Statistical Machine Translation

LECTURE – 6

PHRASE-BASED MODELS

APRIL 19, 2010

Brief Outline

- Introduction
- Mathematical Definition
- Phrase Translation Table
- Consistency
- Phrase Extraction
- Translation Probabilities
- Reordering Models

Introduction

Question: Is word-model the right one?

1. Often group of words taken together are translated in a different way:

   e.g. My daughter is *apple of my eyes*.

   >> mia figlia è mela dei miei occhi

   He was beaten *black and blue*.

   >> di essere stato picchiato nero e blu

2. A polysemic word e.g. *fan, bat, bank* may be better Translated , if translated in a context.

All these translations breaks down in the Word level.

Introduction

Solution is ***phrase based model***

Note: A *phrase* here is NOT the way a parser defines based on syntactic role: NP, VP. Rather it is a *multi-word unit* as a sequence of words.

When such set of consecutive words occur frequently and statistics about their translations are calculated, it may lead to better translation.

Still there can be problems - as we see in the follwoing examples

Introduction

The fan is running	>>	Il ventilatore è in funzione
The fan is on	>>	Il ventilatore è in
The fan is on table	>>	Il ventilatore è in tavola
The fan is on the table	>>	La ventola è sul tavolo
The fan is on the desk	>>	Il ventilatore è sulla sedia
The fan is on test	>>	La ventola è in prova
The fan is waving a flag	>>	la ventola è sventolare una bandiera

The fan is	>>	prashansak hai
The fan is running	>>	pankhaa chal rahaa hai
The fan is on	>>	prashansak par hai
The fan is on the table	>>	prashansak mej par hai
The fan is on the chair	>>	prashansak kursii par hai
The fan is on test	>>	prashansak parikshaa par hai
The fan is waiving a flag	>>	prashansak ek jhandaa lahraate hai

Mathematical Definition

What are the problems?

- Semantics is not always preserved.
- Translation changed with introduction of article
- Preposition changes with article.
- Identification of phrase boundaries is difficult

Still phrase –based makes more sense than pure word-based.

Mathematical Definition

Here we improve upon the Bayesian Model:

$$\mathbf{e}_{Best} = \arg\max_{\mathbf{e}} \; p(\mathbf{e} | \mathbf{f}) = \arg\max_{\mathbf{e}} \; p(\mathbf{f} | \mathbf{e}) p(\mathbf{e})$$

For phrase-based model, the term $p(\mathbf{f} \mid \mathbf{e})$ is *further broken down:*

Let $\mathbf{f}$ be split into I phrases: $\overline{f}_1, \overline{f}_2, \ldots, \overline{f}_I$

Not modeled explicitly – so all segments are Equally likely

Mathematical Definition

Each of the foreign phrases $\overline{f}_1 , \overline{f}_2 , \ldots , \overline{f}_I$
is translated into corresponding **e** phrase $\bar{e}_i$

Hence we get:

$$p(\mathbf{f} \mid \mathbf{e}) = \prod_{i=1}^{I} \phi(\overline{f}_i \mid \overline{e}_i) * pd(d_i)$$

- *pd is the relative distortion probability*
- d_i is an relative distortion of the i^{th} phrase.

This is because the phrases need not be in the same order in **f** as in **e**.

Mathematical Definition

Example:

he has gone into the room

woh kamre mein chalaa gayaa

he peeped into the room

woh kamre mein jhankaa

There should be a way to handle this reordering

Often *pd* is taken as a decaying function : $\alpha^{|x|}$, s..t. $\alpha \in (0, 1)$.

Although it is possible to study their distribution from a corpus.

However, it is not easy.

Mathematical Definition

A simple measure for d_i is $start_i - end_{i-1} - 1$

- Counts no. of **f** words skipped when taken out of sequence.
- Computed on the basis of **e** phrase and **f** words.

Note: wihtout any reordering, $start_i = end_{i-1} + 1$

Ex: natuerlich hat john spass am spiel (DE)

of course john has fun with the game

d1 = 0 d2 = 1 d3 = -2 d4 = 1 d5 = 0

Calculation of Di

Phrase No.	Translates	Skips	D_i
1	1	Start at the beginning	0
2	3	Phrase 2	1
3	2	Moves back 3 & 2	- 2
4	4-5	Phrase 3	1
5	6	Start normally	0

Ex: Try computing the *di*'s for the following

the tall boy riding a bicycle is singing loud songs

Il ragazzo alto in sella a una bicicletta è cantare le canzoni ad alto volume

Phrase Translation table

Gives information about how the phrases are translated.

