

# Statistical Machine Translation

## Lecture 2

### Theory and Praxis of Decoding

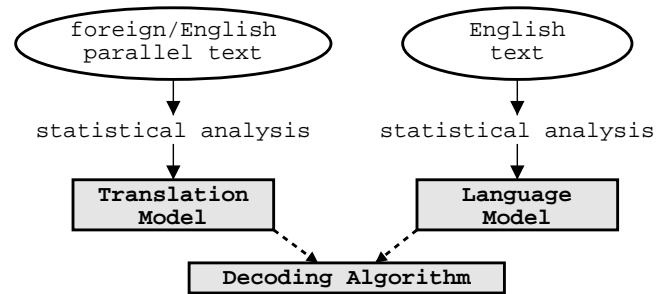
Philipp Koehn  
pkoehn@inf.ed.ac.uk

School of Informatics  
University of Edinburgh



## Statistical Machine Translation

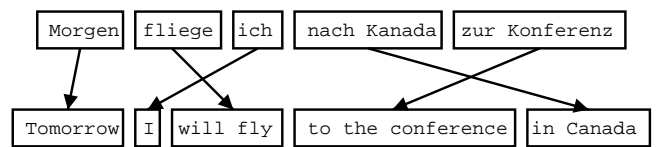
- Components: Translation model, language model, decoder



## Phrase-Based Systems

- A number of research groups developed phrase-based systems ( RWTH Aachen, Univ. of Southern California/ISI, CMU, IBM, Johns Hopkins Univ., Cambridge Univ., Univ. of Catalunya, ITC-irst, Univ. Edinburgh, Univ. of Maryland...)
- Systems differ in
  - training methods
  - model for phrase translation table
  - reordering models
  - additional feature functions
- Currently best method for SMT (MT?)
  - top systems in DARPA/NIST evaluation are phrase-based
  - best commercial system for Arabic-English is phrase-based

## Phrase-Based Translation



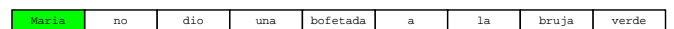
- Foreign input is segmented in phrases
  - any sequence of words, not necessarily linguistically motivated
- Each phrase is translated into English
- Phrases are reordered

## Phrase Translation Table

- Phrase Translations for “den Vorschlag”:

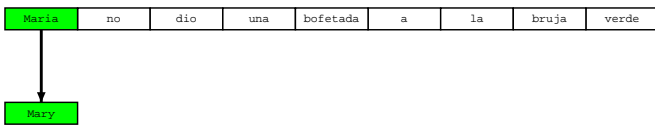
English	$\phi(e f)$	English	$\phi(e f)$
the proposal	0.6227	the suggestions	0.0114
's proposal	0.1068	the proposed	0.0114
a proposal	0.0341	the motion	0.0091
the idea	0.0250	the idea of	0.0091
this proposal	0.0227	the proposal ,	0.0068
proposal	0.0205	its proposal	0.0068
of the proposal	0.0159	it	0.0068
the proposals	0.0159	...	...

## Decoding Process



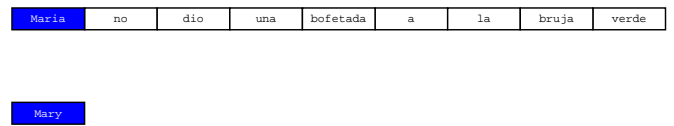
- Build translation left to right
  - select foreign words to be translated

## Decoding Process



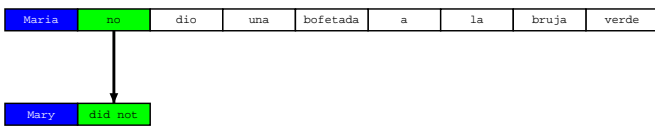
- Build translation left to right
  - select foreign words to be translated
  - find English phrase translation
  - add English phrase to end of partial translation

## Decoding Process



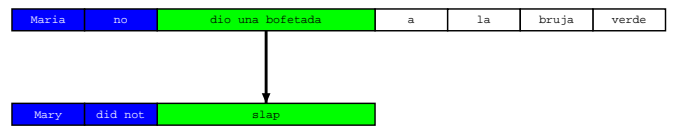
- Build translation left to right
  - select foreign words to be translated
  - find English phrase translation
  - add English phrase to end of partial translation
  - mark foreign words as translated

