UPPSALA
UNIVERSITET

Phrase-based Statistical Machine Translation

Sara Stymne

2018-09-20

Some slides from Philipp Koehn

UPPSALA
UNIVERSITET

Lecture outline

- What is phrase-based SMT?
- Phrase-based modeling
- Training
- Log-linear models and features
- Tuning (?)
- Factored models (?)

UPPSALA
UNIVERSITET

Word-based vs Phrase-based SMT

- Word-based models translate **words** as atomic units
- Phrase-based models translate **phrases** as atomic units

UPPSALA
UNIVERSITET

Word-based vs Phrase-based SMT

- Word-based models translate **words** as atomic units
- Phrase-based models translate **phrases** as atomic units
  - A phrase is a continuous sequence of words
  - Not necessarily a linguistic phrase

UPPSALA
UNIVERSITET

Advantages of Phrase-based SMT

- Many-to-many translation can handle
  - Non-compositional phrases
    kick the bucket – ins Gras beissen (lit: into the grass bite)
  - Compounds
    bläddерblocksblad – flipchart paper
- Use of local context
  - Local word order
    affaires extérieure – external affairs
  - Local agreement issues
    det röda blocket – the red block
    den röda konen – the red cone

UPPSALA
UNIVERSITET

Advantages of Phrase-based SMT II

- Translating phrases helps to reduce translation ambiguities
- Phrases of arbitrary length: sometimes the entire sentence might be covered by a phrase
- Simpler model: no more need to explicitly model the concepts of fertility, insertion and deletion of words

UPPSALA
UNIVERSITET

Word-based SMT: Generative Model

Bakom huset hittade polisen en stor mängd narkotika .

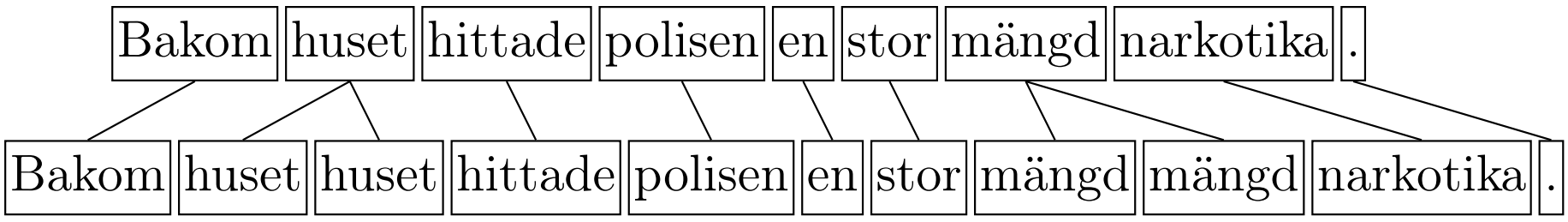
UPPSALA
UNIVERSITET

Word-based SMT: Generative Model

Bakom	huset	hittade	polisen	en	stor	mängd	narkotika	.

UPPSALA
UNIVERSITET

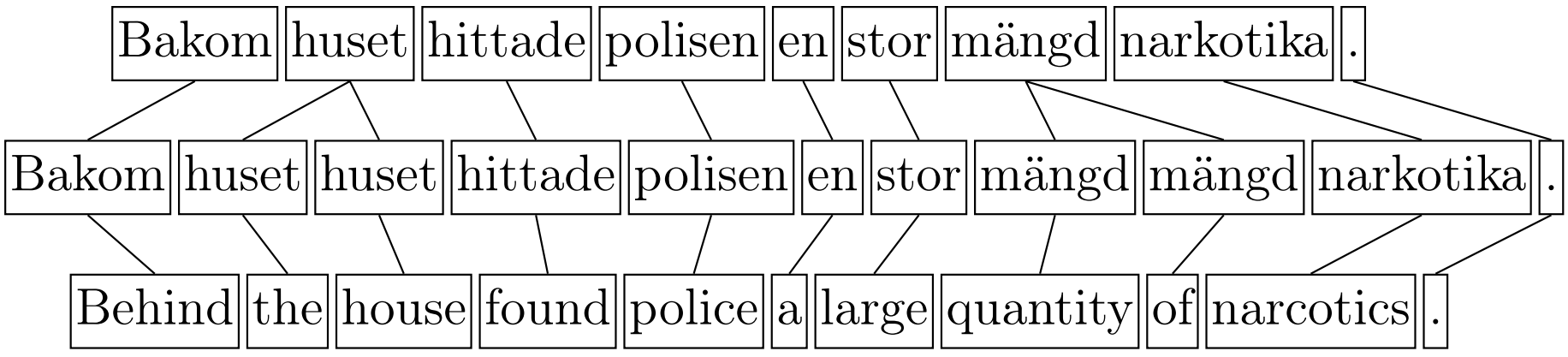
Word-based SMT: Generative Model



1. Fertility

UPPSALA
UNIVERSITET

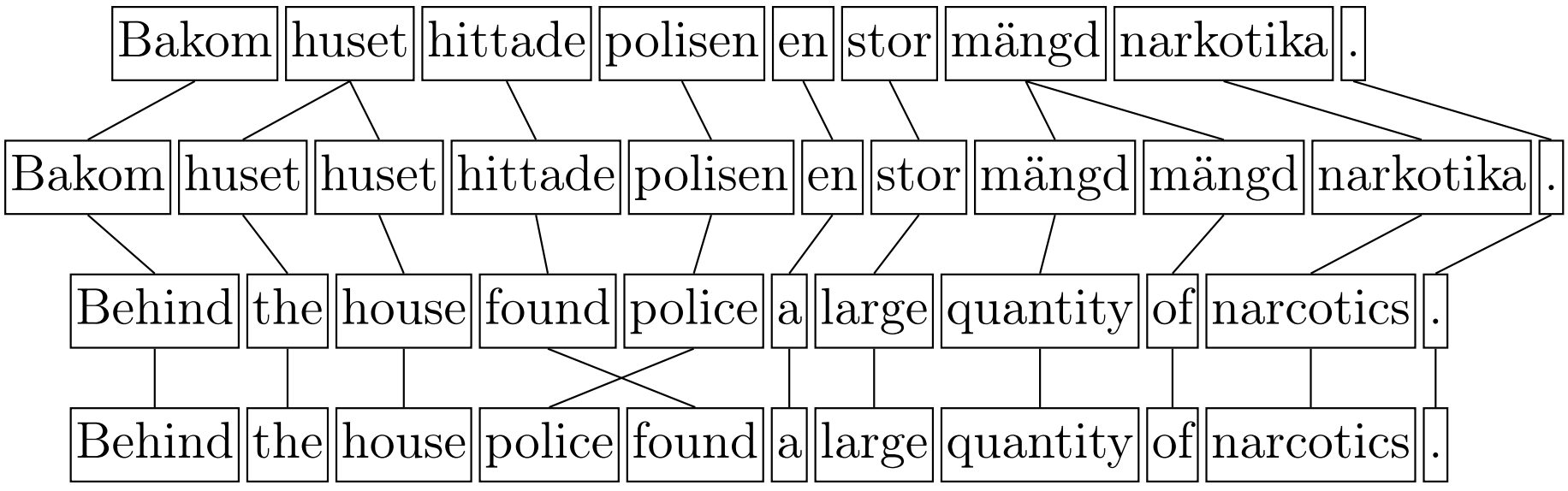
Word-based SMT: Generative Model



1. Fertility
2. Word translation

UPPSALA
UNIVERSITET

Word-based SMT: Generative Model



1. Fertility
2. Word translation
3. Output ordering

UPPSALA
UNIVERSITET

Phrase-based SMT: Generative Model

Bakom huset hittade polisen en stor mängd narkotika .

UPPSALA
UNIVERSITET

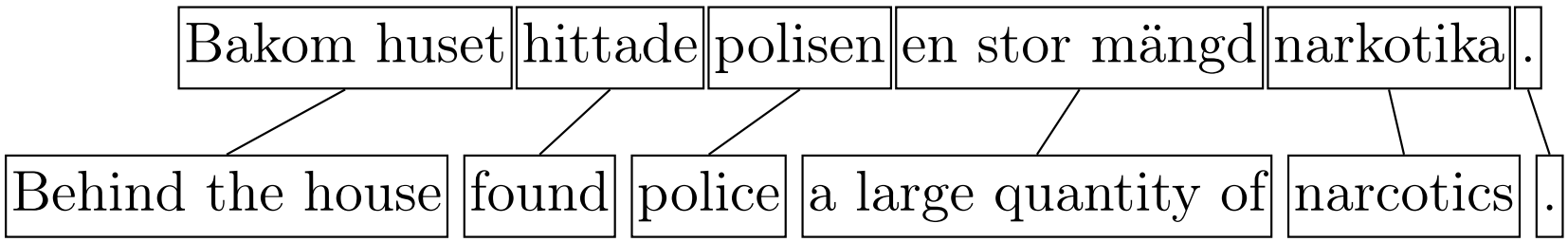
Phrase-based SMT: Generative Model

Bakom huset	hittade	polisen	en stor mängd	narkotika	.

