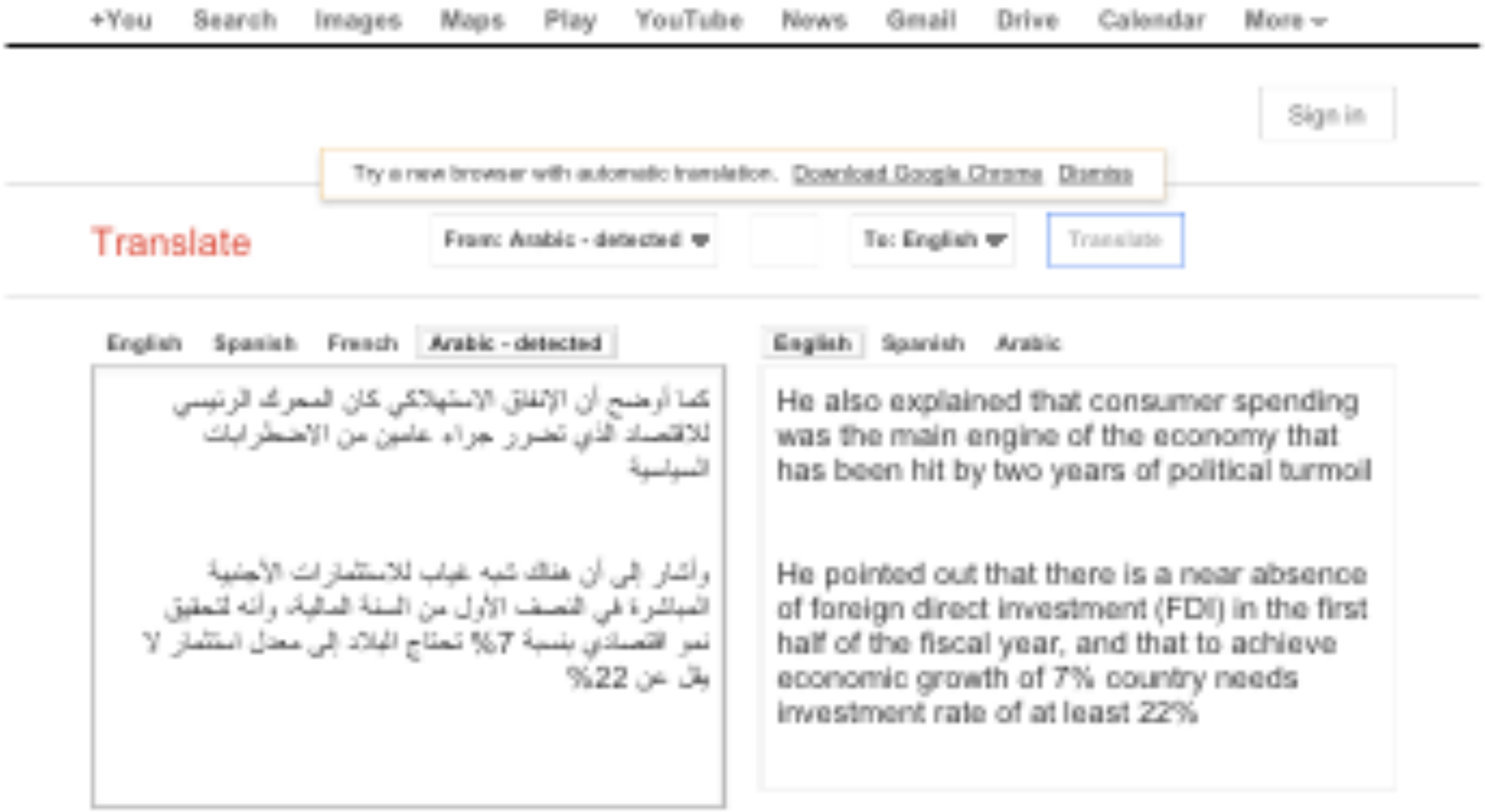
Machine Translation

Many slides from Michael Collins and Chris Calison-Burch



From: Arabic - detected

To: English

كما أوضح أن الإنفاق الاستهلاكي كان المحرك الرئيسي
للاقتصاد الذي تضرر جراء عامين من الاضطرابات
السياسية

وأشار إلى أن هناك شبه غياب للاستثمارات الأجنبية
المباشرة في النصف الأول من السنة المالية، وأنه لتحقيق
نمو اقتصادي بنسبة 7% تحتاج البلاد إلى معدل استثمار لا
يقل عن 22%

He also explained that consumer spending was the main engine of the economy that has been hit by two years of political turmoil

He pointed out that there is a near absence of foreign direct investment (FDI) in the first half of the fiscal year, and that to achieve economic growth of 7% country needs investment rate of at least 22%

Overview

- Challenges in machine translation
- Classical machine translation
- A brief introduction to statistical MT

Challenges: Lexical Ambiguity

(Example from Dorr et. al, 1999)

book the flight ⇒ reservar

read the book ⇒ libro

kill a man ⇒ matar

kill a process ⇒ acabar

Challenges: Differing Word Orders

- English word order is *subject – verb – object*
- Japanese word order is *subject – object – verb*

English:	IBM bought Lotus
Japanese:	*IBM Lotus bought*

English:	Sources said that IBM bought Lotus yesterday
Japanese:	*Sources yesterday IBM Lotus bought that said*

Syntactic Structure is not Preserved Across Translations (Example from Dorr et. al, 1999)

The bottle floated into the cave

$\Downarrow$

La botella entro a la cuerva flotando
(the bottle entered the cave floating)

Syntactic Ambiguity Causes Problems

(Example from Dorr et. al, 1999)

John hit the dog with the stick

$\Downarrow$

John golpeo el perro con el palo/que tenia el palo

Pronoun Resolution (Example from Dorr et. al, 1999)

The computer outputs the data; it is fast.

⇓

La computadora imprime los datos; es rapida

The computer outputs the data; it is stored in ascii.