The phrase translations are extracted from word alignment

Can be accomplished in many ways.

We shall use an algorithm based on *consistency*.

The following algorithm is designed on the top of word alignment in a parallel corpus.

It extracts phrase pairs that are *consistent* with the word alignment

Q: Can we develop a good algorithm for Phrase alignment?

What is Consistency?

Definition: A phrase pair ($\bar{f}, \bar{e}$) is said to be consistent w.r.t an alignment A, if all words $f_1 .. f_n$ in $\bar{f}$ that have alignment points $e_1 .. e_n$ in $\bar{e}$ and vice versa:

($\bar{f}, \bar{e}$) is consistent with A $\Leftrightarrow$

$$\forall e_i \in \bar{e} \mid (e_i, f_j) \in A \Rightarrow f_j \in \bar{f}$$

$$\wedge \quad \forall f_j \in \bar{f} \mid (e_i, f_j) \in A \Rightarrow e_i \in \bar{e}$$

$$\wedge \quad \exists e_i \in \bar{e}, f_j \in \bar{f} \mid (e_i, f_j) \in A$$

What is Consistency?

Example:

The boy is singing loud songs>>

Il ragazzo sta cantando le canzoni ad alto volume

The boy >> Il ragazzo

Is singing >> sta cantando

Loud >> forte

Songs >> canzoni

Loud songs >> canzoni ad alto volume

Singing loud songs >> cantando canzoni ad alto volume

loud songs >> canzoni ad alto volume Is inconsistent??

Phrase Extraction

We need an algorithm to extract consistent phrases:

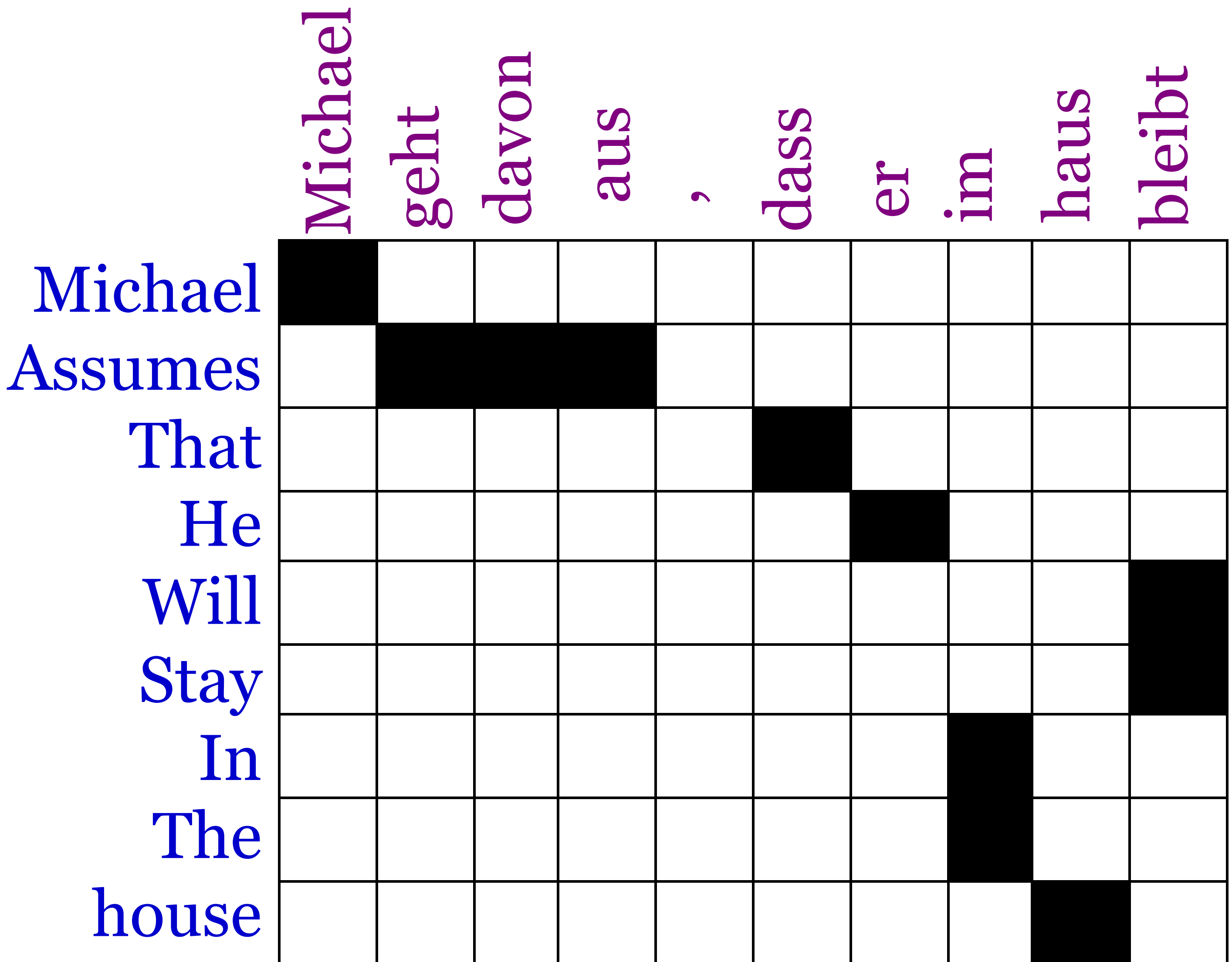
- It should go over all possible sub-sentences
- For each it extracts the minimal foreign phrase
- Matching is done by identifying all alignment points for $\bar{e}$, and then identifying the shortest $\bar{f}$ that includes all the alignment points.

