Machine Translation

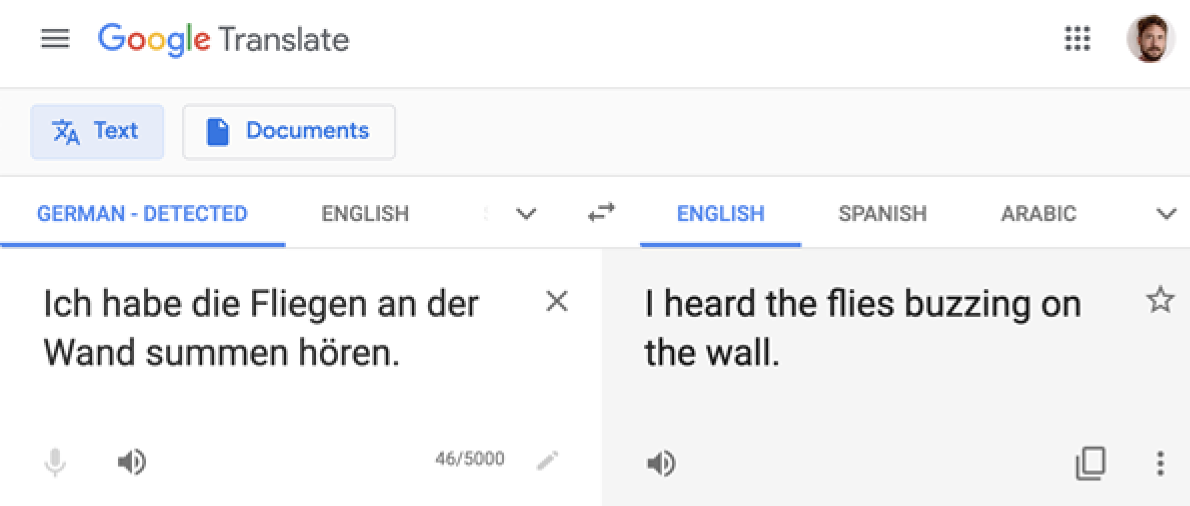
CS-585

Natural Language Processing

Sonjia Waxmonsky

Based on slides from Kai Shu and
Derrick Higgins

Machine Translation



Machine Translation

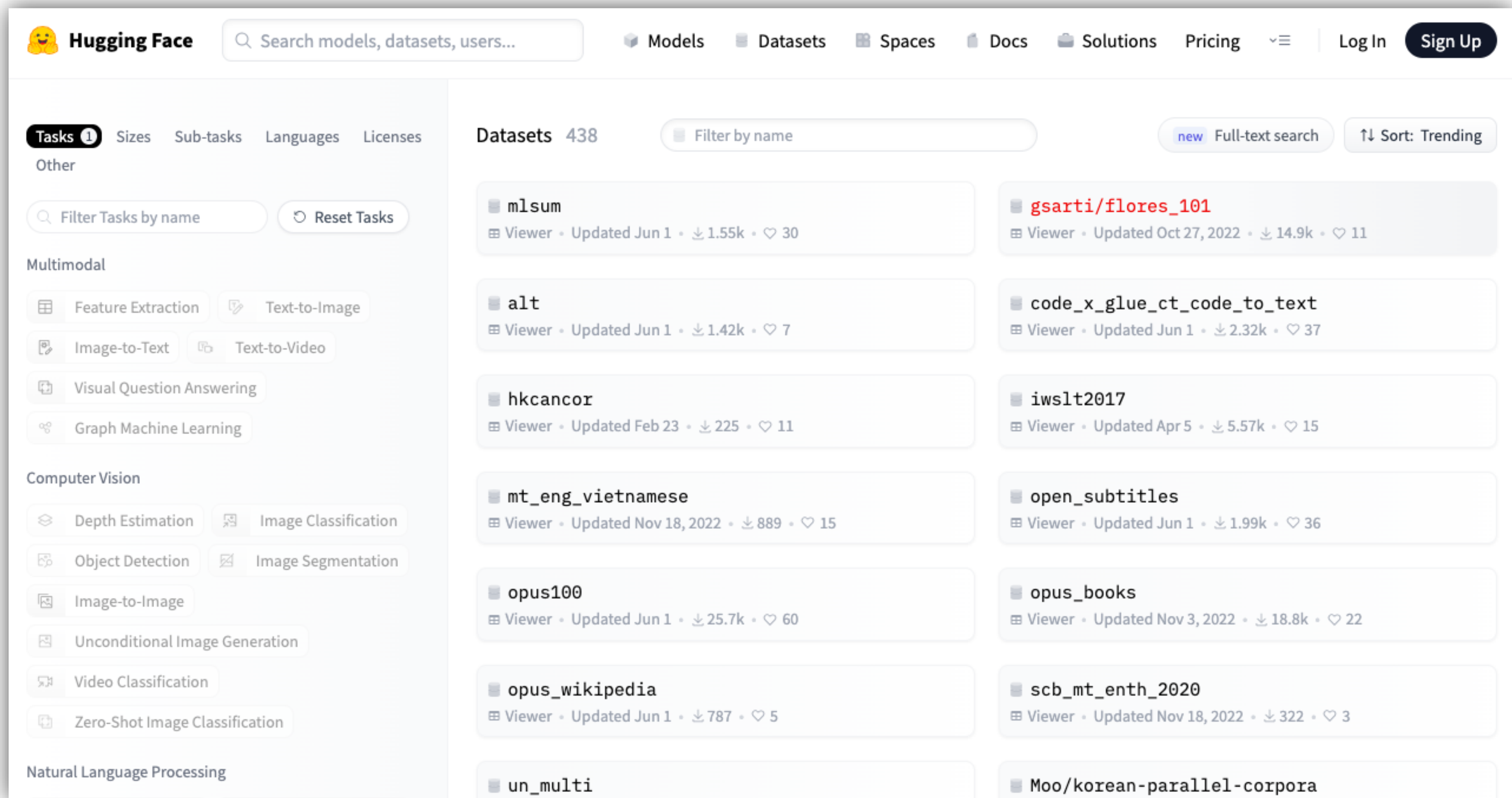
- For a given sentence in the **source language**, predict the most likely sentence in the **target language**

$$\widehat{W}_t = \underset{W_t}{\operatorname{argmax}} \boldsymbol{\Psi}(W_s, W_t)$$

- Large search space: all sequences of words in the target language
- Hard to leverage locality assumptions

Resources for Machine Translation

So much data is available!



https://huggingface.co/datasets?task_categories=task_categories:translation

Parallel Corpora

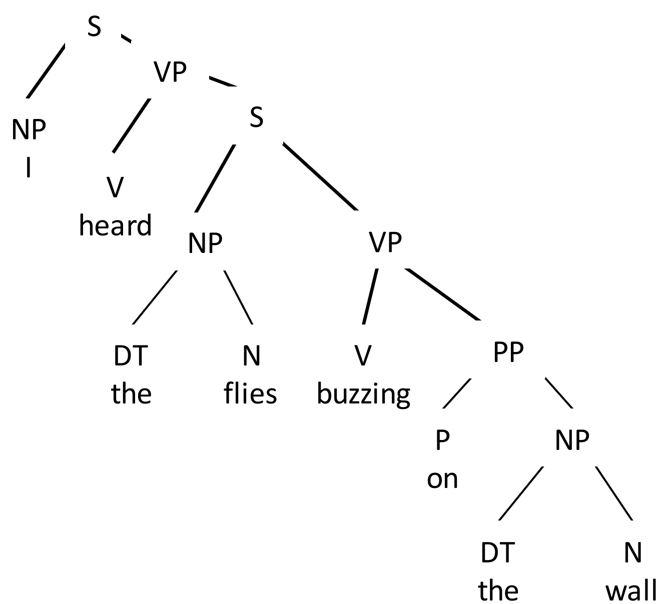
- **Parallel corpora**: data sets pairing source language sentences with their translations to the target language.
- Corpora often available for high-resource languages or language pairs:
  - **Hansards:** Extracted from Canadian Parliament proceedings; French/English
  - **Europarl:** Extracted from the proceedings of the European Parliament; 21 European languages

- Lesser-resourced languages can be related to other languages with a **pivot language**
  - **Samanantar (2021):** English and 11 Indic languages → 55 Indic language pairs

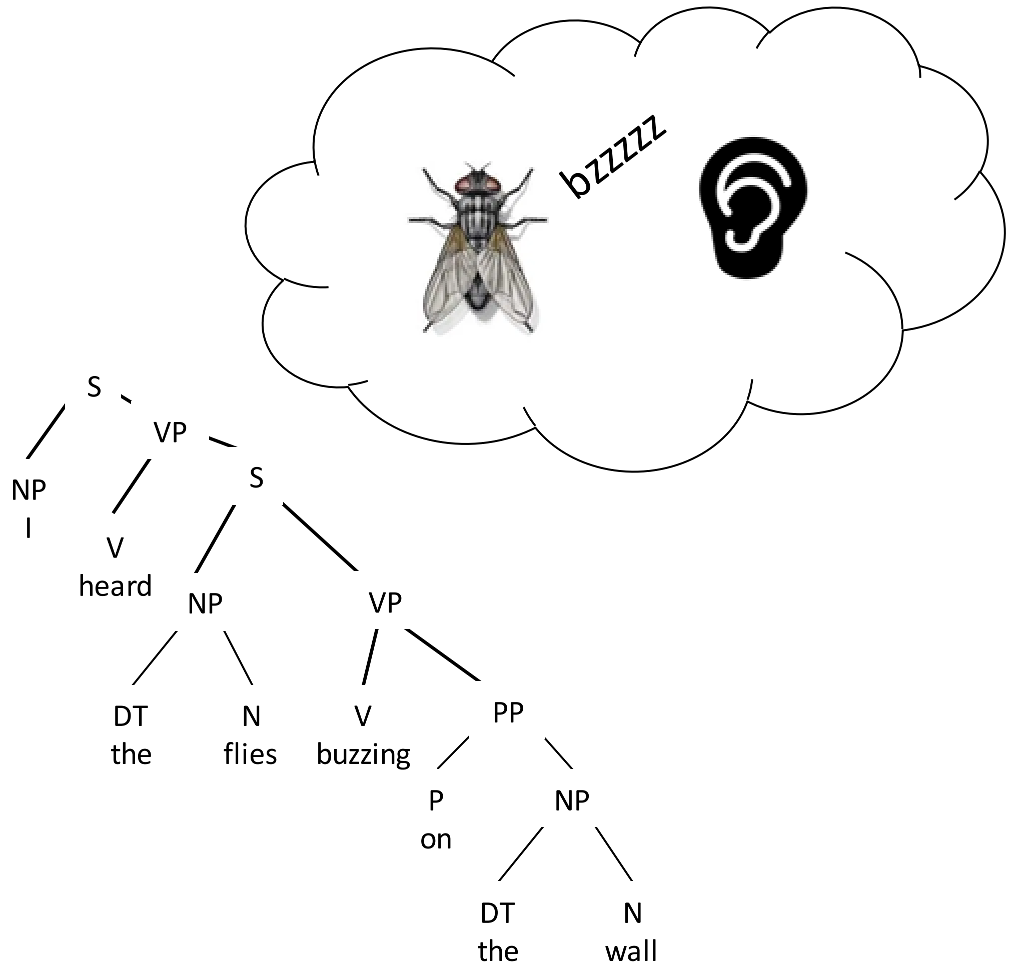
Resources for Machine Translation

- Also less-reliable **comparable corpora** -- e.g., Wikipedia articles, news reports in multiple languages about the same event
  - Comparable corpora can be mined to identify parallel sentence pairs can be extracted
- For evaluation, we may have **multiple reference translations** for each sentence

