The Heidelberg University Machine Translation Systems for IWSLT2013

Institute for Computational Linguistics, Heidelberg University, Germany
Patrick Simianer, Laura Jehl, Stefan Riezler
{simianer,jehl,riezler}@cl.uni-heidelberg.de

UNIVERSITÄT HEIDELBERG ZUKUNFT SEIT 1386

We submitted systems for three translation directions: **German-to-English**, **Russian-to-English** and **English-to-Russian**. The focus of our approaches lies on effective usage of the in-domain parallel training data combined with simple scaling of the language and translation models. We use the **training data** to tune parameter weights for **millions of sparse lexicalized features** using **efficient parallelized stochastic learning techniques**. For German-to-English we incorporate **syntax features**. We combine all systems with **large general-domain language models**; For RU↔EN we use more unfiltered data for the TM.

Sparse, lexicalized features attached to SCFG rules

(1) $X \rightarrow$ X_1 hat X_2 versprochen | X_2 promised X_1
(2) $X \rightarrow$ X_1 hat mir versprochen | X_1 promised me X_2
(3) $X \rightarrow$ X_1 versprach X_2 | X_1 promised X_2

Rule identifiers: unique rule identifier

Rule *n*-grams: bigrams in source and target side of a rule, e.g. `hat` X, X `versprochen`

Rule shape: 39 patterns identifying location of sequences of terminal and non-terminal symbols, e.g. (for rule (1))

`NT, term*, NT, term* | NT, term*, NT`

There is a very large number of potential features ($\gg$ than the number of rules in the grammar).

Pairwise-ranking optimization ("dtrain")

$$\begin{aligned} g(\mathbf{x}_1) > g(\mathbf{x}_2) &\Leftrightarrow f(\mathbf{x}_1) > f(\mathbf{x}_2) \\ &\Leftrightarrow f(\mathbf{x}_1) - f(\mathbf{x}_2) > 0 \\ &\Leftrightarrow \boldsymbol{w}\cdot\mathbf{x}_1 - \boldsymbol{w}\cdot\mathbf{x}_2 > 0 \quad (1) \\ &\Leftrightarrow \boldsymbol{w}\cdot\underbrace{(\mathbf{x}_1 - \mathbf{x}_2)}_{=\bar{\mathbf{x}}_i} > 0 \end{aligned}$$

$\mathbf{x}_{1,2}$ feature representations
$g(\cdot)$ (per-sentence) BLEU score
$f(\cdot)$ model score of the decoder
$\boldsymbol{w}$ weight vector
$\mathbf{x}\cdot\mathbf{y}$ vector dot product

Hinge loss for a stochastic pairwise-ranking perceptron

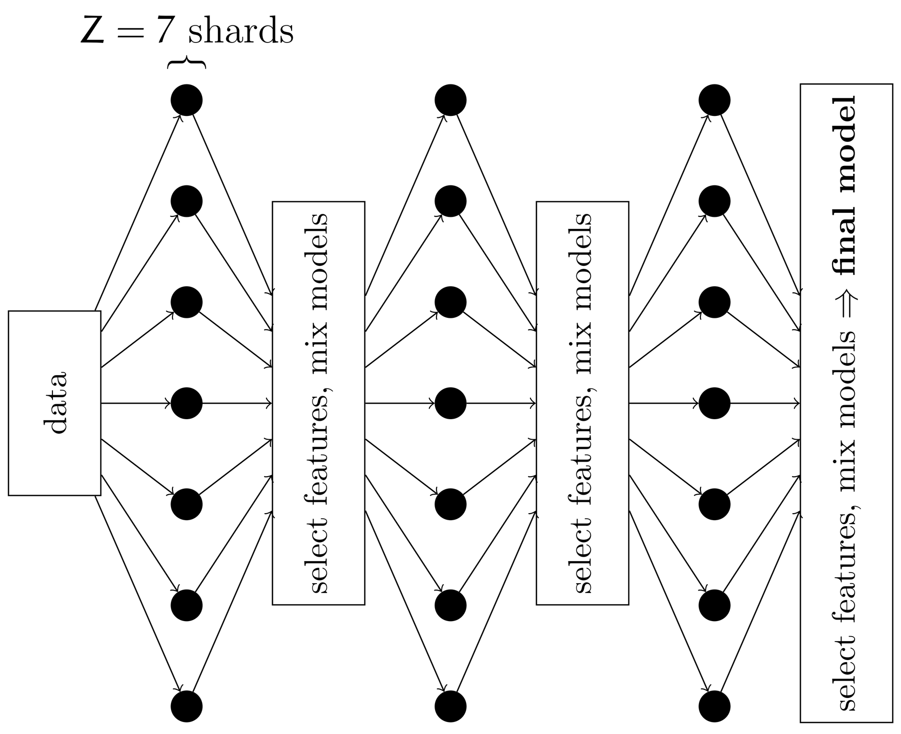
$$L_i(\boldsymbol{w}) = \max(0, -\boldsymbol{w}\cdot\bar{\mathbf{x}}_i) \quad (2)$$

$$\nabla L_i = \begin{cases} -\bar{\mathbf{x}}_i & \text{if } \boldsymbol{w}\cdot\bar{\mathbf{x}}_i \leqslant 0, \\ 0 & \text{otherwise.} \end{cases} \quad (3)$$

