Feature-Rich Translation by Quasi-Synchronous Lattice Parsing

Kevin Gimpel and Noah A. Smith

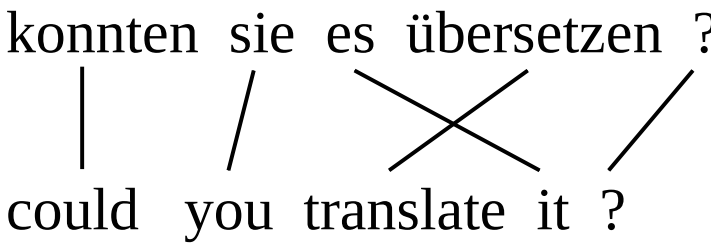
Introduction

- Two trends in machine translation research
  - Many approaches to decoding
    - Phrase-based
    - Hierarchical phrase-based
    - Tree-to-string
    - String-to-tree
    - Tree-to-tree
  - Regardless of decoding approach, addition of richer features can improve translation quality

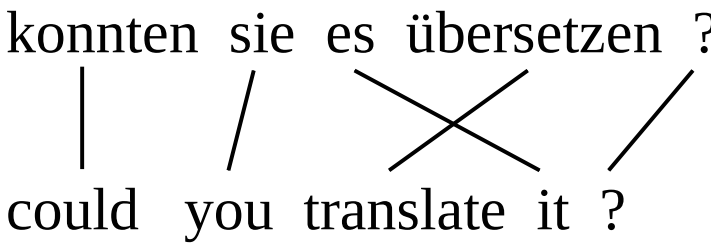
Introduction

- Two trends in machine translation research
  - Many approaches to decoding
    - Phrase-based
    - Hierarchical phrase-based
    - Tree-to-string
    - String-to-tree
    - Tree-to-tree
  - Regardless of decoding approach, addition of richer features can improve translation quality
- Decoding algorithms are strongly tied to features permitted

Phrase-Based Decoding



Phrase-Based Decoding



Phrase Table	
1	konnten → could
2	konnten sie → could you
3	es übersetzen → translate it
4	sie es übersetzen → you translate it
5	es → it
6	? → ?
...	

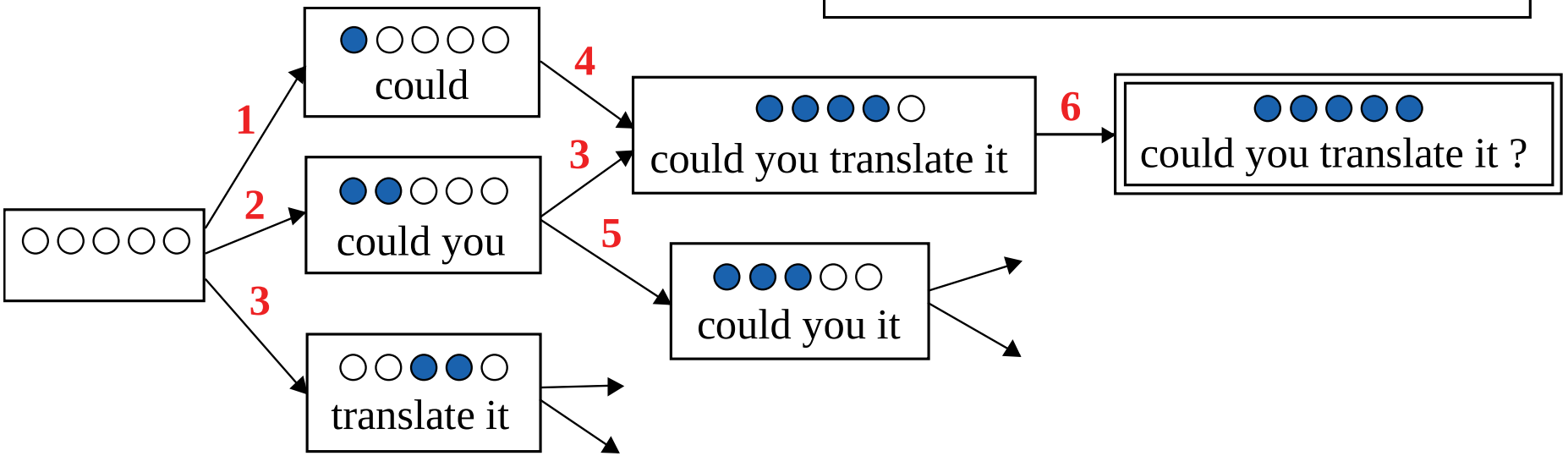
Phrase-Based Decoding

konnten sie es übersetzen ?

could you translate it ?

Phrase Table

1	konnten → could
2	konnten sie → could you
3	es übersetzen → translate it
4	sie es übersetzen → you translate it
5	es → it
6	? → ?
...	



ARK lti Carnegie Mellon

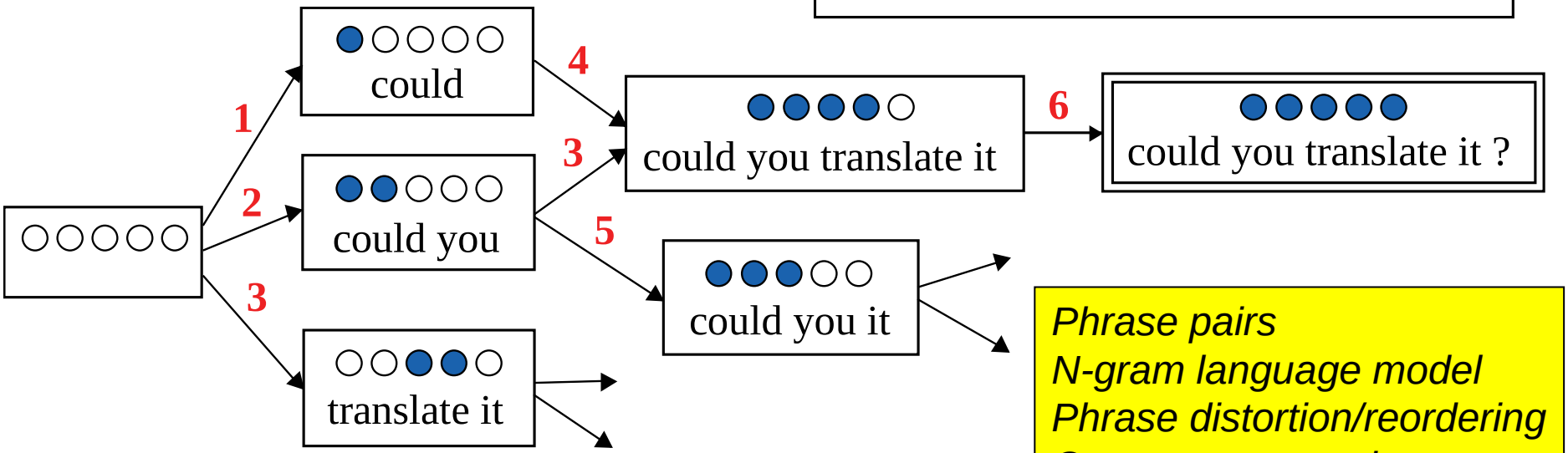
Phrase-Based Decoding

konnten sie es übersetzen ?

could you translate it ?

