Tuning SMT Systems on the Training Set

Chris Dyer, Patrick Simianer, Stefan Riezler, Phil Blunsom, Eva Hasler

Project Report
MT Marathon 2011
FBK Trento

Tuning SMT Systems on the Training Set

Goal: Discriminative training using **sparse features** on the **full training set**

Tuning SMT Systems on the Training Set

Goal: Discriminative training using **sparse features** on the **full training set**

Approach: Picky-picky / elitist learning:

Tuning SMT Systems on the Training Set

Goal: Discriminative training using **sparse features** on the **full training set**

Approach: Picky-picky / elitist learning:

- Stochastic learning with **true random selection of examples**.

Tuning SMT Systems on the Training Set

Goal: Discriminative training using **sparse features** on the **full training set**

Approach: Picky-picky / elitist learning:

- Stochastic learning with **true random selection of examples**.
- **Feature selection** according to various regularization criteria.

Tuning SMT Systems on the Training Set

Dyer, Simianer, Riezler, Blunsom, Hasler

Goal: Discriminative training using **sparse features** on the **full training set**

Approach: Picky-picky / elitist learning:

- Stochastic learning with **true random selection of examples**.
- **Feature selection** according to various regularization criteria.
- **Leave-one-out estimation**: Leave out sentence/shard currently being trained on when extracting rules/features in training.

SMT Framework + Data

- cdec decoder (https://github.com/redpony/cdec)

SMT Framework + Data

- cdec decoder (https://github.com/redpony/cdec)
- Hiero SCFG grammars

SMT Framework + Data

- cdec decoder (https://github.com/redpony/cdec)
- Hiero SCFG grammars
- WMT11 news-commentary corpus

SMT Framework + Data

- cdec decoder (https://github.com/redpony/cdec)
- Hiero SCFG grammars
- WMT11 news-commentary corpus
    - 132,755 parallel sentences

SMT Framework + Data

- cdec decoder (https://github.com/redpony/cdec)
- Hiero SCFG grammars
- WMT11 news-commentary corpus
  - 132,755 parallel sentences
  - German-to-English

Learning Framework: SGD for Pairwise Ranking

Algorithm extended ranking voted perceptron: training

$D = \{D^1, ..., D^M\}$: Development set
$C^m = \{c_1^m, ..., c_N^m\}$: the original N-best list of D^m
c_n^m: n-th candidate in C^m

$X^m = \{x_1^m, ...x_N^m\}$: (reordered) N-best list of D^m
x_i^m: i-th candidate in the (reordered) N-best list X^m
$Ranking(W, C^m)$: returns N-best list of C^m reordered
  based on the score, $s_n^m = < W, \phi(c_n^m) >$
$\phi(x_n^m)$: the feature vector of x_n^m

W: weight vector
$V = \{V_1, ...V_T\}$: set of weight vectors
T: Number of pre-defined iteration

For $t = 1, \ldots, T$
  For $m = 1, \ldots, M$;; for each sample in dev-set
    $X^m \leftarrow Ranking(W, C^m)$
    For $i = 1, \ldots, |X^m|$
      For $j = i + 1, \ldots, |X^m|$
        If $(BLEU(x_j^m) > BLEU(x_i^m)$
            $\& \; WER(x_j^m) <= WER(x_i^m))$
          $s = (BLEU(x_j^m) - BLEU(x_i^m))$
          $W = W + s * (\phi(x_j^m) - \phi(x_i^m))$
        End If
      End For
    End For
    $V_t = W$
  End For
End For
Return V

Constraint Selection = Sampling of Pairs

- Random sampling of pairs from full chart for pairwise ranking:

Constraint Selection = Sampling of Pairs

- Random sampling of pairs from full chart for pairwise ranking:
  - First sample translations according to their model score.

Constraint Selection = Sampling of Pairs

- Random sampling of pairs from full chart for pairwise ranking:
  - First sample translations according to their model score.
  - Then sample pairs.

Constraint Selection = Sampling of Pairs

Dyer, Simianer, Riezler, Blunsom, Hasler

- Random sampling of pairs from full chart for pairwise ranking:
  - First sample translations according to their model score.
  - Then sample pairs.
- Sampling will diminish problem of learning to discriminate translations that are too close (in terms of sentence-wise approx. BLEU) to each other.

