Introduction to Python and Natural Language Technologies

11. Machine Translation

Dávid Márk Nemeskey

MTA SZTAKI

November 15, 2017

Overview

Introduction

History

The need for translation has always been obvious in human history.
Machine translation

- 1954-66: Early systems
  - Important because of the Cold War
  - Computers were in their infancy
- 1966-1980: MT (and AI) "winter"
  - ALPAC report: MT is of low quality
  - Funding reduced considerably
- 1980-: Classical MT
  - Semantic / interlingua approaches in the '80s
  - Statistical methods in the '90s
  - Widely deployed commercial systems (Google Translate) by 2007
- 2014-: Deep Learning

Is it any good?

The main objection against MT in the ALPAC report was low quality.
Example: a round-translation from English through Russian:

- *The spirit is willing, but the flesh is weak.*

Is it any good?

The main objection against MT in the ALPAC report was low quality.
Example: a round-translation from English through Russian:

- *The spirit is willing, but the flesh is weak.*
- *The whisky is strong, but the meat is rotten.*

Is it any good?

The main objection against MT in the ALPAC report was low quality. Example: a round-translation from English through Russian:

- *The spirit is willing, but the flesh is weak.*
- *The whisky is strong, but the meat is rotten.*

Unfortunately, this is an urban legend.

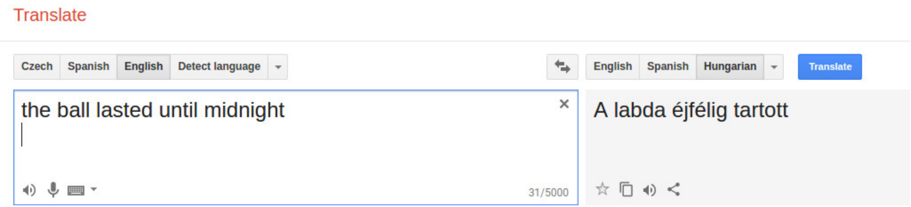
Is it any good?

The main objection against MT in the ALPAC report was low quality.
Example: a round-translation from English through Russian:

- *The spirit is willing, but the flesh is weak.*
- *The whisky is strong, but the meat is rotten.*

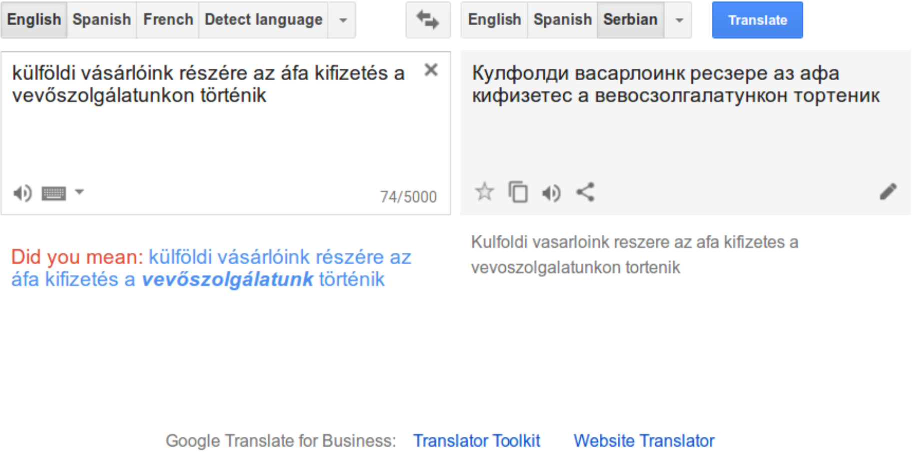
Unfortunately, this is an urban legend.

Still, it is not hard to find errors in Google Translate:



Is it any good?

An even more outrageous example:



Is it any good?

For entertainment...

- Google Translate Songs with Idris Elba (FAKE!)
- Google Translate Sings: Let It Go
- Google Translate Sings: Bohemian Rapsody

Why is it hard?

Machine translation is hard because it involves translating not only between words, but also the various linguistic structures used by the source and target languages. These structures can be different in many ways. These so-called **translation divergences** fall into two main groups:

- Some differences have a **systematic** structure across languages. These can be modelled generally for many languages.
- **Idiosynchretic** differences are arbitrary and must be dealt with one by one.

Translating words is not easy either: languages make different distinctions between concepts. This is called **Lexical divergence**.

Systematic differences

Morphological differences: we have already covered this in lecture 8.

Syntax:

Systematic differences

Morphological differences: we have already covered this in lecture 8.

Syntax:

- **Sentence word order**:
  - *SVO (Subject-Verb-Object)*: English, German. Usually have *prepositions*.
  - *SOV*: Irish, Japanese. Usually use *postpositions*.

Systematic differences

Morphological differences: we have already covered this in lecture 8.

Syntax:

- **Sentence word order**:
  - *SVO (Subject-Verb-Object)*: English, German. Usually have *prepositions*.
  - *SOV*: Irish, Japanese. Usually use *postpositions*.
- **Argument marking**:
  - *Head-marking*: Hungarian, e.g. *az ember ház-a*
  - *Dependent-marking*: English, e.g. *the man-'s house*

Systematic differences

Morphological differences: we have already covered this in lecture 8.

Syntax:

- **Sentence word order**:
  - *SVO (Subject-Verb-Object)*: English, German. Usually have *prepositions*.
  - *SOV*: Irish, Japanese. Usually use *postpositions*.
- **Argument marking**:
  - *Head-marking*: Hungarian, e.g. *az ember ház-a*
  - *Dependent-marking*: English, e.g. *the man-'s house*
- **Marking direction of motion**:
  - *Verb-framed*: Japanese, Romance languages, e.g. *La botella salió flotando*
  - *Satellite-framed*: Hindi, Hungarian, Germanic languages, e.g. *The bottle floated out*

Idiosyncratic differences

Idiosyncratic differences range from syntactic to purely punctuational.

Idiosyncratic differences

Idiosyncratic differences range from syntactic to purely punctuational.

- **Noun phrase order**:
  - English, Hungarian is head last: *the green witch*
  - French, Spanish is head first (mostly): *la bruja verde*

Idiosyncratic differences

Idiosyncratic differences range from syntactic to purely punctuational.

- **Noun phrase order**:
  - English, Hungarian is head last: *the green witch*
  - French, Spanish is head first (mostly): *la bruja verde*
- **POS subdivisions**:
  - Japanese has two types of adjectives (*i*/*na*), with different inflection rules

Idiosyncratic differences

Idiosyncratic differences range from syntactic to purely punctuational.

- **Noun phrase order**:
  - English, Hungarian is head last: *the green witch*
  - French, Spanish is head first (mostly): *la bruja verde*
- **POS subdivisions**:
  - Japanese has two types of adjectives (*i*/*na*), with different inflection rules
- **Date and time**:
  - date format: `YYYY.MM.DD`, `DD/MM/YY`, `MM/DD/YY`, ...
  - calendar used (Gregorian, Chinese, Japanese...)

Lexical divergences

Lexical divergence refers to the phenomenon where languages make different distinctions between concepts.

Lexical divergences

Lexical divergence refers to the phenomenon where languages make different distinctions between concepts.

More specific words for siblings in Hungarian and Japanese then in English:

	brother	sister
older	báty, 兄	nővér, 姉
younger	öccs, 弟	húg, 妹

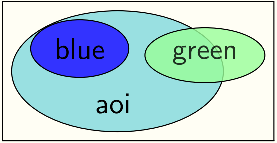
Lexical divergences

Lexical divergence refers to the phenomenon where languages make different distinctions between concepts.

More specific words for siblings in Hungarian and Japanese then in English:

	brother	sister
older	báty, 兄	nővér, 姉
younger	öccs, 弟	húg, 妹

Colors: the Japanese word for blue, 青い (*aoi*), also means green sometimes



The Classical Model

The Classical Model

Here we introduce the main classical (rule-based) MT architectures.
Classical systems have been superseded by statistical models, but the ideas are applicable to even deep learning models.

