The HDU Discriminative SMT System for Constrained Data PatentMT at NTCIR10

Patrick Simianer, Gesa Stupperich, Laura Jehl,
Katharina Wäschle, Artem Sokolov, Stefan Riezler

Institute for Computational Linguistics, Heidelberg University, Germany

Outline

The HDU discriminative SMT system

Intuition: Patents have a twofold nature; They are . . .

1. **easy to translate:** repetitive and formulaic text
2. **hard to translate:** long sentences and unusual jargon

Method: Discriminative SMT

1. **Training:** multi-task learning with large, sparse feature sets via ℓ_1/ℓ_2 regularization
2. **Syntax features:** soft-syntactic constraints for complex word order differences in long sentences

Subtasks/results

Participation in **Chinese-to-English** (ZH-EN) and **Japanese-to-English** (JP-EN) PatentMT subtasks

- **Constrained data** situation where only the parallel corpus provided by the organizers was used
- **Results:**

JP-EN Rank 5 **(constrained: 2)** with regard to **BLEU** on the *Intrinsic Evaluation* (IE) test set; *IE adequacy 8th, IE acceptability 6th*

ZH-EN Rank 9 **(constrained: 3)** for the ZH-EN translation subtask on this subtask's IE test set; *IE adequacy 4th, IE acceptability 4th*

Hierarchical phrase-based translation

(1) $X \rightarrow$ X_1 要件 の X_2 | X_2 of X_1 requirements
(2) $X \rightarrow$ この とき 、 X_1 は | this time , the X_1 is
(3) $X \rightarrow$ テキスト メモリ 41 に X_1 | X_1 in the text memory 41

- Synchronous CFG with rules encoding hierarchical phrases (Chiang, 2007; Adam Lopez, 2007)
- `cdec` decoder (Dyer et al., 2010)

Online pairwise-ranking optimization

$$\overbrace{g(\mathbf{x_1}) > g(\mathbf{x_2})}^{\text{ranking by BLEU should agree with ...}} \Leftrightarrow \overbrace{f(\mathbf{x}_1) > f(\mathbf{x}_2)}^{\text{the model score of the decoder}}$$

$$\Leftrightarrow f(\mathbf{x}_1) - f(\mathbf{x}_2) > 0$$

$$\Leftrightarrow \mathbf{w} \cdot \mathbf{x}_1 - \mathbf{w} \cdot \mathbf{x}_2 > 0$$

$$\Leftrightarrow \underbrace{\mathbf{w} \cdot (\mathbf{x}_1 - \mathbf{x}_2) > 0}_{\text{this can reformulated as a binary classification problem}}$$

- For large feature sets we train a **pairwise ranking** model using algorithms for stochastic gradient descent
- Gold standard training data is obtained by calculating per-sentence BLEU scores of translations of *k*best lists
- Simplest case: several runs of the perceptron algorithm over a single development set
- (data-) Parallelized by sharding (**multi-task learning**), incorporating ℓ_1/ℓ_2 regularization

Online pairwise-ranking optimization

$$\overbrace{g(\mathbf{x_1}) > g(\mathbf{x_2})}^{\text{ranking by BLEU should agree with ...}} \Leftrightarrow \overbrace{f(\mathbf{x}_1) > f(\mathbf{x}_2)}^{\text{the model score of the decoder}}$$
$$\Leftrightarrow f(\mathbf{x}_1) - f(\mathbf{x}_2) > 0$$
$$\Leftrightarrow \mathbf{w}\cdot\mathbf{x}_1 - \mathbf{w}\cdot\mathbf{x}_2 > 0$$
$$\Leftrightarrow \underbrace{\mathbf{w}\cdot(\mathbf{x}_1 - \mathbf{x}_2) > 0}_{\text{this can reformulated as a binary classification problem}}$$

- For large feature sets we train a **pairwise ranking** model using algorithms for stochastic gradient descent
- Gold standard training data is obtained by calculating per-sentence BLEU scores of translations of *k*best lists
- Simplest case: several runs of the perceptron algorithm over a single development set
- (data-) Parallelized by sharding (**multi-task learning**), incorporating ℓ_1/ℓ_2 regularization

Online pairwise-ranking optimization

$$\overbrace{g(\mathbf{x_1}) > g(\mathbf{x_2})}^{\text{ranking by BLEU should agree with ...}} \Leftrightarrow \overbrace{f(\mathbf{x}_1) > f(\mathbf{x}_2)}^{\text{the model score of the decoder}}$$

$$\Leftrightarrow f(\mathbf{x}_1) - f(\mathbf{x}_2) > 0$$

$$\Leftrightarrow \mathbf{w} \cdot \mathbf{x}_1 - \mathbf{w} \cdot \mathbf{x}_2 > 0$$

$$\Leftrightarrow \underbrace{\mathbf{w} \cdot (\mathbf{x}_1 - \mathbf{x}_2) > 0}_{\text{this can reformulated as a binary classification problem}}$$

