

Universität Potsdam

Institut für Informatik
Lehrstuhl Maschinelles Lernen



Machine Translation

Uwe Dick

Google Translate

Google Übersetzer

Sofortübersetzung deaktivieren



Englisch Deutsch Französisch Englisch - erkannt



Deutsch Französisch Englisch

Übersetzen

Bayern Munich on Thursday suggested Mario Götze's playing future at the German champions may remain on the bench, even though he showed his desire to earn a starting spot after turning down a possible to Liverpool.

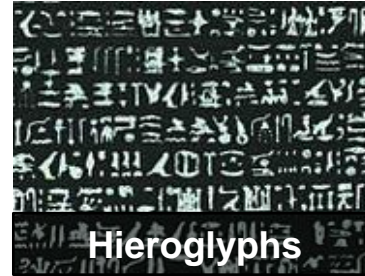
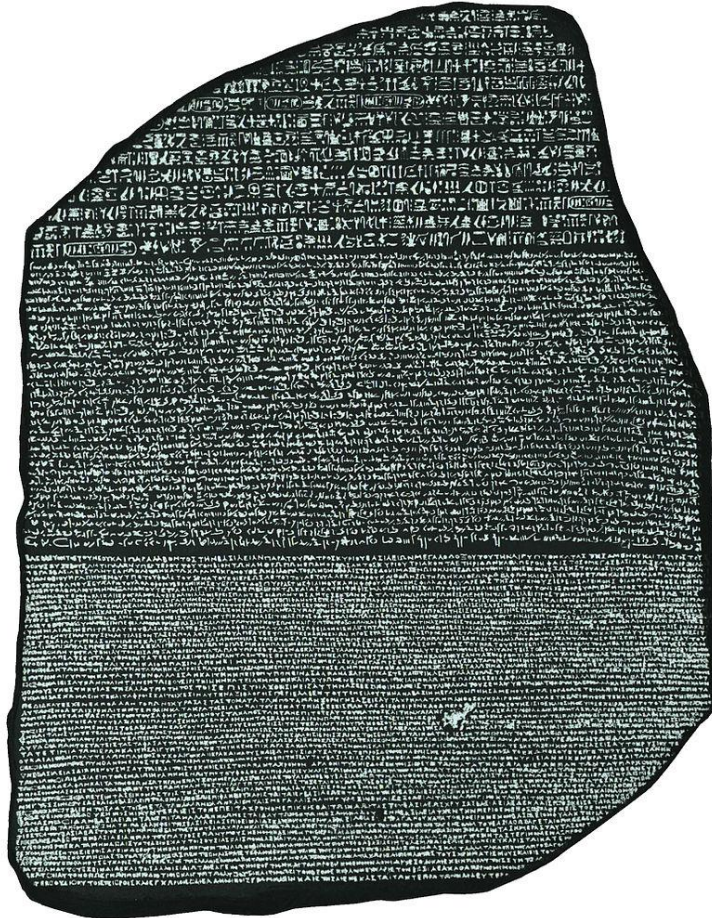
The Germany international, who scored the winning goal in the 2014 World Cup final, has failed to carve out a starting spot since joining Bayern from rivals Borussia Dortmund in 2013 and has been earmarked as a major summer target for Jürgen Klopp.

Bayern München am Donnerstag vorgeschlagen, Mario Götze des Spiel Zukunft bei den Deutschen Meister auf der Bank bleiben kann, obwohl er seinen Wunsch, zeigte einen Startplatz nach dem Ausdrehen eine mögliche nach Liverpool zu verdienen.

Die Deutschland Nationalspieler , der den Siegtreffer in der WM 2014 Finale erzielte, hat es versäumt, aus Bayern einen Startplatz zu schnitzen Rivalen Borussia Dortmund im Jahr 2013 seit dem Beitritt und hat als Haupt Sommer Ziel für Jürgen Klopp bestimmt worden.



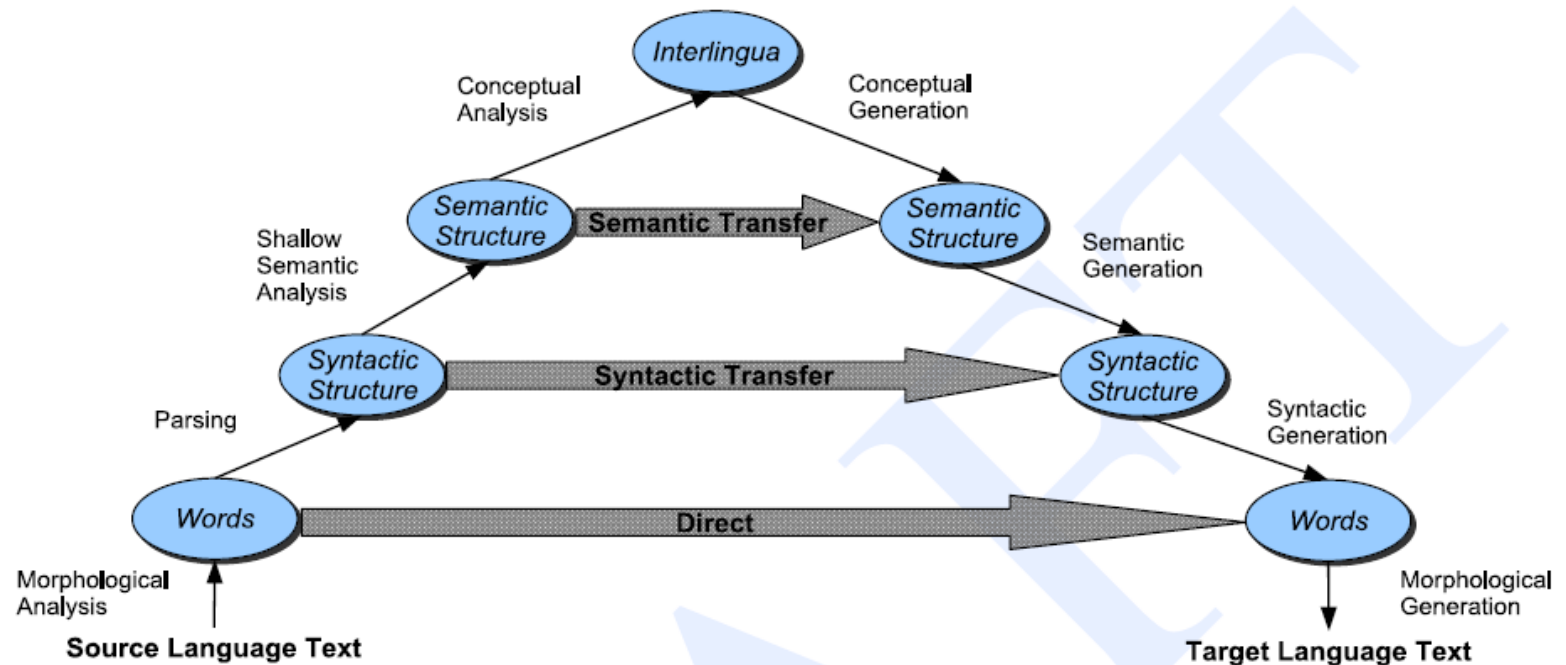
Rosetta Stone



Machine Translation

- Automatically translate text s in source language into text t in target language.
- Tasks and solutions we will elaborate on:
 - ◆ How should we model the translation problem using machine learning methods?
 - ◆ How can we learn such a model?
 - ◆ What is a good model?
 - ★ How can we measure the quality of a translation?

Machine Translation—Classical View



Statistical Machine Translation

- Problem formulation:
 - ◆ Given sentence in source language S ,
 - ◆ Find best sentence in target language T :

$$\operatorname{argmax}_T P(T|S)$$

- Generally, a model of some latent structure A is included (e.g. syntactic parse tree, word alignment. More on that later):
 - ◆ $\operatorname{argmax}_T \sum_A P(T, A|S)$
 - ◆ Sometimes, this is simplified to $\operatorname{argmax}_{T,A} P(T, A|S)$

Statistical Machine Translation

- Often, Bayes' rule is used to split the likelihood of target sentence given source sentence into
 - ◆ (inverse) translation model
 - ◆ and (target) language model.

$$\operatorname{argmax}_T P(T|S) = \operatorname{argmax}_T P(S|T)P(T)$$

Statistical Machine Translation

- You learned a variety of language models $P(T)$ last week.
- The two components can be identified with a
 - ◆ adequacy model $P(S|T)$ which (mainly) determines how much of the information of T is translated to S ,
 - ◆ fluency model $P(T)$ which determines aspects such clarity and naturalness of the target sentence.

Lexical Translation Models and Word Alignment

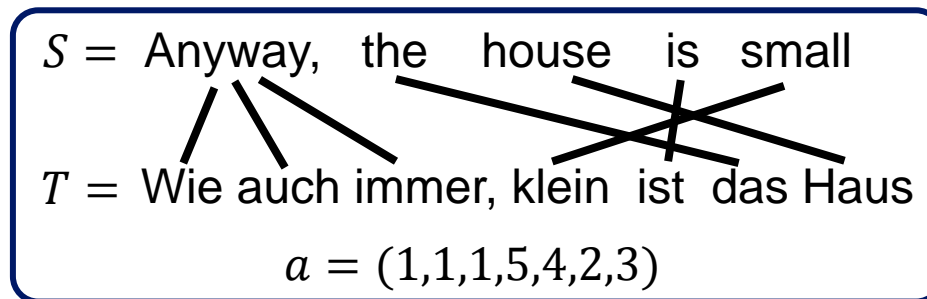
- Simple translation models model the translation process as a sequence of word-to-word translations $P(s|t)$ with possible reorderings of words in the target sentence.
- Reordering describes a word-to-word alignment a , e.g.
 - ◆ The i -th word t_i in T aligns with the a_i -th word s_{a_i} in S .

S	=	The	house	is	small
T	=	Das	Haus	ist	klein

$a = (1,2,3,4)$

Word Alignments

- Vectorial representation a can express one-to-many relations.
- One word s_i can be aligned to several words in T .



- Some words in S or T might not have any aligned words.

Lexical Translation Models

- A simple statistical translation model can model $P(T|S)$ as follows:

$$P(T|S) = \sum_a P(T, a|S)$$

and

$$P(T, a|S) = P(a|S) P(T|a, S)$$

- ◆ $P(A|S)$ can be very simple.

IBM Models

- Statistical alignment models that are also simple translation models.
- Very basic. Proposed in 1993.
- Translate sentence $S = s_1 \dots s_K$ into another language $T = t_1 \dots t_I$.
- Or: Align sentence $S = s_1 \dots s_K$ with sentence $T = t_1 \dots t_I$.

IBM Models

- $P(T|S) = \sum_a P(T, a|S)$
- $P(T, a|S) = P(a|S) P(T|a, S)$
$$= P(a|S) \prod_k P(t_{a_k}|s_k)$$
- Models differ in choice of reordering model $P(a|S)$.

IBM Model 1

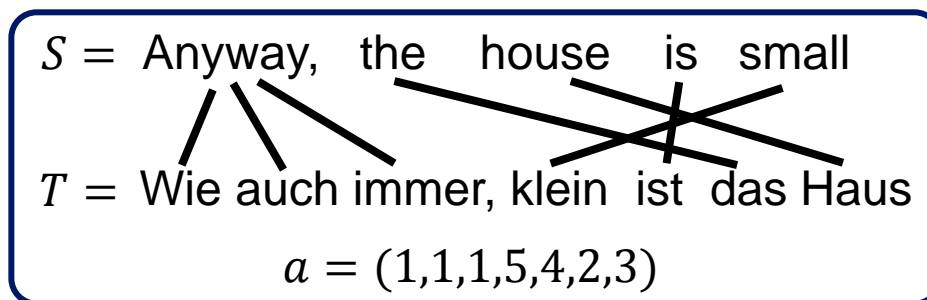
- IBM Model 1 models $P(a|S)$ as

$$P(a|S) = P_l(\text{length}(a)|S) \prod_k^{\text{length}(a)} P_a(a_k|S)$$

- ◆ $P(a_k|S)$ is uniform over all source words and a special NULL word.
- ◆ Length of a is also modeled as uniform distribution over some length interval.