The Classical Model

Here we introduce the main classical (rule-based) MT architectures. Classical systems have been superseded by statistical models, but the ideas are applicable to even deep learning models.

Translation is done in three steps:

1. **Analysis**: parse the input sentence into some representation
2. **Transformation**: between source- and target-language representations
3. **Generation**: target-language text from target-language structures

Classical architectures

Three main classical translation approaches exist:

1. **Direct translation**:
   - word-by-word translation using a bilingual dictionary
   - nominal analysis and generation steps
2. **Transfer**:
   - the input is parsed at some level(s)
   - and transformation rules convert source-language parses to target-language parses
3. **Interlingua**:
   - the input is analyzed into a language-agnostic abstract meaning representation
   - no transformation step

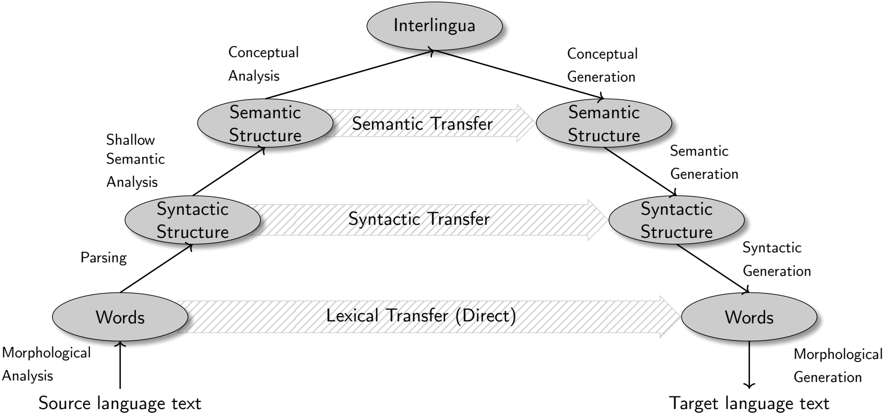
Classical architectures

Three main classical translation approaches exist:

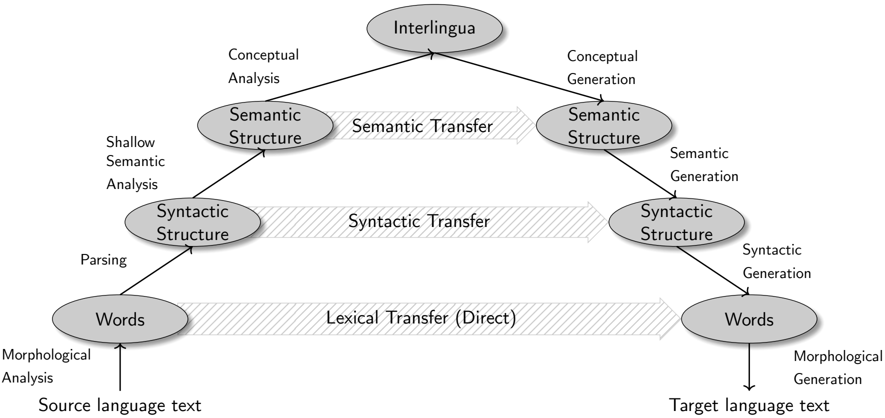
1. **Direct translation**:
   - word-by-word translation using a bilingual dictionary
   - nominal analysis and generation steps
2. **Transfer**:
   - the input is parsed at some level(s)
   - and transformation rules convert source-language parses to target-language parses
3. **Interlingua**:
   - the input is analyzed into a language-agnostic abstract meaning representation
   - no transformation step

The Vauquois triangle helps visualize these approaches.

The Vauquois triangle



The Vauquois triangle



The differences in the three approaches:

- increasing amount of analysis / generation
- decreasing amount of transfer

Example sentences

We shall examine the workings of the various algorithms on the following example sentences.

(1) *Maria no dió una bofetada a la bruja verde*
Mary not gave a slap to the witch green
'Mary didn't slap the green witch'

(2) *Diese Woche ist die grüne Hexe zu Hause*
This week is the green witch at/to house
'The green witch is at home this week'

(3) 彼(かれ)は音楽(おんがく)を聞(き)く のが大好(だいす)きです。
He music listen -ing adore
'He adores listening to music'

Direct translation

In **direct translation**, the source language text is processed word-by-word, and each word is translated separately via a bilingual dictionary.

Direct translation

In **direct translation**, the source language text is processed word-by-word, and each word is translated separately via a bilingual dictionary. This simple process is augmented with two auxiliary steps:

- Morphological analysis: to extract syntactic/semantic content in bound morphemes
- Word reordering: to account for languages with different word order

Direct translation

In **direct translation**, the source language text is processed word-by-word, and each word is translated separately via a bilingual dictionary. This simple process is augmented with two auxiliary steps:

- Morphological analysis: to extract syntactic/semantic content in bound morphemes
- Word reordering: to account for languages with different word order

A schema of direct translation:

1. Morphological analysis
2. Lexical transfer (word-by-word) using a dictionary
3. Word reordering
4. Morphological generation

An example

Input	Mary didn't slap the green witch

Observations:

An example

	Input	Mary didn't slap the green witch
1.	Morph. analysis	Mary, do-PAST, not, slap, the, green, witch

Observations:

- Actually, we would also need POS tagging to know that *slap* is a verb

An example

	Input	Mary didn't slap the green witch
1.	Morph. analysis	Mary, do-PAST, not, slap, the, green, witch
2.	Lexical transfer	Maria, PAST, no, dar una bofetada a, la, verde, bruja

Observations:

- Actually, we would also need POS tagging to know that *slap* is a verb
- Spanish negates verbs with *no* + *verb*, so *do* can be deleted

An example

	Input	Mary didn't slap the green witch
1.	Morph. analysis	Mary, do-PAST, not, slap, the, green, witch
2.	Lexical transfer	Maria, PAST, no, dar una bofetada a, la, verde, bruja

Observations:

- Actually, we would also need POS tagging to know that *slap* is a verb
- Spanish negates verbs with *no + verb*, so *do* can be deleted
- We presume the dictionary includes *slap (V): dar una bofetada*

An example

	Input	Mary didn't slap the green witch
1.	Morph. analysis	Mary, do-PAST, not, slap, the, green, witch
2.	Lexical transfer	Maria, PAST, no, dar una bofetada a, la, verde, bruja
3.	Reordering	Maria, no, dar PAST, una bofetada a, la, bruja, verde

Observations:

- Actually, we would also need POS tagging to know that *slap* is a verb
- Spanish negates verbs with *no + verb*, so *do* can be deleted
- We presume the dictionary includes *slap (V): dar una bofetada*
- We only need *local* reorderings here

An example

	Input	Mary didn't slap the green witch
1.	Morph. analysis	Mary, do-PAST, not, slap, the, green, witch
2.	Lexical transfer	Maria, PAST, no, dar una bofetada a, la, verde, bruja
3.	Reordering	Maria, no, dar PAST, una bofetada a, la, bruja, verde
4.	Morph. generation	Maria no dió una bofetada a la bruja verda

Observations:

- Actually, we would also need POS tagging to know that *slap* is a verb
- Spanish negates verbs with *no + verb*, so *do* can be deleted
- We presume the dictionary includes *slap (V): dar una bofetada*
- We only need *local* reorderings here

Problems with direct translation

1. Word-by-word translation
2. Word-by-word reordering

- Words' meaning depend on context
- A single word usually has several translations
- Even more confusion with homonymous and polysemous words

Problems with direct translation

1. Word-by-word translation
2. Word-by-word reordering

… worked in our example:

Maria PAST no dar una bofetada a la verde bruja

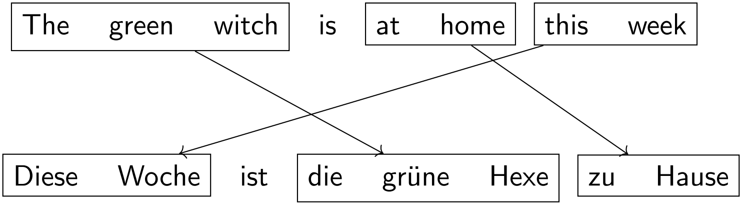
Maria no dar PAST una bofetada a la bruja verde

… because we only needed to reorder single words.

