ACS Statistical Machine Translation

Lecture 9: Hierarchical Translation Grammars, Syntax-based Translation

Department of Engineering
University of Cambridge

Bill Byrne
bill.byrne@eng.cam.ac.uk

Lent 2013

Outline

- Lecture 8: we described **Hierarchical Phrase-based Translation**
  - efficient implementation with **WFSTs**
- Today we will talk about the **Translation Grammar** that is used under this framework
  - Rule Extraction
  - Grammar Redundancy and Overgeneration
  - Role of non-terminals
  - Shallow Grammars
- We will also introduce **Syntax-based Translation** using the same decoding principle

Building the Rule Set

Hierarchical rules are extracted from word-aligned parallel text

- Similarly to the phrase-based approach, standard constraints apply [1]:
  - maximum number of non-terminals is two
  - disallow adjacent non-terminals in the source language
  - unaligned words are not allowed at the edges of the rule
  - require at least one pair of aligned words per rule
- Types of rules that are extracted include:

$X \rightarrow \langle$**source,target**$\rangle$	**times**
$X \rightarrow \langle w,w \rangle$	27759863
$X \rightarrow \langle w\ X\ w,w\ X\ w \rangle$	1715310
$X \rightarrow \langle w\ X,X\ w \rangle$	628313
$X \rightarrow \langle X\ w,w\ X \rangle$	581122
$X \rightarrow \langle X_2\ w\ X_1,X_1\ X_2\ w \rangle$	484803
$X \rightarrow \langle X_2\ w\ X_1,w\ X_1\ X_2 \rangle$	483146
$X \rightarrow \langle X_2\ w\ X_1,X_1\ w\ X_2 \rangle$	186616
$X \rightarrow \langle X\ w,w\ X\ w \rangle$	156697
$X \rightarrow \langle X_2\ w\ X_1,X_1\ w\ X_2\ w \rangle$	147650
$X \rightarrow \langle X_2\ w\ X_1,w\ X_1\ w\ X_2 \rangle$	147443
$X \rightarrow \langle w\ X_1\ w\ X_2,w\ X_1\ X_2 \rangle$	65383
$X \rightarrow \langle w\ X,w\ X\ w \rangle$	62633
$X \rightarrow \langle w\ X\ w,w\ X \rangle$	60112
$X \rightarrow \langle X_1\ w\ X_2\ w,X_1\ w\ X_2 \rangle$	32782
etc...	...

[1] D. Chiang, 2005. A Hierarchical Phrase-Based Model for Statistical Machine Translation. Proc. ACL.

Translation as Parsing under a Synchronous Grammar

Parsing is the syntactic analysis of a sentence,
i.e. finding the **grammatical structure** of a sentence.

- Given a context-free grammar (CFG)
  - e.g. with rules $\langle\ V \rightarrow w\ \rangle$ and $\langle\ V \rightarrow w\ V\ \rangle$
- We want to know if we can generate the sentence given the grammar of rules

With **Hierarchical Translation**, rules are bilingual

- we have a synchronous probabilistic CFG
- both languages have the same number of non-terminals(gaps)
- we will parse the source language
- and automatically generate the target
- apply a language model to the target

Full Hierarchical Grammar

Formally it contains the following rules, where $\mathbf{T}$ is the set of terminals (words).

full hierarchical grammar	
$S \rightarrow \langle X, X \rangle$	glue rule
$S \rightarrow \langle S\ X, S\ X \rangle$	glue rule
$X \rightarrow \langle \gamma, \alpha, \sim \rangle\ ,\ \gamma, \alpha \in \{X \cup \mathbf{T}\}^+$	hiero rules of any level

- Leaving aside concatenation rules, all rules have the same non-terminal X on their left-hand side
- This allows plenty of rules to 'fit in each X gap', which means that many reorderings are possible

