Natural Language Processing II

Khalil Sima'an

Universiteit van Amsterdam

Natural Language Processing II

Why Machine Translation?

Main Questions and General Approach?

Vanilla Treasures of Machine Translation (MT)

Major interest by industry!

Vanilla Treasures of Machine Translation (MT)

Major interest by industry!

GET RICH TODAY: *Translate better than Google!*

Technological motivation

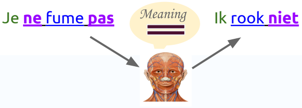
- Cultural, economic and societal impact
- Huge volume that never gets translated
- **MT is enabling**: Speed + Low cost

BUT why conduct research on MT (beside technology)?

Why Machine Translation (MT)?

Technological challenge…

Scientific challenge Human Language Understanding



We never observe "meaning" in the wild.
But translation Data has two crucial properties

- Human *meaning preserving behavior*: Meaning(I) == Meaning(O)
- Both Input and Output observable.

Translation Data == Translation Equations == Meaning Equations

Motivation 1: Find the Latent Structure of Translation Equations

Motivation 2: How to Translate Correctly, i.e., Build new equations

The Structure of Equivalence?

Sentence-level translation equations

(De zonnestralen die door het raam binnenkomen)

==

(The sun rays that infiltrate through the window)

But how are translation equations built-up? Important for generalization.

The Structure of Equivalence? Analogy

Sentence-level translation equations

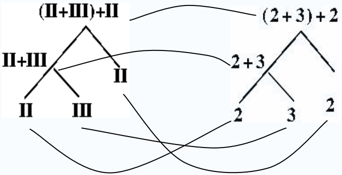
(De zonnestralen die door het raam binnenkomen) == (The sun rays that infiltrate through the window)

But how are translation equations built-up? **Important for generalization.**

Analogy: Decomposition of equations

- Two decimal alphabets;
- We know the “atomic units”, e.g., II=2, III=3
- Non-ambiguous translation
- One composition operator (**+**)
- No idioms, just composition

➪ **Easy to decompose recursively**

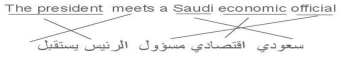


Recursive Structure of Translation Equivalence, How?

Translation Equivalence: Challenges

Induce mapping

Parallel Corpus: A large sample of source-target pairs of human translations.

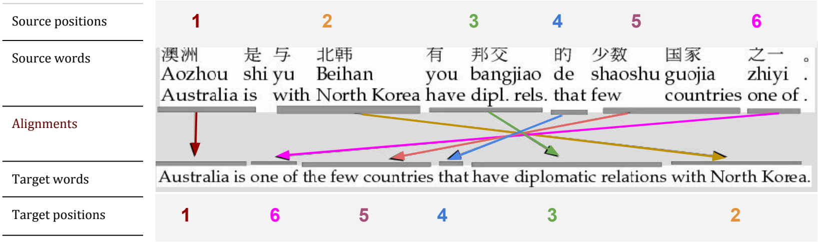
I **ran up** a big bill. I **ran up** a big hill.	Ik **heb** een grote rekening **opgelopen**. Ik **rende** een grote heuvel **op**.	**Ambiguity** Stochastic decisions
He **destroyed** them.	Hij **richtte** hen **ten gronde**.	**Idioms:** how to identify?
Je **ne** fume **pas** Ik rook **niet**	Je **ne** **VP-F** **pas** Ik **VP-N** **niet**.	**Non-Contiguous** mapping.
		**Morph. Variations** Canonical forms?
		**Word Order Differences** Mappings with permuted word order: huge space (n!). Example from (Chiang 2007)

Let us concentrate on Word Order for now

Structure of Translation Equivalence: Word Order

Parallel Corpus: A large sample of source-target pairs of human translations.

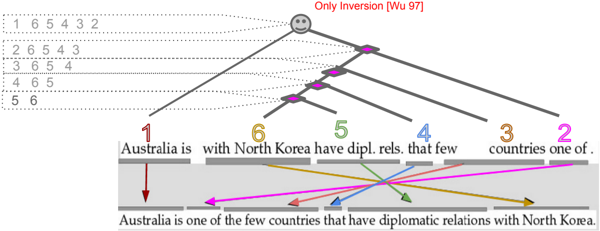
Induce word-level mapping: Many-to-Many Word Alignment induced as latent structure [existing].*



* Up-to some encapsulation of idioms, morphology, unaligned words: First approximation available [Brown et al 1992; Och & Ney 2003].