1. Phrase segmentation

UPPSALA
UNIVERSITET

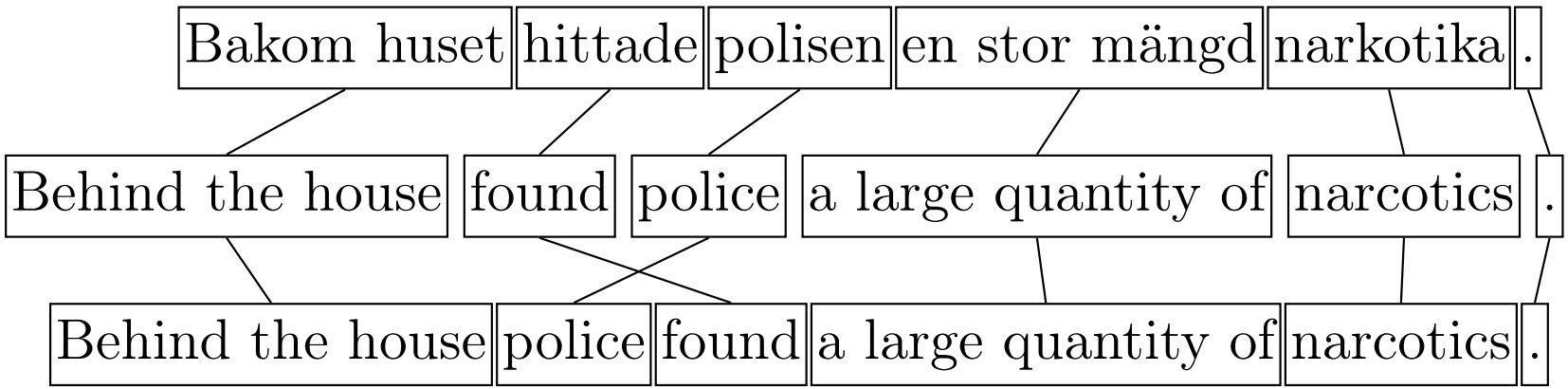
Phrase-based SMT: Generative Model



1. Phrase segmentation
2. Phrase translation

UPPSALA
UNIVERSITET

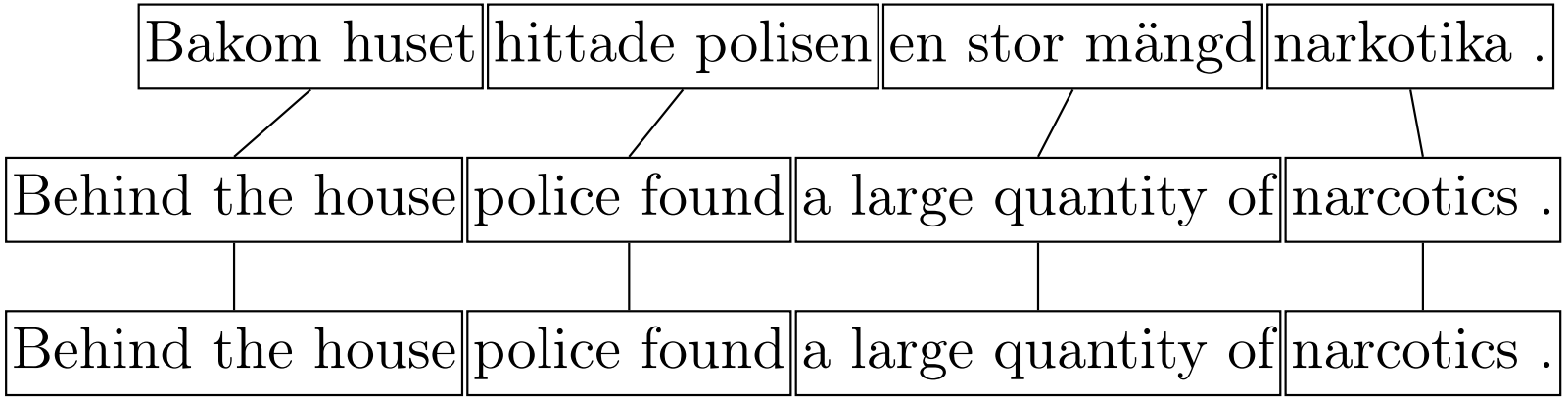
Phrase-based SMT: Generative Model



1. Phrase segmentation
2. Phrase translation
3. Output ordering

UPPSALA
UNIVERSITET

Phrase-based SMT: Alternative segmentation



1. Phrase segmentation
2. Phrase translation
3. Output ordering

UPPSALA
UNIVERSITET

Phrase translation table

- Main knowledge source: table with phrase translations and their probabilities
- Example: phrase translations for **naturligtvis**

Translation	**Probability** $\phi(\bar{t}\|\bar{s})$
of course	0.5
naturally	0.3
of course ,	0.15
, of course ,	0.05

UPPSALA
UNIVERSITET

Real example

- Phrase translations for **begreppet** learned from the Europarl corpus

English	$\phi(\bar{t}\|\bar{s})$	**English**	$\phi(\bar{t}\|\bar{s})$
the	0.226415	the news	0.012816
told	0.169811	the report	0.008544
announcement	0.075472	the information	0.008544
message	0.056604	the back	0.004272
news	0.056604	the suspension	0.004272
information	0.037736	the death	0.004272
informed	0.037736	this announcement	0.002848
learnt	0.037736	this news	0.002136
peace of mind by ensuring	0.027778	a message	0.001539
insight	0.018868	his answer	0.000356
the announcement	0.017088	were told	0.000229
the message	0.012816	the back and	2.917e-05

UPPSALA
UNIVERSITET

Real example

- Phrase translations for **begreppet** learned from the Europarl corpus

English	$\phi(\bar{t}\|\bar{s})$	**English**	$\phi(\bar{t}\|\bar{s})$
the	0.226415	the news	0.012816
told	0.169811	the report	0.008544
announcement	0.075472	the information	0.008544
message	0.056604	the back	0.004272
news	0.056604	the suspension	0.004272
information	0.037736	the death	0.004272
informed	0.037736	this announcement	0.002848
learnt	0.037736	this news	0.002136
peace of mind by ensuring	0.027778	a message	0.001539
insight	0.018868	his answer	0.000356
the announcement	0.017088	were told	0.000229
the message	0.012816	the back and	2.917e-05

- lexical variation (announcement, message, news, told, . . .)

UPPSALA
UNIVERSITET

Real example

- Phrase translations for **begreppet** learned from the Europarl corpus

English	$\phi(\bar{t}\|\bar{s})$	**English**	$\phi(\bar{t}\|\bar{s})$
the	0.226415	the news	0.012816
told	0.169811	the report	0.008544
announcement	0.075472	the information	0.008544
message	0.056604	the back	0.004272
news	0.056604	the suspension	0.004272
information	0.037736	the death	0.004272
informed	0.037736	this announcement	0.002848
learnt	0.037736	this news	0.002136
peace of mind by ensuring	0.027778	a message	0.001539
insight	0.018868	his answer	0.000356
the announcement	0.017088	were told	0.000229
the message	0.012816	the back and	2.917e-05

- lexical variation (announcement, message, news, told, . . .)
- Morphological variation (information, informed)

UPPSALA
UNIVERSITET

Real example

- Phrase translations for **begreppet** learned from the Europarl corpus

English	$\phi(\bar{t}\|\bar{s})$	**English**	$\phi(\bar{t}\|\bar{s})$
the	0.226415	the news	0.012816
told	0.169811	the report	0.008544
announcement	0.075472	the information	0.008544
message	0.056604	the back	0.004272
news	0.056604	the suspension	0.004272
information	0.037736	the death	0.004272
informed	0.037736	**this** announcement	0.002848
learnt	0.037736	this news	0.002136
peace of mind by ensuring	0.027778	**a** message	0.001539
insight	0.018868	his answer	0.000356
the announcement	0.017088	**were** told	0.000229
the message	0.012816	the back and	2.917e-05

- lexical variation (announcement, message, news, told, . . .)
- Morphological variation (information, informed)
- Included function words (the, a, were, this)

UPPSALA
UNIVERSITET

Real example

- Phrase translations for **begreppet** learned from the Europarl corpus

English	$\phi(\bar{t}\|\bar{s})$	**English**	$\phi(\bar{t}\|\bar{s})$
the	0.226415	the news	0.012816
told	0.169811	the report	0.008544
announcement	0.075472	the information	0.008544
message	0.056604	the back	0.004272
news	0.056604	the suspension	0.004272
information	0.037736	the death	0.004272
informed	0.037736	this announcement	0.002848
learnt	0.037736	this news	0.002136
peace of mind by ensuring	0.027778	a message	0.001539
insight	0.018868	his answer	0.000356
the announcement	0.017088	were told	0.000229
the message	0.012816	**the back and**	2.917e-05

- lexical variation (announcement, message, news, told, . . .)
- Morphological variation (information, informed)
- Included function words (the, a, were, this)
- Noise (the, the back and)

UPPSALA
UNIVERSITET

Phrases

- Model is not limited to linguistic phrases
  (noun phrases, verb phrases, prepositional phrases)
- Example of useful non-linguistic phrases:
  - det finns – there is/are
  - put off – skjuta upp

UPPSALA
UNIVERSITET

Phrases

- Model is not limited to linguistic phrases
  (noun phrases, verb phrases, prepositional phrases)
- Example of useful non-linguistic phrases:
  - det finns – there is/are
  - put off – skjuta upp
- Experiments have shown that limitation to only linguistic phrases hurts quality