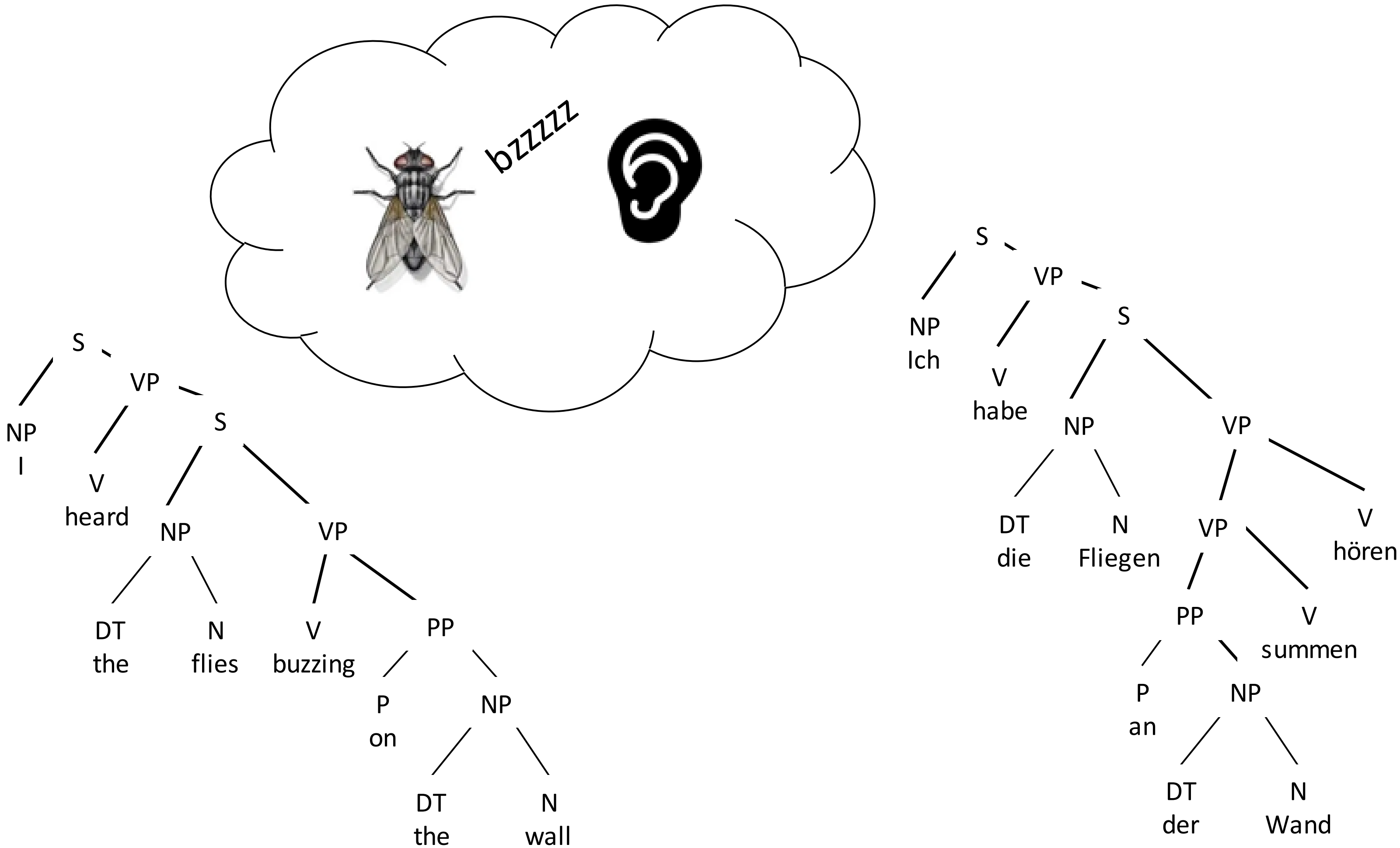
How Machine Translation does NOT work



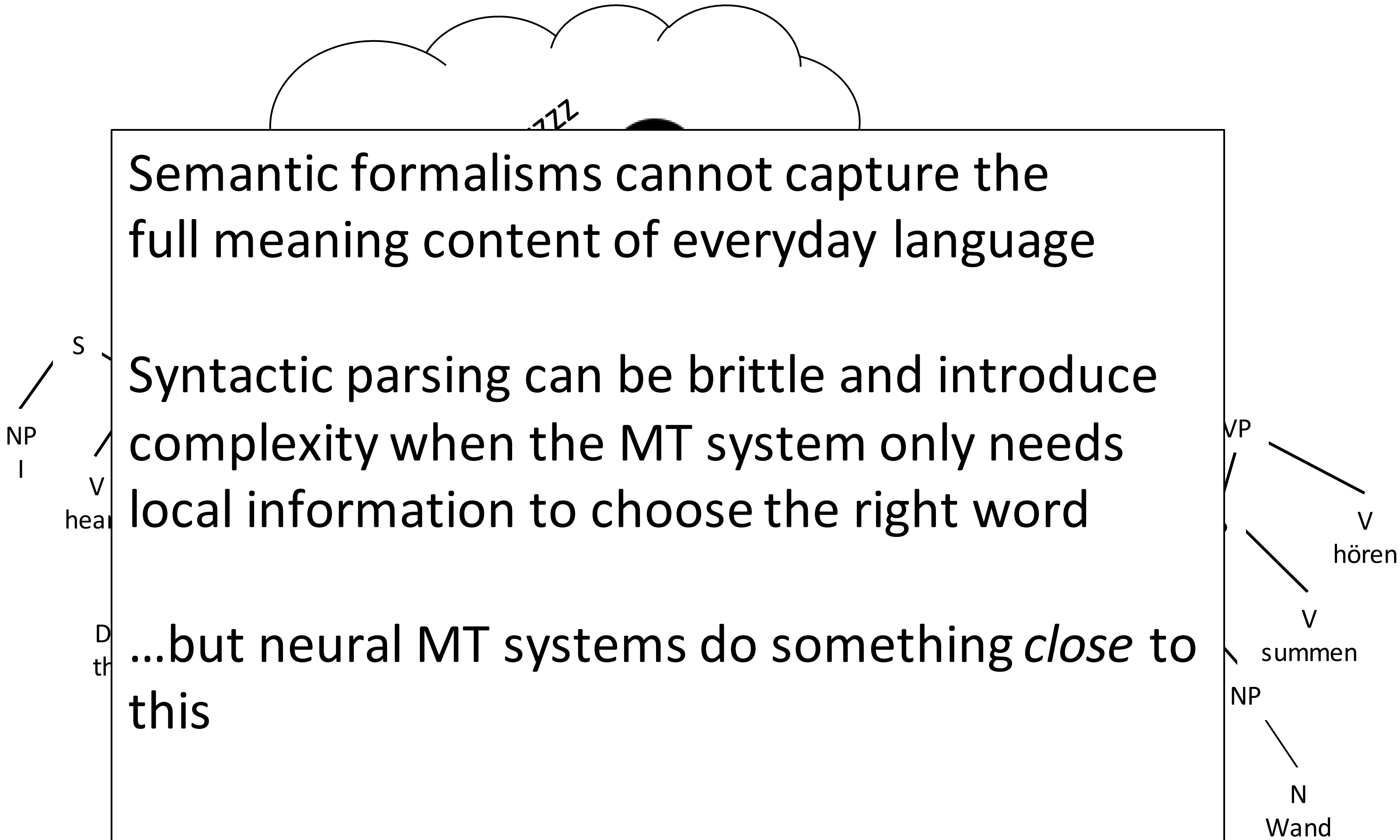
How Machine Translation does NOT work



How Machine Translation does NOT work



How Machine Translation does NOT work

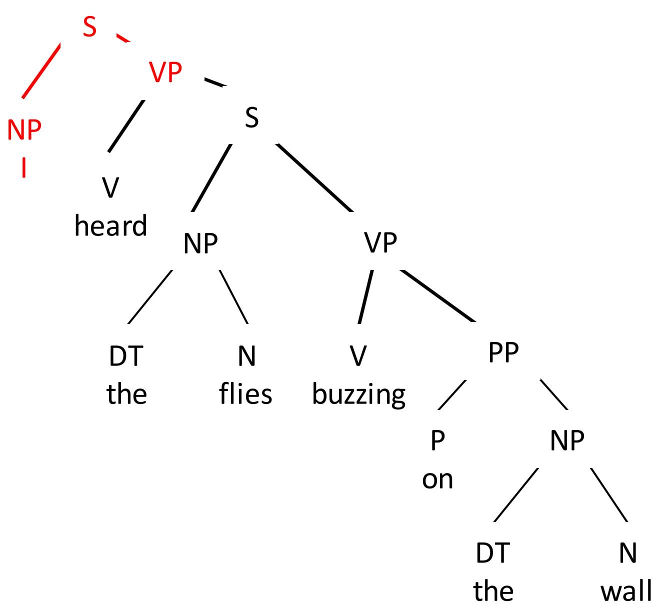


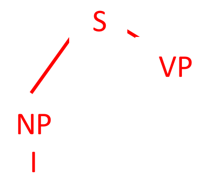
Semantic formalisms cannot capture the full meaning content of everyday language

Syntactic parsing can be brittle and introduce complexity when the MT system only needs local information to choose the right word

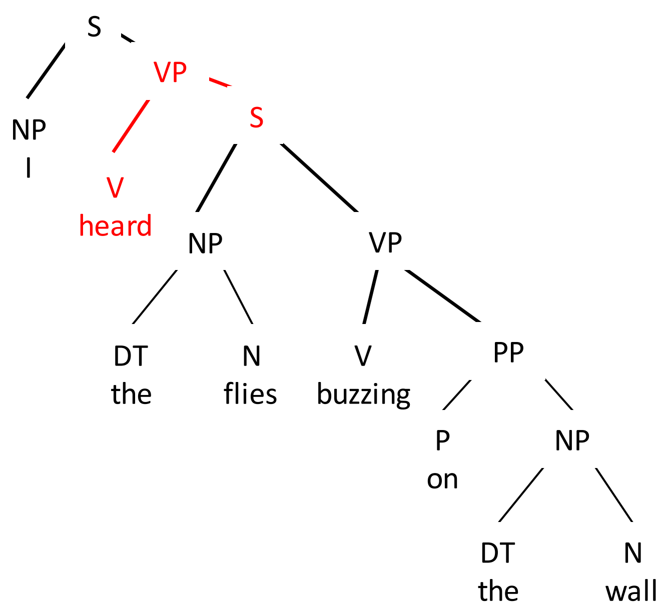
…but neural MT systems do something *close* to this

