Natural Language Processing (CS-472) Spring-2023

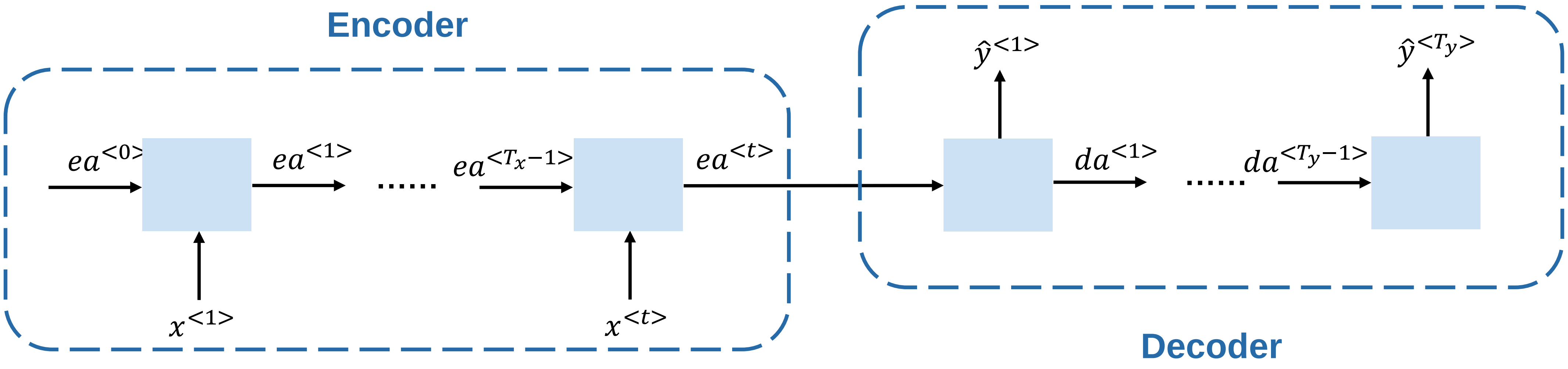
Muhammad Naseer Bajwa
Assistant Professor,
Department of Computing, SEECS
Co-Principal Investigator,
Deep Learning Lab, NCAI
NUST, Islamabad
naseer.bajwa@seecs.edu.pk

Overview of this week’s lecture

Machine Translation

- Statistical machine translation
- Neural network based machine translation
- Attention in seq2seq models

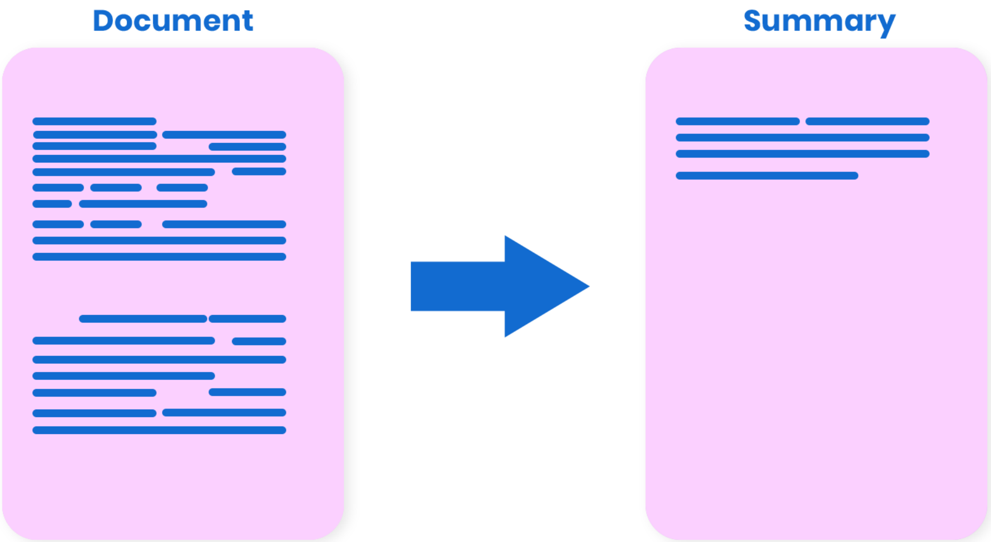
seq2seq models take sequential inputs and generate sequential outputs



Many-to-Many V2

seq2seq models are quite versatile

- Many other NLP tasks can be phrase as seq2seq problems.
  - Summarisation



seq2seq models are quite versatile

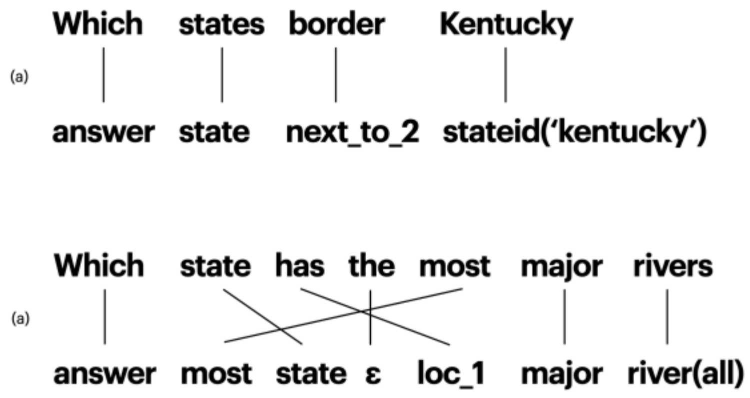
- Many other NLP tasks can be phrase as seq2seq problems.
  - Summarisation
  - Dialogue

	Utterance	Dialogue act
U:	Hi, I am looking for somewhere to eat.	`hello(task = find,type=restaurant)`
S:	You are looking for a restaurant. What type of food do you like?	`confreq(type = restaurant, food)`
U:	I'd like an Italian somewhere near the museum.	`inform(food = Italian, near=museum)`
S:	Roma is a nice Italian restaurant near the museum.	`inform(name = "Roma", type = restaurant, food = Italian, near = museum)`
U:	Is it reasonably priced?	`confirm(pricerange = moderate)`
S:	Yes, Roma is in the moderate price range.	`affirm(name = "Roma", pricerange = moderate)`
U:	What is the phone number?	`request(phone)`
S:	The number of Roma is 385456.	`inform(name = "Roma", phone = "385456")`
U:	Ok, thank you goodbye.	bye()