Alignments $\stackrel{?}{=}$ Sequence of individual alignments

Hierarchical Word Order (Surface Composition)

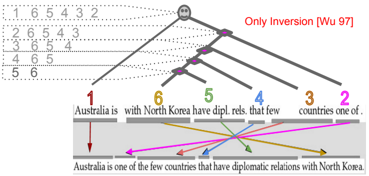


Hierarchical composition could have benefits: Long range reordering

The Questions of Translation Equivalence

Q1. How to learn translation equivalence over “words”?

- What are the units of equivalence?
- Which units map to which?
- What composition is needed to learn this?



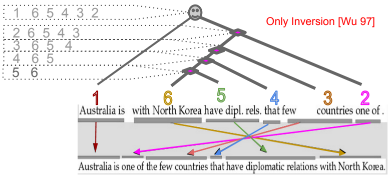
Q3. How to compose new translations from old ones to preserve meaning?

- Which representations?
- Which compositions preserve meaning?

Hierarchical vs. Sequential View: Applications

Q1. How to learn translation equivalence over “words”?

- What are the units of equivalence?
- Which units map to which?
- What composition is needed to learn this?



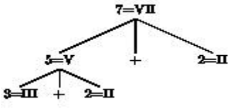
Q.2 How to learn word order from examples?

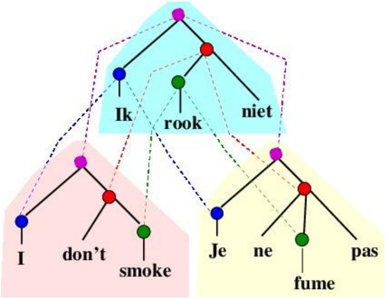
- Word order as first big challenge!
- Structure of equivalence?
- NLP II mostly about this!

Q3. How to compose new translations from old ones to preserve meaning?

- Which representations?
- Which compositions preserve meaning?

Hierarchical Equivalence: Questions





Questions:

- **Lexicon**: Which word-translation pairs?
- **Structure**: Which composition structure?
- **Composition**: Which operators?

. How to learn all this from parallel data?

. How to deal with ambiguity?

The Structure of NLP II

1. How to learn a lexicon and mapping between words? Sequential view.
   a. Word-based models and word alignments (IBM Models)
   b. Inducing alignments and using them for extracting phrases, i.e., translation equations at any level, not only sentence level
2. How to evaluate Machine Translation system output?
3. How to learn hierarchical models based on Synchronous Grammars?
   a. Synchronous grammars
   b. Hierarchical phrased-based model
4. How obtain semantic representations from multilingual data?
5. How to learn models of word-order differences (reordering) between languages?
   a. Permutations and their decomposition/factorization
   b. Synchronous grammars and permutations
   c. Learning from data

Statistical Machine Translation: First Steps

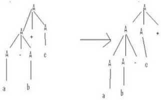
Translation between Languages

Human translators translate texts from a source language to a target language.

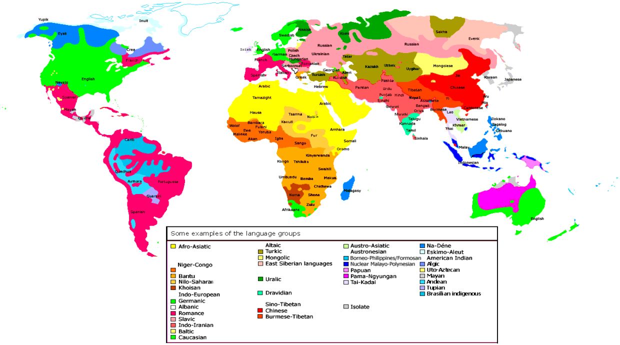
I don't smoke. Je ne fume pas

Can we build a computer program that translates texts from one language to another?

What challenges will we face and how do we tackle them?



So many languages, so little time



- Are the differences between languages arbitrary?
- Are there shared regularities between different languages?

How should we automatically translate?