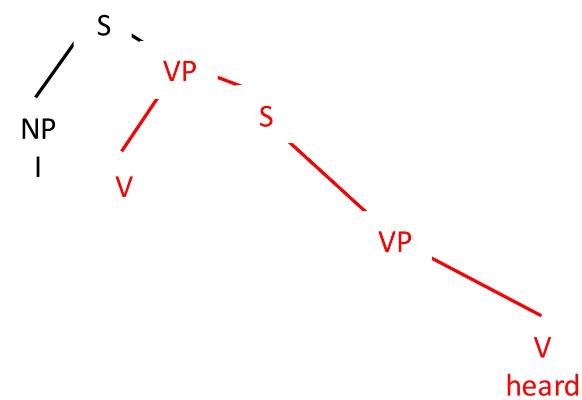
How Machine Translation does NOT work



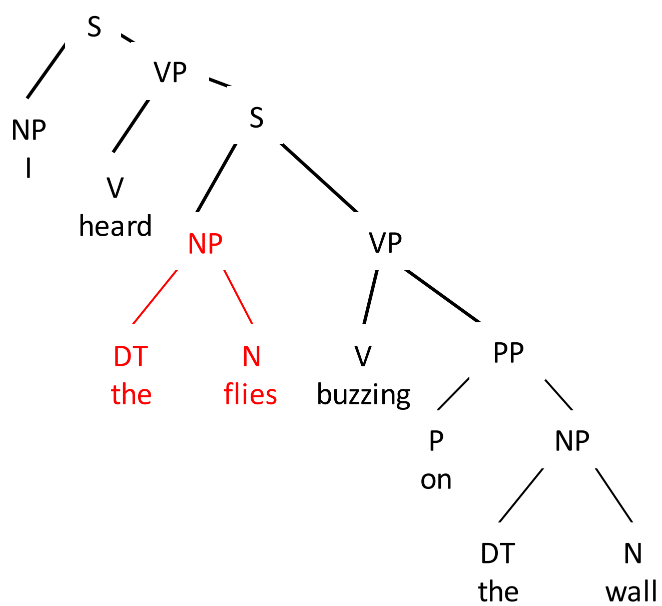


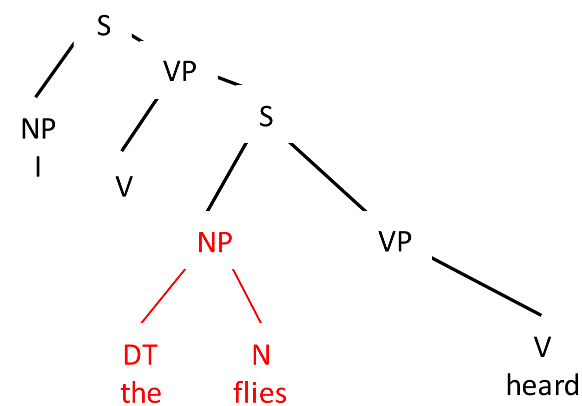
How Machine Translation does NOT work



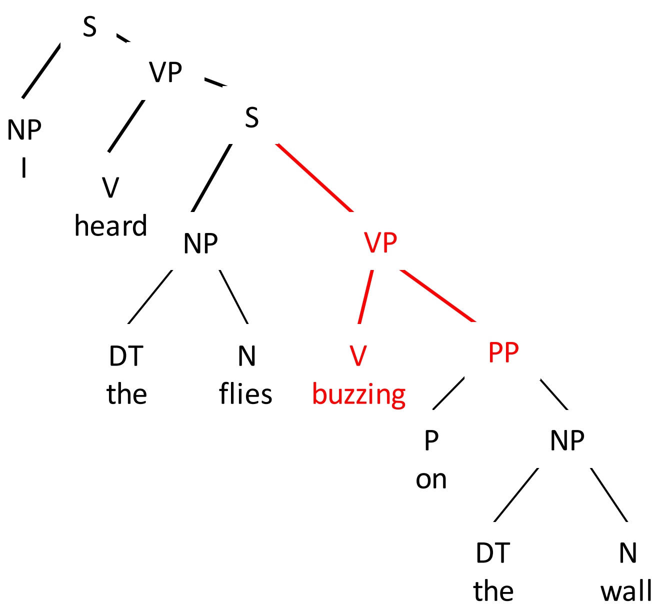


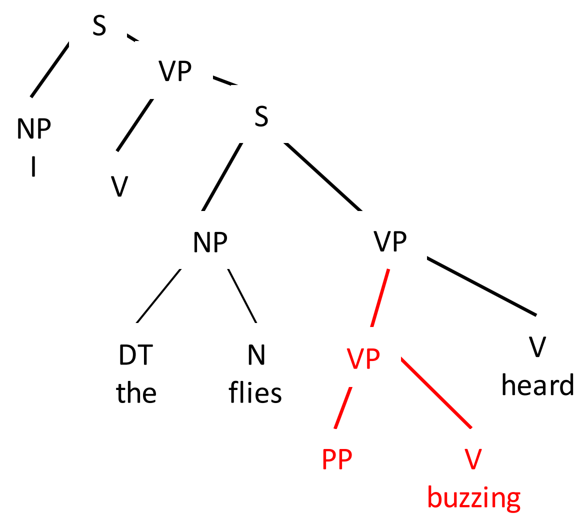
How Machine Translation does NOT work



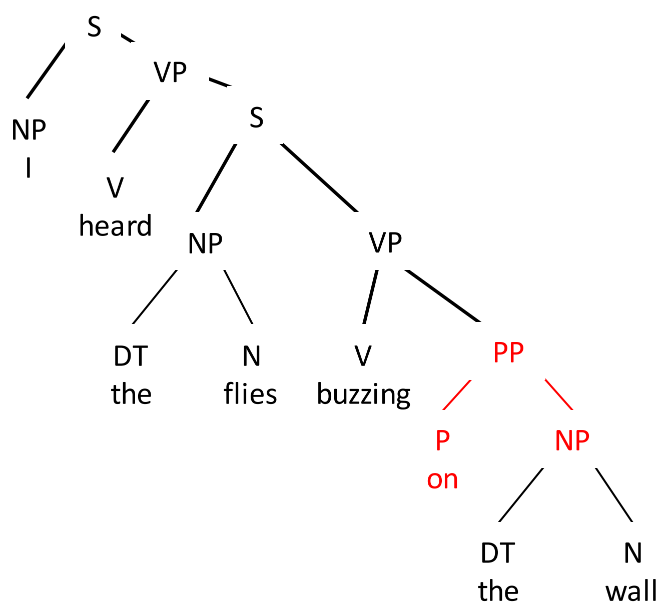


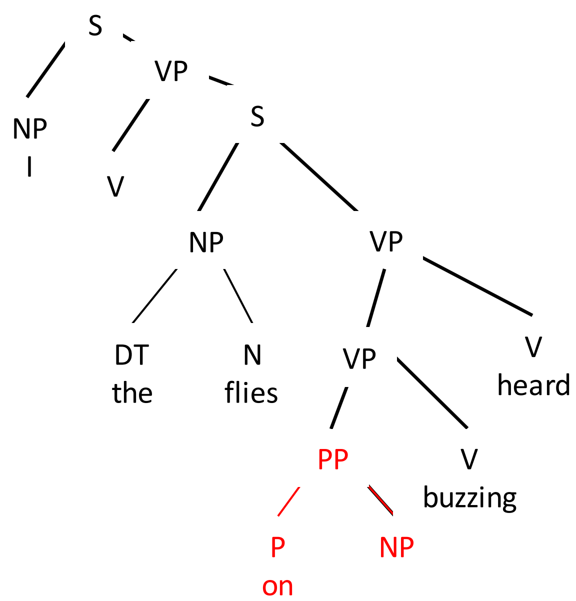
How Machine Translation does NOT work



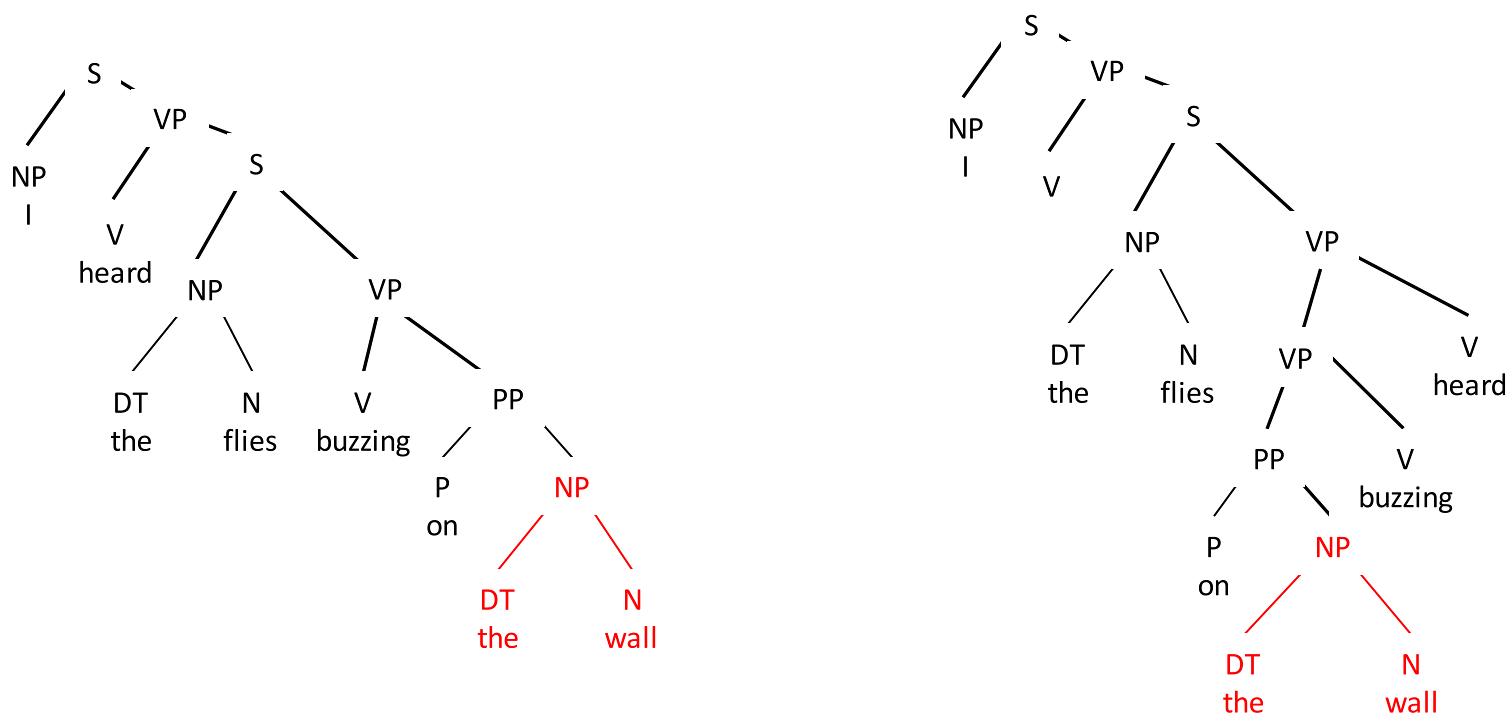


How Machine Translation does NOT work

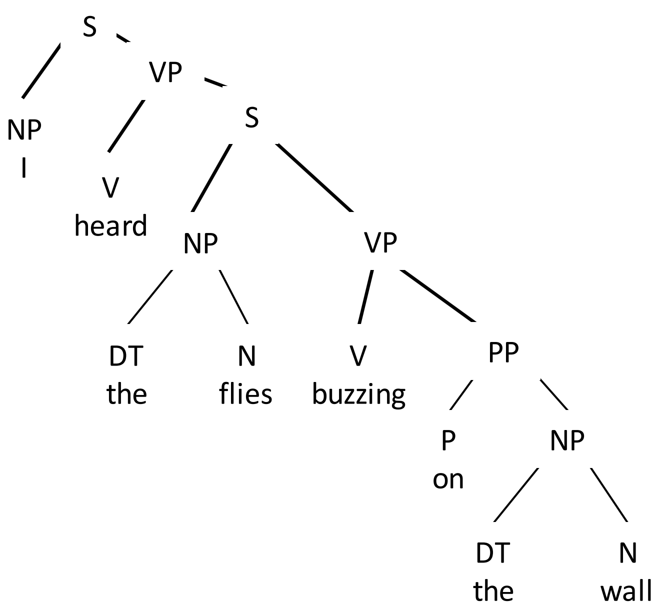


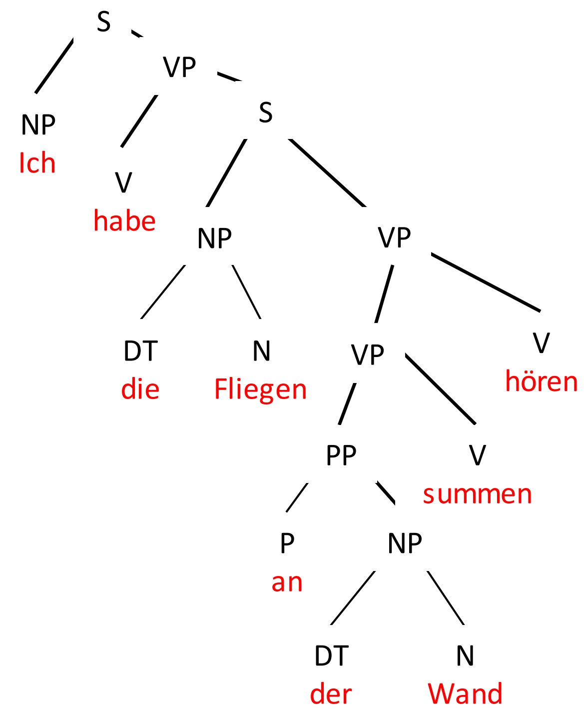


How Machine Translation does NOT work

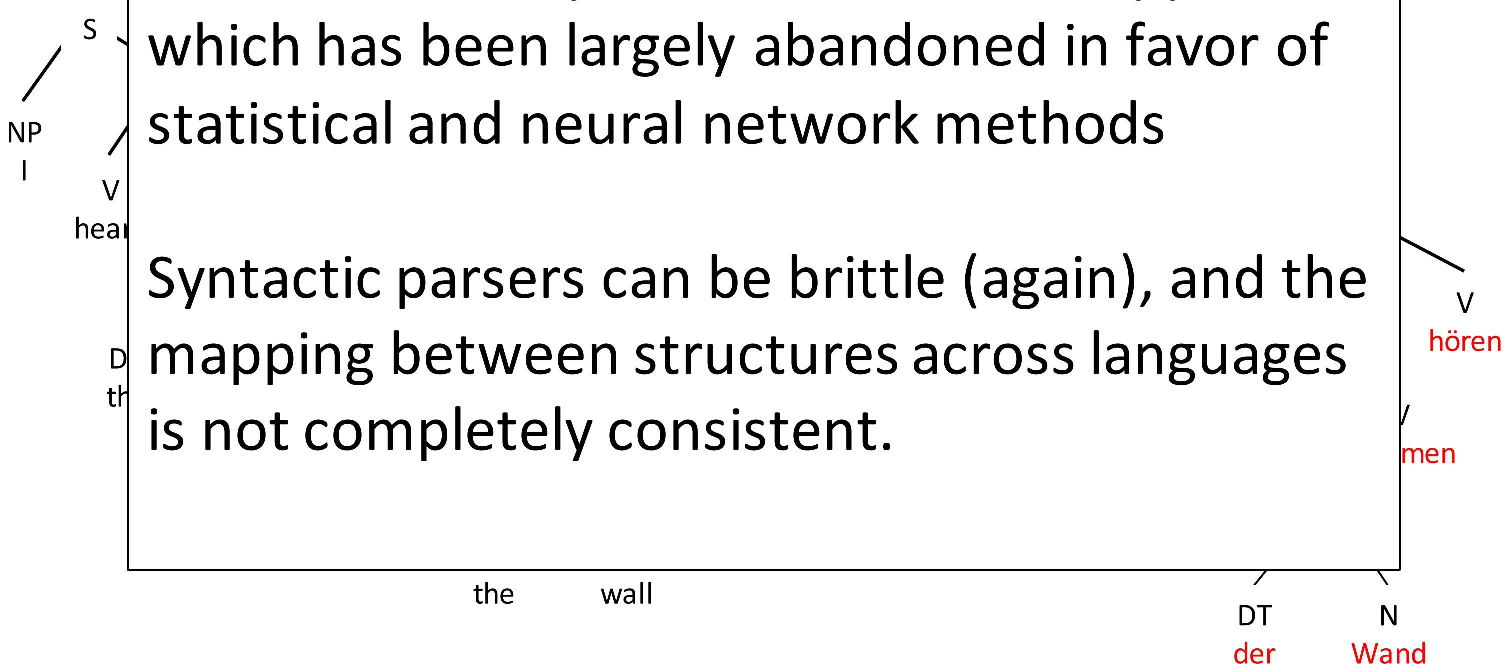


How Machine Translation does NOT work





How Machine Translation does NOT work



This is essentially the rule-based MT approach, which has been largely abandoned in favor of statistical and neural network methods

Syntactic parsers can be brittle (again), and the mapping between structures across languages is not completely consistent.

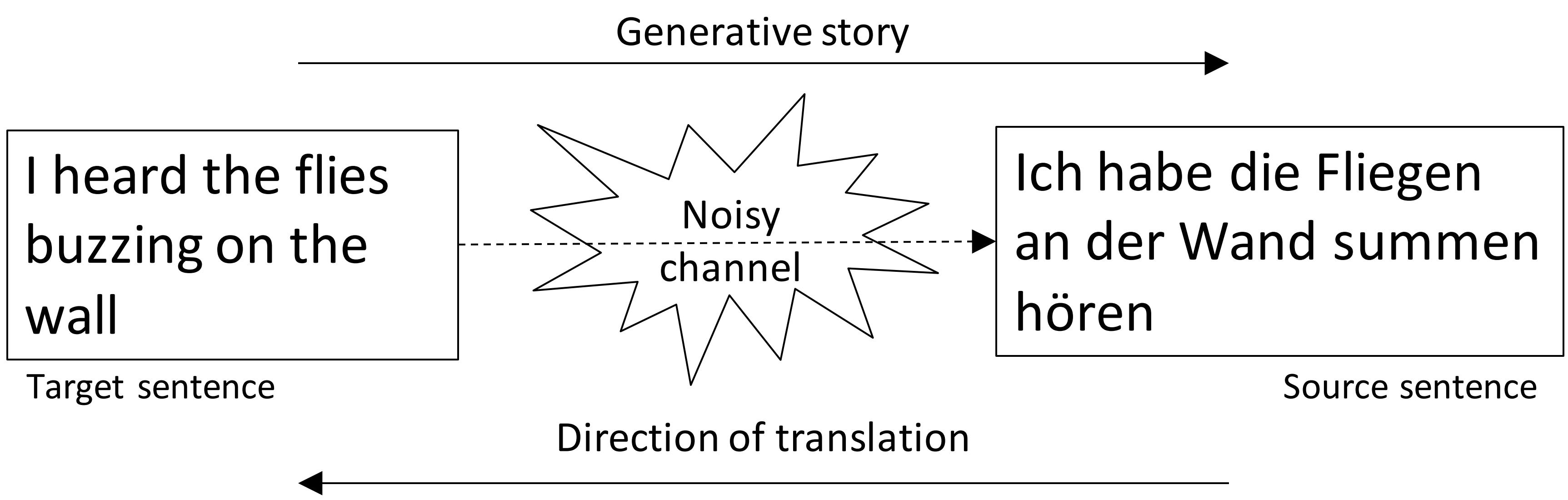
STATISTICAL MT

The Noisy Channel Model

- Statistical machine translation is based on the **noisy channel model**
  - Also used in data compression, speech recognition
- When we observe some sequence of symbols, we hypothesize that it actually came from a noisy encoding process