Problems with direct translation

1. Word-by-word translation
2. Word-by-word reordering

... does not suffice for the second example sentence:



Here we would need to reorder *phrases*, which is impossible when our units are words.

Transfer

Transfer approaches exploit the systematic structural differences between the source and target languages.

Transfer

Transfer approaches exploit the systematic structural differences between the source and target languages.

These systems are positioned at the middle of the Vauquois triangle; i.e. the schema looks like:

1. **Analysis**: parsing the source language text into morphological, syntactic, (shallow) semantic structures
2. **Transfer**: transform these structures based on the structural differences between the source and target languages
3. **Generation**: generate the target language text based on the target language structures

Transfer

Transfer approaches exploit the systematic structural differences between the source and target languages.

Levels on which transfer (can) take place:

- Syntactic transfer
- Semantic transfer
- Lexical transfer

Syntactic transfer

Maps between source and target language tree or dependency structures:

Languages	Difference	Rule		
en $\Rightarrow$ es	NP word order	Nom $\rightarrow$ Adj Noun	$\Rightarrow$	Nom $\rightarrow$ Noun Adj
en $\Rightarrow$ ja	SVO / SOV	VP $\rightarrow$ V NP	$\Rightarrow$	VP $\rightarrow$ NP V
en $\Rightarrow$ ja	Pre / postpositions	PP $\rightarrow$ P NP	$\Rightarrow$	PP $\rightarrow$ NP P

Syntactic transfer

Maps between source and target language tree or dependency structures:

Languages	Difference	Rule		
en $\Rightarrow$ es	NP word order	`Nom` $\rightarrow$ `Adj Noun`	$\Rightarrow$	`Nom` $\rightarrow$ `Noun Adj`
en $\Rightarrow$ ja	SVO / SOV	`VP` $\rightarrow$ `V NP`	$\Rightarrow$	`VP` $\rightarrow$ `NP V`
en $\Rightarrow$ ja	Pre / postpositions	`PP` $\rightarrow$ `P NP`	$\Rightarrow$	`PP` $\rightarrow$ `NP P`

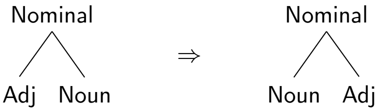


Table 1: NP word order

Syntactic transfer

Maps between source and target language tree or dependency structures:

Languages	Difference	Rule		
en ⇒ es	NP word order	`Nom` → `Adj Noun`	⇒	`Nom` → `Noun Adj`
en ⇒ ja	SVO / SOV	`VP` → `V NP`	⇒	`VP` → `NP V`
en ⇒ ja	Pre / postpositions	`PP` → `P NP`	⇒	`PP` → `NP P`

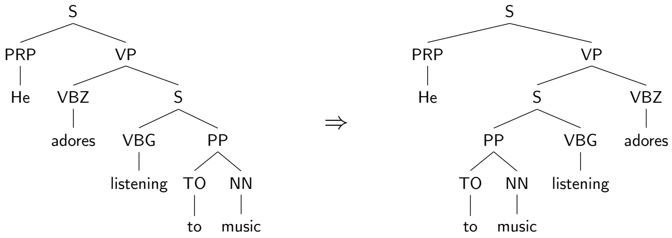


Table 1: SVO / SOV

Syntactic transfer

Maps between source and target language tree or dependency structures:

Languages	Difference	Rule		
en ⇒ es	NP word order	`Nom → Adj Noun`	⇒	`Nom → Noun Adj`
en ⇒ ja	SVO / SOV	`VP → V NP`	⇒	`VP → NP V`
en ⇒ ja	Pre / postpositions	`PP → P NP`	⇒	`PP → NP P`

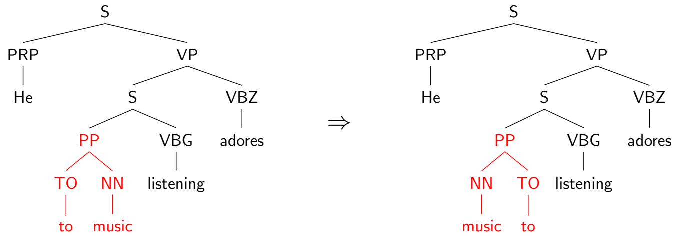


Table 1: Pre / postpositions

Semantic transfer

Syntactic transfer rules handle the systematic differences between the grammar of the two languages. Rules are usually written in terms of nonterminals.

Semantic transfer rules take into account semantic properties, e.g.

- Subcategorization: what argument types a verb (and some other words) accepts
- Lexicalization: some words impose constraints on the tree (e.g. a *ship* is always female in English)

Semantic transfer

An example on subcategorization:

Languages	Verb	Frames
`en` $\Rightarrow$ `ja`	listen / 聞(き)く (*kiku*)	`OBL(TO)` $\Rightarrow$ `ACC`

Semantic transfer

An example on subcategorization:

Languages	Verb	Frames
`en` ⇒ `ja`	listen / 聞く(き) (*kiku*)	`OBL(TO)` ⇒ `ACC`

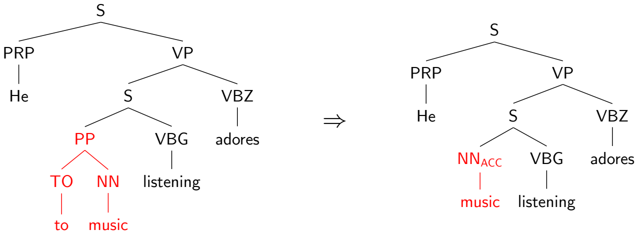


Table 2: listen frame: `OBL(TO)` to `ACC`

Interlingua

Interlingua approaches use a language-agnostic *abstract meaning representation* (*AMR*).

1. Analyze input into the AMR using a source language pipeline
2. No transfer is involved.
3. Generate the output from AMR using target language tools

Interlingua

Interlingua approaches use a language-agnostic *abstract meaning representation* (*AMR*).

1. Analyze input into the AMR using a source language pipeline
2. No transfer is involved.
3. Generate the output from AMR using target language tools

AMRs can be of any type of conceptual representation:

- First (second) Order Logic
- Other kinds of logic (temporal, modal, ...)
- LFG f-structures
- etc.

Interlingua: examples

Representations for "*Mary didn't slap the green witch*".

First Order Logic:

Sentence: $\neg slap(Mary, GreenWitch)$

Interlingua: examples

Representations for "*Mary didn't slap the green witch*".

First Order Logic:

$$\begin{array}{rl} \text{Sentence:} & \neg slap(Mary, GreenWitch) \\ \text{green witch:} & (\forall x)(GreenWitch(x) \rightarrow green(x) \wedge witch(x)) \end{array}$$

Interlingua: examples

Representations for "*Mary didn't slap the green witch*".

First Order Logic:

Sentence:	$\neg slap(Mary, GreenWitch)$
green witch:	$(\forall x)(GreenWitch(x) \rightarrow green(x) \wedge witch(x))$
green:	$(\forall x)(green(x) \rightarrow (\exists r \exists g \exists b)(hasRGB(x, r, g, b) \wedge r \leq ...))$

Interlingua: examples

Representations for "*Mary didn't slap the green witch*".

First Order Logic:

Sentence:	$\neg slap(Mary, GreenWitch)$
green witch:	$(\forall x)(GreenWitch(x) \rightarrow green(x) \wedge witch(x))$
green:	$(\forall x)(green(x) \rightarrow (\exists r \exists g \exists b)(hasRGB(x, r, g, b) \wedge r \leq ...))$
slap:	$(\forall x \forall y)(slap(x, y) \rightarrow (\exists z)(hit(x, z) \wedge face(z) \wedge has(y, z) \wedge ...)$

Interlingua: examples

Representations for "*Mary* **did***n't slap the green witch*".

