# Statistical Machine Translation

LECTURE – 7

## PBMT & DECODING

APRIL 20, 2010

## Brief Outline

- Possible Directions to PBMT
- Decoder
- Decoding Algorithms
- Hypothesis Expansion
- Hypothesis Reduction
- Search Space Reduction
- Stack-Based Implementation
- Pruning Based on Future Cost

## Preamble

Phrase-based methods are more intuitive than word-based.

The overall task however is not so straightforward.

Selection of phrases, storage retrieval etc. are obvious difficulties

A word may not have a single alignment (polysemy, case).

Maintaining Subject –Verb concordance or Anaphora –resolution often difficult.

Consequently an input sentence may lead to a number of translations.

All these problems make Phrase-Based Translations very interesting to study.

## Possible directions

What about Linguistically motivated segment boundaries?

This appears to be the most logical thing to do.

Lots of research is going on

Why go by Word Alignment ?

Can there be phrase Alignment directly?

This leads to Joint Phrase Alignment Model

There are different Phrase translations.

How can they be combined efficiently?

This leads to developing Decoders.

## *Linguistically Motivated Segment Boundaries.*

## *Linguistically Motivated Segment Boundaries.*

Consider translating:

The tall boy who came here riding a bike is good in maths.

What are the linguistic boundaries?

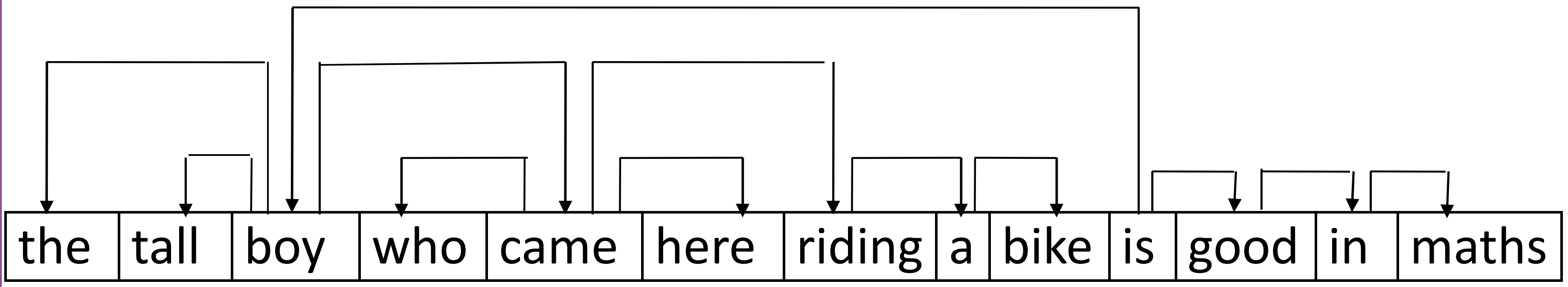
We show two of the most important parsing:

- Phrase structured grammar
- Dependency Grammar

# Phrase-Structured Parse

```
(S  (Subject
        (NP   (Art The)   (Adj   tall) (N boy)
              (AMOD ( WP  who)  (V came)
                          (Vmod  here)
                           (Vmod  (VBG riding)
                                      (NP (Art  a) (N bike))))))
    (Predicate
              (V is)
               (Adj  good)
                (PP  (Prep  in )
                       (N maths)))))
```

## Dependency Parse



## Discussion

It is obvious that they give natural sub-sentences

The advantage of Dependency parse is that they are Good in projecting long distance dependencies.

But the problem of Phrase selection is still there .

How about storing the skeleton – and add additional Features as branches of the tree?

Some people are using CFG.

Still very much a research topic.

## Discussion

Let us see how long distance dependencies affect translation:

The girl is good >> Il ragazza è buona

The girl who comes here everyday is good >>
La ragazza che viene qui ogni giorno è ***buono***

The adjective is not following the noun because of the long sdistance dependencey

## Discussion

This can happen with different languages in different forms:

**English:**

The **boy** who comes here daily says that **his father** is ill.

The **boy** who comes here daily says that **his mother** is ill.

The **girl** who comes here daily says that **her father** is ill.

The **girl** who comes here daily says that **her mother** is ill.

## Discussion

**Italian**

The boy who comes here daily says that his father is ill

>>

Il ragazzo che viene qui ogni giorno dice che **suo** padre è **malato**

The boy who comes here daily says that his mother is ill

>>

Il ragazzo che viene qui ogni giorno dice che **sua** madre è **malata**

In fact google translates even if we write

The boy .... her father

## Discussion

**Hindi**

(1) Hindi will behave same as Italian w.r.t these Sentences.
(2) But Hindi has other problem – Verb follows the number and gender of the actor.