Example Decoding

- $P(T|S) = \sum_a P(T, a|S)$
- Iterate over all alignments. Example:



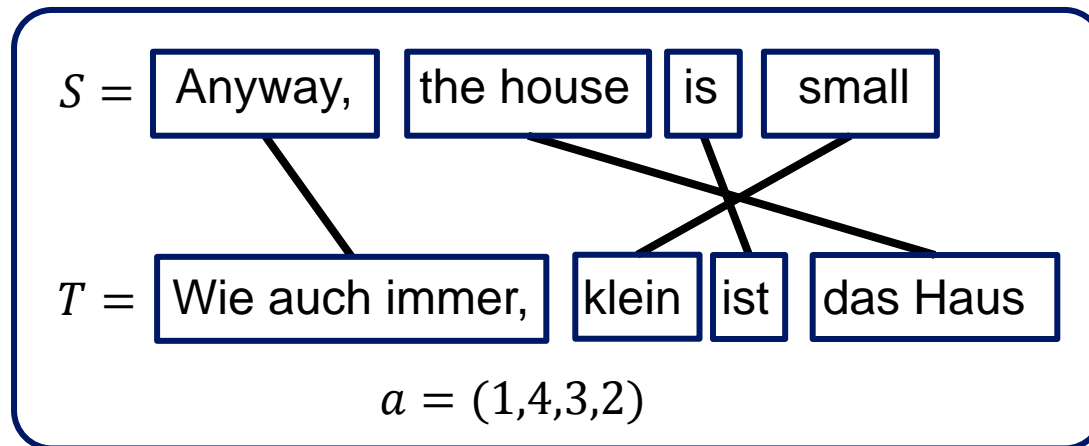
$$P(T, a|S) = P(a|S) \prod_k P(t_k | s_{a_k})$$

$$P(T, a|S)$$

$$\begin{aligned} &= P(a|S) \cdot P(\text{Wie} | \text{anyway}) \cdot P(\text{auch} | \text{anyway}) \\ &\cdot P(\text{immer} | \text{anyway}) \cdot P(\text{klein} | \text{small}) \cdot P(\text{ist} | \text{is}) \\ &\cdot P(\text{das} | \text{the}) \cdot P(\text{haus} | \text{house}) \end{aligned}$$

Phrase Based Translation

- Generally better to use phrases instead of single word translations.
- Here, phrase just means a sequence of words (substrings), without any linguistic meaning.
- Better suited for translation tasks



Phrase-Based Model

- Simple phrase based model:
 - ◆ Let $S = \bar{s}_1 \dots \bar{s}_m$ and $T = \bar{t}_1 \dots \bar{t}_m$ be phrase sequences.

- $P(S, a|T) \sim \prod_{i=1}^m P(\bar{s}_{a_i}|\bar{t}_i) d(\text{start}_{a_i} - \text{end}_{a_{i-1}})$

Reordering / distance score. E.g.
 $d(x) = \alpha^{|x-1|}$

Start position of phrase \bar{s}_{a_i}

End position of phrase $\bar{s}_{a_{i-1}}$

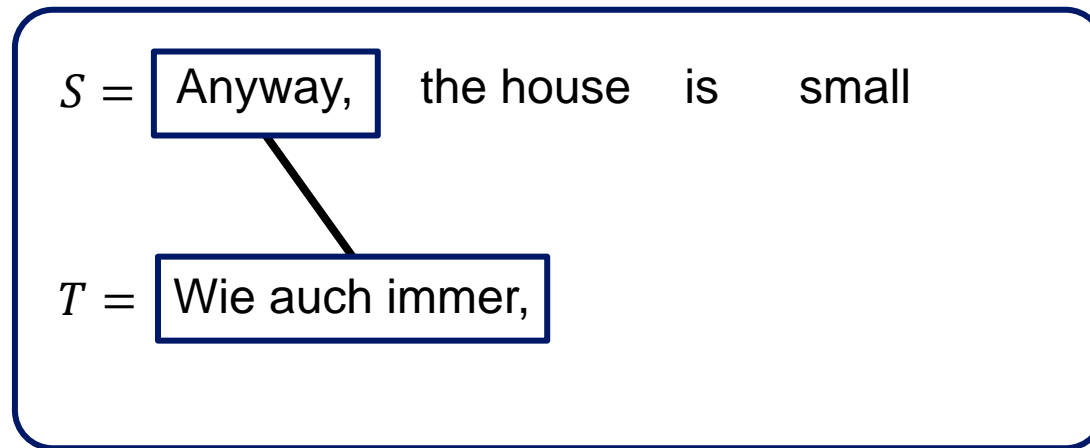
Phrase-Based Model

- This phrase-based model (Koen, 2003) can be defined by:
 - ◆ All phrase translation probabilities known to the model are stored in a *phrase table*.
 - ◆ A *language model* of the target language.
 - ◆ A reordering scoring function.
 - ◆ Potentially also a penalty function that penalizes long target sentences.

$$P(T, a|S) \sim \prod_{i=1}^m P(\overline{s_{a_i}} | \overline{t_i}) \cdot d(\text{start}_{a_i} - \text{end}_{a_{i-1}}) \cdot P_{lm}(T) \cdot W(\text{len}(T))$$

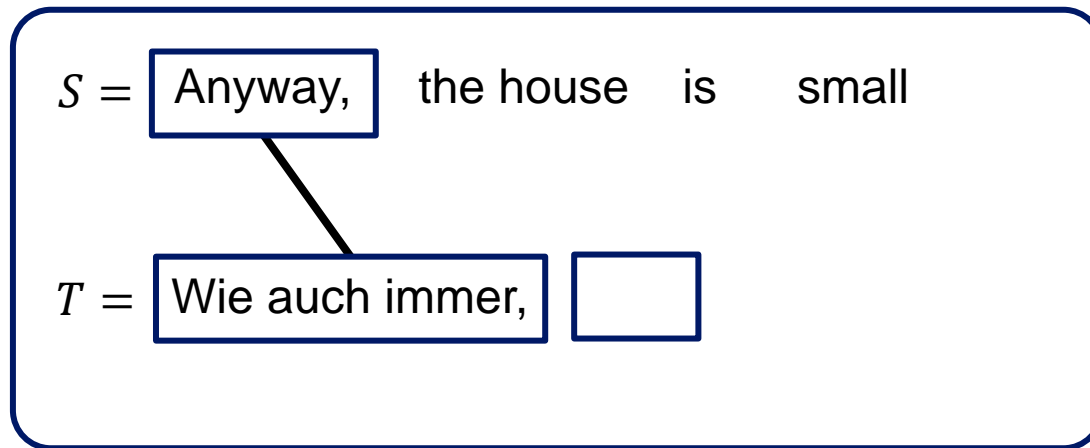
Decoding of Phrase-Based Models

- Find the best derivation out of exponentially many.
- Decoding as search.
- Example:



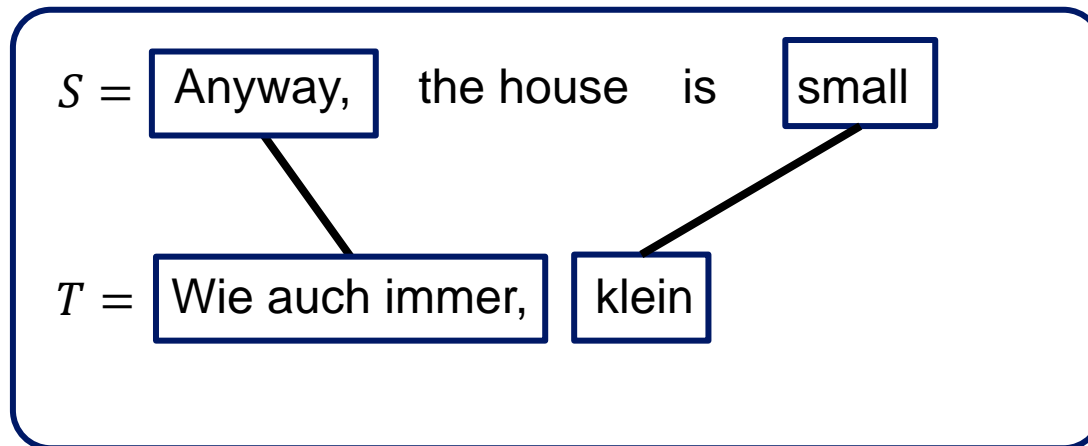
Decoding of Phrase-Based Models

- Find the best derivation out of exponentially many.
- Decoding as search.
- Example:



Decoding of Phrase-Based Models

- Find the best derivation out of exponentially many.
- Decoding as search.
- Example:



- ... and so on

Decoding as Search

- There are several possible translation options.

Anyway,	the	house	is	small
Wie auch immer	der	Haus	ist	klein
sowieso	die	Gebäude	entspricht	lütt
jedenfalls	das	Bank		winzig
überhaupt	des	Publikum		
	das Haus		ist klein	
	des Hauses		ist nicht der Rede wert	
	die Kammer			
	im Haus			

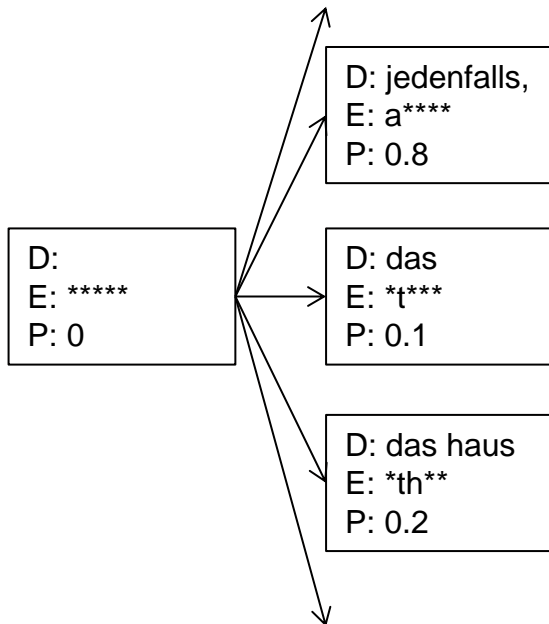
Decoding as Search

- Expand search nodes
 - ◆ P: Probability (score / neg. cost) of current state
 - ◆ E: aligned words in English sentence
 - ◆ D: last expanded word in German

D:
E: *****
P: 0

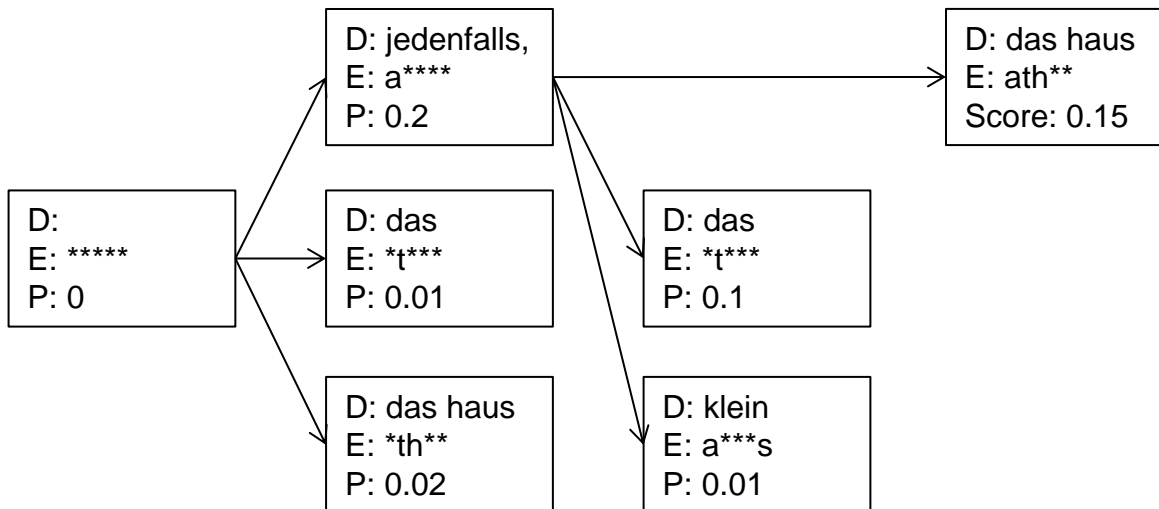
Decoding as Search

- Expand search nodes
 - ◆ P: Probability (score / neg. cost) of current state
 - ◆ E: aligned words in English sentence
 - ◆ D: last expanded word in German