Phrase Table

1	konnten → could
2	konnten sie → could you
3	es übersetzen → translate it
4	sie es übersetzen → you translate it
5	es → it
6	? → ?
...	



Phrase pairs
N-gram language model
Phrase distortion/reordering
Coverage constraints

ARK lti Carnegie Mellon

Hierarchical Phrase-Based Decoding

$_0$konnten$_1$sie$_2$es$_3$übersetzen$_4$?$_5$

could you translate it ?

SCFG Rules	
1	X → es übersetzen / translate it
2	X → es / it
3	X → übersetzen / translate
4	X → konnten sie X ? / could you X ?
5	X → konnten sie X_1 X_2 ? / could you X_2 X_1 ?
...	

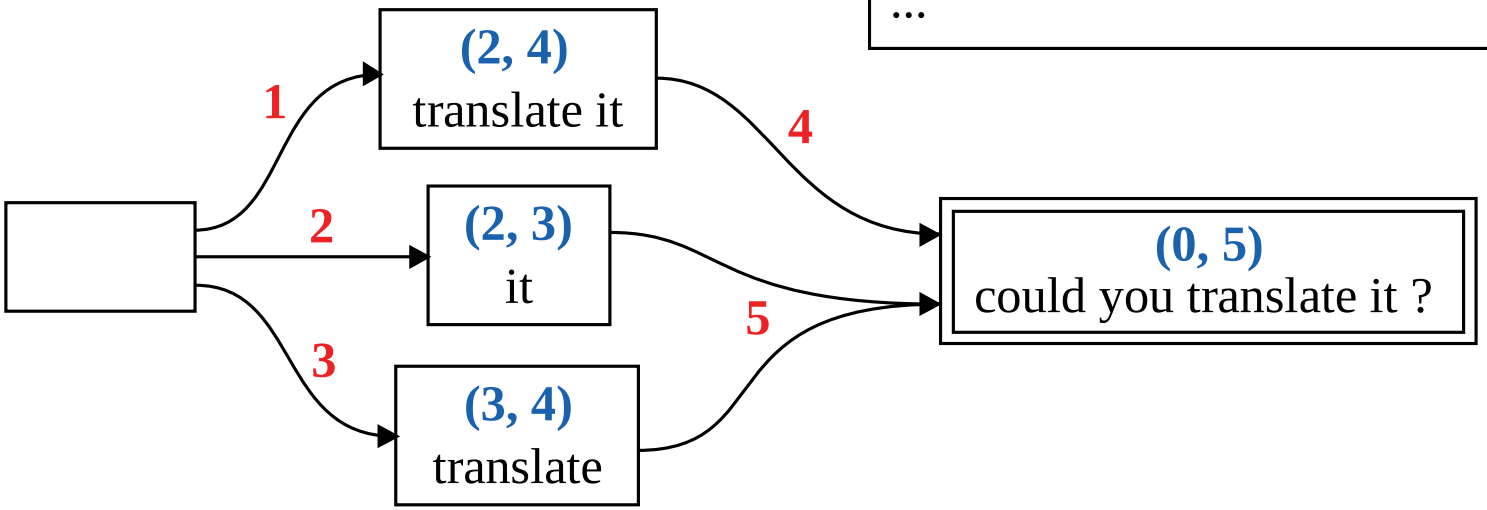
Hierarchical Phrase-Based Decoding

$_0$konnten$_1$sie$_2$es$_3$übersetzen$_4$?$_5$

could you translate it ?

SCFG Rules

1 X → es übersetzen / translate it
2 X → es / it
3 X → übersetzen / translate
4 X → konnten sie X ? / could you X ?
5 X → konnten sie X_1 X_2 ? / could you X_2 X_1 ?
...



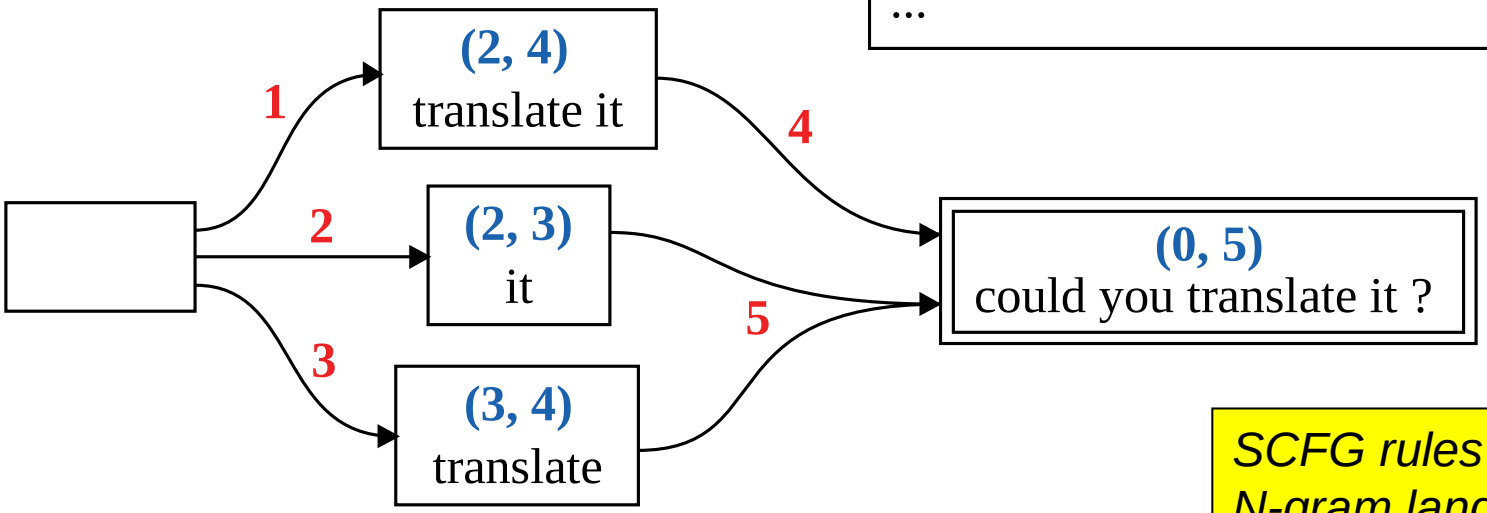
Hierarchical Phrase-Based Decoding

$_0$konnten$_1$sie$_2$es$_3$übersetzen$_4$?$_5$

could you translate it ?

SCFG Rules

1 X → es übersetzen / translate it
2 X → es / it
3 X → übersetzen / translate
4 X → konnten sie X ? / could you X ?
5 X → konnten sie X_1 X_2 ? / could you X_2 X_1 ?
...