First Order Logic:

Sentence (past):	$(\exists t)(\mathbf{at(t,} \neg slap(Mary, GreenWitch)\mathbf{)})$
green witch:	$(\forall x)(GreenWitch(x) \rightarrow green(x) \wedge witch(x))$
green:	$(\forall x)(green(x) \rightarrow (\exists r \exists g \exists b)(hasRGB(x, r, g, b) \wedge r \leq ...))$
slap:	$(\forall x \forall y)(slap(x, y) \rightarrow (\exists z)(hit(x, z) \wedge face(z) \wedge has(y, z) \wedge ...)$

Interlingua: examples

Representations for "*Mary didn't slap the green witch*".

LFG f-structure:

$$\begin{bmatrix} \text{event} & \text{slapping} \\ \text{tense} & \text{past} \\ \text{polarity} & \text{negative} \\ \text{agent} & \text{Mary} \\ \text{theme} & \begin{bmatrix} \text{witch} & \\ \text{definiteness} & \text{def} \\ \text{attributes} & \begin{bmatrix} \text{has-color} & \text{green} \end{bmatrix} \end{bmatrix} \end{bmatrix}$$

Interlingua: pros and cons

Advantages:

- No transfer stage is involved: no need for e.g. bilingual dictionaries
- Direct translation and transfer approaches require resources for each **language pair**; interlingua needs one resource chain per language
- For n languages, transfer requires n^2 systems, interlingua n

Interlingua: pros and cons

Advantages:

- No transfer stage is involved: no need for e.g. bilingual dictionaries
- Direct translation and transfer approaches require resources for each **language pair**; interlingua needs one resource chain per language
- For n languages, transfer requires n^2 systems, interlingua n

Disadvantages:

- Defining an AMR is very difficult
- Full conceptual analysis / generation is hard
- All concepts from all languages need to be present
  - $Brother$ is not enough, need $OlderBrother$ and $YoungerBrother$
  - Colors: $Blue \not\approx JapaneseBlue$,
  - Dragons: $WesternDragon \not\approx EasterDragon \not\approx HungarianDragon$

Interlingua: pros and cons

Advantages:

- No transfer stage is involved: no need for e.g. bilingual dictionaries
- Direct translation and transfer approaches require resources for each **language pair**; interlingua needs one resource chain per language
- For n languages, transfer requires n^2 systems, interlingua n

Disadvantages:

- Defining an AMR is very difficult
- Full conceptual analysis / generation is hard
- All concepts from all languages need to be present
  - $Brother$ is not enough, need $OlderBrother$ and $YoungerBrother$
  - Colors: $Blue \not\approx JapaneseBlue$,
  - Dragons: $WesternDragon \not\approx EasterDragon \not\approx HungarianDragon$

Due to these problems, interlingual translation is used only in sublanguage domains.

Statistical Machine Translation

Statistical Machine Translation

The classical model of translation depended on hand-crafted rules and resources, such as dictionaries. It had several problems:

- It needs translation-specific resources (transfer rules, AMR)
- Creating these requires a huge amount of work
- No guidance on how to choose between available rules

Statistical Machine Translation

The classical model of translation depended on hand-crafted rules and resources, such as dictionaries. It had several problems:

- It needs translation-specific resources (transfer rules, AMR)
- Creating these requires a huge amount of work
- No guidance on how to choose between available rules

Statistical MT learns a probabilistic model and tries to find the *most probable translation*. Mathematically,

- given the foreign language sentence $F = f_1, f_2, ..., f_m$
- we are looking for the best English sentence $\hat{E} = e_1, e_2, ..., e_l$:

$$\hat{E} = \operatorname*{argmax}_{E} P(E|F)$$

Statistical Machine Translation

There are two approaches for finding $\hat{E}$:

1. **Direct approach**: models $P(E|F)$ directly.
   - No way to model the quality of the resulting E sentence
   - Not really used in practice
2. **Noisy channel model**: a paradigm borrowed from information theory

Statistical Machine Translation

There are two approaches for finding $\hat{E}$:

1. **Direct approach**: models $P(E|F)$ directly.
   - No way to model the quality of the resulting E sentence
   - Not really used in practice
2. **Noisy channel model**: a paradigm borrowed from information theory

As with statistical methods generally, statistical MT has two phases:

1. **Training**: learning the probabilistic model from training data
2. **Decoding**: using the trained system to translate a sentence

Noisy Channel Model

Intuition for translating from Spanish to English:

1. we are talking with someone in English
2. the channel between us is noisy, and everything comes out of it in Spanish
3. *translation* is the task of restoring the original signal.

Noisy Channel Model

Intuition for translating from Spanish to English:

1. we are talking with someone in English
2. the channel between us is noisy, and everything comes out of it in Spanish
3. *translation* is the task of restoring the original signal.

Mathematically, given the foreign language sentence $F = f_1, f_2, ..., f_m$, we are looking for the best English sentence $\hat{E} = e_1, e_2, ..., e_l$:

$$\begin{aligned} \hat{E} &= \operatorname*{argmax}_E P(E|F) && \textit{Use Bayes' rule} \\ &= \operatorname*{argmax}_E \frac{P(F|E)P(E)}{P(F)} && P(F) \textit{ is constant} \\ &= \operatorname*{argmax}_E P(F|E)P(E) \end{aligned}$$

Noisy Channel Model: cont.

From the equation

$$\hat{E} = \operatorname*{argmax}_E P(F|E)P(E),$$

we can see we need to model components:

- $P(F|E)$ is the (backward) **translation model**
- $P(E)$ is the (English) **language model**

Noisy Channel Model: cont.

From the equation

$$\hat{E} = \operatorname*{argmax}_{E} P(F|E)P(E),$$

we can see we need to model components:

- $P(F|E)$ is the (backward) **translation model**
- $P(E)$ is the (English) **language model**

These two components correspond to two important properties of a good translation:

- **Fidelity**: how faithful the translation is to the source language and content
- **Fluency**: how natural the resulting sentence in the target language

The language model

A (statistical) **language model** is a probability distribution over a sequence of words. Given $S = w_1 w_2 ... w_N$, we want to estimate

$$P(S) = P(w_1 w_2 ... w_N)$$

The language model

A (statistical) **language model** is a probability distribution over a sequence of words. Given $S = w_1 w_2 ... w_N$, we want to estimate

$$P(S) = P(w_1 w_2 ... w_N)$$

Since this is too difficult in the general case, usually the solution is sought in the format of

$$P(w_1 w_2 ... w_N) = \prod_{i=1}^{N} P(w_i | C_i)$$

, where C is a *context*.

The language model: N-grams and RNNs

The most popular context is the history:

$$P(w_1 w_2 ... w_N) = \prod_{i=1}^{N} P(w_i | H_i) = \prod_{i=1}^{N} P(w_i | w_1^{i-1})$$

This is a *generative context*: we can use it to generate the text on the fly.

The language model: N-grams and RNNs

The most popular context is the history:

$$P(w_1 w_2 ... w_N) = \prod_{i=1}^{N} P(w_i | H_i) = \prod_{i=1}^{N} P(w_i | w_1^{i-1})$$

This is a *generative context*: we can use it to generate the text on the fly.

The two main generative models are:

1. **n-gram models**: for an n-gram model of order n, the history consists of the last $n-1$ words ($H_i = w_{i-n+1}^{i-1}$); e.g. a 4-gram model predicts the 4th word based on the first three.
2. **Recurrent Neural Networks (RNNs)** encode the whole history in their state.

The translation model

Over the time, many translation models have been proposed. Here we introduce two of them.

IBM Model 1 The simplest of the five models published in Brown et al. (1993)

Phrase-based translation model (e.g. Koehn et al. (2003))

The translation model

Over the time, many translation models have been proposed. Here we introduce two of them.