The boy who comes here daily says that his father sings well
>>
Il ragazzo che viene qui ogni giorno dice che suo padre **canta bene**

**(the same for mother)**

But in Hindi we shall have different verb forms.

## Discussion

### Bengali

In bengali gender , number have no effect on Verb and Adjective (barring a few exceptions for Adjectives)

Does it make it easy ??

## *Joint Model for Phrase Alignment*

# Joint Model

It is a generative model – where a pair of SL and TL Phrases are generated using some concept.

- First a set of concepts **c** = $c_1 c_2 .. c_n$ is generated.
- Each concept $c_i$ then generates a phrase pair $(\overline{f}_i, \overline{e}_i)$ with probability $p(\overline{f}_i, \overline{e}_i \mid c_i)$
- The number of phrase pairs are same for TL and SL sentences.
- Hence this is called a joint model

## Joint Model

- Each phrase is placed at some specific position, say *pos*

- *pos* takes care of the language modeling for TL.

- Probability of **e** and **f** can be written as:

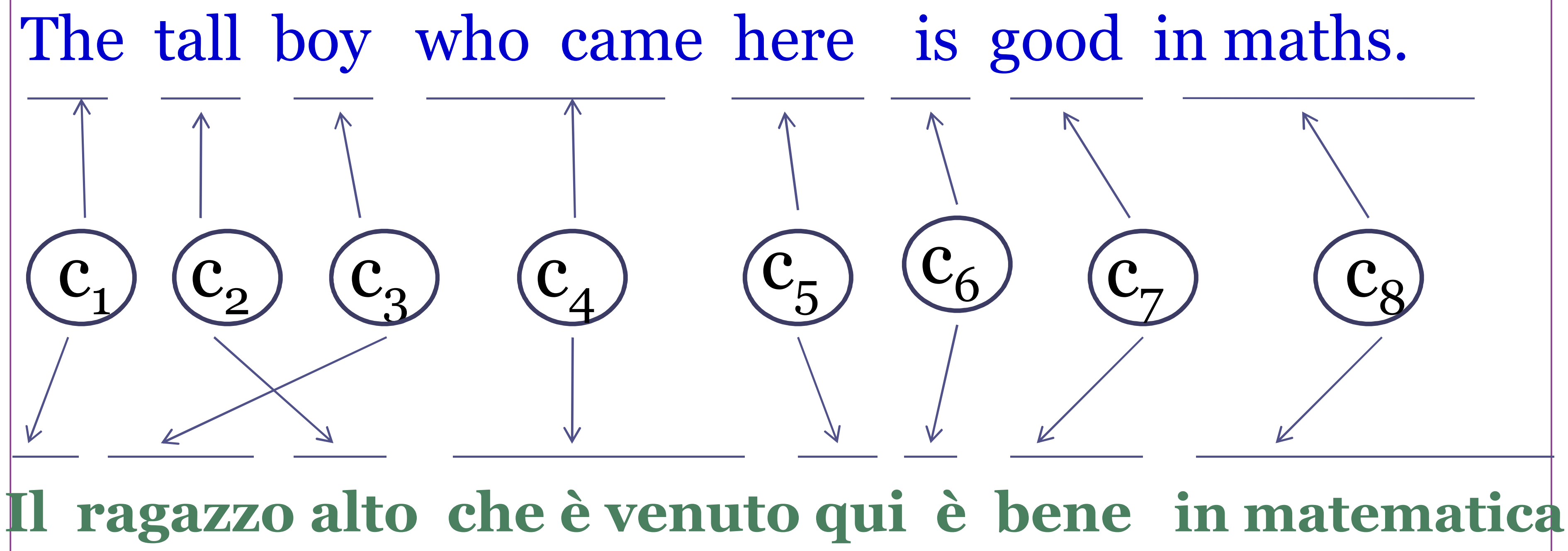
$$\mathrm{p}(\mathbf{e}, \mathbf{f}) = \prod_{i=1}^{n} p(\overline{e_i}, \overline{f_i} \mid c_i)\, d(pos(e_i) \mid pos(e_{i-1}))$$

where d takes care of the reordering/distortion.

- The model now can be trained on a parallel corpus.