Figure 26.13 A sample dialogue from the HIS System of Young et al. (2010) using the dialogue acts in Fig. 26.12.

seq2seq models are quite versatile

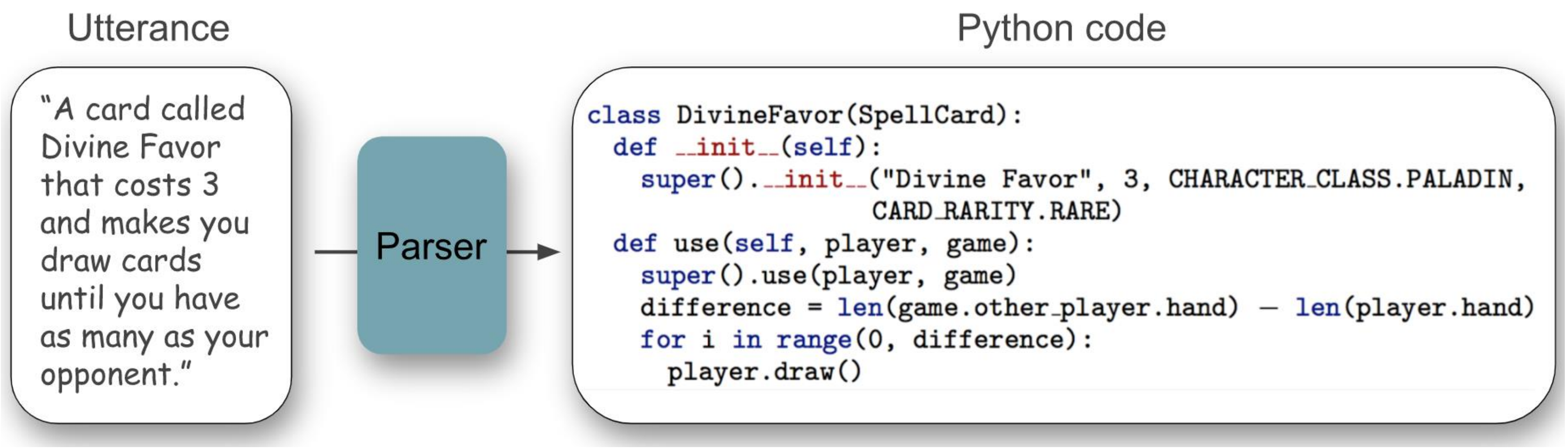
- Many other NLP tasks can be phrase as seq2seq problems.
  - Summarisation
  - Dialogue
  - Semantic Parsing



https://arxiv.org/pdf/2205.08288.pdf

seq2seq models are quite versatile

- Many other NLP tasks can be phrase as seq2seq problems.
  - Summarisation
  - Dialogue
  - Semantic Parsing
  - Code Generation



seq2seq models are quite versatile

- Many other NLP tasks can be phrase as seq2seq problems.
  - Summarisation
  - Dialogue
  - Semantic Parsing
  - Code Generation
  - Language Translation

Diese Woche haben wir einen zusätzlichen Vortrag.

This week we are having on additional lecture.

Language translation is one of the prime use-cases of seq2seq models

- The task of language translation requires converting an input sentence $\boldsymbol{x}$ from a source language to an output sentence $\boldsymbol{y}$ in the target language preserving the semantic information.

I play the flute. ⟹

Language translation is one of the prime use-cases of seq2seq models

- The task of language translation requires converting an input sentence $\boldsymbol{x}$ from a source language to an output sentence $\boldsymbol{y}$ in the target language preserving the semantic information.

I play the flute. → میں بانسری بجاتا ہوں۔

Language translation is one of the prime use-cases of seq2seq models

- The task of language translation requires converting an input sentence $\boldsymbol{x}$ from a source language to an output sentence $\boldsymbol{y}$ in the target language preserving the semantic information.

I play the flute. $\Longrightarrow$ میں بانسری بجاتا ہوں۔

I play cricket. $\Longrightarrow$

Language translation is one of the prime use-cases of seq2seq models

- The task of language translation requires converting an input sentence $\boldsymbol{x}$ from a source language to an output sentence $\boldsymbol{y}$ in the target language preserving the semantic information.

I play the flute. ⟹ میں بانسری بجاتا ہوں۔

I play cricket. ⟹ میں کرکٹ کھیلتا ہوں۔

Language translation is one of the prime use-cases of seq2seq models

- The task of language translation requires converting an input sentence $\boldsymbol{x}$ from a source language to an output sentence $\boldsymbol{y}$ in the target language preserving the semantic information.

I play the flute. ⟹ میں بانسری بجاتا ہوں۔

I play cricket. ⟹ میں کرکٹ کھیلتا ہوں۔

- If the semantic information is not preserved, the objective of language translation is not achieved.

Language translation is one of the prime use-cases of seq2seq models

- The task of language translation requires converting an input sentence $\boldsymbol{x}$ from a source language to an output sentence $\boldsymbol{y}$ in the target language preserving the semantic information.

I play the flute.	→	میں بانسری بجاتا ہوں-
I play cricket.	→	میں کرکٹ کھیلتا ہوں-

- If the semantic information is not preserved, the objective of language translation is not achieved.

The spirit is willing but the flesh is weak.	→	The liquor is good but the meat is spoiled.
Out of sight, out of mind.	→	Invisible, idiot.

The need for machine translation emerged during the cold war

- The early machine translation models were developed to eavesdrop on Russians around 1950.
  - Rule-based models relied heavily on bilingual dictionaries.
  - Limitations of such models?

The need for machine translation emerged during the cold war

- The early machine translation models were developed to eavesdrop on Russians around 1950.
  - Rule-based models relied heavily on bilingual dictionaries.
  - Limitations of such models? Different languages have different sentence structures.

The need for machine translation emerged during the cold war

- The early machine translation models were developed to eavesdrop on Russians around 1950.
  - Rule-based models relied heavily on bilingual dictionaries.
  - Limitations of such models? Different languages have different sentence structures.
- A better approach is to learn probabilistic models from the data.
  - Learn how?