IBM Model 1
: The simplest of the five models published in Brown et al. (1993)

Phrase-based
: translation model (e.g. Koehn et al. (2003))

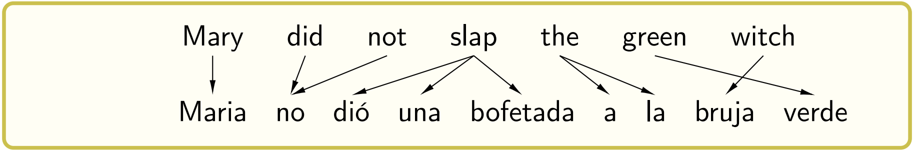
Most statistical translation systems are based on the idea of **word alignment**; though how they use it varies from model to model.

Alignment

Statistical translation models are based on **word alignment**: a mapping between the words of the source and target sentences.

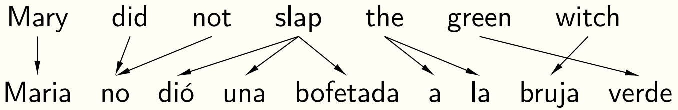
Alignment

Statistical translation models are based on **word alignment**: a mapping between the words of the source and target sentences.



Alignment

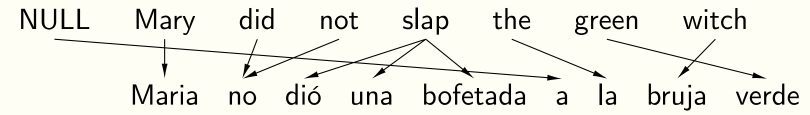
Statistical translation models are based on **word alignment**: a mapping between the words of the source and target sentences.



- Alignments can be $1:n$, $n:1$ or $n:n$ (phrasal alignment)
- IBM 1 only handles $1:n$

Alignment

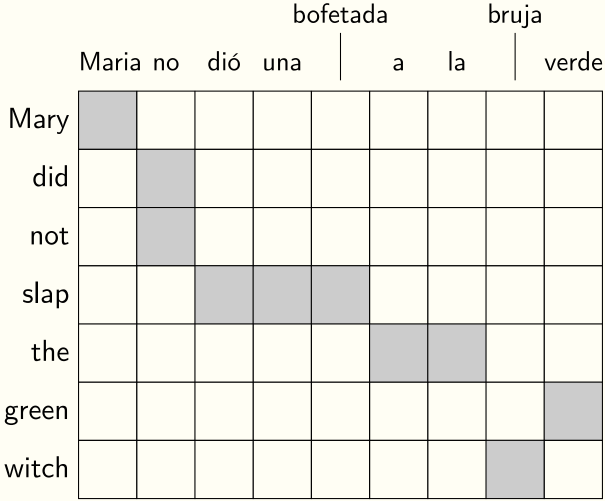
Statistical translation models are based on **word alignment**: a mapping between the words of the source and target sentences.



- Alignments can be $1:n$, $n:1$ or $n:n$ (phrasal alignment)
- IBM 1 only handles $1:n$
- The NULL word can model the appearance of *spurious* words in the output

Alignment

Another way to represent or visualize alignments is the matrix format:



IBM Model 1

IBM Model 1 uses

- word alignment directly as the translation model:

$$P(F|E) = \sum_{A} P(F, A|E)$$

- only word-word translation probabilities: $P(f_i|e_j)$
  (the probability that the j^{th} English word translates to the i^{th} foreign word)

IBM Model 1

IBM Model 1 uses

- word alignment directly as the translation model:

$$P(F|E) = \sum_A P(F, A|E)$$

- only word-word translation probabilities: $P(f_i|e_j)$
  (the probability that the j^{th} English word translates to the i^{th} foreign word)

The only trainable part of the model is the *translation table*, which stores the word-word translation probabilities. The alignment probabilities are assumed to be uniform.

IBM Model 1 – in more detail

IBM Model 1 uses word alignment directly as the translation model:

$$P(F|E) = \sum_{A} P(F, A|E)$$

IBM Model 1 – in more detail

IBM Model 1 uses word alignment directly as the translation model:

$$P(F|E) = \sum_{A} P(F, A|E)$$

More specifically, the source sentence $E = e_1, e_2, ..., e_J$ of length J is translated as follows:

NULL | Mary | did | not | slap | the | green | witch

IBM Model 1 – in more detail

IBM Model 1 uses word alignment directly as the translation model:

$$P(F|E) = \sum_{A} P(F, A|E)$$

More specifically, the source sentence $E = e_1, e_2, ..., e_J$ of length J is translated as follows:

1. Choose a length K for the Spanish sentence: $F = f_1, f_2, ..., f_K$

NULL	Mary	did	not	slap	the	green	witch

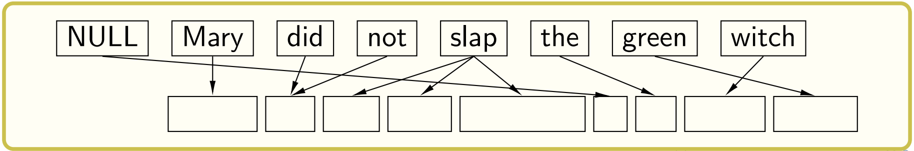
IBM Model 1 – in more detail

IBM Model 1 uses word alignment directly as the translation model:

$$P(F|E) = \sum_{A} P(F, A|E)$$

More specifically, the source sentence $E = e_1, e_2, ..., e_J$ of length J is translated as follows:

1. Choose a length K for the Spanish sentence: $F = f_1, f_2, ..., f_K$
2. Choose a $1 : n$ alignment between E and F: $A = a_1, a_2, ..., a_K$



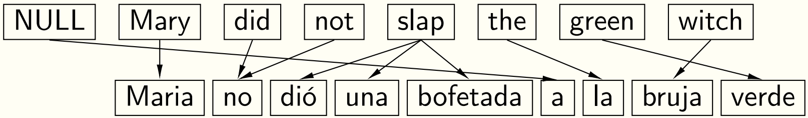
IBM Model 1 – in more detail

IBM Model 1 uses word alignment directly as the translation model:

$$P(F|E) = \sum_{A} P(F, A|E)$$

More specifically, the source sentence $E = e_1, e_2, ..., e_J$ of length J is translated as follows:

1. Choose a length K for the Spanish sentence: $F = f_1, f_2, ..., f_K$
2. Choose a $1 : n$ alignment between E and F: $A = a_1, a_2, ..., a_K$
3. Fill each position k in F by the translation of the English word that is aligned to it



IBM Model 1 – in yet more detail

1. Choose a length K for the Spanish sentence: $F = f_1, f_2, ..., f_K$.
2. Choose a $1 : n$ alignment between E and F: $A = a_1, a_2, ..., a_K$.
3. Fill each position k in F by the translation of the English word that is aligned to it.

IBM Model 1 – in yet more detail

1. Choose a length K for the Spanish sentence: $F = f_1, f_2, ..., f_K$.

   K is chosen with the small constant probability ϵ.
2. Choose a $1:n$ alignment between E and F: $A = a_1, a_2, ..., a_K$.
3. Fill each position k in F by the translation of the English word that is aligned to it.

IBM Model 1 – in yet more detail

1. Choose a length K for the Spanish sentence: $F = f_1, f_2, ..., f_K$.
2. Choose a $1 : n$ alignment between E and F: $A = a_1, a_2, ..., a_K$.

   Each alignment is chosen with a uniform probability. We can think of selecting one alignment as a task to color one cell in each column of the aligment matrix. How many colorations are there?

3. Fill each position k in F by the translation of the English word that is aligned to it.

IBM Model 1 – in yet more detail

1. Choose a length K for the Spanish sentence: $F = f_1, f_2, ..., f_K$.
2. Choose a $1 : n$ alignment between E and F: $A = a_1, a_2, ..., a_K$.

   Each alignment is chosen with a uniform probability. We can think of selecting one alignment as a task to color one cell in each column of the aligment matrix. How many colorations are there?
   - There are $J + 1$ rows (*NONE* included)
3. Fill each position k in F by the translation of the English word that is aligned to it.

IBM Model 1 – in yet more detail

1. Choose a length K for the Spanish sentence: $F = f_1, f_2, ..., f_K$.
2. Choose a $1 : n$ alignment between E and F: $A = a_1, a_2, ..., a_K$.