## Joint Model

Example:


The tall boy who came here is good in maths.
$c_1$
$c_2$
$c_3$
$c_4$
$c_5$
$c_6$
$c_7$
$c_8$
Il ragazzo alto che è venuto qui è bene in matematica

## Joint Model

Typically phrase translation probabilities are used to estimate the concept driven translation Probabilities – i.e.

$$p(\bar{e}, \bar{f} \mid c) = t(\bar{e} \mid \bar{f})$$

But the major decision is what is the value of n.

As this value has huge effect on the alignment space.

## Complexity of the Alignment Space

If we assume only to contiguous spaces –

1) a phrase can start from any position, and end anywhere later.

[

Hence there can be $O(n^2)$ phrases

2) The same is for both TL and SL

3) Thus there are $O(n^4)$ possible pairs - or $O(n^3)$ if we follow the phrase sequence of one language.

## Complexity of the Alignment Space

Another question is in how many ways a pair of sentences can be generated of length n each.

4) Some concepts can be merged.
Hence we need to consider m concepts out of n.
Thus the number of possibilities is $O(m\string^n^3)$.

So complexity wise it is becoming much worse than alignment based on word alignment points.

EM algorithm still can be used.
But with a lot of heuristics to keep the size under control

## Training the Model

Basic Steps

Init: Give uniform probability to all phrase pairs

E-step: To estimate the likelihood of all possible phrase alignments for all sentence pairs. Weight the count with probability of the alignment in which they occur

M-Step: Given the counts, the probability estimates are updated with MLE.

Greedy searches used to keep check on space and time

## Training the Model

Typically, a greedy search works as follows:

1. Create initial alignment from the highest probability $t(\overline{e} \mid \overline{f})$ from the Phrase Translation Table.

2. Then Hill-climbing is applied by:
   a) breaking and merging concepts.
   b) moving words across concepts.
   c) swapping words between concepts.

To collect counts, for high probability alignments,
Sampling is done: - along the hill-climbing path
- in the neighbourhood of the highest probability alignment

## *Decoder*

## What is decoding

Mathematical models (word based, phrase based)
Assign probabilistic scores to different possible Translations.

The task of Decoder is to find the best sentence.

But the problem of exhaustive search is NP-complete. [Kevin Knight]

Heuristic search techniques are employed.

Thus though the best solution is not always guaranteed, We expect one very close to that.

## Why Search

Consider: er geht ja nicht nach hause (Gr)

Top 3 translation options of the words (Koehn)

| SL words | Tran 1 | Tran 2 | Tran 3 |
|---|---|---|---|
| Er | he | it | , it |
| Geht | is | are | goes |
| Ja | yes | is | of course |
| Nicht | not | do not | does not |
| Nach | after | to | according to |
| Hause | house | home | chamber |

The actual translation: He does not go home

## Why Search

Therefore a monotone translation with the most probable meanings do not work.

Consider 2-word phrases ( 3 most probables)

| 2-gram phrases | Trans -1 | Trans-2 | Trans-3 |
|---|---|---|---|
| Er geht | it is | he will be | it goes |
| Geht ja | in | are | is after all |
| Ja nicht | not | is not | does not |
| Nicht nach | to | following | not after |
| Nach hause | home | under house | return home |

## Why Search

Similarly one can think of phrases of bigger size and their probabilities.

The message is: there are plenty of choices.

A system trained on the Europarl corpus may have as many as 2727 options for this sentence (Philip Koehn).

The  solution space grows at a very fast rate as the Sentence length increase.

Consequently heuristic searching is needed.

## *Decoding Algorithms*

# Hypothesis Expansion

Hypothesis is built Left to Right, one phrase at a time.

Each partial translation made is called a Hypothesis

The process start with Null Hypothesis.

Expansion:  Picking a translation option, and construct a new hypothesis.

Hence the search space is progressive.

Also, the above scheme gives the tree-like structure.

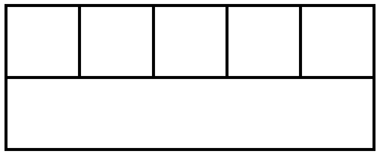
## Hypothesis Expansion

Illustration:

Hypotheses: A record like structure

Contains information about the current translation and the progress made so far.

