6.891: Lecture 13 (October 22nd, 2003)

Machine Translation Part IV

Announcements

- Philipp Koehn talk tomorrow:

> Advances in Statistical Machine Translation: Phrases, Noun Phrases and Beyond
> Speaker: Philipp Koehn
> Speaker Affiliation: Information Sciences Institute / Univ. of Southern California
>
> Date: 10-23-2003 Time: 2:30 PM - 3:30 PM Refreshments: 2:15 PM Location: NE43-518 / 200 Technology Square (Room 518)
> I will review the state of the art in statistical machine translation (SMT), present my dissertation work, and sketch out the research challenges of syntactically structured statistical machine translation.
>
> The best methods currently in SMT build on the translation of phrases (any sequence of words) instead of single words. Phrase translation pairs are automatically learned from parallel corpora. While SMT systems generate translation output that often conveys a lot of the meaning of the original text, it is frequently ungrammatical and incoherent.
>
> The research challenge at this point is to introduce syntactic knowledge to the state of the art in order to improve translation quality. My approach breaks up the translation process along linguistic lines. I will present my thesis work on noun phrase translation and ideas about clause structure.

- I'll post an announcement about projects before the next lecture: please arrange a meeting with me to discuss possible projects

Overview

- Syntax Based Model 1: (Yamada and Knight 2001)
- Syntax Based Model 2: (Wu 1995)
- A Phrase-Based Model: (Koehn, Och and Marcu 2003)

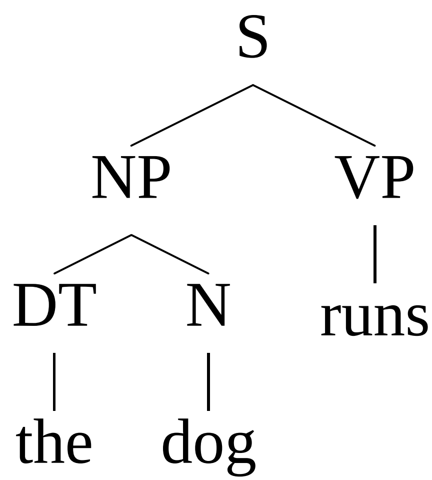
Methods that go beyond word-word alignments

(Yamada and Knight 2001)

- Task: English to Japanese translation
- IBM Models may be poor for languages with very different word orders?
- Task is Japanese $\rightarrow$ English translation,
  and we have an English parser
- Notation: as before we'll use $\mathbf{f}$ as the source language (was French, now Japanese), and $\mathbf{e}$ as the target language
- Notation: we'll use $\mathcal{E}$ to refer to an English **tree**

An Example $(\mathcal{E}, \mathrm{f})$ Pair

$\mathcal{E}$:



$\mathbf{f}$: arun athe adog anow

Preprocessing of the training set:
Parse all the English strings

Problems that Need to be Solved

- How to model $P(\mathbf{f} \mid \mathcal{E})$?
  i.e., how is a French **string** generated from an English **tree**?

- How do we train the parameters of the model?

- How do we decode with the model, i.e., find

$$\mathrm{argmax}_{\mathbf{e}} P(\mathbf{f} \mid \mathcal{E}) P(\mathbf{e})$$

  where $\mathbf{e}, \mathcal{E}$ is a sentence/tree pair in English?

How to model $P(\mathrm{f} \mid \mathrm{e})$?: Three Operations that Modify Trees

- **Reordering** operations
- **Insertion** of French words
- **Translation** of English words

Reordering Operations

- For each rule with n children, there are $n!$ possible reorderings
- For example, S $\rightarrow$ ADVP NP VP can be reordered in 6 possible ways

S	$\rightarrow$	ADVP	NP	VP
S	$\rightarrow$	ADVP	VP	NP
S	$\rightarrow$	NP	ADVP	VP
S	$\rightarrow$	NP	VP	ADVP
S	$\rightarrow$	VP	NP	ADVP
S	$\rightarrow$	VP	ADVP	NP