UPPSALA
UNIVERSITET

Probabilistic model

- Bayes rule

$$t_{best} = \arg\max_t p(t|s) \tag{1}$$
$$= \arg\max_t p(s|t)p_{LM}(t)$$

  - translation model: $p(s|t)$
  - language model: $p_{LM}(t)$

UPPSALA
UNIVERSITET

Probabilistic model

- Bayes rule

$$t_{best} = \arg\max_t p(t|s) \tag{1}$$
$$= \arg\max_t p(s|t)p_{LM}(t)$$

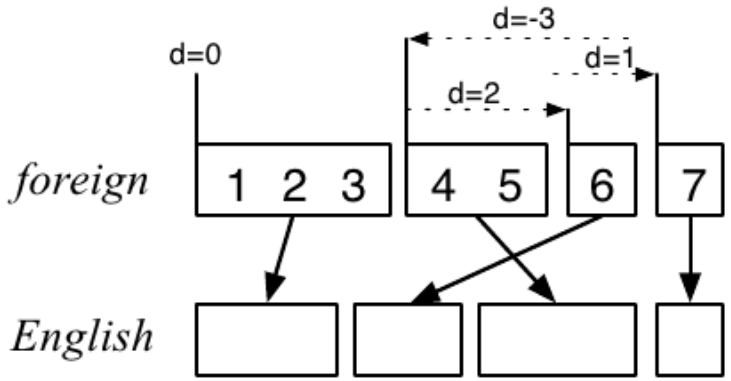
  - translation model: $p(s|t)$
  - language model: $p_{LM}(t)$
- Decomposition of translation model

$$p(\bar{s}_1^I|\bar{t}_1^I) = \prod_{i=1}^{I} \phi(\bar{s}_i|\bar{t}_i)d(\text{start}_i - \text{end}_{i-1} - 1)$$

  - phrase translation model: ϕ
  - reordering probability: d

UPPSALA
UNIVERSITET

Distance-based reordering



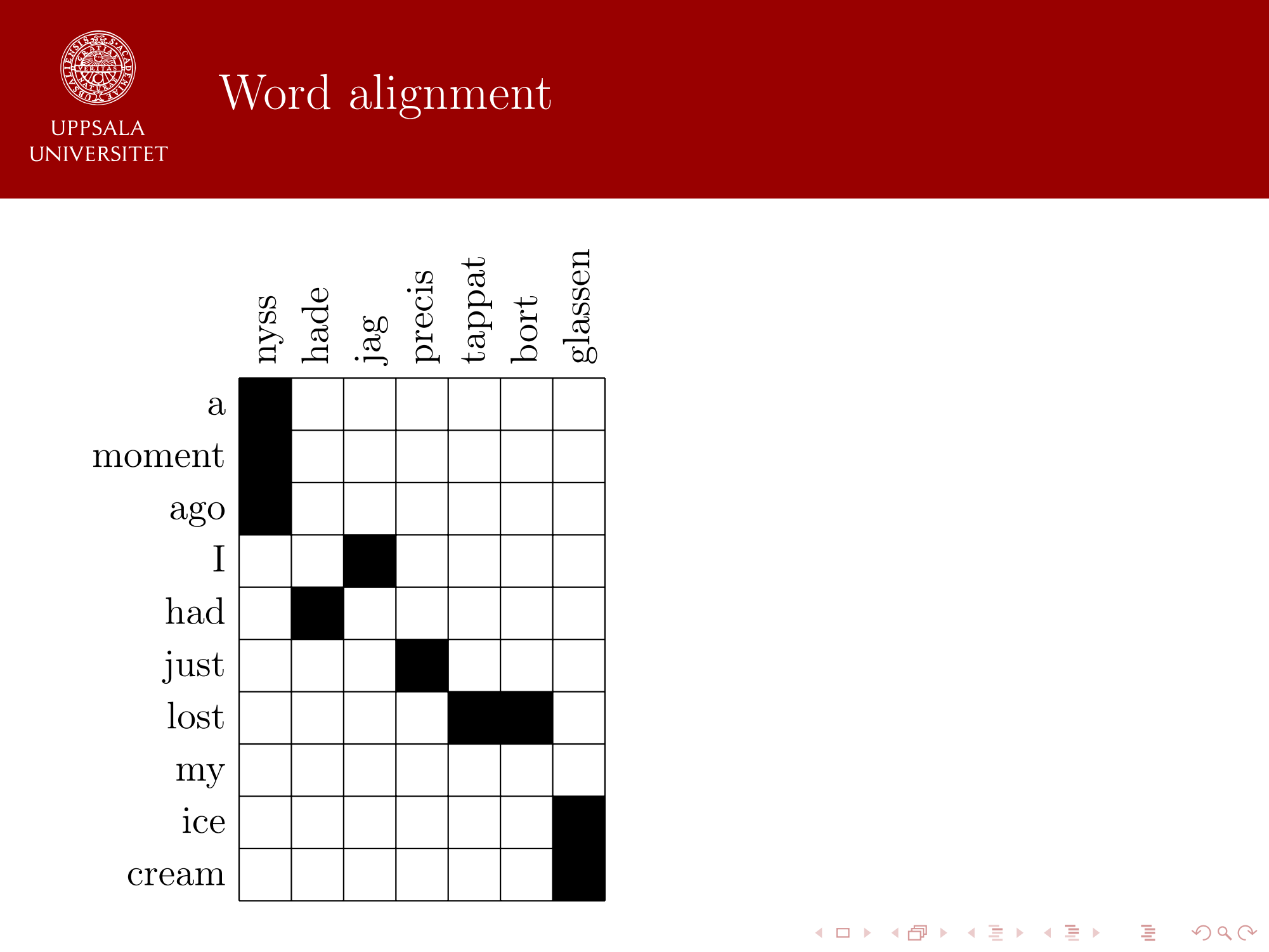
phrase	translates	movement	distance
1	1–3	start at beginning	0
2	6	skip over 4–5	+2
3	4–5	move back over 4–6	-3
4	7	skip over 6	+1

- distance $= \text{start}_i - \text{end}_{i-1} - 1$
- Scoring function: $d(x) = \alpha^{|x|}$ – exponential with distance

UPPSALA
UNIVERSITET

Learning a Phrase Translation Table

- Task: learn the model from a parallel corpus
- Three stages:
  - Word alignment
  - Extraction of phrase pairs
  - Scoring of phrase pairs



Word alignment

UPPSALA
UNIVERSITET

Obtaining a word alignment

- Using some word alignment software and models, often:
  - GIZA++
  - IBM1 – HMM – IBM3 – IBM4

UPPSALA
UNIVERSITET

Obtaining a word alignment

- Using some word alignment software and models, often:
  - GIZA++
  - IBM1 – HMM – IBM3 – IBM4
- Such models are directional
  - Gives 1–N links
  - Does not give M–1 or M–N links

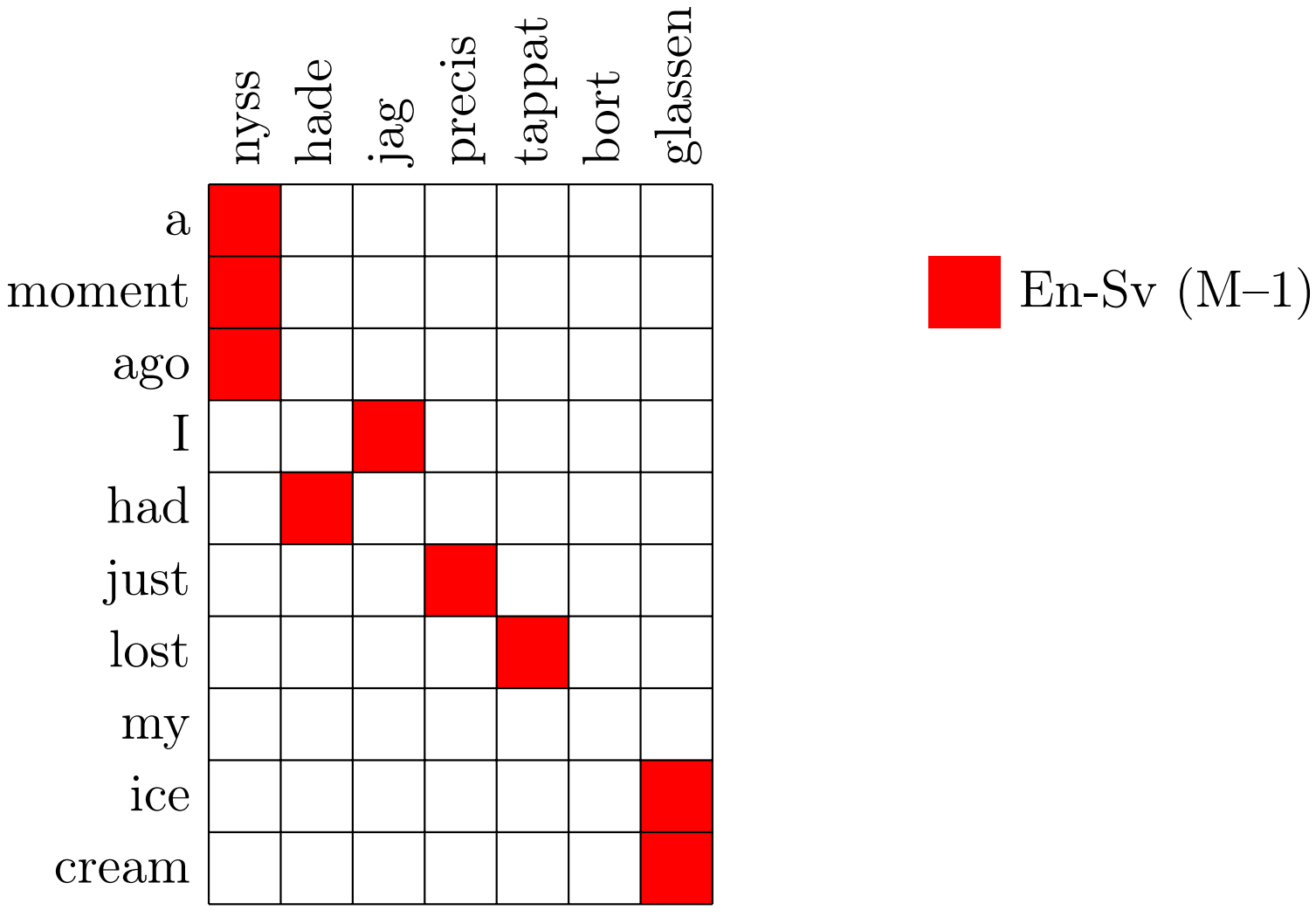
UPPSALA
UNIVERSITET

Obtaining a word alignment

- Using some word alignment software and models, often:
  - GIZA++
  - IBM1 – HMM – IBM3 – IBM4
- Such models are directional
  - Gives 1–N links
  - Does not give M–1 or M–N links
- We want all types of links!
- Solution: symmetrize directional alignments