History I: Premature Optimism and Failure

Some history on translation and speech recognition:

- During 50's and 60's
  First computers; Chomsky's grammars; programming languages; big optimism and huge funding

  `Translation is Easy:  we can program this!!`

  ALPAC (Automatic Language Processing Advisory Committee) Report 1966 (U.S. Government).
  Failure: AI abandons NLP, NLP abandons Translation
- During 70's and 80's:
  AI: "You need world-knowledge: build an ontology"
  CS: Concentrate on Information Retrieval
  Linguistics: We need better theory

History II: Renewed Optimism

- During 70's: A group of statisticians at IBM TJ Watson "digs up" an old idea ([Weaver 1948, 1949]):

  > *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.*

  Communication and Information Theory (Shannon, Weaver); Code breaking (Turing).
- During 80's: Success in ASR; Look at Translation
- During 90's: Success in parsing and Translation
- By 2006: Google introduces "Google Translate"!

Next: How good is statistical MT these days?

Modeling Human Translation Expertise

- Translators are Experts in Translation
- Humans: Study, work, acquire by experience . . .

Can we model "expertise acquisition" from experience?

- Observe and learn how humans translate?
- Use input-output translation examples: Parallel corpora
  No access to what happens in between
- How do we build and select the correct translation?
  Ambiguity is stalking us all the way.

How can we learn translation regularities from data?

Data and Statistical Models

Parallel corpus = a collection of text-chunks and their translations.
Parallel corpora are the by-product of *human translation*.
Every source chunk is paired with a target chunk.

Dutch	English
De prijs van het huis is gestegen.	The price of the house has risen.
Het huis kan worden verkocht.	The house can be sold.
Als het de marktprijs daalt zullen sommige gezinnen een zware tijd doormaken.	If the market price goes down, some families will go through difficult times.
⋮ ⋮ ⋮	⋮ ⋮ ⋮
⋮ ⋮ ⋮	⋮ ⋮ ⋮

- Hansards Canadian Parliament Proc. (English-French).
- European Parliament Proc. (23 languages).
- United Nations documents.
- Newspapers: Chinese-English; Arabic-English; Urdu-English.

The hidden structure of translation

How to model the translation mapping in the data?

The big cities will join forces if the prime minister maintains his present policy in the long run.

????

De grote steden zullen samen optrekken als de premier zijn huidige beleid op lange termijn blijft handhav

What is the nature of the mapping?

$$\text{“}\mathit{Translate}(\mathit{sentence}) = \hat{\sum}_i \mathit{Translate}(\mathit{part}_i)\text{”} \ ??$$

- What are $part_i$ and $Translate(part_i)$ in the data?
- What is $\hat{\sum}_i$?
- How to model differences in word-order, morphology etc?
- What about ambiguity, idioms etc?

Probabilistic Modeling: Simple Noisy Channel

Source sentence $\mathbf{s} = s_1, \ldots s_n$
Target sentence $\mathbf{t} = t_1, \ldots t_n$

$$\arg\max_{\mathbf{t}} P(\mathbf{t} \mid \mathbf{s}) = \arg\max_{\mathbf{t}} P(\mathbf{t}) \times P(\mathbf{s} \mid \mathbf{t})$$

- **Target Language Model** $P(\mathbf{t})$=?
  How regular is a given string **t** in the target language?

  $$P(\mathbf{t}) = \sum_{\mathbf{d}} P(\mathbf{t}, \mathbf{d})$$

  Derivations **d**: Finite-State / Context-Free Grammar
- **Translation Model** $P(\mathbf{s} \mid \mathbf{t})$=?
  How to model the mapping $\mathbf{t} \rightarrow \mathbf{s}$?

This course: Learning translation models from data

Modeling Parallel Corpus Data

- How to represent the source sentence?
- How to represent the target sentence?
- How to model the mapping between these representations?
  We need to model sentence pairs!!
- Is translation compositional?
- Some options: Probabilistic Synchronous Grammars, Probabilistic Tree Transducers, etc.
- What learning algorithms?
- How to automatically evaluate translation output?

Data and Models: Structure of lecture

- General statistical framework
- Word-based models: word alignments
- Phrase-based models: phrase-alignments
- Tree-based models: tree-alignments

Introduction to Statistical Machine Translation

Statistical Approach: Parallel Corpora

Task: Translate a source sentence **f** to a target sentence **e**.
Data: Parallel corpus (source-target sentence pairs).

Source-Channel Approach: IBM Models (1990's)

Parallel Corpus Example

Parallel corpus **C** = a collection of text-chunks and their translations.
Parallel corpora are the by-product of *human translation*.
Every source chunk is paired with a target chunk.

