Machine Translation

Ramaseshan Ramachandran

LET US TRANSLATE - SWAHILI TO ENGLISH

My dog

mbwa	wangu
My	My
Dog	dog

My house

Nyumba	Yangu
My	My
House	House

My cycle

mzunguko	wangu
My	My
Cycle	Cycle

Your house

Nyumba	yako
My	My
House	House

Your cycle

mzunguko	Wako
Your	Your
Cycle	Cycle

This is a house

hii	ni	nyumba
This	This	This
is	is	is
a	a	a
house	house	house

This is your cat

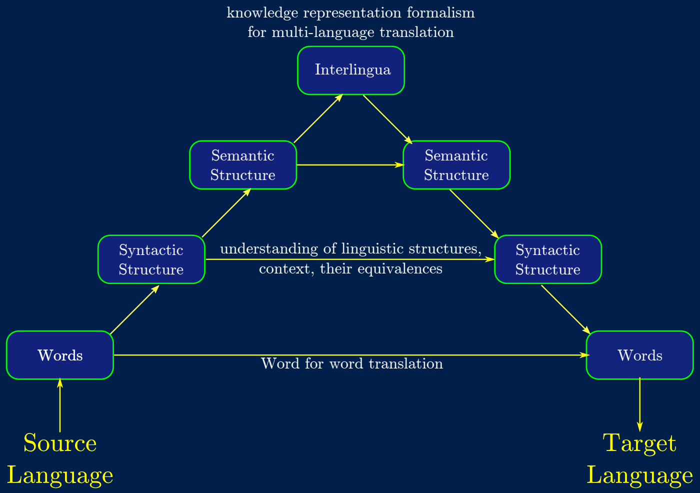
huyu	ni	paka	wako
This	This	This	This
is	is	is	is
your	your	your	your
cat	cat	cat	cat

nyumba	yangu	ni	nyumba	yako

When I look at an article in Russian, I say "This is really written in English, but it has been coded in some strange symbols. I will now proceed to decode." (Warren Weaver, 1947)

VAUQUOIS PYRAMID - VARIOUS APPROACHES TO MT

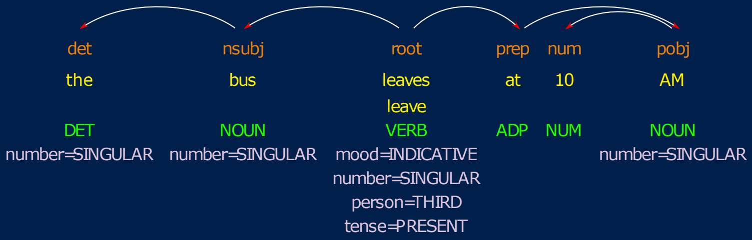
Vauquois pyramid [1]



WORD2WORD OR LITERAL TRANSLATION

Every word from the source language is converted into the target language, one word at a time with out considering the whole sentence as context

SYNTACTIC TRANSLATION



1. The source sentence is parsed to create a syntax tree
2. The nodes of the source tree is mapped to the nodes the similar syntax tree created for the target language -

   $(\text{subject})_s \longrightarrow (\text{subject})_t$

   $(\text{noun})_s \longrightarrow (\text{noun})_t$

   $(\text{det})_s \longrightarrow (\text{det})_t$

   $(\text{adj})_s \longrightarrow (\text{adj})_t$
3. Generate the sentence in the target language sentence from the parse tree

SEMANTICS-BASED TRANSLATION

- The meaning of the source sentence is obtained
- Using the semantics derived from the source sentence, the target sentence is generated

INTERLINGUA TRANSLATION

- A meta-language format for representing knowledge independent of any language
- Instead of Translation systems for all possible pairs of languages, one representation would be used to generate translations
- $O(n^2) \rightarrow O(n)$
- Difficult to design efficient and comprehensive knowledge representation formalisms and due to the large amount of ambiguity

AUTOMATIC MACHINE TRANSLATION

- The idea of *the ability to make anyone speak to anyone without the boundary of languages* is the most appealing idea
- The goal of the automatic translation is to produce error-free translation
  - Preserve the meaning of the source language
- AMT is a hard problem
- Parallel corpora aid in the development of AMT

- Translation by analogy: Example based machine translation (EBMT) (lazy learning)

  Hii ni nyumba yangu - This is my house

  Mbwa wangu anapenda kukimbia - My dog loves to run

  Mimi kukimbia na mbwa wangu - I run with my dog

  This is my dog -

MT FROM EXAMPLES

- Translation by analogy: Example based machine translation (EBMT) (lazy learning)

  Hii ni nyumba yangu - This is my house

  Mbwa wangu anapenda kukimbia - My dog loves to run

  Mimi kukimbia na mbwa wangu - I run with my dog

  This is my dog - Huyu ni mbwa wangu
- Learn MT models from data: Statistical Machine Learning
  - Translation models with language-specific parameters
  - Train model parameters & apply to unseen data

Translations are generated using parameters and models which are derived from the analysis of bilingual text corpora.

SMT

- Every French string, f, is a possible translation of e. We assign to every pair of strings $\{e, f\}$ a number $P(f|e)$, which we interpret as the probability that a translator, when presented with e, will produce f as his translation
- Given a French string f, the job of our translation system [2] is to find the string e that the native speaker had in mind when he produced f

F	E
comment allez-vous?	How are you?
	How do you do?
	How are you doing?
Comment ça va ?	
Vous allez bien?	
Ça va ?	