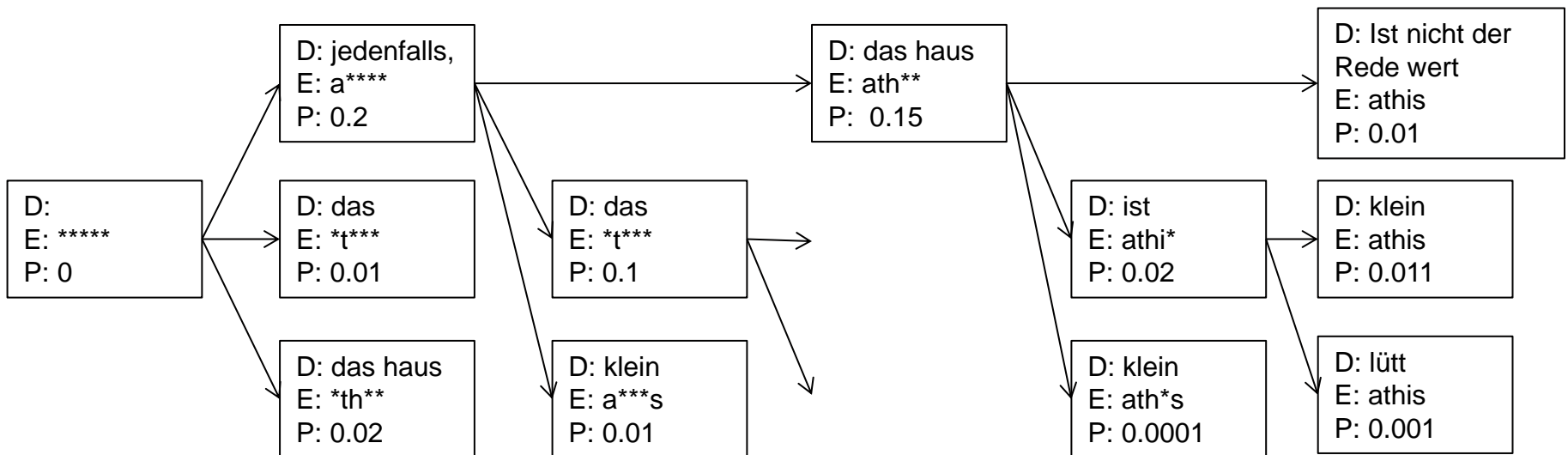
Decoding as Search

- Expand search nodes
- P: Probability (score / neg. cost) of current state
 - ◆ E.g. $P \sim P(\text{jedenfalls}|\text{anyway}) \cdot d(0) \cdot P(\text{klein}|\text{small}) \cdot d(3) \cdot P_{lm}(\text{jedenfalls klein})$



Decoding as Search

- Expand search nodes
- P: Probability (score / neg. cost) of current state
 - ◆ E.g. $P \sim P(\text{jedenfalls}|\text{anyway}) \cdot d(0) \cdot P(\text{klein}|\text{small}) \cdot d(3) \cdot P_{lm}(\text{jedenfalls klein})$



Decoding as Search

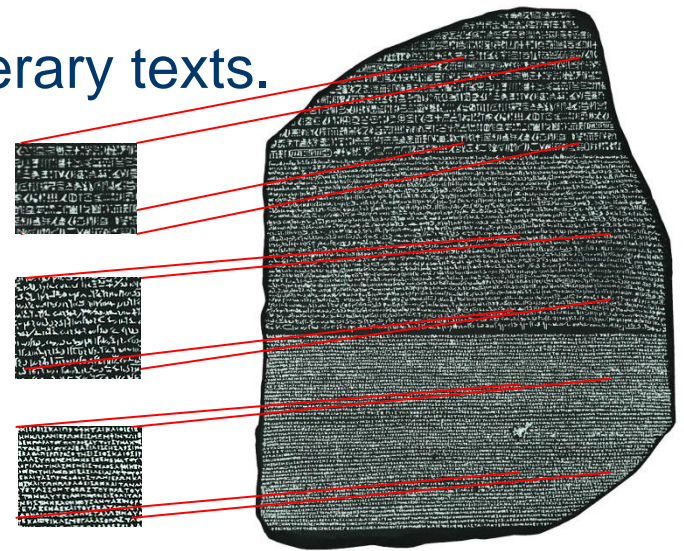
- Several search heuristics are possible
 - ◆ Beam search: stack decoding
 - ★ Maintain priority queue for each prefix substring of source sentence.
 - ★ Discard unlikely search states
 - ◆ A* search
 - ★ Visit hypotheses in order of $\text{score}(\text{state}) = \text{state}.P + h(\text{state})$, where $h(\text{state})$ is an estimate of future log-probabilities.
 - ★ Estimate future log-probabilities by using best phrase translations for remaining words and discarding reordering costs.

Learning Translation Models

- Fine, now we have seen very simple translation models (such as IBM1) as well as more useful ones (phrase-based models)
- We have also seen how we can find a good decoding of such models, that is, translations.
- However, so far we assumed that model parameters, such as phrase translation probabilities $P(\bar{s}|\bar{t})$, are given, e.g. as phrase tables.
- Now we will focus on more interesting things: Learning.

Learning of Phrase-Based Models

- Translation models are learned from *parallel corpora*, e.g.
 - ◆ Europarl: official translations by the European parliament. Translations for EU member languages.
 - ◆ Hansards: Canadian parliament. French/English
 - ◆ Hongkong Hansards: Chinese/English
 - ◆ News articles
 - ◆ Less popular: translation of literary texts.



Training Data for Learning

- As a first step, parallel corpora have to be sentence aligned.
- Many publicly available corpora are already sentence aligned (e.g. europarl)
- Otherwise, a set of heuristics are available.
 - ◆ Based on length of sentences
 - ◆ Based on letter N-grams
 - ◆ lexically

Word Alignment

- Phrase-based approaches have to estimate probabilities $P(\bar{s}|\bar{t})$ for translating phrase \bar{t} to \bar{s} .
- In order to do so, phrases need to be aligned in a parallel corpus.
- Phrase alignments can be generated from word alignments.
- IBM Models are not well suited for translation tasks but are used a lot for word alignments.

IBM Model 1

- Alignment decisions are independent
- $P(t|s)$ is categorical (multinomial)
- It follows ($\text{length}(a)$ is omitted because length of sentence S is fixed)

$$\begin{aligned} P(T, a|S) &= P(a|S) P(T|a, S) \\ &= \frac{1}{k+1} \prod_k P(t_k|s_{a_k}) \end{aligned}$$

and $P(T|S) \sim \sum_a \prod_k P(t_k|s_{a_k})$

IBM Model 1 — Decoding

- Find best word alignment

$$\begin{aligned} a^* &= \operatorname{argmax}_a P(T, a|S) \\ &= \operatorname{argmax}_a \prod_k P(t_k | s_{a_k}) \end{aligned}$$

- Independence of alignments:

$$a_k^* = \operatorname{argmax}_{a_k} P(t_k | s_{a_k})$$

IBM Model 1 — Learning

- How do we learn the lexical translation probabilities?
- If alignments were known, estimation would only consist of counting:

$$P(\text{ist}|\text{is}) = \frac{\text{count}(\text{ist} \sim \text{is})}{\text{count}(\text{is})}$$

- Unfortunately, alignments are not known.
- Instead, alignments and probabilities both have to be estimated using the EM algorithm.

IBM Model 1 — Learning

EM Algorithm

- Initialize random translation probabilities $P(t|s)$
- Iterate:
 - ◆ Estimate alignments based on current estimates of $P(t|s)$.

$$P(a|S, T) = \frac{P(T, a|S)}{P(T|S)} = \frac{P(T, a|S)}{\sum_{a'} P(T, a'|S)}$$

- ◆ Estimate ML probabilities by computing the expected counts

$$E[\text{count}(s, t)] = \sum_{\langle S, T \rangle} \sum_a P(a|S, T) \sum_{i=1}^{|a|} \delta(s_{a_i} = t_i)$$

and compute $P(t|s) = \frac{E[\text{count}(s, t)]}{E[\text{count}(s)]}$

IBM Models 2,3

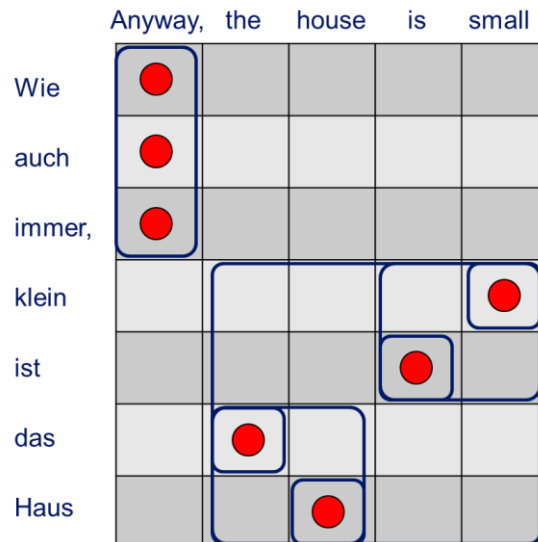
- IBM Models 2 to 5 introduce non-uniform distributions for $P(a|T)$.
- Model 2 introduces a position dependent *reordering model*.
- Model 3 introduces a *fertility model* that predicts the number of words a word t is aligned to in S .
- Models 4 and 5 introduce more complex reordering models.

IBM Models

- IBM models implemented in GIZA++, a toolbox which is widely used for word alignments.
- Practical issue: IBM models can only model many-to-one relations.
 - ◆ In practice alignments in both directions are computed and then the union and intersection between both alignments are computed.
- Other alignment models also possible.
 - ◆ E.g. HMM model

Phrase-Based Models

- For phrase based models, we need phrase pairs.
- Based on word alignments, phrase pairs can be easily computed for training data.
- Phrases are rectangles, word alignments are points










Phrase-Based Models


- Construction of phrases from word alignment.
- Given a pair of aligned sentences S and T with alignment \sim , such that $s_{k_1} \sim t_{k_2}$ iff k_1 -th word in source sentence is aligned with k_2 -th word in target sentence.
- Then $\langle s_{i_1} \dots s_{j_1}, t_{i_2} \dots t_{j_2} \rangle$ is a phrase pair if
 - ◆ $s_{k_1} \sim t_{k_2}$ for at least one $k_1 \in [i_1, j_1]$ and $k_2 \in [i_2, j_2]$
 - ◆ $s_{k_1} \not\sim t_{k_2}$ for all $k_1 \notin [i_1, j_1]$ and $k_2 \in [i_2, j_2]$
 - ◆ $s_{k_1} \not\sim t_{k_2}$ for all $k_1 \in [i_1, j_1]$ and $k_2 \notin [i_2, j_2]$

Phrase Generation

Anyway, the house is small

Wie					
auch					
immer,					
klein					
ist					
das					
Haus					









Phrase Generation

	Anyway,	the	house	is	small
Wie					
auch					
immer,					
klein					
ist					
das					
Haus					

Phrase Generation

	Anyway,	the	house	is	small
Wie					
auch					
immer,					
klein					
ist					
das					
Haus					

Phrase Generation

	Anyway,	the	house	is	small
Wie					
auch					
immer,					
klein					
ist					
das					
Haus					

Phrase Generation

	Anyway,	the	house	is	small
Wie	<input checked="" type="checkbox"/>				
auch	<input checked="" type="checkbox"/>				
immer,	<input checked="" type="checkbox"/>				
klein					<input checked="" type="checkbox"/>
ist				<input checked="" type="checkbox"/>	
das		<input checked="" type="checkbox"/>			
Haus			<input checked="" type="checkbox"/>		

Phrase-Based Model — Learning

- We can now compute phrase alignments for parallel corpora.
- We still have to learn phrase translation probabilities.
- Luckily, this is very simple if we have phrase alignments:

- ◆ Just count!
$$P(\bar{s}|\bar{t}) = \frac{\text{count}(\bar{s},\bar{t})}{\text{count}(\bar{t})} = \frac{\text{count}(\bar{s},\bar{t})}{\sum_{\bar{s}} \text{count}(\bar{s},\bar{t})}$$

Tree (Syntax)-Based Models

- So far, we did neither use nor model any grammatical / structural / syntactical information.
- But could certainly be helpful:
 - ◆ E.g. in German: hat ... gekauft
 - ◆ In English has bought ...
 - ◆ Phrase-based model: potentially hundreds of different phrases

hat Schuhe gekauft	hat Hosen gekauft	hat Brötchen gekauft	...
has bought shoes	has bought pants	has bought rolls

Syntax-based Models

- Better:

- ◆ Rule like

$X \rightarrow \text{hat } Y \text{ gekauft}$

should be translated to

$X \rightarrow \text{has bought } Y$

- Possible realization:

- ◆ Parse source sentence.

