Phrase-based Statistical Machine Translation

Ulrich Germann, University of Edinburgh

September 16, 2016

(Lecture on decoding algorithm, B Pouliquen, Gian course, Varanasi)
Original slides available at
http://ufal.mff.cuni.cz/mtm16/files/14-phrase-based-smt-ulrich-germann.pdf

Decoding

based on slides originally by P. Koehn, edited by M. Huck (and possibly others)

Given the model, find the best translation

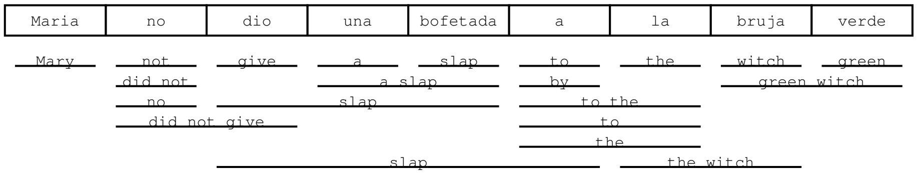
$$\mathbf{e}_{\text{best}} = \text{argmax}_{\mathbf{e}}\ p(\mathbf{e}\,|\,\mathbf{f})$$

We use the "Viterbi approximation"

$$(a, \mathbf{e})_{\text{best}} = \text{argmax}_{(a,\mathbf{e})}\ p(a, \mathbf{e}\,|\mathbf{f})$$

- This is a search problem - a big one.
  - Dynamic programming
  - Approximation (beam search)
  - Model restrictions (reordering)

Decoding



Maria	no	dio	una	bofetada	a	la	bruja	verde
Mary	not	give	a	slap	to	the	witch	green
	did not		a slap		by		green witch	
	no	slap			to the			
	did not give				to			
					the			
		slap				the witch		

- many different ways to *segment* the input sentence into phrases
- many different ways to *translate* each phrase

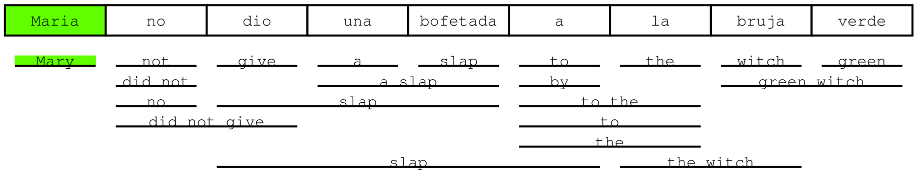
Hypothesis Expansion

Maria	no	dio	una	bofetada	a	la	bruja	verde
Mary	not	give	a	slap	to	the	witch	green
	did not		a slap		by		green witch	
	no	slap			to the			
	did not give				to			
					the			
		slap				the witch		

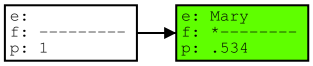
```
e:
f: ---------
p: 1
```

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

Hypothesis Expansion


Maria no dio una bofetada a la bruja verde

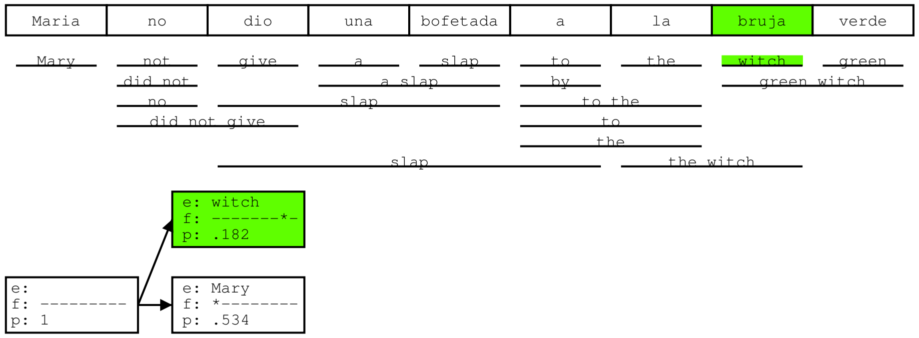
Mary not give a slap to the witch green
did not a slap by green witch
no slap to the
did not give to
the
slap the witch


e:
f: ---------
p: 1

e: Mary
f: *--------
p: .534

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

Hypothesis Expansion



Maria	no	dio	una	bofetada	a	la	bruja	verde

Mary
not
did not
no
did not give
give
a
a slap
slap
slap
slap
to
by
to the
to
the
the
the witch
witch
green
green witch