DEFINITIONS

Let us assume that the task is to translate a French sentence f with a sequence $(f_1, f_2, f_3, ...f_m)$ of length m and f_j for $j \in (1, 2, 3, ...m)$ is the j^{th} word.
The translated English sentence will be assumed to have the sequence $(e_1, e_2, e_3, ...e_n)$ and n is the length of the English sentence.
Let us assume that the corpus consists of the pair of source and translated sentences, $(f^k \, and \, e^k)$.
$f^k = (f^k_1, f^k_2, \ldots, f^k_m)$ where f^k_j is the j^{th} word in the k^{th} French sentence of length m
$e^k = (e^k_1, e^k_2, \ldots, e^k_n)$ where e^k_j is the j^{th} word in the k^{th} English sentence of length n

The parallel copora are available from Canadian parliamentary proceedings (the *Hansards*) and from Europarl data. Europarl data consists of proceedings from the European parliament, and consists of translations between several European languages

PARALLEL CORPORA

A parallel corpora is a collection of corpus that contains a collection of original text and its translation in various languages. In most cases, parallel corpora contain data from two languages.

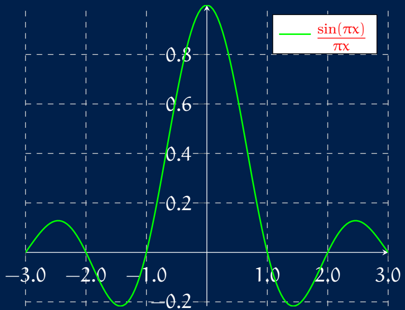
English	French
Resumption of the session	Reprise de la session
I declare resumed the session of the European Parliament adjourned on Friday 17 December 1999, and I would like once again to wish you a happy new year in the hope that you enjoyed a pleasant festive period.	Je déclare reprise la session du Parlement européen qui avait été interrompue le vendredi 17 décembre dernier et je vous renouvelle tous mes vux en espérant que vous avez passé de bonnes vacances.
You have requested a debate on this subject in the course of the next few days, during this part-session	Vous avez souhaitété un détébat à ce sujet dans les prochains jours, au cours de cette pétériode de session.

ARGMAX

The *arguments of the maxima* function f is defined for a set D as

$$\arg\max_{x \in D} f(x) = x | f(x) \geq f(y), \forall y \in D$$

In other words, the argmax are the points of the domain of some function at which the function values are maximized



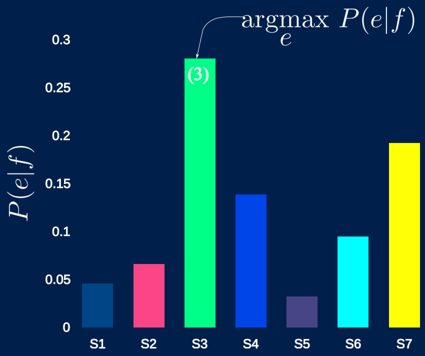
The argmax of the function $\frac{\sin(\pi x)}{\pi x}$ is 0 as the function has the global maximum value of 1

ARGMAX - MT

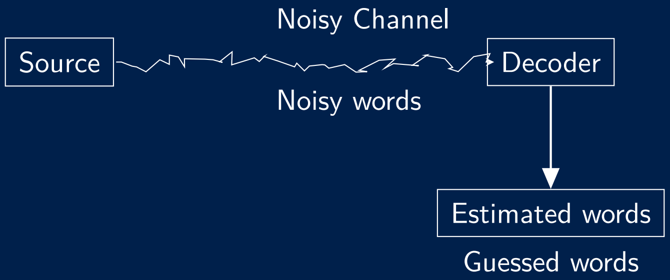
Given a French sentence f, find the most likely English sentence e that maximizes $P(e|f)$. The *arguments of the maxima* function f is defined as,

$$\arg\max_{e} P(e|f) \tag{1}$$

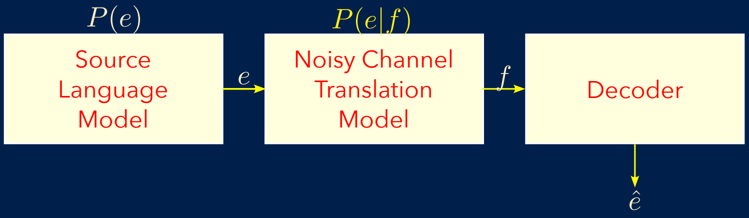
The English sentence e, out of all such sentences, which yields the highest value for $P(e|f)$. It is possible to have more than one translation for a given sentence. In such cases, *argmax* finds one English sentence e that yields the highest value for $P(e|f)$.



THE NOISY CHANNEL MODEL



NOISY CHANNEL FOR MT



$$\hat{e} = p(e) \times p(f|e)$$

The Language Model generates an English sentence e. The Translation Model transmits e as the French sentence f. The decoder finds the English sentence $hate$ which is most likely to have given rise to f[3]. $P(e)$ is the distribution over which sentences are likely in English and $P(f|e)$ is the translation model that indicates the likelihood seeing the French sentence f as a translation of e

Many bilinguals, whose mother tongue is not English, may think of the sentence they want to speak in their mother tongue first and then speak out the translated version in English

BAYES' RULE FOR MT

By applying Bayes' Theorem, the translation problem is broken down into two smaller problems. Assume that we have a French sentence f and we would like to translate into an English sentence e.