- ◆ Convert source parse tree to parse tree for target language according to grammatical rules.

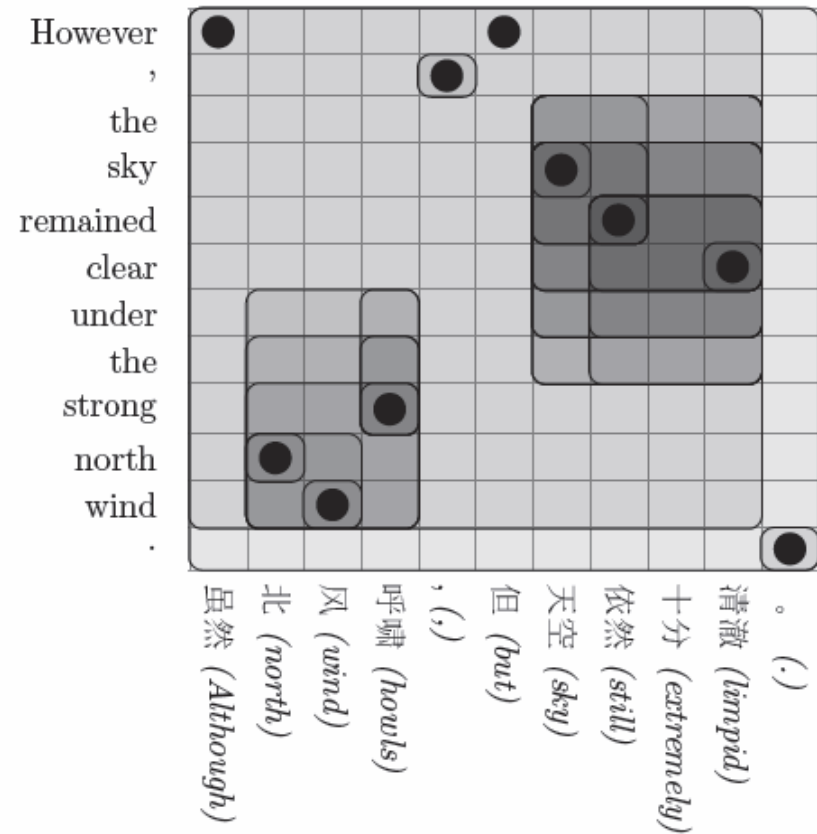
- ◆ Generate target sentence.

Synchronous PCFGs

- Simultaneous generation of 2 sentences from 2 context free languages
- Definition:
 - ◆ Finite set of non-terminals $\{N_i\}$, start symbol N_1
 - ◆ Terminals in source language $\{s_k\}$ and in target language $\{t_k\}$
 - ◆ Rules $\{N_i \rightarrow \langle \alpha, \beta, \sim \rangle\}$, where α sequence of source terminals and non-terminals, β sequence of target terminals and non-terminals, \sim alignment between non-terminals in α and β .
 - ◆ Probabilities for rules $P(N_i \rightarrow \langle \alpha, \beta, \sim \rangle)$

Synchronous PCFGs

- Example sentence pair (English / Mandarin) with phrase pairs



Synchronous PCFGs

- Example (Alignment denoted by indices)

$NP \rightarrow DT_{\boxed{1}}NPB_{\boxed{2}} / DT_{\boxed{1}}NPB_{\boxed{2}}$

$NPB \rightarrow JJ_{\boxed{1}}NN_{\boxed{2}} / JJ_{\boxed{1}}NN_{\boxed{2}}$

$NPB \rightarrow NPB_{\boxed{1}}JJ_{\boxed{2}} / JJ_{\boxed{2}}NPB_{\boxed{1}}$

$DT \rightarrow \text{the} / \varepsilon$

$JJ \rightarrow \text{strong} / \text{呼啸}$

$JJ \rightarrow \text{north} / \text{北}$

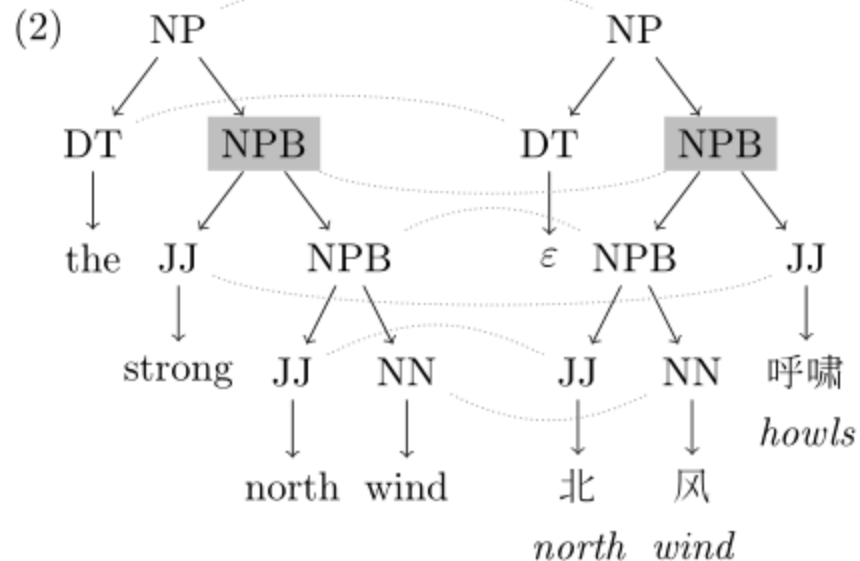
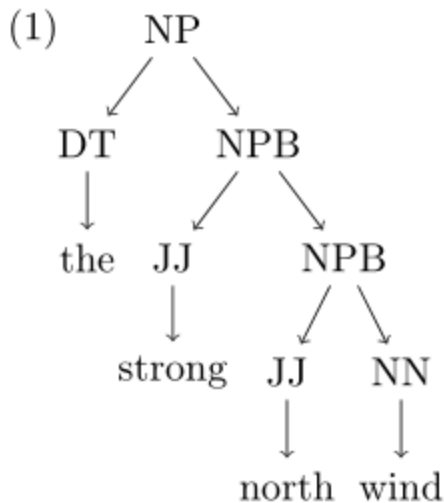
$NN \rightarrow \text{wind} / \text{风}$

Synchronous PCFGs: Parameter Estimation

- Rules have to be predefined
- Word alignment has to be known
- Learn with EM-Algorithm: Iterate
 - ◆ For each rule: compute expected counts in training data with current parameter estimates. (with inside-outside algorithm (Forward-Backward for PCFGs))
 - ◆ Estimate new parameters using counts

Synchronous PCFGs: Decoding

- For each sentence in source language:
 - ◆ Find most likely parse tree using only the source language part of the rules.
 - ◆ Infer target language part with help of alignment



Synchronous PCFGs

- Advantage: Elegant handling of reordering of subphrases
- Disadvantage: Rules have to be known in advance.
But:
 - ◆ Definition of synchronous rules are hard
 - ◆ Especially for languages with very different grammars
- Solution: Synchronous PCFGs based on phrases with automatically generated rules!

Hierarchical Phrase Translations

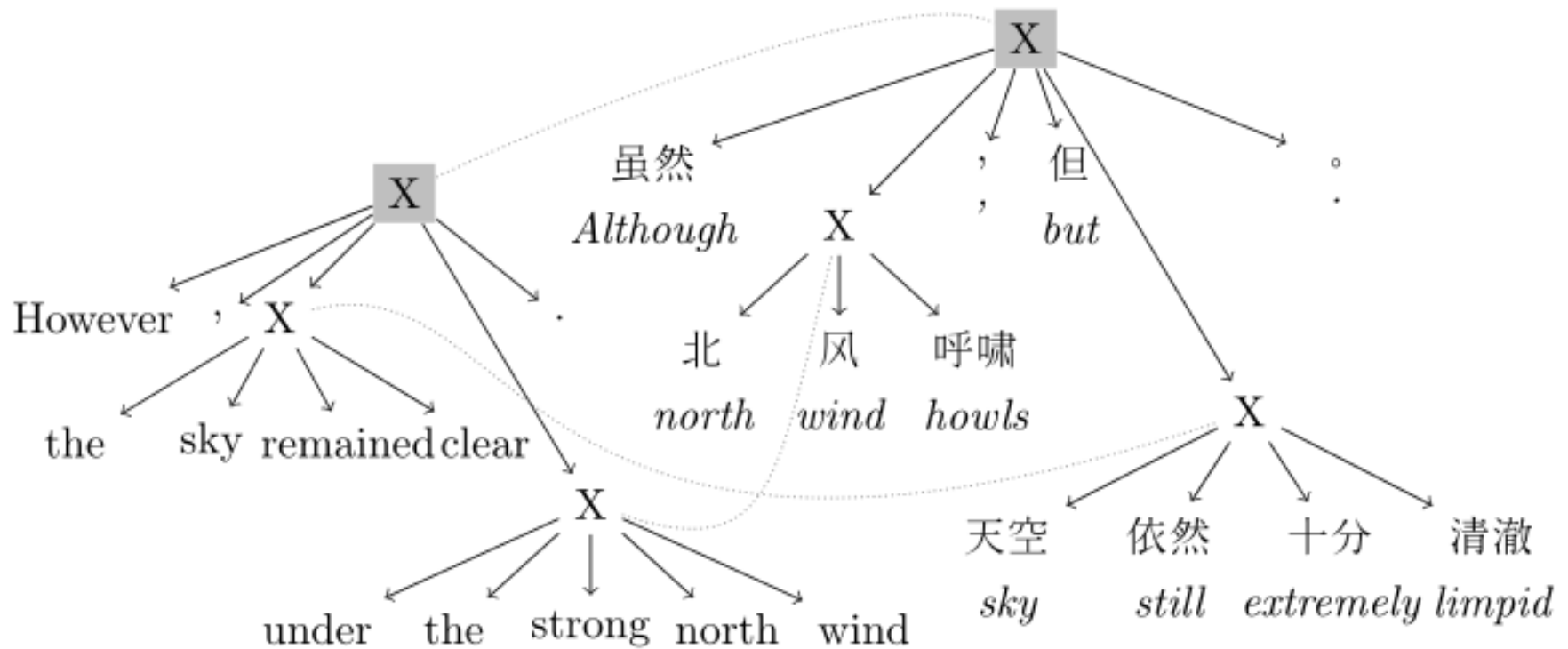
- Rules are automatically extracted from phrase pairs.
- Example:

$X \longrightarrow$ However , $X_{[1]}X_{[2]}$. / 虽然 $X_{[2]}$, 但 $X_{[1]}$ 。

$X \longrightarrow$ under the strong north wind / 北 风 呼 啸

$X \longrightarrow$ the sky remained clear / 天 空 依 然 十 分 清 澈

Hierarchical Phrase Translations



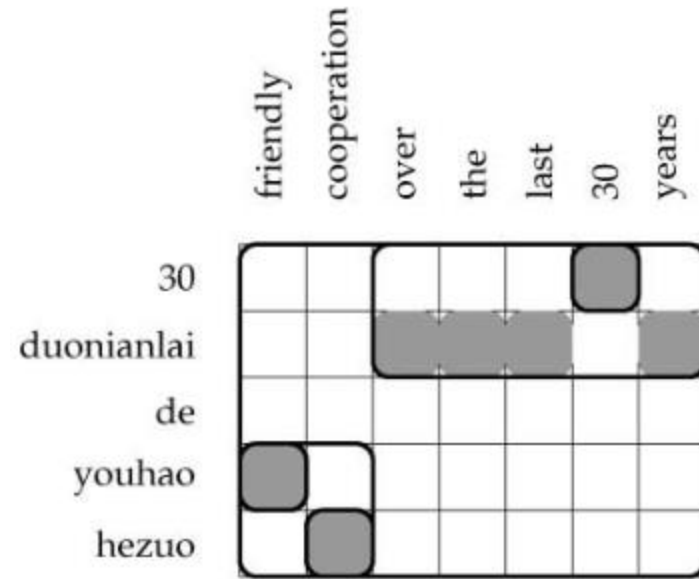
Hierarchical Phrase Translations

- Algorithm for generating rules:
 1. For all phrase pairs $\langle s_{i_1} \dots s_{j_1}, t_{i_2} \dots t_{j_2} \rangle$ add a rule
$$X \rightarrow \langle s_{i_1} \dots s_{j_1}, t_{i_2} \dots t_{j_2} \rangle$$
 2. For all rules $r = X \rightarrow \langle \alpha, \beta \rangle$:

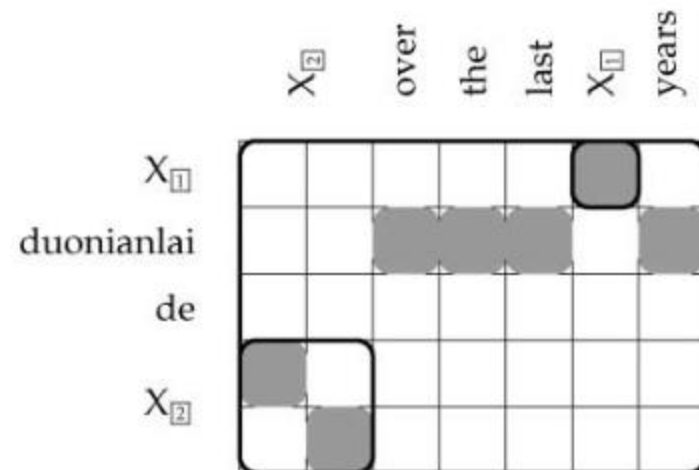
If $\langle s_{i_1} \dots s_{j_1}, t_{i_2} \dots t_{j_2} \rangle$ is a phrase pair, such that $\alpha = \alpha_1 s_{i_1} \dots s_{j_1} \alpha_2$ and $\beta = \beta_1 t_{i_2} \dots t_{j_2} \beta_2$, then
$$X \rightarrow \langle \alpha_1 X_k \alpha_2, \beta_1 X_k \beta_2 \rangle$$
is a rule and k is an index not used in r
 3. Repeat step 2 until no new rules can be added
- In practice some pruning mechanism has to be applied in order to reduce number of rules.

Rule Generation: Example

- Start:
Word and phrase alignments



- After 2 iterations:



String-to-Tree models

- An alternative approach is to use syntactic information only for one of the languages.
- In general, string-to-tree methods are models that use syntax information such as parse trees for the target language.
- With this approach we can use linguistic knowledge!

String-to-Tree model: GHKM

- One example is the GHKM formalism (also considered tree-to-string model)
 - ◆ Learn rules of the form

Syntax \rightarrow *String*

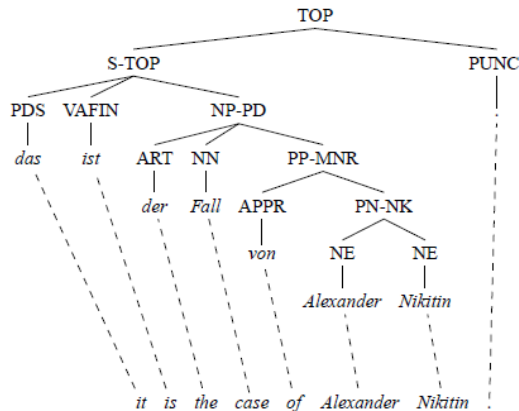
$VP(AUX(\textit{does}), RB(\textit{not}), x_1: VB) \rightarrow \textit{ne } x_1 \textit{ pas}$

Also expressible as:

$VP \rightarrow \textit{ne } x_1 \textit{ pas /does not } VB$

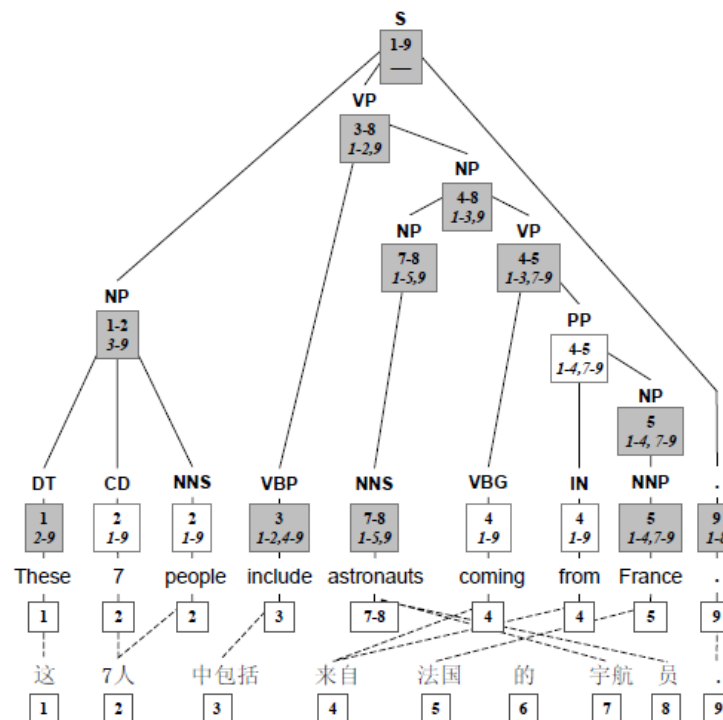
Training of String-to-Tree Model

- Assume that we have source sentence s , a parse tree for target sentence π which yields target sentence t and a word alignment a .
- Create rule set from this training example.



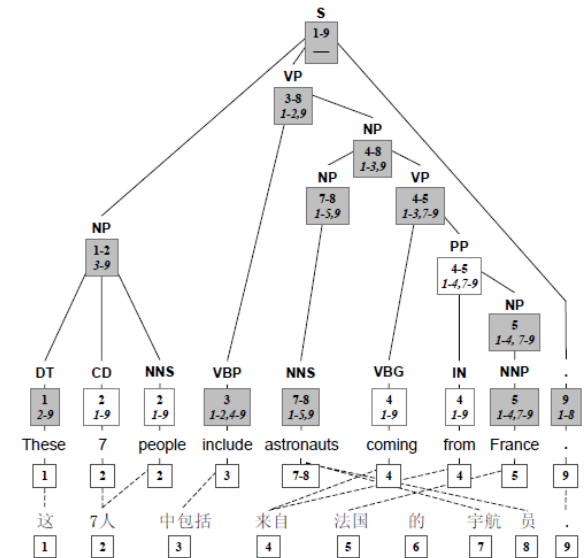
Rule Generation for GHKM

- Example triple (π, s, a) can be visualized as directed graph G consisting of nodes and edges of π and the alignment edges corresponding to a .



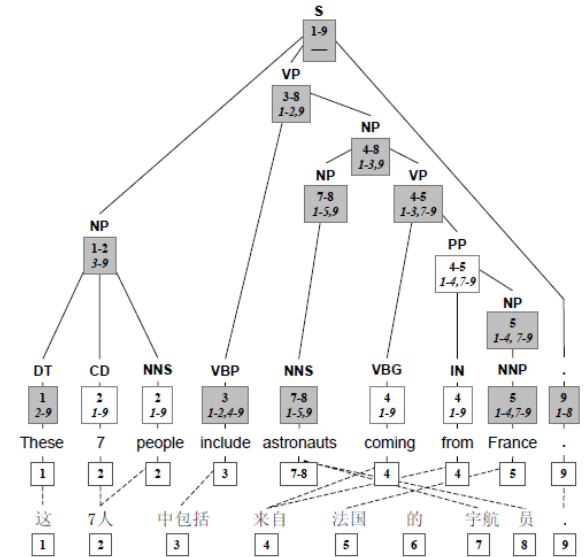
Rule Generation for GHKM

- Nodes are labeled with its *span* and *complement span*
 - ◆ The span of a node n is defined by the indices of the first and last word in s that are reachable from n .
 - ◆ The complement span of n is the union of the spans of all nodes n' in G that are neither descendants nor ancestors of n .
 - ◆ Nodes of G whose spans and complement spans are non-overlapping form the *frontier set* $F \subset G$



Rule Generation for GHKM

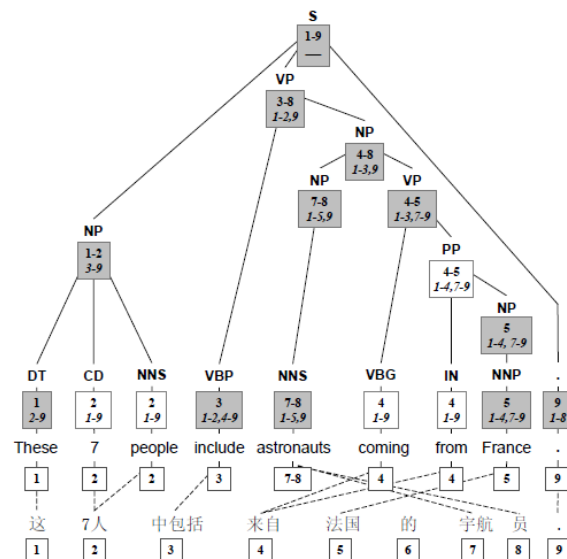
- Frontier nodes are depicted in gray in the example graph.
- Frontier nodes are important for choosing rules of the form
 - ◆ $S(x_0:NP, x_1:VP, x_2:..) \rightarrow x_0, x_1, x_2$
 - ◆ Only frontier nodes are allowed as variables in the left hand side of rules.
 - ◆ $NP(x_0:DT, x_1:CD, x_2:NNS) \rightarrow x_0, x_1, x_2$ is NOT allowed.
 - ◆ Instead, $NP(x_0:DT, CD(7), NNS(people)) \rightarrow x_0, 7人$



Rule Generation for GHKM

- Such rules are said to be induced by G .
- Minimal rules defined over G cannot be decomposed into simpler rules.
- Minimal rule set for example graph:

- $S(x_0:NP, x_1:VP, x_2:.) \rightarrow x_0, x_1, x_2$
- $NP(x_0:DT, CD(7), NNS(people)) \rightarrow x_0, 7人$
- $DT(these) \rightarrow 这$
- $VP(x_0:VBP, x_1:NP) \rightarrow x_0, x_1$
- $VBP(include) \rightarrow 中包括$
- $NP(x_0:NP, x_1:VP) \rightarrow x_1, 的, x_0$
- $NP(x_0:NNS) \rightarrow x_0$
- $NNS(astronauts) \rightarrow 宇航, 员$
- $VP(VBG(coming), PP(IN(from), x_0:NP)) \rightarrow 来自, x_0$
- $NP(x_0:NNP) \rightarrow x_0$
- $NNP(France) \rightarrow 法国$
- $.(.) \rightarrow .$



Rule Generation for GHKM

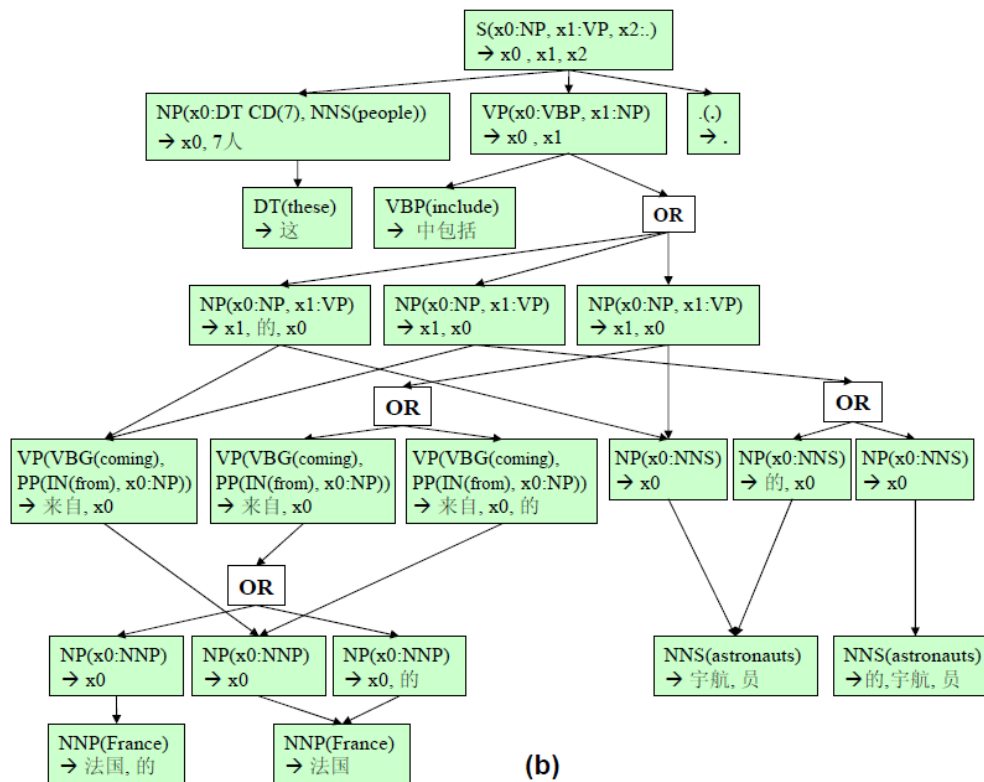
- Composed rules result from composition of two or more rules. E.g. rules (b) and (c) compose into:

$NP(DT(\textit{these}), CD(7), NNS(\textit{people})) \rightarrow \text{这}, 7\text{人}$

- Minimal rules for a graph G are unique iff there are no unaligned words in s .
- Otherwise, many minimal derivations of G are possible.