## Decoding Process



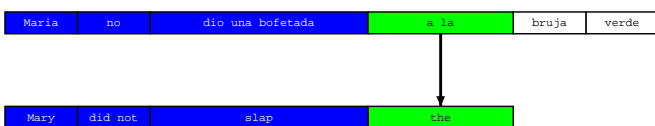
- One to many translation

## Decoding Process



- Many to one translation

## Decoding Process



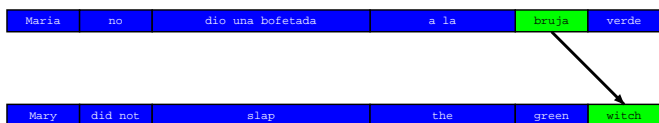
- Many to one translation

## Decoding Process



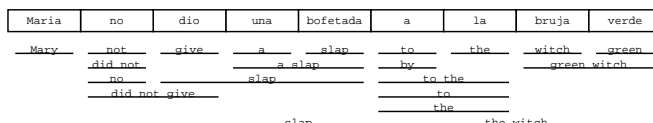
- Reordering

## Decoding Process



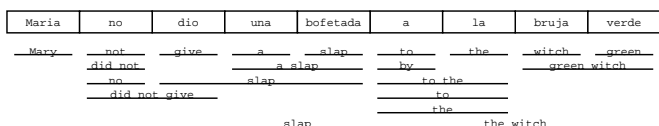
- Translation finished

## Translation Options



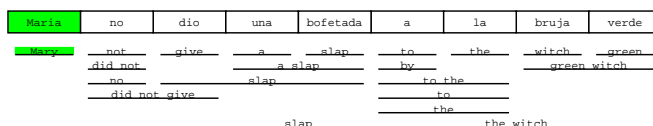
- Look up possible phrase translations
  - many different ways to segment words into phrases
  - many different ways to translate each phrase

## Hypothesis Expansion



- Start with empty hypothesis
  - e: no English words
  - f: no foreign words covered
  - p: probability 1

## Hypothesis Expansion

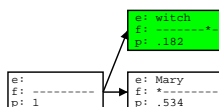
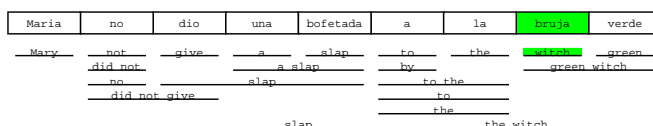


- Pick translation option
- Create hypothesis
  - e: add English phrase Mary
  - f: first foreign word covered
  - p: probability 0.534

## A Quick Word on Probabilities

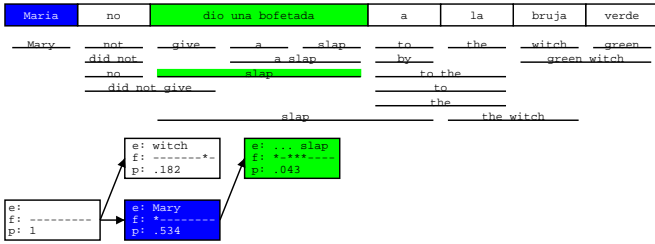
- Not going into detail here, but...
- Translation Model
  - phrase translation probability  $p(\text{Mary}|\text{Maria})$
  - reordering costs
  - phrase/word count costs
  - ...
- Language Model
  - uses trigrams:
  - $p(\text{Mary did not}) = p(\text{Mary} | \langle s \rangle) * p(\text{did} | \text{Mary}, \langle s \rangle) * p(\text{not} | \text{Mary did})$

## Hypothesis Expansion



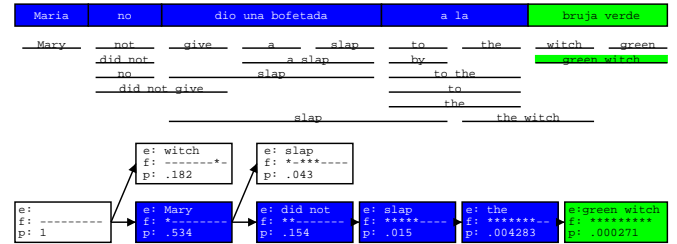
- Add another hypothesis

## Hypothesis Expansion



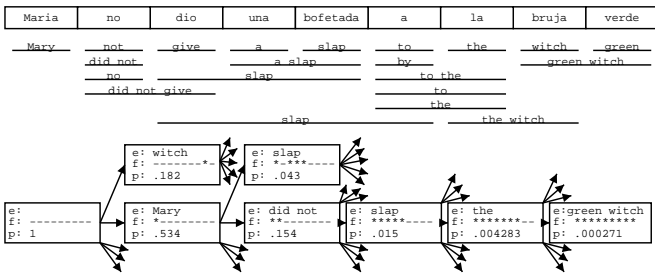
- Further hypothesis expansion

## Hypothesis Expansion



- ... until all foreign words covered
  - find best hypothesis that covers all foreign words
  - backtrack to read off translation