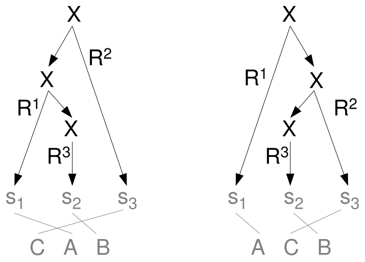
- In theory, rule nesting is unlimited
- In practice, there are limits imposed by:
  - which rules have been extracted from the parallel corpus used in training
  - which words occur in the source sentence to be translated, as at least one terminal is consumed by each hierarchical rule

Some Modelling Issues – Overgeneration

Overgeneration: different translations arising from the same set of rules

Translations of the source sequence $s_1\ s_2\ s_3$

R^1: $X \rightarrow \langle s_1\ X, A\ X \rangle$
R^2: $X \rightarrow \langle X\ s_3, X\ C \rangle$
R^3: $X \rightarrow \langle s_2, B \rangle$



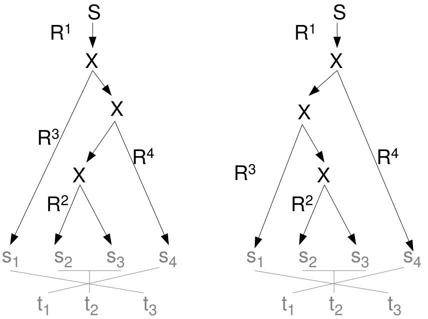
- not necessarily a bad thing in that new translations can be synthesized from rules extracted from training data
- a strong target language model, such as a high order n-gram, is typically relied upon to discard unsuitable hypotheses

Overgeneration complicates translation in that many hypotheses are introduced only to be subsequently discarded. These must be kept until the LM can be applied to discard them.

Some Modelling Issues – Spurious Ambiguity

Spurious ambiguity: *a situation where the decoder produces many derivations that are distinct yet have the same model feature vectors and give the same translation*

R^1: $S \rightarrow \langle X,X \rangle$
R^2: $X \rightarrow \langle \mathrm{s}_2\ \mathrm{s}_3,\mathrm{t}_2 \rangle$
R^3: $X \rightarrow \langle \mathrm{s}_1\ X,X\ \mathrm{t}_3 \rangle$
R^4: $X \rightarrow \langle X\ \mathrm{s}_4,\mathrm{t}_1\ X \rangle$

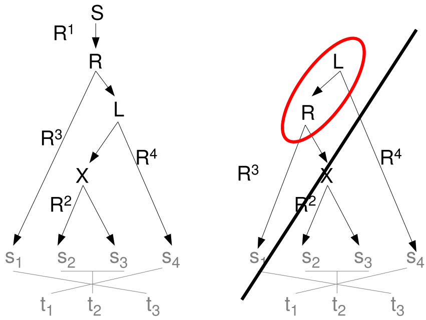


- the use of a single non-terminal X makes the grammar flexible, but redundant
- this can have a big impact in decoding time and memory requirements, even with efficient implementations

Some Modelling Issues – Addressing Grammar Redundancy

Some of the previous problems can be partly addressed with additional non-terminals

R¹: $S \rightarrow \langle \{X, R, L\}, \{X, R, L\} \rangle$
R²: $X \rightarrow \langle s_2\ s_3, t_2 \rangle$
R³: $R \rightarrow \langle s_1\ \{X, R, L\}, \{X, R, L\}\ t_3 \rangle$
R⁴: $L \rightarrow \langle \{X, L\}\ s_4, t_1\ \{X, L\} \rangle$



- this is really tough when many different rule types are included!

Some Modelling Issues – Grammars and Language Pairs

Discussion on Hierarchical Grammar:

- rule concatenation (S rules) is very efficient
- computational complexity is due to rule nesting (X rules)
- rule nesting should be used for placing words in different order for each language (reordering), not for placing them in the same order (this is already done by the S rules)

Not all language pairs require the complete power of the Full Hierarchical Grammar

- we can introduce new non-terminals to limit rule nesting
- for a given language pair, grammar should be defined so that the word movement needed is allowed, and any extra excessive reordering disallowed

Shallow Hierarchical Grammars

Formally we can define the following grammar, where $\mathbf{T}$ is the set of terminals (words).