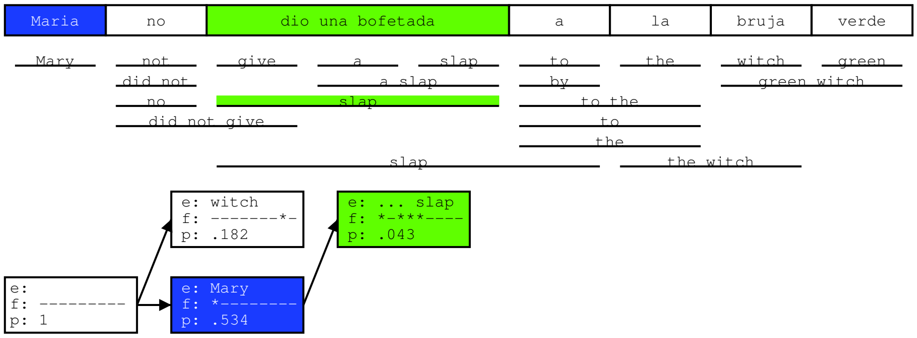
```
e: witch
f: -------*-
p: .182

e:
f: ---------
p: 1

e: Mary
f: *--------
p: .534
```

- Add another *hypothesis*

Hypothesis Expansion



Maria | no | dio una bofetada | a | la | bruja | verde

Mary
not
did not
no
did not give
give
a
a slap
slap
slap
to
by
to the
to
the
the
witch
the witch
green
green witch

e:
f: ---------
p: 1

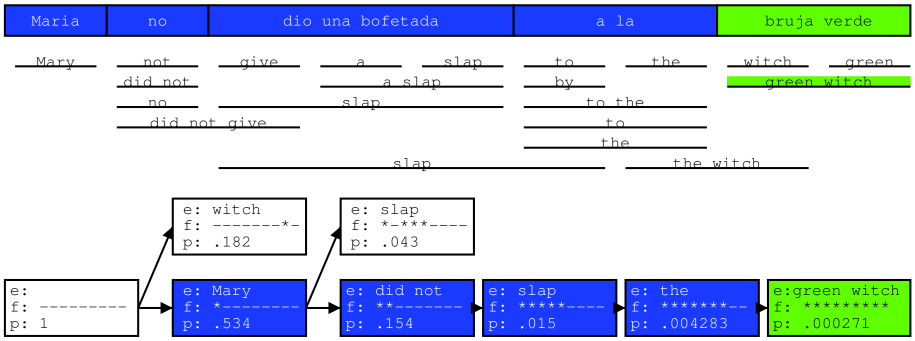
e: witch
f: -------*-
p: .182

e: Mary
f: *--------
p: .534

e: ... slap
f: *-***----
p: .043

- Further *hypothesis expansion*

Hypothesis Expansion



Maria | no | dio una bofetada | a la | bruja verde

Mary

not

did not

no

did not give

give

a

slap

a slap

slap

slap

to

by

to the

to

the

the

the witch

witch

green

green witch

e:
f: ---------
p: 1

e: witch
f: -------*-
p: .182

e: slap
f: *-***----
p: .043

e: Mary
f: *--------
p: .534

e: did not
f: **-------
p: .154

e: slap
f: *****----
p: .015

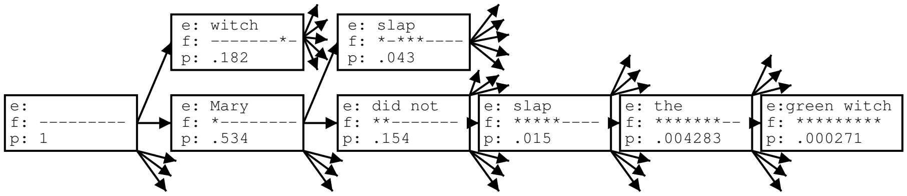
e: the
f: *******--
p: .004283

e:green witch
f: *********
p: .000271

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

Hypothesis Expansion

Maria	no	dio	una	bofetada	a	la	bruja	verde
Mary	not	give	a	slap	to	the	witch	green
	did not		a slap		by		green witch	
	no	slap			to the			
	did not give				to			
					the			
		slap				the witch		



e:
f: ---------
p: 1

e: witch
f: -------*-
p: .182

e: slap
f: *-***----
p: .043

e: Mary
f: *--------
p: .534

e: did not
f: **-------
p: .154

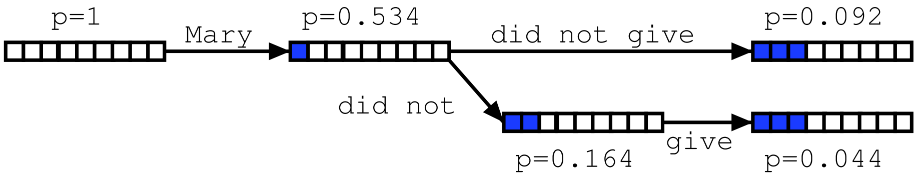
e: slap
f: *****----
p: .015

e: the
f: *******--
p: .004283

e:green witch
f: *********
p: .000271

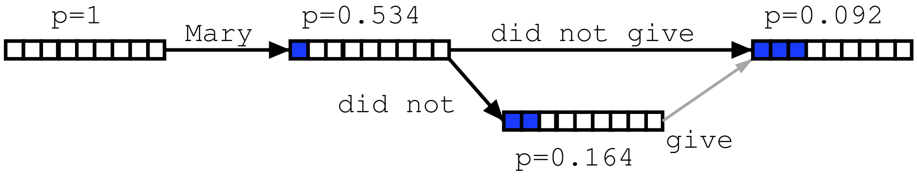
Adding more hypothesis ⇒ *Explosion* of search space