Reordering Operations

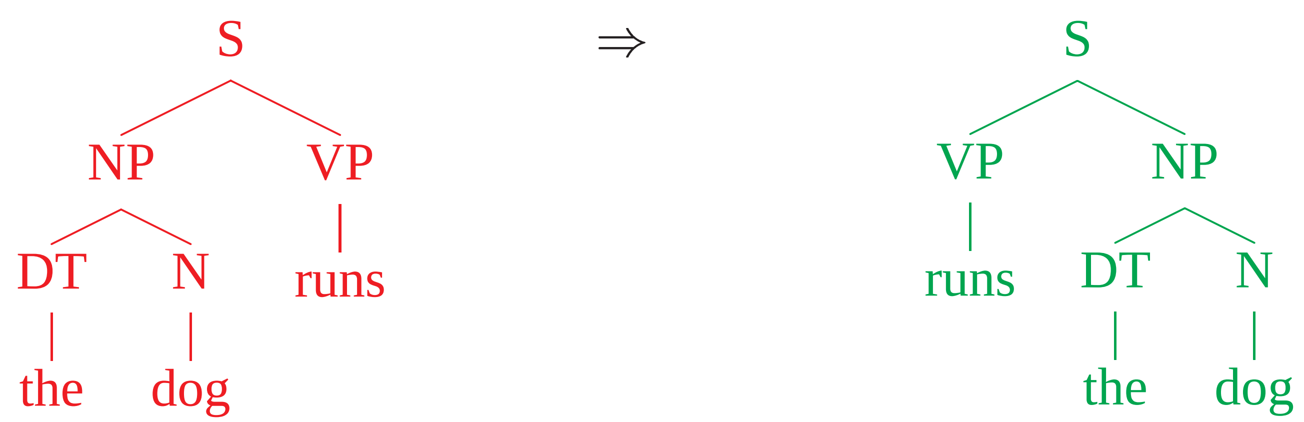
- Introduce $\rho(r' \mid r)$ as probability of r being reordered as r'
- For example,

$$\rho(\texttt{S} \rightarrow \texttt{VP ADVP NP} \mid \texttt{S} \rightarrow \texttt{ADVP NP VP})$$

- We now have a table of these probabilities for each rule:

r′					$\rho(r' \mid \texttt{S} \rightarrow \texttt{ADVP NP VP})$
S	→	ADVP	NP	VP	0.5
S	→	ADVP	VP	NP	0.1
S	→	NP	ADVP	VP	0.3
S	→	NP	VP	ADVP	0.03
S	→	VP	NP	ADVP	0.04
S	→	VP	ADVP	NP	0.03

An Example of Reordering Operations



Has probability:

$$\rho(\texttt{S} \rightarrow \texttt{VP NP} \mid \texttt{S} \rightarrow \texttt{NP VP}) \times$$
$$\rho(\texttt{NP} \rightarrow \texttt{DT N} \mid \texttt{NP} \rightarrow \texttt{DT N})$$
$$\rho(\texttt{DT} \rightarrow \texttt{the} \mid \texttt{DT} \rightarrow \texttt{the})$$
$$\rho(\texttt{N} \rightarrow \texttt{dog} \mid \texttt{N} \rightarrow \texttt{dog})$$
$$\rho(\texttt{VP} \rightarrow \texttt{runs} \mid \texttt{VP} \rightarrow \texttt{runs})$$

Note: Unary rules can only "reorder" in one way, with probability 1
e.g., $\rho(\texttt{VP} \rightarrow \texttt{runs} \mid \texttt{VP} \rightarrow \texttt{runs}) = 1$

Insertion Operations

- At any node in the tree, we can either:

  – Generate no "inserted" foreign words
  e.g., has probability

  $$\mathbf{I_1}(none \mid NP, S)$$

  here NP is the node in the tree, S is its parent

  – Generate an inserted foreign word to the left of the node
  e.g., has probability

  $$\mathbf{I_1}(left \mid NP, S)\mathbf{I_2}(anow)$$

  here NP is the node in the tree, S is its parent, and $anow$ is inserted to the left of the node

– Generate an inserted foreign word to the right of the node

$$\mathbf{I_1}(right \mid NP, S)\mathbf{I_2}(anow)$$

here NP is the node in the tree, S is its parent, and $anow$ is inserted to the right of the node