⇓

La computadora imprime los datos; estan almacendos en ascii

Overview

- Challenges in machine translation
- Classical machine translation
- A brief introduction to statistical MT

Direct Machine Translation

- Translation is word-by-word

- Very little analysis of the source text (e.g., no syntactic or semantic analysis)

- Relies on a large bilingual directionary. For each word in the source language, the dictionary specifies a set of rules for translating that word

- After the words are translated, simple reordering rules are applied (e.g., move adjectives after nouns when translating from English to French)

An Example of a set of Direct Translation Rules

(From Jurafsky and Martin, edition 2, chapter 25. Originally from a system from Panov 1960)

Rules for translating *much* or *many* into Russian:

```
if preceding word is how return skol'ko
else if preceding word is as return stol'ko zhe
else if word is much
    if preceding word is very return nil
    else if following word is a noun return mnogo
else (word is many)
    if preceding word is a preposition and following word is noun return mnogii
    else return mnogo
```

Some Problems with Direct Machine Translation

- Lack of any analysis of the source language causes several problems, for example:
  - Difficult or impossible to capture long-range reorderings

English:	Sources said that IBM bought Lotus yesterday
Japanese:	*Sources yesterday IBM Lotus bought that said*

  - Words are translated without disambiguation of their syntactic role
    e.g., *that* can be a complementizer or determiner, and will often be translated differently for these two cases

    They said *that* ...

    They like *that* ice-cream

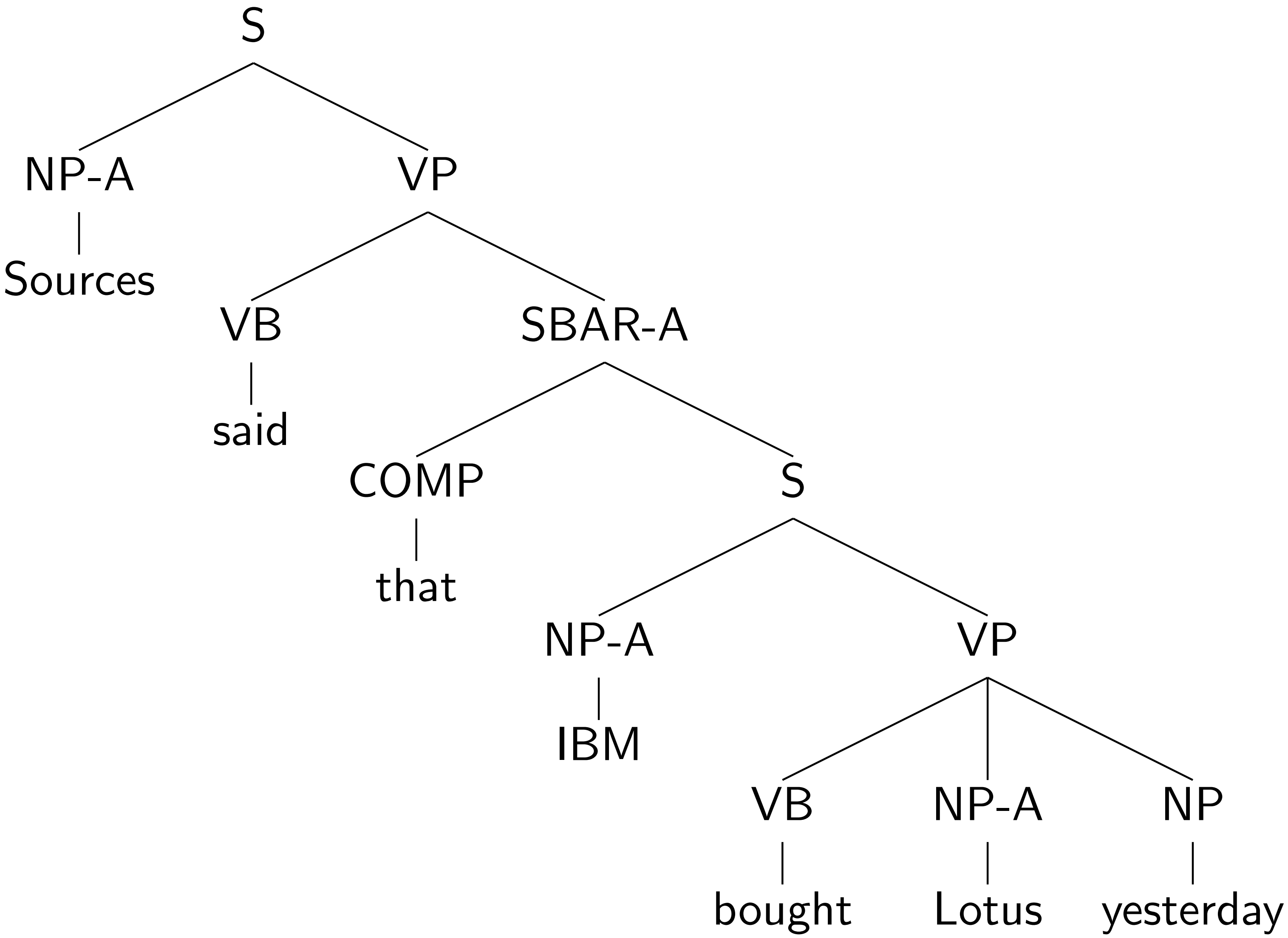
Transfer-Based Approaches

Three phases in translation:

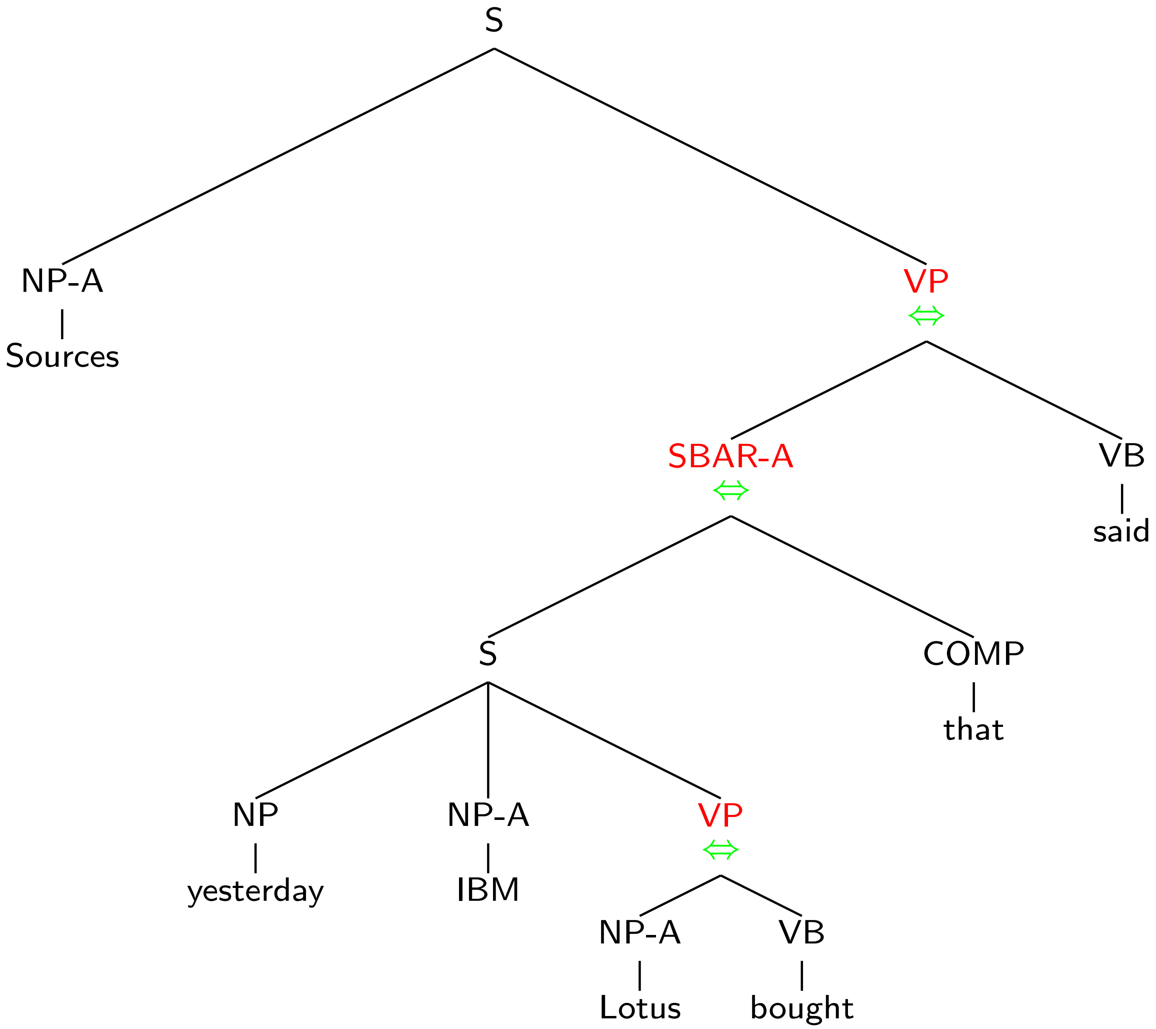
- Analysis: Analyze the source language sentence; for example, build a syntactic analysis of the source language sentence.
- Transfer: Convert the source-language parse tree to a target-language parse tree.
- Generation: Convert the target-language parse tree to an output sentence.

Transfer-Based Approaches

- The “parse trees” involved can vary from shallow analyses to much deeper analyses (even semantic representations).
- The transfer rules might look quite similar to the rules for direct translation systems. But they can now operate on syntactic structures.
- It’s easier with these approaches to handle long-distance reorderings
- The *Systran* systems are a classic example of this approach



⇒ Japanese: *Sources yesterday IBM Lotus bought that said*


S
NP-A
Sources
VP
⇔
SBAR-A
⇔
VB
said
S
COMP
that
NP
yesterday
NP-A
IBM
VP
⇔
NP-A
Lotus
VB
bought

Interlingua-Based Translation

Two phases in translation:

- Analysis: Analyze the source language sentence into a (language-independent) representation of its meaning.
- Generation: Convert the meaning representation into an output sentence.

Interlingua-Based Translation

One Advantage: If we want to build a translation system that translates between n languages, we need to develop n analysis and generation systems. With a transfer based system, we'd need to develop $O(n^2)$ sets of translation rules.

Disadvantage: What would a language-independent representation look like?

Interlingua-Based Translation

- How to represent different concepts in an interlingua?
- Different languages break down concepts in quite different ways:

  German has two words for *wall*: one for an internal wall, one for a wall that is outside

  Japanese has two words for *brother*: one for an elder brother, one for a younger brother

  Spanish has two words for *leg*: *pierna* for a human's leg, *pata* for an animal's leg, or the leg of a table
- An interlingua might end up simple being an intersection of these different ways of breaking down concepts, but that doesn't seem very satisfactory...

Overview

- Challenges in machine translation
- Classical machine translation
- A brief introduction to statistical MT

A Brief Introduction to Statistical MT

- Parallel corpora are available in several language pairs

- Basic idea: use a parallel corpus as a training set of translation examples

- Classic example: IBM work on French-English translation, using the Canadian Hansards. (1.7 million sentences of 30 words or less in length).

- Idea goes back to Warren Weaver (1949): suggested applying statistical and cryptanalytic techniques to translation.

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 (1949)

The Noisy Channel Model

- Goal: translation system from French to English
- Have a model $p(e \mid f)$ which estimates conditional probability of any English sentence e given the French sentence f. Use the training corpus to set the parameters.
- A Noisy Channel Model has two components:

$$\begin{array}{rl} p(e) & \textbf{the language model} \\ p(f \mid e) & \textbf{the translation model} \end{array}$$

- Giving:

$$p(e \mid f) = \frac{p(e, f)}{p(f)} = \frac{p(e)p(f \mid e)}{\sum_e p(e)p(f \mid e)}$$

and

$$\text{argmax}_e p(e \mid f) = \text{argmax}_e p(e)p(f \mid e)$$

More About the Noisy Channel Model

- The **language model** $p(e)$ could be a trigram model, estimated from any data (parallel corpus not needed to estimate the parameters)

- The **translation model** $p(f \mid e)$ is trained from a parallel corpus of French/English pairs.

- Note:
  - The translation model is backwards!
  - The language model can make up for deficiencies of the translation model.
  - Later we'll talk about how to build $p(f \mid e)$
  - Decoding, i.e., finding

    $$\text{argmax}_e p(e) p(f \mid e)$$

    is also a challenging problem.

