Jane: A Guide to RWTH's Hierarchical Machine Translation Toolkit

Daniel Stein, David Vilar, Stephan Peitz and Hermann Ney

jane@i6.informatik.rwth-aachen.de

`http://www.hltpr.rwth-aachen.de/jane`

Sep 2010

Human Language Technology and Pattern Recognition
Lehrstuhl für Informatik 6
Computer Science Department
RWTH Aachen University, Germany

1 Introduction

- Hierarchical phrase-based translation toolkit, including
  - Phrase extraction
  - Decoding
  - MERT training
- Toolkit written in C++, with tools in Python and Bash/Zsh
- Focus on efficiency and flexibility
- Parallelized operation under the Sun Grid Engine
- Extensions include syntax augmented models, advanced lexicon models, MIRA, . . .
- Jane is open-source for non-commercial purposes
- `http://www.hltpr.rwth-aachen.de/jane`

Outline

1 Introduction

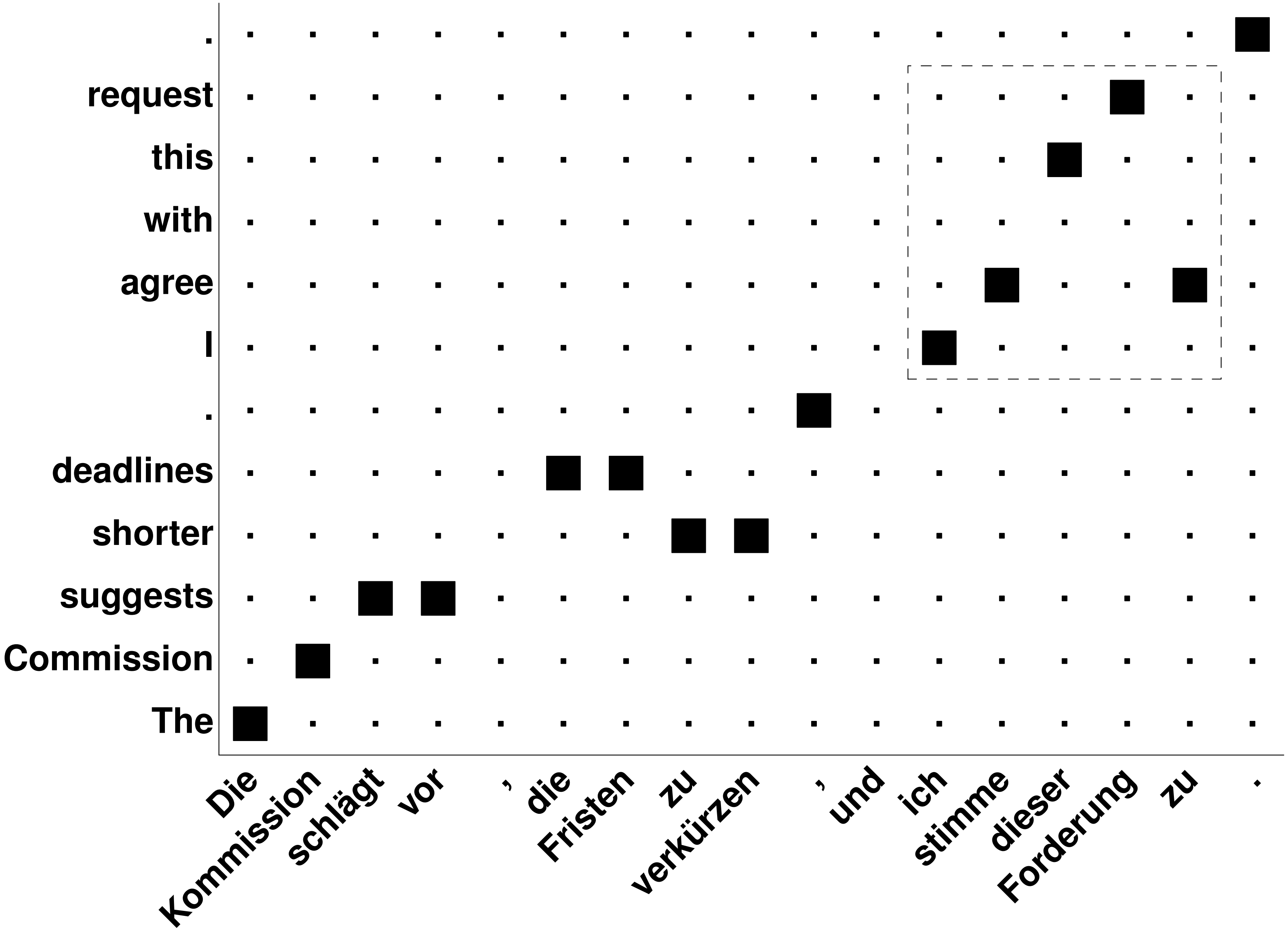
2 Hierarchical Phrases

3 Extraction

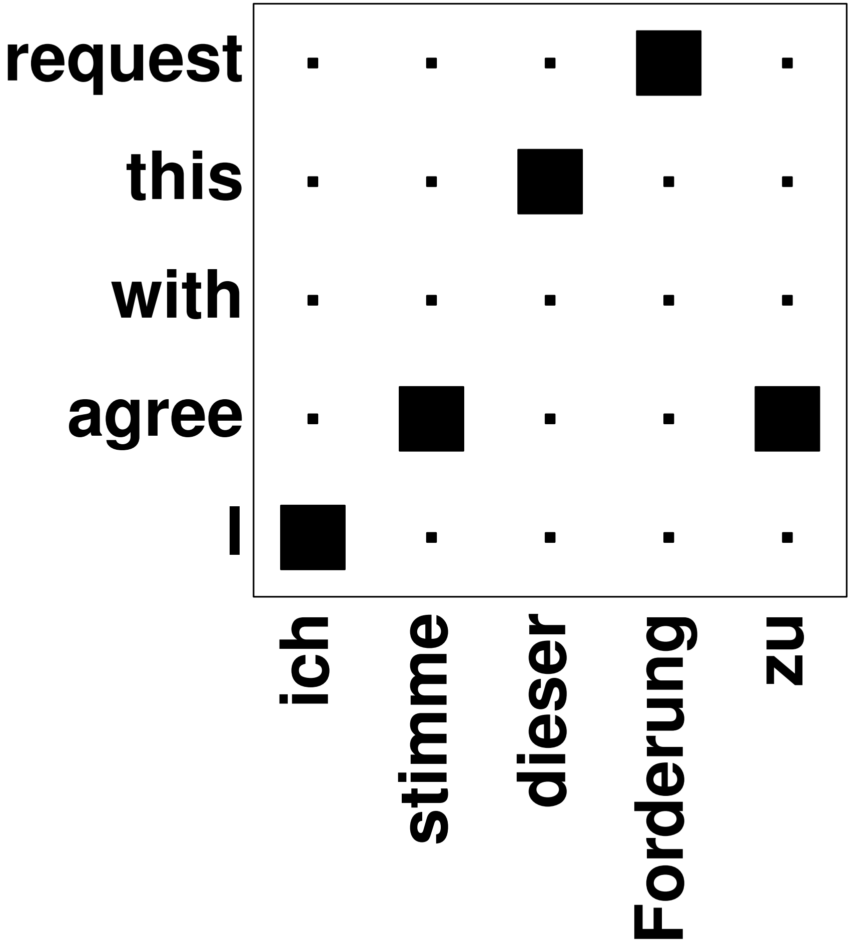
4 Translation

5 Conclusions