An Example of Insertion Operations

Has probability:

$$
\begin{aligned}
&\mathbf{I_1}(right \mid NP, S) \times \mathbf{I_2}(anow) \times \\
&\mathbf{I_1}(none \mid S, TOP) \times \\
&\mathbf{I_1}(none \mid VP, S) \times \\
&\mathbf{I_1}(none \mid runs, VP) \times \\
&\mathbf{I_1}(none \mid DT, NP) \times \\
&\mathbf{I_1}(none \mid N, NP) \times \\
&\mathbf{I_1}(none \mid the, DT) \times \\
&\mathbf{I_1}(none \mid dog, N)
\end{aligned}
$$

Translation Operations

For each English word, translate it to French word f with probability $\mathbf{T}(f \mid e)$ (note that f can be $NULL$)

Has probability:

$\mathbf{T}(aruns \mid runs) \times \mathbf{T}(athe \mid the) \times \mathbf{T}(adog \mid dog)$

Summary: Three Operations that Modify Trees

- The three operations:
  - **Reordering** operations with parameters ρ
  - **Insertion** of French words with parameters $\mathbf{I_1}$, $\mathbf{I_2}$
  - **Translation** of English words with parameters $\mathbf{T}$

- In this case, the **alignment** $\mathbf{a}$ is the sequence of reordering, insertion and translation operations used to build $\mathbf{f}$

- We have a model of $P(\mathbf{f}, \mathbf{a} \mid \mathcal{E})$

- Note that each $(\mathcal{E}, \mathbf{f})$ pair may have many possible alignments

- Two questions:

   1. How do we train the ρ, $\mathbf{I_1}$, $\mathbf{I_2}$, $\mathbf{T}$ parameters?
   2. How do we find

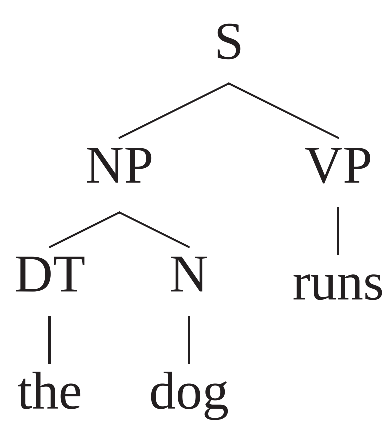
$$\text{argmax}_{\mathcal{E},\mathbf{e},\mathbf{a}} P(\mathbf{f},\mathbf{a} \mid \mathcal{E}) P(\mathbf{e})$$

   where $(\mathcal{E}, \mathbf{e}, \mathbf{a})$ is an English tree, sentence, alignment triple?

The translation problem:

Input: arun athe adog anow

Output:



A Slightly Simpler Translation Problem

- For now, instead of trying to find

$$\text{argmax}_{\mathcal{E},\mathbf{e},\mathbf{a}} P(\mathbf{f}, \mathbf{a} \mid \mathcal{E}) P(\mathbf{e})$$

  we'll consider a method that finds

$$\text{argmax}_{\mathcal{E},\mathbf{e},\mathbf{a}} P(\mathbf{f}, \mathbf{a} \mid \mathcal{E})$$

  (no language model)

- This can be done by transforming our model into a probabilistic context-free grammar, then parsing the French sentence using dynamic programming!!!

Constructing a PCFG

- For each English/French word pair (e, f), construct rules

  `e` $\rightarrow$ `f`

  with probabilities $\mathbf{T}(f \mid e)$

- For example, `dog` $\rightarrow$ `adog` with probability $\mathbf{T}(adog \mid dog)$

- Also construct rules

  `e` $\rightarrow$ ϵ

  with probabilities $\mathbf{T}(NULL \mid e)$ (where ϵ is the empty string)

Constructing a PCFG

- For every pair of non-terminals construct rules such as

  `NP-S` $\rightarrow$ `NP` with probability $\mathbf{I_1}(none \mid NP, S)$
  `NP-S` $\rightarrow$ `INS NP` with probability $\mathbf{I_1}(left \mid NP, S)$
  `NP-S` $\rightarrow$ `NP INS` with probability $\mathbf{I_1}(right \mid NP, S)$