SCFG rules
N-gram language model
Coverage constraints

ARK lti Carnegie Mellon

Our goal:

An MT framework that allows as many features as possible without committing to any particular decoding approach

Overview

- Initial step towards a “universal decoder” that can permit any feature of source and target words/trees/alignments
- Experimental platform for comparison of formalisms, feature sets, and training methods
- Building blocks:
  - Quasi-synchronous grammar (Smith & Eisner 2006)
  - Generic approximate inference methods for non-local features (Chiang 2007; Gimpel & Smith 2009)

Outline

- Introduction
- Model
- Quasi-Synchronous Grammar
- Training and Decoding
- Experiments
- Conclusions and Future Work

$$\langle \boldsymbol{t}^*, \tau_{\boldsymbol{t}}^*, \boldsymbol{a}^* \rangle = \underset{\langle \boldsymbol{t}, \tau_{\boldsymbol{t}}, \boldsymbol{a} \rangle}{\operatorname{argmax}} \; p(\boldsymbol{t}, \tau_{\boldsymbol{t}}, \boldsymbol{a} \mid \boldsymbol{s}, \tau_{\boldsymbol{s}})$$

target words

target tree

alignment of target tree nodes to source tree nodes

source words

source tree

Parameterization

$$\langle \boldsymbol{t}^*, \tau_{\boldsymbol{t}}^*, \boldsymbol{a}^* \rangle = \underset{\langle \boldsymbol{t}, \tau_{\boldsymbol{t}}, \boldsymbol{a} \rangle}{\operatorname{argmax}} \; p(\boldsymbol{t}, \tau_{\boldsymbol{t}}, \boldsymbol{a} \mid \boldsymbol{s}, \tau_{\boldsymbol{s}})$$

- We use a single globally-normalized log-linear model:

$$p(\boldsymbol{t}, \tau_{\boldsymbol{t}}, \boldsymbol{a} \mid \boldsymbol{s}, \tau_{\boldsymbol{s}}) = \frac{\exp\{\boldsymbol{\theta}^\top \boldsymbol{g}(\boldsymbol{s}, \tau_{\boldsymbol{s}}, \boldsymbol{a}, \boldsymbol{t}, \tau_{\boldsymbol{t}})\}}{\sum_{\boldsymbol{a}', \boldsymbol{t}', \tau_{\boldsymbol{t}}'} \exp\{\boldsymbol{\theta}^\top \boldsymbol{g}(\boldsymbol{s}, \tau_{\boldsymbol{s}}, \boldsymbol{a}', \boldsymbol{t}', \tau_{\boldsymbol{t}}')\}}$$

- Features can look at any part of any structure

Features

- Log-linear models allow “arbitrary” features, but in practice inference algorithms must be developed to support feature sets
- Many types of features appear in MT:
  - lexical word and phrase mappings
  - *N*-gram and syntactic language models
  - distortion/reordering
  - hierarchical phrase mappings
  - syntactic transfer rules
- We want to use all of these!

Outline

- Introduction
- Model
- Quasi-Synchronous Grammar
- Training and Decoding
- Experiments
- Conclusions and Future Work

Quasi-Synchronous Grammar

(Smith & Eisner 06)

- A **quasi-synchronous grammar** (QG) is a model of

$$p(\boldsymbol{t}, \tau_{\boldsymbol{t}}, \boldsymbol{a} \mid \boldsymbol{s}, \tau_{\boldsymbol{s}})$$

Quasi-Synchronous Grammar

(Smith & Eisner 06)

- A **quasi-synchronous grammar** (QG) is a model of

$$p(\boldsymbol{t}, \tau_{\boldsymbol{t}}, \boldsymbol{a} \mid \boldsymbol{s}, \tau_{\boldsymbol{s}})$$

- $\tau_{\boldsymbol{t}}$
  - To model target trees, any monolingual formalism can be used
  - We use a **quasi-synchronous dependency grammar** (QDG)

Quasi-Synchronous Grammar

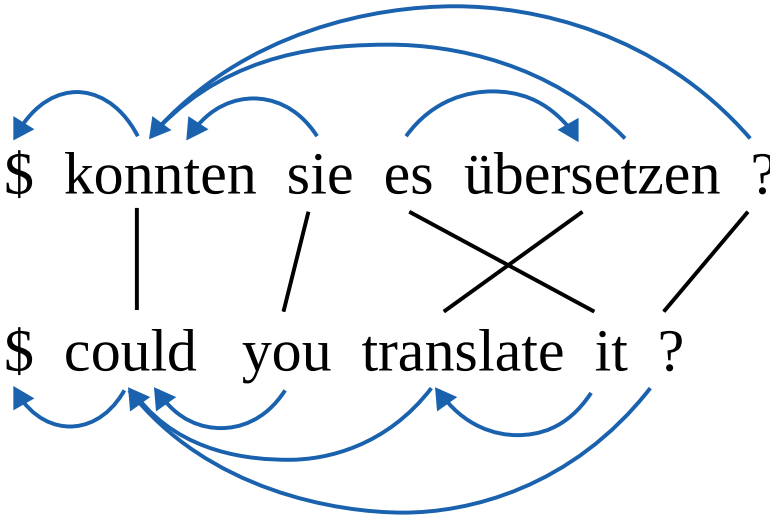
(Smith & Eisner 06)

- A **quasi-synchronous grammar** (QG) is a model of

$$p(\boldsymbol{t}, \tau_{\boldsymbol{t}}, \boldsymbol{a} \mid \boldsymbol{s}, \tau_{\boldsymbol{s}})$$

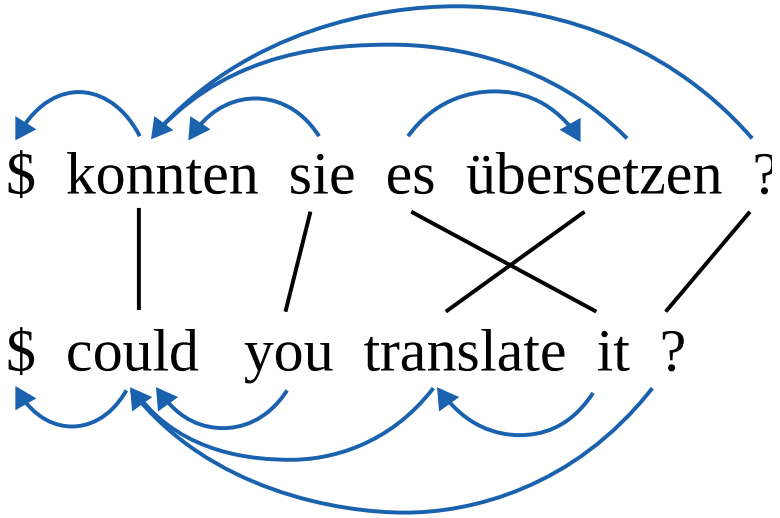
- $\tau_{\boldsymbol{t}}$
  - To model target trees, any monolingual formalism can be used
  - We use a **quasi-synchronous dependency grammar** (QDG)

- $\boldsymbol{a}$
  - Each node in the target tree is aligned to zero or more nodes in the source tree (for a QDG, nodes = words)
  - Constraints on the alignments → synchronous grammar
  - In QG, departures from synchrony are penalized *softly* using features