## Hypothesis Expansion



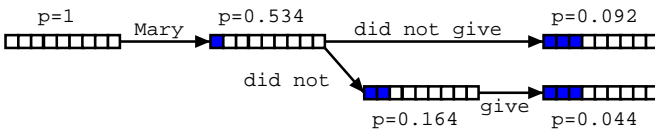
- Adding more hypothesis

⇒ Explosion of search space

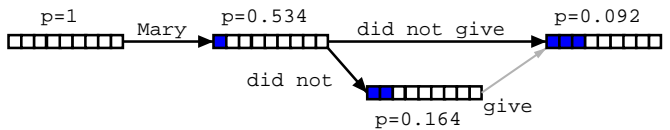
## Explosion of Search Space

- Number of hypotheses is exponential with respect to sentence length
- ⇒ Decoding is NP-complete [Knight, 1999]
- ⇒ Need to reduce search space
- risk free: hypothesis recombination
  - risky: histogram/threshold pruning

## Hypothesis Recombination

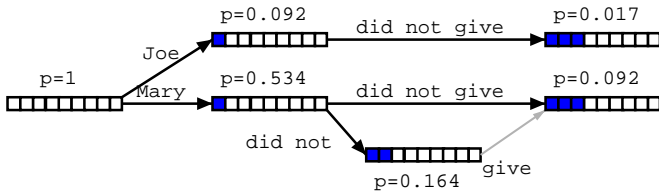


- Different paths to the same partial translation



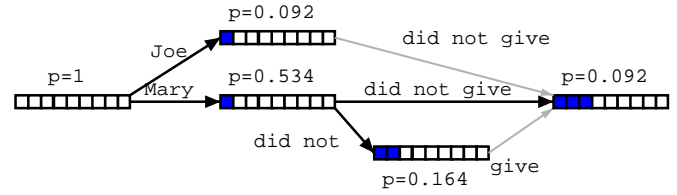
- Different paths to the same partial translation
- ⇒ Combine paths
- drop weaker hypothesis
  - keep pointer from worse path

## Hypothesis Recombination



- Recombined hypotheses do not have to match completely
- No matter what is added, weaker path can be dropped, if:
  - last two English words match (matters for language model)
  - foreign word coverage vectors match (effects future path)

## Hypothesis Recombination



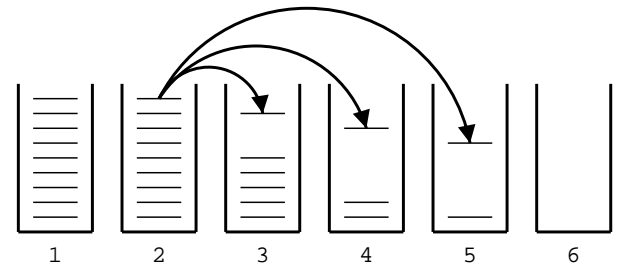
- Recombined hypotheses do not have to match completely
- No matter what is added, weaker path can be dropped, if:
  - last two English words match (matters for language model)
  - foreign word coverage vectors match (effects future path)

⇒ Combine paths

## Pruning

- Hypothesis recombination is not sufficient
- ⇒ Heuristically discard weak hypotheses
- Organize Hypothesis in stacks, e.g. by
    - same foreign words covered
    - same number of foreign words covered (Pharaoh does this)
    - same number of English words produced
  - Compare hypotheses in stacks, discard bad ones
    - histogram pruning: keep top  $n$  hypotheses in each stack (e.g.,  $n=100$ )
    - threshold pruning: keep hypotheses that are at most  $\alpha$  times the cost of best hypothesis in stack (e.g.,  $\alpha = 0.001$ )

## Hypothesis Stacks



- Organization of hypothesis into stacks
  - here: based on number of foreign words translated
  - during translation all hypotheses from one stack are expanded
  - expanded Hypotheses are placed into stacks