NOTE:

- If $\bar{e}$ contains only nonaligned points then $\exists$ no $\bar{f}$
- If the matched $\bar{f}$ has additional alignment points, then *it cannot be extracted.*
- If the minimal phrase borders unaligned words, then the extended phrase also extracted.

Phrase Extraction Example

From Philip Koehn.



Phrase Extraction Example

The extracted phrases are: From Philip Koehn.

Michael – michael

Michael assumes – michael geht davon aus

michael geht davon aus ,

Michael assumes that –

michael geht davon aus , dass

Michael assumes that he –

michael geht davon aus , dass er

Michael assumes that he will stay in the house–

michael geht davon aus , dass er im haus bleibt

assumes – geht davon aus | geht davon aus ,

assumes that – geht davon aus , dass

Phrase Extraction Example

assumes that he – geht davon aus , dass er
assumes that he will stay in the house–
geht davon aus , dass er im haus bleibt
that – dass | , dass
that he - dass er | , dass er
that he will stay in the house – dass er im haus bleibt |
, dass er im haus bleibt
he – er
he will stay in the house – er im haus bleibt
will stay – bleibt
will stay in the house - im haus bleibt
in the - im
in the house - im haus
house - haus

Phrase Extraction Algorithm

Input: Word alignment for sentence pair (e ,f)
Output: Set of Phrase pairs PP.

For es = 1 to m // m is length of e, n is length of f
  For ee = es to m

    // Find the minimally matching phrase of f
    fs = n; fe = 0;

  For all $(e, f) \in$ A // A is the set of alignments
  { if $es \le e \le ee$ then
    { fs = min (f, fs)
      fe = max (f, fe)
    }}
  PP = PP $\cup$ *extractfrom* (fs, fe, es, ee)

Phrase Extraction Algorithm

Function *extractfrom* (fs, fe, es, ee)

if $fe = 0$ then return {}
For all $(e, f) \in A$
  if $(e < es)$ or $(e > ee)$ then return {}
$S = \varphi$
$ffs = fs$
repeat
  $ffe = fe$
  repeat
    $S = S \cup (es \ldots ee, ffs \ldots ffe)$
    $ffe{+}{+}$
    until *ffe* is aligned
  $ffs --$
until *ffs* is aligned
return S

Phrase Extraction Example

Some Statistics:

- 9 english Words vs. 10 German words
- 11 alignment points
- 45 contiguous English phrases
- 55 contiguous German phrases
- 24 pairs have been extracted.

Phrase Extraction

Points to note about Phrase Extraction:

- Unaligned words -> Multiple matches
  (Consider for example, the effect of the comma (,))
- No restriction on phrase length.
- Leads to huge number of extracted pairs
- Most long phrases of the training data are unlikely to occur in the test data
- Often a restriction is kept on the max length of a phrase.
- Extracting a huge no of phrases lead to computational burden.
- It is NOT clear whether it has effect on output.