From the probabilistic perspective, we want to find the English sentence e that has maximal probability given the French sentence f. Using Bayes rule we can write this problem as

$P(e|f) = \frac{P(f|e)P(e)}{P(f)}$ We can find the English sentence using **the** $\arg\max$

$$\arg\max_e = \arg\max_e P(e|f)$$

$$= \arg\max_e \frac{P(f|e)P(e)}{P(f)}$$

$$\boxed{\hat{e} = \arg\max_e P(f|e)P(e)}$$

$P(f|e)$ – the translation model and
$P(e)$ – the English Language Model
The problem is reduced to modeling these 2 distributions
Now we have to estimate the parameters of the $P(f|e)$ from the training examples (f^k, e^k) for $k = 1 \ldots n$

BIGRAM AND TRIGRAM PROBABILITIES

$$P(w_2|w_1) = \frac{f(w_1, w_2)}{f(w_1)}$$

$f(w1, w2)$ is the number of times w_2 appeared after w_1

$$P(w_3|w_1, w_2) = \frac{f(w_1, w_2, w_3)}{f(w_1, w_2)}$$

$f(w_1, w_2, w_3)$is the number of times w_3 appeared after w_1 and w_2

Or we could use the non-Markovian neurql Language models for LM

SPARSITY AND SMOOTHING

- Newer ways of forming a sentence is common.
- It is possible that a trained model will see a new n-gram
- These new n-grams results in $P(x|y) = 0$
- $P(x|y) = 0$ will propagate through and produce a zero probability for the entire sentence
- Smaller probabilities too create a very small value

To avoid $P(x|y) = 0$, linear interpolation is used.
$P(w_3|w_2, w_1) = \lambda_1 P(w_3|w_2, w_1) + \lambda_2 P(w_2|w_1) + \lambda_3 P(w_1) + \lambda_4$
where $\lambda_1(0.95) + \lambda_2(0.04) + \lambda_3(0.008) + \lambda_4(0.002) = 1$
For new words and n-grams, $P(x|y)$ will always have a small value

KNOWLEDGE CAPTURED BY THE MODEL

I want to eat Chinese food. I want English food. I want to eat english food

$$P_1(english|want) = 0.0011$$
$$P_2(chinese|want) = 0.0065$$
$$P_3(to|want) = 0.66$$
$$P_4(eat|to) = 0.28$$
$$P_4(order|to) = 0.18$$
$$P_5(want|I) = 0.32$$

$$P_6(food|english) = 0.015$$
$$P_7(food|chinese) = 0.15$$
$$P_8(chinese|eat) = 0.34$$
$$P_{10}(english|eat) == 0.001$$
$$P_{11}(i| < s >) = 0.25$$
$$P_{12}(< /s > |food) = 0.12$$

I want ______ food

I want to ______ ______ food.

To avoid underflow values of multiplication to find $P(e)$, one can use log

$$\log(P_1 * P_2 * P_3 * P_4 ... P_n) = \log(P_1) + \log(P_2) + \log(P_3) + \log(P_4) ... \log(P_n)$$

EVALUATION OF THE LANGUAGE MODEL

Can we apply Bayes rule for evaluation of the model? A model can be evaluated based on the test data

$$P(model|testingdataset) = \frac{P(model)P(testing\,data\,set)}{P(testing\,data\,set)} \tag{2}$$

- A better model is one which assigns a higher probability to the word that actually occurs
- The best model is the one that optimizes the $P(model)P(testing\,data\,set)$
- A model that outputs zero probability for any unknown sentence will be discarded

PERPLEXITY

- The tiny numbers of $\mathrm{P}(e)$ may underflow any floating point scheme.
- An n-gram model will assign a very tiny $\mathrm{P}(e)$ for long sequences.
- Many n-gram conditional probabilities may also be a very small value
- The product for $\mathrm{P}(e)$ will be tiny

To compare models, $\mathbb{P} = 2^{-\log_2(\mathrm{P}(e))/|V|}$ is computed. $|V|$ is the number of words in the test data. $\mathbb{P}$ is known as the perplexity score.

$$\mathbb{P} \propto \frac{1}{\mathrm{P}(e)}$$

A good model will have a relatively small perplexity score. The lower the perplexity, the better the model is.

TRANSLATION MODEL

$P(f|e)$ is the chance that upon seeing e, a translator will produce f.

$$P(f|e) = \frac{\text{Count of (f,e)}}{\text{Count of } (e)}$$

In simple terms, translating from French to English is to identify the bag of words in English and later form syntactically correct sentences.

In this model, there is no need to use any French to English translated corpus to train the language model.
Is this correct and will it work?

TRANSLATION

The	book	is	on	the	ta-ble
Le	livre	est	sur	la	ta-ble

$$P(french|english)$$

Le	livre	est	sur	la	ta-ble
The	book	is	on	the	ta-ble

HOW CAN WE TRANSLATE?

- What steps do we take to translate a language?
- As non-native speakers, how do we frame English sentences?
- Do we have a BoW for English, before writing any English sentences?
- Do we assemble word-for-word translation in mind before writing any English sentences?
- Do we assemble BoW in both languages before writing?
- Can it be thought of string rewriting?
- Identify a corresponding word in the other language and use its language model to build the sentence?

TRANSLATION MODEL WITH A FIXED LENGTH OF FRENCH SENTENCE

By fixing the size of the French sentence to m words, we will assume that there is some distribution $P(m|n)$ that models the conditional distribution of French sentence length m conditioned on the English sentence length n. We could also choose a set of words $(f_1, f_2, f_3, ...f_m)$

Now, we can write – the conditional probability of the French sentence is conditioned on the English words of length n and the French sentence of length m.

$$P(f_1, f_2, f_3, ...f_m|e_1, e_2, e_3, ...e_n, m) \tag{3}$$

Is it easy or hard to estimate the distribution of equation (3)?