   Each alignment is chosen with a uniform probability. We can think of selecting one alignment as a task to color one cell in each column of the aligment matrix. How many colorations are there?
   - There are $J + 1$ rows (*NONE* included)
   - K columns
3. Fill each position k in F by the translation of the English word that is aligned to it.

IBM Model 1 – in yet more detail

1. Choose a length K for the Spanish sentence: $F = f_1, f_2, ..., f_K$.
2. Choose a $1 : n$ alignment between E and F: $A = a_1, a_2, ..., a_K$.

   Each alignment is chosen with a uniform probability. We can think of selecting one alignment as a task to color one cell in each column of the aligment matrix. How many colorations are there?

   - There are $J + 1$ rows (*NONE* included)
   - K columns
   - This is K-ary Cartesian power:

3. Fill each position k in F by the translation of the English word that is aligned to it.

IBM Model 1 – in yet more detail

1. Choose a length K for the Spanish sentence: $F = f_1, f_2, ..., f_K$.
2. Choose a $1 : n$ alignment between E and F: $A = a_1, a_2, ..., a_K$.

   Each alignment is chosen with a uniform probability. We can think of selecting one alignment as a task to color one cell in each column of the aligment matrix. How many colorations are there?
   - There are $J + 1$ rows (*NONE* included)
   - K columns
   - This is K-ary Cartesian power:

   $$P(A|E) = \frac{\epsilon}{(J+1)^K}$$

3. Fill each position k in F by the translation of the English word that is aligned to it.

IBM Model 1 – in yet more detail

1. Choose a length K for the Spanish sentence: $F = f_1, f_2, ..., f_K$.
2. Choose a $1 : n$ alignment between E and F: $A = a_1, a_2, ..., a_K$.
3. Fill each position k in F by the translation of the English word that is aligned to it.

Given A, let

IBM Model 1 – in yet more detail

1. Choose a length K for the Spanish sentence: $F = f_1, f_2, ..., f_K$.
2. Choose a $1 : n$ alignment between E and F: $A = a_1, a_2, ..., a_K$.
3. Fill each position k in F by the translation of the English word that is aligned to it.

Given A, let

- e_{a_k} the English word that is aligned to f_k

IBM Model 1 – in yet more detail

1. Choose a length K for the Spanish sentence: $F = f_1, f_2, ..., f_K$.
2. Choose a $1 : n$ alignment between E and F: $A = a_1, a_2, ..., a_K$.
3. Fill each position k in F by the translation of the English word that is aligned to it.

Given A, let

- e_{a_k} the English word that is aligned to f_k
- $P(f_x|e_y)$ the probability of translating e_y to f_x

IBM Model 1 – in yet more detail

1. Choose a length K for the Spanish sentence: $F = f_1, f_2, ..., f_K$.
2. Choose a $1 : n$ alignment between E and F: $A = a_1, a_2, ..., a_K$.
3. Fill each position k in F by the translation of the English word that is aligned to it.

Given A, let

- e_{a_k} the English word that is aligned to f_k
- $P(f_x|e_y)$ the probability of translating e_y to f_x

The probability of the Spanish sentence is thus:

$$P(F|E, A) = \prod_{k=1}^{K} P(f_k|e_{a_k})$$

IBM Model 1 – put it all together

The probability of a Spanish sentence through alignment A is:

$$
\begin{aligned}
P(F, A|E) &= P(F|E, A) \times P(A|E) \\
&= \frac{\epsilon}{(J+1)^K} \prod_{k=1}^{K} P(f_k|e_{a_k})
\end{aligned}
$$

IBM Model 1 – put it all together

The probability of a Spanish sentence through alignment A is:

$$\begin{aligned} P(F, A|E) &= P(F|E, A) \times P(A|E) \\ &= \frac{\epsilon}{(J+1)^K} \prod_{k=1}^{K} P(f_k|e_{a_k}) \end{aligned}$$

To get the probability of the translation, we sum over all alignments:

$$\boxed{\begin{aligned} P(F|E) &= \sum_A P(F, A|E) \\ &= \sum_A \frac{\epsilon}{(J+1)^K} \prod_{k=1}^{K} P(f_k|e_{a_k}) \end{aligned}}$$

Phrase-based translation

In phrase-based models, the unit of translation is the phrase. The main steps are:

1. Group the source words into I phrases: $E = \bar{e}_1, \bar{e}_2, ..., \bar{e}_I$
2. Translate the phrases into the target language phrases one-by-one: each $\bar{e}_i$ to $\bar{f}_i$
3. Reorder the phrases.

Very similar to direct translation, only with phrases instead of words.

Phrase-based translation

In phrase-based models, the unit of translation is the phrase. The main steps are:

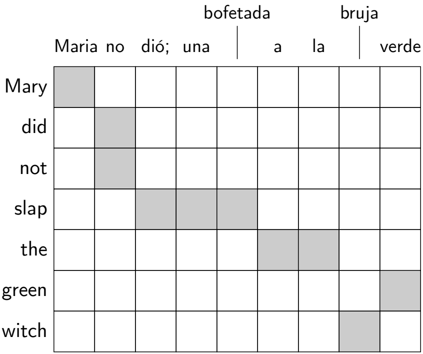
1. Group the source words into I phrases: $E = \bar{e}_1, \bar{e}_2, ..., \bar{e}_I$
2. Translate the phrases into the target language phrases one-by-one: each $\bar{e}_i$ to $\bar{f}_i$
3. Reorder the phrases.

Very similar to direct translation, only with phrases instead of words.

Two important subtasks during training:

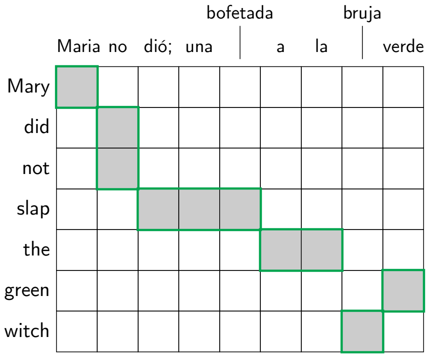
- Finding the phrases $\bar{e}, \bar{f}$ using word alignments
- Building a **phrase translation table**: it stores the probability of each $\bar{e} \rightarrow \bar{f}$ translation

Phrase acquisition using word alignments



A consistent phrase pair is where the words align only with each other, and no external words.

Phrase acquisition using word alignments



(Maria, Mary)

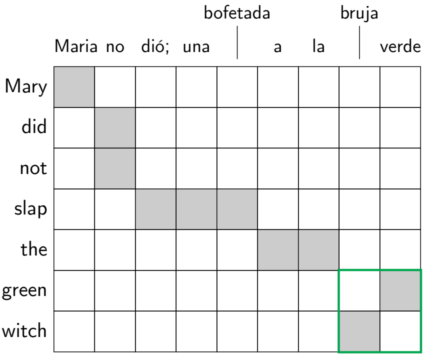
(no, did not)

...

(witch, bruja)

A consistent phrase pair is where the words align only with each other, and no external words.

Phrase acquisition using word alignments



(Maria, Mary)

(no, did not)

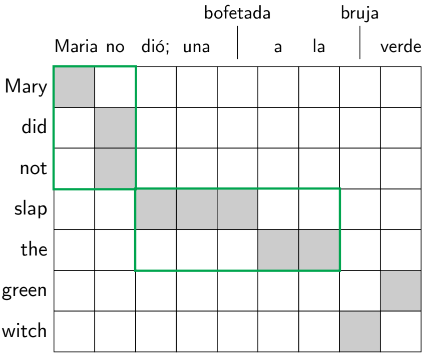
...

(witch, bruja)

(green witch, bruja verde)

A consistent phrase pair is where the words align only with each other, and no external words.