Translation Process

Translation Process

Let us now look at the most practical aspect:

How the translation is generated for a given new sentence **f** ‘

For illustration consider the Italian sentence:

non gli piace il cibo indiano

We expect one person non-conversant with the language will proceed as follows

Translation Process

Non gli piace il cibo indiano

No

Translation Process

Non gli piace il cibo indiano

No Indian

Translation Process

Non gli piace il cibo indiano

No the Indian

Translation Process

Non gli piace il cibo indiano

No the food Indian

Translation Process

Non gli piace il cibo indiano

No the food Indian

Apply Reordering

Translation Process

Non gli piace il cibo indiano

No like the food Indian

Translation Process

Non gli piace il cibo indiano

No he like the food Indian

Translation Process

Non gli piace il cibo indiano

He no like the food Indian

Apply Reordering

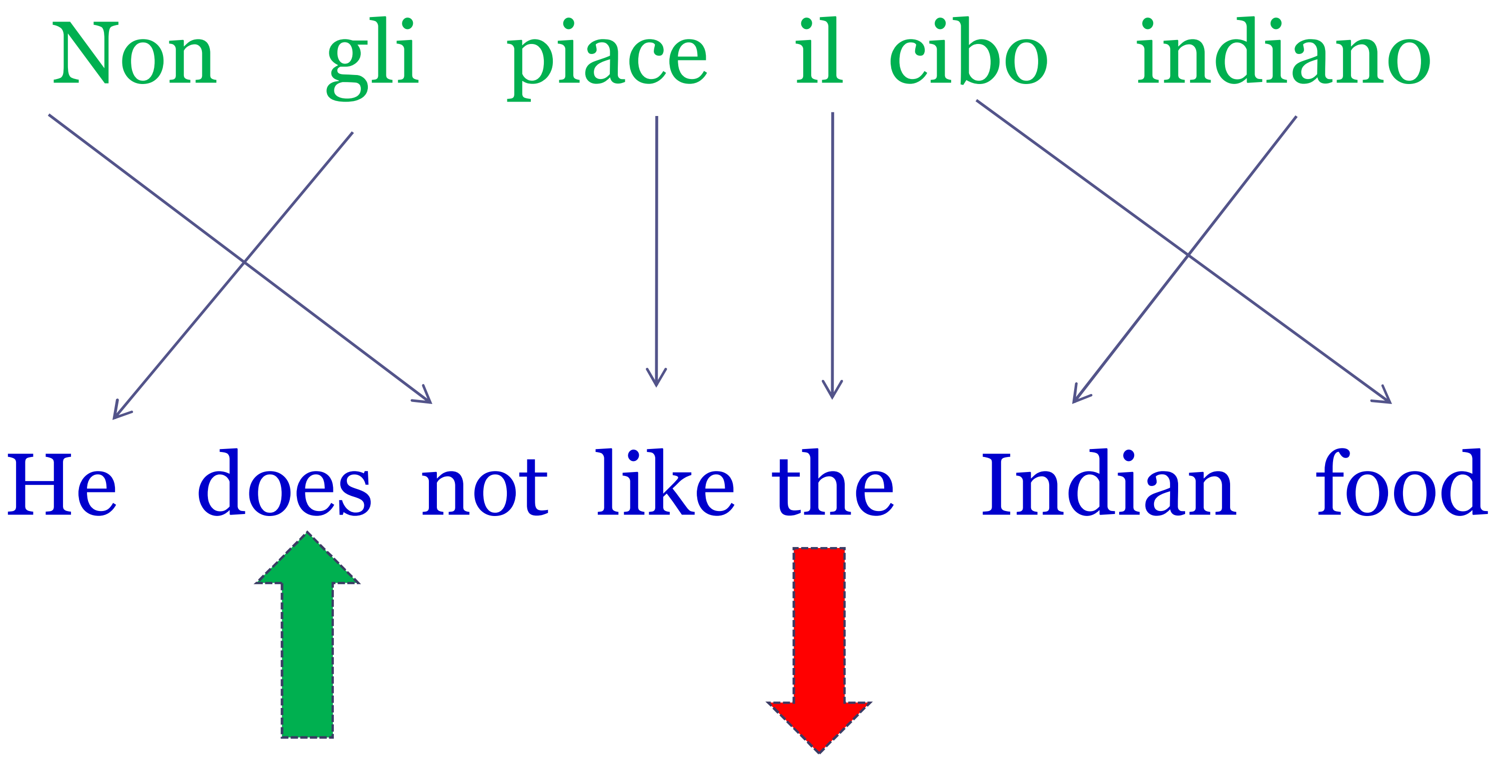
Translation Process

Non gli piace il cibo indiano

He does not like the Indian food

Apply Language Modeling

Translation Process



Apply Language Modeling

Translation Process

Non gli piace il cibo indiano

He does not like Indian food

Got the final Translation!!!

Comments

This example is done at a word level.

But suppose we have Phrase Translation tables, which Gives the following translations directly:

Il cibo indian >> Indian food
Non gli piace >> he does not like

Then the whole translation process will be easier.

Hence Phrase-Based Methods are pursued.