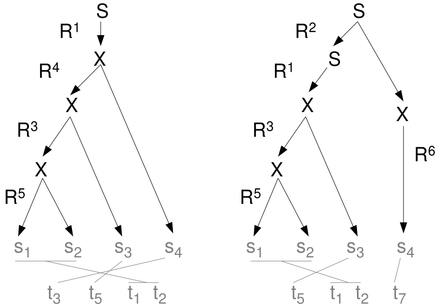
shallow-1 grammar	
S$\rightarrow\langle X$,$X\rangle$	glue rule
S$\rightarrow\langle$S X,S $X\rangle$	glue rule
$X\rightarrow\langle\gamma,\alpha,\sim\rangle$, $\gamma,\alpha \in \{\{V\}\cup\mathbf{T}\}^+$	hiero rules level 1
$V\rightarrow\langle\gamma^p,\alpha^p\rangle$, $\gamma^p,\alpha^p \in \mathbf{T}^+$	regular phrases

- For Arabic-to-English and Spanish-to-English, shallow-1 grammar performs similarly to full hiero - **but $\sim$20$\times$ faster**
- Constrained search space, but can be built exactly and quickly - **no pruning required**
- If full hiero could be searched without errors, we would expect minor translation quality improvements

Shallow Hierarchical Grammars (2)

Example:

R[1]: $S \rightarrow \langle X, X \rangle$
R[2]: $S \rightarrow \langle S\ X, S\ X \rangle$
R[3]: $X \rightarrow \langle X\ \mathrm{s}_3, \mathrm{t}_5\ X \rangle$
R[4]: $X \rightarrow \langle X\ \mathrm{s}_4, \mathrm{t}_3\ X \rangle$
R[5]: $X \rightarrow \langle \mathrm{s}_1\ \mathrm{s}_2, \mathrm{t}_1\ \mathrm{t}_2 \rangle$
R[6]: $X \rightarrow \langle \mathrm{s}_4, \mathrm{t}_7 \rangle$



- Tree on the left uses rule nesting twice, so it is not possible under shallow-1 grammar

Shallow Hierarchical Grammars (3)

Formally we can control the level of nesting we want with the following grammars, where $\mathbf{T}$ is the set of terminals (words).

shallow-N grammar	
S$\rightarrow\langle X^N, X^N\rangle$	glue rule
S$\rightarrow\langle$S X^N,S $X^N\rangle$	glue rule
$X^n\rightarrow\langle\gamma,\alpha,\sim\rangle$, $\gamma,\alpha \in \{\{X^{n-1}\}\cup\mathbf{T}\}^+$	hiero rules levels $n = 1,\ldots,N$
with the requirement that α and γ contain at least one X^{n-1}	
$X^0\rightarrow\langle\gamma,\alpha\rangle$, $\gamma,\alpha \in \mathbf{T}^+$	regular phrases

- In Arabic-to-English, shallow-2 grammar does not provide improvements over shallow-1
- In Chinese-to-English, shallow-3 grammar is better than shallow-1 or shallow-2
- Note that for n=1, this is equivalent to previous slide where $X^1 \equiv X$ and $X^0 \equiv V$

Practical 3/3

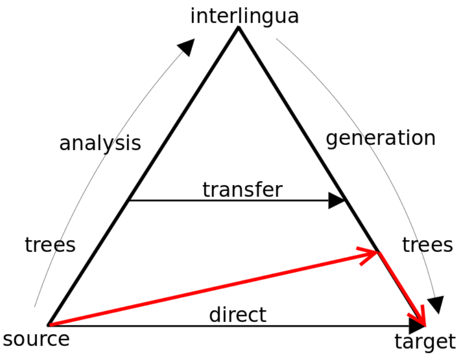
- Hierarchical Translation with alternative translation grammars
- Handout available at:

  `http://www.cl.cam.ac.uk/teaching/1213/L102/practicals//handout-3.pdf`
- Demonstrated Sessions: 4th and 11th March
- Answers to practical questions should be included in a single practical report to be handed at the end of term

Hierarchical Translation Grammars

Syntax-based SMT

String to Tree Translation [2]

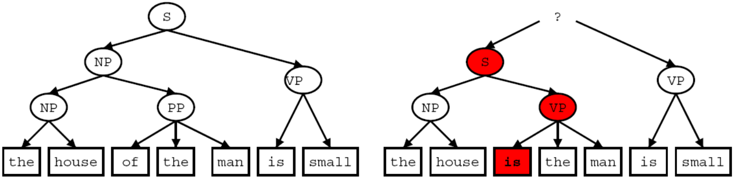


- exploit rich resources on the target side (usually English)
- include syntactic information intro translation unit
- induce a foreign tree via steps of **reordering, insertion and translation**
- use of a target-language **syntactic language model**

[2] Yamada, K. and Knight, K. 2001. A Syntax-based Statistical Translation Model. Proc. ACL

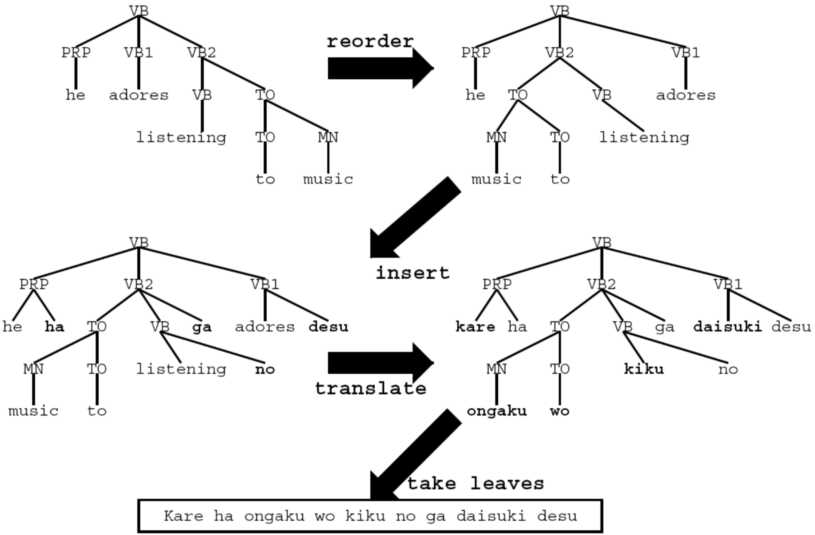
Syntactic Language Model

- Good syntax tree ⇒ good English sentence
- Allows for long-distance dependencies, unlike N-grams



✓ left translation is prefered by syntactic LM

Train Models from Parsed Corpus

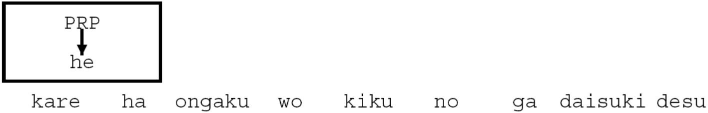


Reordering Table

Original Order	Reordering	p(reorder|original)
PRP VB1 VB2	PRP VB1 VB2	0.074
PRP VB1 VB2	**PRP VB2 VB1**	**0.723**
PRP VB1 VB2	VB1 PRP VB2	0.061
PRP VB1 VB2	VB1 VB2 PRP	0.037
PRP VB1 VB2	VB2 PRP VB1	0.083
PRP VB1 VB2	VB2 VB1 PRP	0.021
VB TO	VB TO	0.107
VB TO	**TO VB**	**0.893**
TO NN	TO NN	0.251
TO NN	**NN TO**	**0.749**