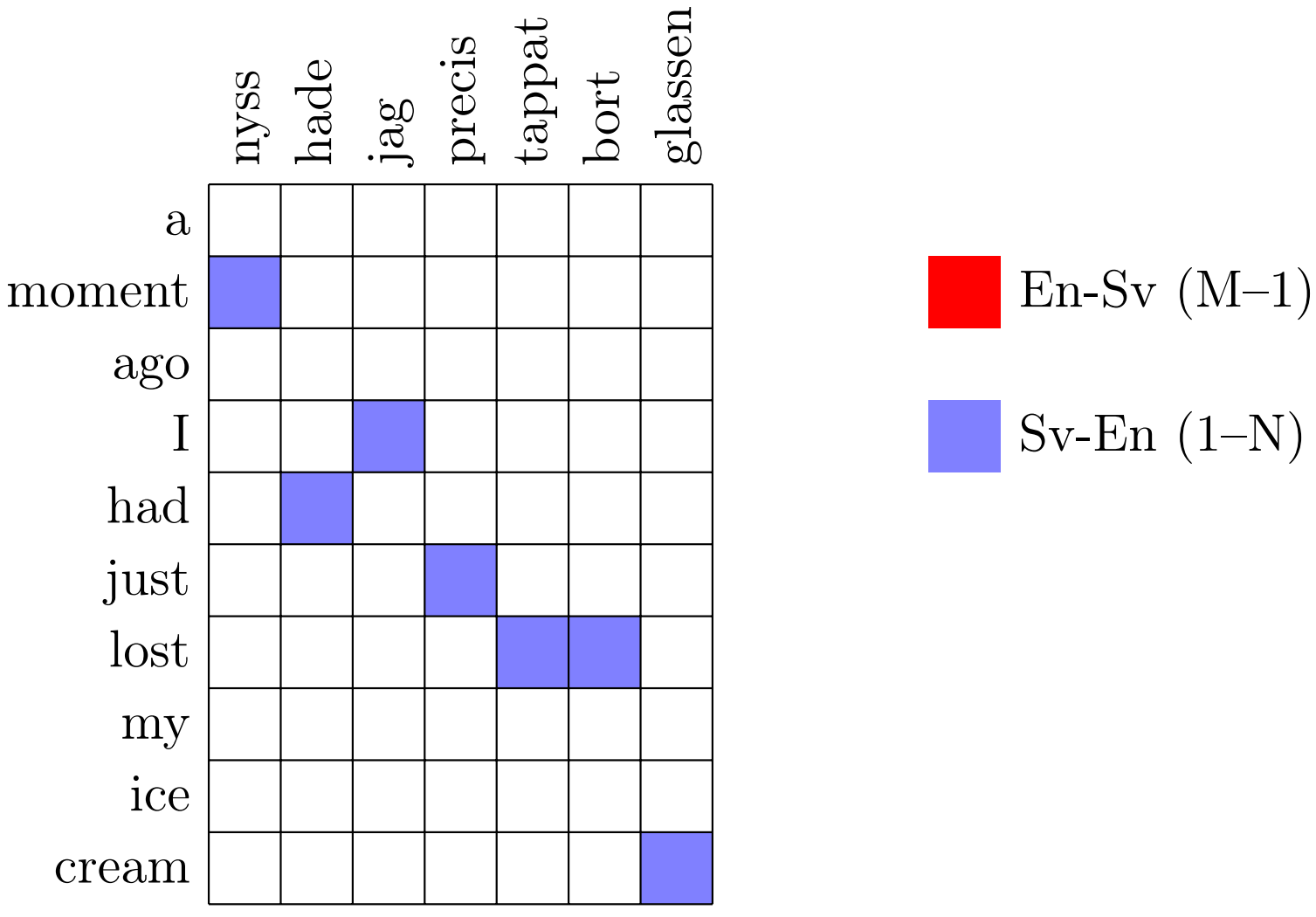
UPPSALA
UNIVERSITET

Directional word alignment



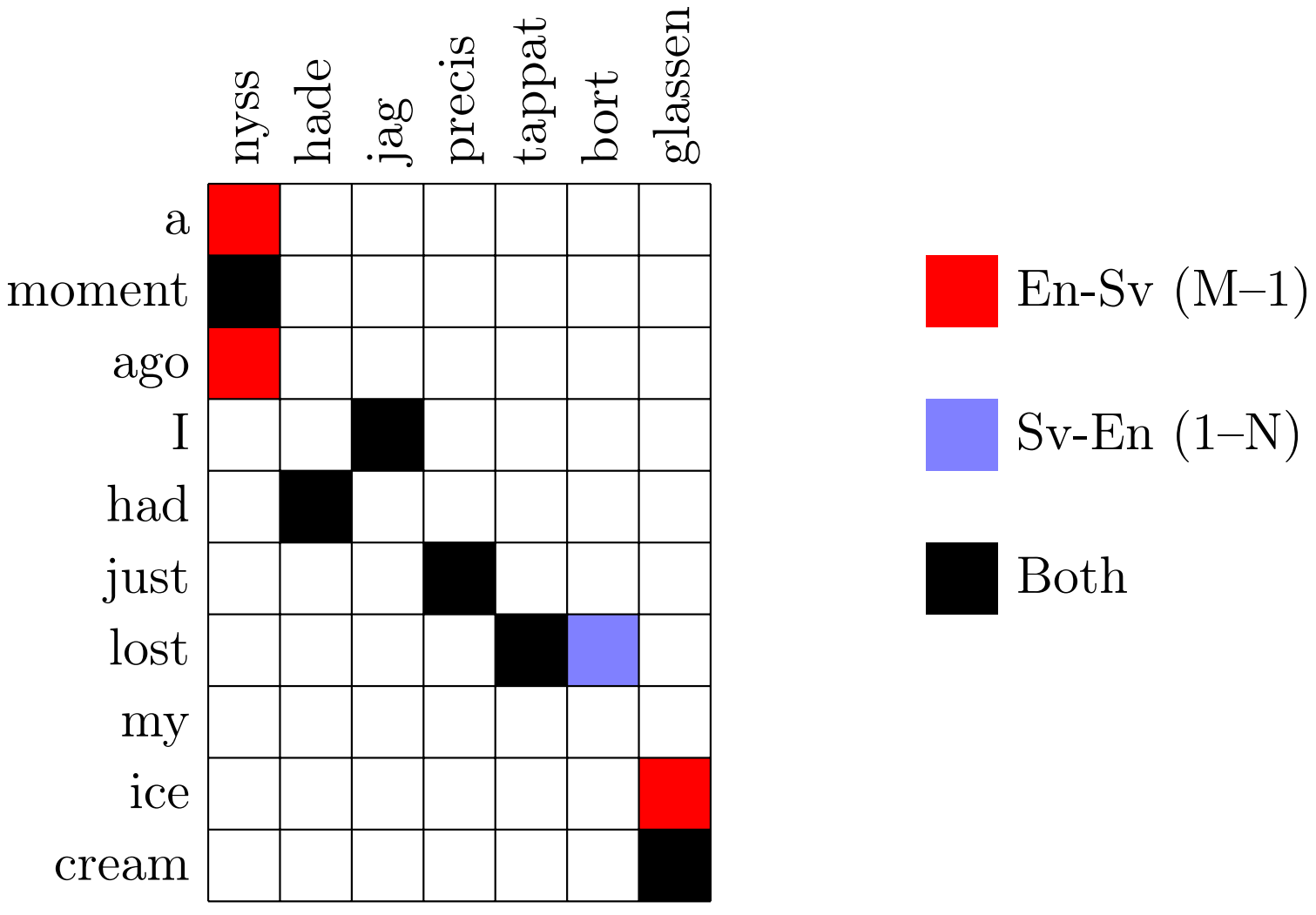
UPPSALA UNIVERSITET

Directional word alignment

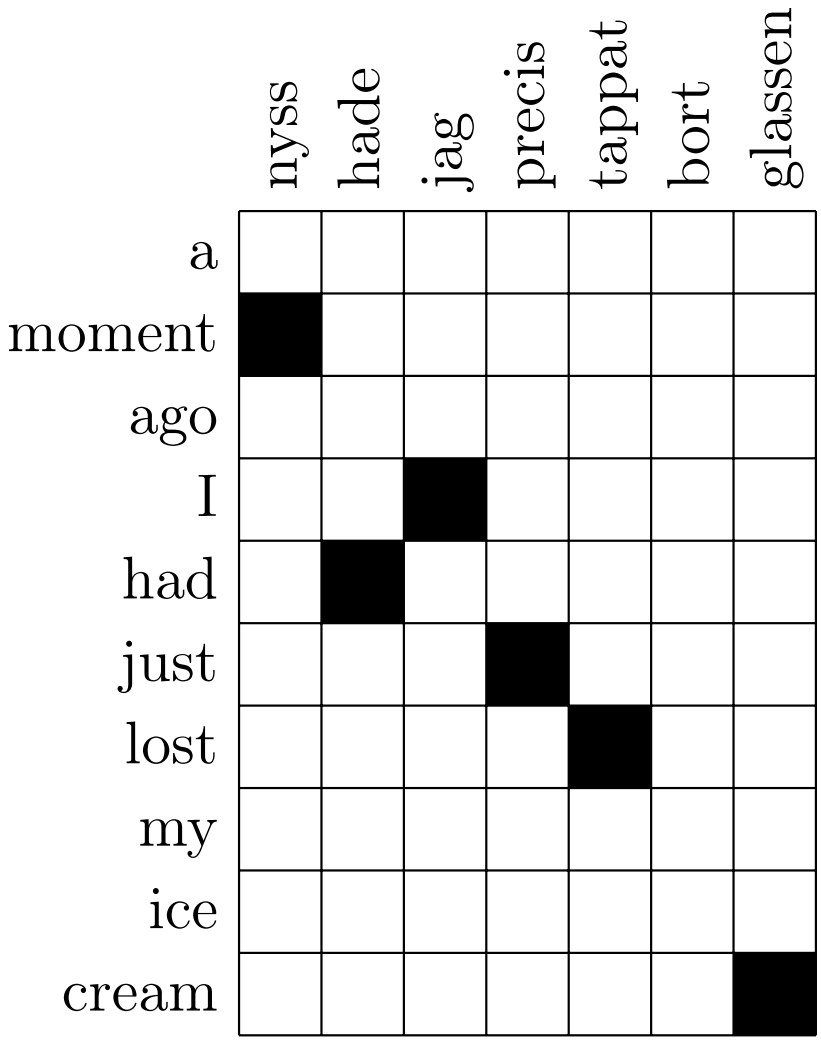


UPPSALA UNIVERSITET

Word alignment – symmetrization



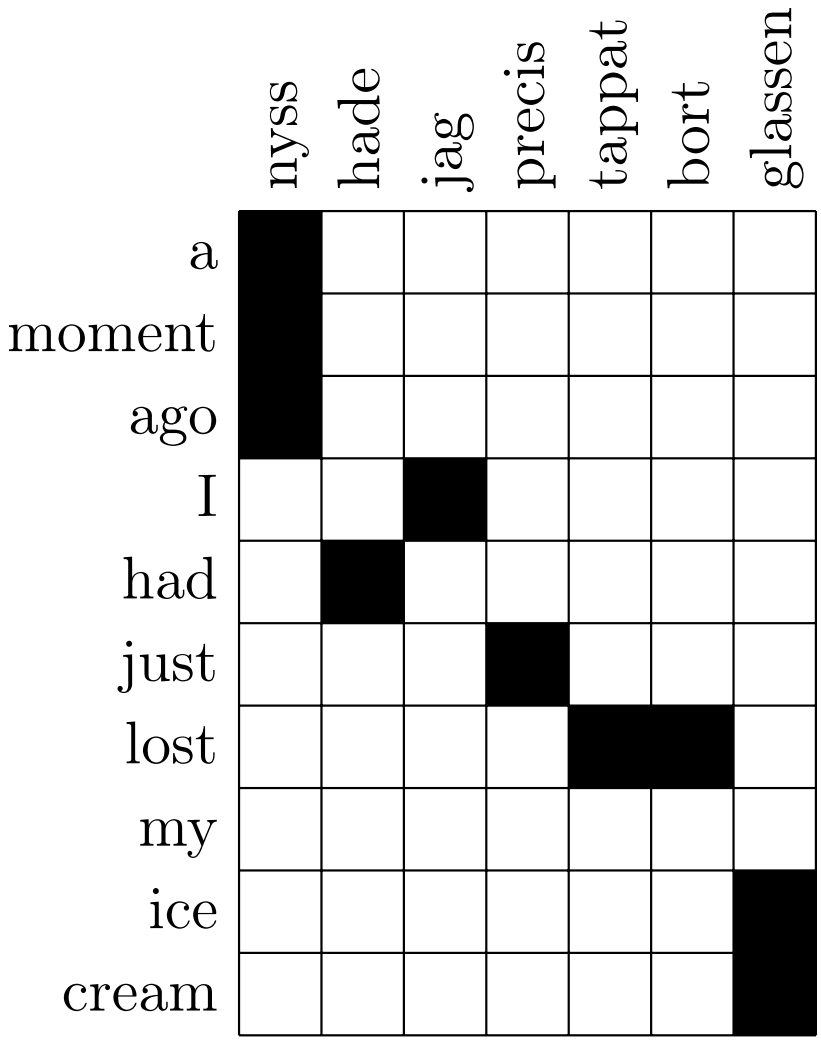
Word alignment – intersection



Intersection

UPPSALA
UNIVERSITET

Word alignment – union



Union

UPPSALA
UNIVERSITET

Word alignment symmetrization

- Intersection: too few links
- Union: too many links

English–Swedish alignment (Holmqvist, 2008)

	Precision	Recall
Intersection	90	75
Union	60	91

UPPSALA
UNIVERSITET

Word alignment symmetrization 2

- Solution: use heuristics!

UPPSALA
UNIVERSITET

Word alignment symmetrization 2

- Solution: use heuristics!
- grow-diag-final-and
  - Start with the intersection
  - Add links from the union:
    - Add diagonally adjacent links
    - Add links for unaligned words in a final step

UPPSALA
UNIVERSITET