2 Hierarchical Phrases

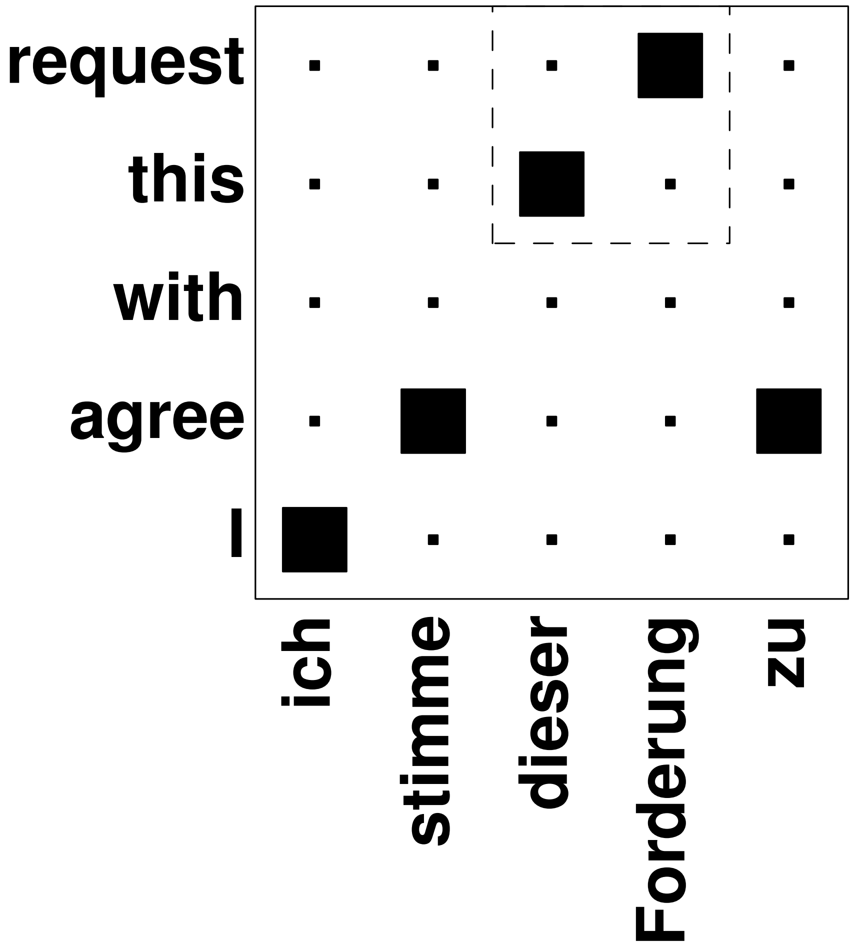


Illustration

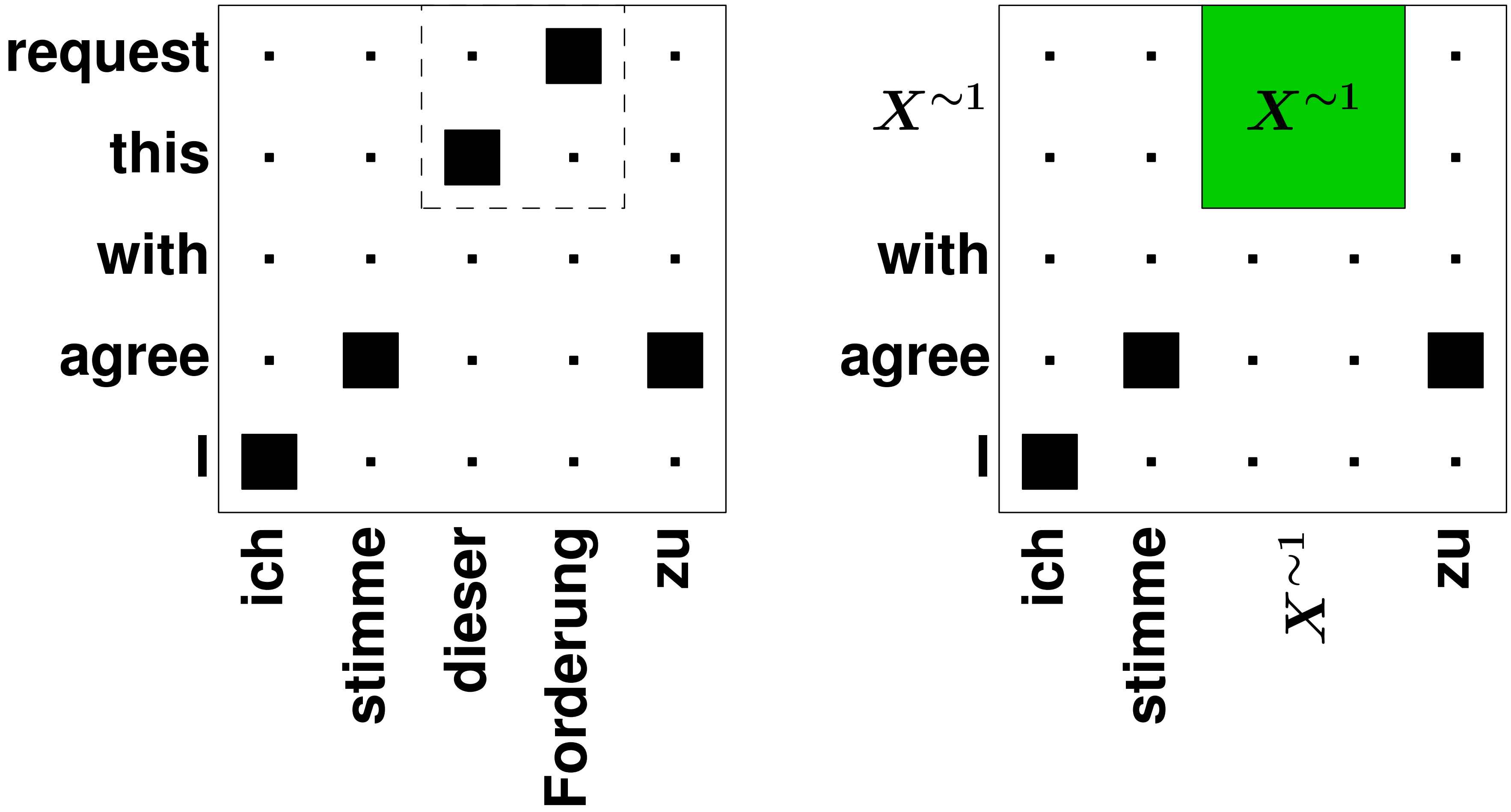


	ich	stimme	dieser	Forderung	zu
request	▪	▪	▪	■	▪
this	▪	▪	■	▪	▪
with	▪	▪	▪	▪	▪
agree	▪	■	▪	▪	■
I	■	▪	▪	▪	▪

Illustration



Illustration



Hierarchical Phrases

- Formalization as a synchronous CFG
- Rules of the form $X \rightarrow \langle \gamma, \alpha, \sim \rangle$, where:
  - X is a non-terminal
  - γ and α are strings of terminals and non-terminals
  - $\sim$ is a one-to-one correspondence between the non-terminals of α and γ
- Example:

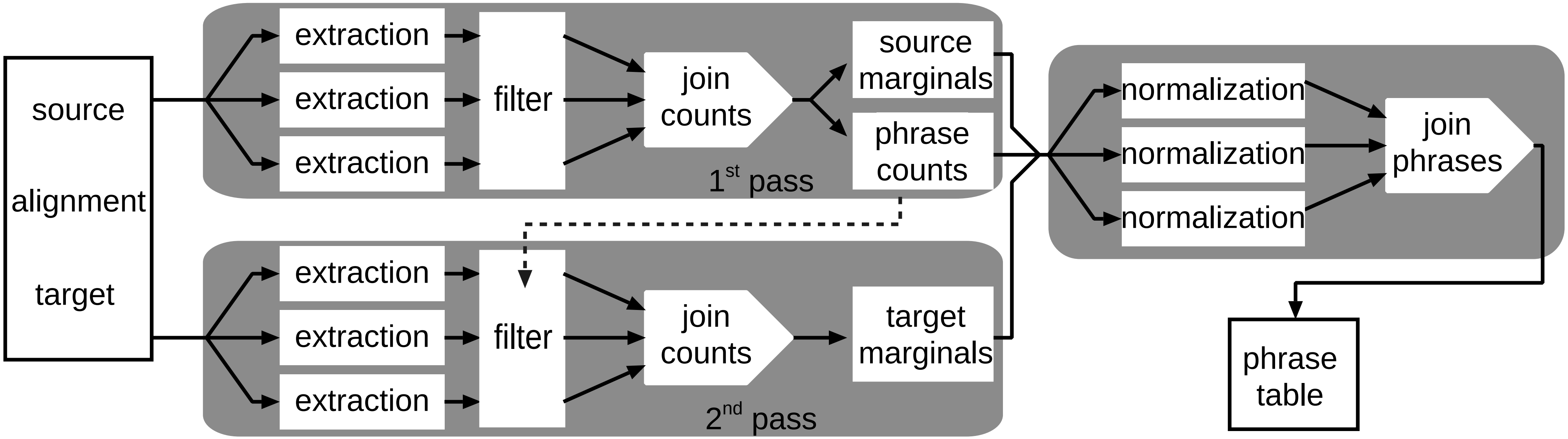
$$X \rightarrow \langle \text{Ich stimme } X^{\sim 1} \text{ zu, I agree with } X^{\sim 1} \rangle$$
$$X \rightarrow \langle \text{weil andere } X^{\sim 1} \text{ nicht } X^{\sim 2}\text{, because others have not } X^{\sim 2}\ X^{\sim 1} \rangle$$

- Additionally: Glue rules

$$S \rightarrow \langle S^{\sim 1} X^{\sim 2}, S^{\sim 1} X^{\sim 2} \rangle$$
$$S \rightarrow \langle X^{\sim 1}, X^{\sim 1} \rangle$$

3 Extraction

- **Parallelized extraction and normalization of counts**



- **2-pass extraction for filtering the target marginals**

Additional Models

- Modular implementation of additional features
- Example usage (config file):

```
source=f.gz
target=e.gz
alignment=Alignment.gz
filter=devAndTest

additionalModels="syntax,parsematch"
extractOpts="--syntax.targetParsefile target.tree \
   --parsematch.sourceParseFile source.tree \
   --parsematch.targetParseFile targetTree"
```

DIY: Additional Models

- **Inherit from:**

  AdditionalExtractionInformationCreator **Produces instances of AdditionalExtractionInformation**
  Main functions:

  `newSentence` **Notifies of a new sentence pair**

  `processCount` **Called when a new phrase(-count) is created**

  AdditionalExtractionInformation **Wrapper class for the additional information required for the feature**
  Main functions:

  `add` **Combines two instances of the class**
  (e.g. the same feature is extracted from two different sentence pairs)

  `writePlain` **For writing the information to disk**

  `writePlainFinal` **For writing the normalized score (if needed)**

- **Add your model to** `AdditionalExtractionInformationFactory.cc`

Additional Models

Already implemented

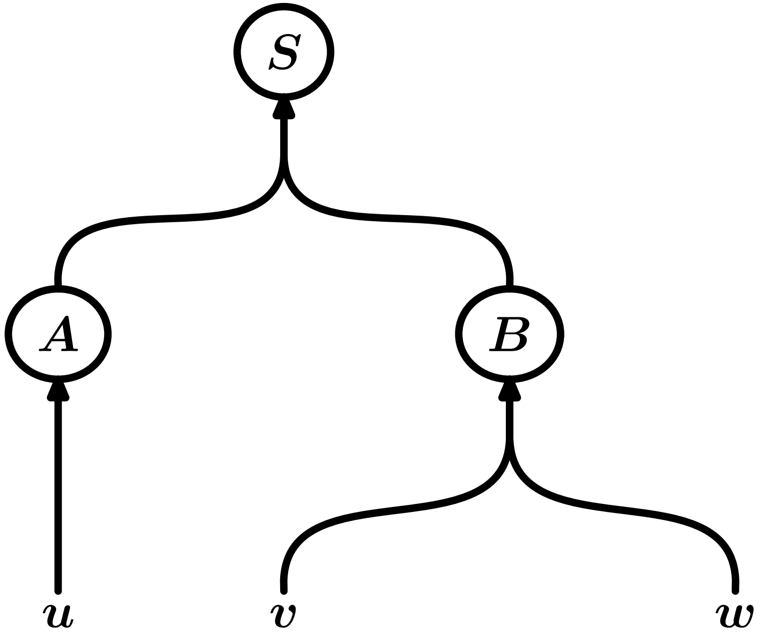
- **Soft syntactic labels [Venugopal & Zollmann$^+$ 09]**
- **Dependency information [Shen & Xu$^+$ 08]**
- **Parsematch information [Vilar & Stein$^+$ 08]**
- **Heuristic extraction features (non-aligned words, single word phrases, etc.)**
- **Alignment information**

4 Translation

- Three running modes:
  - Single best translation
  - n-best translation
  - Server mode
- Cube pruning and cube growing
- On-demand loading of phrases for reduced memory footprint
- Four LM formats
  - Arpa
  - SRI binary format
  - RandLM
  - In-house binary format with on-demand loading
- Arbitrary number of LMs in search
- Sentence-level parallelization (Sun Grid Engine)

Translation: Principles

- Two passes: parsing and LM computation
- Parsing
  - CYK+ algorithm
  - Generation of an hypergraph
  - No LM scores are taken into account (directly)
  - Translations only implicitly computed