Gold standard ranking: BLEU+1 scores of translations of *k*best lists

Tuning on the training set with ℓ_1/ℓ_2 regularization and parallelization

(Simianer et al, 2012)



(a) Parallelization strategy

	f_1	f_2	f_3	f_4	f_5
$\boldsymbol{w}_1$	[5	4	3	4	0]
$\boldsymbol{w}_2$	[2	0	4	1	1]
$\boldsymbol{w}_3$	[4	0	0	3	0]
ℓ_2 norms	9	4	5	5	1
sort	f_1	f_3	f_4	f_2	f_5
select K = 3	f_1	f_3	f_4		
mix	11/3	7/3	7/3		

(b) Feature selection

Figure 1 : Visualization of the learning algorithm

- Randomly split data into Z shards
- Select top K feature columns that have highest ℓ_2 norm over shards (or equivalently, by setting a threshold λ)
- Average weights of selected features over shards
- Resend reduced weight vector to shards for new epoch

SMT Setup

- `cdec` SCFG decoder (Dyer et al, 2009)
- Word alignments with a variant of IBM's model 2 (Dyer et al, 2013)
- Hiero grammars (2 non-terminals max., ...) built with impl. of the suffix array extraction technique of (Lopez, 2007)
- Language models built with `lmplz` (Heafield, 2013)
- Tokenization, compound splitting and recasing with `moses tools`

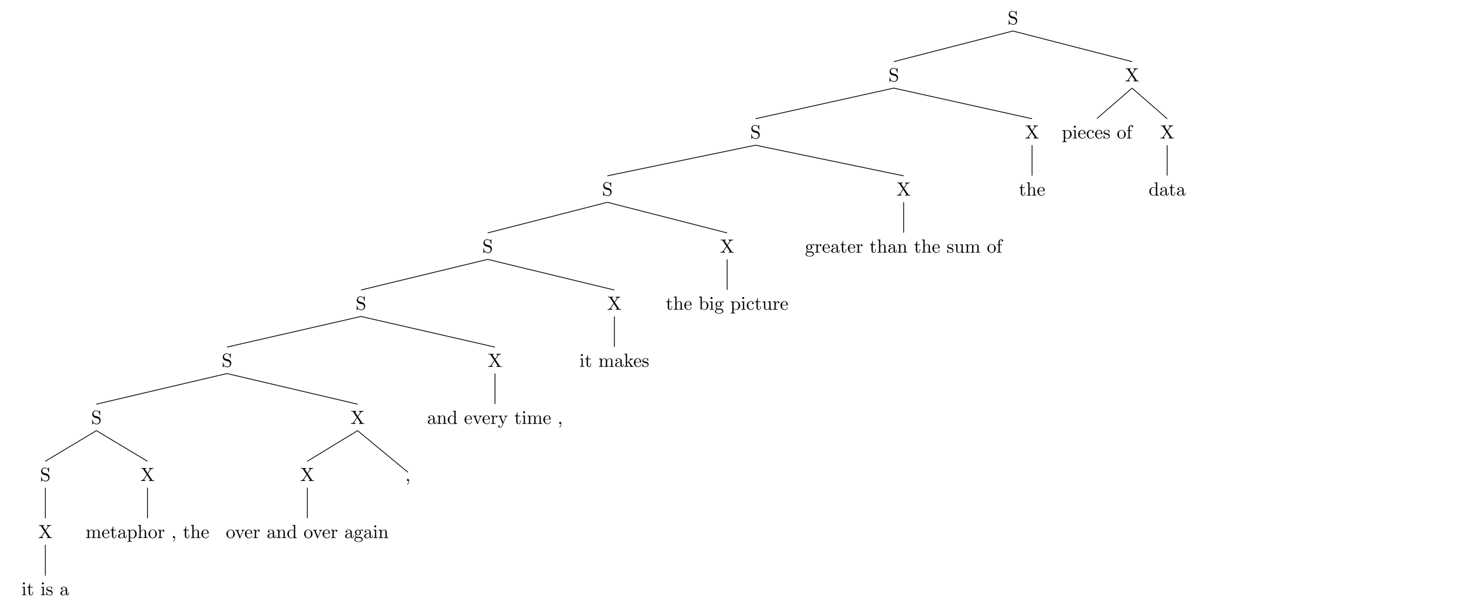
(Simianer et al, 2012) *Joint Feature Selection in Distributed Stochastic Learning for Large-Scale Discriminative Training in SMT*; **(Dyer et al, 2010)** *cdec: A Decoder, Alignment, and Learning framework for finite-state and context-free translation models*; **(Dyer et al, 2013)** *A Simple, Fast, and Effective Reparameterization of IBM Model 2*; **(Lopez, 2007)** *Hierarchical Phrase-Based Translation with Suffix Arrays*; **(Heafield, 2013)** *Efficient Language Modeling Algorithms with Applications to Statistical Machine Translation*; **(Marton & Resnik, 2008)** *Soft Syntactic Constraints for Hierarchical Phrased-Based Translation*