Difficulties

- In each case we may have different options

E.G non >> not, none
gli >> the, him, it
piace >> like
cibo >> food, grub, viands, meat, fare, scoff, cheer
indiano >> Indian, Hindu

So the task is NOT as easy as we thought!!

Difficulties

If we look at from phrases we can get multiple translations: E.g.

non gli piace >> not like (got it from google)
dislikes,
does not like,
is not fond of

Hence most our actions will be probabilistic.

Divergence Observation

Simple sentence translations of this structure from English to Italian :

He does not read >> egli non legge
He does not sing >> egli non canta
He does not cry >> egli non piange

But

He does not like >> non gli piace
He does not speak >> non parla
He does not write >> non scrive

Computing Translation Probability

To make mathematically sound we resort to the following model:

The best translation is obtained from the eq:

$$e_{Best} = \underset{e}{\arg\max} \prod_{i=1}^{I} \phi(\overline{f_i} \mid \overline{e_i}) * pd(d_i)\, p_{Lm}(e)$$

How do we get the probabilities?

Note that there are three components:

- ϕ - matching the SL phrase with the TL phrase.
- *pd* – the phrases are rearranged appropriately
- p_{LM} – the output is fluent as per the TL is concerned

They are now used to compute sentence probabilities

Computing Translation Probability

Note: Each of the individual probabilities are computed separately

- $\varphi(\ )$ - from phrase translation table.
- $pd(\ )$ - as a phrase is translated one stores its end position. This is used as end_{i-1}. For the next phrase we note its position as $start_i$. As these numbers are known one can compute the distortion probability.
- P_{LM} - gives the prob. of a sentence based on n-grams.

As the translation is being constructed the *Partial scores* are built

Phrase Translation Probabilities

Phrase Translation Probabilities

The huge number of extracted phrases prohibits a generative modeling.

Note: In word modeling, where on the basis of word alignment (between input and output text) and counting we could estimate probabilities – designed mathematically.

Here the situation is different:

- there are *finer* phrases and *coarser* phrases.
- we do not know usefulness of them.
- We do not want to eliminate anyone.
- *Counting* does not give the solution.

Let us illustrate from google:

Phrase Translation Probabilities

he >> egli ; does >> fa ; not >> non ; like >> piacere

he does >> lo fa ; does not >> non ; not like >> non come

he does not >> egli non ; does not like >> non piace

he does not like >> non gli piace

How do we store phrases in the Phrase Translation Table?

As a consequence we have to approach Estimation of Phrase translation Probabilities in a different way.

Phrase Translation Probabilities

- For each sentence pair extract a no. of phrase pairs $(\bar{f}, \bar{e})$.
- Count in how many sentence pairs a particular phrase pair is extracted.
- Estimate for $\phi(\bar{f} \mid \bar{e})$ is then the *relative frequency*:

$$\frac{count(\bar{e}, \bar{f})}{\sum_{\bar{f}_k} count(\bar{e}, \bar{f}_k)}$$

Note:

- For large corpus the Table may be several GBs.
- This makes it difficult to use for future translation.
- Typically kept sorted, and partially loaded into RAM.
- Can be used in translation for better performance.

Extension to Translation Model

We started with the following equation:

$$\mathbf{e}_{Best} = \underset{\mathbf{e}}{\arg\max}\, p(\mathbf{e}|\mathbf{f}) = \underset{\mathbf{e}}{\arg\max}\, p(\mathbf{f}|\mathbf{e})p(\mathbf{e})$$

Then $p(\mathbf{f} \mid \mathbf{e})$ is further reduced to

$$\prod_{i=1}^{I} \phi(\overline{f_i} \mid \overline{e_i}) * pd(d_i)$$

Thus our translation model has three components:

1. The phrase translation Table $\phi(\overline{f}, \bar{e})$.
2. The language model $p(e)$
3. The model for reordering $pd(d_i)$, $d_i = \text{start}_i - \text{end}_{i-1} - 1$

This gives the following translation model

Extension to Translation Model

$$e_{Best} = \underset{e}{\operatorname{argmax}} \prod_{i=1}^{I} \phi(\overline{f_i} \mid \overline{e_i}) * pd(d_i) \prod_{i=1}^{m} p(e_i \mid e_1 \ldots e_{i-1})$$

However, not all of them are equally important!

Often the words are well translated but the translation is not o.k.

Hence we prefer more weight to Language Model.

By introducing three weighting constants, we have:

$$e_{Best} = \underset{e}{\operatorname{argmax}} \prod_{i=1}^{I} \phi(\overline{f_i} \mid \overline{e_i})^{\lambda_\phi} * pd(d_i)^{\lambda_d} \prod_{i=1}^{m} p(e_j \mid e_1 \ldots e_{i-1})^{\lambda_L}$$

Log-Linear Model

The above model however mathematically not straightforward.