Rule Generation for GHKM

- Several derivations due to unaligned source word 的 : Derivation forest.



Rule Generation for GHKM

- Algorithm:
- Maintain a table to store OR-nodes which can uniquely be defined with its span $l - u$ and its syntactic category c (e.g. NP).
 1. Assign spans und complement spans to each node in the graph, determine frontier set F .
 2. Extract minimal rule for each $n \in F$.

Rule Generation for GHKM

- Algorithm:
 3. For each node n in the frontier set (top-down traversal):
 - ◆ Explore all tree fragments rooted at n by maintaining open and closed queues q_o of rules.
 - ◆ At each step, take smallest rule from q_o and try for each of its variables to discover new rules by means of composition until threshold on rule size is reached.
 - ◆ Add new rules to OR table.

Probability Estimation for GHKM

- The OR-table contains a representation that encodes valid derivations.
- Next, we have to estimate probabilities $P(rhs(r)|lhs(r))$ for rules r of the form $lhs(r) \rightarrow rhs(r)$.
- Note that, according to Bayes' rule,
$$\operatorname{argmax}_t P(t|s) = \operatorname{argmax}_t P(t)P(s|t) = \operatorname{argmax}_t P(t) \sum_{\pi} P(s|\pi)P(\pi|t)$$
- $P(\pi|t)$ is probability for certain parse tree. Not covered here (Will be covered later)

Probability Estimation for GHKM

- We want to estimate $P(s|\pi)$ by estimating rule probabilities. $P(s|\pi) = \sum_{\theta} \prod_{r \in \theta} P(rhs(r)|lhs(r))$
 - ◆ θ are derivations that are constructible from G .
 - ◆ $\theta = r_1 \circ \dots \circ r_k$ where r_i are the rules that constitute the derivation.

- If we assume that all derivations are equally likely, estimation is done by counting

$$P(rhs(r)|lhs(r)) = \frac{count(r)}{\sum_{r': lhs(r')=lhs(r)} count(r')}$$

- Otherwise, we have to estimate derivation probabilities $P(\theta|G)$.

Probability Estimation for GHKM

- If we would know $P(\theta|G)$, we could weight the counts $P(rhs(r)|lhs(r)) = \frac{\sum_{\theta:r \in \theta} P(\theta|G)}{\sum_{r':lhs(r')=lhs(r)} \sum_{\theta:r' \in \theta} P(\theta|G)}$
- Use EM to estimate those the adjusted $P(rhs(r)|lhs(r))$.
- Iterate:
 - ◆ $P(\theta|G) = \frac{\prod_{r \in \theta} P(rhs(r)|lhs(r))}{\sum_{\theta'} \prod_{r \in \theta'} P(rhs(r)|lhs(r))}$
 - ◆ $P(rhs(r)|lhs(r)) = \frac{\sum_{\theta:r \in \theta} P(\theta|G)}{\sum_{r':lhs(r')=lhs(r)} \sum_{\theta:r' \in \theta} P(\theta|G)}$

General Decoding for Translation Models

- We have seen several machine translation models that each solve the same problem.

- Find the best target sentence T .

- Using Bayes' rule, $P(T|S) = \frac{P(S|T)P(T)}{P(S)}$

- Find most likely sentence T given S

$$\operatorname{argmax}_T P(T|S) = \operatorname{argmax}_T P(S|T)P(T)$$

$$= \operatorname{argmax}_T \sum_A P(S, A|T) P(T)$$

where A denotes some structure, such as alignment, phrase-based derivation, or PCFG derivation.

General Decoding

- Often, instead of finding the best target sentence, one resorts to finding only the best derivation

$$\operatorname{argmax}_{T,A} P(S, A|T)P(T)$$

- For convenience, we omit derivations from the notation in the next slides.
- In general, scoring of derivations and decoding can be realized using log-linear models.
 - ◆ E.g. we could generalize the phrase-based scoring function.
 - ◆ $\operatorname{argmax}_{T,A} P_r(S|T)^{\lambda_r} P_{lm}(T)^{\lambda_{lm}} W(\operatorname{len}(T))^{\lambda_W} d(A)^{\lambda_A}$

Decoding—Linear Models

- $\lambda_r, \lambda_{lm}, \lambda_W, \lambda_d$ are weights for each individual term of the scoring function.
- Taking the logarithm yields
 - ◆ $\operatorname{argmax}_{T,A} \lambda_r \log P_r(S|T) + \lambda_{lm} \log P_{lm}(T) + \lambda_W \log W(\operatorname{len}(T)) + \lambda_A \log d(A)$

$$\text{◆ } \operatorname{score}(S, T, A) = \begin{pmatrix} \lambda_r \\ \lambda_{lm} \\ \lambda_W \\ \lambda_A \\ \dots \end{pmatrix}^T \begin{pmatrix} \log P_r(S|T) \\ \log P_{lm}(T) \\ \log W(\operatorname{len}(T)) \\ \log d(A) \\ \dots \end{pmatrix}$$

Decoding—Linear Models

- The weights λ can be learned by supervised learning methods such as CRFs (cf. lecture on Basic Models) that maximize the score for the best translation.
 - ◆ More specialized models were also proposed e.g. [Och,2003]

Features for Linear Models

- In modern translation systems, a large variety of features is generated and used in such linear models, e.g.
 - ◆ the direct translation probability $\log P(T|S)$ or $\log P(T, A|S)$ or either without the log,
 - ◆ scores $P(S|T)$ computed by several different translation models such as phrase-based and syntax-based,
 - ◆ scores based on additional lexical likelihoods, e.g. by using 'real' lexicons,
 - ◆
- We will learn about some additional features today!

Evaluation

- How can we evaluate the performance of one translation model?
- How can we compare different translation models?
- Good evaluation metrics of translation models not obvious
 - ◆ (Expert) human translations serve as ground truth / reference translations.
 - ◆ There will generally be several correct (human) translations of the same text.
 - ◆ Some reference translations might not even be agreed upon by all experts.

Evaluation

- Evaluation of translation can be separated into two parts. Humans can give their verdict on each of those.
 - ◆ Fluency
 - ★ Evaluates the fluency, naturalness, or style of the translated target text.
 - ◆ Adequacy
 - ★ Evaluates the informativeness of the translated text. That is, how much of the information of the source sentence is transported to the translation

Evaluation

- This scenario demands humans to evaluate each translated sentence.
 - ◆ Very time consuming!
 - ◆ Evaluate feedback for new methods or parameter settings becomes bottleneck for development of machine translation system.
- Instead, one should resort to automatically evaluable performance metrics.

Evaluation — BLEU

- Several such performance metrics were proposed, e.g. BLEU, NIST, TER, ROUGE, ...
- The most commonly used performance metric is the BLEU score (Bilingual Evaluation Understudy).
 - ◆ It allows to compare to several reference translations at once.
 - ◆ It compares n-grams of reference translations and candidate translations (machine translation).

BLEU — Example

- Score and compare two candidate translations

Cand 1: It is a guide to action which ensures that the military always obeys the commands of the party

Cand 2: It is to insure the troops forever hearing the activity guidebook that party direct

Ref 1: It is a guide to action that ensures that the military will forever heed Party command

Ref 2: It is the guiding principle which guarantees the military forces always being under the command of the Party

Ref 3: It is the practical guide for the army always to heed the directions of the party

Example taken from: Daniel Jurafsky & James H. Martin.

Speech and Language Processing: An introduction to natural language processing, computational linguistics, and speech recognition.

BLEU

- BLEU uses a modified n-gram precision metric.
- A candidate sentence is evaluated by a weighted average of the number of n-gram overlaps.
- Let $cand$ denote a candidate translation and $cand(s)$ the translation of sentence s . Let ref be a reference translation and $ref(s)$ its translation of s .

BLEU

- We define the clipped counts for n -grams as:

For some $n \in \{1, \dots, N(4)\}$:

For all distinct n -grams α_n in candidate sentence $\text{cand}(s)$ let:

$$c(\alpha_n, \text{cand}(s)) = \min\{\text{count}(\alpha_n, \text{cand}(s)), \max_{\text{ref}}\{\text{count}(\alpha_n, \text{ref}(s))\}\}$$

- $\text{score}(n, \text{cand}) = \frac{\sum_s \sum_{\alpha_n} c(\alpha_n, \text{cand}(s))}{\sum_s \sum_{\alpha_n} \text{count}(\alpha_n, \text{cand}(s))}$

BLEU

- With normal counts, the following candidate translation would have a score of

$$\text{score}(1, \text{cand}) = \frac{7}{7}$$

Candidate: the the the the the the

Reference 1: the cat is on the mat

Reference 1: there is a cat on the mat

BLEU

- $BLEU(cand) = BP \exp\left(\frac{1}{N} \sum_{n=1}^N w_n \log(score(n, cand))\right)$
 - ◆ BP penalizes translations that are too short.
 - ◆ Let $c = \text{len}(\text{candidate corpus})$.
 - ◆ and $r = \text{effective length of reference corpus}$
 - ★ Effective length: for each candidate sentence, find reference sentence that is closest in length. Sum length of all those reference sentences.
 - ◆ $BP = \begin{cases} 1 & \text{if } c > r \\ e^{(1-r/c)} & \text{otherwise} \end{cases}$
 - ◆ w_n can be used to give more weight to certain n .

BLEU — Example

Cand 1: It is a guide to action which ensures that the military always obeys the commands of the party

Cand 2: It is to insure the troops forever hearing the activity guidebook that party direct

Ref 1: It is a guide to action that ensures that the military will forever heed Party command

Ref 2: It is the guiding principle which guarantees the military forces always being under the command of the Party

Ref 3: It is the practical guide for the army always to heed the directions of the party

- $c('the', cand1(s)) = \min\{3, \max\{1, 4, 4\}\} = 3$

- $score(1, cand1) = \frac{\sum_s \sum_{\alpha_n} c(\alpha_n, cand(s))}{\sum_s \sum_{\alpha_n} count(\alpha_n, cand(s))} = \frac{17}{18}$

- $score(2, cand1) = \frac{10}{17}$

- $score(3, cand1) = \frac{7}{16}, score(4, cand1) = \frac{4}{15}$

- $BP = 1, BLEU(cand1) =$

$$\exp\left(\frac{1}{4}(-0.024 - 0.230 - 0.359 - 0.574)\right) = 0.74$$

BLEU vs. Human Evaluation

- At WMT15 (EMNLP Workshop on Statistical Machine Translation 2015), human evaluation was compared to BLEU scores.
- 137 researchers contributed to the annotation campaign. They had to decide for binary rankings such as ,translation A > translation B‘.