It is hard to estimate $P(f|e,m)$ directly. Let us introduce the concept of alignment variables

ALIGNMENT

- Consider a seed word in English that starts the translation process
- Assume this seed word, a_j, as the alignment word at the position j^{th} in the English sentence
- The alignment a is $\{a_1, a_2, a_3, ...a_m\}$, where $a_j \in \{0, n\}$
- The possible alignments are $(n+1)^m$
- The idea is to find the most likely alignment

Alignment probability depends on positions of the words, and position relative to neighbors. The likelihood of an alignment depends on how many words align to a certain position

Automatic alignment is the backbone of SMT

BIJECTIVE ALIGNMENT

- Every word in each text is coupled to exactly one word in the other text.
- No word remains uncoupled or left out

(1)	(2)		......		(6)
the	book	is	on	the	ta-ble
|	|	|	|	|	|
Le	livre	est	sur	la	ta-ble

ALIGNMENT - EXAMPLE 1

Alignment with respect to the translation model $P(f|e,m,n)$

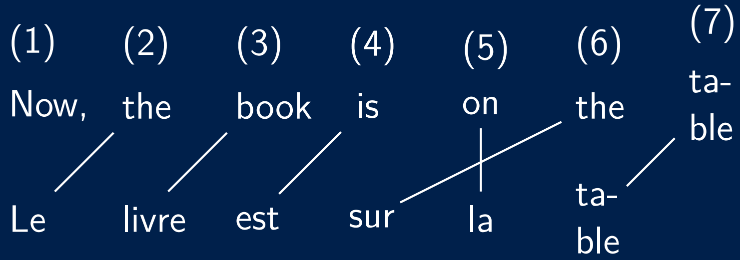
(1)	(2)	(3)	(4)	(5)	(6)	(7)
Now,	the	book	is	on	the	ta-ble
Le	livre	est	sur	la	ta-ble	

$n = 7$ and $m = 6$

The alignment $(a_1, a_2, a_3, a_4, a_5, a_6) = \{2,3,4,5,6,7\}$

ALIGNMENT - EXAMPLE 2

Alignment with respect to the translation model $P(f|e, m, n)$



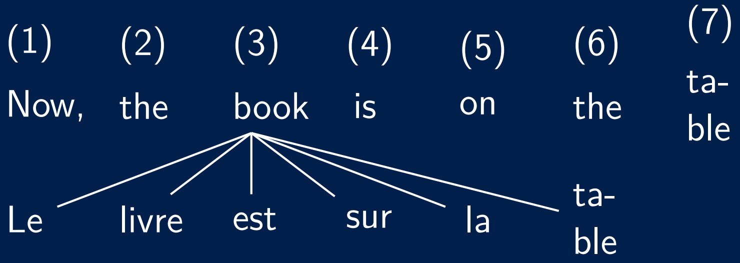
$n = 7$ and $m = 6$

The alignment $(a_1, a_2, a_3, a_4, a_5, a_6) = \{2, 3, 4, 6, 5, 7\}$

The index of the alignment refers to the location of the French word and the value refers to the location of the English word

ALIGNMENT - EXAMPLE 3

Alignment with respect to the translation model $P(f|e, m, n)$



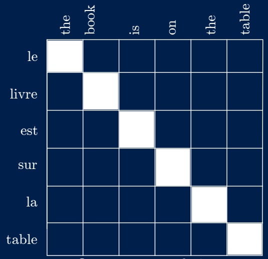
$n = 7$ and $m = 6$

The alignment $(a_1, a_2, a_3, a_4, a_5, a_6) = \{3, 3, 3, 3, 3, 3\}$

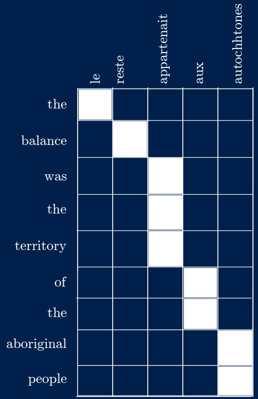
The index of the alignment refers to the location of the French word and the value refers to the location of the English word

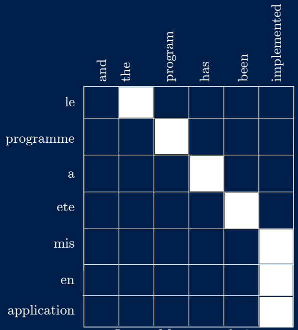
ALIGNMENT - ANOTHER REPRESENTATION

The alignment for the translation model $P(f|e, m, n)$



One-to-one translation





One-to-Many translation

Some examples are from the paper "The Mathematics of Statistical Machine Translation: Parameter Estimation"

https://www.aclweb.org/anthology/J93-2003

ALIGNMENTS

- Insertion - A NULL token is inserted if the target language does not have the equivalent source language word
- One2Many - A source word may translate into more than one target word
- Many2One - Many source words translate into one target word

SAMPLE TABLE FOR TRANSLATION PROBABILITY - $P(f|e,m,n)$

$e =$ Now the book is on the table

$f =$ Le livre est sur la table

	Now	the	book	is	on	the	table
Le	0.006	**0.47**	0.341	0.018	0.128	0.023	0.014
livre	0.108	0.076	**0.416**	0.046	0.048	0.241	0.065
est	0.194	0.101	0.03	**0.421**	0.15	0.057	0.047
sur	0.035	0.116	0.075	0.197	**0.434**	0.121	0.022
la	0.244	0.023	0.289	0.013	0.159	**0.289**	0.242
table	0.108	0.136	0.099	0.035	0.136	0.05	**0.436**

$$p(le|the) > p(le|on) > \ldots > p(le|book) > p(le|now)$$

The parameter, $p(f|e)$, is the conditional probability of generating a French word f from an English word e.