Constraint Selection = Sampling of Pairs

Dyer, Simianer, Riezler, Blunsom, Hasler

- Random sampling of pairs from full chart for pairwise ranking:
  - First sample translations according to their model score.
  - Then sample pairs.
- Sampling will diminish problem of learning to discriminate translations that are too close (in terms of sentence-wise approx. BLEU) to each other.
- Sampling will also speed up learning.

Constraint Selection = Sampling of Pairs

- Random sampling of pairs from full chart for pairwise ranking:
  - First sample translations according to their model score.
  - Then sample pairs.
- Sampling will diminish problem of learning to discriminate translations that are too close (in terms of sentence-wise approx. BLEU) to each other.
- Sampling will also speed up learning.
- Lots of variations on sampling possible ...

Candidate Features

- Efficient computation of features from local rule context:

Candidate Features

- Efficient computation of features from local rule context:
  - Hiero SCFG rule identifier

Candidate Features

- Efficient computation of features from local rule context:
  - Hiero SCFG rule identifier
  - target n-grams within rule

Candidate Features

- Efficient computation of features from local rule context:
  - Hiero SCFG rule identifier
  - target n-grams within rule
  - target n-gram with gaps (X) within rule

Candidate Features

- Efficient computation of features from local rule context:
  - Hiero SCFG rule identifier
  - target n-grams within rule
  - target n-gram with gaps (X) within rule
  - binned rule counts in full training set

Candidate Features

- Efficient computation of features from local rule context:
  - Hiero SCFG rule identifier
  - target n-grams within rule
  - target n-gram with gaps (X) within rule
  - binned rule counts in full training set
  - rule length features

Candidate Features

- Efficient computation of features from local rule context:
  - Hiero SCFG rule identifier
  - target n-grams within rule
  - target n-gram with gaps (X) within rule
  - binned rule counts in full training set
  - rule length features
  - rule shape features

Candidate Features

- Efficient computation of features from local rule context:
  - Hiero SCFG rule identifier
  - target n-grams within rule
  - target n-gram with gaps (X) within rule
  - binned rule counts in full training set
  - rule length features
  - rule shape features
  - word alignments in rules

Candidate Features

- Efficient computation of features from local rule context:
    - Hiero SCFG rule identifier
    - target n-grams within rule
    - target n-gram with gaps (X) within rule
    - binned rule counts in full training set
    - rule length features
    - rule shape features
    - word alignments in rules
- ... and many more!

Feature Selection

Dyer,
Simianer,
Riezler,
Blunsom,
Hasler

- ℓ_1/ℓ_2-regularization

Feature Selection

Dyer, Simianer, Riezler, Blunsom, Hasler

- ℓ_1/ℓ_2-regularization
  - Compute ℓ_2-norm of column vectors ($=$ vector of examples/shards for each of n features), then ℓ_1-norm of resulting n-dimensional vector.

Feature Selection

Dyer, Simianer, Riezler, Blunsom, Hasler

- ℓ_1/ℓ_2-regularization
  - Compute ℓ_2-norm of column vectors ($=$ vector of examples/shards for each of n features), then ℓ_1-norm of resulting n-dimensional vector.

$$\mathbf{W}_a : \begin{bmatrix} 4 & 0 & 0 & 3 \\ 0 & 4 & 3 & 0 \end{bmatrix} \quad \mathbf{W}_b : \begin{bmatrix} 4 & 3 & 0 & 0 \\ 0 & 4 & 3 & 0 \end{bmatrix}$$

$$4 \quad 4 \quad 3 \quad 3 \rightarrow 14 \qquad 4 \quad 5 \quad 3 \quad 0 \rightarrow 12$$

Feature Selection

- ℓ_1/ℓ_2-regularization
  - Compute ℓ_2-norm of column vectors (= vector of examples/shards for each of n features), then ℓ_1-norm of resulting n-dimensional vector.

$$\mathbf{W}_a : \begin{bmatrix} 4 & 0 & 0 & 3 \\ 0 & 4 & 3 & 0 \end{bmatrix} \quad \mathbf{W}_b : \begin{bmatrix} 4 & 3 & 0 & 0 \\ 0 & 4 & 3 & 0 \end{bmatrix}$$

$$\begin{array}{cccc} 4 & 4 & 3 & 3 \rightarrow 14 \end{array} \qquad \begin{array}{cccc} 4 & 5 & 3 & 0 \rightarrow 12 \end{array}$$