Example from Koehn and Knight tutorial

Translation from Spanish to English, candidate translations based on $p(Spanish \mid English)$ alone:

Que hambre tengo yo

$\rightarrow$

What hunger have	$p(s\|e) = 0.000014$
Hungry I am so	$p(s\|e) = 0.000001$
I am so hungry	$p(s\|e) = 0.0000015$
Have i that hunger	$p(s\|e) = 0.000020$

$\dots$

Example from Koehn and Knight tutorial (continued)

With $p(Spanish \mid English) \times p(English)$:

Que hambre tengo yo

$\rightarrow$

What hunger have	$p(s\|e)p(e) = 0.000014 \times 0.000001$
Hungry I am so	$p(s\|e)p(e) = 0.000001 \times 0.0000014$
I am so hungry	$p(s\|e)p(e) = 0.0000015 \times 0.0001$

Have i that hunger	$p(s\|e)p(e) = 0.000020 \times 0.00000098$

$\ldots$

Recap: The Noisy Channel Model

- Goal: translation system from French to English
- Have a model $p(e \mid f)$ which estimates conditional probability of any English sentence e given the French sentence f. Use the training corpus to set the parameters.
- A Noisy Channel Model has two components:

$p(e)$ **the language model**

$p(f \mid e)$ **the translation model**

- Giving:

$$p(e \mid f) = \frac{p(e, f)}{p(f)} = \frac{p(e)p(f \mid e)}{\sum_e p(e)p(f \mid e)}$$

and

$$\text{argmax}_e p(e \mid f) = \text{argmax}_e p(e)p(f \mid e)$$

Roadmap for the Next Few Lectures

- IBM Models 1 and 2
- *Phrase-based* models

Overview

- IBM Model 1
- IBM Model 2
- EM Training of Models 1 and 2

IBM Model 1: Alignments

- How do we model $p(f \mid e)$?
- English sentence e has l words $e_1 \ldots e_l$,
  French sentence f has m words $f_1 \ldots f_m$.
- An alignment a identifies which English word each French word originated from
- Formally, an alignment a is $\{a_1, \ldots a_m\}$, where each $a_j \in \{0 \ldots l\}$.
- There are $(l+1)^m$ possible alignments.

IBM Model 1: Alignments

- e.g., $l = 6$, $m = 7$

$$e = \text{And the program has been implemented}$$

$$f = \text{Le programme a ete mis en application}$$

- One alignment is

$$\{2, 3, 4, 5, 6, 6, 6\}$$

- Another (bad!) alignment is

$$\{1, 1, 1, 1, 1, 1, 1\}$$

Alignments in the IBM Models

- We'll define models for $p(a \mid e, m)$ and $p(f \mid a, e, m)$, giving
$$p(f, a \mid e, m) = p(a \mid e, m)p(f \mid a, e, m)$$

- Also,
$$p(f \mid e, m) = \sum_{a \in \mathcal{A}} p(a \mid e, m)p(f \mid a, e, m)$$
where $\mathcal{A}$ is the set of all possible alignments

A By-Product: Most Likely Alignments

- Once we have a model $p(f, a \mid e, m) = p(a \mid e)p(f \mid a, e, m)$ we can also calculate

  $$p(a \mid f, e, m) = \frac{p(f, a \mid e, m)}{\sum_{a \in \mathcal{A}} p(f, a \mid e, m)}$$

  for any alignment a

- For a given f, e pair, we can also compute the most likely alignment,

  $$a^* = \arg\max_a p(a \mid f, e, m)$$

- Nowadays, the original IBM models are rarely (if ever) used for translation, but they are used for recovering alignments

An Example Alignment

French:
le conseil a rendu son avis , et nous devons à présent adopter un nouvel avis sur la base de la première position .

English:
the council has stated its position , and now , on the basis of the first position , we again have to give our opinion .

Alignment:
the/le council/conseil has/à stated/rendu its/son position/avis ,/, and/et now/présent ,/NULL on/sur the/le basis/base of/de the/la first/première position/position ,/NULL we/nous again/NULL have/devons to/a give/adopter our/nouvel opinion/avis ./.

IBM Model 1: Alignments

- In IBM model 1 all allignments a are equally likely:

$$p(a \mid e, m) = \frac{1}{(l+1)^m}$$

- This is a **major** simplifying assumption, but it gets things started...

IBM Model 1: Translation Probabilities

- Next step: come up with an estimate for

$$p(f \mid a, e, m)$$

- In model 1, this is:

$$p(f \mid a, e, m) = \prod_{j=1}^{m} t(f_j \mid e_{a_j})$$

- e.g., $l = 6$, $m = 7$

$$e = \text{And the program has been implemented}$$

$$f = \text{Le programme a ete mis en application}$$

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

$$\begin{aligned} p(f \mid a, e) \quad = \quad & t(Le \mid the) \times \\ & t(programme \mid program) \times \\ & t(a \mid has) \times \\ & t(ete \mid been) \times \\ & t(mis \mid implemented) \times \\ & t(en \mid implemented) \times \\ & t(application \mid implemented) \end{aligned}$$

IBM Model 1: The Generative Process

To generate a French string f from an English string e:

- **Step 1:** Pick an alignment a with probability $\frac{1}{(l+1)^m}$
- **Step 2:** Pick the French words with probability

$$p(f \mid a, e, m) = \prod_{j=1}^{m} t(f_j \mid e_{a_j})$$

The final result:

$$p(f, a \mid e, m) = p(a \mid e, m) \times p(f \mid a, e, m) = \frac{1}{(l+1)^m} \prod_{j=1}^{m} t(f_j \mid e_{a_j})$$

An Example Lexical Entry

English	French	Probability
position	position	0.756715
position	situation	0.0547918
position	mesure	0.0281663
position	vue	0.0169303
position	point	0.0124795
position	attitude	0.0108907

. . . de la situation au niveau des négociations de l ' ompi . . .
. . . of the current position in the wipo negotiations . . .

nous ne sommes pas en mesure de décider , . . .
we are not in a position to decide , . . .

. . . le point de vue de la commission face à ce problème complexe .
. . . the commission 's position on this complex problem .

Overview

- IBM Model 1
- IBM Model 2
- EM Training of Models 1 and 2

IBM Model 2

- Only difference: we now introduce **alignment** or **distortion** parameters

$\mathbf{q}(i \mid j, l, m)$ = Probability that j'th French word is connected to i'th English word, given sentence lengths of e and f are l and m respectively

- Define

$$p(a \mid e, m) = \prod_{j=1}^{m} \mathbf{q}(a_j \mid j, l, m)$$

where $a = \{a_1, \ldots a_m\}$

- Gives

$$p(f, a \mid e, m) = \prod_{j=1}^{m} \mathbf{q}(a_j \mid j, l, m) \mathbf{t}(f_j \mid e_{a_j})$$

An Example

$$\begin{aligned} l &= 6 \\ m &= 7 \\ e &= \text{And the program has been implemented} \\ f &= \text{Le programme a ete mis en application} \\ a &= \{2, 3, 4, 5, 6, 6, 6\} \end{aligned}$$

$$\begin{aligned} p(a \mid e, 7) = \; & \mathbf{q}(2 \mid 1, 6, 7) \times \\ & \mathbf{q}(3 \mid 2, 6, 7) \times \\ & \mathbf{q}(4 \mid 3, 6, 7) \times \\ & \mathbf{q}(5 \mid 4, 6, 7) \times \\ & \mathbf{q}(6 \mid 5, 6, 7) \times \\ & \mathbf{q}(6 \mid 6, 6, 7) \times \\ & \mathbf{q}(6 \mid 7, 6, 7) \end{aligned}$$