IBM MODEL 1

IBM models [2] are statistical machine translation models. They learn the model parameters by using bilingual corpus. The were part of many SMT systems for more than 20 years

- Lexical translation model (word2word)
- Alignment decisions are independent
- All alignments are equally likely
- The length of the source language sentence is fixed, m
- More than one source language word, (f_j), can be aligned to a single target language word (e_{a_j})

IBM MODEL 1 - TRANSLATION PROBABILITY

The goal is to estimate ***P(e|f)***. This is broken into two small distributions
(1) Translation model - ***P(f,a|e,m)***)
(2) Language Model - ***P(e)***

English sentence - $e_1, e_2, e_3, \ldots, e_n$
French Sentence - $f_1, f_2, f_3, \ldots, f_m$
$a = \{a_1, a_2, a_3, \ldots, a_m\}$ - alignment indicates that from which English word each French word originated from - each alignment, $a_j \in [0, m]$. Estimate the translation probability

$$P(f, a|e, m) = P(a|e, m) \times P(f|a, e, m) \tag{4}$$

where $P(a|e, m)$ is the probability distribution of possible alignments

$$\begin{aligned} P(f|e, m) &= \sum_{a \in A} P(f, a|e, m) \\ &= \sum_{a \in A} P(a|e, m) \times P(f|a, e, m) \end{aligned} \tag{5}$$

IBM MODEL 1 - TRANSLATION PROBABILITY

1. Find the alignment -
$$P(a|e,m) = \frac{1}{(1+n)^m}$$
2. Find the French word alignment probability, given the alignment variable, English word and fixed length of French Sentence
$$P(f|a,e,m) = \prod_{j=1}^{m} p(f_j|e_{a_j})$$
3. Find the most probable alignment variables for every pair of e and f

using,

$$P(f,a|e,m) = P(a|e,m) \times P(f|a,e,m)$$
$$= \frac{1}{(1+n)^m} \times \prod_{j=1}^{m} p(f_j|e_{a_j})$$
$$p(f_j|e_{a_j}) = \frac{C(f_j, e_{a_j})}{\sum_{a \in A} C(f_j, e_{a_j})}$$

where $C(f_j, e_{a_j})$ is the count of the french word f_j aligned with the english word e_{aj}

$$\boxed{\text{Finally, } \hat{e} = \arg\max_{e \in E} = P(e)P(f,a|e,m)}$$

IBM MODEL 1 - EXAMPLE

$n = 7$ and $m = 6$

$e =$ Now the book is on the table

$f =$ Le livre est sur la table

$a = \{2, 3, 4, 5, 6, 7\}$

$$\begin{aligned} P(f|a,e,m) =& p(Le|the) \times p(livre|book) \\ & \times p(est|is) \times p(sur|on) \times p(la|the) \\ & \times p(table|table) \end{aligned}$$

$$p(le|the) = \frac{Count(the, Le)}{Count(the)} \ldots$$

$$P(f, a|e, 6) = \frac{1}{(1+7)^6} \times P(f|a, e, 6)$$

IBM MODEL 1 - TRAINING

- If the alignments are known, then it is possible to estimate the translation probabilities by counting the aligned words
- If the translation probabilities are known, then it is possible to estimate the alignments
- We do not know both - Incomplete data
- Hence an iterative approach with refinement of these values over time is used

If we had complete data, we could estimate model
If we had the model, we could fill in the missing information
To solve this incomplete problem, we use ***Expectation maximization*** algorithm

1. Initialize model parameters (equally likely)
2. Assign probabilities to the missing data
3. Estimate model parameters from completed data
4. Iterate steps 2-3 until convergence

PYTHON CODE FOR EM - SWAHILI → ENGLISH

1. Initialization

```
import numpy as np

# Define the source and target vocabularies
src_vocab = ['my', 'house', 'cycle', 'his','dog']
tgt_vocab = ['mbwa', 'wangu', 'nyumba', 'yangu','mzunguko', 'wake

# Define the parallel corpus
src_sents = [['my', 'dog'],      ['my', 'house'],
                 ['my', 'cycle'], ['his', 'dog']]
tgt_sents = [['mbwa', 'wangu'],      ['nyumba', 'yangu'],
                     ['mzunguko', 'wangu'],  ['mbwa','wake

# Initialize the translation probability matrix
# The matrix has shape (num_source_words, num_target_words)
```

PYTHON CODE FOR EM - SWAHILI → ENGLISH - E-STEP

```
for iter in range(num_iterations):
    #initialize count_matrix and total probability- not shown here
    for i in range(len(src_sents)):
            norm_factor = np.zeros(len(tgtt_sents[i]))
    # Compute the posterior probability of each alignment
    post_prob = np.zeros((len(src_sents[i]), len(tgt_sents[i])))
    for j in range(len(src_sents[i])):
        for k in range(len(tgt_sents[i])):
            norm_factor[k] += trans_prob[k, tgt_vocab.index(tgt_sents[i][j])]
            if norm_factor[k]  == 0 :   norm_factor[k] = 1
            post_prob[j, k] = trans_prob[src_vocab.index(src_sents[i][j]), tgt_
    # Update the count matrix and total probability vector
    for j in range(len(src_sents[i])):
        for k in range(len(tgt_sents[i])):
            count_matrix[src_vocab.index(src_sents[i][j]), target_vocab.index(t
            total_prob[src_vocab.index(src_sents[i][j])] += post_prob[j, k]
```

PYTHON CODE FOR EM - SWAHILI → ENGLISH - M-STEP

```
# Maximization step
for i in range(len(source_vocab)):
    for j in range(len(target_vocab)):
        if total_prob[i] == 0: # Avoid division by zero
            translation_prob[i, j] = 1 / len(target_vocab)
        else:
            translation_prob[i, j] = count_matrix[i, j] / total_p
```

SAMPLE TRACE OF TRANSLATION TABLE - ITERATION 1

	my	dog	house	cycle	your	this	is	a	cat
mbwa	0.50	0.50	0.0000	0.00	0.0000	0.000	0.000	0.0000	0.0000
wangu	0.50	0.25	0.0000	0.25	0.0000	0.000	0.000	0.0000	0.0000
nyumba	0.15	0.00	0.4000	0.00	0.1500	0.100	0.100	0.1000	0.0000
yangu	0.50	0.00	0.5000	0.00	0.0000	0.000	0.000	0.0000	0.0000
mzunguko	0.25	0.00	0.0000	0.50	0.2500	0.000	0.000	0.0000	0.0000
hii	0.00	0.00	0.2500	0.00	0.0000	0.250	0.250	0.2500	0.0000
wako	0.00	0.00	0.0000	0.25	0.3750	0.125	0.125	0.0000	0.1250
yako	0.00	0.00	0.5000	0.00	0.5000	0.000	0.000	0.0000	0.0000
ni	0.00	0.00	0.1429	0.00	0.1071	0.250	0.250	0.1429	0.1071
huu	0.00	0.00	0.0000	0.00	0.2500	0.250	0.250	0.0000	0.2500
paka	0.00	0.00	0.0000	0.00	0.2500	0.250	0.250	0.0000	0.2500