Empty hypothesis:



The algorithm starts with Empty Hypothesis

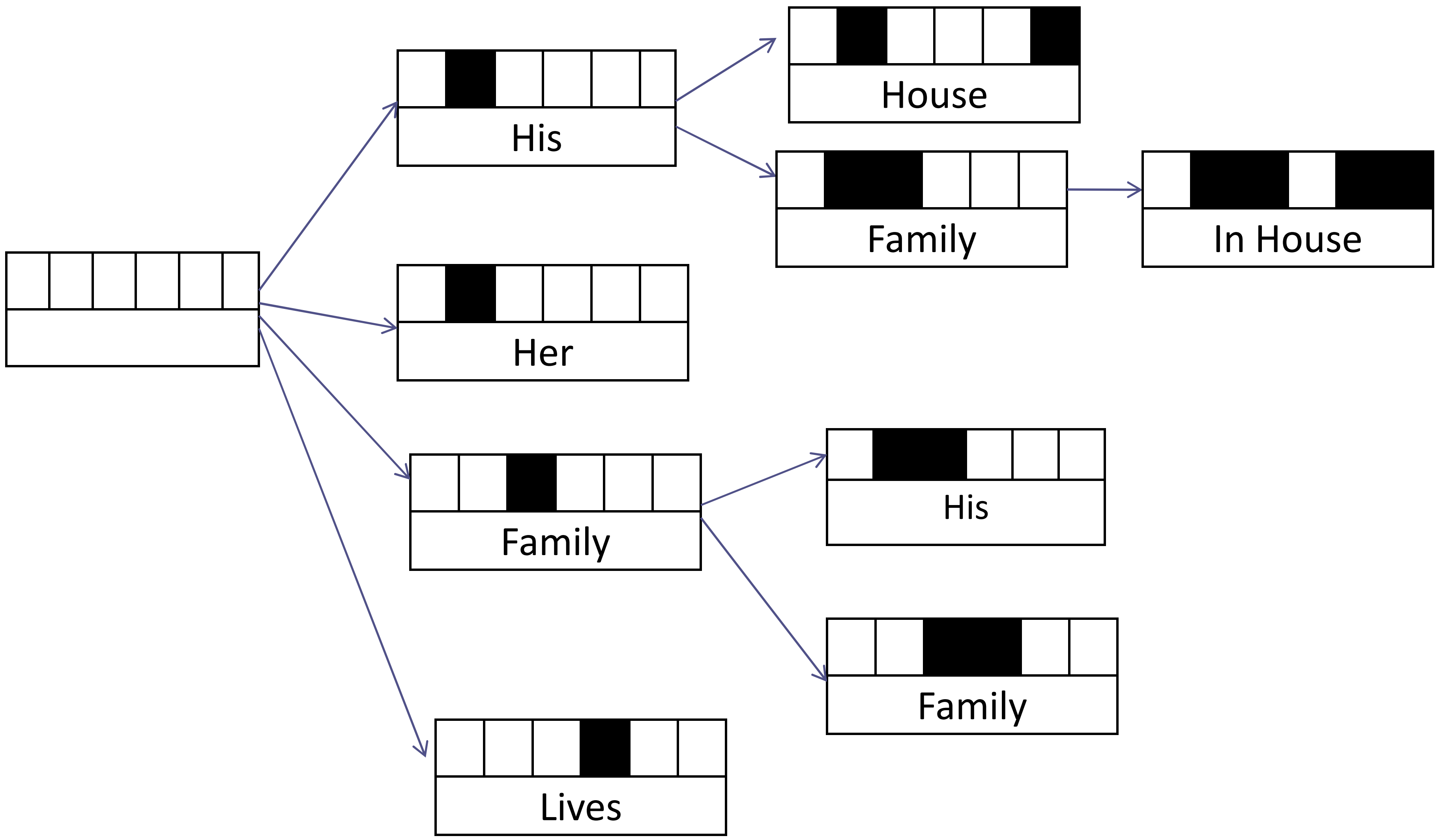
## Illustration

Consider: la sua famiglia resta nella casa

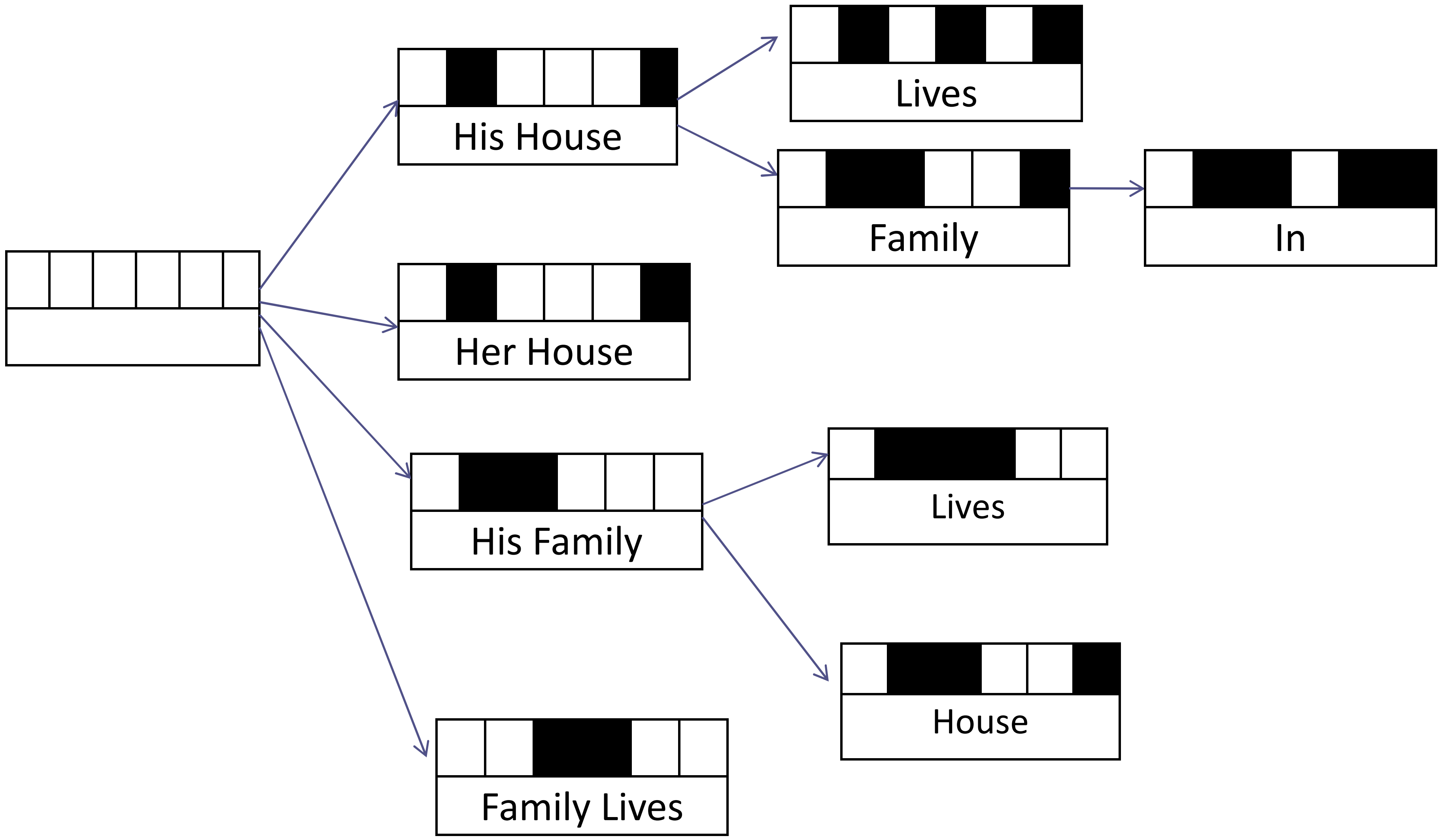
| | |
|---|---|
| Sua | - his her here its |
| Famiglia | - family household home people stock |
| Resta | - remains lives |
| Casa | - house home flat family household |

One can pick a word in many ways:
which leads to exponential growth.

# Hypothesis Expansion



# Hypothesis Expansion



## Hypothesis Expansion

Note: Each time one phrase translation is added the score is updated using the three factors:

- Translation Model: Collect the Phrase Translation $\phi$ Probability from the Phrase Translation Table.

- Distortion model: Get the distortion probability $d$ from the start position of the current phrase and end position of the previous phrase translated.

- Language Model: The bi-gram, tri-gram probabilities are used to score the translation generated so far.

Thus at each step partial scores are maintained.

## Hypothesis Expansion

The expansion comes to an end point when there is no More phrase/word to expand.

Thus there will be a large number of endpoints.

The number depends upon : Length of the sentence
No. of options

Different endpoints will have different probabilities.

The probability will depend upon : Language Model
Translation Probs.

Typically, the highest prob. will be chosen.

## Computational Complexity

The best solution can be obtained after building the whole search Space

This is exponential in the size of the input sentence.

This problem is NP-Complete (K. Knight).