Phrase acquisition using word alignments

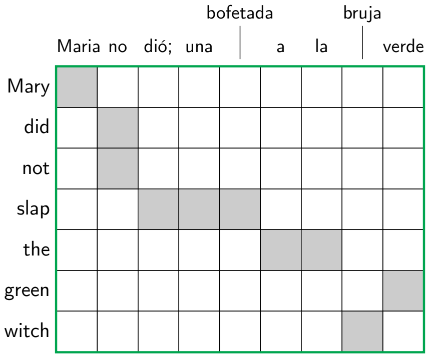


(Maria, Mary)
(no, did not)
...
(witch, bruja)
(green witch, bruja verde)
(Mary did not, Maria no)
(slap the, dió una bofetada a la)

A consistent phrase pair is where the words align only with each other, and no external words.

- Not necessarily phrases in the syntactic sense

Phrase acquisition using word alignments



(Maria, Mary)
(no, did not)
...
(witch, bruja)
(green witch, bruja verde)
(Mary did not, Maria no)
(slap the, dió una bofetada a la)
(Mary did not slap the green witch, Maria no dió una bofetada a la bruja verde)

A consistent phrase pair is where the words align only with each other, and no external words.

- Not necessarily phrases in the syntactic sense
- Less frequent / accidental phrases can be discarded

Training

Training is the process of learning the probabilistic model from training data. MT models are trained on **parallel corpora**.

- Parliamentary proceedings (e.g. Canada, EU, Hong Kong)
- Literature
- Software documentation
- Movie subtitles

Training

Training is the process of learning the probabilistic model from training data. MT models are trained on **parallel corpora**.

- Parliamentary proceedings (e.g. Canada, EU, Hong Kong)
- Literature
- Software documentation
- Movie subtitles

The sentences in the parallel texts are aligned using **sentence aligner** programs. Features used:

- sentence length
- bilingual dictionary

Training

Training is the process of learning the probabilistic model from training data. MT models are trained on **parallel corpora**.

- Parliamentary proceedings (e.g. Canada, EU, Hong Kong)
- Literature
- Software documentation
- Movie subtitles

The sentences in the parallel texts are aligned using **sentence aligner** programs. Features used:

- sentence length
- bilingual dictionary

There is **no** gold standard data for machine translation!

Decoding

Decoding is the act of using the trained system to actually translate a sentence.

Decoding

Decoding is the act of using the trained system to actually translate a sentence.

In our case, we take the foreign sentence F and aim to return the best translation according to

$$\hat{E} = \operatorname*{argmax}_{E} P(E|F).$$

Decoding

Decoding is the act of using the trained system to actually translate a sentence.

In our case, we take the foreign sentence F and aim to return the best translation according to

$$\hat{E} = \underset{E}{\text{argmax}}\, P(E|F).$$

The task is to find $\hat{E}$ (implement argmax) without enumerating all possible translations.

Why not enumerate?

Let's see how many possible translations does a sentence have.
For IBM Model 1, we have the following parameters:

- a number K of sentence lengths to try: k
- a number of alignments for a k is $(J+1)^k$
- a word can have n translations on average

We get about[1] $\sum_{k=1}^{K}(J+1)^k n^J$ possible translations.

[1] Of course, the formula is not exact because of $1:n$ alignments.

Why not enumerate?

Let's see how many possible translations does a sentence have.
For IBM Model 1, we have the following parameters:

- a number K of sentence lengths to try: k
- a number of alignments for a k is $(J+1)^k$
- a word can have n translations on average

We get about[1] $\sum_{k=1}^{K}(J+1)^k n^J$ possible translations.

For an English sentence of length $J = 15$, $k \in \{12, 18\}$, and $n = 2$ we get

[1] Of course, the formula is not exact because of $1:n$ alignments.

Why not enumerate?

Let's see how many possible translations does a sentence have.
For IBM Model 1, we have the following parameters:

- a number K of sentence lengths to try: k
- a number of alignments for a k is $(J+1)^k$
- a word can have n translations on average

We get about[1] $\sum_{k=1}^{K}(J+1)^k n^J$ possible translations.

For an English sentence of length $J=15$, $k \in \{12, 18\}$, and $n=2$ we get 165,058,671,289,825,900,862,898,176.

[1] Of course, the formula is not exact because of $1:n$ alignments.

Decoding as search

Decoding is usually implemented as a search problem in the space of translations:

- the translation is built incrementally
  - translate a single word at a time
  - draw a single alignment arrow
- we execute the next step according to some strategy

Decoding as search

Decoding is usually implemented as a search problem in the space of translations:

- the translation is built incrementally
  - translate a single word at a time
  - draw a single alignment arrow
- we execute the next step according to some strategy

There are several strategies we can use to select the next step:

Greedy search Executes the step with the highest trained probability

Beam search Keeps the most probable n paths and always continues the top one

Best-first search Like beam search, but also uses heuristics beside the trained probability

MT Evaluation

Evaluating machine translation is highly subjective: any sentence can have multiple 'good'translations. As such, there is no undisputed method of choice for evaluation.

MT Evaluation

Evaluating machine translation is highly subjective: any sentence can have multiple ‘good’translations. As such, there is no undisputed method of choice for evaluation.

There are two main approaches for rating translations:

1. Manual evaluation
2. Automatic evaluation

Manual evaluation

Manual evaluation relies on the judgement of human raters.

Manual evaluation

Manual evaluation relies on the judgement of human raters.

What to test?

- Fidelity:
- Fluency:

Manual evaluation

Manual evaluation relies on the judgement of human raters.

What to test?

- Fidelity:
  - *Adequacy*: whether the translation contains the information in the source sentence
  - *Informativeness*: is the information in the translation enough to complete a task; e.g. answer certain questions about the text
- Fluency:

Manual evaluation

Manual evaluation relies on the judgement of human raters.

What to test?

- Fidelity:
  - *Adequacy*: whether the translation contains the information in the source sentence
  - *Informativeness*: is the information in the translation enough to complete a task; e.g. answer certain questions about the text
- Fluency:
  - *Clarity*
  - *Naturalness*
  - *Style*

Manual evaluation

How to test?

- *Ratings*: rate aspects of the translation e.g. on a 5-point scale
- *Psycholinguistic tasks*:
  - The *cloze task*: some words are replaced by an underscore and raters must guess it. Correlates with *fluency*.
  - *Multi-choice questions*: good for evaluating *informativeness*.
- *Edit cost*: how much effort it takes to convert the MT output into a good translation.

Automatic evaluation

Manual rating is expensive, so most of the time we rely on automatic evalutation. The most widespread measure is the **BLEU metric** (**Bi**lingual **E**valuation **U**nderstudy) (Papineni et al., 2002):

- Scores a sentence translation *candidate* based on several *reference translations*
- Scores range from 0 to 1
- Similarly to F-score, it has two components:
  - **Precision**: ratio of overlapping n-grams between candidate and references
  - *Recall* is not usable with multiple references: **brevity penalty**
- Several orders of n-grams are used, up N (usually 4)
- The final score for the whole text is the micro average of the sentence scores

Precision

Precision example for $N = 1$: the ratio of overlapping unigrams.

Candidate	the	cat	sits	on	a	mat	
Reference 1	the	cat	sat	on	the	mat	
Reference 2	a	cat	was	sitting	on	the	mat

Precision

Precision example for $N = 1$: the ratio of overlapping unigrams.

Candidate	the	cat	sits	on	a	mat	
Reference 1	the	cat	sat	on	the	mat	
Reference 2	a	cat	was	sitting	on	the	mat

Precision

Precision example for $N = 1$: the ratio of overlapping unigrams.

Candidate	the	cat	sits	on	a	mat	
Reference 1	the	cat	sat	on	the	mat	
Reference 2	a	cat	was	sitting	on	the	mat

Precision

Precision example for $N = 1$: the ratio of overlapping unigrams.

Candidate	the	cat	sits	on	a	mat	
Reference 1	the	cat	sat	on	the	mat	
Reference 2	a	cat	was	sitting	on	the	mat

Precision

Precision example for $N = 1$: the ratio of overlapping unigrams.

Candidate	the	cat	sits	on	a	mat	
Reference 1	the	cat	sat	on	the	mat	
Reference 2	a	cat	was	sitting	on	the	mat

Precision

Precision example for $N = 1$: the ratio of overlapping unigrams.