The need for machine translation emerged during the cold war

- The early machine translation models were developed to eavesdrop on Russians around 1950.
  - Rule-based models relied heavily on bilingual dictionaries.
  - Limitations of such models? Different languages have different sentence structures.
- A better approach is to learn probabilistic models from the data.
  - Learn how?
    1. Statistical Methods
    2. Neural Network based Methods

- Suppose we are translating from English to German.

Statistical Machine Translation (SMT) learns probabilistic language models

- Suppose we are translating from English to German.
- The task of SMT can be formulated as,
  - Given an English sentence e, find a German sentence g, which best captures the semantics of e.

$$\tilde{g} = argmax_g \; P(g|e)$$

  - This model should have knowledge of both languages to perform well.

Statistical Machine Translation (SMT) learns probabilistic language models

- Suppose we are translating from English to German.

- The task of SMT can be formulated as,

  - Given an English sentence e, find a German sentence g, which best captures the semantics of e.

$$\tilde{g} = argmax_g \, P(g|e)$$

  - This model should have knowledge of both languages to perform well.

- Using Bayes' rule, the probability may be divided into following components

$$\tilde{g} = argmax_g \frac{P(e|g)P(g)}{P(e)}$$

Statistical Machine Translation (SMT) learns probabilistic language models

- Suppose we are translating from English to German.
- The task of SMT can be formulated as,
  - Given an English sentence e, find a German sentence g, which best captures the semantics of e.

$$\tilde{g} = argmax_g \, P(g|e)$$

  - This model should have knowledge of both languages to perform well.
- Using Bayes' rule, the probability may be divided into following components

$$\tilde{g} = argmax_g \frac{P(e|g)P(g)}{P(e)}$$

  - However, $P(e)$ does not depend on the German sentence, it can be considered as a constant.

$$\tilde{g} = argmax_g \, P(e|g)P(g)$$

The task of machine translation can be divided into two sub-tasks

- The probabilistic model for language translation takes the form,

$$\tilde{g} = argmax_g \, P(e|g)P(g)$$

The task of machine translation can be divided into two sub-tasks

- The probabilistic model for language translation takes the form,

$$\tilde{g} = argmax_g \, P(e|g)P(g)$$

  - $P(e|g)$: A translation model.
    - Ensures fidelity/adequacy of translation.
    - Requires bilingual data (parallel corpus).
    - Learns the mapping between two languages at words or small phrase level.

The task of machine translation can be divided into two sub-tasks

- The probabilistic model for language translation takes the form,

$$\tilde{g} = argmax_g \, P(e|g)P(g)$$

  - $P(e|g)$: A translation model.
    - Ensures fidelity/adequacy of translation.
    - Requires bilingual data (parallel corpus).
    - Learns the mapping between two languages at words or small phrase level.
  - $P(g)$: A German language model.
    - Ensures fluency of German sentences.
    - Checks if the output sentence conforms to German grammar and sentence structure.

The task of machine translation can be divided into two sub-tasks

- The probabilistic model for language translation takes the form,

$$\tilde{g} = argmax_g \ P(e|g)P(g)$$

  - $P(e|g)$: A translation model.
    - Ensures fidelity/adequacy of translation.
    - Requires bilingual data (parallel corpus).
    - Learns the mapping between two languages at words or small phrase level.
  - $P(g)$: A German language model.
    - Ensures fluency of German sentences.
    - Checks if the output sentence conforms to German grammar and sentence structure.
  - $argmax_g$: Finds the sentence g that maximises this probability.

How to learn translation model?

- The model $P(e|g)$ can be further divided into two components.

$$P(e|g) = P(e,a|g)$$

  - Here, a represent alignment (correspondence) between German and English words.

How to learn translation model?

- The model $P(e|g)$ can be further divided into two components.

$$P(e|g) = P(e, a|g)$$

  - Here, a represent alignment (correspondence) between German and English words.

Japan shaken by two new quakes

Japan von zwei neuen Beben erschüttert

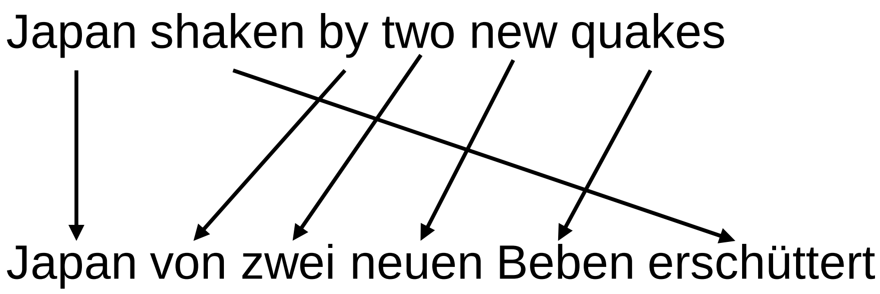
One to One Alignment

How to learn translation model?

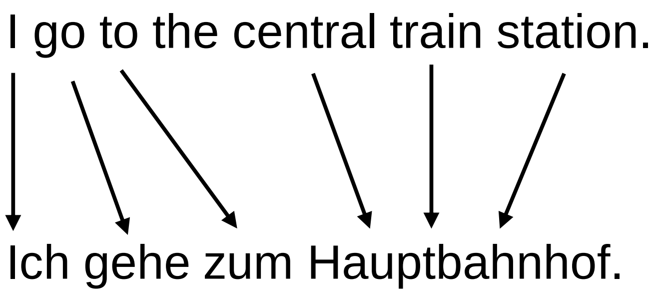
- The model $P(e|g)$ can be further divided into two components.

$$P(e|g) = P(e,a|g)$$

  - Here, a represent alignment (correspondence) between German and English words.



One to One Alignment



Many to One Alignment

Alignment is complex

- If a word corresponds to more than one words in other language, it’s called a fertile word.

Er hat Backpfeifengesicht

He has a face in need of a fist.

Alignment is complex

- If a word corresponds to more than one words in other language, it’s called a fertile word.

Er hat Backpfeifengesicht

He has a face in need of a fist.

- There can also be many to many alignment at phrase level.