- Also, for every French word f that can be inserted, construct rules such as

  `INS` $\rightarrow$ `f` with probability $\mathbf{I_2}(f)$

  e.g.,

  `INS` $\rightarrow$ `anow` with probability $\mathbf{I_2}(anow)$

Constructing a PCFG

- For every rule in English r, for every reordering of r, construct following rules
(example with $r = \texttt{S} \rightarrow \texttt{ADVP NP VP}$,
$r' = \texttt{S} \rightarrow \texttt{VP ADVP NP}$)

$\texttt{S} \rightarrow \texttt{S(ADVP,NP,VP)}$ with probability 1

$\texttt{S(ADVP,NP,VP)} \rightarrow \texttt{VP-S ADVP-S NP-S}$
with probability $\rho(\texttt{S} \rightarrow \texttt{VP ADVP NP} \mid \texttt{S} \rightarrow \texttt{ADVP NP VP})$

Constructing a PCFG

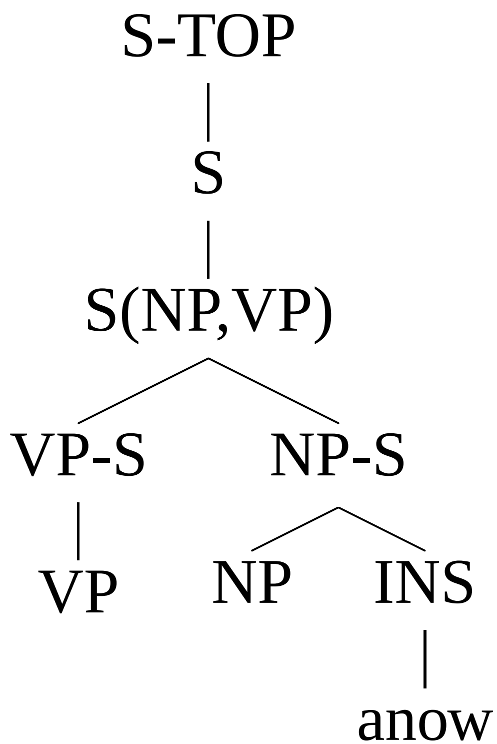
- Finally, for every non-terminal X, construct a start symbol

  ```
  X-TOP
  ```