Hypothesis Recombination



- Different paths to the *same* partial translation

Hypothesis Recombination

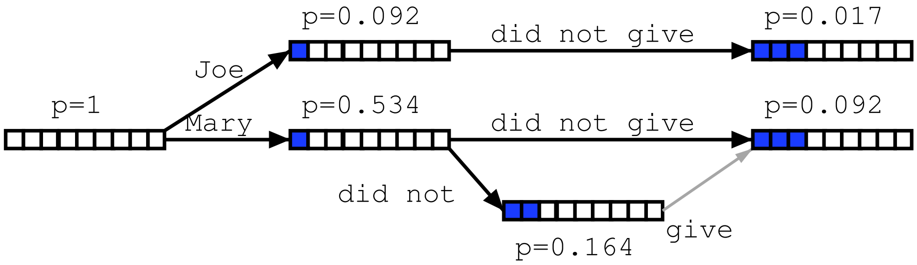


- Different paths to the same partial translation

⇒ *Combine paths*

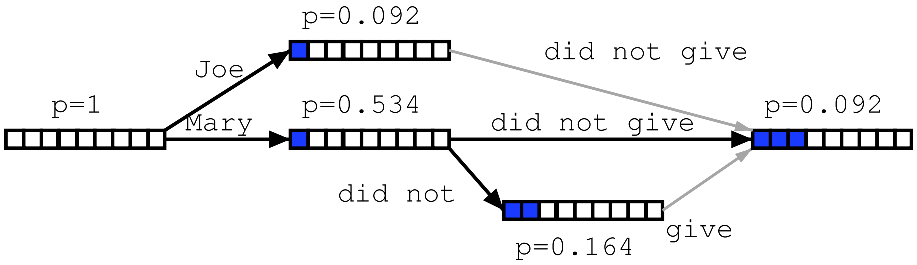
  - *drop weaker* path
  - keep pointer from weaker path (for lattice generation)

Hypothesis Recombination



- Recombined hypotheses do *not* have to *match completely*
- No matter what is added, weaker path can be dropped, if:
  - *last* $n - 1$ *English words* match (matters for language model)
  - *foreign word coverage* vectors match (affects future path)

Hypothesis Recombination



- Recombined hypotheses do not have to match completely
- No matter what is added, weaker path can be dropped, if:
  - last $n - 1$ English words match (matters for language model)
  - foreign word coverage vectors match (effects future path)

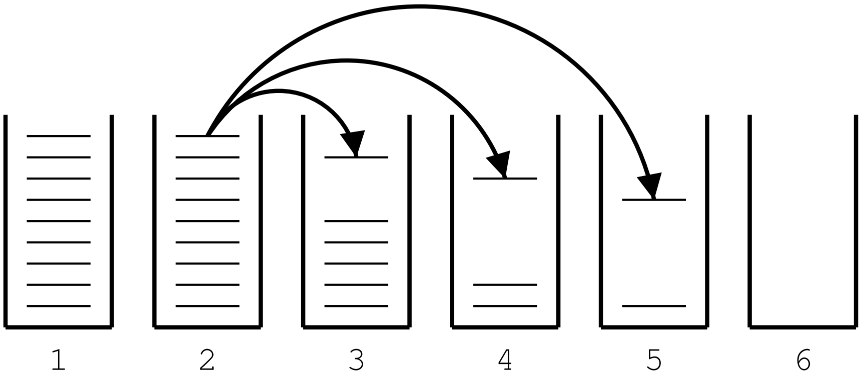
⇒ *Combine paths*

Beam Search

heuristically *discard* weak hypotheses early

- it is better to organize hypotheses in stacks (actually: priority queues), e.g. by
  - *same* foreign words covered
  - *same number* of foreign words covered
- compare hypotheses in stacks, discard bad ones
  - histogram pruning: keep top *k* hypotheses in each stack (e.g., *k*=100)
  - threshold pruning: keep hypotheses that are at most α times the cost of best hypothesis in stack (e.g., $\alpha = 0.001$)

