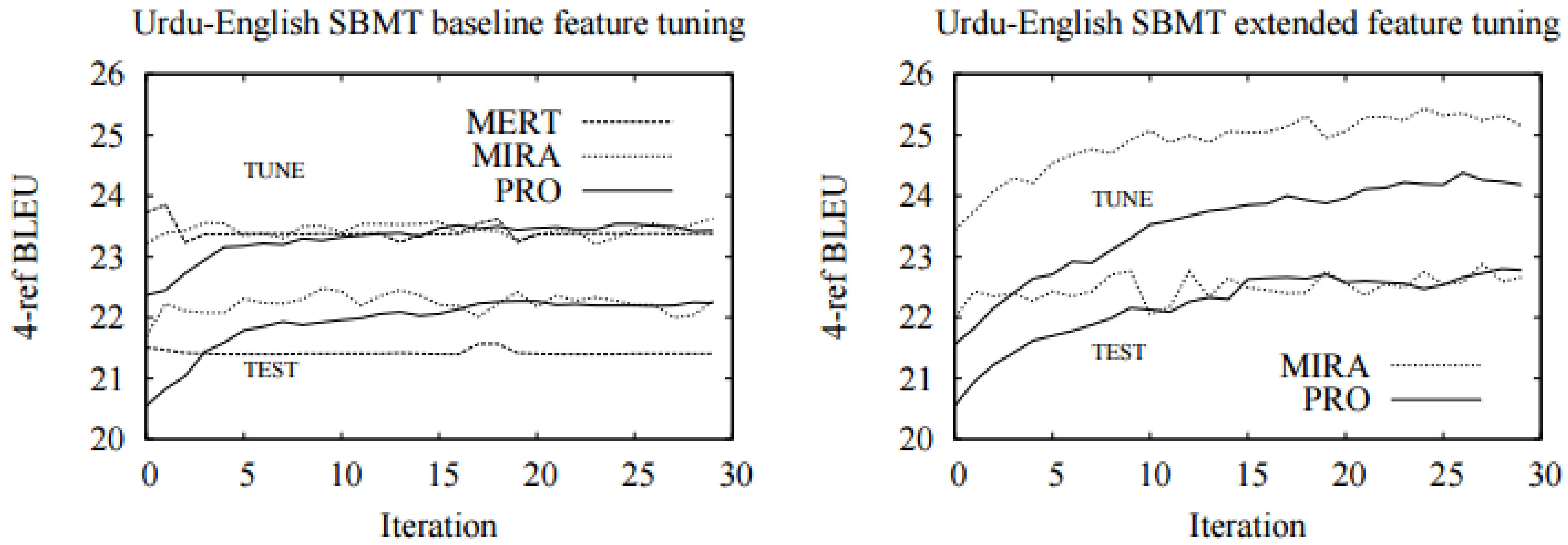
- Baseline feature set
  - Correspond to a typical small feature set in MT literature
  - Gives a low (around 20) dimensional candidate space
- Extended feature set
  - Only used with MIRA and PRO
  - Gives a high (thousands) dimensional candidate space

Results

PBMT				
Language	Experiment feats	Experiment method	BLEU tune	BLEU test
Urdu-English	base	MERT	20.5	17.7
		MIRA	20.5	17.9
		PRO	20.4	18.2
	ext	MIRA	21.8	17.8
		PRO	21.6	18.1
Arabic-English	base	MERT	46.8	41.2
		MIRA	47.0	41.1
		PRO	46.9	41.1
	ext	MIRA	47.5	41.7
		PRO	48.5	41.9
Chinese-English	base	MERT	23.8	22.2
		MIRA	24.1	22.5
		PRO	23.8	22.5
	ext	MIRA	24.8	22.6
		PRO	24.9	22.7

SBMT				
Language	Experiment feats	Experiment method	BLEU tune	BLEU test
Urdu-English	base	MERT	23.4	21.4
		MIRA	23.6	22.3
		PRO	23.4	22.2
	ext	MIRA	25.2	22.8
		PRO	24.2	22.8
Arabic-English	base	MERT	44.7	39.0
		MIRA	44.6	39.0
		PRO	44.5	39.0
	ext	MIRA	45.8	39.8
		PRO	45.9	40.3
Chinese-English	base	MERT	25.5	22.7
		MIRA	25.4	22.9
		PRO	25.5	22.9
	ext	MIRA	26.0	23.3
		PRO	25.6	23.5

Monotonicity



Summary

Successes of this Publication

- Thorough explanation of background and concepts
- Appears to perform comparably to contemporary systems
- Illustrates idea of mapping to a well-solved problem
- Surprisingly good results by solving an apparently simpler problem
- Source code not released, which is a pity
- Comparisons to alternative baselines might be interesting