  for example,

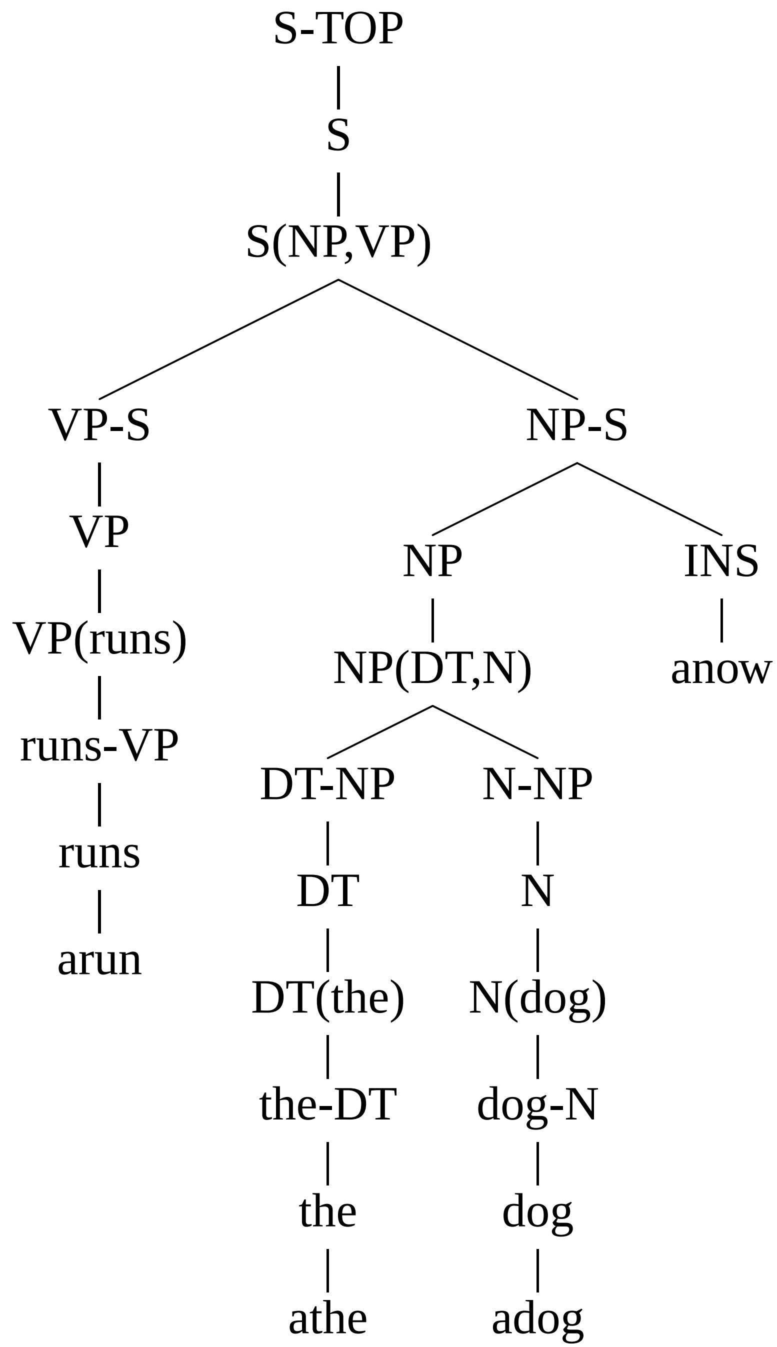
  ```
  S-TOP
  ```

An example:



This subtree has probability:

$\mathbf{I_1}(none \mid S, TOP) \times \rho(\texttt{S} \rightarrow \texttt{VP NP} \mid \texttt{S} \rightarrow \texttt{NP VP}) \times$
$\mathbf{I_1}(none \mid VP, S) \times \mathbf{I_1}(right \mid NP, S) \times \mathbf{I_2}(anow)$


S-TOP
S
S(NP,VP)
VP-S
NP-S
VP
NP
INS
VP(runs)
NP(DT,N)
anow
runs-VP
DT-NP
N-NP
runs
DT
N
arun
DT(the)
N(dog)
the-DT
dog-N
the
dog
athe
adog

Other Points

- Once we've constructed the PCFG, finding the most likely parse for a French string $\rightarrow$ finding the most likely English parse tree, English string, and alignment

- The model can be trained using EM:
  dynamic programming approach is possible

- Can parse a French sentence to produce a **forest**:
  a compact representation of all possible English translations

- A trigram language model can be used to pick the highest scoring string from the forest (although I'm not sure about the computational complexity of this...)

- (Yamada and Knight 2002) describe newer models

Overview

- Syntax Based Model 1: (Yamada and Knight 2001)
- Syntax Based Model 2: (Wu 1995)
- A Phrase-Based Model: (Koehn, Och and Marcu 2003)

Methods that go beyond word-word alignments

(Wu 1995)

- Standard probabilistic context-free grammars:
  probabilities over rewrite rules define probabilities over trees, strings, in one language

- **Transduction grammars**:
  Simultaneously generate strings in two languages

A Probabilistic Context-Free Grammar

S	$\Rightarrow$	NP	VP	1.0
VP	$\Rightarrow$	Vi		0.4
VP	$\Rightarrow$	Vt	NP	0.4
VP	$\Rightarrow$	VP	PP	0.2
NP	$\Rightarrow$	DT	NN	0.3
NP	$\Rightarrow$	NP	PP	0.7
PP	$\Rightarrow$	P	NP	1.0

Vi	$\Rightarrow$	sleeps	1.0
Vt	$\Rightarrow$	saw	1.0
NN	$\Rightarrow$	man	0.7
NN	$\Rightarrow$	woman	0.2
NN	$\Rightarrow$	telescope	0.1
DT	$\Rightarrow$	the	1.0
IN	$\Rightarrow$	with	0.5
IN	$\Rightarrow$	in	0.5