An Example

$$\begin{aligned} l &= 6 \\ m &= 7 \\ e &= \text{And the program has been implemented} \\ f &= \text{Le programme a ete mis en application} \\ a &= \{2, 3, 4, 5, 6, 6, 6\} \end{aligned}$$

$$\begin{aligned} p(f \mid a, e, 7) &= \mathbf{t}(Le \mid the) \times \\ &\quad \mathbf{t}(programme \mid program) \times \\ &\quad \mathbf{t}(a \mid has) \times \\ &\quad \mathbf{t}(ete \mid been) \times \\ &\quad \mathbf{t}(mis \mid implemented) \times \\ &\quad \mathbf{t}(en \mid implemented) \times \\ &\quad \mathbf{t}(application \mid implemented) \end{aligned}$$

IBM Model 2: The Generative Process

To generate a French string f from an English string e:

- **Step 1:** Pick an alignment $a = \{a_1, a_2 \ldots a_m\}$ with probability

$$\prod_{j=1}^{m} \mathbf{q}(a_j \mid j, l, m)$$

- **Step 3:** Pick the French words with probability

$$p(f \mid a, e, m) = \prod_{j=1}^{m} \mathbf{t}(f_j \mid e_{a_j})$$

The final result:

$$p(f, a \mid e, m) = p(a \mid e, m)p(f \mid a, e, m) = \prod_{j=1}^{m} \mathbf{q}(a_j \mid j, l, m)\mathbf{t}(f_j \mid e_{a_j})$$

Recovering Alignments

- If we have parameters q and t, we can easily recover the most likely alignment for any sentence pair

- Given a sentence pair $e_1, e_2, \ldots, e_l$, $f_1, f_2, \ldots, f_m$, define

$$a_j = \arg \max_{a \in \{0 \ldots l\}} q(a|j,l,m) \times t(f_j|e_a)$$

for $j = 1 \ldots m$

e = And the program has been implemented

f = Le programme a ete mis en application

Overview

- IBM Model 1
- IBM Model 2
- EM Training of Models 1 and 2

The Parameter Estimation Problem

- Input to the parameter estimation algorithm: $(e^{(k)}, f^{(k)})$ for $k = 1 \ldots n$. Each $e^{(k)}$ is an English sentence, each $f^{(k)}$ is a French sentence

- Output: parameters $t(f|e)$ and $q(i|j, l, m)$

- A key challenge: **we do not have alignments on our training examples**, e.g.,

$$\begin{aligned} e^{(100)} &= \text{And the program has been implemented} \\ f^{(100)} &= \text{Le programme a ete mis en application} \end{aligned}$$

Parameter Estimation if the Alignments are Observed

- First: case where alignments are observed in training data. E.g.,

$$\begin{aligned} e^{(100)} &= \text{And the program has been implemented} \\ f^{(100)} &= \text{Le programme a ete mis en application} \\ a^{(100)} &= \langle 2, 3, 4, 5, 6, 6, 6 \rangle \end{aligned}$$

- Training data is $(e^{(k)}, f^{(k)}, a^{(k)})$ for $k = 1 \ldots n$. Each $e^{(k)}$ is an English sentence, each $f^{(k)}$ is a French sentence, each $a^{(k)}$ is an alignment
- Maximum-likelihood parameter estimates in this case are trivial:

$$t_{ML}(f|e) = \frac{\text{Count}(e,\ f)}{\text{Count}(e)} \quad q_{ML}(j|i,l,m) = \frac{\text{Count}(j|i,l,m)}{\text{Count}(i,l,m)}$$

Input: A training corpus $(f^{(k)}, e^{(k)}, a^{(k)})$ for $k = 1 \ldots n$, where $f^{(k)} = f_1^{(k)} \ldots f_{m_k}^{(k)}$, $e^{(k)} = e_1^{(k)} \ldots e_{l_k}^{(k)}$, $a^{(k)} = a_1^{(k)} \ldots a_{m_k}^{(k)}$.

Algorithm:

- Set all counts $c(\ldots) = 0$
- For $k = 1 \ldots n$
  - For $i = 1 \ldots m_k$, For $j = 0 \ldots l_k$,

$$\begin{array}{rcl} c(e_j^{(k)}, f_i^{(k)}) & \leftarrow & c(e_j^{(k)}, f_i^{(k)}) + \delta(k,i,j) \\ c(e_j^{(k)}) & \leftarrow & c(e_j^{(k)}) + \delta(k,i,j) \\ c(j|i,l,m) & \leftarrow & c(j|i,l,m) + \delta(k,i,j) \\ c(i,l,m) & \leftarrow & c(i,l,m) + \delta(k,i,j) \end{array}$$

where $\delta(k,i,j) = 1$ if $a_i^{(k)} = j$, 0 otherwise.

Output: $t_{ML}(f|e) = \frac{c(e,f)}{c(e)}$, $q_{ML}(j|i,l,m) = \frac{c(j|i,l,m)}{c(i,l,m)}$

Parameter Estimation with the EM Algorithm

- Training examples are $(e^{(k)}, f^{(k)})$ for $k = 1 \ldots n$. Each $e^{(k)}$ is an English sentence, each $f^{(k)}$ is a French sentence
- The algorithm is related to algorithm when alignments are observed, but two key differences:
  1. The algorithm is *iterative*. We start with some initial (e.g., random) choice for the q and t parameters. At each iteration we compute some "counts" based on the data together with our current parameter estimates. We then re-estimate our parameters with these counts, and iterate.
  2. We use the following definition for $\delta(k, i, j)$ at each iteration:

$$\delta(k,i,j) = \frac{q(j|i,l_k,m_k)t(f_i^{(k)}|e_j^{(k)})}{\sum_{j=0}^{l_k} q(j|i,l_k,m_k)t(f_i^{(k)}|e_j^{(k)})}$$

Input: A training corpus $(f^{(k)}, e^{(k)})$ for $k = 1 \ldots n$, where $f^{(k)} = f_1^{(k)} \ldots f_{m_k}^{(k)}$, $e^{(k)} = e_1^{(k)} \ldots e_{l_k}^{(k)}$.