Er geht ja nicht nach hause

He is not going home

The SMT was very popular between 1990 – 2010

- The best SMT models were extremely complex.
  - Required separately designed subcomponents.
  - Heavily involved feature engineering.

The SMT was very popular between 1990 – 2010

- The best SMT models were extremely complex.
  - Required separately designed subcomponents.
  - Heavily involved feature engineering.
- Each language required different features to be captured.
  - Many extra resources needed to be compiled and maintained.
  - For instance, tables of equivalent phrases.

The SMT was very popular between 1990 – 2010

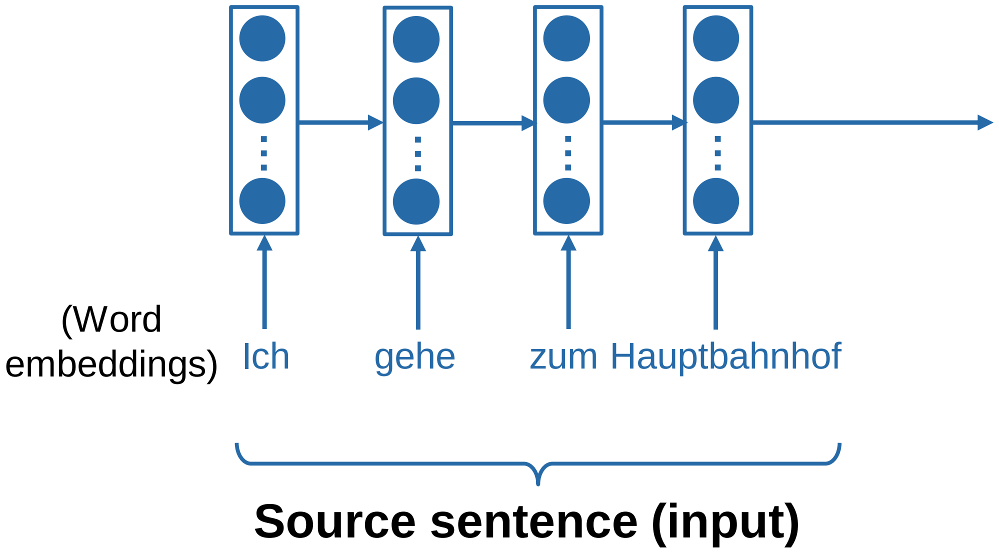
- The best SMT models were extremely complex.
  - Required separately designed subcomponents.
  - Heavily involved feature engineering.
- Each language required different features to be captured.
  - Many extra resources needed to be compiled and maintained.
  - For instance, tables of equivalent phrases.
- Depended on manual labour which was not reusable.
  - For every language pair, the whole process needs to be repeated.

Neural Machine Translation (NMT) made its entry in 2014

Neural Machine Translation (NMT) made its entry in 2014

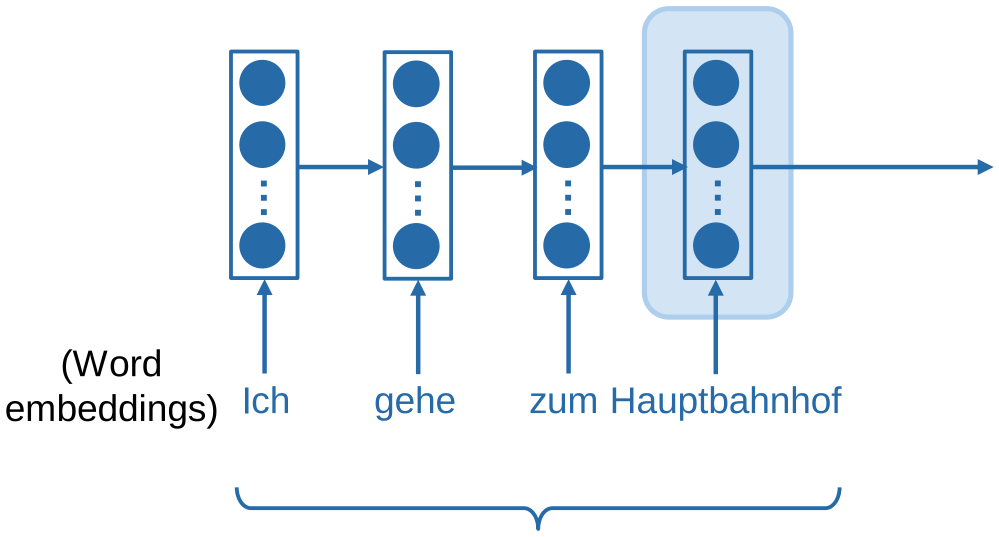
$\underbrace{\text{Ich} \quad \text{gehe} \quad \text{zum Hauptbahnhof}}_{\textbf{Source sentence (input)}}$

Neural Machine Translation (NMT) made its entry in 2014



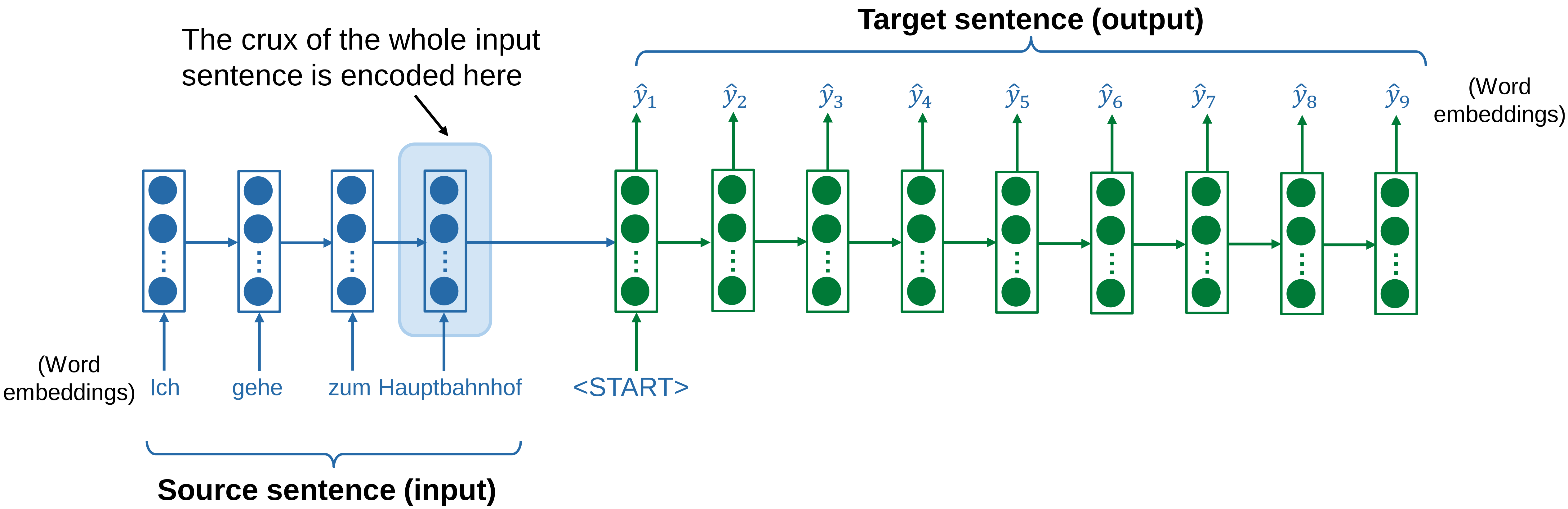
Neural Machine Translation (NMT) made its entry in 2014