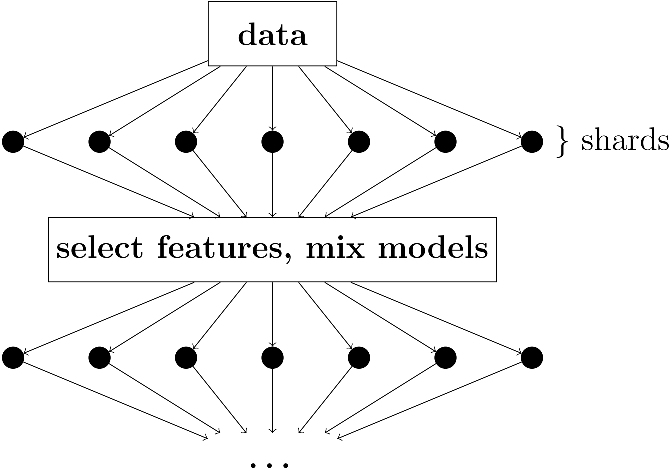
- For large feature sets we train a **pairwise ranking** model using algorithms for stochastic gradient descent
- Gold standard training data is obtained by calculating per-sentence BLEU scores of translations of *k*best lists
- Simplest case: several runs of the perceptron algorithm over a single development set
- (data-) Parallelized by sharding (**multi-task learning**), incorporating ℓ_1/ℓ_2 regularization

Algorithm for Multi-Task Learning

- Randomly split data into Z shards
- Run optimization on each shard separately for one iteration
- Collect and stack resulting weight vectors
- 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 $k \leftarrow 1 \ldots K$ over shards

$$\mathbf{v}[k] = \frac{1}{Z} \sum_{z=1}^{Z} \mathbf{W}[z][k]$$

- Resend reduced weight vector $\mathbf{v}$ to shards for new iteration


data
} shards
select features, mix models
...

Feature sets

12 **dense features** of the default translation model

- **Sparse lexicalized features**, defined locally on SCFG rules:

  Rule identifiers: unique rule identifier

  Rule *n*-grams: bigrams in source and target side of a rule, e.g. `of` X_1, X_1 `requirements`

  Rule shape: 39 patterns identifying location of sequences of terminal and non-terminal symbols, e.g. `NT, term*, NT -- NT, term*, NT, term*`

  (1) $X \rightarrow X_1$ 要件 の $X_2 \mid X_2$ of X_1 requirements

- **Soft-syntactic constraints** on source side:
  - 20 features for matching/non-matching of 10 most common constituents (Marton and Resnik, 2008)

Marton & Resnik's soft-syntactic constraints

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

- These features indicate if spans in parses of the decoder **match =** or **cross +** constituents in syntactic trees
- We compare these on the source of the data; syntactic trees are pre-computed; lookup is done online
- In contrast to (Chiang, 2005) these features include the actual phrase labels

JP-EN: System Setup

Training data: three million parallel sentences of NTCIR10, **constrained data**

Standard SMT pipeline: `GIZA` word alignments; `MeCab` for Japanese segmentation; `moses tools` for English; lowercased models; 5gram `SRILM` language model; grammars with max. two non-terminals

Extensive preprocessing

HDU-1 Multi-task training with **sparse rule features** combining all four available development sets

HDU-2 Identical to HDU-1 but training stopped early

JP-EN: Preprocessing

- English tokenization: we slightly extended the non-breaking prefixes list (e.g. including FIG., PAT., . . .)
- **Consistent tokenization** (Ma and Matsoukas, 2011)
  - Training data was aligned using regular expressions
  - In test and development data we use the most common variant observed in training data
- Applied a compound splitter to split **Katakana terms** (Feng et al., 2011) to decrease the number of OOVs

JP-EN: Development tuning

	tuning set			
tuning method	*dev1*	*dev2*	*dev3*	*dev1,2,3*
MERT baseline (avg)	27.85	27.63	27.6	27.76
single dev, dense	27.83	–	–	–
single dev, +sparse	28.84	28.08	28.71	29.03
multi-task, +sparse	–	–	–	28.92

ZH-EN: System Setup

Training and development data of NTCIR10 (one million/2000 parallel sentences), **constrained setup**