Хотите светящегося в темноте мороженого?
Британский предприниматель создал первое в мире светящееся в темноте мороженое с помощью медузы.
— Source

Fancy a glow-in-the-dark ice cream? A British entrepreneur has created the world's first glow-in-the-dark ice cream - using jellyfish.
— Reference

Best ← Rank 1 ● Rank 2 ● Rank 3 ● Rank 4 ● Rank 5 ● → **Worst**
You do want ice cream luminous in the darkness?
— Translation 1

Best ← Rank 1 ● Rank 2 ● Rank 3 ● Rank 4 ● Rank 5 ● → **Worst**
You want to glowing in the dark ice cream?
— Translation 2

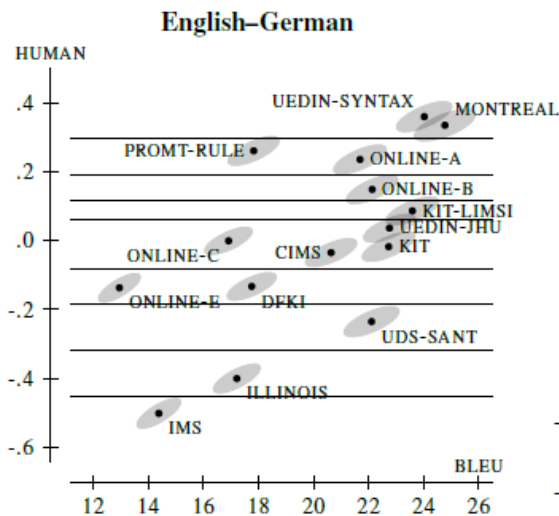
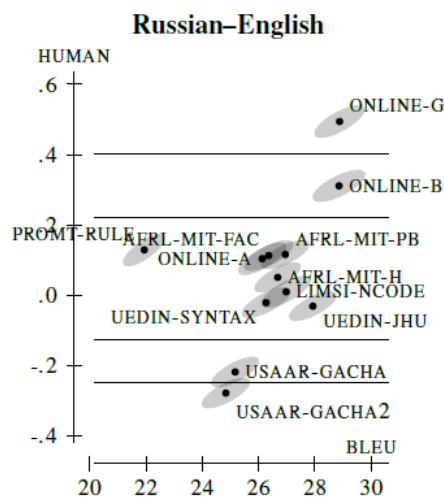
Best ← Rank 1 ● Rank 2 ● Rank 3 ● Rank 4 ● Rank 5 ● → **Worst**
You want the luminous in the dark ice cream?
— Translation 3

Best ← Rank 1 ● Rank 2 ● Rank 3 ● Rank 4 ● Rank 5 ● → **Worst**
Want luminous in the dark ice cream?
— Translation 4

Best ← Rank 1 ● Rank 2 ● Rank 3 ● Rank 4 ● Rank 5 ● → **Worst**
Want to illuminate the Dark with Ice Cream?
— Translation 5

BLEU vs. Human Evaluation

- They computed general scores based on those ranking decisions and compared them to BLEU scores.
- As a result, it has to be said the BLEU alone is not a very reliable evaluation metric for translation tasks.



Neural Network-Based Translation Models

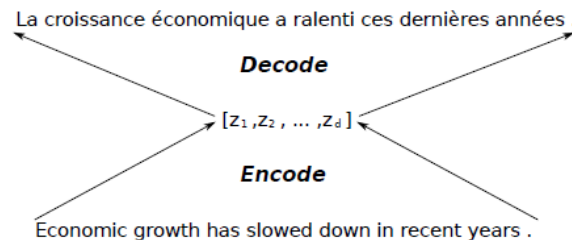
- Recently, neural network based translation models gained a lot of interest and showed very good empirical performance
- Many deep learning architectures are possible for tackling translation tasks.
- We focus here on Encoder-Decoder approaches that have shown promising results and have strong relations to NN-based continuous language models.

Encoder-Decoder Approaches

- Idea: Learn intermediate representation of sentences or phrases.
 - ◆ Encoder reads whole sentence/phrase and computes hidden representation .
 - ◆ Decoder uses the fixed-length representation to decode translated sentence.
 - ◆ Representation idea similar to (single) language models such as Word2Vec, i.e. word embedding.
 - ◆ Encoder and decoder generally Recurrent NNs with some kind of memory cells (e.g. LSTM).

Encoder-Decoder Approaches

- Several similar approaches use this idea.
 - ◆ Cho et al., 2014a: Learning Phrase Representations using RNN Encoder–Decoder for Statistical Machine Translation
 - ◆ Sutskever et al., 2014: Sequence to Sequence Learning with Neural Networks
 - ◆ Cho et al, 2014b: On the Properties of Neural Machine Translation: Encoder–Decoder Approaches

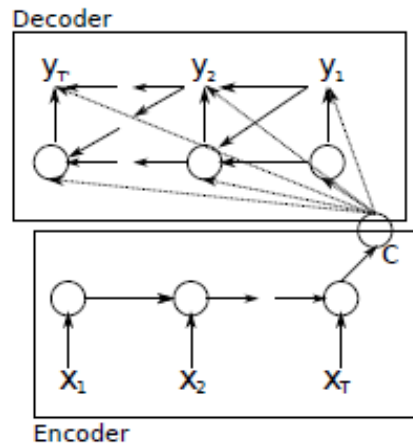


Encoder-Decoder Approaches

- Encoder-Decoder approaches can be used
 - ◆ to rescore n-best lists of standard phrase-based SMT models.
(I.e. after other model created a list of n best translations, ED-model computes likelihood of each of those translations and reranks hypotheses accordingly)
 - ★ Like using (log-)probabilities of translations / phrases as feature of linear model
 - ◆ to rescore of phrase pair scores (i.e. phrase translation probabilities).
 - ◆ as a standalone (direct) decoding mechanism.

Cho et al., 2014a

- RNN Encoder-Decoder.
 - ◆ Two RNNs, one for encoding, one for decoding.



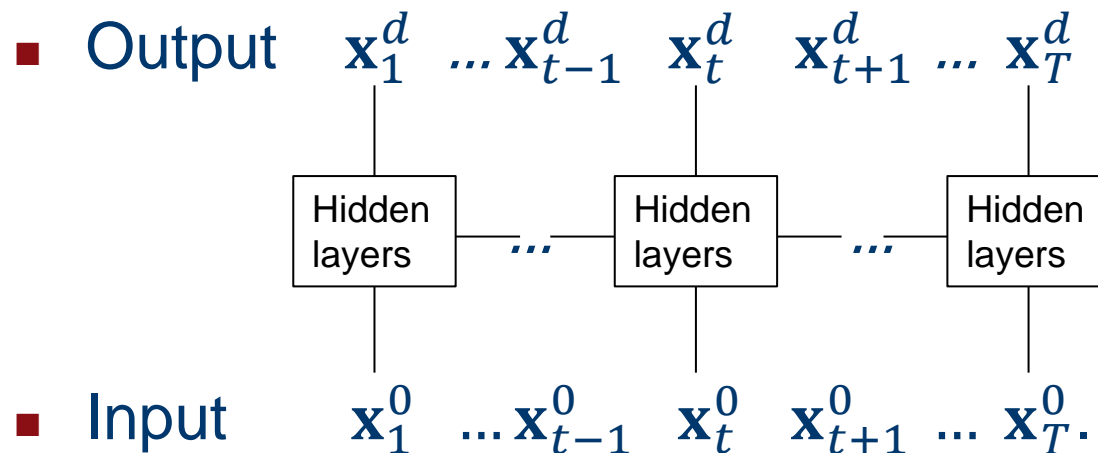
- Used as part of standard phrase-based SMT system
 - ◆ Used for (re-)scoring phrase pairs (i.e. $P(\bar{s}|\bar{t})$, $P(\bar{s}, \bar{t})$, $P(\bar{t}|\bar{s})$) from training set.
 - ◆ Learned on phrase pairs

RNNs and LSTM

- Input:
 - ◆ Sequence of word representations.
 - ★ One-hot encodings.
 - ★ Word embeddings from pre-trained language model.
- Output:
 - ◆ Softmax layer: word probabilities.

Recurrent Neural Networks

- Identical network “unfolded” in time.
- Units on hidden layer propagate to the right.
- Hidden layer activation stores context information.



LSTM

- Memory in RNNs
- Input gate scales input to memory.
- Forget gate scales old memory value.
- Output gate scales output.

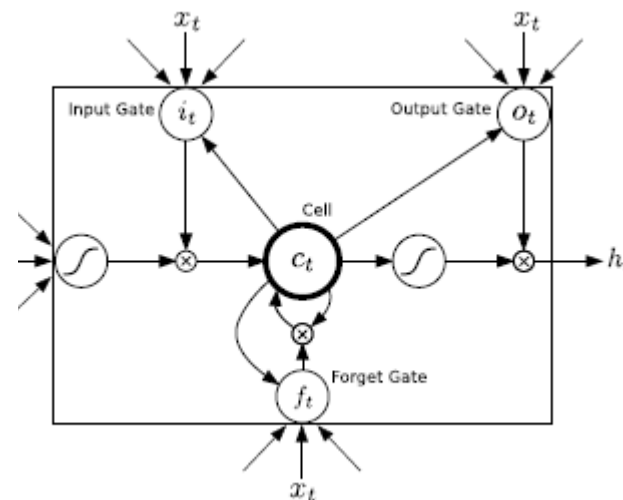
$$i_t = \sigma(W_{xi}x_t + W_{hi}h_{t-1} + W_{ci}c_{t-1} + b_i)$$

$$f_t = \sigma(W_{xf}x_t + W_{hf}h_{t-1} + W_{cf}c_{t-1} + b_f)$$

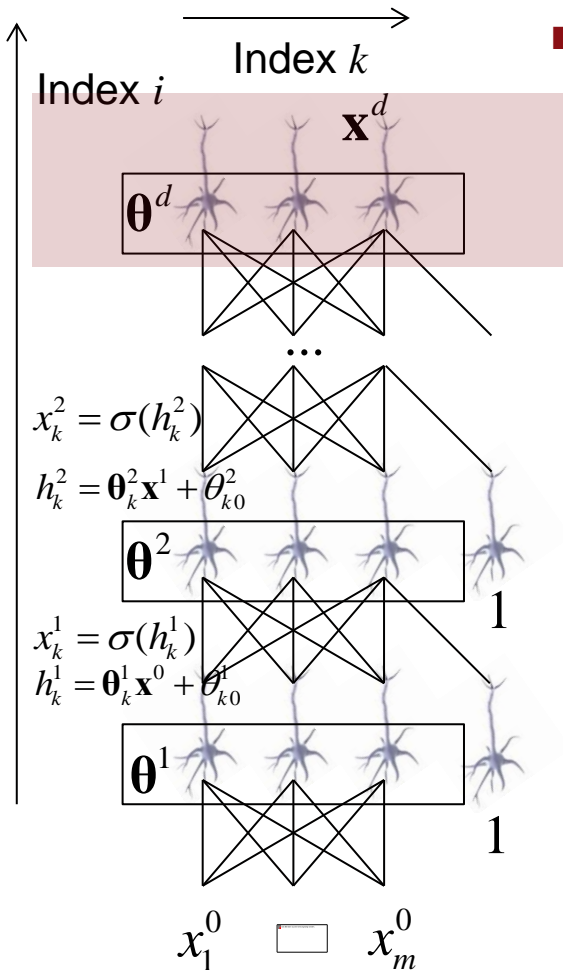
$$c_t = f_t c_{t-1} + i_t \tanh(W_{xc}x_t + W_{hc}h_{t-1} + b_c)$$

$$o_t = \sigma(W_{xo}x_t + W_{ho}h_{t-1} + W_{co}c_t + b_o)$$

$$h_t = o_t \tanh(c_t)$$



Classification: Softmax Layer

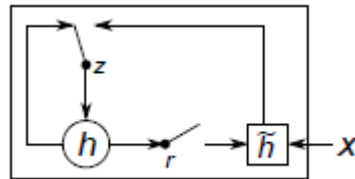


- One output unit per class:

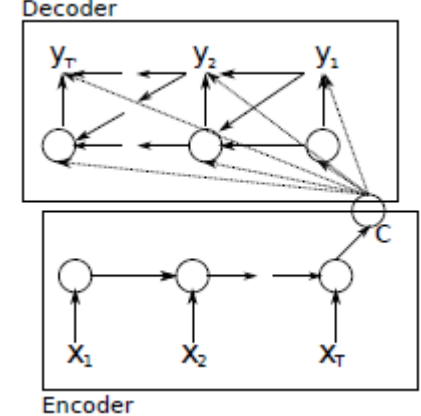
- ◆ $x_k^d = \sigma_{sm}(h_k^d) = \frac{e^{h_k^d}}{\sum_{k'} e^{h_{k'}^d}}$
- ◆ x_k^d : predicted probability for class k .