Dutch	English
De prijs van het huis is gestegen.	The price of the house has risen.
Het huis kan worden verkocht.	The house can be sold.
Als het de marktprijs daalt zullen sommige gezinnen een zware tijd doormaken.	If the market price goes down, some families will go through difficult times.
⋮ ⋮ ⋮	⋮ ⋮ ⋮
⋮ ⋮ ⋮	⋮ ⋮ ⋮

- Hansards Canadian Parliament Proc. (English-French).
- European Parliament Proc. (23 languages).
- United Nations documents.
- Newspapers: Chinese-English; Arabic-English; Urdu-English.
- TAUS corpora.

Generative Source-Channel Framework

Given source sentence **f**, select target sentence **e**

$$\arg\max_{\mathbf{e}\in E(\mathbf{f})}\{\ P(\mathbf{e}\mid\mathbf{f})\ \} = \arg\max_{\mathbf{e}\in E(\mathbf{f})}\{\ \overbrace{P(\mathbf{e})}^{L.M.} \times \overbrace{P(\mathbf{f}\mid\mathbf{e})}^{T.M.}\ \}$$

Set $E(\mathbf{f})$ is the set of hypothesized translations of **f**.

$P(\mathbf{f}\mid\mathbf{e})$: accounts for <u>divergence</u> in ...

- word order
- morphology
- syntactic relations
- idiomatic ways of expression
- $\vdots$

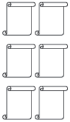
How to estimate $P(\mathbf{e}\mid\mathbf{f})$? Sparse-data problem!

Inducing The Structure of Translation Data

e = Mary did not slap the green witch .

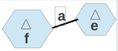
? ? ? ?

f = Maria no dio una bofetada a la bruja verde .



The latent structure of translation equivalence

- Graphical representations $\Delta_{\mathbf{f}}$ and $\Delta_{\mathbf{e}}$ for **f** and **e**
- Relation **a** between $\Delta_{\mathbf{f}}$ and $\Delta_{\mathbf{e}}$



$$\arg\max_{\mathbf{e}\in E(\mathbf{f})}\{\ P(\mathbf{e}\mid\mathbf{f})\ \} =$$

$$\arg\max_{\mathbf{e}\in E(\mathbf{f})}\{\ \sum_{\langle\Delta_{\mathbf{f}},\mathbf{a},\Delta_{\mathbf{e}}\rangle}\ P(\mathbf{e},\Delta_{\mathbf{f}},\Delta_{\mathbf{e}},\mathbf{a}\mid\mathbf{f})\ \}$$

The difficult question: Which $\Delta_{\mathbf{f}/\mathbf{e}}$ and **a** fit data best?

Structure in current models

$$\Delta_{\mathbf{f}} \xrightarrow{\mathbf{a}} \Delta_{\mathbf{e}}$$

In most current models structure of reordering:

- $\Delta_{\mathbf{f}/\mathbf{e}}$ are structures over word positions.
- **a** is an **alignment** between groups of word positions in $\Delta_{\mathbf{f}}$ and $\Delta_{\mathbf{e}}$.
- Challenge: Number of permutations of n words is $n!$

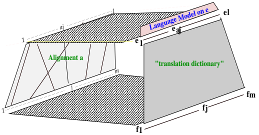
Structure shows translation units composing together

- What are the atomic translation units?
- How these compose together efficiently?
- How to put probs. on these structures?

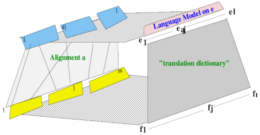
Structure helps combat sparsity and complexity

Structure in Existing Models: Sketch

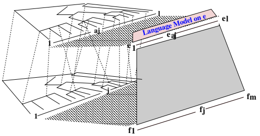
Word-based



Phrase-based



Tree-based



Problem: No sufficient stats to estimate $P(\mathbf{e} \mid \mathbf{f})$ from data

Word-Based Models: Word Alignments

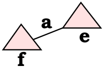
Some History and References

Statistical models with word-alignments:

- Brown, Cocke, Della Pietra, Della Pietra, Jelinek, Lafferty, Mercer and Roossin. A statistical approach to machine translation. Computational Linguistics, 1990.
- Brown, Della Pietra, Della Pietra and Mercer. The mathematics of statistical machine translation: parameter estimation., Computational Linguistics, 1993.
- Och and Ney: A Systematic Comparison of Various Statistical Alignment Models. Computational Linguistics, 2003.

Word-Based Models and Word-Alignment

a is a mapping between word positions.