Initialization: Initialize $t(f|e)$ and $q(j|i, l, m)$ parameters (e.g., to random values).

```
For $s = 1 \ldots S$
    Set all counts $c(\ldots) = 0$
    For $k = 1 \ldots n$
        For $i = 1 \ldots m_k$, For $j = 0 \ldots l_k$
            $c(e_j^{(k)}, f_i^{(k)}) \leftarrow c(e_j^{(k)}, f_i^{(k)}) + \delta(k, i, j)$
            $c(e_j^{(k)}) \leftarrow c(e_j^{(k)}) + \delta(k, i, j)$
            $c(j|i, l, m) \leftarrow c(j|i, l, m) + \delta(k, i, j)$
            $c(i, l, m) \leftarrow c(i, l, m) + \delta(k, i, j)$
        where
            $\delta(k, i, j) = \frac{q(j|i, l_k, m_k) t(f_i^{(k)}|e_j^{(k)})}{\sum_{j=0}^{l_k} q(j|i, l_k, m_k) t(f_i^{(k)}|e_j^{(k)})}$
    Recalculate the parameters:
        $t(f|e) = \frac{c(e, f)}{c(e)}$    $q(j|i, l, m) = \frac{c(j|i, l, m)}{c(i, l, m)}$
```

$$\delta(k,i,j) = \frac{q(j|i,l_k,m_k)t(f_i^{(k)}|e_j^{(k)})}{\sum_{j=0}^{l_k} q(j|i,l_k,m_k)t(f_i^{(k)}|e_j^{(k)})}$$

$e^{(100)}$ = And the program has been implemented

$f^{(100)}$ = Le programme a ete mis en application

Justification for the Algorithm

- Training examples are $(e^{(k)}, f^{(k)})$ for $k = 1 \ldots n$. Each $e^{(k)}$ is an English sentence, each $f^{(k)}$ is a French sentence

- The log-likelihood function:

$$L(t, q) = \sum_{k=1}^{n} \log p(f^{(k)}|e^{(k)}) = \sum_{k=1}^{n} \log \sum_{a} p(f^{(k)}, a|e^{(k)})$$

- The maximum-likelihood estimates are

$$\arg\max_{t,q} L(t, q)$$

- The EM algorithm will converge to a *local maximum* of the log-likelihood function

Summary

- Key ideas in the IBM translation models:
  - Alignment variables
  - Translation parameters, e.g., $t(\text{chien}|\text{dog})$
  - Distortion parameters, e.g., $q(2|1, 6, 7)$
- The EM algorithm: an iterative algorithm for training the q and t parameters
- Once the parameters are trained, we can recover the most likely alignments on our training examples

$$\begin{aligned} e &= \text{And the program has been implemented} \\ f &= \text{Le programme a ete mis en application} \end{aligned}$$

Phrase-Based Translation

Overview

- Learning phrases from alignments
- A phrase-based model
- Decoding in phrase-based models

Phrase-Based Models

- First stage in training a phrase-based model is extraction of a *phrase-based (PB) lexicon*

- A PB lexicon pairs strings in one language with strings in another language, e.g.,

nach Kanada	↔	in Canada
zur Konferenz	↔	to the conference
Morgen	↔	tomorrow
fliege	↔	will fly
. . .		

An Example (from tutorial by Koehn and Knight)

- A training example (Spanish/English sentence pair):

  Spanish: Maria no daba una bofetada a la bruja verde

  English: Mary did not slap the green witch

- Some (not all) phrase pairs extracted from this example:

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

- We'll see how to do this using *alignments* from the IBM models (e.g., from IBM model 2)

Recap: IBM Model 2

- IBM model 2 defines a distribution $p(a, f|e, m)$ where f is foreign (French) sentence, e is an English sentence, a is an *alignment*, m is the length of the foreign sentence
- A useful by-product: once we've trained the model, for any (f, e) pair, we can calculate

  $$a^* = \arg\max_a p(a|f, e, m) = \arg\max_a p(a, f|e, m)$$

  under the model. a^* is the **most likely alignment**

English: Mary did not slap the green witch

Spanish: Maria no daba una bofetada a la bruja verde

Representation as Alignment Matrix

	Maria	no	daba	una	bof’	a	la	bruja	verde
Mary	●								
did						●			
not		●							
slap			●	●	●				
the							●		
green									●
witch								●	

(Note: “bof”’ = “bofetada”)

In IBM model 2, each foreign (Spanish) word is aligned to exactly one English word. The matrix shows these alignments.

Finding Alignment Matrices

- Step 1: train IBM model 2 for $p(f \mid e)$, and come up with most likely alignment for each (e, f) pair

- Step 2: train IBM model 2 for $p(e \mid f)$ and come up with most likely alignment for each (e, f) pair

- We now have two alignments:
  take intersection of the two alignments as a starting point

Alignment from $p(f \mid e)$ model:

	Maria	no	daba	una	bof'	a	la	bruja	verde
Mary	●								
did						●			
not		●							
slap			●	●	●				
the							●		
green									●
witch								●	

Alignment from $p(e \mid f)$ model:

	Maria	no	daba	una	bof'	a	la	bruja	verde
Mary	●								
did		●							
not		●							
slap					●				
the							●		
green									●
witch								●	

Intersection of the two alignments:

	Maria	no	daba	una	bof’	a	la	bruja	verde
Mary	●								
did									
not		●							
slap					●				
the							●		
green									●
witch								●	

The intersection of the two alignments has been found to be a very reliable starting point

Heuristics for Growing Alignments

- Only explore alignment in **union** of $p(f \mid e)$ and $p(e \mid f)$ alignments
- Add one alignment point at a time
- Only add alignment points which align a word that currently has no alignment
- At first, restrict ourselves to alignment points that are "neighbors" (adjacent or diagonal) of current alignment points
- Later, consider other alignment points

The final alignment, created by taking the intersection of the two alignments, then adding new points using the growing heuristics:

	Maria	no	daba	una	bof’	a	la	bruja	verde
Mary	●								
did		●							
not		●							
slap			●	●	●				
the						●	●		
green									●
witch								●	

Note that the alignment is no longer many-to-one: potentially multiple Spanish words can be aligned to a single English word, and vice versa.

Extracting Phrase Pairs from the Alignment Matrix

	Maria	no	daba	una	bof’	a	la	bruja	verde
Mary	●								
did		●							
not		●							
slap			●	●	●				
the						●	●		
green									●
witch								●	

- A phrase-pair consists of a sequence of English words, e, paired with a sequence of foreign words, f
- A phrase-pair (e, f) is *consistent* if: 1) there is at least one word in e aligned to a word in f; 2) there are no words in f aligned to words outside e; 3) there are no words in e aligned to words outside f

  e.g., (Mary did not, Maria no) is consistent. (Mary did, Maria no) is *not* consistent
- We extract all consistent phrase pairs from the training example.