Two step reduction of complexity:

- Hypothesis reduction
- Pruning

## Hypothesis Reduction

The same hypothesis can be reached through different Paths: E.g. His Family (In the above example)

Let $x_f >> x_e$ $y_f >> y_e$ and $z_f >> z_e$

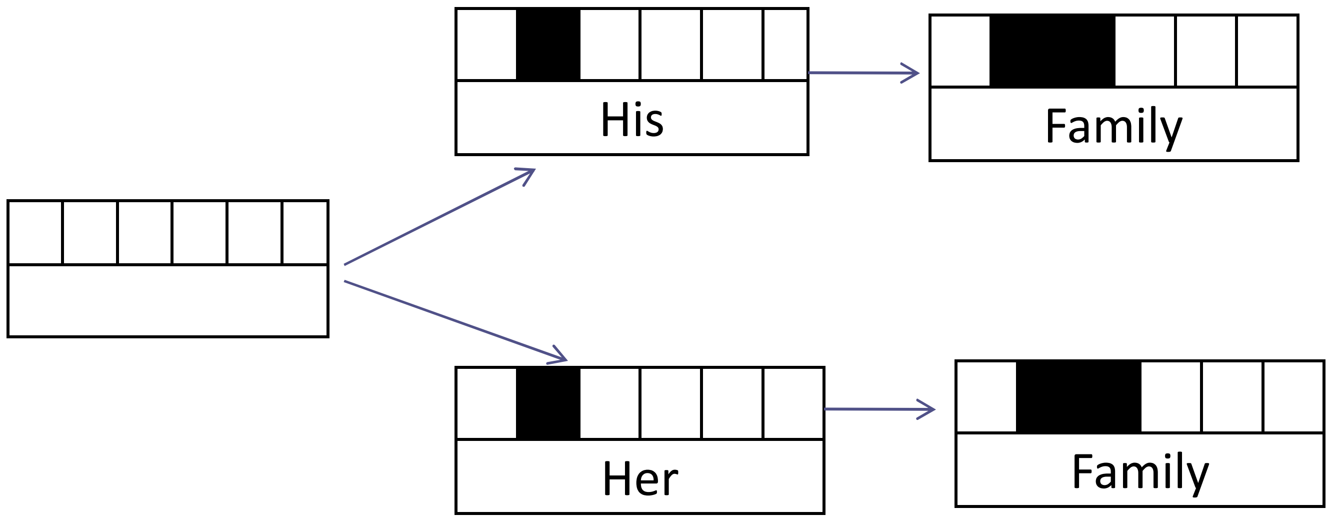
Then the translation decision $[x_f\ y_f\ z_f] >> [x_e\ y_e\ z_e]$
Can be reached in 6 + 3*2 + 1 = 13 possible ways.

Do we need to preserve all?

We can drop the worst hypothesis. It will never be Part of the solution.

## Hypothesis Reduction

Also consider the following situation:



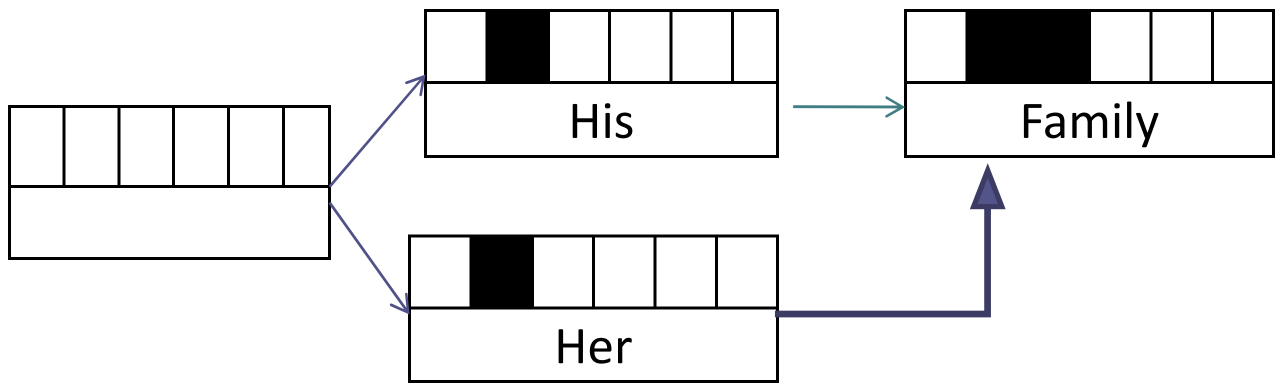
The last production is same for both the path.

Hence both cannot be part of the solution.

Here too we can retain only the better one!!

# Hypothesis Reduction

Typically instead of complete deleting a Link is maintained.



This helps in getting second/third/ … best Solutions.

## Hypothesis Reduction

-Translation model: has no effect on reduction

- Reordering model.

  Requires the start and end positions of the current phrase and last phrase.