- $\Delta_{\mathbf{f}}$ and $\Delta_{\mathbf{e}}$ are sequences of word positions. $\mathbf{e} = e_1^l = e_1 \ldots e_l$ and $\mathbf{f} = f_1^m = f_1 \ldots f_m$
- A hidden word-alignment **a**:

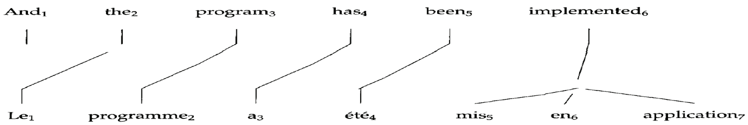
$$P(\mathbf{f} \mid \mathbf{e}) = \sum_{\mathbf{a}} P(\mathbf{a}, \mathbf{f} \mid \mathbf{e})$$

- Each source position has a single link to a target position or to position zero

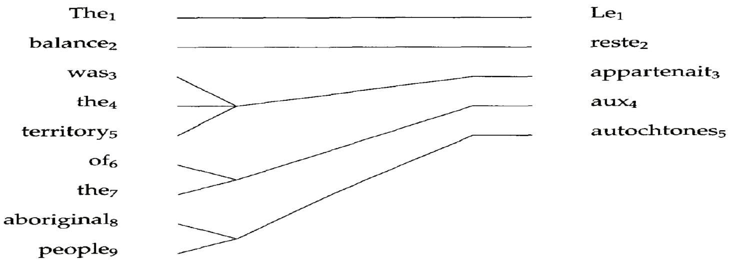
$$\mathbf{a} : \{pos_{\mathbf{f}}\} \rightarrow (\{pos_{\mathbf{e}}\} \cup \{0\})$$

- $\mathbf{a}_i$ or $\mathbf{a}(i)$, i.e., word position in **e** with which $\mathbf{f}_i$ is aligned.

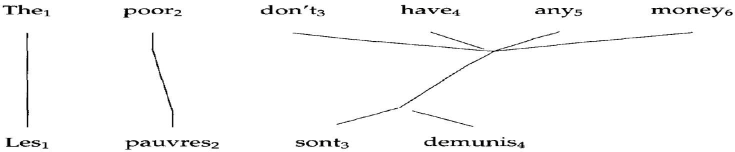
Word Alignment Example



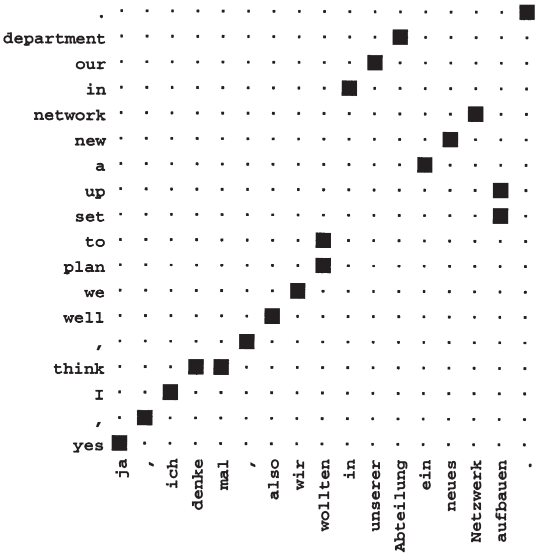
Word Alignment Example



Word Alignment Example: Not covered in this setting



Word Alignment Matrix Example



Translation model with word alignment

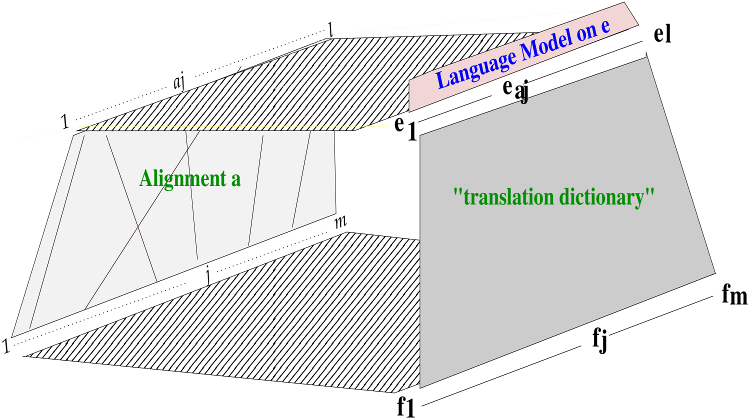
$$\arg\max_{\mathbf{e}} P(\mathbf{e} \mid \mathbf{f}) = \arg\max_{\mathbf{e}} P(\mathbf{e}) \times P(\mathbf{f} \mid \mathbf{e})$$

$$P(\mathbf{f} \mid \mathbf{e}) = \sum_{\mathbf{a}} P(\mathbf{a}, \mathbf{f} \mid \mathbf{e}) = \sum_{\mathbf{a}} P(\mathbf{a} \mid \mathbf{e}) \times P(\mathbf{f} \mid \mathbf{a}, \mathbf{e})$$