Word alignment symmetrization 2

- Solution: use heuristics!
- grow-diag-final-and
  - Start with the intersection
  - Add links from the union:
    - Add diagonally adjacent links
    - Add links for unaligned words in a final step

	Precision	Recall
Intersection	90	75
Union	60	91
grow-diag-final-and	70	88

UPPSALA
UNIVERSITET

Word alignment symmetrization 2

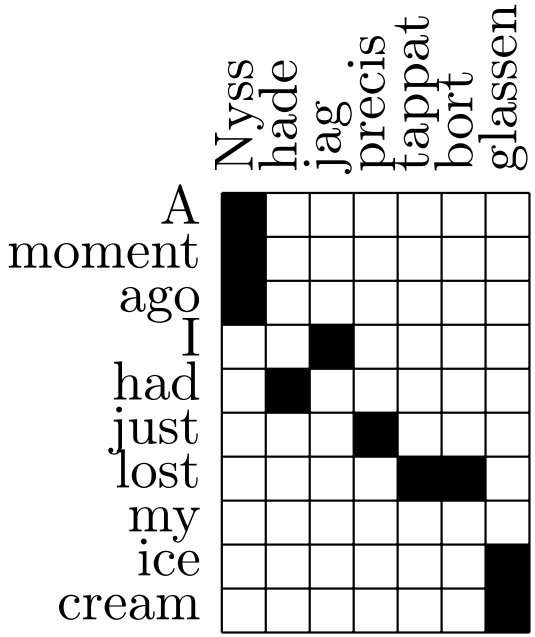
- Solution: use heuristics!
- grow-diag-final-and
  - Start with the intersection
  - Add links from the union:
    - Add diagonally adjacent links
    - Add links for unaligned words in a final step
- There are other symmetrization heuristics and word alignment methods
- In general: high recall is important for good SMT quality (with large training data)

	Precision	Recall
Intersection	90	75
Union	60	91
grow-diag-final-and	70	88

UPPSALA
UNIVERSITET

Extracting phrase pairs

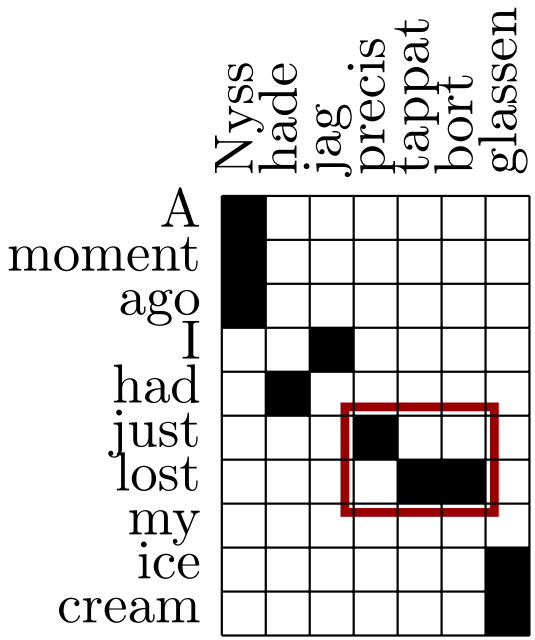
- Extract phrase pairs that are consistent with word alignments