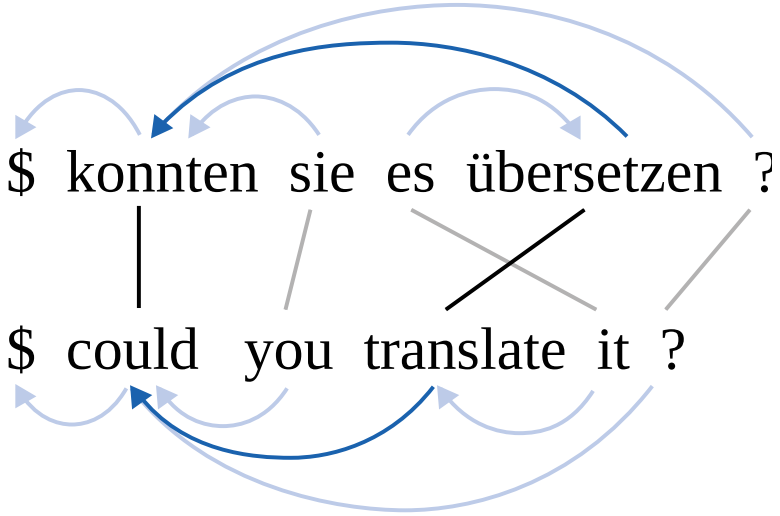

$ konnten sie es übersetzen ?
$ could you translate it ?


ARK
lti
Carnegie Mellon

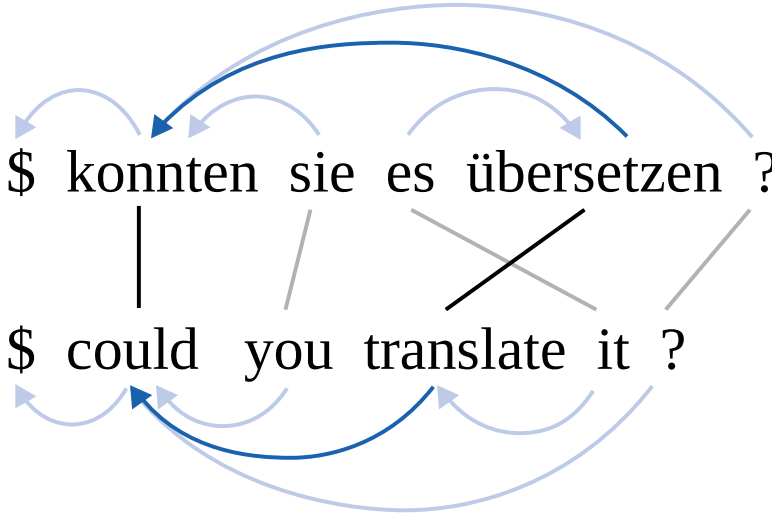
For every parent-child pair in the **target** sentence, what is the relationship of the **source** words they are linked to?



For every parent-child pair in the **target** sentence, what is the relationship of the **source** words they are linked to?



For every parent-child pair in the **target** sentence, what is the relationship of the **source** words they are linked to?

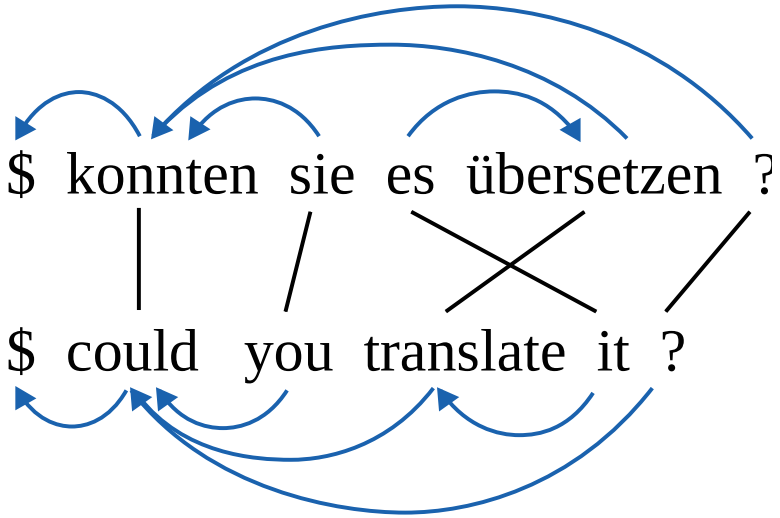


Parent-child

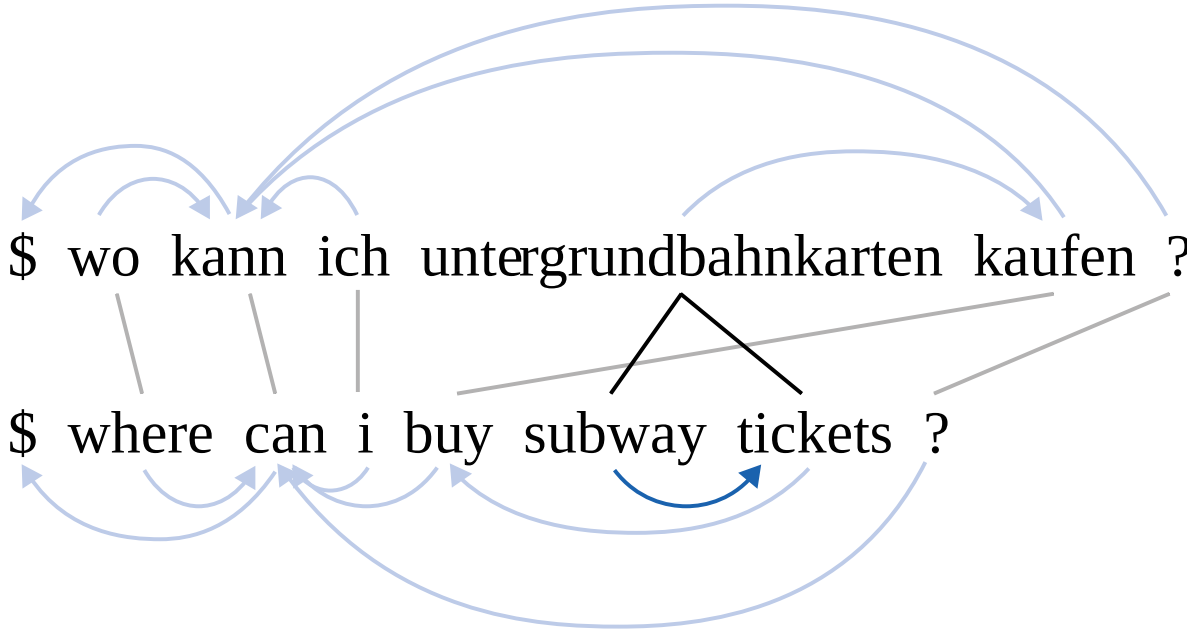
ARK lti Carnegie Mellon

For every parent-child pair in the **target** sentence, what is the relationship of the **source** words they are linked to?