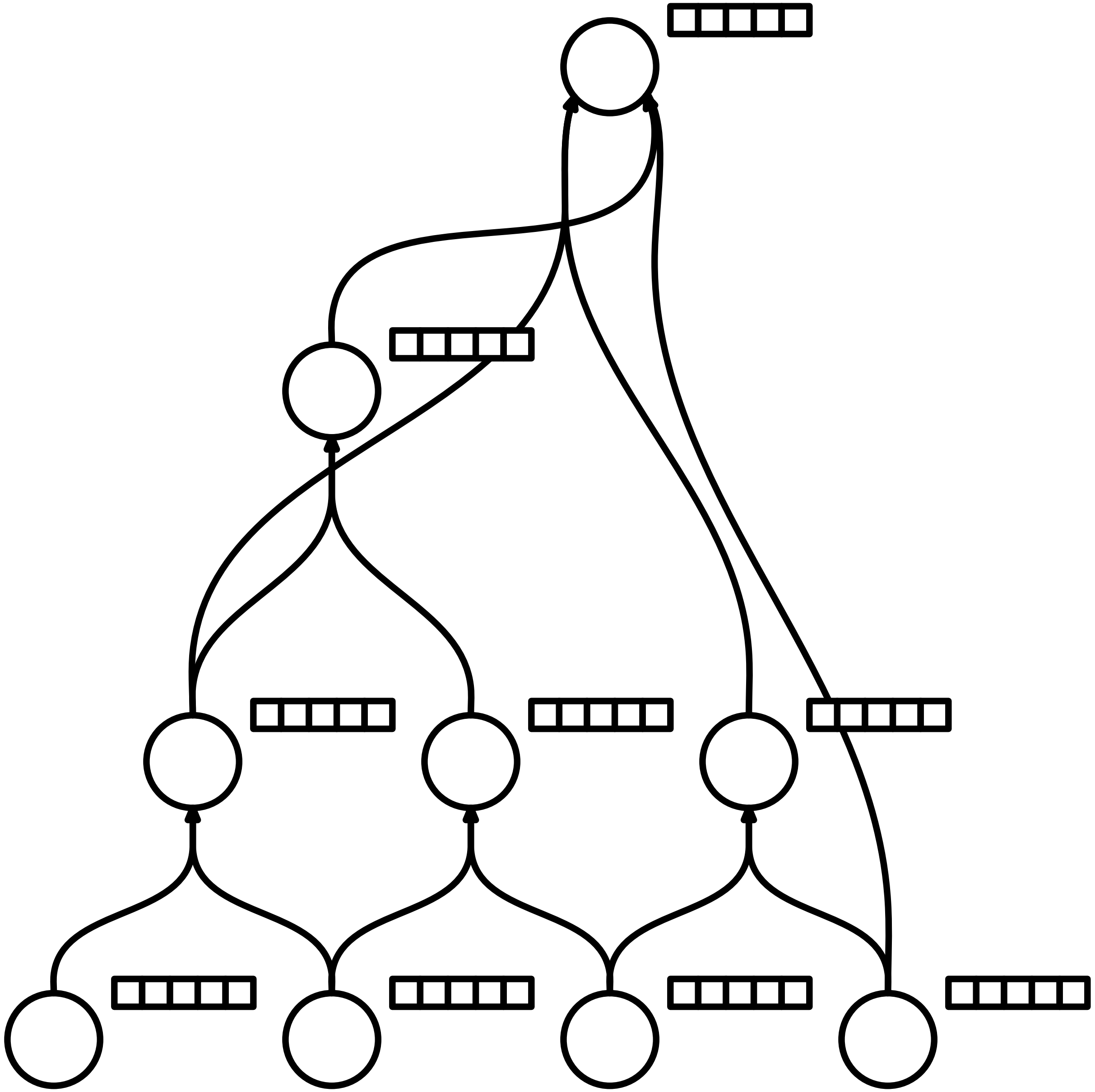


$$S \rightarrow AB$$
$$A \rightarrow u$$
$$B \rightarrow vw$$

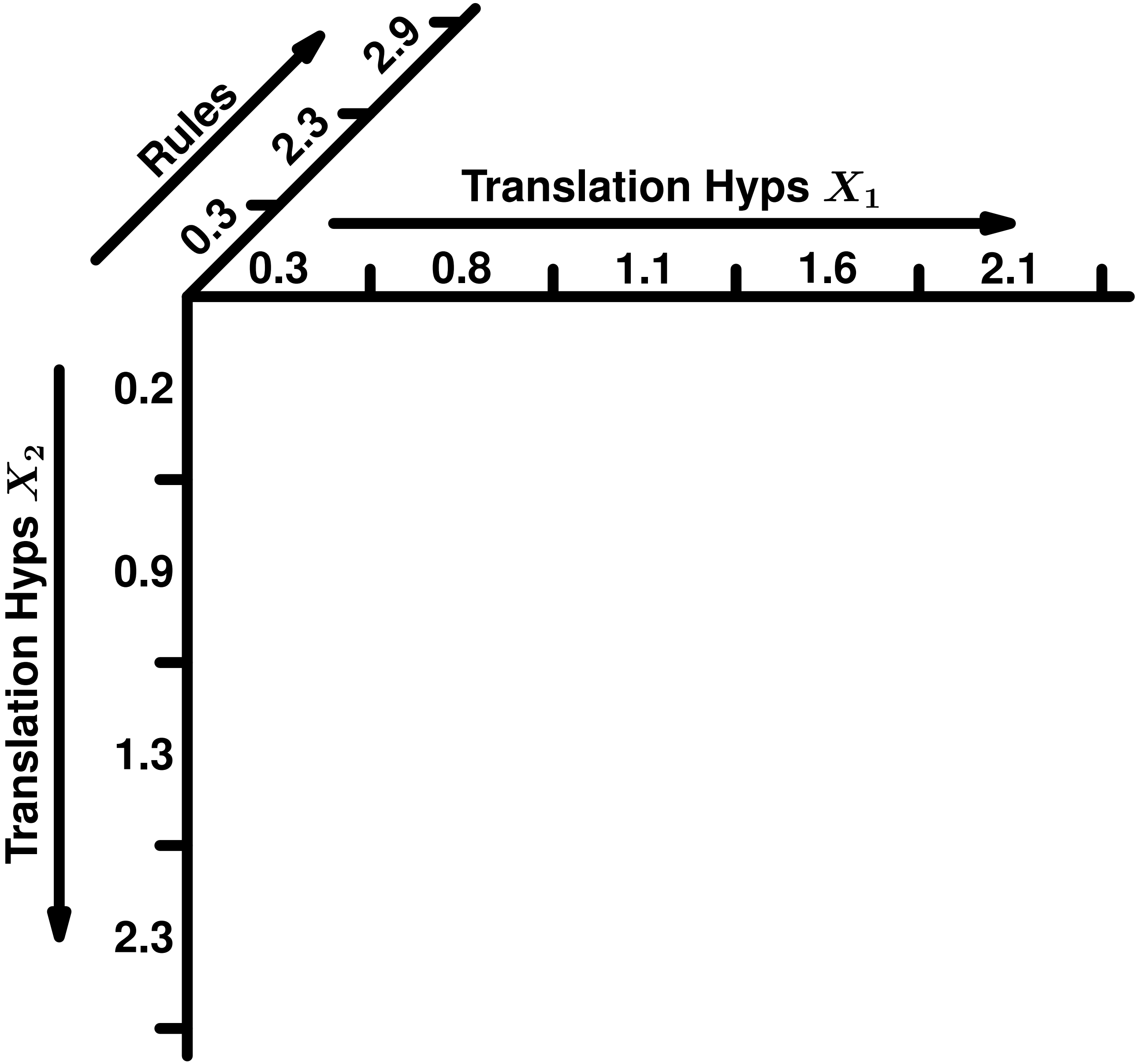
Translation: Principles

- LM computation
  - Traverse the hypergraph and compute n-best lists of derivations
    - Fixed size: cube pruning
    - On demand: cube growing