UPPSALA
UNIVERSITET

Extracting phrase pairs

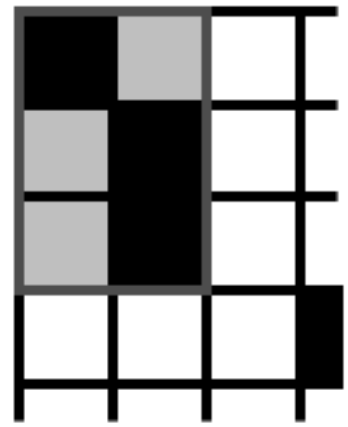
- Extract phrase pairs that are consistent with word alignments



just lost–precis tappat bort

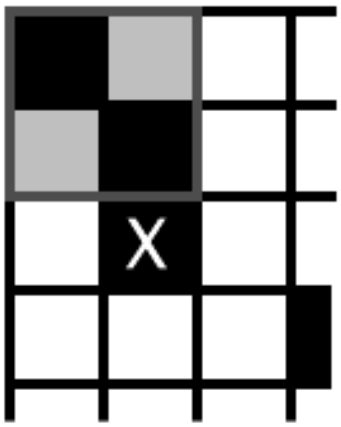
Consistent phrase pairs

All words of the phrase pairs have to align to each other



consistent

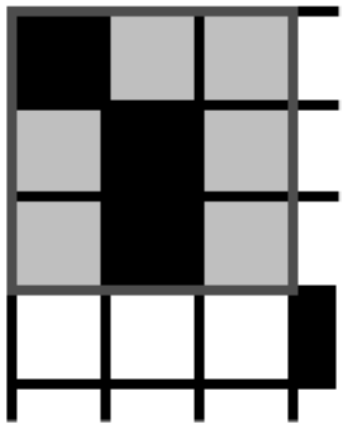
ok



inconsistent

violated

one alignment
point outside

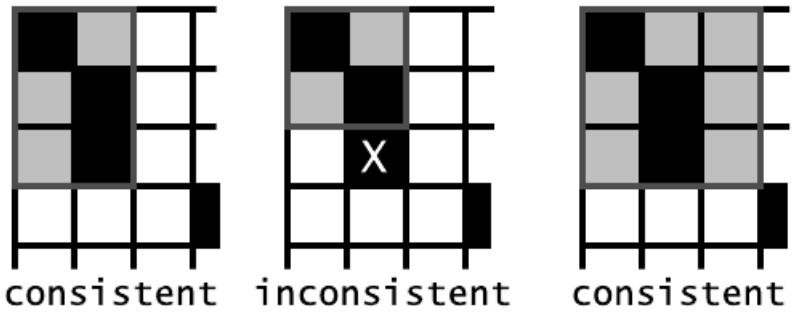


consistent

ok

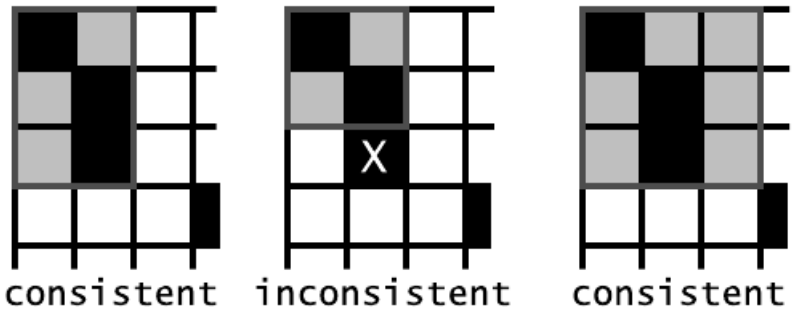
unaligned
word is fine

Phrase extraction definition



A phrase pair $(\overline{t}, \overline{s})$ is consistent with an alignment A, if all words $s_1, \ldots, s_m$ in $\overline{s}$ that have alignment points in A have these with words $t_1, \ldots, t_n$ in $\overline{t}$ and vice versa and at least one word in $\overline{t}$ is aligned to at least one word in $\overline{s}$

Phrase extraction definition

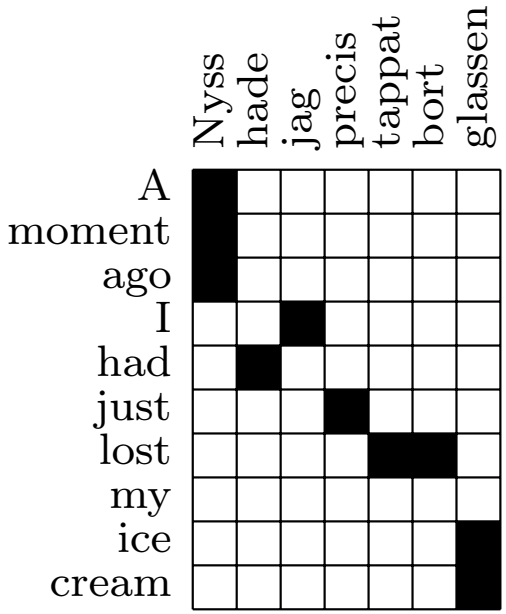


A phrase pair $(\overline{t}, \overline{s})$ is consistent with an alignment A, if all words $s_1, \ldots, s_m$ in $\overline{s}$ that have alignment points in A have these with words $t_1, \ldots, t_n$ in $\overline{t}$ and vice versa and at least one word in $\overline{t}$ is aligned to at least one word in $\overline{s}$

$$(\overline{t}, \overline{s}) \text{ consistent with } A \Leftrightarrow \forall t_i \in \overline{t} : (t_i, s_j) \in A \rightarrow s_j \in \overline{s}$$
$$\text{AND } \forall s_i \in \overline{s} : (t_i, s_j) \in A \rightarrow t_j \in \overline{t}$$
$$\text{AND } \exists t_i \in \overline{t}, s_i \in \overline{s} : (t_i, s_j) \in A$$

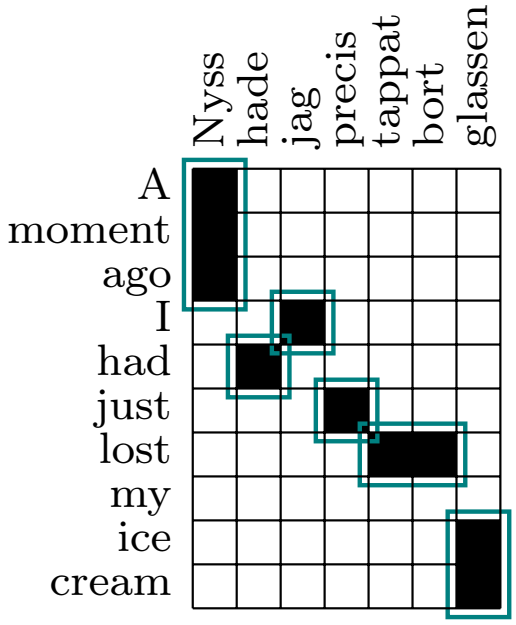
UPPSALA
UNIVERSITET

Phrases extracted



UPPSALA
UNIVERSITET

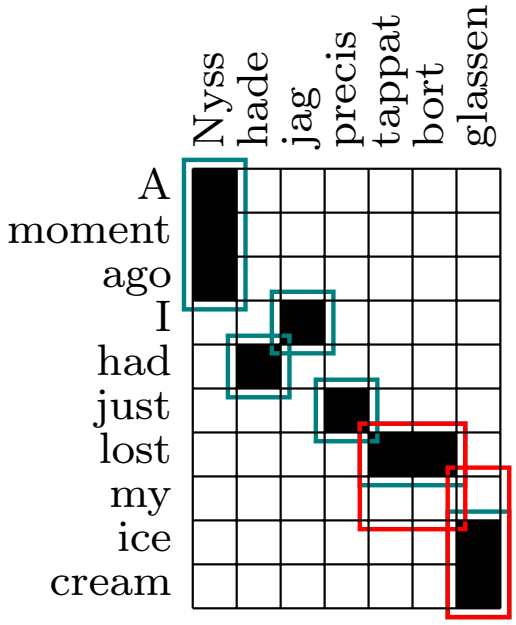
Phrases extracted



a moment ago–nyss, I–jag, had–hade, just–precis
lost–tappat bort, ice cream–glassen

UPPSALA
UNIVERSITET

Phrases extracted

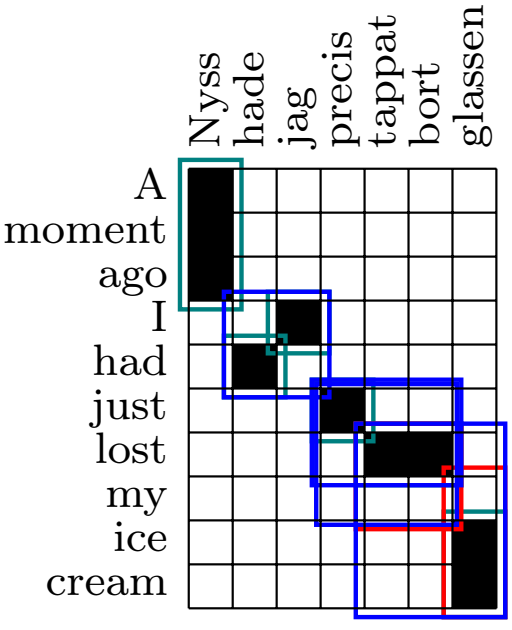


a moment ago–nyss, I–jag, had–hade, just–precis
lost–tappat bort, ice cream–glassen

lost my–tappat bort, my ice cream–glassen

UPPSALA
UNIVERSITET

Phrases extracted



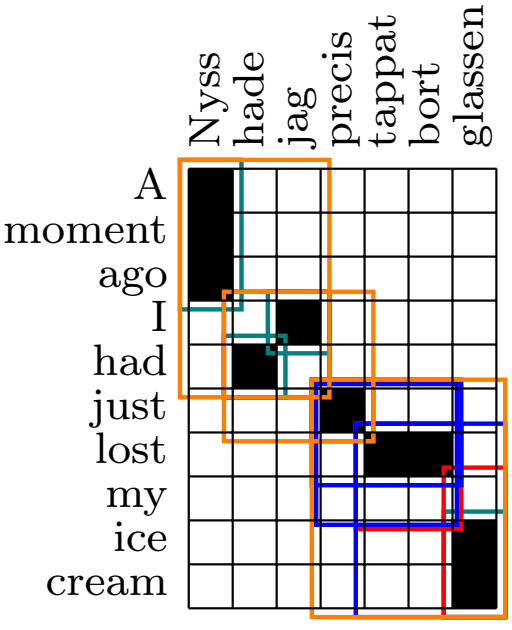
a moment ago–nyss, I–jag, had–hade, just–precis
lost–tappat bort, ice cream–glassen

lost my–tappat bort, my ice cream–glassen

I had–hade jag, lost my ice cream–tappat bort glassen
just lost–precis tappat bort, just lost my–precis tappat bort