Decode as Parsing

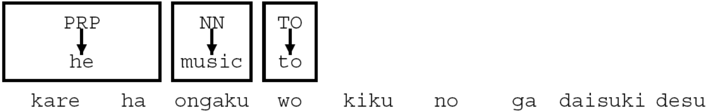
- chart parsing:



- pick Japanese words
- translate into tree stumps

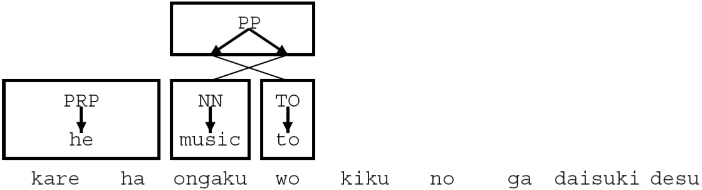
Decode as Parsing

- chart parsing:



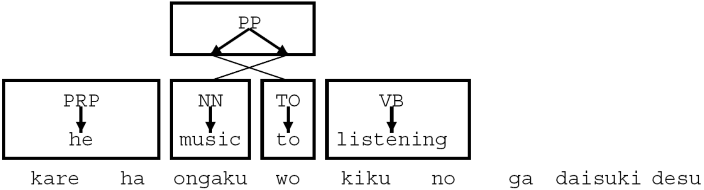
- pick Japanese words
- translate into tree stumps

Decode as Parsing



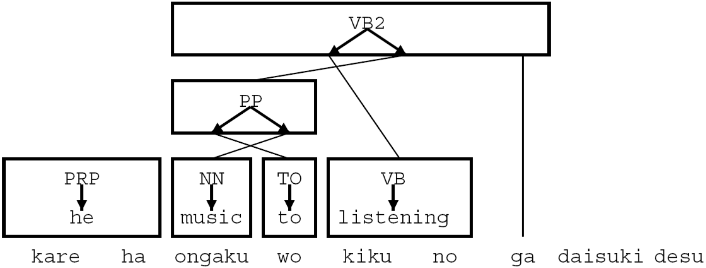
- add and combine more entries (reorder model)

Decode as Parsing

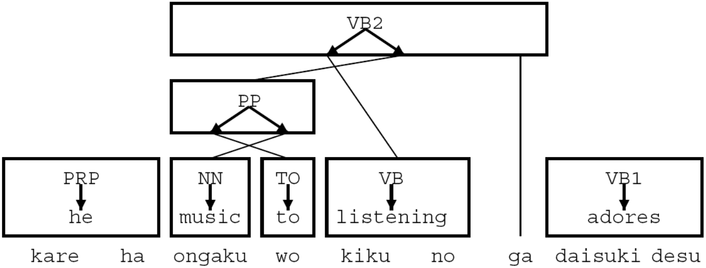


- add and combine more entries (reorder model)

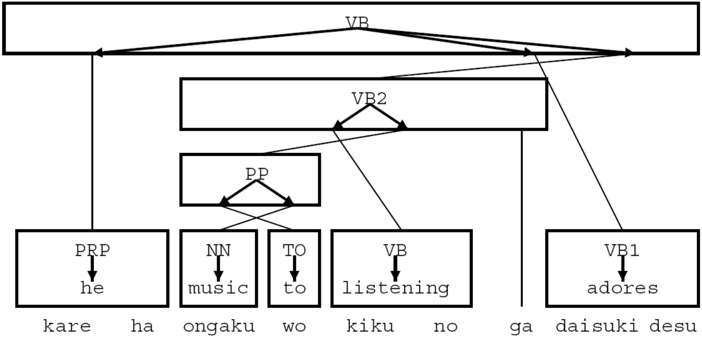
Decode as Parsing



Decode as Parsing



Decode as Parsing



- finished when all source words covered → target tree produced