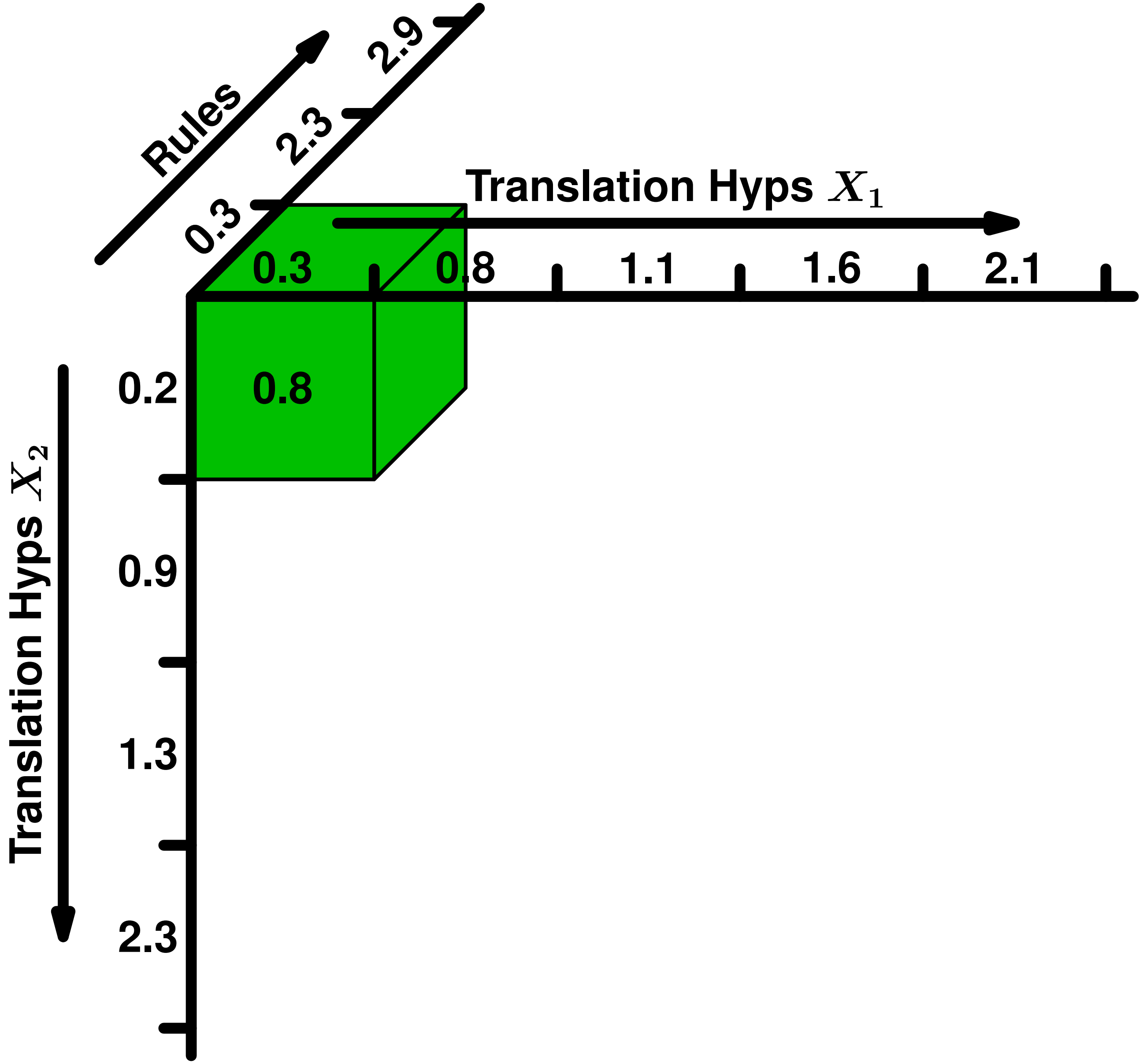
Cube Pruning



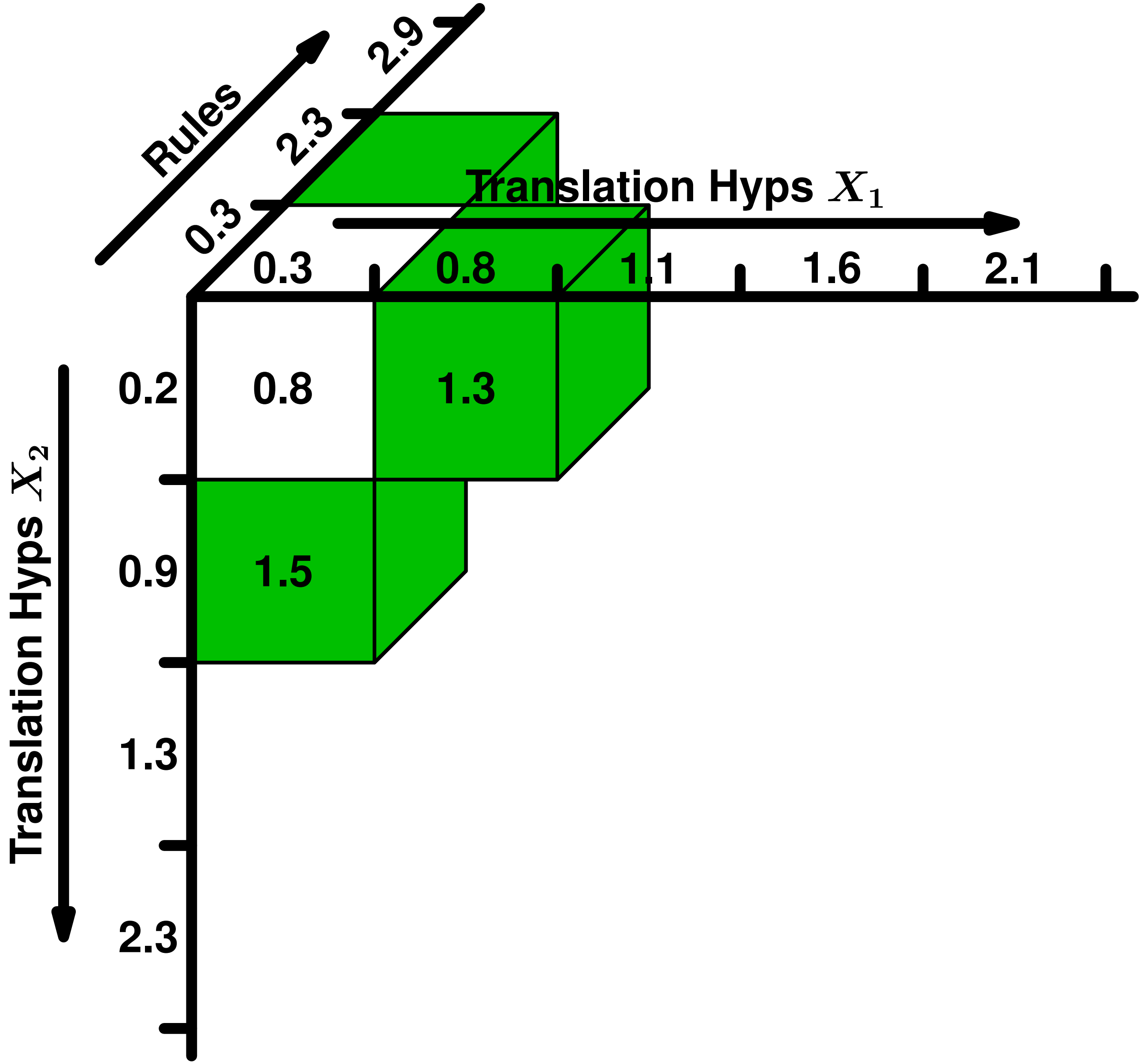
Cube Pruning



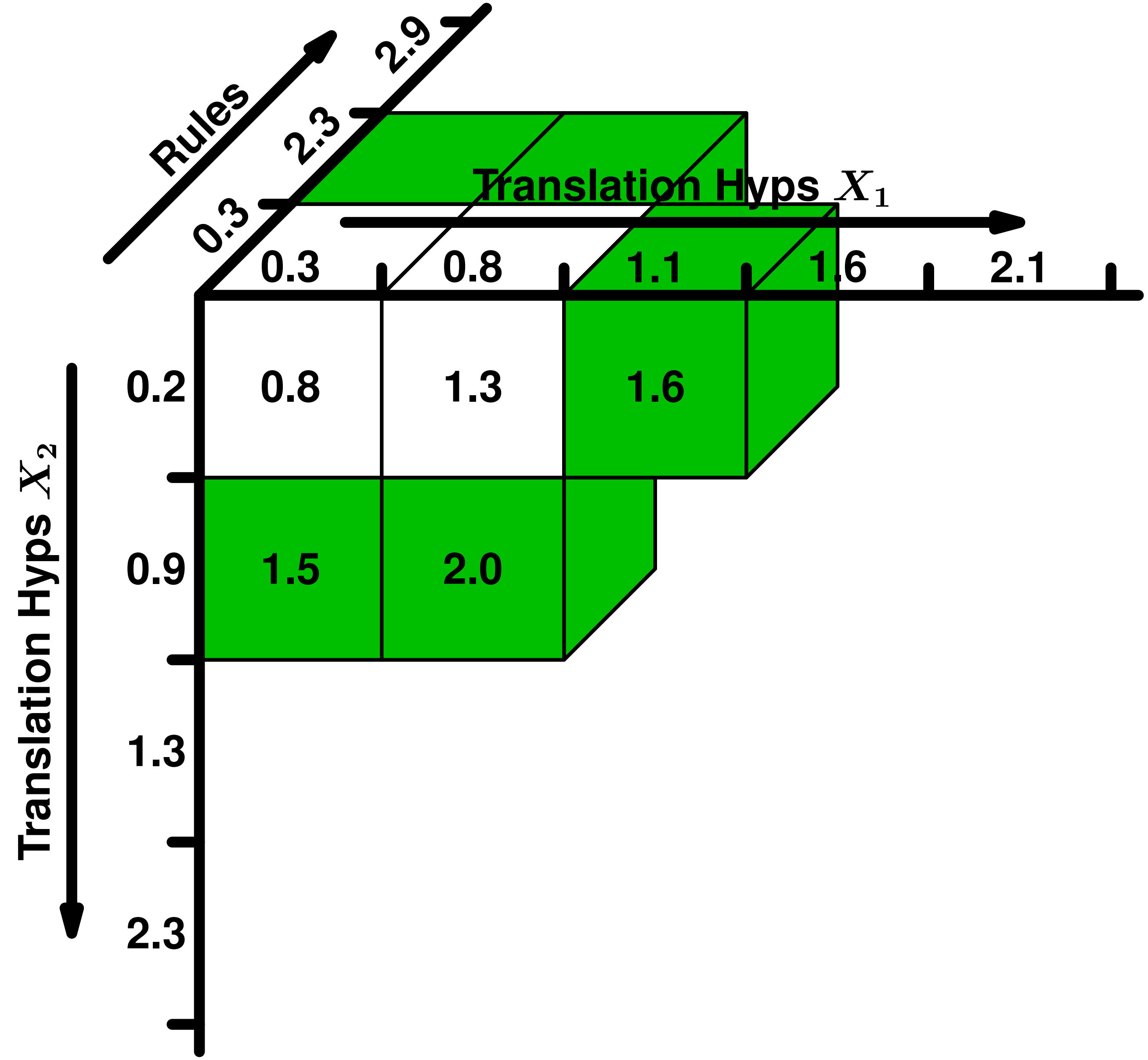
Cube Pruning



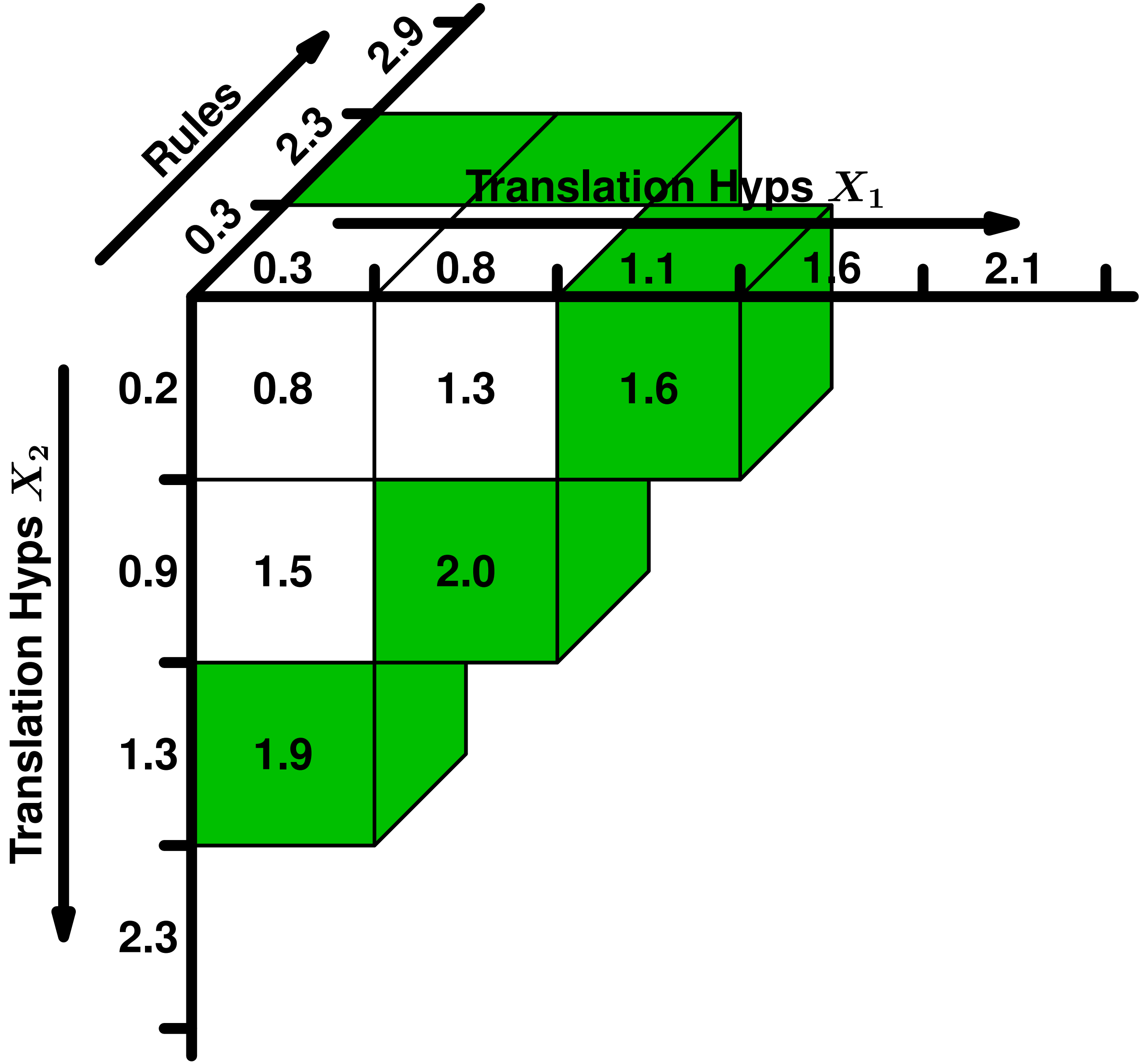
Cube Pruning



Cube Pruning



Cube Pruning



Additional Models

- **Modular implementation of additional features**
- **Example usage (config file):**

```
[Jane]
decoder = cubePrune

[Jane.singleBest]
fileIn = f-dev
fileOut = f-dev.hyp

[Jane.CubePrune]
generationNbest = 100
secondaryModels = Syntax

[Jane.CubePrune.rules]
file = rules.bin
```

```
[Jane.CubePrune.LM]
file = lm.5gram.gz

[Jane.scalingFactors]
phraseS2T = 0.0391947693
phraseT2S = 0.0160933791
ibm1S2T = 0.0353934023
...
LM = 0.0881110196
Syntax = 0.00236455511
syntaxPenalty = 0.0628653661
```

DIY: Additional Models

- Inherit from:

  SecondaryModel Computes additional scores

  Main functions:

  `newSentence` Notifies of a new sentence to translate

  `scoreBackpointer` Compute scores for a derivation

- Add your model to `SecondaryModelCreator.hh`

Additional Models

Already implemented

- **Extended lexicon models [Mauser & Hasan$^+$ 09]**
- **Soft syntax labels [Venugopal & Zollmann$^+$ 09]**