All “parent-child” configurations → synchronous dependency grammar



Many other configurations are possible:



Same node

Many other configurations are possible:

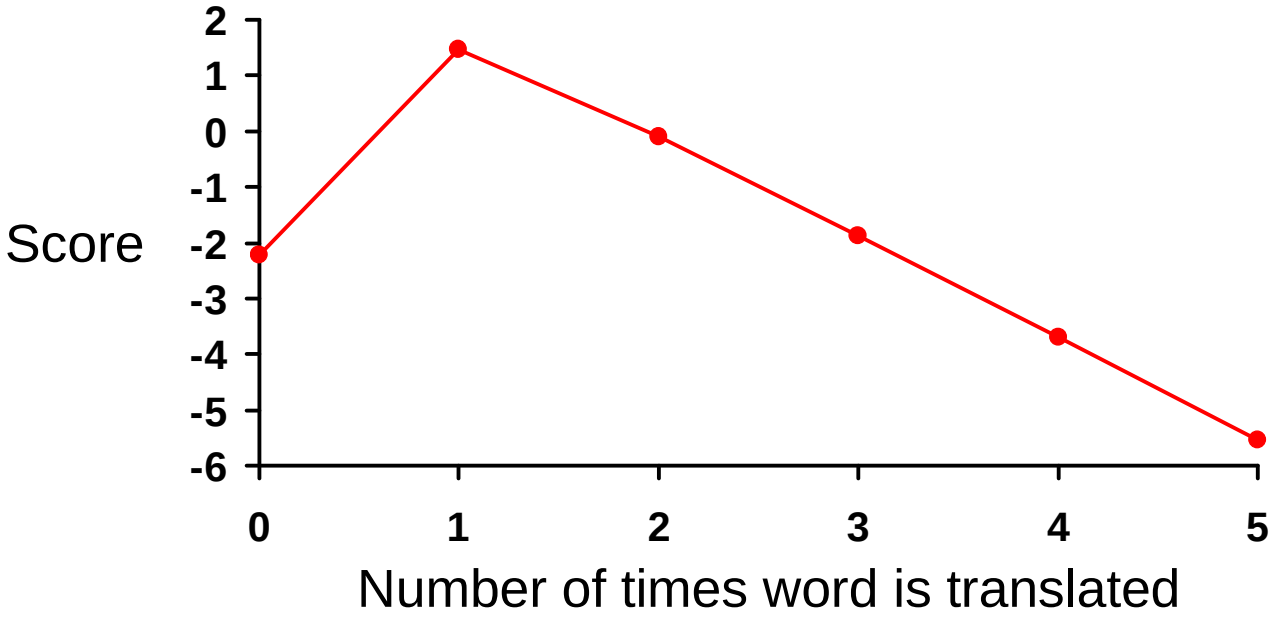
Parent-child
Child-parent
Same node
Sibling
Grandparent/child
Grandchild/parent
C-Command
Parent null
Child null
Both null
Other

Coverage Features

- There are no hard constraints to ensure that all source words get translated
- While QG has been used for several tasks, it has not previously been used for generation
- We add **coverage features** and learn their weights

Coverage Feature	Weight
Word never translated	-2.21
Word translated that was translated at least N times already:	
$N = 0$	1.48
$N = 1$	-3.04
$N = 2$	-0.22
$N = 3$	-0.05

Coverage Feature	Weight
Word never translated	-2.21
Word translated that was translated at least N times already:	
$N = 0$	1.48
$N = 1$	-3.04
$N = 2$	-0.22
$N = 3$	-0.05



Outline

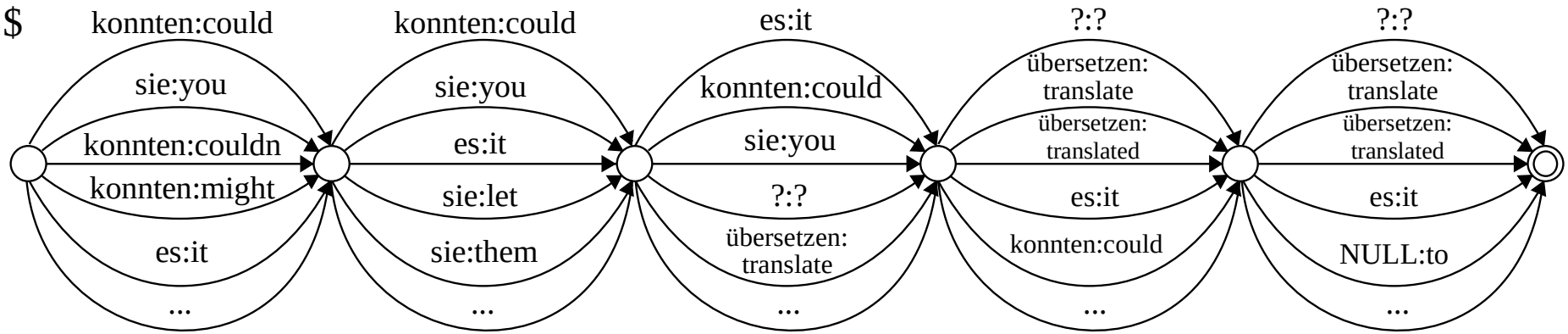
- Introduction
- Model
- Quasi-Synchronous Grammar
- Training and Decoding
- Experiments
- Conclusions and Future Work

Decoding

- A QDG induces a monolingual grammar for a source sentence whose language consists of all possible translations
- Decoding:
  - Build a weighted lattice encoding the language of this grammar
  - Perform lattice parsing with a dependency grammar
    - Extension of dependency parsing algs for strings (Eisner 97)
  - Integrate non-local features via cube pruning/decoding (Chiang 07, Gimpel & Smith 09)