Questions

- How to parametrize the model?
  How are **e**, **f** and **a** composed from basic units?
- How to train the model?
  How to acquire word alignment?
- How to translate with this model?
  Decoding and computational issues (for second part)

Word-Alignment As Hidden Structure



We need to decompose

- The alignment **a** and the length m: $P(\mathbf{a} \mid \mathbf{e})$
- "Translation dictionary" $P(\mathbf{f} \mid \mathbf{e}, \mathbf{a})$

Word Alignment Models: General Scheme

Alignment of positions in **f** with positions in **e**:
$\mathbf{a} = a_1^m = a_1 \ldots a_m$

Markov process over **a**

$$P(a_1^m, f_1^m \mid e_1^l) = P(m \mid \mathbf{e}) \times \prod_{j=1}^{m} P(a_j \mid a_1^{j-1}, f_1^{j-1}, m, \mathbf{e}) \times P(f_j \mid a_1^j, f_1^{j-1}, m, \mathbf{e})$$

In words: to generate alignment **a** and foreign sentence **f**

1. Choose a length m for **f**
2. Generate alignment a_j given the preceding alignments, words in **f**, m, and **e**
3. Generate word f_j conditioned on structure so far and **e**.

IBM models are obtained by simplifications of this formula.

IBM Model I

$$P(a_1^m, f_1^m \mid e_1 \ldots e_l) = P(m \mid \mathbf{e}) \times \prod_{j=1}^{m} P(a_j \mid a_1^{j-1}, f_1^{j-1}, m, \mathbf{e}) \times P(f_j \mid a_1^j, f_1^{j-1}, m, \mathbf{e})$$

IBM Model I:

Length: $P(m \mid \mathbf{e}) =\approx P(m \mid l) \approx = \epsilon$ A fixed probability ϵ.

Align with uniform probability j with any a_j in $\mathbf{e}_1^l$ or NULL: $P(a_j \mid a_1^{j-1}, f_1^{j-1}, m, \mathbf{e}) \approx (l+1)^{-1}$

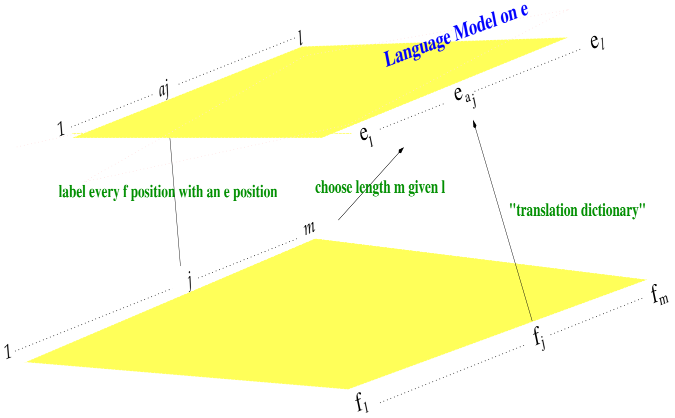
Note that a_j can be linked with l positions in $\mathbf{e}$ or with NULL.

Lexicon: lexicon parameters $\pi_t(f \mid e)$

$$P(f_j \mid a_1^j, f_1^{j-1}, m, \mathbf{e}) \approx P(f_j \mid e_{a_j}) = \pi_t(f_j \mid e_{a_j})$$

Parameters: ϵ and $\{\pi_t(f \mid e) \mid \langle f, e \rangle \in \mathbf{C}\}$.

Sketch IBM Model I



IBM Model I Explicit

IBM Model I altogether

$$P(\mathbf{f} \mid \mathbf{e}) = \sum_{a_1^m} P(a_1^m, f_1^m \mid e_1 \ldots e_l)$$

$$= \frac{\epsilon}{(l+1)^m} \times \sum_{a_1=0}^{l} \cdots \sum_{a_m=0}^{l} \prod_{j=1}^{m} \pi_t(f_j \mid e_{a_j})$$

Parameters: ϵ and $\{\pi_t(f \mid e) \mid \langle f, e \rangle \in \mathbf{C}\}$.