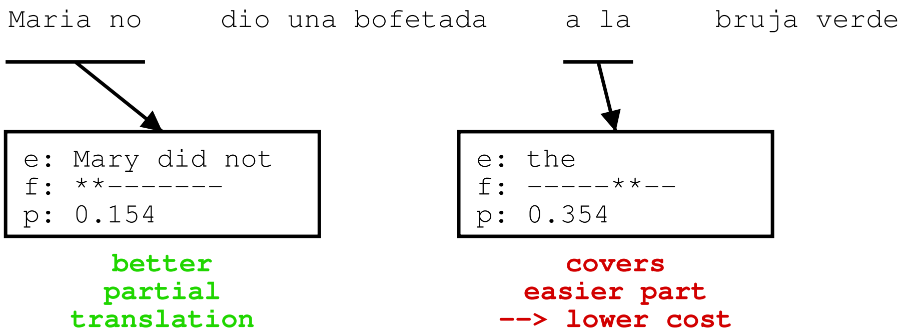
Hypothesis Stacks



- Organization of hypotheses 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



Maria 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

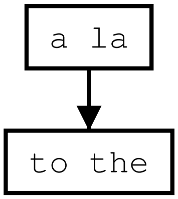
covers easier part --> lower cost

- Hypothesis that covers *easy part* of sentence is preferred

⇒ Need to consider future cost of uncovered parts

Future Cost Estimation

Step 1: estimate future cost for each *translation option*

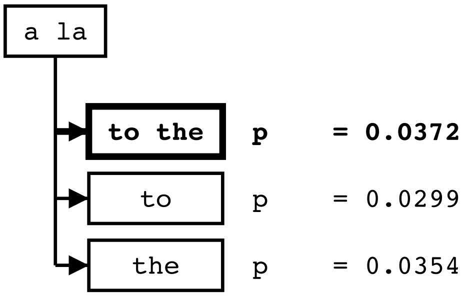


- look up translation model cost
- estimate language model cost (no prior context)
- ignore reordering model cost

⇒ LM * TM = p(to) * p(the|to) * p(to the|a la)

Future Cost Estimation

Step 2: find *cheapest cost* (highest probability) among translation options



a la

to the **p = 0.0372**

to p = 0.0299

the p = 0.0354

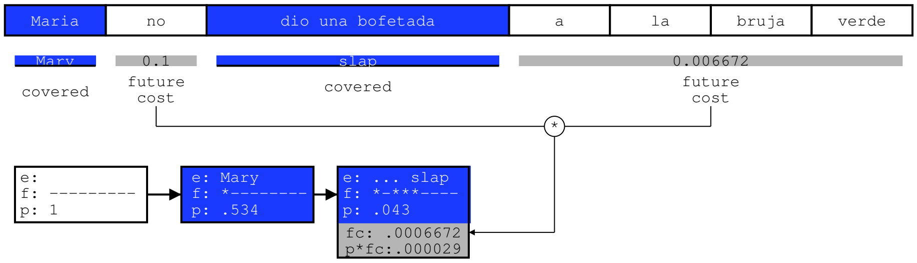
Future Cost Estimation

Step 3: Find *lowest future cost* for each possible span

- Cost of translation option for that span, *or*
- Sum of costs of covering subspans

⇒ Pre-compute future costs, bottom up., via dynamic programming.

Future Cost Estimation: Application



Maria | no | dio una bofetada | a | la | bruja | verde

Mary | 0.1 | slap | 0.006672

covered | future cost | covered | future cost

*

```
e:
f: ---------
p: 1
```

```
e: Mary
f: *--------
p: .534
```

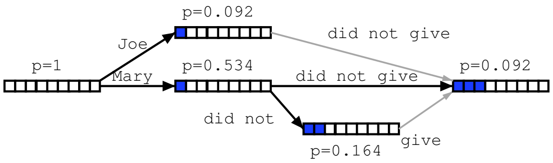
```
e: ... slap
f: *-***----
p: .043
fc: .0006672
p*fc:.000029
```

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

Limits on Reordering

- Reordering may be limited
  - Monotone translation: No reordering at all
  - Only phrase movements of at most *d* words
- Reordering limits *speed* up search (polynomial instead of exponential)
- Current reordering models are weak, so limits *improve* translation quality

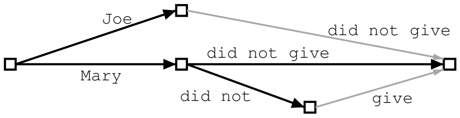
Word Lattice Generation



- Search graph can be easily converted into a word lattice
  - can be further mined for N-best lists

  ⇒ enables reranking approaches

  ⇒ enables discriminative training



Sample N-Best List

- Simple N-best list:

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

Summary

- Left-to-right decoding as search
- Hypothesis recombination
- Pruning
- Future cost estimation
- Word lattices and *n*-best lists