- **Dependency models [Stein & Peitz$^+$ 10] based on [Shen & Xu$^+$ 08]**
- **Reordering models [Vilar & Stein$^+$ 10]**

5 Conclusions

- Efficient toolkit for hierarchical phrase-based translation
- Easily extensible
- Parallelized operation
- Open source, free for non-commercial use
- `http://www.hltpr.rwth-aachen.de/jane`

Jane: A Guide to RWTH's Hierarchical Machine Translation Toolkit

Daniel Stein, David Vilar, Stephan Peitz and Hermann Ney

jane@i6.informatik.rwth-aachen.de

http://www.hltpr.rwth-aachen.de/jane

Sep 2010

Human Language Technology and Pattern Recognition
Lehrstuhl für Informatik 6
Computer Science Department
RWTH Aachen University, Germany

References

[Mauser & Hasan^{+} 09] A. Mauser, S. Hasan, H. Ney: Extending Statistical Machine Translation with Discriminative and Trigger-Based Lexicon Models. In *Proc. of the Conf. on Empirical Methods for Natural Language Processing (EMNLP)*, pp. 210–218, Singapore, Aug. 2009. 18

[Shen & Xu^{+} 08] L. Shen, J. Xu, R. Weischedel: A New String-to-Dependency Machine Translation Algorithm with a Target Dependency Language Model. In *Proc. of the Annual Meeting of the Association for Computational Linguistics (ACL)*, pp. 577–585, Columbus, Ohio, June 2008. 10, 18