SAMPLE TRACE OF TRANSLATION TABLE - ITERATION 5

	my	dog	house	cycle	your	this	is	a	cat
mbwa	0.1724	0.8276	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
wangu	0.9410	0.0368	0.0000	0.0222	0.0000	0.0000	0.0000	0.0000	0.0000
nyumba	0.0066	0.0000	0.9541	0.0000	0.0066	0.0068	0.0068	0.0190	0.0000
yangu	0.8499	0.0000	0.1501	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
mzunguko	0.0179	0.0000	0.0000	0.9638	0.0183	0.0000	0.0000	0.0000	0.0000
hii	0.0000	0.0000	0.0442	0.0000	0.0000	0.1999	0.1999	0.5559	0.0000
wako	0.0000	0.0000	0.0000	0.0200	0.9800	0.0000	0.0000	0.0000	0.0000
yako	0.0000	0.0000	0.1501	0.0000	0.8499	0.0000	0.0000	0.0000	0.0000
ni	0.0000	0.0000	0.0023	0.0000	0.0000	0.4842	0.4842	0.0292	0.0000
huu	0.0000	0.0000	0.0000	0.0000	0.0000	0.1206	0.1206	0.0000	0.7588
paka	0.0000	0.0000	0.0000	0.0000	0.0000	0.1206	0.1206	0.0000	0.7588

SAMPLE TRACE OF TRANSLATION TABLE - ITERATION 10

	my	dog	house	cycle	your	this	is	a	cat
mbwa	0.0819	**0.9181**	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0
wangu	**0.9938**	0.0045	0.0000	0.0017	0.0000	0.0000	0.0000	0.0000	0.0
nyumba	0.0002	0.0000	**0.9966**	0.0000	0.0002	0.0005	0.0005	0.0020	0.0
yangu	**0.9278**	0.0000	0.0722	0.0000	0.0000	0.0000	0.0000	0.0000	0.0
mzunguko	0.0012	0.0000	0.0000	**0.9976**	0.0012	0.0000	0.0000	0.0000	0.0
hii	0.0000	0.0000	0.0079	0.0000	0.0000	0.1697	0.1697	0.6527	0.0
wako	0.0000	0.0000	0.0000	0.0013	**0.9987**	0.0000	0.0000	0.0000	0.0
yako	0.0000	0.0000	0.0722	0.0000	**0.9278**	0.0000	0.0000	0.0000	0.0
ni	0.0000	0.0000	0.0000	0.0000	0.0000	**0.4993**	**0.4993**	0.0013	0.0
huu	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	**1.0**
paka	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	**1.0**

IBM MODEL 2

Additional Step: Estimating Alignment Probabilities

- IBM Model 1 estimates translation probabilities for each word pair in the parallel corpus.
- IBM Model 2 includes an additional step to estimate alignment probabilities.
- Alignment probabilities represent the probability that a French word aligns with an English word.
- These probabilities are used to re-estimate the translation probabilities in Model 2.
- The alignment probabilities are estimated using the following formula:

$$q(j|i,m,n) = \frac{t(f_i|e_j)}{\sum_{i'=1}^{m} t(f_{i'}|e_j)}$$

where $t(f_i|e_j)$ is the translation probability of French word f_i given English word e_j, and m and n are the lengths of the French and English sentences, respectively.

IBM MODEL 2

Two parameters of the alignment model are defined as

1. The conditional probability of generating a French word f_j, given the English word, e_j - $p(f_j|e_i)$, where n and m are the lengths of the English and French sentences, respectively
2. $q(j|i,n,m)$ is the probability of alignment variable a_i taking the value j, conditioned on the lengths n and m of the English and French sentences, respectively.

$$P(a|e,m) = \prod_{j=1}^{m} q(a_j|j,n,m), \text{ where } a = \{a_1, a_2, a_3, \ldots, a_m\}$$

$$\therefore P(f,a|e,m) = \prod_{j=1}^{m} q(a_j|j,n,m)t(f_j|e_{a_j})$$

$$\tilde{e} = \underset{e \in E}{\arg\max} = P(e) \times P(a|e,m) \times P(f,a|e,m)$$

IBM MODEL 2 - SWAHILI-ENGLISH TRANSLATION TABLE

IBM MODEL 2 - FRENCH → ENGLISH EXAMPLE

$n = 7$ and $m = 6$

$e =$Now the book is on the table

$f =$Le livre est sur la table

$a = \{2, 3, 4, 5, 6, 7\}$

$$\begin{aligned} P(a|e, m) = & q(2|1, 7, 6) \\ & \times q(3|2, 7, 6) \\ & \times q(4|3, 7, 6) \\ & \times q(5|4, 7, 6) \\ & \times q(6|5, 7, 6) \\ & \times q(7|6, 7, 6) \end{aligned}$$

$$\begin{aligned} P(f|a, e, m) = & P(Le|the) \\ & \times t(livre|book) \\ & \times t(est|is) \\ & \times t(sur|on) \\ & \times t(la|the) \\ & \times t(table|table) \end{aligned}$$

$$P(le|the) = \frac{Count(the, Le)}{Count(the)} \ldots$$

$$P(f, a|e, 6) = P(a|e, 6) \times P(f|a, e, m)$$

If we know the parameters q and t, it is easy to find the most probable alignment sequence a for any pair of French and English sentences.