  - We started with a different symbol sequence
  - We passed it through a “noisy channel” that **obscured the original message**

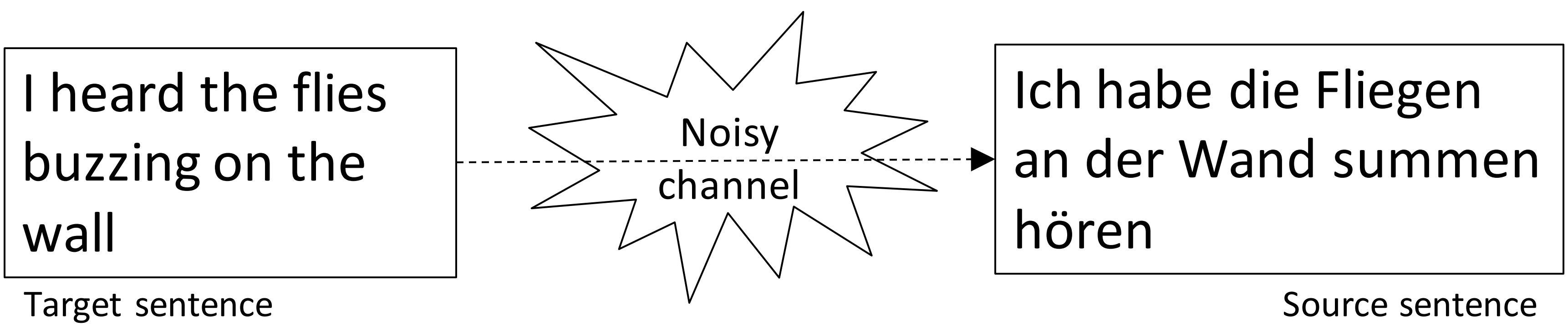
The Noisy Channel Model

- In speech recognition, we observe a sequence of auditory signals, and hypothesize that they came from an underlying sequence of words
- In MT, we observe a sequence of German words and hypothesize that they came from an underlying sequence of English words



The Noisy Channel Model

- The noisy channel model for statistical MT is a generative model, similar to HMMs in which we hypothesize that words are generated from a sequence of latent states (tags)

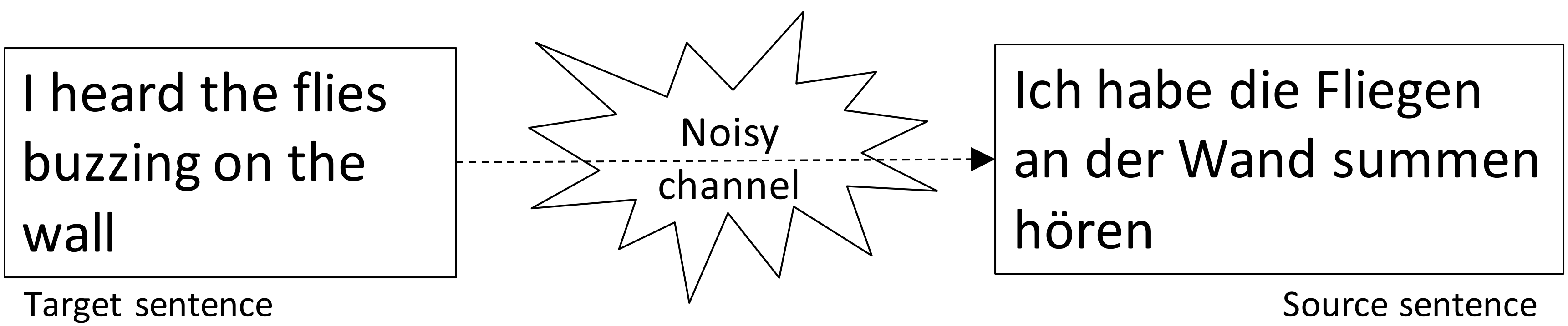


The Noisy Channel Model

- What we care about is $P(W_t|W_s)$, the probability of the target sentence given the source sentence that we observe. By Bayes' rule:

$$P(W_t|W_s) = \frac{P(W_t)P(W_s|W_t)}{P(W_s)}$$

$$\underset{W_t}{\operatorname{argmax}} P(W_t|W_s) = \underset{W_t}{\operatorname{argmax}}\big(P(W_t)P(W_s|W_t)\big)$$



The Noisy Channel Model

- $$\underset{W_t}{\operatorname{argmax}} P(W_t|W_s) = \underset{W_t}{\operatorname{argmax}} \left(P(W_t)P(W_s|W_t)\right)$$

Language model: how likely is a given sequence of words in the target language (English)

Translation model: how likely is a given sequence of words in the source language (German) given a sentence in the target language (English)