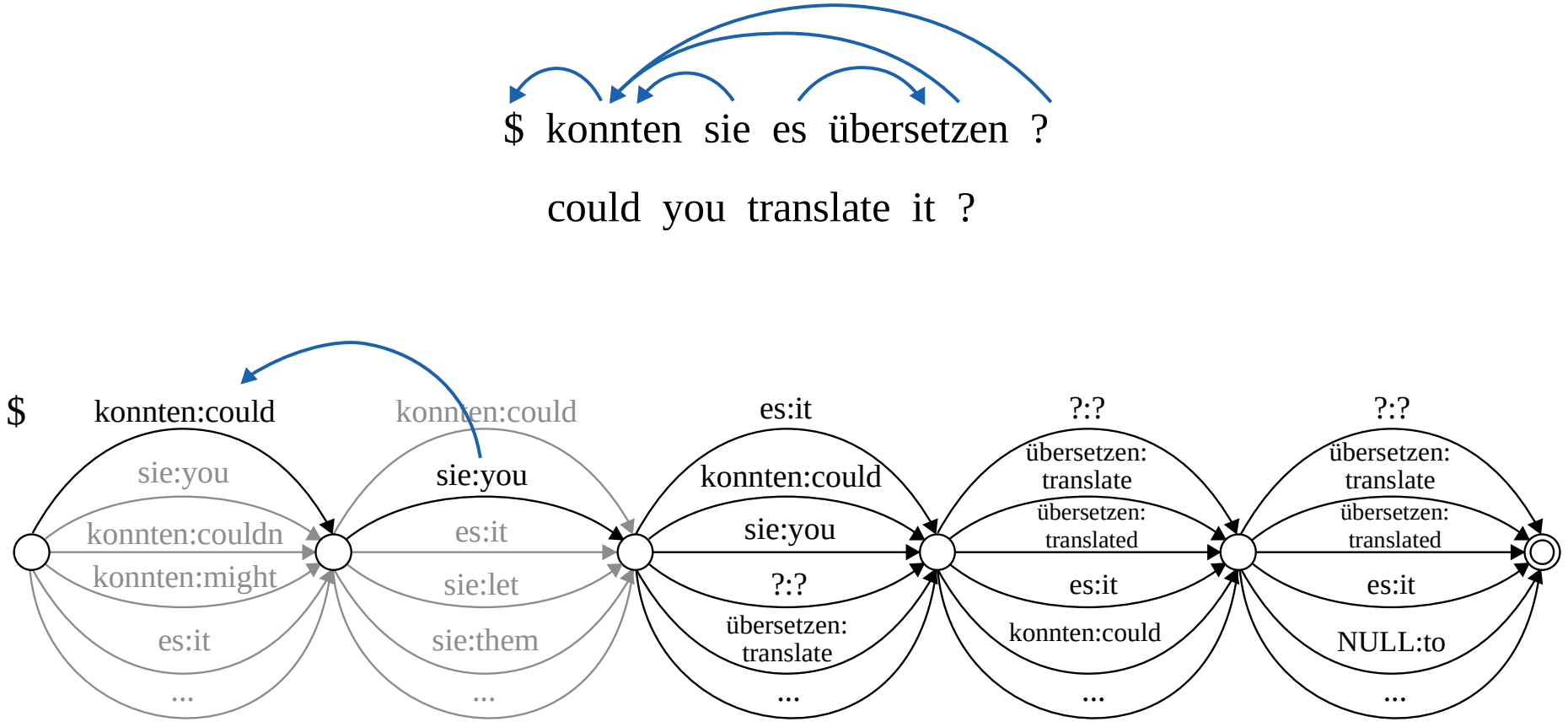
could you translate it ?

could you translate it ?

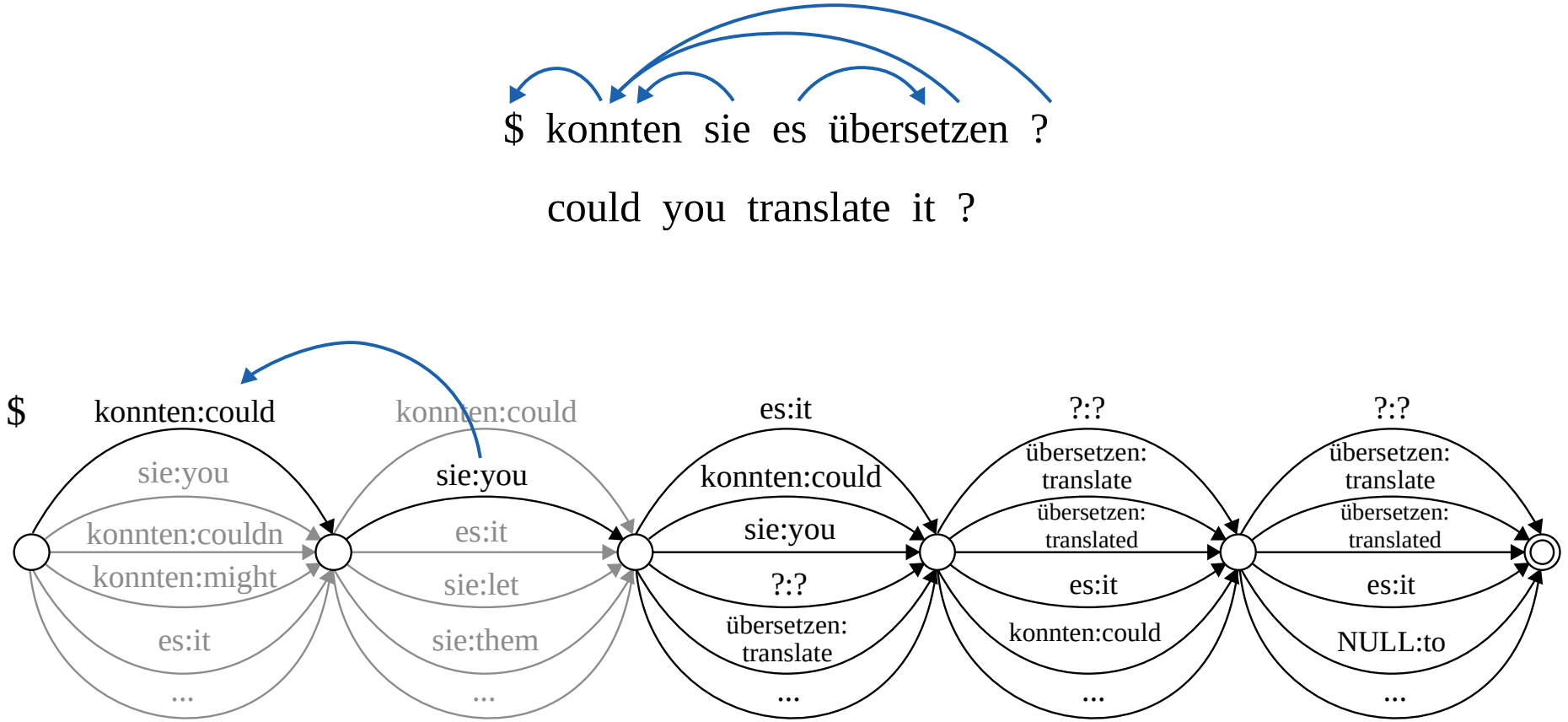


- Lattice arcs are weighted using lexical translation and distortion features
- Top 5 arcs shown in each bundle
- Hard limit on sentence length, multiple final states