So we maximize exp(*log p*) instead of the original function *p*.
Hence our function is:

$$\exp\left[\begin{array}{l}\lambda_\phi \sum_{i=1}^{I} \log \phi(\overline{f_i} \mid \overline{e_i}) + \\ \qquad \lambda_d \sum_{i=1}^{I} \log \; pd\,(start_i - end_{i-1} - 1) + \\ \qquad\qquad \lambda_L \sum_{i=1}^{m} p\,(e_i \mid e_1 \ldots e_{i-1})\end{array}\right]$$

Log-Linear Model

- Each translation is considered to be a data point.
- Each data point is considered to be a vector (of features) (Similar to Word Space Model)
- A model has corresponding set of feature functions.
- The feature functions are trained separately, assuming they are independent
- This allows to experiment with different weights for different functions.
- This also allows to add additional modeling (e.g. lexical weighting, word penalty, phrase length penalty) if deemed necessary.

Reordering Model

Reordering Model

Reordering seems to be the most difficult of the tasks.
As most of the reordering are based on language pairs

It is difficult to design a general reordering model –

E.g. Adj NN (English) >> NN Adj (Fr)

Aux V Main V (English) >> Main V Aux V (Hindi)

I am eating Indian food

Je mange la nourriture indienne

I am eating Indian food

main bharatiya khaana khaa rahaa hoon

Consequently total reordering requires much more information!!

Reordering Model

The reordering model we discussed is based on the movement distance d_i.

Does not take care of the underlying word (or its class).

Hence there is a lot a scope to improve reordering:
- e.g *hierarchical*, *reordering with syntactic knowledge*

We discuss one important Reordering Model viz. Lexical Reordering.

Reordering Model (Lexical)

Here we observe that some phrases switch positions more often than others:

Studente di dottorato $_{(It)}$ >> Doctoral student
Traduzione automatica >> Machine translation
Stati Uniti d'America >> United States of America
European Parliament >> Parlamento europeo

Parlement Européen $_{(Fr)}$ >> European Parliament
Bombe atomique >> Atom bomb

America yuktarashtra $_{(B)}$ >> United States of America

Lexical Reordering

Reordering is done on the basis of the Actual phrases.

Here we take statistics of three possible directions of reordering:

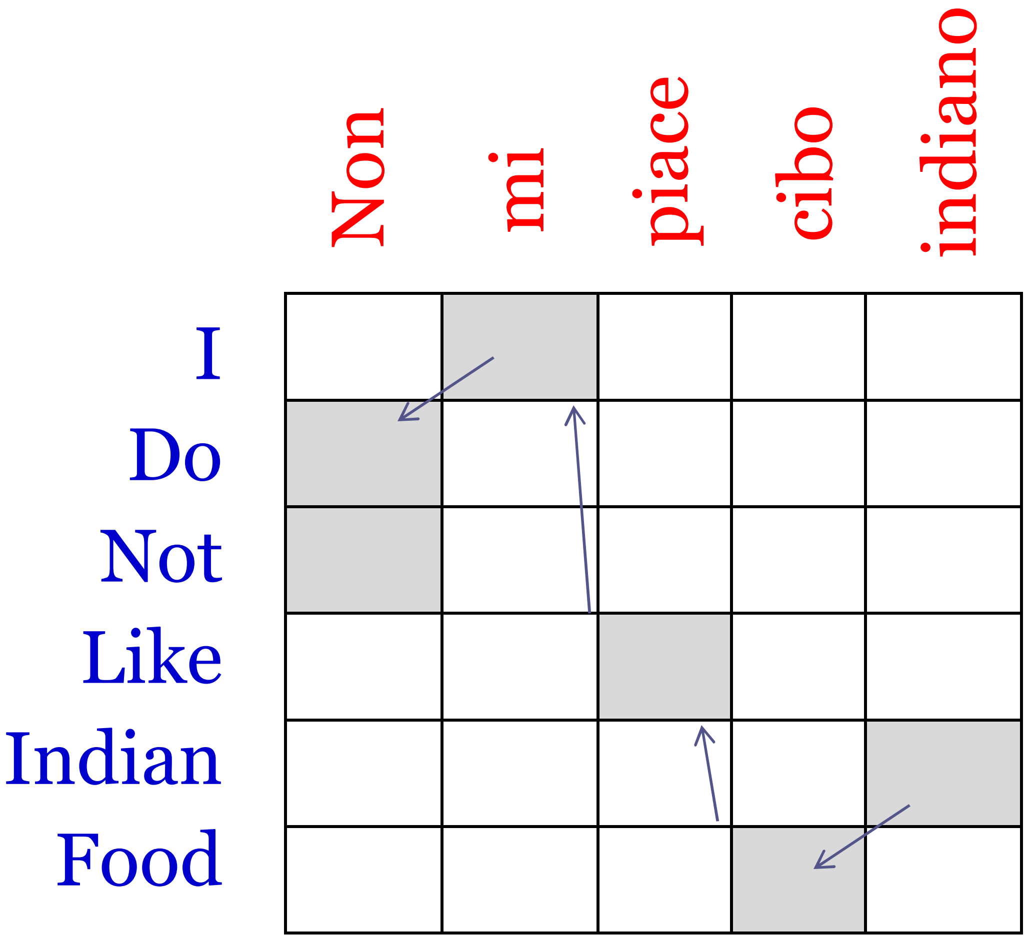
-monotone (m) : two successive phrases in **f** translate into successive phrases in **e**.