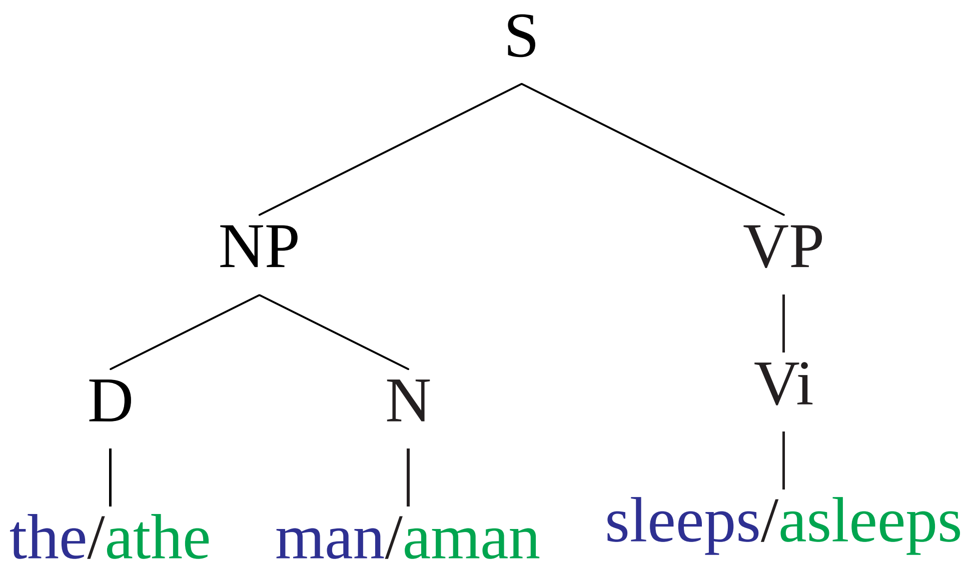
- Probability of a tree with rules $\alpha_i \rightarrow \beta_i$ is $\prod_i P(\alpha_i \rightarrow \beta_i | \alpha_i)$

Transduction PCFGs

- First change to the rules: **lexical** rules generate a pair of words

Vi	$\Rightarrow$	sleeps/asleeps	1.0
Vt	$\Rightarrow$	saw/asaw	1.0
NN	$\Rightarrow$	man/aman	0.7
NN	$\Rightarrow$	woman/awoman	0.2
NN	$\Rightarrow$	telescope/atelescope	0.1
DT	$\Rightarrow$	the/athe	1.0
IN	$\Rightarrow$	with/awith	0.5
IN	$\Rightarrow$	in/ain	0.5

<u>Transduction PCFGs</u>



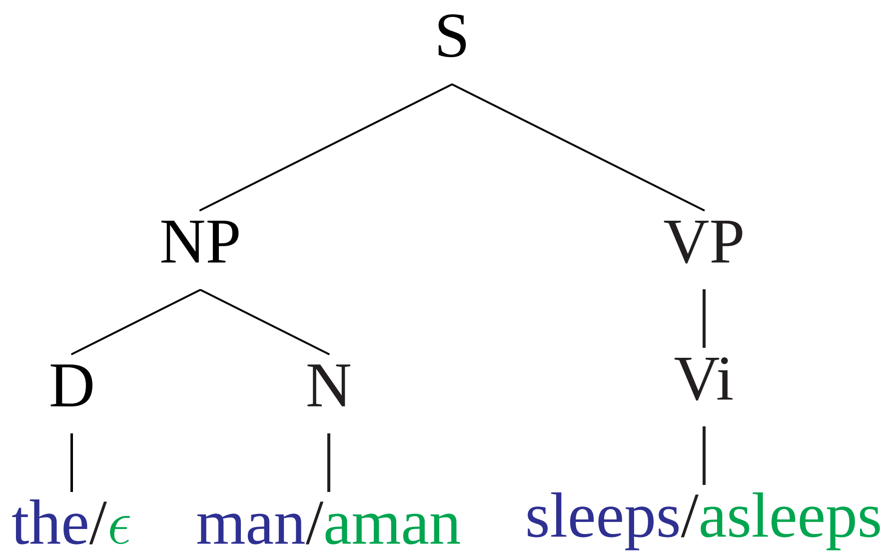
- The modified PCFG gives a distribution over $(\mathbf{f}, \mathbf{e}, T)$ triples, where $\mathbf{e}$ is an English string, $\mathbf{f}$ is a French string, and T is a tree

Transduction PCFGs

- Another change: allow empty string ϵ to be generated in either language, e.g.,

DT	$\Rightarrow$	the/ϵ	1.0
IN	$\Rightarrow$	ϵ/awith	0.5

Transduction PCFGs



- Allows strings in the two languages to have different lengths

  the man sleeps $\Rightarrow$ aman asleeps

Transduction PCFGs

- Final change: currently formalism does not allow different word orders in the two languages