  Nodes where they differ cannot be recombined

-Language model.

  Nodes with different histories cannot be recombined.

Note: It gives a smarter structure.

Does not reduce exponential complexity.

## *Search Space Reduction*

## The Philosophy

Instead growing the full search space, can we  prune some parts?

If a branch does not appear to be probable it  can be eliminated. ( Of course has its own risk.)

There will be a search error – if we happen to prune A branch that might yield the final solution.

But exhaustive search is too expensive.

Depends upon the heuristic search technique  applied

## Search Space Reduction

Various heuristic search techniques exist:

- Depth First ( exhaustive )
- Breadth First (exhaustive)
- Best First (expands the most promising node chosen according to a specified rule.)
- Hill Climbing (starts with an initial translation. It is recursively improved by changing it in steps)

-Branch& Bound (a branch is stopped from expansion if its max probability is certain to be less than one that is already found )

## Search Space Reduction

However the one that is mostly used is Beam Search.

It is an optimization of best-first search

Best-first search works in the following way:

- orders all partial solutions (states) according to some heuristic
- which attempts to predict how close a partial solution is to a complete solution (goal state).
- Uses already travelled cost and an estimate of the remaining cost.
- Typically implemented on a Priority Queue

## Search Space Reduction

Beam Search is implemented on stacks.

In Beam search, only a predetermined number of best partial solutions are kept as candidates.

The beam width can either be fixed or variable.

- In a fixed beam width, a maximum number of successor states is kept.
- In a variable beam width, a threshold is set around the current best state.

All states that fall outside this threshold are discarded.

## Stack-Based implementation

n+1 stacks are used, where sentence length is n words

The stacks are numbered 0,1, …, n

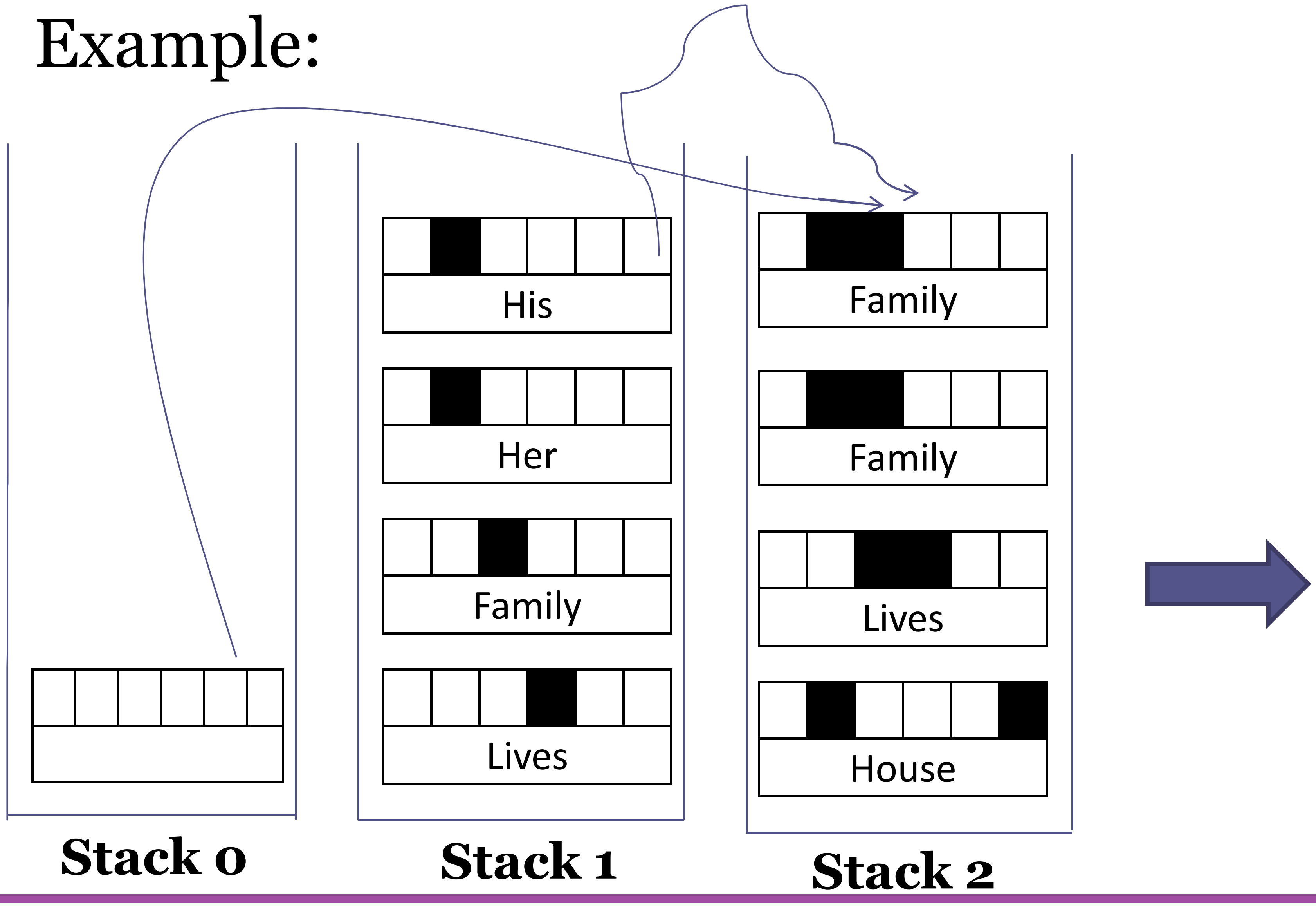
The *i*th stack contains hypotheses where *i* original Words have been translated