-swap (s) : two successive phrases in **f** $(\bar{f}_j, \bar{f}_{j+1})$ translate into two successive phrases $(\bar{e}_i, \bar{e}_{i+1})$ of **e** but word alignment is in reverse order.

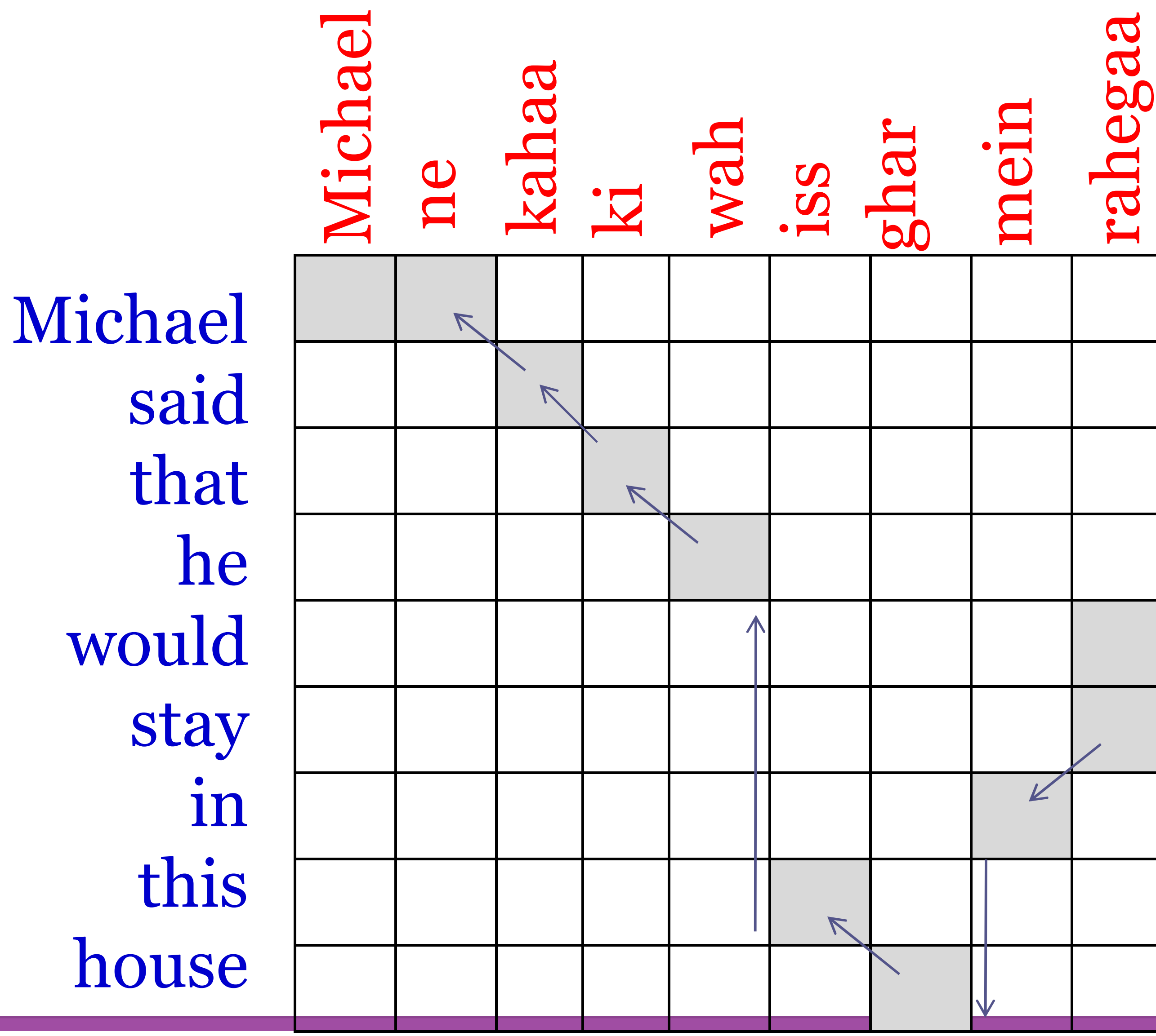
-discontinuous (d): translations of two successive phrases in *f* are not successive in *e*.