ARK lti Carnegie Mellon

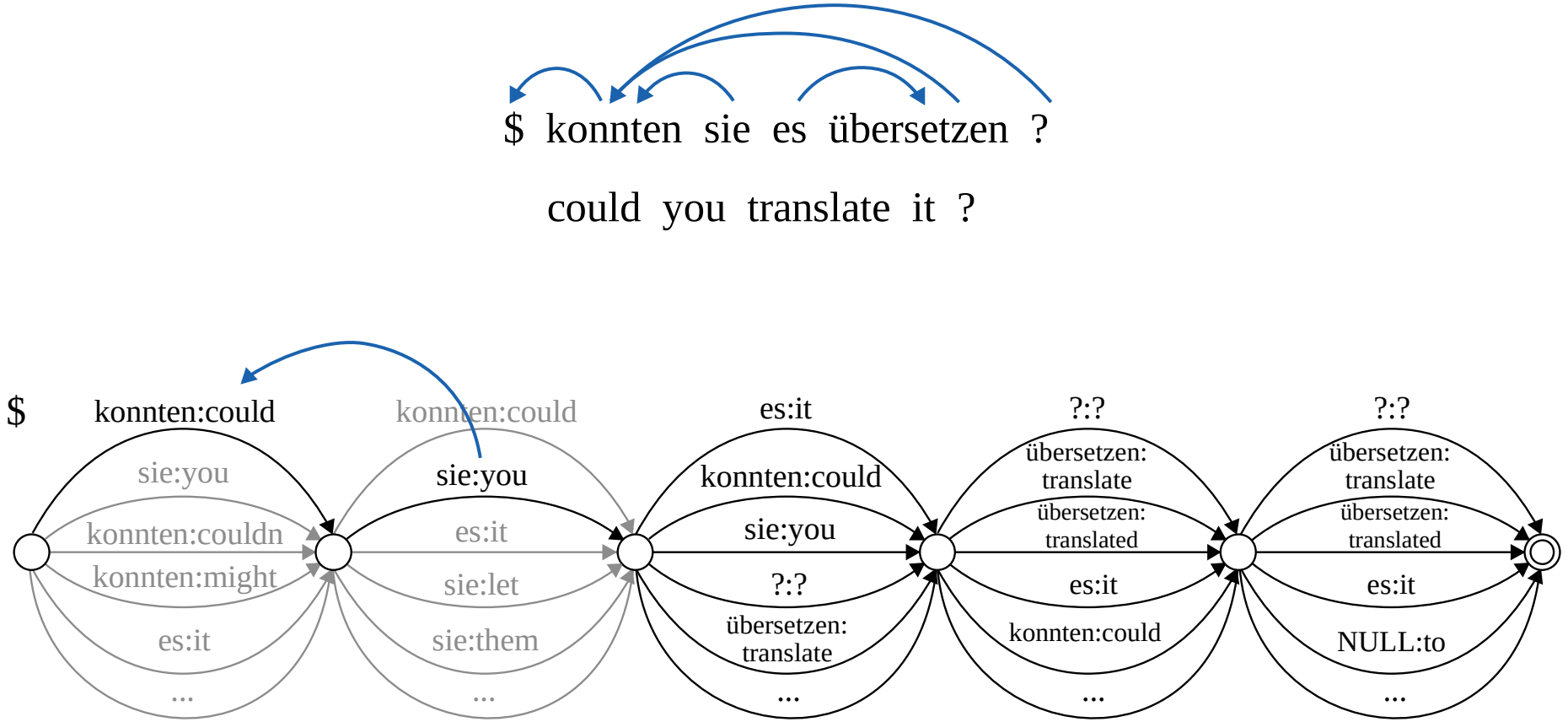


ARK lti Carnegie Mellon



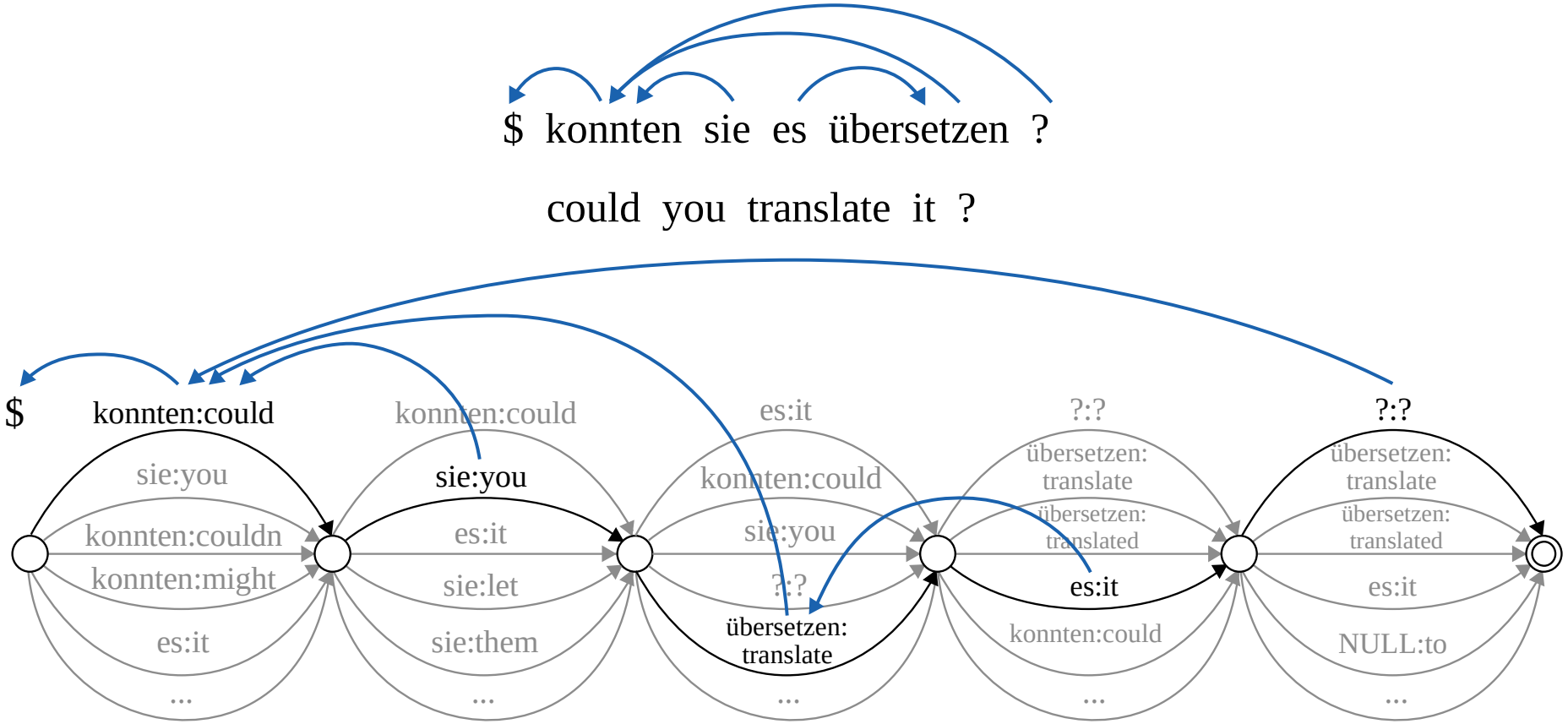
Bigram feature: “could you”

ARK lti Carnegie Mellon



Bigram feature: “could you”

Phrase features: “konnten sie” → “could you”


$ konnten sie es übersetzen ?
could you translate it ?
$
konnten:could
sie:you
konnten:couldn
konnten:might
es:it
...
konnten:could
sie:you
es:it
sie:let
sie:them
...
es:it
konnten:could
sie:you
?:?
übersetzen:
translate
...
?:?
übersetzen:
translate
übersetzen:
translated
es:it
konnten:could
...
?:?
übersetzen:
translate
übersetzen:
translated
es:it
NULL:to
...