I heard the flies buzzing on the wall

Target sentence

Noisy channel

Ich habe die Fliegen an der Wand summen hören

Source sentence

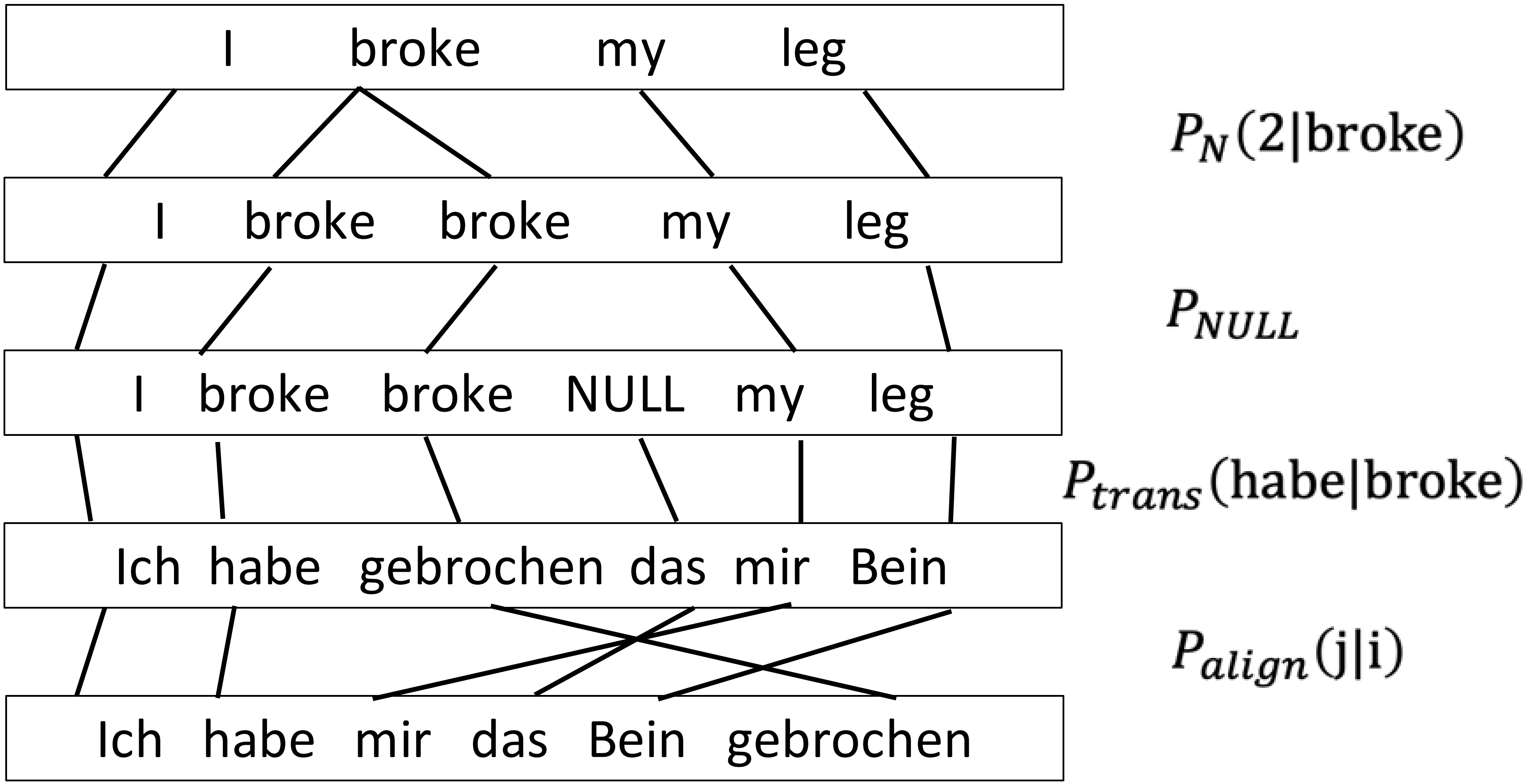
Three Problems in Statistical MT

- Language model $P(W_t)$
  - Assign high probabilities to well-formed sequences (English sentences) and low probabilities to random word sequences
  - We know how to do this!
- Translation model $P(W_s|W_t)$
  - Assign high probabilities to sentences that look like translations of one another, and low probabilities to random sentence pairs
- Decoding algorithm
  - Given a language model, a translation model, and a new sentence W_s … find translation W_t maximizing $P(W_t)P(W_s|W_t)$

Language model

- $P(W_t)$ can be estimated with language model tools
  - N-gram models
  - Count smoothing
  - Backoff and interpolation
  - …

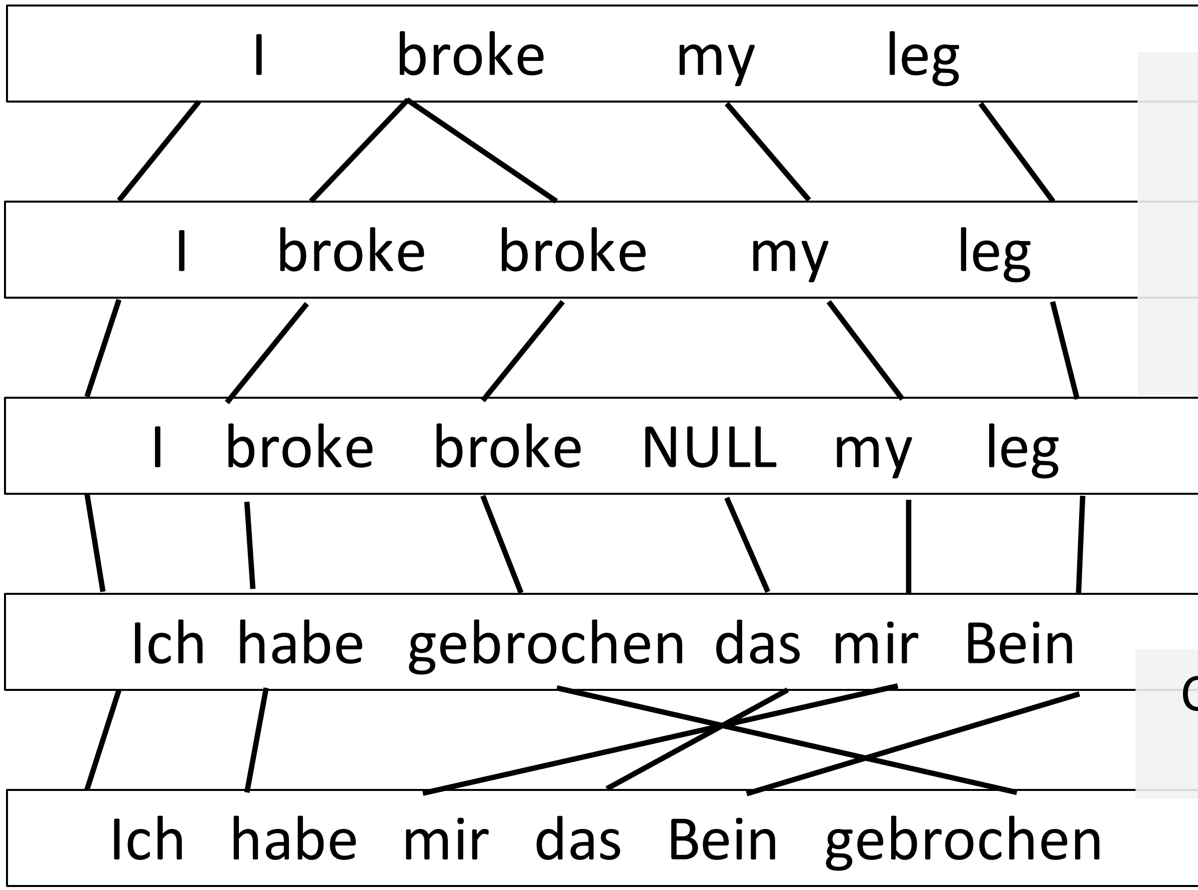
Translation model: generative story



Translation model: estimation

- We need to estimate
  - Fecundity: $P_N(n|\mathrm{w})$ – how often each word in the target language corresponds to multiple words in the source language
  - Null probabilities: P_{NULL} – the likelihood of a null insertion
  - Translation probabilities: $P_{trans}(w_s|w_t)$ – the probability of each source word being generated by a given target word
  - Alignment probabilities: $P_{align}(\mathrm{j}|\mathrm{i})$ = the likelihood of a word at position I aligning with a word at position j

Translation model: estimation



If we had a bunch of sentences annotated like this, we could just use maximum-likelihood estimation

Can we do it with just sentence pairs?

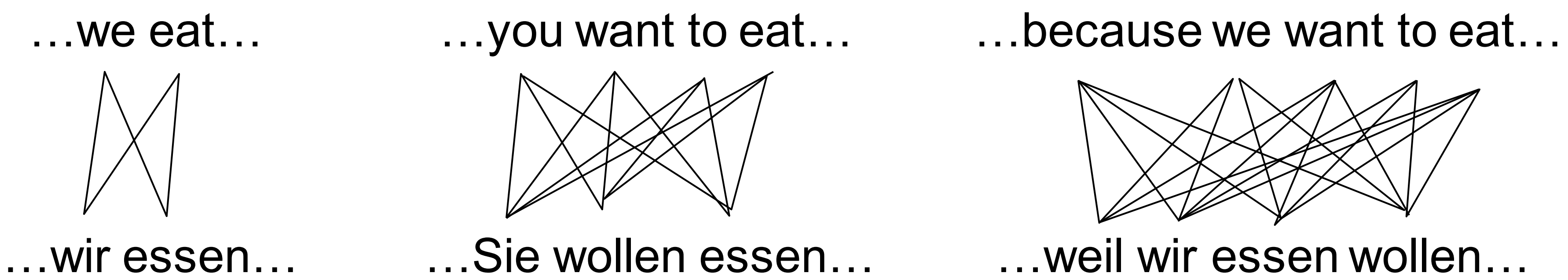
Translation model: estimation

- So, if we knew the **word alignments**, we could estimate all the necessary parameters of our model
- And if we had the **model parameters**, we could figure out the most likely alignments of words between sentences
- So, we're stuck.
- Or are we?

Remember EM?

- We used it for unsupervised learning of HMMs and Naïve Bayes
- Iteratively learn latent structure and model parameters for a generative model
- **Expectation**: Calculate the expected values of latent variables (alignments) using model parameters
- **Maximization**: Update model parameters to maximize likelihood of data under hypothesized alignments