Standard SMT pipeline, segmentation of Chinese with the Stanford Segmenter, **no additional preprocessing**

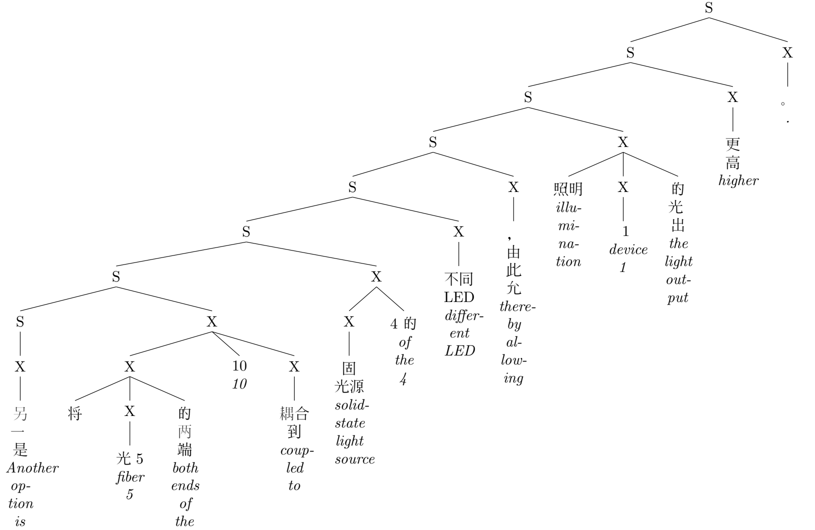
HDU-1 Marton & Resnik's soft-syntactic features, 20 additional weights tuned with MERT

HDU-2 System as JP-EN with sparse rule features, but unregularized training on a single development set

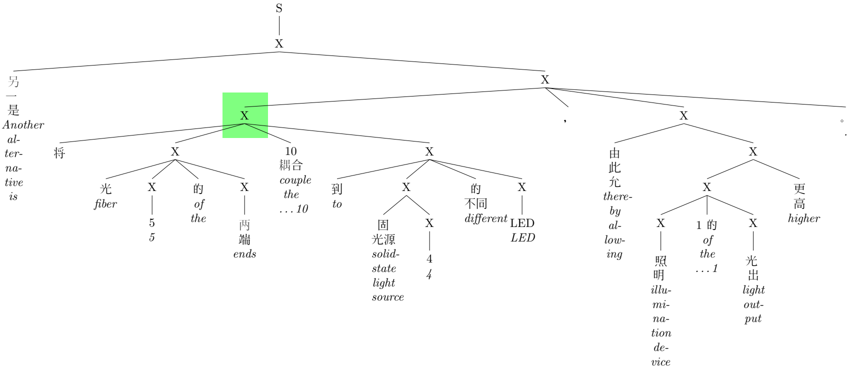
Effects of soft-syntactic constraints I

baseline	Another option is coupled to both ends of the fiber . . . , thereby allowing . . .
soft-syntax	Another alternative is to couple the ends of the fiber . . . , thereby allowing . . .
reference	A further option is to optically couple both ends 10 of the optical fiber 5 . . . , thus allowing . . .

Effects of soft-syntactic constraints II



Effects of soft-syntactic constraints III



The HDU discriminative SMT system: Conclusion

- We achieved solid results for both subtasks with good automatic and manual evaluation results
- Training a model of **sparse features** is a very good approach for patent translation, with improvements of about 1 BLEU point by just enabling them
- **Multi-task learning** enables the use of more training data, newer experiments even point to further possibilities of improvement with this technique
- **Soft-syntactic constraints** show the desired effect, incorporating proper syntax into Hiero models, leading to better translations (and prettier derivations!)

References I

Adam Lopez. Hierarchical phrase-based translation with suffix arrays. Technical report, University of Maryland, College Park, 2007.

David Chiang. A hierarchical phrase-based model for statistical machine translation. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, 2005.

David Chiang. Hierarchical phrase-based translation. *Computational Linguistics*, 33(2), June 2007.

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

References II

Minwei Feng, Christoph Schmidt, Joern Wuebker, Stephan Peitz, Markus Freitag, and Hermann Ney. The RWTH Aachen system for NTCIR-9 PatentMT, 2011.

Jeff Ma and Spyros Matsoukas. BBN's systems for the Chinese-English sub-task of NTCIR-9 PatentMT evaluation, 2011.

Yuval Marton and Philip Resnik. Soft syntactic constraints for hierarchical phrased-based translation. In *Proceedings of ACL 2008*, 2008.