Marton & Resnik's (2008) soft-syntactic constraints

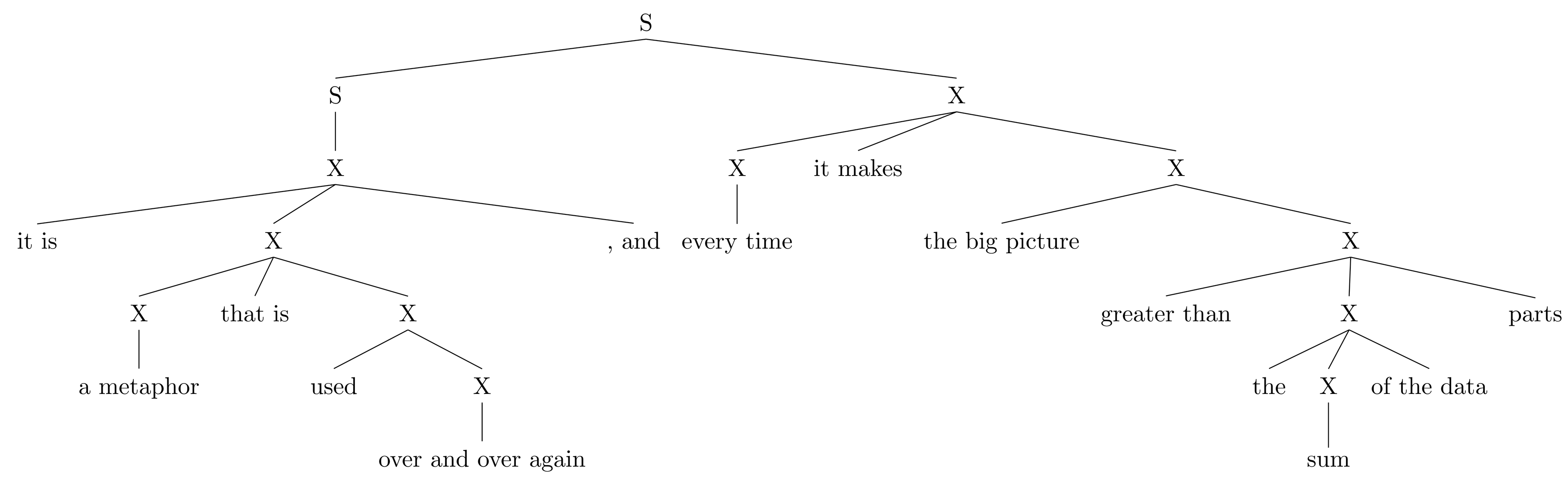
$\{$ADJP,ADVP,CP,DNP,IP,LCP,NP,PP,QP,VP$\} \times \{=,+\}$

- Indicate if spans in decoder derivations **match =** or **cross +** constituents of syntactic trees
- In contrast to the syntax feature in Chiang's original Hiero paper these features do include the actual phrase labels

Effects of soft-syntactic constraints



(a) Baseline derivation with lots of gluing



(b) Derivation using soft-syntactic constraints depicting a sensible parse tree

(Large) Language and Translation Models

German-to-English TM: just TED data ⇒ about 150,000 tokens
English LM: 10^9 FR-EN, Europarl, News Commentary, News Crawl, UN corpus, LDC2011T07 ⇒ 7,245,227,502 tokens
Russian↔English TM: Common Crawl, Yandex 1M, News Commentary, Wiki Headlines, TED data ⇒ 44,042,275 Russian and 48,677,800 English tokens
Russian LM: Common Crawl, News Commentary, Yandex 1M, News Crawl, TED data ⇒ 335,023,785 tokens

Development Results (`tst2010`)

results on `tst2010`; * primary/† secondary submission; *baseline* is a standard system with dense features trained with MERT on the dev set

German-to-English:

System	TED 4-gram LM	Large 5-gram LM
baseline	26.7	+1.7
dtrain-dev	+0.9	+2.1
dtrain-train(clustered)*	+1.3	**+2.9**
dtrain-train+soft-syntax†	+1.4	-

Russian-to-English:

System	TED 4-gram LM	Large 5-gram LM
baseline	17.0	+0.5
dtrain-dev	+0.2	+0.8
dtrain-dev+large TM+large LM	–	+3.1
dtrain-train†	+0.7	+1.4
dtrain-train+large LM+large TM*	–	**+3.7**

English-to-Russian:

System	TED 4-gram LM	Large 5-gram LM
baseline	12.4	+0.7
baseline+large TM	+0.1	+1.1
dtrain-dev	+0.4	+1.3
dtrain-dev+large TM*	+0.7	**+2.4**
dtrain-train†	-0.6	+0.8

Official Results for Primary Submissions (`tst2013`)

- German-to-English: 23.06/22.91* (24.07)
- Russian-to-English: 23.78 (25.00)
- English-to-Russian: 15.87 (15.95)

lowercase scores in brackets; * calculated with disfluencies in the references