EM for Alignment

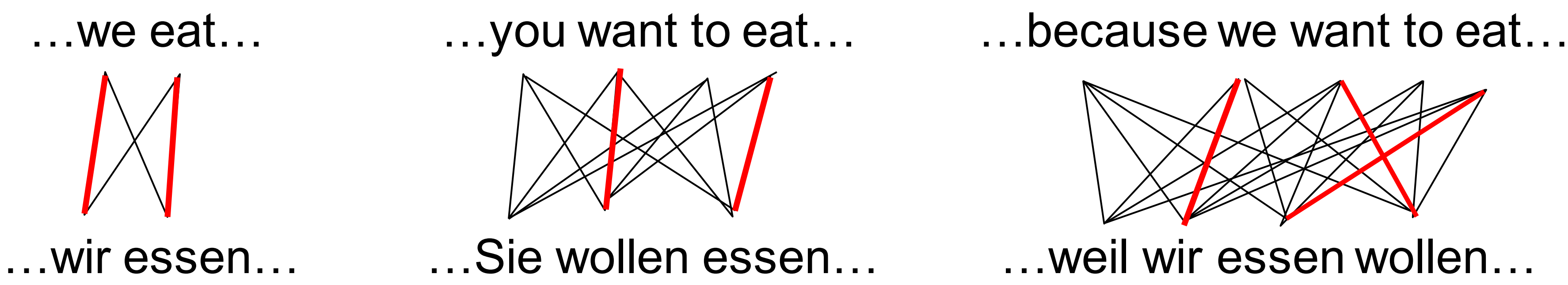


Initialization: assume all word alignments are equally probable

$$\forall w_s^1, w_s^2 \colon P_{trans}(w_s^1 | w_t) = P_{trans}(w_s^2 | w_t)$$

A given English word is equally likely to be the translation of any German word

EM for Alignment

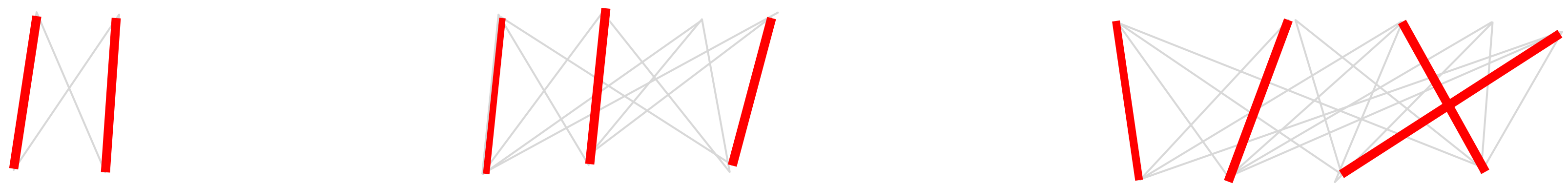


Certain words appear together frequently as possible alignments

P_{trans}(wollen|want) goes up

EM for Alignment

…we eat… …you want to eat… …because we want to eat…

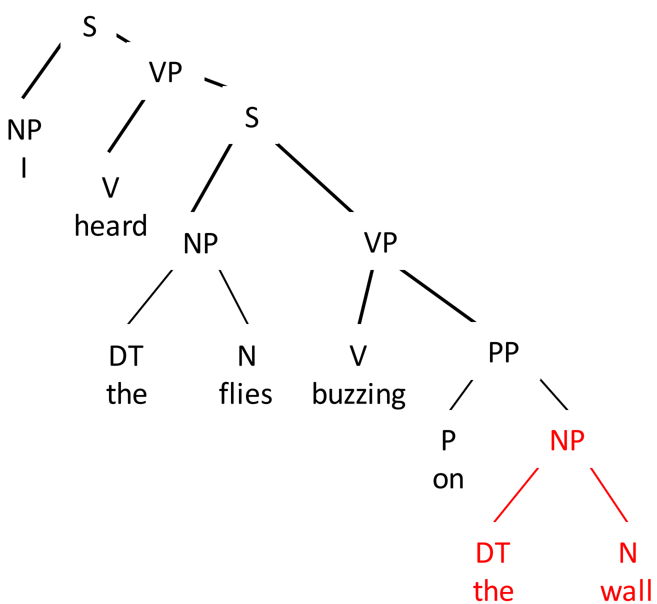


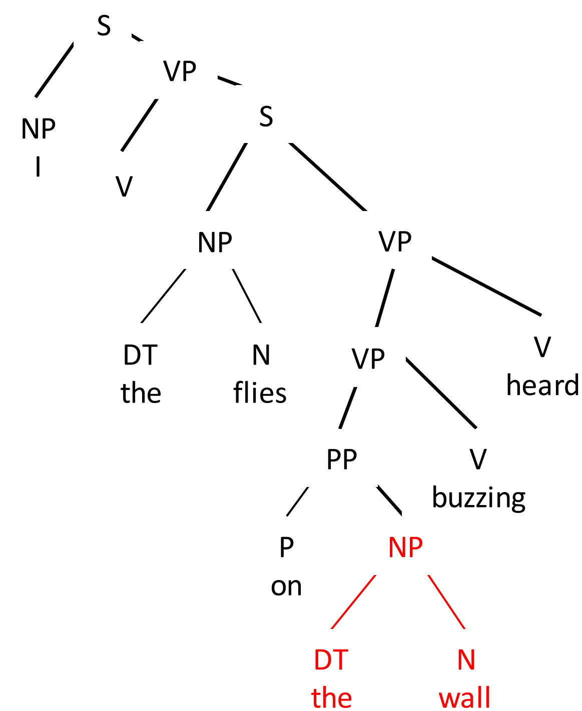
…wir essen… …Sie wollen essen… …weil wir essen wollen…

Associations are strengthened after a few iterations

Latent alignments are uncovered by EM algorithm

Remember: How Machine Translation does NOT work





Remember: How Machine Translation does NOT work

- A naïve approach:
  - use **syntactic** transformations to alter the structure from the source language into one suitable for the target language,
  - and then swap in the right words

- This is not how MT works, but there are **similarities** with the statistical approach
  - The alignment probabilities express structural differences between languages
  - The translation probabilities between words are a lot like the relexicalization step of the naïve model

Decoding for statistical MT

- Decoding: find the most likely translation for a given sentence in the source language

$$\underset{W_t}{\operatorname{argmax}}\big(P(W_t)P(W_s|W_t)\big)$$

- Here as well, we can use the Viterbi algorithm to decode efficiently
  - Keep track of best path resulting in a given partial analysis with specific alignment characteristics

Phrase-based machine translation

- So far, we have been talking about ***word-based*** statistical MT: each source word is aligned with at most one target word
- This has some drawbacks
  - Some weird alignments due to grammatical differences between languages: `mir – my`
  - Asymmetry: if we're translating German to English, we can have multiple German words aligned with an English word, but not vice-versa
  - Non-compositional meaning: "hot potato", "hard drive", "real estate"
  - Difficulty in handling widely-differing word orders (German verb-final)

Phrase-based machine translation

I | heard | the flies | buzzing | on the wall

Ich | habe | die Fliegen | an der Wand | summen | hören

- An alternative is **phrase-based MT**
- Source (foreign) input segmented in to phrases
- Each phrase is probabilistically translated into English
  - P(on the wall | an der Wand)
  - **Huge table of translation probabilities**
- Phrases are then probabilistically re-ordered

NEURAL MT

Machine translation with neural networks

- Primary framework: *encoder-decoder* model
- Source sentence is encoded using one part of the model to produce a meaning representation (a vector)

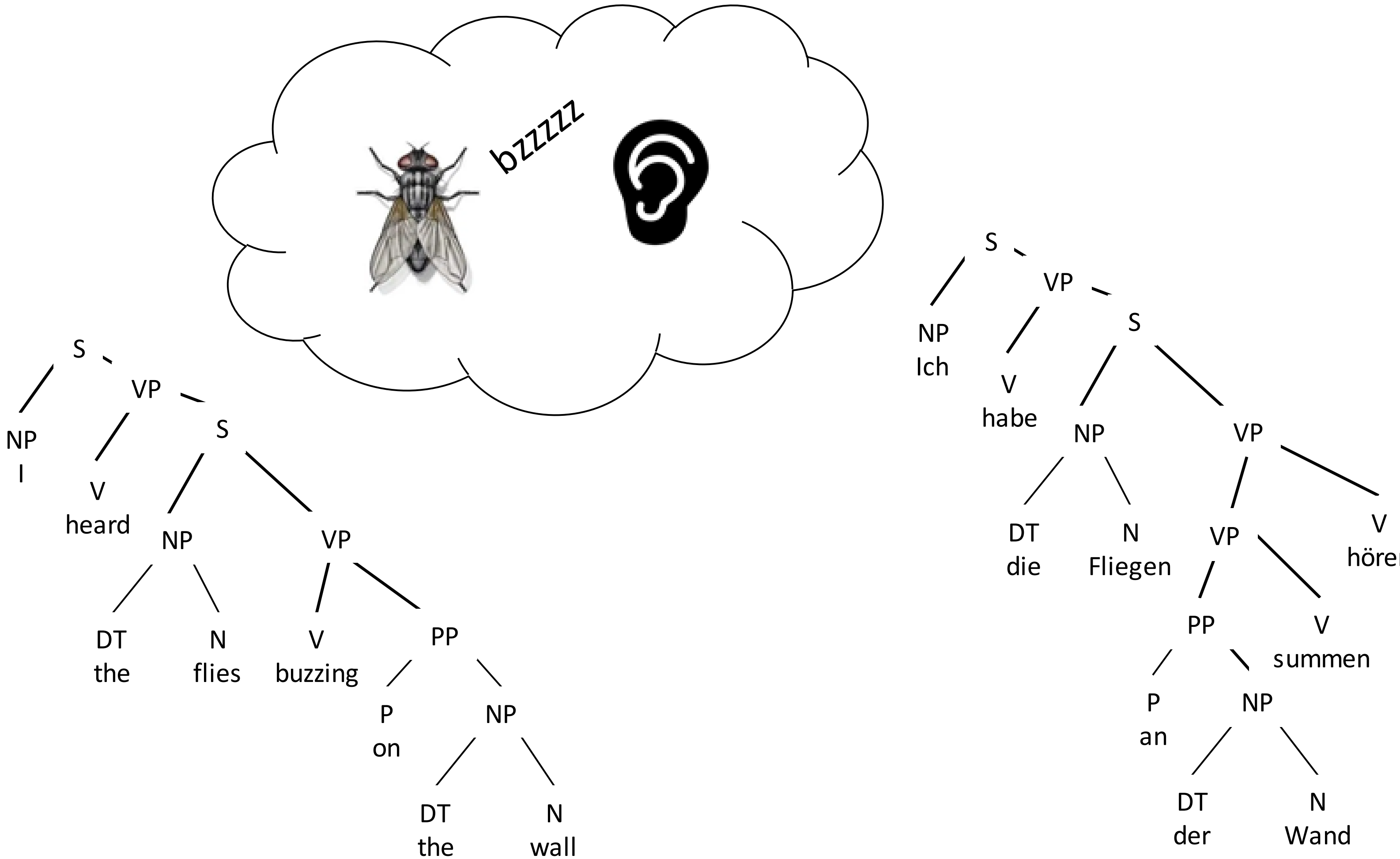
$$z = \text{Encode}(W_s)$$

- That vector is fed to a decoder model that predicts each word of the target sentence in turn (or an `END` token)

$$W_t = \text{Decode}(z)\text{, or}$$

$$W_t = \text{Decode}(z, W_s)$$

Remember: How Machine Translation does NOT work



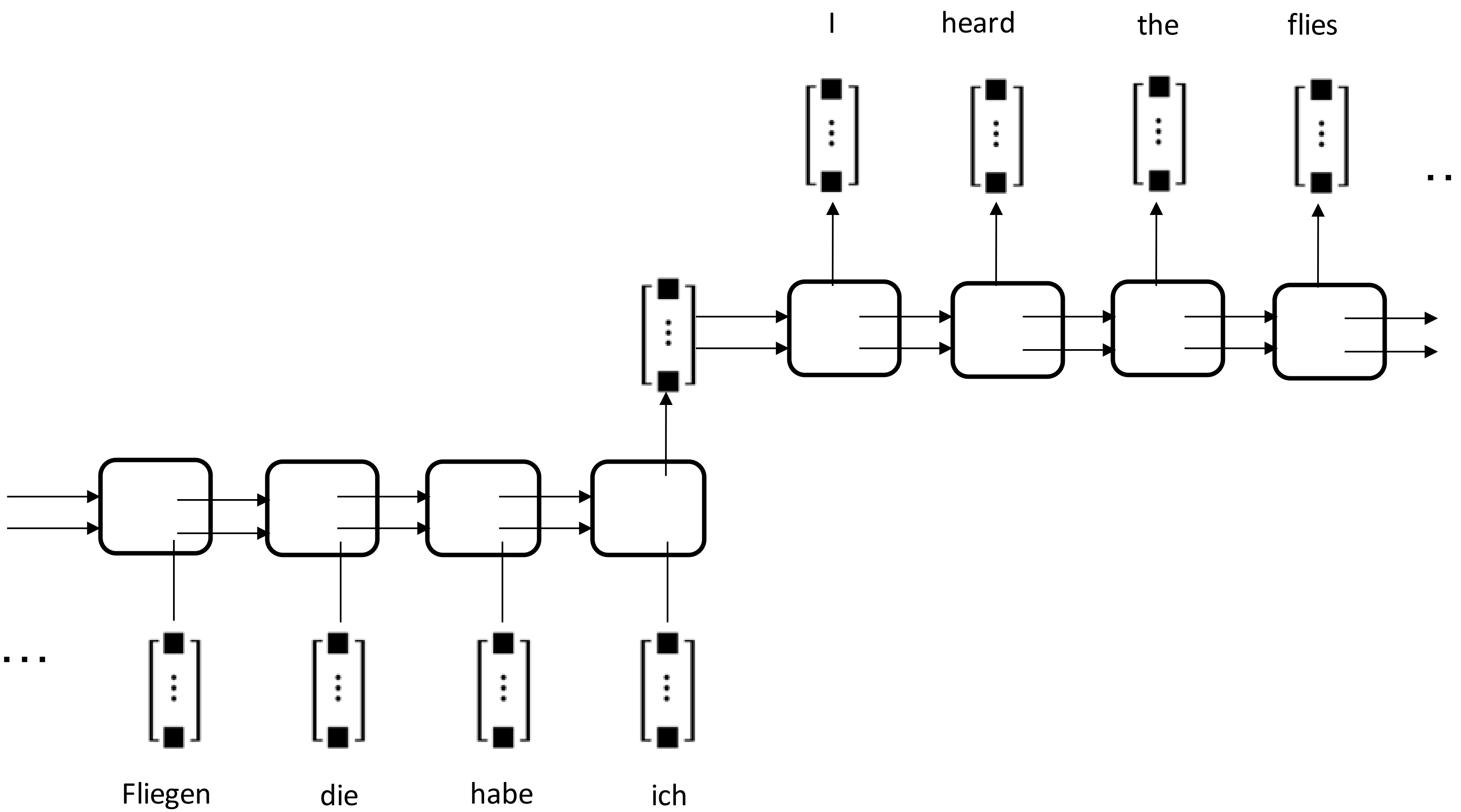
Remember: How Machine Translation does NOT work