The progress starts with stack 0, which contains null Hypothesis.

If a stack is full, least probable hypothesis is pruned.

# Stack-Based implementation

Example:



## Stack-Based implementation

The Algorithm

```
Set stack 0 = null Hypothesis
For i = 0 to n – 1
   For j = 0 to $n_i$      // $n_i$ is the no. of hypotheses
                              in stack i
      For all translations options
         Try  option,   if applicable
          Apply recombination, if applicable
          Place  hypothesis  in appropriate stack
          If there is a stack overflow -  prune it.
```

**As each stack has limited capacity only best few retained**

## Stack-Based implementation

Note that the algorithm maintains not only The best hypothesis, but a few more.

Thus the solution is achieved along a line.

Hence this is called the Beam Search.

## Stack-Based implementation

Complexity

The stack size is directly related to decoding speed
Suppose the max no. of hypothesis in a stack = k.

Assume all stacks are filled & all translation options are available.

The number of hypothesis expansions is: $k * m * n$,
where m is the maximum translation options

If the no. of translation options is linear with sentence length then complexity = $O(kn^2)$

But if we choose all phrases can go exponential !!

## Stack-Based implementation

Shortcoming

Since stack sizes are limited:

- May prune some very good translation
- May retain  some very poor score.

Solution?

- threshold pruning

## Threshold Pruning

Use variable size stack.

- If there are many good hypothesis try to retain as many.
- If there are very few good hypothesis prune all The bad ones.

To effect this variable size stacks are used.

Pruning criterion: score < α. Max score (α is a pre-decided constant).

This is called: Threshold Pruning.

## Further improvement.

Note that still the pruning method is ad hoc. Some part may be easy to translate, some part is difficult. E.g.

It is about joining forces economically, and it has to be recognized unassumingly.

In any language the green parts are easy to translate, whereas red part is more difficult.

The stack based algorithm prunes a hypothesis Without taking care of this!!!!

## *Pruning based on Future cost.*

## Estimation of Future cost.

It implies *estimating* the difficulty of translating the Remaining part of the input sentence.

We need to consider the 3 models:
--Translation model
-- Reorder model
-- Language Model

How do they affect the computation??

## Estimation of Future cost.

Reordering Model: It is difficult to gauge what reorder will be needed when the complete translation is done.

Hence it is not used in estimating future cost.

Translation Model: The Phrase Translation Table can be used.

Language Model: Cannot use n-grams in an effective way, as words are not translated monotonically.

Hence unigram, bigrams are typically used.

## Estimation of Future cost.

The aim is to compute the cheapest cost.

A dynamic programming Based Algorithm Is used.

The cost is computed for various spans of Words along the sentence.

## Estimation of Future cost.

The algorithm:

```
For gap = 1 to n-1
  for start = 1 to n – gap
      end = start + gap
      cost(start, end) ← ∞
      cost(start, end) ← translation cost (if exists)
      For k = start to end – 1
        if
          cost (start,k) + cost (k+1,end) < cost(start,end)
            update cost (start,end)
```

This Future Cost can now be used to prune a more efficiently.

## Other Algorithms

Other search algorithms have also been tried:

A* - puts admissibility constraint on heuristic.

Greedy Hill Climbing:

- Start with an initial word translation
- Iteratively improve it by incorporating language model/ reorder model (by merging, splitting/swapping)
- Stop when no further improvements possible.

Thank You.