UPPSALA
UNIVERSITET

Phrases extracted



a moment ago–nyss, I–jag, had–hade, just–precis
lost–tappat bort, ice cream–glassen

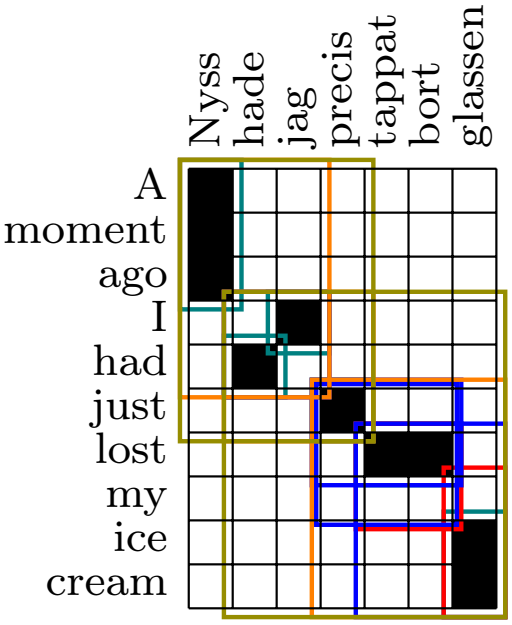
lost my–tappat bort, my ice cream–glassen

I had–hade jag, lost my ice cream–tappat bort glassen
just lost–precis tappat bort, just lost my–precis tappat bort

a moment ago I had–nyss hade jag, I had just–hade jag precis
just lost my ice cream–precis tappat bort glassen

UPPSALA
UNIVERSITET

Phrases extracted



a moment ago–nyss, I–jag, had–hade, just–precis
lost–tappat bort, ice cream–glassen

lost my–tappat bort, my ice cream–glassen

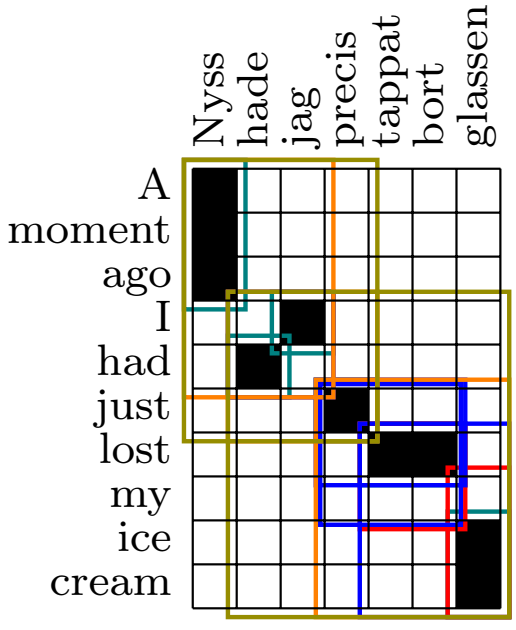
I had–hade jag, lost my ice cream–tappat bort glassen
just lost–precis tappat bort, just lost my–precis tappat bort

a moment ago I had–nyss hade jag, I had just–hade jag precis
just lost my ice cream–precis tappat bort glassen

a moment ago I had just–nyss hade jag precis
I had just lost my ice cream–hade jag precis tappat bort glassen

UPPSALA
UNIVERSITET

Phrases extracted



a moment ago–nyss, I–jag, had–hade, just–precis
lost–tappat bort, ice cream–glassen

lost my–tappat bort, my ice cream–glassen

I had–hade jag, lost my ice cream–tappat bort glassen
just lost–precis tappat bort, just lost my–precis tappat bort

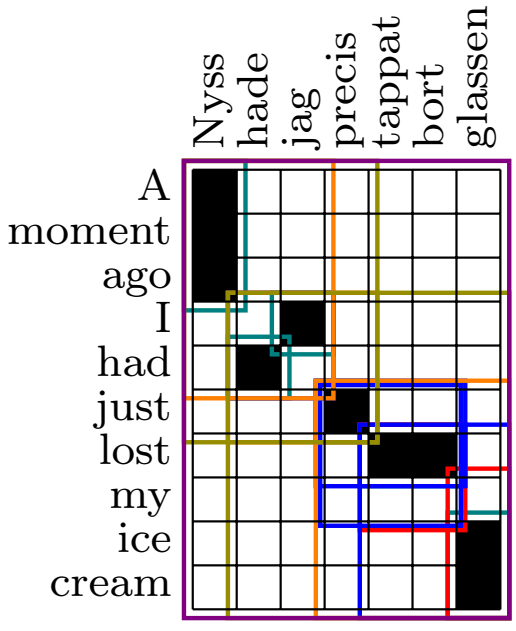
a moment ago I had–nyss hade jag, I had just–hade jag precis
just lost my ice cream–precis tappat bort glassen

a moment ago I had just–nyss hade jag precis
I had just lost my ice cream–hade jag precis tappat bort glassen

...

UPPSALA UNIVERSITET

Phrases extracted



a moment ago–nyss, I–jag, had–hade, just–precis
lost–tappat bort, ice cream–glassen

lost my–tappat bort, my ice cream–glassen

I had–hade jag, lost my ice cream–tappat bort glassen
just lost–precis tappat bort, just lost my–precis tappat bort

a moment ago I had–nyss hade jag, I had just–hade jag precis
just lost my ice cream–precis tappat bort glassen

a moment ago I had just–nyss hade jag precis
I had just lost my ice cream–hade jag precis tappat bort glassen

...

a moment ago I had just lost my ice cream–
nyss hade jag precis tappat bort glassen

UPPSALA
UNIVERSITET

Phrase extraction exercise

Exercise on phrase pair extraction!

UPPSALA
UNIVERSITET

Scoring phrase translations

- Phrase pair extraction: collect all phrase pairs from the data
- Phrase pair scoring: assign probabilities to phrase translations

UPPSALA
UNIVERSITET

Scoring phrase translations

- Phrase pair extraction: collect all phrase pairs from the data
- Phrase pair scoring: assign probabilities to phrase translations
- Score by relative frequency:

$$\phi(\bar{t}|\bar{s}) = \frac{count(\bar{s}, \bar{t})}{\sum_{\bar{t}_i} count(\bar{s}, \bar{t}_i)}$$

UPPSALA
UNIVERSITET

Scoring phrase translations

- Phrase pair extraction: collect all phrase pairs from the data
- Phrase pair scoring: assign probabilities to phrase translations
- Score by relative frequency:

$$\phi(\bar{t}|\bar{s}) = \frac{count(\bar{s}, \bar{t})}{\sum_{\bar{t}_i} count(\bar{s}, \bar{t}_i)}$$

- Potentially improve scoring by smoothing

UPPSALA
UNIVERSITET

Size of the phrase table

- Phrase translation table typically much bigger than corpus
- Limit the length of phrase pairs (often to 7 tokens)

UPPSALA
UNIVERSITET

Size of the phrase table

- Phrase translation table typically much bigger than corpus
- Limit the length of phrase pairs (often to 7 tokens)
- Too big to store in memory?
  - Store on disk
  - Use smart data structures

UPPSALA
UNIVERSITET

Size of the phrase table

- Phrase translation table typically much bigger than corpus
- Limit the length of phrase pairs (often to 7 tokens)
- Too big to store in memory?
  - Store on disk
  - Use smart data structures
- Prune phrase table – i.e., remove non-useful phrase pairs
  - Limit translation options for each phrase (often to 20–30)
  - Prune table based on statistics, such as χ^2

UPPSALA
UNIVERSITET

Size of the phrase table

- Phrase translation table typically much bigger than corpus
- Limit the length of phrase pairs (often to 7 tokens)
- Too big to store in memory?
  - Store on disk
  - Use smart data structures
- Prune phrase table – i.e., remove non-useful phrase pairs
  - Limit translation options for each phrase (often to 20–30)
  - Prune table based on statistics, such as χ^2
- Filter phrase table towards a test set