Probabilities for Phrase Pairs

- For any phrase pair (f, e) extracted from the training data, we can calculate

$$t(f|e) = \frac{Count(f, e)}{Count(e)}$$

e.g.,

$$t(\text{daba una bofetada} \mid \text{slap}) = \frac{Count(\text{daba una bofetada}, \text{slap})}{Count(\text{slap})}$$

An Example Phrase Translation Table

An example from Koehn, EACL 2006 tutorial. (Note that we have $t(e|f)$ not $t(f|e)$ in this example.)

- Phrase Translations for *den Vorschlag*

English	t(e|f)	**English**	t(e|f)
the proposal	0.6227	the suggestions	0.0114
's proposal	0.1068	the proposed	0.0114
a proposal	0.0341	the motion	0.0091
the idea	0.0250	the idea of	0.0091
this proposal	0.0227	the proposal ,	0.0068
proposal	0.0205	its proposal	0.0068
of the proposal	0.0159	it	0.0068
the proposals	0.0159	...	...

Overview

- Learning phrases from alignments
- A phrase-based model
- Decoding in phrase-based models

Phrase-Based Systems: A Sketch

$\boxed{\text{Today}}$

$\boxed{\text{Heute}}$ werden wir uber die Wiedereroffnung des Mont-Blanc-Tunnels diskutieren

$$\begin{aligned}\text{Score} \quad = & \quad \underbrace{\log q(\text{Today} \mid \text{*}, \text{*})}_{\text{Language model}} \\ + & \quad \underbrace{\log t(\text{Heute} \mid \text{Today})}_{\text{Phrase model}} \\ + & \quad \underbrace{\eta \times 0}_{\text{Distortion model}}\end{aligned}$$

Phrase-Based Systems: A Sketch

Today $\boxed{\text{we shall be}}$

Heute $\boxed{\text{werden wir}}$ uber die Wiedereroffnung des Mont-Blanc-Tunnels diskutieren

$$\begin{aligned}\text{Score} \quad = \quad & \underbrace{\log q(\text{we}|\text{*, Today}) + \log q(\text{shall}|\text{Today, we}) + \log q(\text{be}|\text{we, shall})}_{\text{Language model}} \\ + \quad & \underbrace{\log t(\text{werden wir} \mid \text{we shall be})}_{\text{Phrase model}} \\ + \quad & \underbrace{\eta \times 0}_{\text{Distortion model}}\end{aligned}$$

Phrase-Based Systems: A Sketch

Today we shall be $\boxed{\text{debating}}$

Heute werden wir uber die Wiedereroffnung des Mont-Blanc-Tunnels $\boxed{\text{diskutieren}}$

$$\text{Score} = \underbrace{\log q(\text{debating}|\text{shall, be})}_{\text{Language model}}$$

$$+ \underbrace{\log t(\text{diskutieren} \mid \text{debating})}_{\text{Phrase model}}$$

$$+ \underbrace{\eta \times 6}_{\text{Distortion model}}$$

Phrase-Based Systems: A Sketch

Today we shall be debating the reopening

Heute werden wir uber die Wiedereroffnung
des Mont-Blanc-Tunnels diskutieren

Phrase-Based Systems: A Sketch

Today we shall be debating the reopening
of the Mont Blanc tunnel

Heute werden wir uber die Wiedereroffnung
des Mont-Blanc-Tunnels diskutieren

Decoding

Phrase-based Translation

An example sentence:

wir müssen auch diese kritik ernst nehmen

A phrase-based lexicon contains phrase entries (f, e) where f is a sequence of one or more foreign words, e is a sequence of one or more English words. Example phrase entries that are relevant to our example:

(wir müssen, we must)

(wir müssen auch, we must also)

(ernst, seriously)

Each phrase (f, e) has a score $g(f, e)$. E.g.,

$$g(f, e) = \log\left(\frac{\text{Count}(f, e)}{\text{Count(e)}}\right)$$

Phrase-based Models: Definitions

- A phrase-based model consists of:
  1. A phrase-based lexicon, consisting of entries (f, e) such as

     (wir müssen, we must)

     Each lexical entry has a score $g(f, e)$, e.g.,

     $$g(\text{wir müssen, we must}) = \log\left(\frac{\text{Count(wir müssen, we must)}}{\text{Count(we must)}}\right)$$

  2. A trigram language model, with parameters $q(w|u, v)$. E.g., $q(\text{also}|\text{we, must})$.
  3. A "distortion parameter" η (typically negative).

Phrase-based Translation: Definitions

An example sentence:

wir müssen auch diese kritik ernst nehmen

- For a particular input (source-language) sentence $x_1 \ldots x_n$, a phrase is a tuple (s, t, e), signifying that the subsequence $x_s \ldots x_t$ in the source language sentence can be translated as the target-language string e, using an entry from the phrase-based lexicon. E.g., $(1, 2, \text{we must})$
- $\mathcal{P}$ is the set of all phrases for a sentence.
- For any phrase p, $s(p)$, $t(p)$ and $e(p)$ are its three components. $g(p)$ is the score for a phrase.

Definitions

- A derivation y is a finite sequence of phrases, $p_1, p_2, \ldots p_L$, where each p_j for $j \in \{1 \ldots L\}$ is a member of $\mathcal{P}$.
- The length L can be any positive integer value.
- For any derivation y we use $e(y)$ to refer to the underlying translation defined by y. E.g.,

  $y =$ (1, 3, we must also), (7, 7, take), (4, 5, this criticism), (6, 6, seriously)

  and

  $$e(y) = \text{we must also take this criticism seriously}$$

Valid Derivations

- For an input sentence $x = x_1 \ldots x_n$, we use $\mathcal{Y}(x)$ to refer to the set of valid derivations for x.
- $\mathcal{Y}(x)$ is the set of all finite length sequences of phrases $p_1 p_2 \ldots p_L$ such that:
  - Each p_k for $k \in \{1 \ldots L\}$ is a member of the set of phrases $\mathcal{P}$ for $x_1 \ldots x_n$.
  - Each word in x is translated exactly once.
  - For all $k \in \{1 \ldots (L-1)\}$, $|t(p_k) + 1 - s(p_{k+1})| \leq d$ where $d \geq 0$ is a parameter of the model. In addition, we must have $|1 - s(p_1)| \leq d$