How to obtain ?

Lexical Ordering



Lexical Ordering



Lexical Reordering

Note:

a) If there is an alignment point at top left of the cell then it is monotone.
b) If there is an alignment point at the bottom left then it is swap.
c) Otherwise it is a discontinuous.

While doing phrase alignment we take the statistics on *Orientation.*

$$p_o(x \mid \overline{f}, \overline{e}) = \frac{count(x, \overline{f}, \overline{e})}{\sum_o count(o, \overline{f}, \overline{e})}, o, x \in \{m, s, d\}$$

Because of data sparseness often some modified Formula/ techniques are also used.

Lexical Reordering

For example, one can make use of the overall probability of some particular orientation:

Let

$$p_o(x) = \frac{\sum_{\bar{f}} \sum_{\bar{e}} count(x, \bar{f}, \bar{e})}{\sum_{o} \sum_{\bar{f}} \sum_{\bar{e}} count(o, \bar{f}, \bar{e})}, o, x \in \{m, s, d\}$$

Then for a given constant α, one can use the following:

$$p_o(x \mid \bar{f}, \bar{e}) = \frac{count(x, \bar{f}, \bar{e}) + \alpha * p_o(x)}{\sum_{o} count(o, \bar{f}, \bar{e}) + \alpha}, o, x \in \{m, s, d\}$$

Other issues of Reordering

Reordering needs to be dealt with more judiciously:

-It has been noticed that in reordering some groups of words move together

-This happens depending upon the role of the phrase

- There may be *local* reordering and *global*

  E.g. intra phrase and inter-phrase

  * hence distance is not the best measurement
  * as it penalizes large movements
  * but some languages demand it (e.g. SOV vs. SVO)

- Typically reordering is guided by the *Language Model*
- Question is: can a typical 3-gram / 4-gram model can guide long distance reordering?

Other issues

Since reordering is difficult, people advocate:

(1) monotone translation

- it does not give perfect translation.
- but it reduces search complexity from exponential to polynomial.

(2) limited reordering

- allowing only local reordering
- typically intraphrase – as suggested by SL-TL pair.
- Controlled by a small-size window.
- Often gives better translation compared to unrestricted reordering.

Other issues

Lexical Weighting

- Infrequent phrase pairs often cause problems – particularly if collected from noisy data.
- If both $\bar{f}$ and $\bar{e}$ occur only once giving $\phi(\bar{f} \mid \bar{e}) = 1 = \phi(\bar{e} \mid \bar{f})$.
- Lexical weighting is a smoothing method where we back off on more reliable probabilities.

One possible formula (Based on word alignment):

$$lex(\bar{e} \mid \bar{f}, a) = \prod_{i=1}^{length(\bar{e})} \frac{1}{\#\{ j \mid (i, j) \in a \}} \sum_{\forall (i,j) \in a} w(e_i \mid f_j)$$

$w(e_i \mid f_j)$ s are calculated from word-aligned corpus.

Other issues

Word penalty and Phrase Penalty

- comes into consideration if we try to model the length of the translation.

Note that – the language model prefers shorter translations – as fewer n-grams need to be scored.

-Word penalty controls the length of the translation by adding a factor w for each produced word.

$w < 1 \Rightarrow$ shorter translation is preferred

$w > 1 \Rightarrow$ longer translation is preferred

-Phrase penalty – tries to control the number of phrases using a factor ρ

$\rho < 1 \Rightarrow$ fewer phrases are preferred

$\rho > 1 \Rightarrow$ more phrases are preferred

Concluding Remarks

Phrase based seems to be more intuitive than Word Based

But the Algorithm is based on Word Alignment

Hence two questions arise:

1) Can't we extract Phrases directly?

   A joint model of phrase alignment has also been proposed.

   Which uses the EM algorithm

2) How to combine the phrase translations?
   This needs development of Decoder.

In the next class we shall focus on these issues.

Thank you