The crux of the whole input sentence is encoded here



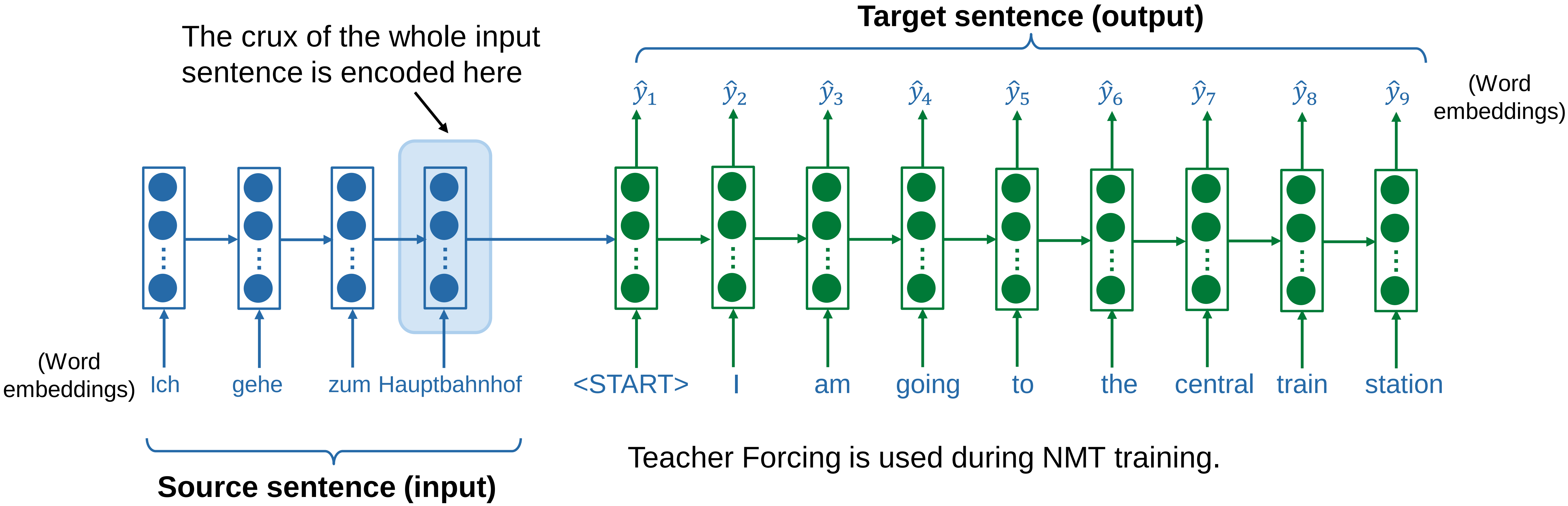
Source sentence (input)

Neural Machine Translation (NMT) made its entry in 2014



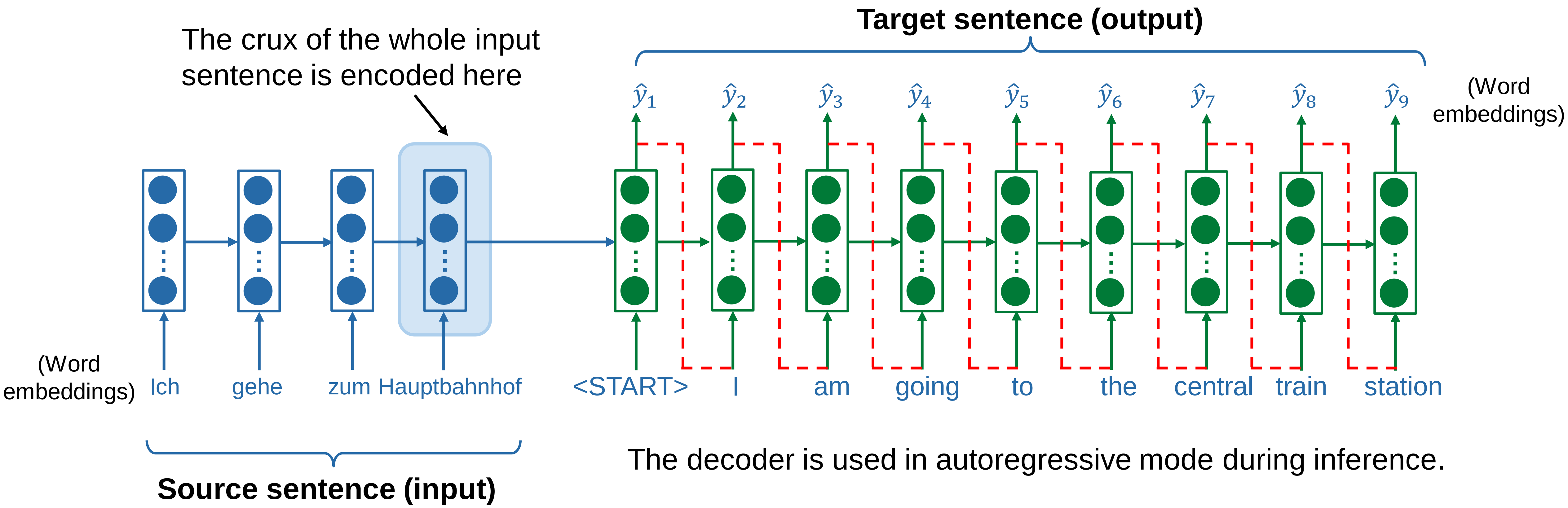
NUST Defining futures
School of Electrical Engineering & Computer Science

Neural Machine Translation (NMT) made its entry in 2014



Teacher Forcing is used during NMT training.