Fix ϵ, i.e., in practice put a uniform probability over a range $[1..m]$, for some natural number m.

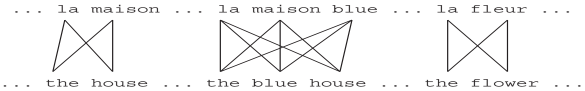
Crucial step: Efficiency (trick A)

$$= \frac{\epsilon}{(l+1)^m} \times \prod_{j=1}^{m} \sum_{i=0}^{l} \pi_t(f_j \mid e_i)$$

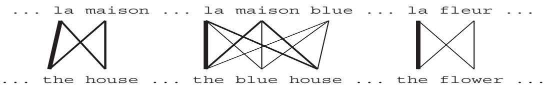
Questions regarding IBM Model I

- How to parametrize the model?
- How to train the model?
  How to acquire word alignment?
- How to translate with this model?
  Decoding and computational issues (for second part)

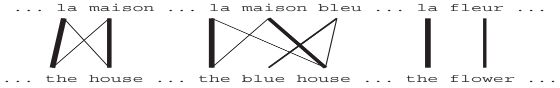
EM for Lexicon and Word Alignment Probs



EM for Lexicon and Word Alignment Probs



EM for Lexicon and Word Alignment Probs



EM for Lexicon and Word Alignment Probs

IBM Model II

Extends IBM Model I at alignment probs:

$$P(a_1^m, f_1^m \mid e_1 \ldots e_l) \approx \epsilon \times \prod_{j=1}^{m} \underline{P(a_j \mid a_1^{j-1}, f_1^{j-1}, m, \mathbf{e})} \times \pi_t(f_j \mid e_{a_j})$$

IBM Model II: changes only one element in IBM Model I:

- IBM Model I does not take into account the position of words in both strings

$$P(a_j \mid a_1^{j-1}, f_1^{j-1}, m, \mathbf{e}) = P(a_j \mid j, l, m) := \pi_A(a_j \mid j, l, m)$$

Where $\pi_A(.|.)$ are parameters to be learned from data.

IBM Models III, IV and V concentrate on more complex alignments allowing, e.g., $1 - to - n$ (fertility)

IBM Model II Parameters

$$P(a_1^m, f_1^m \mid e_1 \ldots e_l) \approx \epsilon \times \prod_{j=1}^{m} \pi_A(a_j \mid j, l, m) \times \pi_t(f_j \mid e_{a_j})$$

Parameters: $\{\pi_A(a_j \mid j, l, m)\}$ and $\{\pi_t(f_j \mid e_{a_j})\}$

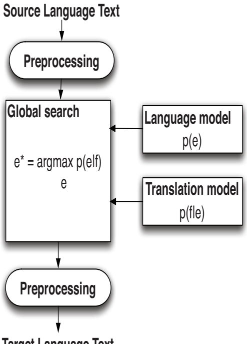
Estimation

Very similar to IBM Model I: EM estimation with the same complexity.

Translation Using EM Estimates

- Lexicon probability estimates: $\{\hat{\pi}_t(f_j \mid e_{a_j})\}$
- Alignment probabilities: $\{\hat{\pi}_A(a_j \mid j, m, l)\}$
- Translation Model + Language Model + Decoder

$$\arg\max_{\mathbf{e}} P(\mathbf{e} \mid \mathbf{f}) = \arg\max_{\mathbf{e}} P(\mathbf{e}) \times \sum_{\mathbf{a}} P(\mathbf{a}, \mathbf{f} \mid \mathbf{e})$$



Viterbi Word-Alignment using EM estimates

After EM has stabilized on estimates

$$\{\hat{\pi}_t(f_j \mid e_{a_j})\} \quad \textit{and} \quad \{\hat{\pi}_A(a_j \mid j, m, l)\}$$

For every $\langle \mathbf{f}, \mathbf{e} \rangle$ in $\mathbf{C}$ apply the following

$$\arg\max_{a_1^m} P(a_1^m \mid f_1^m, e_1^l) \;=$$

$$\arg\max_{a_1^m} \prod_{j=1}^{m} \hat{\pi}_A(a_j \mid j, m, l)\}\pi_t(f_j \mid e_{a_j})$$