ARK
lti
Carnegie Mellon

Training

- Recall that we use a single globally-normalized log-linear model:

$$p(\boldsymbol{t}, \tau_{\boldsymbol{t}}, \boldsymbol{a} \mid \boldsymbol{s}, \tau_{\boldsymbol{s}}) = \frac{\exp\{\boldsymbol{\theta}^\top \boldsymbol{g}(\boldsymbol{s}, \tau_{\boldsymbol{s}}, \boldsymbol{a}, \boldsymbol{t}, \tau_{\boldsymbol{t}})\}}{\sum_{\boldsymbol{a}', \boldsymbol{t}', \tau'_{\boldsymbol{t}}} \exp\{\boldsymbol{\theta}^\top \boldsymbol{g}(\boldsymbol{s}, \tau_{\boldsymbol{s}}, \boldsymbol{a}', \boldsymbol{t}', \tau'_{\boldsymbol{t}})\}}$$

  - If all structures are given, this becomes a convex, supervised learning problem
  - If a structure is not given, it can be marginalized out during training (or simply ignored during both training and testing)
  - Here, we assume alignments are not given and marginalize them out during training

Training

- Standard approach is to optimize conditional likelihood

$$\mathrm{LL}(\boldsymbol{\theta}) = \sum_{i=1}^{N} \log p(\boldsymbol{t}^{(i)}, \tau_{\boldsymbol{t}}^{(i)} \mid \boldsymbol{s}^{(i)}, \tau_{\boldsymbol{s}}^{(i)})$$

$$= \sum_{i=1}^{N} \log \frac{\sum_{\boldsymbol{a}} \exp\{\boldsymbol{\theta}^\top \boldsymbol{g}(\boldsymbol{s}^{(i)}, \tau_{\boldsymbol{s}}^{(i)}, \boldsymbol{a}, \boldsymbol{t}^{(i)}, \tau_{\boldsymbol{t}}^{(i)})\}}{\sum_{\boldsymbol{t}, \tau_{\boldsymbol{t}}, \boldsymbol{a}} \exp\{\boldsymbol{\theta}^\top \boldsymbol{g}(\boldsymbol{s}^{(i)}, \tau_{\boldsymbol{s}}^{(i)}, \boldsymbol{a}, \boldsymbol{t}, \tau_{\boldsymbol{t}})\}}$$

problem: must sum over words + trees + alignments!

Pseudo-likelihood

- Solution: optimize *pseudo-likelihood* (Besag, 1975) by making the following approximation:

$$p(\boldsymbol{t}, \tau_{\boldsymbol{t}} \mid \boldsymbol{s}, \tau_{\boldsymbol{s}}) \approx p(\boldsymbol{t} \mid \tau_{\boldsymbol{t}}, \boldsymbol{s}, \tau_{\boldsymbol{s}}) \times p(\tau_{\boldsymbol{t}} \mid \boldsymbol{t}, \boldsymbol{s}, \tau_{\boldsymbol{s}})$$

- The objective function becomes:

$$\mathrm{PL}(\boldsymbol{\theta}) = \sum_{i=1}^{N} \log \left(\sum_{\boldsymbol{a}} p(\boldsymbol{t}^{(i)}, \boldsymbol{a} \mid \tau_{\boldsymbol{t}}^{(i)}, \boldsymbol{s}^{(i)}, \tau_{\boldsymbol{s}}^{(i)}) \right)$$

sum over words + alignments

$$+ \sum_{i=1}^{N} \log \left(\sum_{\boldsymbol{a}} p(\tau_{\boldsymbol{t}}^{(i)}, \boldsymbol{a} \mid \boldsymbol{t}^{(i)}, \boldsymbol{s}^{(i)}, \tau_{\boldsymbol{s}}^{(i)}) \right)$$

sum over trees + alignments

  - Integrate non-local features via “cube summing” [Gimpel & Smith 09]

Outline

- Introduction
- Model
- Quasi-Synchronous Grammar
- Training and Decoding
- Experiments
- Conclusions and Future Work

Experiments

- One of our goals was an experimental platform to address questions like the following:
  - How do phrase features interact with syntactic features?
  - How do synchronous (isomorphism) constraints affect translation quality?
  - How do string-to-tree, tree-to-string, and tree-to-tree approaches compare in terms of runtime and translation quality?
  - Does a small number of feature templates work better than a large number of binary features?
  - How do MERT/MIRA compare with optimization of conditional likelihood?

Experiments

- One of our goals was an experimental platform to address questions like the following:
  - How do phrase features interact with syntactic features?
  - How do synchronous (isomorphism) constraints affect translation quality?
  - How do string-to-tree, tree-to-string, and tree-to-tree approaches compare in terms of runtime and translation quality?
  - Does a small number of feature templates work better than a large number of binary features?
  - How do MERT/MIRA compare with optimization of conditional likelihood?

Experimental Setup

- Data
  - German-English Basic Travel Expression Corpus (BTEC)
  - Only sentences of length ≤ 15
  - 80k sentences for training, 1k for tuning, 500 for testing
- Features
  - Parsed source and target text using Stanford parser
  - Phrase extraction using Moses (max phrase length = 3)
  - Trigram language model

Experiments

- This is not a state-of-the-art MT system
  - Moses obtains 68.4 BLEU and 85.2 METEOR on this dataset
  - Our best scores are 52 BLEU and 75 METEOR

Features

Lexical Translation
$p(s \mid t)$
$p(t \mid s)$

Language Model
BigramProbability
TrigramProbability

Reordering
AbsoluteDistortion

Coverage
WordLeftUntranslated
UsedWordAlreadyUsedNTimes
(N in {0,1,2,3})

Phrase Translation
$p(\boldsymbol{s} \mid \boldsymbol{t})$
$p(\boldsymbol{t} \mid \boldsymbol{s})$
$lex(\boldsymbol{s} \mid \boldsymbol{t})$
$lex(\boldsymbol{t} \mid \boldsymbol{s})$

Target Dependency	
words & word classes	$p(root)$
	$p(child \mid parent, left)$
	$p(child \mid parent, right)$
	(+ 4 valence distributions)

QG Configuration
(14 binary features, one for each configuration)

Features

Lexical Translation

$p(s \mid t)$
$p(t \mid s)$

Language Model

BigramProbability
TrigramProbability

Reordering

AbsoluteDistortion

Coverage

WordLeftUntranslated
UsedWordAlreadyUsedNTimes
(N in {0,1,2,3})

Phrase Translation

$p(\boldsymbol{s} \mid \boldsymbol{t})$
$p(\boldsymbol{t} \mid \boldsymbol{s})$
$lex(\boldsymbol{s} \mid \boldsymbol{t})$
$lex(\boldsymbol{t} \mid \boldsymbol{s})$

Target Dependency

words & word classes

$p(root)$
$p(child \mid parent, left)$
$p(child \mid parent, right)$
(+ 4 valence distributions)

QG Configuration

(14 binary features, one for each configuration)

ARK lti Carnegie Mellon

Feature Set Comparison: BLEU Scores

	No Syntax Features	Target Syntax Features Only	Source & Target Syntax Features
No Phrase Features	37.3	44.6	44.2
Phrase Features	46.8	49.7	51.4

QDG Configuration Comparison

	BLEU	METEOR
synchronous	40.1	69.5
+ nulls, root-any	41.1	69.3
+ child-parent, same-node	43.4	68.2
+ sibling	48.8	72.2
+ grandparent/child	50.2	73.7
+ c-command	51.6	74.4
+ other	51.4	74.7

Conclusions and Ongoing Work

- We have described an MT system based on quasi-synchronous grammar that can use features from many types of MT systems
- We reported on preliminary experiments comparing feature sets and synchronous dependency constraints
- Ongoing work in improving decoder efficiency, adding features, and conducting additional experiments

Thanks!