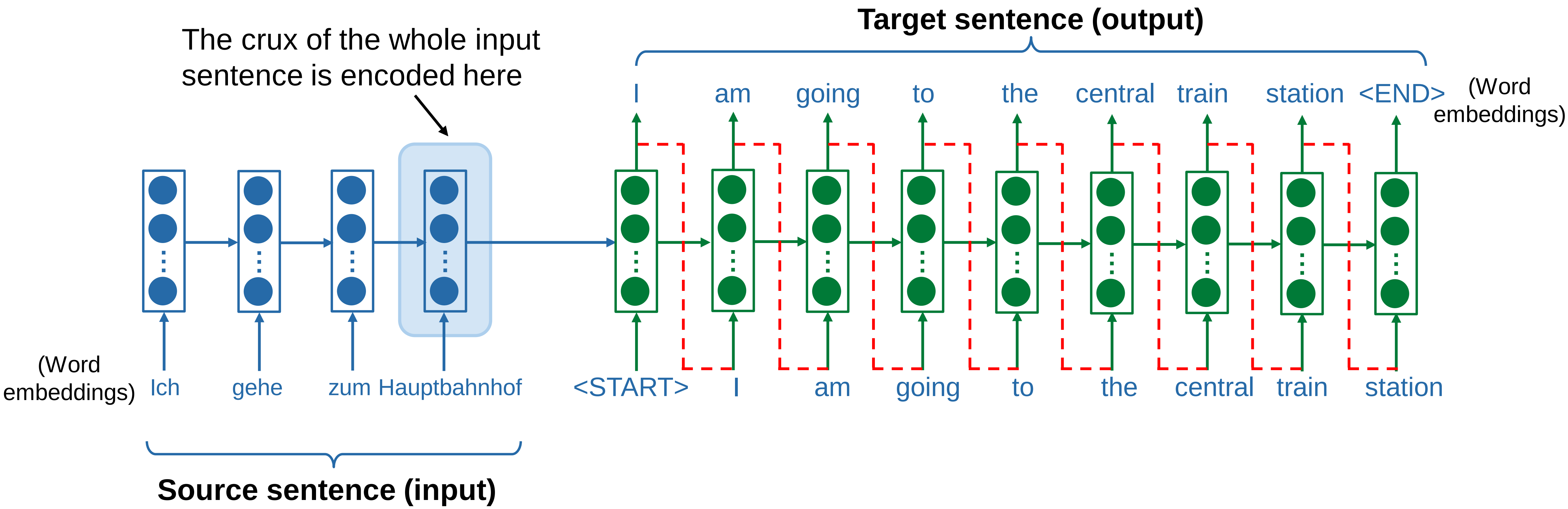
Neural Machine Translation (NMT) made its entry in 2014



The decoder is used in autoregressive mode during inference.

Neural Machine Translation (NMT) made its entry in 2014

- Decoder generates target language sentence conditioned on the encoding of the source sentence.



seq2seq models is a type of conditional language model

- Language Model generates coherent and grammatically correct sequence of words.
  - Predicts next word in the target sentence.

seq2seq models is a type of conditional language model

- Language Model generates coherent and grammatically correct sequence of words.
  - Predicts next word in the target sentence.
- Conditional Language Models generate output considering a given condition.
  - The next word in the target sequence is conditioned on the encoding of source sentence and the previously predicted word in the target sentence.

seq2seq models is a type of conditional language model

- Language Model generates coherent and grammatically correct sequence of words.
  - Predicts next word in the target sentence.
- Conditional Language Models generate output considering a given condition.
  - The next word in the target sequence is conditioned on the encoding of source sentence and the previously predicted word in the target sentence.
- Mathematically, the task of NMT is to predict;

$$P(g|e) = P(g_1|e)P(g_2|g_1, e)P(g_3|g_1, g_2, e) \dots P(g_T|g_1, g_2, \dots g_{T-1}, e)$$

Here g represents German sentence (target) and e stands for English sentence (source).

seq2seq models is a type of conditional language model

- Language Model generates coherent and grammatically correct sequence of words.
    - Predicts next word in the target sentence.
- Conditional Language Models generate output considering a given condition.
    - The next word in the target sequence is conditioned on the encoding of source sentence and the previously predicted word in the target sentence.
- Mathematically, the task of NMT is to predict;

$$P(g|e) = P(g_1|e)P(g_2|g_1, e)P(g_3|g_1, g_2, e) \dots P(g_T|g_1, g_2, \dots g_{T-1}, e)$$

Here g represents German sentence (target) and e stands for English sentence (source).