- We discussed a naïve approach:
  - construct a semantic representation for the meaning of the source sentence
  - and then use this to generate an appropriate target sentence with appropriate syntactic structure and lexical choices
- Actually kind of similar to encoder-decoder model
  - But semantic representation is learned and not amenable to inspection
  - Syntactic generation rules are latent in structure of neural network

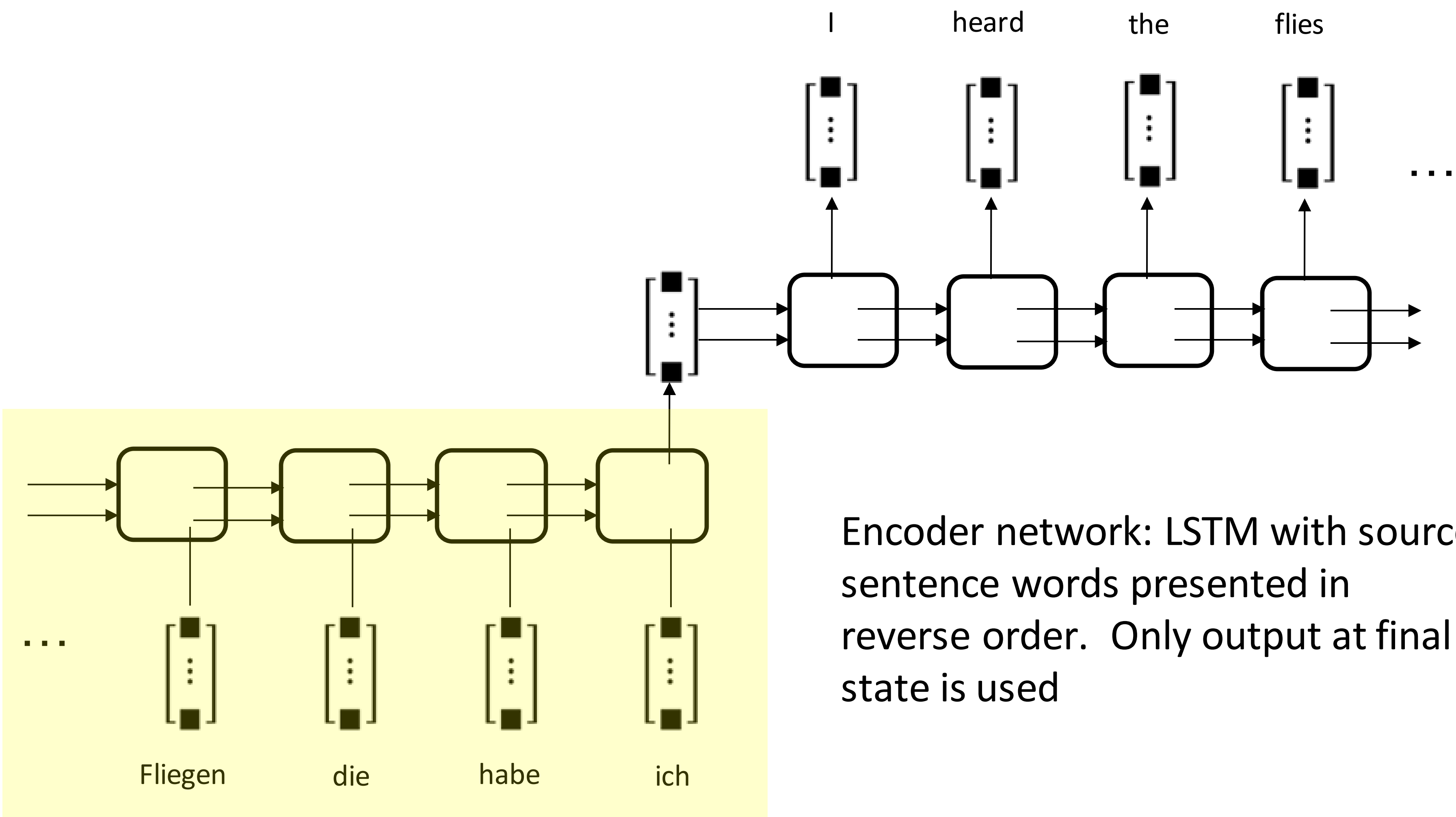
`seq2seq`

- Different encoder-decoder models may have slightly different architectures
  - Encoder and decoder may be recurrent, convolutional, or transformers
  - Decoder may have inputs other than the encoder output z
- The sequence-to-sequence (`seq2seq`) model is a relatively simple version of an encoder-decoder approach
  - Encoder and decoder are LSTMs
  - Decoder operates directly on the encoded representation z, with no other inputs

`seq2seq` model

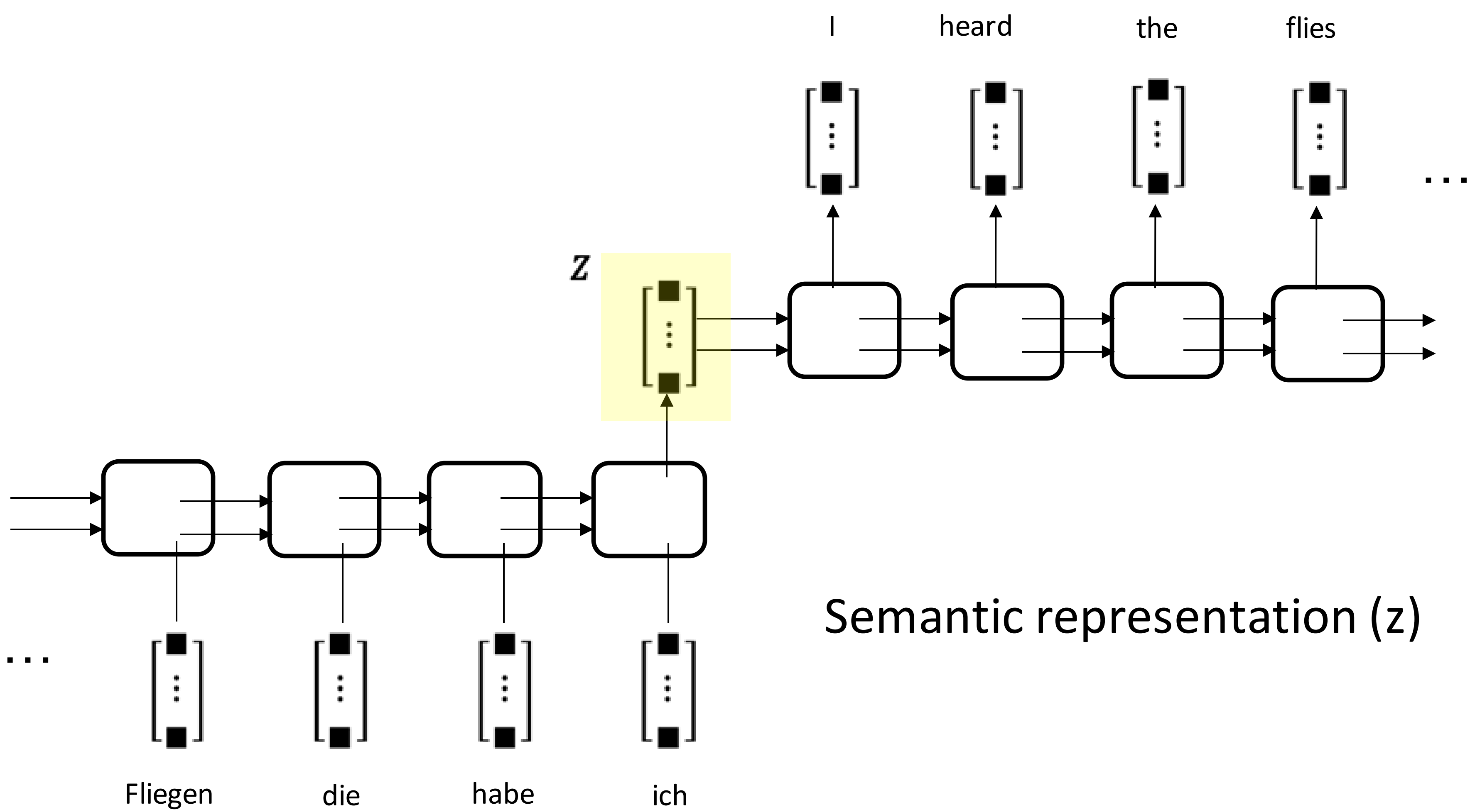


seq2seq model



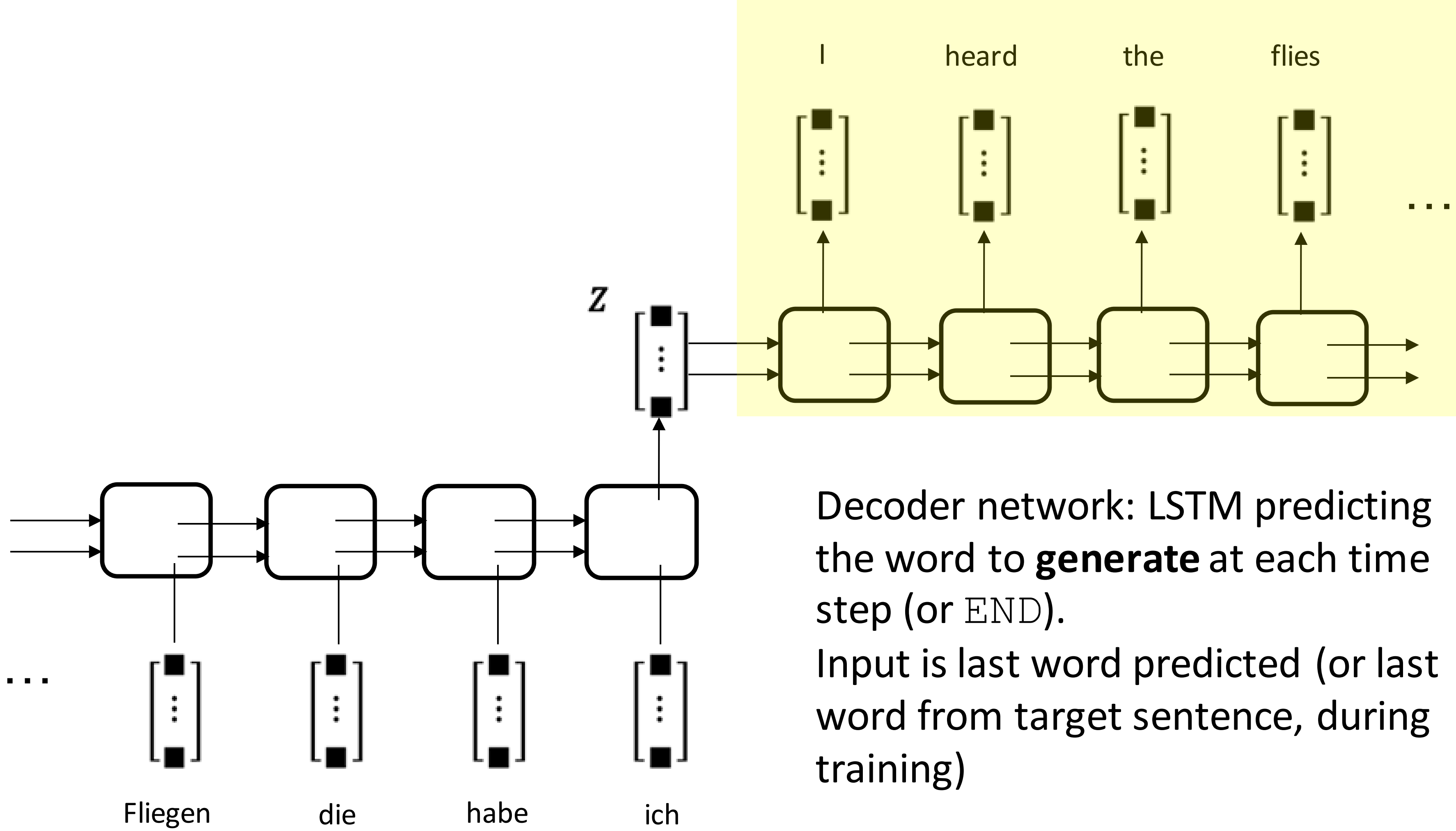
Encoder network: LSTM with source sentence words presented in reverse order. Only output at final state is used

seq2seq model



Semantic representation (z)