UPPSALA
UNIVERSITET

Weighted models

- Described model consists of three sub-models:
  - Phrase translation models $\phi(\overline{s}|\overline{t})$
  - Reordering model d
  - Language model $p_{LM}(t)$

$$t_{best} = \arg\max_{t} \prod_{i=1}^{I} \phi(\overline{s}_i|\overline{t}_i) d(\text{start}_i - \text{end}_{i-1} - 1) \prod_{i=1}^{|t|} p_{LM}(t_i|t_{i-(n-1)} \dots t_{i-1})$$

UPPSALA
UNIVERSITET

Weighted models

- Described model consists of three sub-models:
  - Phrase translation models $\phi(\overline{s}|\overline{t})$
  - Reordering model d
  - Language model $p_{LM}(t)$

$$t_{best} = \arg\max_{t} \prod_{i=1}^{I} \phi(\overline{s}_i|\overline{t}_i) d(\text{start}_i - \text{end}_{i-1} - 1) \prod_{i=1}^{|t|} p_{LM}(t_i|t_{i-(n-1)} \dots t_{i-1})$$

- Some sub-models may be more important than others

UPPSALA
UNIVERSITET

Weighted models

- Described model consists of three sub-models:
  - Phrase translation models $\phi(\overline{s}|\overline{t})$
  - Reordering model d
  - Language model $p_{LM}(t)$

$$t_{best} = \arg\max_{t} \prod_{i=1}^{I} \phi(\overline{s}_i|\overline{t}_i) d(\text{start}_i - \text{end}_{i-1} - 1) \prod_{i=1}^{|t|} p_{LM}(t_i|t_{i-(n-1)} \dots t_{i-1})$$

- Some sub-models may be more important than others
- Add weights $\lambda_\phi, \lambda_d, \lambda_{LM}$

UPPSALA
UNIVERSITET

Weighted models

- Described model consists of three sub-models:
  - Phrase translation models $\phi(\overline{s}|\overline{t})$
  - Reordering model d
  - Language model $p_{LM}(t)$

$$t_{best} = \arg\max_{t} \prod_{i=1}^{I} \phi(\overline{s}_i|\overline{t}_i)^{\lambda_\phi} d(\text{start}_i - \text{end}_{i-1} - 1)^{\lambda_d} \prod_{i=1}^{|t|} p_{LM}(t_i|t_{i-(n-1)} \dots t_{i-1})^{\lambda_{LM}}$$

- Some sub-models may be more important than others
- Add weights $\lambda_\phi, \lambda_d, \lambda_{LM}$

UPPSALA
UNIVERSITET

Log-linear models

- Such a weighted model is a log-linear model:

$$p(x) = \exp \sum_{i=1}^{n} \lambda_i h_i(x)$$

- Our feature functions:
  - three feature functions $n = 3$
  - random variable $x = (s, t, \text{start}, \text{end})$
  - feature function $h_1 = \log \phi$
  - feature function $h_2 = \log d$
  - feature function $h_3 = \log p_{LM}$

UPPSALA
UNIVERSITET

Weighted model as a log-linear model

$$p(t, a|s) = \exp(\lambda_\phi \sum_{i=1}^{I} \log \phi(\overline{s}_i|\overline{t}_i) +$$
$$\lambda_d \sum_{i=1}^{I} \log d(\text{start}_i - \text{end}_{i-1} - 1) +$$
$$\lambda_{LM} \sum_{i=1}^{|t|} \log p_{LM}(t_i|t_{i-(n-1)} \dots t_{i-1}))$$

UPPSALA
UNIVERSITET

More feature functions

$$t^* = \arg\max_t \sum_i \lambda_i h_i(s, t)$$

- Easy and useful to add more feature functions

UPPSALA
UNIVERSITET

More feature functions

$$t^* = \arg\max_t \sum_i \lambda_i h_i(s,t)$$

- Easy and useful to add more feature functions
  - Bidirectional alignment probabilities $\phi(\overline{s}|\overline{t})$ and $\phi(\overline{t}|\overline{s})$

UPPSALA
UNIVERSITET

More feature functions

$$t^* = \arg\max_t \sum_i \lambda_i h_i(s,t)$$

- Easy and useful to add more feature functions
  - Bidirectional alignment probabilities $\phi(\bar{s}|\bar{t})$ and $\phi(\bar{t}|\bar{s})$
  - Lexical weighting of phrase pairs:

UPPSALA
UNIVERSITET

More feature functions

$$t^* = \arg\max_t \sum_i \lambda_i h_i(s,t)$$

- Easy and useful to add more feature functions
  - Bidirectional alignment probabilities $\phi(\overline{s}|\overline{t})$ and $\phi(\overline{t}|\overline{s})$
  - Lexical weighting of phrase pairs:

$$\text{lex}(\overline{t}|\overline{s},a) = \prod_{i=1}^{\text{length}(\overline{t})} \frac{1}{|\{j|(i,j)\in a\}|} \sum_{\forall(i,j)\in a} w(t_i|s_j)$$

UPPSALA
UNIVERSITET

More feature functions

$$t^* = \arg\max_t \sum_i \lambda_i h_i(s,t)$$

- Easy and useful to add more feature functions
  - Bidirectional alignment probabilities $\phi(\overline{s}|\overline{t})$ and $\phi(\overline{t}|\overline{s})$
  - Lexical weighting of phrase pairs:

$$\text{lex}(\overline{t}|\overline{s},a) = \prod_{i=1}^{\text{length}(\overline{t})} \frac{1}{|\{j|(i,j)\in a\}|} \sum_{\forall(i,j)\in a} w(t_i|s_j)$$

  - Useful since rare phrase pairs have unreliable probability estimates

UPPSALA
UNIVERSITET

More feature functions 2

- Language model has a bias towards short translations
  - word count: $\mathrm{wc}(t) = \log |t|^{\omega}$

UPPSALA
UNIVERSITET

More feature functions 2

- Language model has a bias towards short translations
  - word count: $\mathrm{wc}(t) = \log |t|^{\omega}$
- We may prefer finer or coarser segmentations
  - phrase count: $\mathrm{pc}(t) = \log |I|^{\rho}$

UPPSALA
UNIVERSITET

More feature functions 2

- Language model has a bias towards short translations
  - word count: $\mathrm{wc}(t) = \log |t|^{\omega}$
- We may prefer finer or coarser segmentations
  - phrase count: $\mathrm{pc}(t) = \log |I|^{\rho}$
- Multiple language models

UPPSALA
UNIVERSITET

More feature functions 2

- Language model has a bias towards short translations
  - word count: $\mathrm{wc}(t) = \log |t|^{\omega}$
- We may prefer finer or coarser segmentations
  - phrase count: $\mathrm{pc}(t) = \log |I|^{\rho}$
- Multiple language models
- Other knowledge sources

UPPSALA
UNIVERSITET

More feature functions 2

- Language model has a bias towards short translations
  - word count: $\mathrm{wc}(t) = \log |t|^{\omega}$
- We may prefer finer or coarser segmentations
  - phrase count: $\mathrm{pc}(t) = \log |I|^{\rho}$
- Multiple language models
- Other knowledge sources
- Lexicalized reordering models

UPPSALA
UNIVERSITET

Translation model learning

- Phrase-based model, but trained using word alignments

UPPSALA
UNIVERSITET

Translation model learning

- Phrase-based model, but trained using word alignments
- Can we not train a phrase-based model directly?

UPPSALA
UNIVERSITET

Translation model learning

- Phrase-based model, but trained using word alignments
- Can we not train a phrase-based model directly?
- Yes, with EM!
  - But, difficult, and often overfits

UPPSALA
UNIVERSITET

Translation model learning

- Phrase-based model, but trained using word alignments
- Can we not train a phrase-based model directly?
- Yes, with EM!
  - But, difficult, and often overfits
- The pipeline I have shown is the state-of-the-art

UPPSALA
UNIVERSITET

Moses

- SMT toolkit
- Free, open source
- Implements several models:
  - Phrase-based
  - Hierarchical
  - Syntax-based
- Decoding (next lecture)
- Training pipeline
  - Training translation models
  - Training language models
  - Optimizing feature weights

UPPSALA
UNIVERSITET

Moses training pipeline

1. Prepare data
2. Run GIZA to create one-way alignments
3. Symmetrize alignment
4. Calculate lexical translation probabilities
5. Extract phrases
6. Score phrases
7. Train reordering model
8. (Train generation model)
9. Create configuration file

UPPSALA
UNIVERSITET

Moses training pipeline

- Preprocess data (tokenization, casing, et.c.)

1. Prepare data
2. Run GIZA to create one-way alignments
3. Symmetrize alignment
4. Calculate lexical translation probabilities
5. Extract phrases
6. Score phrases
7. Train reordering model
8. (Train generation model)
9. Create configuration file

- Tune the system

UPPSALA
UNIVERSITET

Tuning – Optimizing feature weights

- How do we learn the best weights λ_i?

UPPSALA
UNIVERSITET

Tuning – Optimizing feature weights

- How do we learn the best weights λ_i?
- Optimize the weights on a small corpus

UPPSALA
UNIVERSITET

Tuning – Optimizing feature weights

- How do we learn the best weights λ_i?
- Optimize the weights on a small corpus
- Called **Tuning**

UPPSALA
UNIVERSITET

Coming up

- Next week:
  - Monday September 24: deadline for wishing for project topics
  - Monday: Lecture on decoding
  - Tuesday: Assignment 4: PBSMT and Moses
  - Wednesday: Lab 1 (not 5LN718)
  - Thursday: Lecutre cancelled
- Later:
  - Assignment 5: decoding and document-level MT
  - Start working on your projects
  - Lectures and assignment on NMT
  - Guest lecture