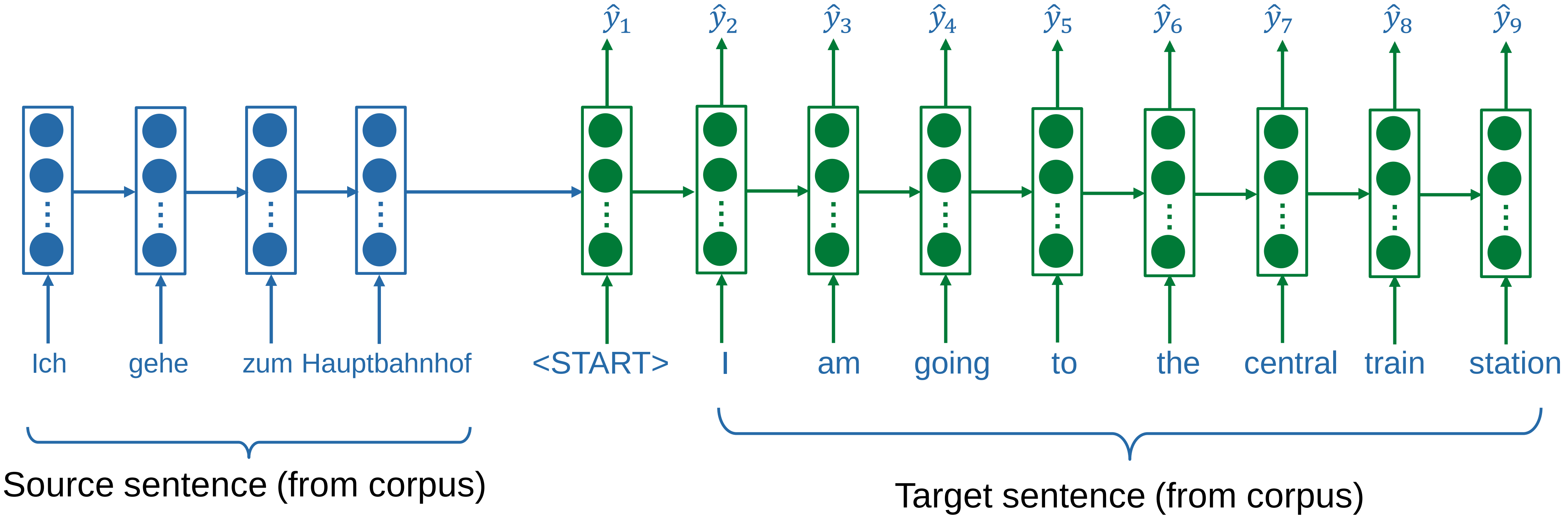
- No need to break down $P(g|e)$ into smaller components as in SMT.

How to train your NMT model?

- Get a parallel corpus.

How to train your NMT model?

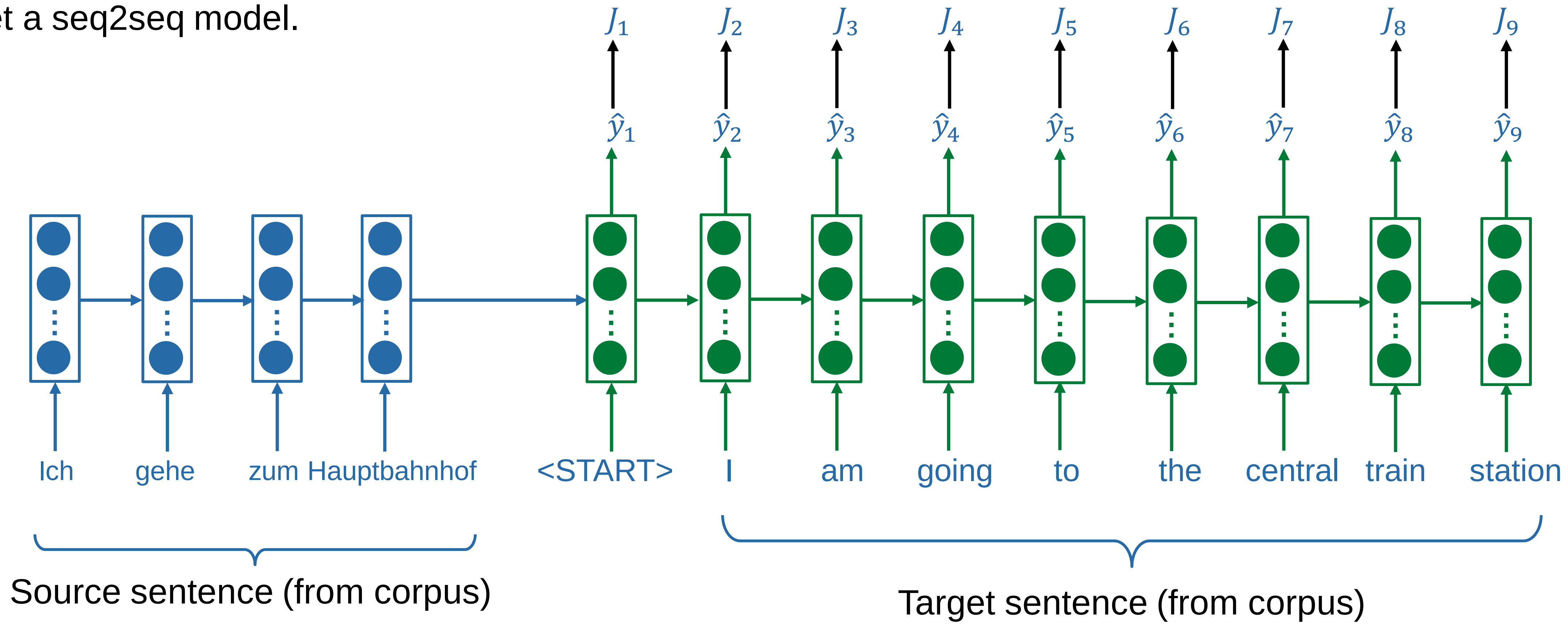
- Get a parallel corpus.
- Get a seq2seq model.



NUST Defining futures
School of Electrical Engineering & Computer Science

How to train your NMT model?