- Modify the method to allow two types of rules, for example

$$
\begin{array}{lllll}
S & \Rightarrow & [NP \quad VP] & & 0.7 \\
S & \Rightarrow & \langle NP \quad VP \rangle & & 0.3
\end{array}
$$

- Define:
  - E_X is the English string under non-terminal X
    e.g., E_{NP} is the English string under the NP
  - F_X is the French string under non-terminal X

- Then for $\mathsf{S} \Rightarrow [\mathtt{NP}\ \ \mathtt{VP}]$ we define

$$\begin{aligned} E_S &= E_{NP}.E_{VP} \\ F_S &= F_{NP}.F_{VP} \end{aligned}$$

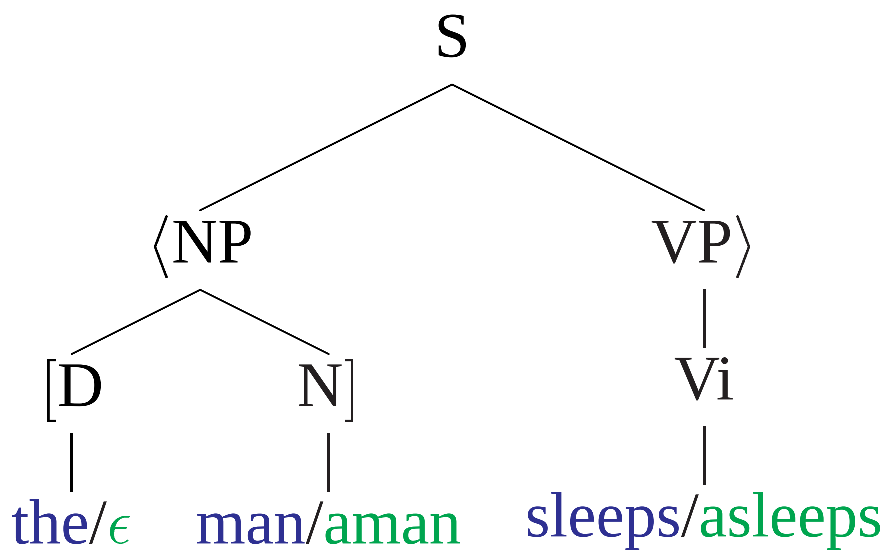
  where . is concatentation operation

- For $\mathsf{S} \Rightarrow \langle\mathtt{NP}\ \ \mathtt{VP}\rangle$ we define

$$\begin{aligned} E_S &= E_{NP}.E_{VP} \\ F_S &= F_{VP}.F_{NP} \end{aligned}$$

In the second case, the string order in French is reversed

Transduction PCFGs



- This tree represents the correspondance

  the man sleeps ⇒ asleeps aman

A Transduction PCFG

S	$\Rightarrow$	$[$NP	VP$]$	0.7
S	$\Rightarrow$	$\langle$NP	VP$\rangle$	0.3
VP	$\Rightarrow$	Vi		0.4
VP	$\Rightarrow$	$[$Vt	NP$]$	0.01
VP	$\Rightarrow$	$\langle$Vt	NP$\rangle$	0.79
VP	$\Rightarrow$	$[$VP	PP$]$	0.2
NP	$\Rightarrow$	$[$DT	NN$]$	0.55
NP	$\Rightarrow$	$\langle$DT	NN$\rangle$	0.15
NP	$\Rightarrow$	$[$NP	PP$]$	0.7
PP	$\Rightarrow$	$\langle$P	NP$\rangle$	1.0

Vi	$\Rightarrow$	sleeps/ϵ	0.4
Vi	$\Rightarrow$	sleeps/asleeps	0.6
Vt	$\Rightarrow$	saw/asaw	1.0
NN	$\Rightarrow$	ϵ/aman	0.7
NN	$\Rightarrow$	woman/awoman	0.2
NN	$\Rightarrow$	telescope/atelescope	0.1
DT	$\Rightarrow$	the/athe	1.0
IN	$\Rightarrow$	with/awith	0.5
IN	$\Rightarrow$	in/ain	0.5

(Wu 1995)

- Dynamic programming algorithms exist for "parsing" a pair of English/French strings (finding most likely tree underlying an English/French pair). Runs in $O(|\mathbf{e}|^3|\mathbf{f}|^3)$ time.