## Comparing Hypotheses

- Comparing hypotheses with same number of foreign words covered

María no dio una bofetada a la bruja verde

```
e: Mary did not
f: **-----
p: 0.154
```

better partial translation

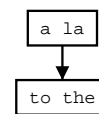
```
e: the
f: -----**--
p: 0.354
```

covers easier part --> lower cost

- Hypothesis that covers easy part of sentence is preferred

⇒ Need to consider future cost of uncovered parts

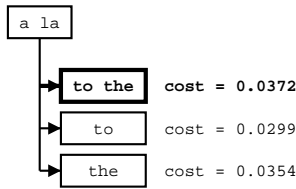
## Future Cost Estimation



- Estimate cost to translate remaining part of input
- Step 1: estimate future cost for each translation option
  - look up translation model cost
  - estimate language model cost (no prior context)
  - ignore reordering model cost

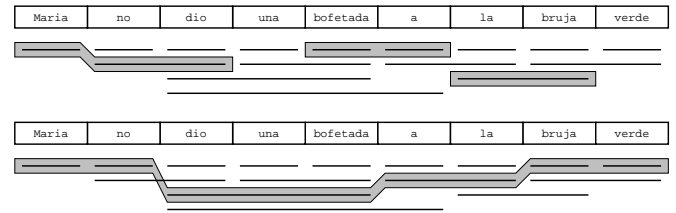
$$\rightarrow LM * TM = p(\text{to}) * p(\text{the}|\text{to}) * p(\text{to the}|a la)$$

## Future Cost Estimation: Step 2



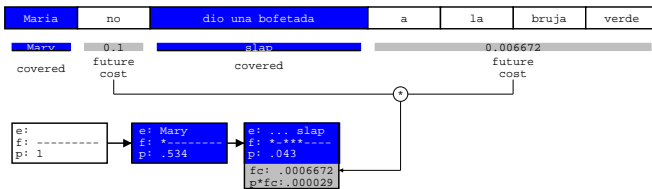
- Step 2: find cheapest cost among translation options

## Future Cost Estimation: Step 3



- Step 3: find cheapest future cost path for each span
  - can be done efficiently by dynamic programming
  - future cost for every span can be precomputed

## Future Cost Estimation: Application



- Use future cost estimates when pruning hypotheses
- For each uncovered contiguous span:
  - look up future costs for each maximal contiguous uncovered span
  - factor them to actually accumulated cost for translation option for pruning

## Pharaoh

- A beam search decoder for phrase-based models
  - works with various phrase-based models
  - beam search algorithm
  - time complexity roughly linear with input length
  - good quality takes about 1 second per sentence
- Very good performance in DARPA/NIST Evaluation
- Freely available for researchers
  - <http://www.isi.edu/licensed-sw/pharaoh/>

## Running the decoder

- An example run of the decoder:

```

% echo 'das ist ein kleines haus' | pharaoh -f pharaoh.ini > out
Pharaoh v1.2.9, written by Philipp Koehn
a beam search decoder for phrase-based statistical machine
translation models
(c) 2002-2003 University of Southern California
(c) 2004 Massachusetts Institute of Technology
(c) 2005 University of Edinburgh, Scotland
loading language model from europarl.srilm
loading phrase translation table from phrase-table, stored 21, pruned
0, kept 21
loaded data structures in 2 seconds
reading input sentences
translating 1 sentences.translated 1 sentences in 0 seconds

% cat out
this is a small house
  
```

## Phrase Translation Table

- Core model component is the phrase translation table:

```

der ||| the ||| 0.3
das ||| the ||| 0.4
das ||| it ||| 0.1
das ||| this ||| 0.1
die ||| the ||| 0.3
ist ||| is ||| 1.0
ist ||| 's ||| 1.0
das ist ||| it is ||| 0.2
das ist ||| this is ||| 0.8
es ist ||| it is ||| 0.8
es ist ||| this is ||| 0.2
ein ||| a ||| 1.0
ein ||| an ||| 1.0
klein ||| small ||| 0.8
klein ||| little ||| 0.8
kleines ||| small ||| 0.2
kleines ||| little ||| 0.2
haus ||| house ||| 1.0
alt ||| old ||| 0.8
altes ||| old ||| 0.2
gibt ||| gives ||| 1.0
es gibt ||| there is ||| 1.0
  
```

## Trace

- Running the decoder with switch “-t”

```
% echo 'das ist ein kleines haus' | pharaoh -f pharaoh.ini -t
[...]
```

this is	0.014086 0 1	a	0.188447 2 2	small	0.000706353 3 3
house	1.46468e-07 4 4				

- Trace for each applied phrase translation:

- output phrase (there is)
- cost incurred by this phrase (0.014086)
- coverage of foreign words (0-1)

## Hypothesis Accounting

- The switch “-v” allows for detailed run time information:

```
% echo 'das ist ein kleins haus' | pharaoh -f pharaoh.ini -v 2
[...]
```

HYP: 114 added, 284 discarded below threshold, 0 pruned, 58 merged.  
BEST: this is a small house -28.9234

- Statistics over how many hypothesis were generated

- 114 hypotheses were added to hypothesis stacks
- 284 hypotheses were discarded because they were too bad
- 0 hypotheses were pruned, because a stack got too big
- 58 hypotheses were merged due to recombination

- Probability of the best translation:  $\exp(-28.9234)$

## Future Cost Estimation

- Pre-computation of the future cost estimates:

```
future costs from 0 to 0 is -5.78855
future costs from 0 to 1 is -10.207
future costs from 0 to 2 is -15.7221
future costs from 0 to 3 is -25.4433
future costs from 0 to 4 is -34.7094
future costs from 1 to 1 is -4.92223
future costs from 1 to 2 is -10.4373
future costs from 1 to 3 is -20.1585
future costs from 1 to 4 is -29.4246
future costs from 2 to 2 is -5.5151
future costs from 2 to 3 is -15.2363
future costs from 2 to 4 is -24.5023
future costs from 3 to 3 is -9.72116
future costs from 3 to 4 is -18.9872
future costs from 4 to 4 is -9.26607
```

## Reordering Example

- Sometimes phrases have to be reordered:

```
% echo 'ein kleines haus ist das' | pharaoh -f pharaoh.ini -t -d 0.5
[...]
```

this	0.000632805 4 4	is	0.13853 3 3	a	0.0255035 0 0
small	0.000706353 1 1	house	1.46468e-07 2 2		

- First output phrase (this) is translation of the 4th word

## Translation Options

- Even more run time information is revealed with “-v 3”:

```
[das:2]
the<1>, pC=-0.916291, c=-5.78855
it<2>, pC=-2.30259, c=-8.0761
this<3>, pC=-2.30259, c=-8.00205

[ist:4]
is<4>, pC=0, c=-4.92223
's<5>, pC=0, c=-6.11591

[ein:7]
a<8>, pC=0, c=-5.5151
an<9>, pC=0, c=-6.41298

[kleines:9]
small<10>, pC=-1.60944, c=-9.72116
little<11>, pC=-1.60944, c=-10.0953

[haus:10]
house<12>, pC=0, c=-9.26607

[das ist:5]
it is<6>, pC=-1.60944, c=-10.207
this is<7>, pC=-0.223144, c=-10.2906
```

- Translation model cost (pC) and future cost estimates (c)

## Hypothesis Expansion

- Start of beam search: First hypothesis (das → the)

```
creating hypothesis 1 from 0 ( ... </s> <s> )
base score 0
covering 0-0: das
translated as: the => translation cost -0.916291
distance 0 => distortion cost 0
language model cost for 'the' -2.03434
word penalty -0
score -2.95064 + futureCost -29.4246 = -32.3752
new best estimate for this stack
merged hypothesis on stack 1, now size 1
```

## Hypothesis Expansion

- Another hypothesis (das ist → this is)

```
creating hypothesis 12 from 0 ( ... </s> <s> )
base score 0
covering 0-1: das ist
translated as: this is => translation cost -0.223144
distance 0 => distortion cost 0
language model cost for 'this' -3.06276
language model cost for 'is' -0.976669
word penalty -0
score -4.26258 + futureCost -24.5023 = -28.7649
new best estimate for this stack
merged hypothesis on stack 2, now size 2
```

## Hypothesis Expansion

- Bad hypothesis that falls out of the beam

```
creating hypothesis 52 from 6 ( ... <s> a )
base score -6.65992
covering 0-0: das
translated as: this => translation cost -2.30259
distance -3 => distortion cost -3
language model cost for 'this' -8.69176
word penalty -0
score -20.6543 + futureCost -23.9095 = -44.5637
estimate below threshold, discarding
```

## Translation Table Pruning

- Limiting translation table size speeds up search
- Histogram pruning: keeping only top  $n$  entries
- Threshold pruning: keep only entries that score  $\alpha$  times worse than best