Candidate	the	cat	sits	on	a	mat	
Reference 1	the	cat	sat	on	the	mat	
Reference 2	a	cat	was	sitting	on	the	mat

Precision

Precision example for $N = 1$: the ratio of overlapping unigrams.

Candidate	the	cat	sits	on	a	mat	
Reference 1	the	cat	sat	on	the	mat	
Reference 2	a	cat	was	sitting	on	the	mat

Precision

Precision example for $N = 1$: the ratio of overlapping unigrams.

Candidate	the	cat	sits	on	a	mat	
Reference 1	the	cat	sat	on	the	mat	
Reference 2	a	cat	was	sitting	on	the	mat

Precision is defined for a sentence as follows.

$$p_n = \frac{\sum\limits_{n\text{-}gram \in C \cap \{R\}} \text{Count}(n\text{-}gram)}{\sum\limits_{n\text{-}gram' \in C} \text{Count}(n\text{-}gram')}$$

Precision

Precision example for $N = 1$: the ratio of overlapping unigrams.

Candidate	the	cat	sits	on	a	mat	
Reference 1	the	cat	sat	on	the	mat	
Reference 2	a	cat	was	sitting	on	the	mat

Precision is defined for a sentence as follows. In this case,

$$p_n = \frac{\sum_{n\text{-}gram \in C \cap \{R\}} \mathsf{Count}(n\text{-}gram)}{\sum_{n\text{-}gram' \in C} \mathsf{Count}(n\text{-}gram')} = 5/6$$

Modified n-gram precision

N-gram ratio can be tricked (unigram example):

Candidate	$\boxed{\text{the}}$	$\boxed{\text{the}}$	$\boxed{\text{the}}$	$\boxed{\text{the}}$	$\boxed{\text{the}}$	$\boxed{\text{the}}$	$\boxed{\text{the}}$
Reference	$\boxed{\text{the}}$	cat	sat	on	$\boxed{\text{the}}$	mat	

Problem This "translation" gets a perfect score, since all of its 7 words are contained in a reference translation.

Modified n-gram precision

N-gram ratio can be tricked (unigram example):

Candidate	$\boxed{\text{the}}$	$\boxed{\text{the}}$	the	the	the	the	the
Reference	$\boxed{\text{the}}$	cat	sat	on	$\boxed{\text{the}}$	mat	

Solution Clip the n-gram count at the maximum reference value (here: 2). This modifies the precision to a much more modest $2/7$.

$$p_n = \frac{\sum_{n\text{-}gram \in C \cap \{R\}} \mathsf{Count}(n\text{-}gram)}{\sum_{n\text{-}gram' \in C} \mathsf{Count}(n\text{-}gram')}$$

The raw precision formula

Modified n-gram precision

N-gram ratio can be tricked (unigram example):

Candidate	$\boxed{\text{the}}$	$\boxed{\text{the}}$	the	the	the	the	the
Reference	$\boxed{\text{the}}$	cat	sat	on	$\boxed{\text{the}}$	mat	

Solution Clip the n-gram count at the maximum reference value (here: 2). This modifies the precision to a much more modest $2/7$.

$$p_n = \frac{\sum\limits_{n\text{-}gram \in C \cap \{R\}} \text{Count}_{\text{clip}}(n\text{-}gram)}{\sum\limits_{n\text{-}gram' \in C} \text{Count}(n\text{-}gram')}$$

Modified precision

Micro-averaged precision

There are two ways to average quality measures (which are already forms of averages): micro- and macro-average.

- A macro-average is just the average of averages
- BLEU uses *micro-average*:

One sentence:

$$p_n = \frac{\sum_{n\text{-}gram \in C \cap \{R\}} \mathsf{Count}_{\mathsf{clip}}(n\text{-}gram)}{\sum_{n\text{-}gram' \in C} \mathsf{Count}(n\text{-}gram')}$$

Micro-averaged precision

There are two ways to average quality measures (which are already forms of averages): micro- and macro-average.

- A macro-average is just the average of averages
- BLEU uses *micro-average*:

M sentences:

$$p_n = \frac{\sum_{i=1}^{M} \sum_{n\text{-}gram \in C_i \cap \{R_i\}} \text{Count}_{\text{clip}}(n\text{-}gram)}{\sum_{i=1}^{M} \sum_{n\text{-}gram' \in C_i} \text{Count}(n\text{-}gram')}$$

Brevity Penalty

Too short, incomplete translations pose another problem.

Candidate	$\boxed{\text{the}}$					
Reference	$\boxed{\text{the}}$	cat	sat	on	the	mat

Problem This "translation" gets a perfect score, since the only unigram it has is in the reference.

Brevity Penalty

Too short, incomplete translations pose another problem.

Candidate	the					
Reference	the	cat	sat	on	the	mat

Solution In lieu of *recall*, BLEU penalizes candidates shorter than the reference:

$$\text{BP} = \begin{cases} 1 & \text{if } c > r \\ e^{(1-r/c)} & \text{if } c <= r \end{cases},$$

where c is the length of C and r is the length of R.

BLEU

The final BLEU score for n-grams up to order N is then

$$\mathsf{BLEU} = \mathsf{BP} \cdot \left(\prod_{n=1}^{N} p_n \right)^{\frac{1}{N}}$$

BLEU

The final BLEU score for n-grams up to order N is then

$$\text{BLEU} = \text{BP} \cdot \left(\prod_{n=1}^{N} p_n \right)^{\frac{1}{N}}$$

Properties of BLEU:

- Supposedly correlate with human judgement
- In reality, it only considers local information, and misses global problems
- Performs poorly when comparing systems with radically different architectures
- Might be useful when evaluating incremental changes to a single system

Resources

- Machine translation systems
  - Google Translate
  - Apertium: an open source rule-based MT system (Forcada et al., 2011)
  - MOSES: an open source, statistical MT system (Koehn et al., 2007)
- Parallel corpora
  - Europarl: Proceedings of the European Parliament, 21 languages (Koehn, 2005)
  - Hunglish Corpus: a Hungarian–English bicorpus (Varga et al., 2007)
- Sentence aligners
  - hunalign: used to align Europarl (Varga et al., 2007)

Appendix: bibliography

Brown, Peter F, Vincent J Della Pietra, Stephen A Della Pietra, and Robert L Mercer (1993). "The mathematics of statistical machine translation: Parameter estimation". In: *Computational linguistics* 19.2, pp. 263–311.

Forcada, Mikel L, Mireia Ginestí-Rosell, Jacob Nordfalk, Jim O'Regan, Sergio Ortiz-Rojas, Juan Antonio Pérez-Ortiz, Felipe Sánchez-Martínez, Gema Ramírez-Sánchez, and Francis M Tyers (2011). "Apertium: a free/open-source platform for rule-based machine translation". In: *Machine translation* 25.2, pp. 127–144.

Koehn, P (2005). "Europarl: A Parallel Corpus for Statistical Machine Translation". In:

Appendix: bibliography

Koehn, Philipp, Franz Josef Och, and Daniel Marcu (2003). “Statistical phrase-based translation”. In: *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology-Volume 1*. Association for Computational Linguistics, pp. 48–54.

Koehn, Philipp et al. (2007). “Moses: Open source toolkit for statistical machine translation”. In: *Proceedings of the 45th annual meeting of the ACL*. Association for Computational Linguistics, pp. 177–180.

Papineni, Kishore, Salim Roukos, Todd Ward, and Wei-Jing Zhu (2002). “BLEU: A Method for Automatic Evaluation of Machine Translation”. In: *Proceedings of the 40th Annual Meeting of the association for Computational Linguistics*. Philadelphia, pp. 311–318.

Appendix: bibliography

Varga, Daniel, Peter Halacsy, Andras Kornai, Viktor Nagy, Laszlo Nemeth, and Viktor Tron (2007). "Parallel corpora for medium density languages". In: *Recent Advances in Natural Language Processing IV. Selected papers from RANLP-05*. Ed. by N Nicolov, K Bontcheva, G Angelova, and R Mitkov. Amsterdam: Benjamins, pp. 247–258.