`seq2seq` model

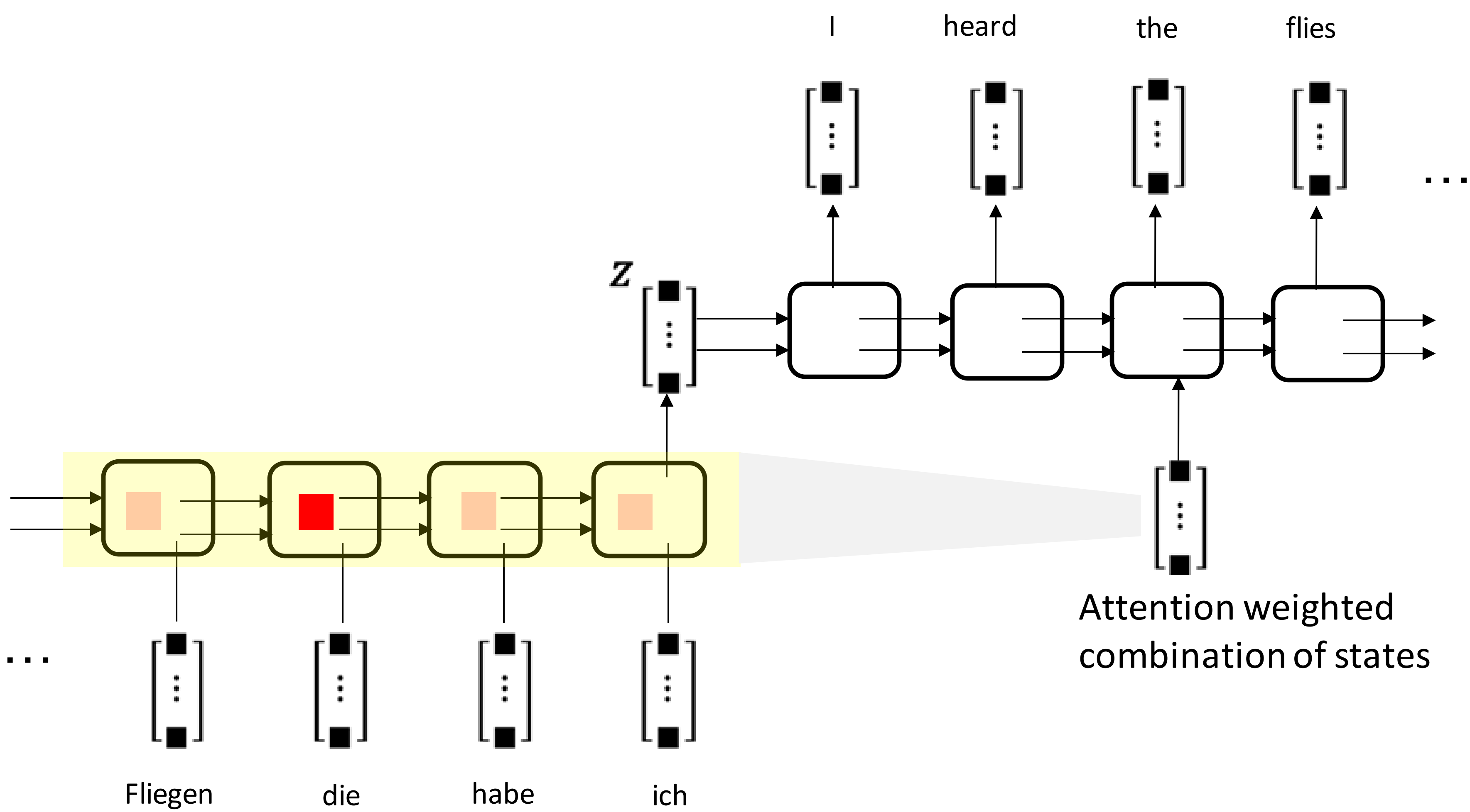


Decoder network: LSTM predicting the word to **generate** at each time step (or `END`).
Input is last word predicted (or last word from target sentence, during training)

Attention for Neural MT

- More complex neural translation models also use information from the source word sequence in the decoder phase
- An **attention function** is used to weight the representations at each word index from the source sentence
- Resulting attention weights are analogous to alignment links in statistical MT

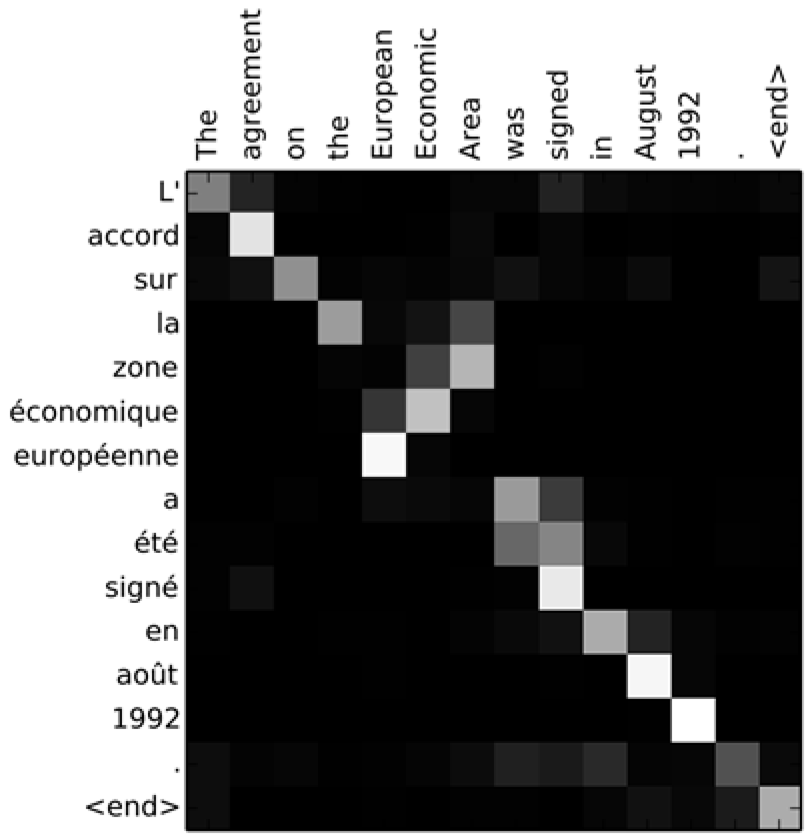
Attention



I
heard
the
flies
...
z
...
Fliegen
die
habe
ich
Attention weighted
combination of states

Attention

Attention weights show a *soft alignment* of input and output sequences



Bahdanau, D. et al. 2015. Neural Machine Translation by Jointly Learning to Align and Translate. ICLR.

Decoding for Neural MT

- Choosing the highest-probability word from the decoder as we go is **greedy search**
- Could run into problems with garden-path structures, where a translation starts off looking promising, but runs into a dead end

The cotton clothing is made of grows in Mississippi	⟹	Kleidung aus Baumwolle …

- Alternatively, use **beam search** to retain the *k* (beam width) best hypotheses at each decoding step

MT EVALUATION

The MT evaluation problem

- We have
  - a source language sentence paired with a gold translation in the target language, produced by a human translator
    - Maybe more than one
  - a translation for that source language sentence produced by our MT system
- We want to know
  - How good is our translation?
  - Is it better than translations produced by other systems?

Approach 1: exact match

Die einzige überstehende deutsche Mannschaft in der Achtelfinale war Dortmund.

source

The only outstanding German team in the second round was Dortmund.

predicted target translation

Dortmund was the only remaining German team in the round of 16.

Of the German sides, only Dortmund advanced to the round of 16.

Dortmund was the sole German representative in the round of 16.

reference translations

No partial credit! But is our translation that bad?

Approach 2: qualitative review

Die einzige überstehende deutsche Mannschaft in der Achtelfinale war Dortmund.

source

The only outstanding German team in the second round was Dortmund.

predicted target translation

Adequacy: 4/10
Fluency: 9/10

Anthea Bell, German-English translator

Not feasible - human review is expensive and time consuming

Approach 3: approximate metrics

Die einzige überstehende deutsche Mannschaft in der Achtelfinale war Dortmund.

source

The only outstanding German team in the second round was Dortmund.

predicted target translation

Dortmund was the only remaining German team in the round of 16.

Of the German sides, only Dortmund advanced to the round of 16.

Dortmund was the sole German representative in the round of 16.

reference translations

BLEU: 0.279 ← Automatically generated score

BLEU Score

- The idea of the BLEU score is that better translations will have more **word sequences that overlap** with reference translations than bad translations do
- We can get a finer grained evaluation than exact match, but without the need to consult human translators for every prediction
- Related metrics: ROUGE (based on Recall), METEOR (based on F1)

BLEU Score: Details

Concretely, BLEU calculates an average of n-gram **precision** across different n-gram orders

- N-gram: sequence of n adjacent words
- N-gram orders: unigrams, bigrams, trigrams, etc.
- How many of the words (or bigrams, or trigrams, etc.) in the predicted translation are found in the reference translations?

BLEU Score: Details

- $$\text{BLEU}{-}4\left(\widehat{W}_t, W_t^{ref_1} \dots W_t^{ref_n}\right) = \left(\prod_{i=1}^{4} \text{ModifiedPrecision}_i\left(\widehat{W}_t, W_t^{ref_1} \dots W_t^{ref_n}\right)\right)^{\frac{1}{4}}$$

Precision of words/n-grams in translation relative to reference

- Modified to prevent credit being given for including a word more often than it shows up in reference

BLEU tends to favor shorter translations, so a "brevity penalty" is also applied

BLEU Example

Unigram precision:

9/11

The only outstanding German team in the second round was Dortmund.

Dortmund was the only remaining German team in the round of 16.

Of the German sides, only Dortmund advanced to the round of 16.

Dortmund was the sole German representative in the round of 16.

BLEU Example

Bigram precision:

4/10

The only outstanding German team in the second round was Dortmund.

the only
only outstanding
outstanding German
German team
team in
in the
the second
second round
round was
was Dortmund

Dortmund was the only remaining German team in the round of 16.

Of the German sides, only Dortmund advanced to the round of 16.

Dortmund was the sole German representative in the round of 16.

BLEU Example

Trigram precision:
2/9

The only outstanding German team in the second round was Dortmund.

the only outstanding
only outstanding German
outstanding German team
German team in
team in the
in the second
the second round
second round was
round was Dortmund

Dortmund was the only remaining German team in the round of 16.

Of the German sides, only Dortmund advanced to the round of 16.

Dortmund was the sole German representative in the round of 16.

BLEU Example

4-gram precision:

1/8

The only outstanding German team in the second round was Dortmund.

the only outstanding German
only outstanding German team
outstanding German team in
German team in the
team in the second
in the second round
the second round was
second round was Dortmund

Dortmund was the only remaining German team in the round of 16.

Of the German sides, only Dortmund advanced to the round of 16.

Dortmund was the sole German representative in the round of 16.