This can be done efficiently:

$$\text{solution} \;\; a_i = \arg\max_{j \in [0..l]} \hat{\pi}_A(a_j \mid j, m, l)\hat{\pi}_t(f_i \mid e_j)$$

HMM Alignment Model: General Form

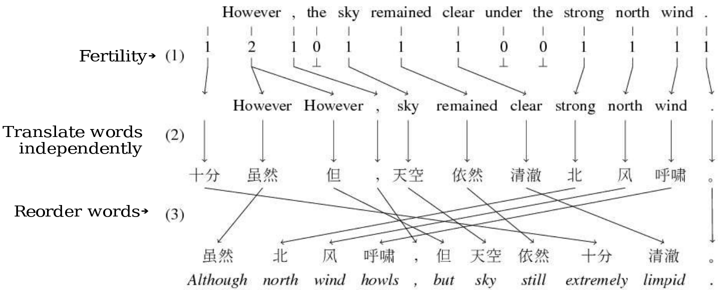
$$P(a_1^m, f_1^m \mid e_1 \ldots e_l) \approx \epsilon \times \prod_{j=1}^{m} \underline{P(a_j \mid a_1^{j-1}, f_1^{j-1}, m, \mathbf{e})} \times \pi_t(f_j \mid e_{a_j})$$

- Words do not move independently of each other: condition word movement on previous word movement

$$P(a_j \mid a_1^{j-1}, f_1^{j-1}, m, \mathbf{e}) \approx P(a_j \mid a_{j-1}, m)$$

IBM Model III (and IV): Example

- A hidden word-alignment **a**: $P(\mathbf{f} \mid \mathbf{e}) = \sum_{\mathbf{a}} P(\mathbf{a}, \mathbf{f} \mid \mathbf{e})$



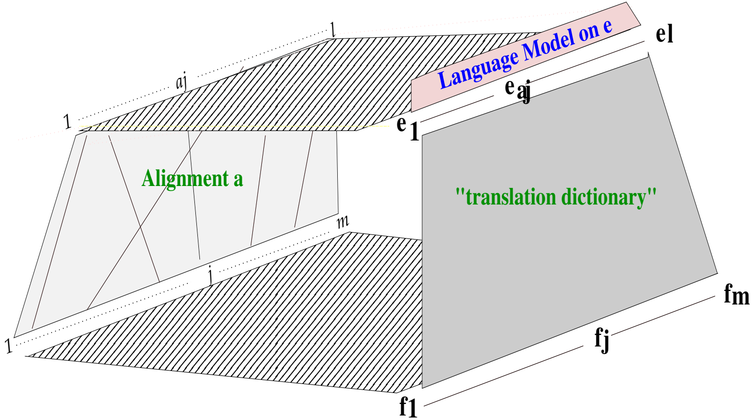
Estimate alignment + lexicon + reordering + fertility parameters.

Word-based Models (Och & Ney 2003)

Table 1
Overview of the alignment models.

Model	Alignment model	Fertility model	E-step	Deficient
Model 1	uniform	no	exact	no
Model 2	zero-order	no	exact	no
HMM	first-order	no	exact	no
Model 3	zero-order	yes	approximative	yes
Model 4	first-order	yes	approximative	yes
Model 5	first-order	yes	approximative	no
Model 6	first-order	yes	approximative	yes

Word-Alignment As Hidden Structure: Sufficient?



We assumed alignment between words and dictionary:

- Alignment **a** and the length m: $P(\mathbf{a} \mid \mathbf{e})$
- Dictionary $P(\mathbf{f} \mid \mathbf{e}, \mathbf{a})$

Limitations of Word-based Models

Limitations of word-based translation:

- Many-to-one and many-to-many is common:
  "Makes more difficult"/bemoeilijkt "Dat richtte (hen) ten gronde"/"That destroyed (them)"
- Reordering takes place (often) by whole blocks.
  Reordering individual words increases *ambiguity.*
  "The (big heavy) cow/la vaca (pesada grande)"
- Translation works by "fixed expressions" (idiomatic).
  Concatenating word-translations increases *ambiguity.*

Estimates of $P(\mathbf{f} \mid \mathbf{e})$ by word-based models are inaccurate.

Instead of words as basic events: multi-word events in corpus.

NLP II topics

We will cover literature (mostly articles) about

- Translation models: word-, phrase-, syntax-based
- Reordering models and synchronous grammars
- MT evaluation
- Paraphrasing and semantic models from parallel data
- Decoding algorithms