[Stein & Peitz^{+} 10] D. Stein, S. Peitz, D. Vilar, H. Ney: A Cocktail of Deep Syntactic Features for Hierarchical Machine Translation. In *Conference of the Association for Machine Translation in the Americas 2010*, Oct. 2010. 18

[Venugopal & Zollmann^{+} 09] A. Venugopal, A. Zollmann, N. Smith, S. Vogel: Preference Grammars: Softening Syntactic Constraints to Improve Statistical Machine Translation. In *Proc. of the Human Language Technology Conf. / North American Chapter of the Association for Computational Linguistics (HLT-NAACL)*, pp. 236–244, Boulder, Colorado, June 2009. 10, 18

[Vilar & Stein^{+} 08] D. Vilar, D. Stein, H. Ney: Analysing Soft Syntax Features and Heuristics for Hierarchical Phrase Based Machine Translation. In *Proc. of the Int. Workshop on Spoken Language Translation (IWSLT)*, pp. 190–197, Waikiki, Hawaii, Oct. 2008. 10

[Vilar & Stein^{+} 10] D. Vilar, D. Stein, M. Huck, H. Ney: Jane: Open Source Hierarchical Translation, Extended with Reordering and Lexicon Models. In *Proc. of the Workshop on Statistical Machine Translation*, pp. 262–270, Uppsala, Sweden, July 2010. 18