- Effect is to choose small subset of features that are useful across all examples/shards

Feature Selection, done properly

- Incremental gradient-based selection of column vectors (Obozinski, Taskar, Jordan: Joint covariant selection and joint subspace selection for multiple classification problems. Stat Comput (2010))

Feature Selection, done properly

- Incremental gradient-based selection of column vectors (Obozinski, Taskar, Jordan: Joint covariant selection and joint subspace selection for multiple classification problems. Stat Comput (2010))

Algorithm 1 Approximate block-Lasso path

Given ϵ and ξ,
while $\lambda^t > \lambda_{\min}$ **do**
  Set $j^* = \operatorname{argmax}_j \|\nabla_{w_j} J(W^t)\|$
  Update $w_{j^*}^{(t+1)} = w_{j^*}^{(t)} - \epsilon u^t$ with $u^t = \frac{\nabla_{w_{j^*}} J}{\|\nabla_{w_{j^*}} J\|}$
  $\lambda^{t+1} = \min\left(\lambda^t, \frac{J(W^t) - J(W^{t+1})}{\epsilon}\right)$
  Add j^* to the active set
  Enforce (4) for covariates in the active set with $\xi_0 = \xi$.
end while

Feature Selection, quick and dirty

- Combine feature selection with averaging:

Feature Selection, quick and dirty

- Combine feature selection with averaging:
  - Keep only those features with large enough ℓ_2-norm computed over examples/shards.

Feature Selection, quick and dirty

- Combine feature selection with averaging:
  - Keep only those features with large enough ℓ_2-norm computed over examples/shards.
  - Then average feature values over examples/shards.

How far did we get in a few days?

- First full training run finished!

How far did we get in a few days?

- First full training run finished!
  - 150k parallel sentences from news commentary data, German-to-English

How far did we get in a few days?

- First full training run finished!
  - 150k parallel sentences from news commentary data, German-to-English
  - pairwise ranking perceptron

How far did we get in a few days?

- First full training run finished!
  - 150k parallel sentences from news commentary data, German-to-English
  - pairwise ranking perceptron
  - sample 100 translations from chart, use all 100*(99)/2 pairs

How far did we get in a few days?

- First full training run finished!
  - 150k parallel sentences from news commentary data, German-to-English
  - pairwise ranking perceptron
  - sample 100 translations from chart, use all 100*(99)/2 pairs
  - OR: use n-best list
  - sparse rule-id features AND/OR dense features

How far did we get in a few days?

Dyer, Simianer, Riezler, Blunsom, Hasler

- First full training run finished!
  - 150k parallel sentences from news commentary data, German-to-English
  - pairwise ranking perceptron
  - sample 100 translations from chart, use all 100*(99)/2 pairs
  - OR: use n-best list
  - sparse rule-id features AND/OR dense features
  - 200 shards (25 machines with 8 cores)

Results

- Still a lot of bugs due to integration of code from different sources

Results

- Still a lot of bugs due to integration of code from different sources
- Infrastructure is working

Results

- Still a lot of bugs due to integration of code from different sources
- Infrastructure is working
- Experiments still running

Results

- Still a lot of bugs due to integration of code from different sources
- Infrastructure is working
- Experiments still running
- Sensible things happening:
  - Best rule $X \rightarrow X_1\ ,\ \text{dass}\ X_2,\ X_1\ \text{that}\ X_2$
  - Bad rule $X \rightarrow X_1\ \text{oder}\ X_2,\ X_1\ \text{and}\ X_2$

Results

- Still a lot of bugs due to integration of code from different sources
- Infrastructure is working
- Experiments still running
- Sensible things happening:
  - Best rule $X \rightarrow X_1$, dass X_2, X_1 that X_2
  - Bad rule $X \rightarrow X_1$ oder X_2, X_1 and X_2
- At the moment still trailing behind MERT ...

Results

- Still a lot of bugs due to integration of code from different sources
- Infrastructure is working
- Experiments still running
- Sensible things happening:
  - Best rule $X \rightarrow X_1\ ,\ \text{dass}\ X_2,\ X_1\ \text{that}\ X_2$
  - Bad rule $X \rightarrow X_1\ \text{oder}\ X_2,\ X_1\ \text{and}\ X_2$
- At the moment still trailing behind MERT ...
- We'll catch up!

Thanks

Thanks to organizers for great opportunity to learn/chat/hobnob!