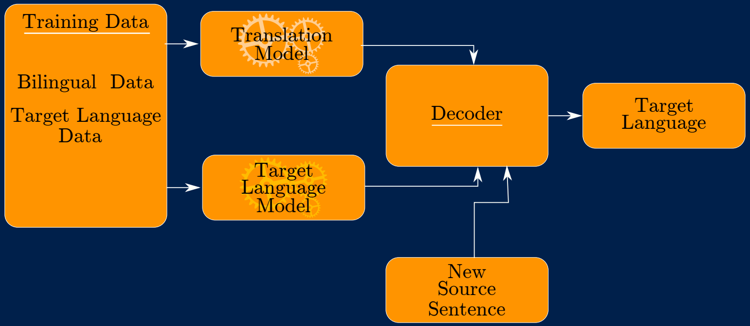
$$a_j = \arg\max_{e \in E} q(a|j, l, m) \times p(f_j|e_a), \qquad \text{for } j = 1..m$$

IBM MODELS

There other models that improve the translation probability. These model are no longer used, but they are used in state of the art NMT models

- To estimate the lexical probability $t(f|e)$
- To derive alignments

Statistical Machine Translation



The translation model represents the probable word translations. The language model encodes the generative model that computes the sentence confidence in terms of probability. The decoder searches for the most likely target word sequence from a large amount of hypotheses using these two models

DECODING PROCESS - 1

le	livre	est	sur	la	table
the	book	been	about	the	table
it	pound	have	over	it	desk
	ledger	belong	out		tableware
	volume	eastern	of		table-top
	novel	eastward	after		booth
	textbook	easterly	on		bench
	0.07781586	is	to		chart
	0.19699646	was	in		desktop
	0.05338291	has	of		panel
	0.27595864	are	at		board
	0.2202764		for		
	0.17556973		with		

What next?

A phrase-based translation system can consider inputs and outputs in terms of sequences of phrases and can handle more complex syntaxes than word-based systems. However, long-term dependencies are still difficult to capture in phrase-based systems

PHRASE-BASED TRANSLATION

- Uses Noisy-channel model
- Uses phrase (contiguous subsequence of a sentence or a span of tokens) as the atomic unit - not to be confused with the Linguistic phrases
- Four stages
  1. Use IBM model to align words
  2. Phrase-to-Phrase alignments
  3. Extraction of phrases
  4. Construct phrase probability table

Reference: Statistical Phrase-Based Translation,Philipp Koehn, et al, 2003

Let e be the target language and f be the foreign language. Let e_i be the i^{th} word and f_j be the j^{th} word for e, f, respectively

$$\hat{e} = \arg\max_{e \in E} P(e)P(f|e) \tag{6}$$

$\arg\max$ is a search operation to predict the English sentence with the highest probability

ADVANTAGES OVER WORD2WORD TRANSLATION

Syntactic models such as IBM models map source tokens into foreign tokens does not account for phrases. Hence, they do not lead to better translations

- Many to many translation possible - can handle non-compositional phrases and idioms
- Use of local context - using nearest neighbors
- The number words in the phrase may dictate the correct word order
- If the learned phrases are longer, the whole sentence is translated

HOW TO FIND PHRASE ALIGNMENTS?

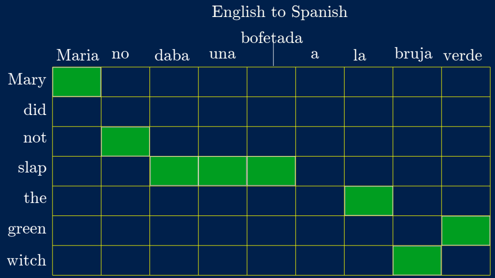
Use symmetrization of the alignments -

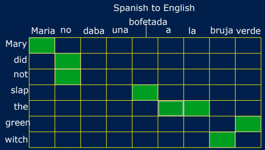
- Use alignment in both directions

  Find (Source $\rightarrow$ Target) and (Target $\rightarrow$ Source) alignments
- Apply ”Intersect” to get precise alignments
- Apply ”Union to find intermediate points

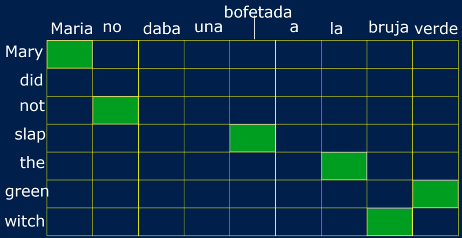
SYMMETRIZATION OF ALIGNMENTS

A method for aligning phrase-to-phrase alignments for a pair of sentences (F,E)is called as ***symmetrization***

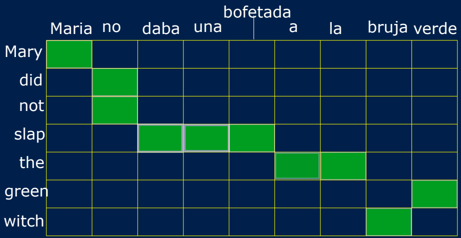




$$(f \rightarrow e) \cap (e \rightarrow f)$$



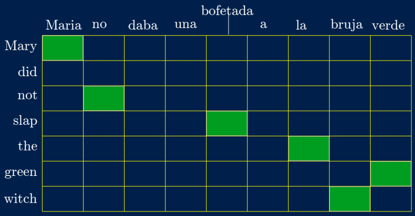
$$(e \rightarrow f) \cup (f \rightarrow e)$$



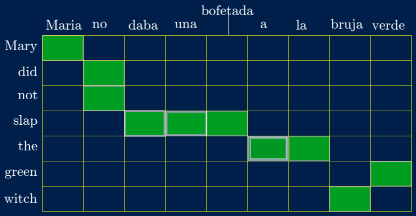
HEURISTICS FOR GROWING ALIGNMENTS

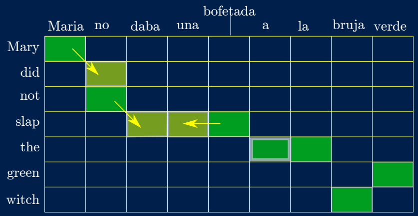
1. To insert new alignment point, search for the alignment points in $P(e|f) \cup P(f|e)$ alignments
2. If not available in (1), do not fill alignment points
3. Check for points that are not aligned already
4. Start filling the diagonal neighbors and adjacent points