- Training the model: given $(\mathbf{e}_k, \mathbf{f}_k)$ pairs in training data, the model gives

$$P(T, \mathbf{e}_k, \mathbf{f}_k \mid \Theta)$$

where T is a tree, Θ are the parameters. Also gives

$$P(\mathbf{e}_k, \mathbf{f}_k \mid \Theta) = \sum_T P(T, \mathbf{e}_k, \mathbf{f}_k \mid \Theta)$$

Likelihood function is then

$$L(\Theta) = \sum_k \log P(f_k, e_k \mid \Theta) = \sum_k \log \sum_T P(T, f_k, e_k \mid \Theta)$$

Wu gives a dynamic programming implementation for EM

Overview

- Syntax Based Model 1: (Yamada and Knight 2001)
- Syntax Based Model 2: (Wu 1995)
- A Phrase-Based Model: (Koehn, Och and Marcu 2003)

Methods that go beyond word-word alignments

A Phrase-Based Model

(Koehn, Och and Marcu 2003)

- Intuition: IBM models have word-word translation
- Intuition: in IBM models each French word is aligned with only one English word
- A new type of model:
  align phrases in English with phrases in French

- An example from Koehn and Knight tutorial:

  Morgen fliege ich nach Kanada zur Konferenz

  Tomorrow I will fly to the conference in Canada

Morgen	Tomorrow
fliege	will fly
ich	I
nach Kanada	in Canada
zur Konderenz	to the conference

Representation as Alignment "Matrix"

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

(Note: "bof"' = "bofetada")

(Another example from the Koehn and Knight tutorial)

The Issues Involved

- Finding alignment matrices for all English/French pairs in training corpora

- Coming up with a model that incorporates phrases

- Training the model

- Decoding with the model

Finding Alignment Matrices

- Step 1: train model 4 for $P(f \mid e)$, and come up with most likely alignment for each (e, f) pair
- Step 2: train model 4 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 was 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 Issues Involved

- Finding alignment matrices for all English/French pairs in training corpora
- Coming up with a model that incorporates phrases
- Training the model
- Decoding with the model

The Model

- The probability model again models $P(\mathbf{f} \mid \mathbf{e})$
- The steps:
  - Choose a segmentation of $\mathbf{e}$ (all segmentations are equally likely)
  - For each English phrase e, choose a French phrase f with probability

    $$\mathbf{T}(f \mid e)$$

    for example

    $$\mathbf{T}(\text{daba una bofetada} \mid \text{slap})$$

  - Choose positions for the French phrases: if start position of the i'th French phases is a_i, and end point of $(i-1)$'th French phrase is b_{i-1}, then this has probability

    $$\mathbf{R}(a_i - b_{i-1})$$

Training the Model

Simple once we have the alignment matrices!:

- Take maximum-likelihood estimates, e.g.,

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

- Take similar estimates for the alignment probabilities

The Issues Involved

- Finding alignment matrices for all English/French pairs in training corpora

- Coming up with a model that incorporates phrases

- Training the model

- Decoding with the model

The Decoding Method

- Goal is to find a high probability English string $\mathbf{e}$ under

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

where

$$P(\mathbf{f}, \mathbf{a} \mid \mathbf{e}) = \prod_{i=1}^{n} \mathbf{T}(f_i \mid e_i)\mathbf{R}(a_i - b_{i-1})$$

where f_i and e_i are the n *phrases* in the alignment,
a_i and b_i are start/end points of the i'th phrase

The Decoding Method

- A **partial hypothesis** is an English prefix, aligned with some of the French sentence

```
Maria      no     daba una bofetada a la bruja verde
  |         |
Mary     did not
```

- S_m is a **stack** which stores n most likely partial analyses that account for m French words

- At each point, pick a partial hypothesis, and **advance** it by choosing a substring of the French string

Maria no daba una bofetada a la bruja verde

Mary did not

$$\Downarrow$$

Maria no daba una bofetada a la bruja verde

Mary did not slap

- In this case, we create a new member of the stack S_5