Cho et al., 2014a

- RNN Encoder-Decoder.
 - ◆ Two RNNs, one for encoding, one for decoding.
 - ◆ Uses special memory cells



- Used as part of standard phrase-based SMT system
 - ◆ Used for (re-)scoring phrase pairs (i.e. $P(\bar{s}|\bar{t})$, $P(\bar{s}, \bar{t})$, $P(\bar{t}|\bar{s})$) from training set.
 - ◆ Learned on phrase pairs.



Cho et al., 2014a

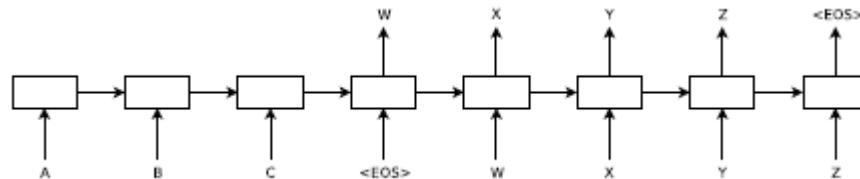
- Worked best in combination with NN language model: CSLM.
- CSLM used for scoring of partial translations during decoding of phrase-based system
 - ◆ Apparently works better than rescoring/reranking of n-best list. [Vaswani,2013]

Models	BLEU	
	dev	test
Baseline	30.64	33.30
RNN	31.20	33.87
CSLM + RNN	31.48	34.64
CSLM + RNN + WP	31.50	34.54

- Learns embedding of phrases. Visualization possible as in Word2Vec.

Sutskever et al., 2014

- 2 RNN with LSTM hidden layers.
- Input sentence is read in reverse order.
- After $\langle \text{EOS} \rangle$ tag is read, second RNN starts from last hidden state h_k .
- Models $P(t_1 \dots t_k | s_1 \dots s_l) = \prod_i P(t_i | h_k, t_{i-1}, \dots, t_1)$



- Decoding is done using beam search. Beam size of 2 already very good.
 - ◆ Log-probabilities are used as scores. Log-probs are normalized with length of decoding.

Sutskever et al., 2014

- Evaluated for both direct decoding as well as rescoring of 1000-best sentences
 - ◆ 1000-best lists of baseline phrase-based SMT model.

Method	test BLEU score (ntst14)
Bahdanau et al. [2]	28.45
Baseline System [29]	33.30
Single forward LSTM, beam size 12	26.17
Single reversed LSTM, beam size 12	30.59
Ensemble of 5 reversed LSTMs, beam size 1	33.00
Ensemble of 2 reversed LSTMs, beam size 12	33.27
Ensemble of 5 reversed LSTMs, beam size 2	34.50
Ensemble of 5 reversed LSTMs, beam size 12	34.81

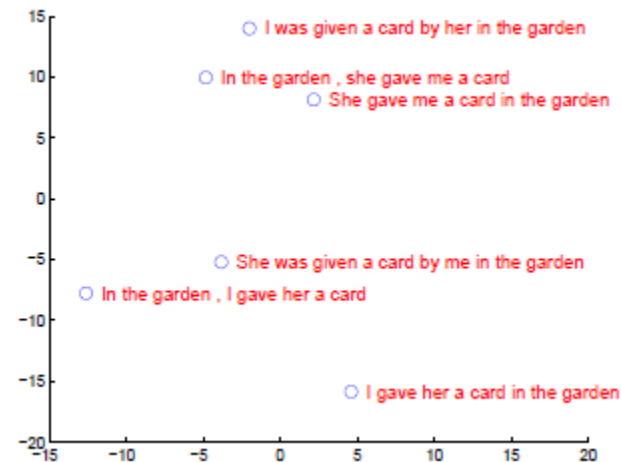
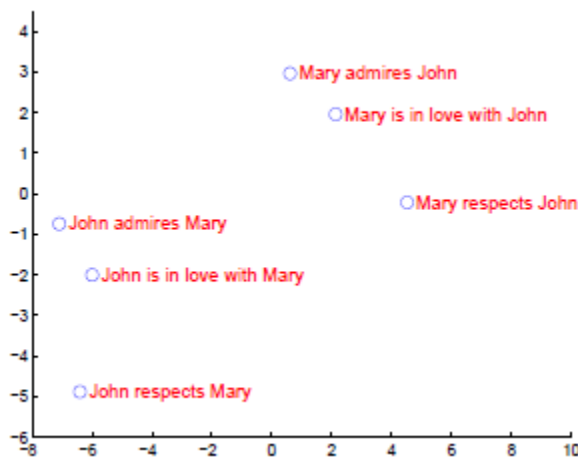
Table 1: The performance of the LSTM on WMT'14 English to French test set (ntst14). Note that an ensemble of 5 LSTMs with a beam of size 2 is cheaper than of a single LSTM with a beam of size 12.

Method	test BLEU score (ntst14)
Baseline System [29]	33.30
Cho et al. [5]	34.54
State of the art [9]	37.0
Rescoring the baseline 1000-best with a single forward LSTM	35.61
Rescoring the baseline 1000-best with a single reversed LSTM	35.85
Rescoring the baseline 1000-best with an ensemble of 5 reversed LSTMs	36.5
Oracle Rescoring of the Baseline 1000-best lists	~45

Table 2: Methods that use neural networks together with an SMT system on the WMT'14 English to French test set (ntst14).

Sutskever et al., 2014

- Visualization of phrase embedding.
 - ◆ 2-D PCA projection of LSTM hidden states.



Bahdanau et.al. 2015

- Neural Machine Translation by Jointly Learning to Align and Translate.
- Encoder realized as bidirectional RNN (No LSTM).
 - ◆ Bidirectional RNN: Two sets of hidden states, one for each direction.
- An alignment is learned simultaneously with encoder and decoder.
- Instead of only using the last state hidden state for conditioning the decoder on, all hidden states are used for decoding.
 - ◆ Alignment (soft-)assigns hidden states to outputs.

Encoder-Decoder Approaches

- Other Encoder-Decoder approaches use different encoder structure and/or other decoders, that is, other neural language (generation) models.
- Cho et al., 2014b showed that direct decoding for long sentences is problematic
 - ◆ However, empirical evaluation in Sutskever et al., 2014 suggests that they did not exhibit such a behavior.

Resources for Machine Translation

- Several resources for MT are available online.
 - ◆ <http://www.statmt.org/>
 - ◆ Moses. Translation tool. <http://www.statmt.org/moses>
 - ◆ Workshop/Conference on Statistical Machine Translation.
 - ★ E.g. <http://www.statmt.org/wmt16/>
 - ★ Several data sets, useful scripts, baseline models.
 - ★ Each year, a translation competition is held as part of the workshop.

- <http://matrix.statmt.org/> : lists results for several translation tasks.

newstest2016 using metric BLEU-cased

Account

		output language						
		Czech	German	English	Finnish	Romanian	Russian	Turkish
input language	Czech	31.4						
	German	38.6						
	English	25.8	34.2	17.4	28.9	26.0	9.8	
	Finnish		23.4					
	Romanian		35.2					
	Russian		29.1					
	Turkish		14.5					

Tricks

- Cho:
 - ◆ Do not use frequency of phrase pairs for learning NN models. (Use each phrase pair just once.). Probabilities of phrase tables are used in linear model, which use relative frequencies. NN score only additional model.

Summary

- Learn Machine Translation models from parallel corpora.
- For ,classical‘ statistical MT models, first step of learning generally involves computing word alignment.
- Phrase-based methods learn phrase-to-phrase translations.
 - ◆ For training, generate phrase pairs from word alignments.

Summary

- Syntax information can be incorporated into MT models.
 - ◆ Hierarchical phrase translation algorithms generate synchronous syntax rule without linguistic meaning.
 - ◆ String-to-tree (or similar) models use linguistic knowledge on target side.
- Evaluation of MT models problematic.
 - ◆ BLEU score widely used (de-facto standard)
 - ◆ But does not necessarily match human judgement.
- Neural translation models show very promising results.
 - ◆ Encoder-Decoder approaches.

Automatic Description Generation from Images

- Goal: Automatically generate descriptions for previously unknown images and videos.



there is a cat sitting on a shelf .



a plate with a fork and a piece of cake .



a black and white photo of a window .



a young boy standing on a parking lot next to cars .



a wooden table and chairs arranged in a room .



a kitchen with stainless steel appliances .



this is a herd of cattle out in the field .



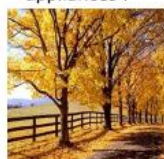
a car is parked in the middle of nowhere .



a ferry boat on a marina with a group of people .



a little boy with a bunch of friends on the street .



a giraffe is standing next to a fence in a field .
(hallucination)



the two birds are trying to be seen in the water .
(counting)



a parked car while driving down the road .
(contradiction)



the handlebars are trying to ride a bike rack .
(nonsensical)



a woman and a bottle of wine in a garden .
(gender)

Automatic Description Generation

- Learn model that learns to generate image descriptions from training data consisting of image-description pairs.
- Verbalize visual and conceptual information depicted in the image, i.e. descriptions that refer to the depicted entities, their attributes and relations, and the actions they are involved in.

Automatic Description Generation

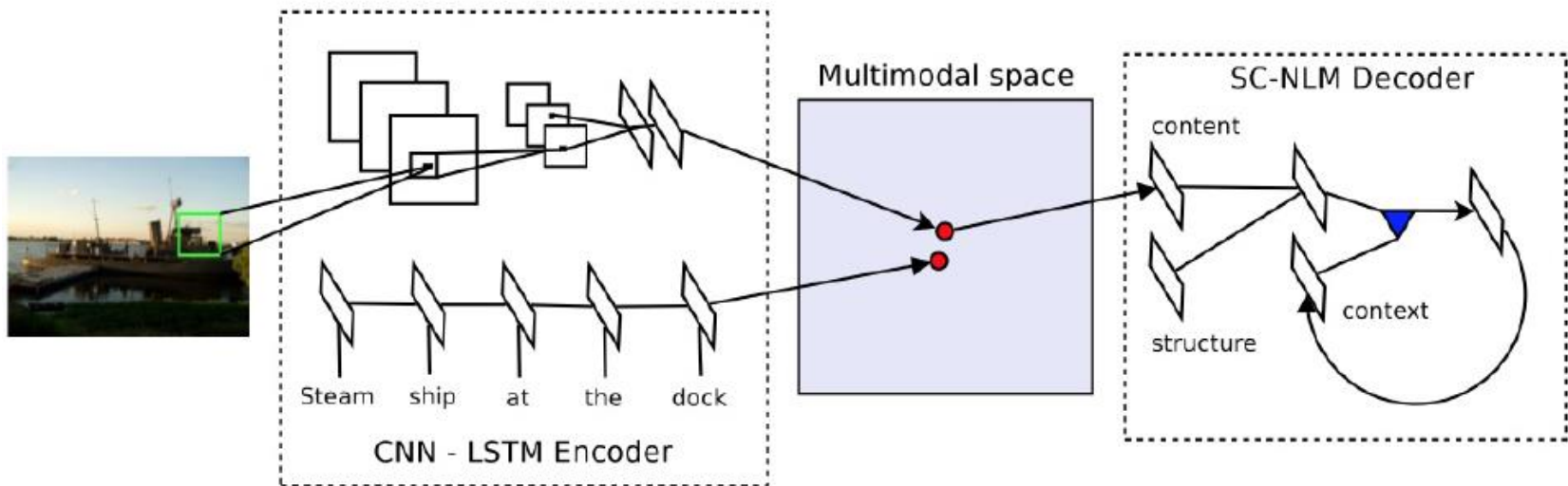
- Existing literature can be categorized into different modeling approaches.
 - ◆ Direct generation models: First detect image content and generate description based on image features
 - ◆ Retrieval based models: Image description generation as retrieval problem.
 - ★ Find most similar image in database, use same description.
 - ★ Synthesize new description out of descriptions of similar images.

Automatic Description Generation

- We only present neural network models because of the apparent similarity to machine translation tasks.
- To be more precise, we present encoder-decoder approaches for image description generation.
- Idea: Consider image as text in source language and translate it into text in description language.

Automatic Description Generation

- General approach:
 - ◆ Learn hidden representation of images (encoder) and, based on the representation, generate description using some neural language model conditioned on the representation (decoder).



Kiros et al, 2015

Automatic Description Generation

- Main principal difference is that the encoder is e.g. a convolutional network on image data.
- Output generation is then conditioned on the hidden representation of the image encoder.
- As for machine translation, several decoder architectures are possible.