Joshua 2.0: A Toolkit for Parsing-Based Machine Translation with Syntax, Semirings, Discriminative Training and Other Goodies

Zhifei Li, Chris Callison-Burch, Chris Dyer[1], Juri Ganitkevitch, Ann Irvine, Sanjeev Khudanpur, Lane Schwartz[2], Wren N. G. Thornton[3], Ziyuan Wang, Jonathan Weese, and Omar F. Zaidan

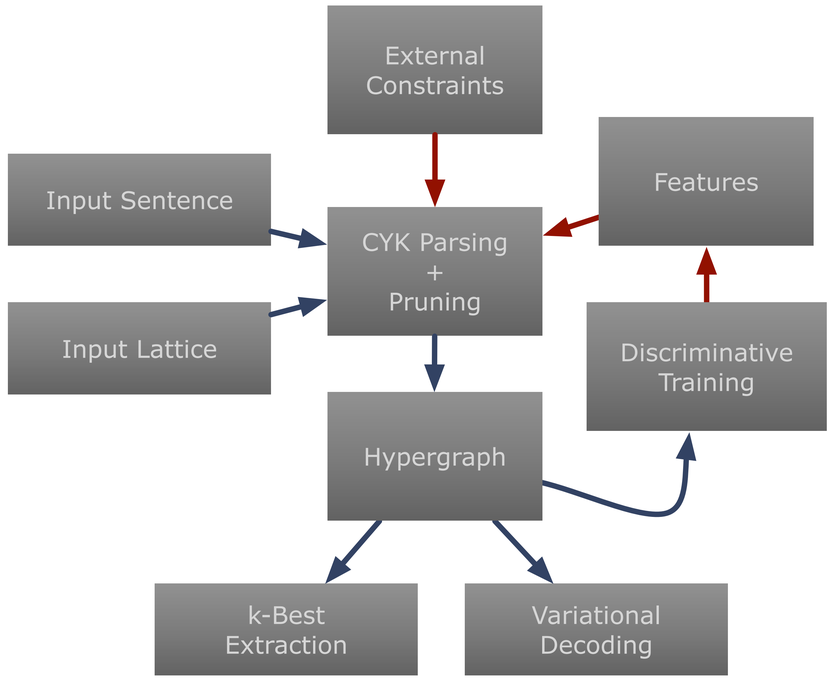
Johns Hopkins University - [1]University of Maryland - [2]University of Minnesota - [3]University of Indiana

JOHNS HOPKINS UNIVERSITY

Joshua Toolkit Features

- A fully featured parsing-based decoder
  - Chart-parsing, n-gram language model integration, beam and cube pruning, k-best extraction, parallel and distributed decoding
  - Easy to extend due to modular design
  - Scalable and fast
  - On-the-fly suffix-array grammar extraction
- Includes minimum error rate training (MERT)
  - Optimizes decoder feature weights
  - Easy to add new metrics
  - Highly parallelized, supports document-level metrics (Zaidan, 2010)
  - Also available as a standalolne toolkit:
    http://cs.jhu.edu/~ozaidan/zmert
- Visualization tools (Weese & Callison-Burch, 2010)
  - GUIs for visualizing parse trees and hypergraphs, facilitating grammar analysis and debugging
- Comes with MT pipeline management
  - Make-based pipeline (Schwartz, 2010)
  - Integrates with LoonyBin (Clark et al., 2010)
- Incorporates external constraints (Irvine et al., 2010)
  - Enables integration of specialized modules into the decoding process as soft and hard constraints
  - Example uses: transliteration, specialized rules for named entities
- Decodes with syntax-based translation grammars
  - Support for arbitrary non-terminal labels
  - Incorporates syntactic information crucial for reordering
- Efficiently handles lattice input
  - Crucial for integration with speech recognition, segmenters or for translation via a bridge language
- Implements Viterbi k-best extraction and variational decoding on the hypergraph
- Discriminative minimum-risk training framework based on gradient descent algorithms capable of handling a large number of features

The Chart-Parsing Decoder

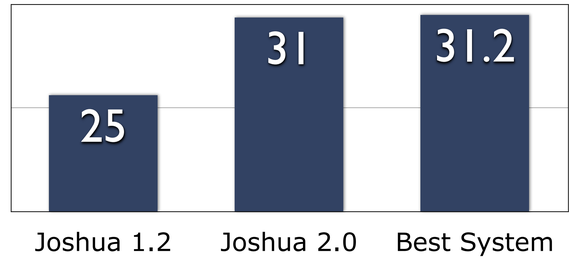


- CYK parser extended to efficiently handle input text and lattices (Dyer et al., 2008)
- High-cost constituents are pruned during parsing
- Information from external modules (such as transliteration) is incorporated by converting XML markup into rules that are considered by the decoder
  - Supports hard per word span constrains as well as soft constraints that compete with regular grammar rules
- Parsing results are stored in a hypergraph
  - Supports Viterbi k-best extraction and variational decoding on the hypergraph (Li et al., 2009)
  - Includes semiring framework for computations on the hypergraph, with implementations of the inside and outside algorithms (Li & Eisner, 2009)
- The discriminative training framework allows for training the weights for a large number of features

Performance

- Joshua achieves state-of-the-art translation quality

NIST Urdu-English (BLEU)



- In WMT10, the Joshua system ranked best in TER for the German-English translation task

Source	Target	BLEU	TER
de	en	21.3	**0.660**
en	de	15.2	0.738
fr	en	27.7	0.614
en	fr	23.8	0.681
es	en	29.0	0.595
en	es	28.1	0.596

Getting Started With Joshua

- Download the latest Joshua release or check out the most recent version via:

  svn co https://joshua.svn.sf.net/svnroot/joshua/trunk joshua
- Set up a baseline MT system:
  - Prepare monolingual and bilingual training data
  - Train a language model using the SRILM toolkit
  - Train a translation model
    - Sub-sample bilingual data (optional)
    - Create word alignments using the Berkeley Aligner
    - Run suffix-array grammar extraction
  - Perform minimum error rate training
  - Decode test sets
- For a detailed walkthrough refer to:

  http://cs.jhu.edu/~ccb/joshua/