$e \to f \cap f \to e$

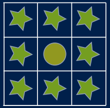


$e \to f \cup f \to e$





Symmetrization heuristic adds neighboring alignment points from the union and unaligned points to the intersection



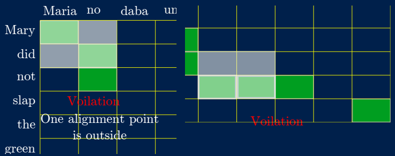
Alignment filling Heuristics

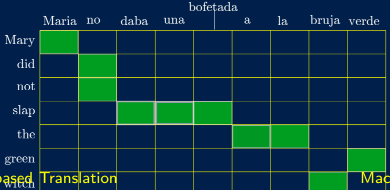
1. $A = f2E \cap e2f$
2. Grow alignment points using $f2E \cup e2f$
3. Finalize

Och and Ney, A Systematic Comparison of Various Statistical Alignment Models, Comp. Linguistics 2003)Sta

EXTRACTION OF PHRASES

The goal is to extract every possible pair of (f, e)





A phrase-pair (e, f) is consistent only when

- There is at least one word in e aligned to a word in f
- There are no words in f aligned to words outside e
- There are no words in e aligned to words outside f

- (Maria, Mary)
- (no, did not)
- (Maria no, Mary did not)
- x (no daba, did not slap)
- (no dabaunabof', did not slap)
- (daba una bof', slap)
- (a la, the)
- (verde, green)
- (bruja, witch)
- (brujaverde, green witch)
- x (Maria no daba una bofetada, Mary did not slap)
- (Maria no daba una bofetada a la, Mary did not slap the)
- (daba una bofetada a la bruja verde, slap the green witch)
- (Maria no daba una bofetada a la bruja verde, Mary did not slapthe green witch)

EXAMPLE - TAMIL2ENGLISH

English to Tamil

	அரசன்	தனது	கண்ணாடியுடன்	முயலைப்	பார்த்தார்
The					
king					
saw					
the					
rabbit					
with					
his					
glasses					

Tamil to English

	அரசன்	தனது	கண்ணாடியுடன்	முயலைப்	பார்த்தார்
The					
king					
saw					
the					
rabbit					
with					
his					
glasses					

English to Tamil

	அரசன்	தனது	கண்ணாடியுடன்	முயலைப்	பார்த்தார்
The					
king					
saw					
the					
rabbit					
with					
his					
glasses					

Tamil to English

	அரசன்	தனது	கண்ணாடியுடன்	முயலைப்	பார்த்தார்
The					
king					
saw					
the					
rabbit					
with					
his					
glasses					

Tamil to English

	அரசன்	தனது	கண்ணாடியுடன்	முயலைப்	பார்த்தார்
The					
king					
saw					
the					
rabbit					
with					
his					
glasses					

SIZE OF THE PHRASE TABLE

- Very large size - bigger than the parallel corpora - reside in memory
- Extract all the phrases and store them in a database or disk

TRANSLATION PROBABILITIES

- Collect all the phrase pairs from the parallel corpora
- Assign probabilities to phrase translations

$$\text{Relative frequency} = p(\bar{f}|\bar{e}) = \frac{count(\bar{e}, \bar{f})}{\sum_i count(\bar{e}, \bar{f}_i)} \tag{7}$$

$$Example$$
$$p(daba\,una\,bofetada|slap) = \frac{C(daba\,una\,bofetada, slap)}{C(slap)} \tag{8}$$

$$\begin{aligned}\hat{e} &= \arg\max_{e \in E} P(e|f) \\ &= P(e) \times P(f|e) \quad (9) \\ &= \arg\max_{e \in E} \prod_{j=1}^{J} p(\bar{f}_j|\bar{e}_j) d(a_j - b_{j-1}) P(e) \quad (10)\end{aligned}$$

- $p(\bar{f}_j|\bar{e}_j)$ is the probability score for the translation of the phrase f, given e
- $d(a_j - b_{j-1})$ is the reordering score for the phrase which is modeled by the distortion probability distribution. a_j denotes the start position of the foreign word and b_{j-1} denotes the end position of the foreign phrase translated into the $j-1$ English phrase.
- This could be simplified by $\alpha^{|a_j - b_{j-1} - 1|}$
- $P(e)$ is the language model - could be a trigram/fourgram model $p(w_i|w_{i-(n-1)}, \ldots w_{i-1})$

- Start with an empty hypothesis
- A sequence of untranslated foreign words and a possible set of phrases for English are chosen
- The foreign words are marked as translated and the probability cost of the hypothesis is updated
    - $\text{cost} = p(e) \times p(\bar{f_i}|\bar{e_i}) \times d(.)$

REFERENCES I

[1] Bernard Vauquois. "A survey of formal grammars and algorithms for recognition and transformation in mechanical translation.". In: *Ifip congress (2)*. Vol. 68. 1968, pp. 1114–1122.

[2] Peter F. Brown et al. "The Mathematics of Statistical Machine Translation: Parameter Estimation". In: *Comput. Linguist.* 19.2 (June 1993), pp. 263–311. ISSN: 0891-2017. URL: http://dl.acm.org/citation.cfm?id=972470.972474.

[3] Manning et al. *Foundations of statistical natural language processing*. Mit Press. MIT Press, 1999. ISBN: 9780262133609. URL: https://books.google.co.in/books?id=YiFDxbEX3SUC.