Examples

wir müssen auch diese kritik ernst nehmen

$y =$ (1, 3, we must also), (7, 7, take), (4, 5, this criticism), (6, 6, seriously)

Examples

wir müssen auch diese kritik ernst nehmen

$y =$ (1, 3, we must also), (7, 7, take), (4, 5, this criticism), (6, 6, seriously)

$y =$ (1, 3, we must also), (1, 2, we must), (4, 5, this criticism), (6, 6, seriously)

Examples

wir müssen auch diese kritik ernst nehmen

$y =$ (1, 3, we must also), (7, 7, take), (4, 5, this criticism), (6, 6, seriously)

$y =$ (1, 3, we must also), (1, 2, we must), (4, 5, this criticism), (6, 6, seriously)

$y =$ (1, 2, we must), (7, 7, take), (3, 3, also), (4, 5, this criticism), (6, 6, seriously)

Scoring Derivations

The optimal translation under the model for a source-language sentence x will be

$$\arg\max_{y \in \mathcal{Y}(x)} f(y)$$

In phrase-based systems, the score for any derivation y is calculated as follows:

$$h(e(y)) + \sum_{k=1}^{L} g(p_k) + \sum_{k=0}^{L-1} \eta \times |t(p_k) + 1 - s(p_{k+1})|$$

where the parameter η is the distortion penalty (typically negative). (We define $t(p_0) = 0$).

$h(e(y))$ is the trigram language model score. $g(p_k)$ is the phrase-based score for p_k.

An Example

wir müssen auch diese kritik ernst nehmen

$y =$ (1, 3, we must also), (7, 7, take), (4, 5, this criticism), (6, 6, seriously)

Decoding Algorithm: Definitions

- A state is a tuple

$$(e_1, e_2, b, r, \alpha)$$

  where e_1, e_2 are English words, b is a bit-string of length n, r is an integer specifying the end-point of the last phrase in the state, and α is the score for the state.

- The initial state is

$$q_0 = (*, *, 0^n, 0, 0)$$

  where 0^n is bit-string of length n, with n zeroes.

States, and the Search Space

wir müssen auch diese kritik ernst nehmen

$(*, *, 0000000, 0, 0)$

Transitions

- We have $ph(q)$ for any state q, which returns set of phrases that are allowed to follow state $q = (e_1, e_2, b, r, \alpha)$.
- For a phrase p to be a member of $ph(q)$, it must satisfy the following conditions:
  - p must not overlap with the bit-string b. I.e., we need $b_i = 0$ for $i \in \{s(p) \ldots t(p)\}$.
  - The distortion limit must not be violated. More formally, we must have $|r + 1 - s(p)| \leq d$ where d is the distortion limit.

An Example of the Transition Function

wir müssen auch diese kritik ernst nehmen

$(\text{must}, \text{also}, 1110000, 3, -2.5)$

An Example of the Transition Function

wir müssen auch diese kritik ernst nehmen

$(\text{must, also}, 1110000, 3, -2.5)$

In addition, we define $next(q, p)$ to be the state formed by combining state q with phrase p.

The $next$ function

Formally, if $q = (e_1, e_2, b, r, \alpha)$, and $p = (s, t, \epsilon_1 \ldots \epsilon_M)$, then next$(q, p)$ is the state $q' = (e'_1, e'_2, b', r', \alpha')$ defined as follows:

- First, for convenience, define $\epsilon_{-1} = e_1$, and $\epsilon_0 = e_2$.
- Define $e'_1 = \epsilon_{M-1}$, $e'_2 = \epsilon_M$.
- Define $b'_i = 1$ for $i \in \{s \ldots t\}$. Define $b'_i = b_i$ for $i \notin \{s \ldots t\}$
- Define $r' = t$
- Define

$$\alpha' = \alpha + g(p) + \sum_{i=1}^{M} \log q(\epsilon_i | \epsilon_{i-2}, \epsilon_{i-1}) + \eta \times |r + 1 - s|$$

The Equality Function

- The function

  $$\mathsf{eq}(q, q')$$

  returns true or false.

- Assuming $q = (e_1, e_2, b, r, \alpha)$, and $q' = (e'_1, e'_2, b', r', \alpha')$, $eq(q, q')$ is true if and only if $e_1 = e'_1$, $e_2 = e'_2$, $b = b'$ and $r = r'$.

The Decoding Algorithm

- Inputs: sentence $x_1 \ldots x_n$. Phrase-based model $(\mathcal{L}, h, d, \eta)$. The phrase-based model defines the functions $ph(q)$ and next(q, p).
- Initialization: set $Q_0 = \{q_0\}$, $Q_i = \emptyset$ for $i = 1 \ldots n$.
- For $i = 0 \ldots n - 1$
    - For each state $q \in$ beam(Q_i), for each phrase $p \in ph(q)$:
      (1) $q' =$ next(q, p)
      (2) Add(Q_i, q', q, p) where $i =$ len(q')
- Return: highest scoring state in Q_n. Backpointers can be used to find the underlying sequence of phrases (and the translation).

Definition of Add(Q, q', q, p)

- If there is some $q'' \in Q$ such that $eq(q'', q') =$ True:
    - If $\alpha(q') > \alpha(q'')$
        - $Q = \{q'\} \cup Q \setminus \{q''\}$
        - set $bp(q') = (q, p)$
    - Else return
- Else
    - $Q = Q \cup \{q'\}$
    - set $bp(q') = (q, p)$

Definition of beam(Q)

Define

$$\alpha^* = \max_{q \in Q} \alpha(q)$$

i.e., α^* is the highest score for any state in Q.

Define $\beta \geq 0$ to be the *beam-width* parameter
Then

$$\text{beam}(Q) = \{q \in Q : \alpha(q) \geq \alpha^* - \beta\}$$

The Decoding Algorithm

- Inputs: sentence $x_1 \ldots x_n$. Phrase-based model $(\mathcal{L}, h, d, \eta)$. The phrase-based model defines the functions $ph(q)$ and next(q, p).
- Initialization: set $Q_0 = \{q_0\}$, $Q_i = \emptyset$ for $i = 1 \ldots n$.
- For $i = 0 \ldots n - 1$
  - For each state $q \in$ beam(Q_i), for each phrase $p \in ph(q)$:
    (1) $q' =$ next(q, p)
    (2) Add(Q_i, q', q, p) where $i =$ len(q')
- Return: highest scoring state in Q_n. Backpointers can be used to find the underlying sequence of phrases (and the translation).