## Hypothesis Expansion

- Hypothesis recombination

```
creating hypothesis 27 from 3 ( ... <s> this )
base score -5.36535
covering 1-1: ist
translated as: is => translation cost 0
distance 0 => distortion cost 0
language model cost for 'is' -0.976669
word penalty -0
score -6.34202 + futureCost -24.5023 = -30.8443
worse than existing path to 12, discarding
```

## Generating Best Translation

- Generating best translation
  - find best final hypothesis (442)
  - trace back path to initial hypothesis

```
best hypothesis 442
[ 442 => 343 ]
[ 343 => 106 ]
[ 106 => 12 ]
[ 12 => 0 ]
```

## Beam Size

- Trade-off between speed and quality via beam size

```
% echo 'das ist ein kleines haus' | pharaoh -f pharaoh.ini -s 10 -v 2
[...]
collected 12 translation options
HYP: 78 added, 122 discarded below threshold, 33 pruned, 20 merged.
BEST: this is a small house -28.9234
```

Beam size	Threshold	Hyp. added	Hyp. discarded	Hyp. pruned	Hyp. merged
1000	unlimited	634	0	0	1306
100	unlimited	557	32	199	572
100	0.00001	144	284	0	58
10	0.00001	78	122	33	20
1	0.00001	9	19	4	0



## Limits on Reordering

- Reordering may be limited
  - Monotone Translation: No reordering at all
  - Only phrase movements of at most  $n$  words
- Reordering limits speed up search
- Current reordering models are weak, so limits improve translation quality

## Sample N-Best List

- N-best list from Pharaoh:

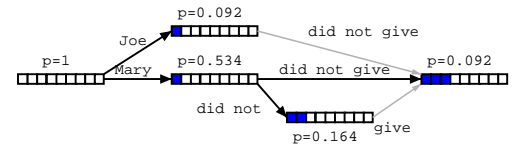
```

Translation ||| Reordering LM TM WordPenalty ||| Score
this is a small house ||| 0 -27.0908 -1.83258 -5 ||| -28.9234
this is a little house ||| 0 -28.1791 -1.83258 -5 ||| -30.0117
it is a small house ||| 0 -27.108 -3.21888 -5 ||| -30.3268
it is a little house ||| 0 -28.1963 -3.21888 -5 ||| -31.4152
this is an small house ||| 0 -31.7294 -1.83258 -5 ||| -33.562
it is an small house ||| 0 -32.3094 -3.21888 -5 ||| -35.5283
this is an little house ||| 0 -33.7639 -1.83258 -5 ||| -35.5965
this is a house small ||| -3 -31.4851 -1.83258 -5 ||| -36.3176
this is a house little ||| -3 -31.5689 -1.83258 -5 ||| -36.4015
it is an little house ||| 0 -34.3439 -3.21888 -5 ||| -37.5628
it is a house small ||| -3 -31.5022 -3.21888 -5 ||| -37.7211
this is an house small ||| -3 -32.8999 -1.83258 -5 ||| -37.7325
it is a house little ||| -3 -31.586 -3.21888 -5 ||| -37.8049
this is an house little ||| -3 -32.9837 -1.83258 -5 ||| -37.8163
the house is a little ||| -7 -28.5107 -2.52573 -5 ||| -38.0364
the is a small house ||| 0 -35.6899 -2.52573 -5 ||| -38.2156
is it a little house ||| -4 -30.3603 -3.91202 -5 ||| -38.2723
the house is a small ||| -7 -28.7683 -2.52573 -5 ||| -38.294
it 's a small house ||| 0 -34.8557 -3.91202 -5 ||| -38.7677
this house is a little ||| -7 -28.0443 -3.91202 -5 ||| -38.9563
it 's a little house ||| 0 -35.1446 -3.91202 -5 ||| -39.0566
this house is a small ||| -7 -28.3018 -3.91202 -5 ||| -39.2139
    
```

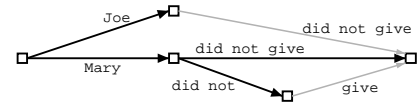
## Thank You!

- Questions?

## Word Lattice Generation



- Search graph can be easily converted into a word lattice
  - can be further mined for n-best lists
  - enables reranking approaches
  - enables discriminative training



## XML Interface

Er erzielte <NUMBER english='17.55'>17,55</NUMBER> Punkte .

- Add additional translation options
  - number translation
  - noun phrase translation [Koehn, 2003]
  - name translation
- Additional options
  - provide multiple translations
  - provide probability distribution along with translations
  - allow bypassing of provided translations