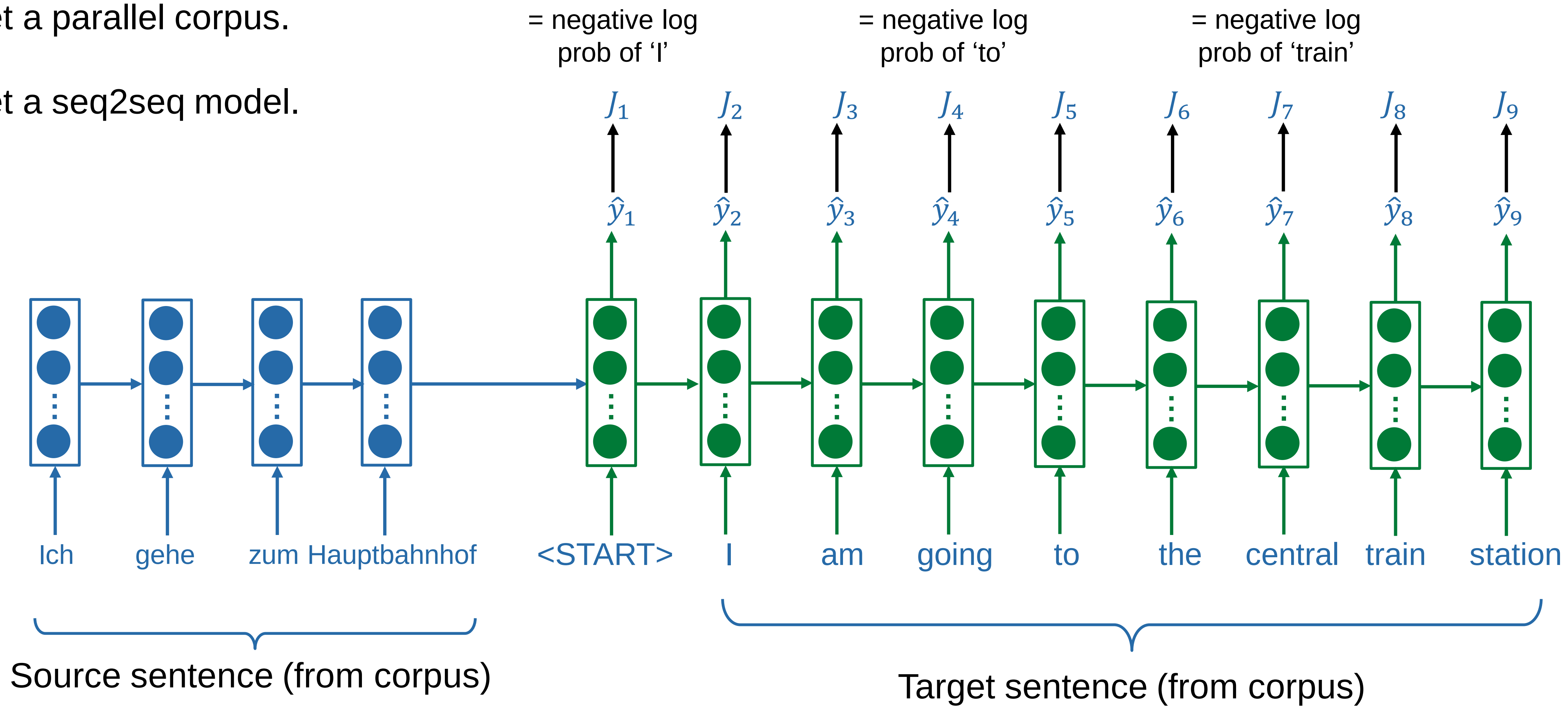
- Get a parallel corpus.
- Get a seq2seq model.



How to train your NMT model?

- Get a parallel corpus.

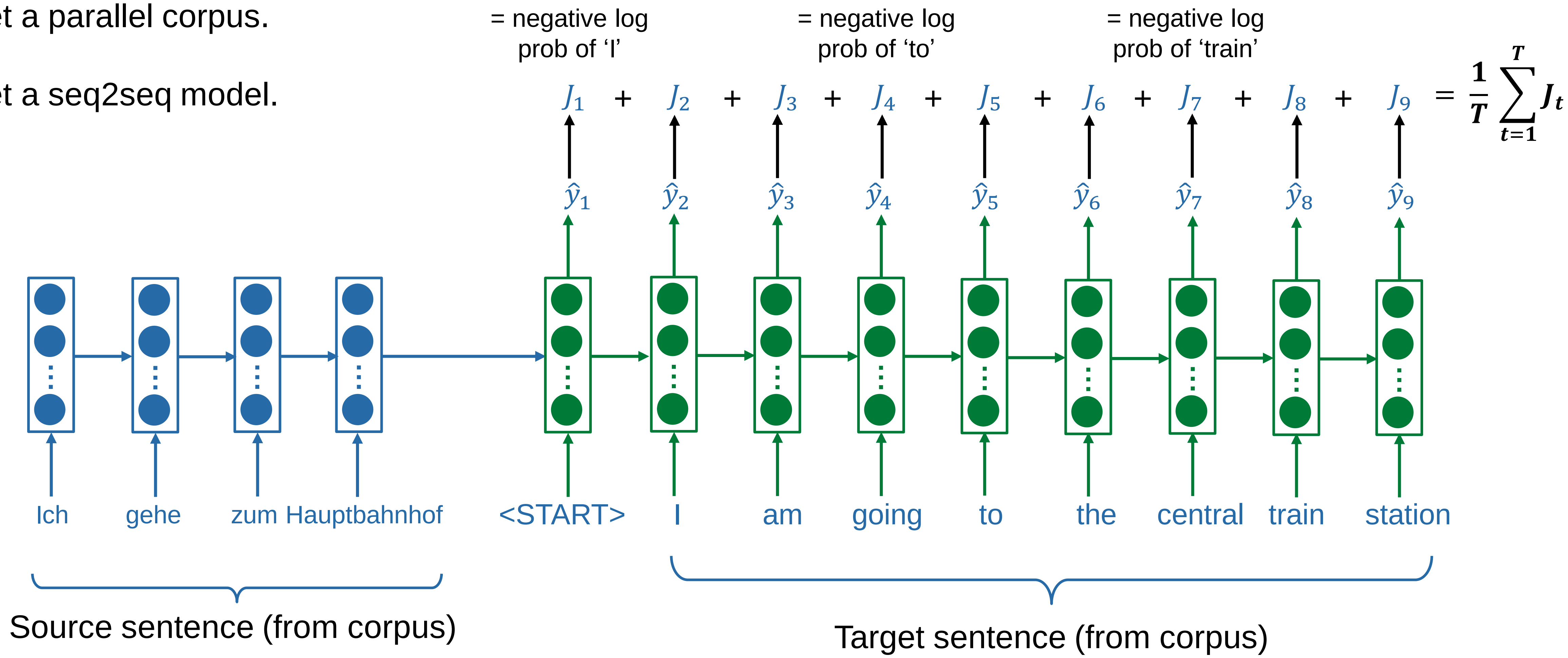
- Get a seq2seq model.



How to train your NMT model?

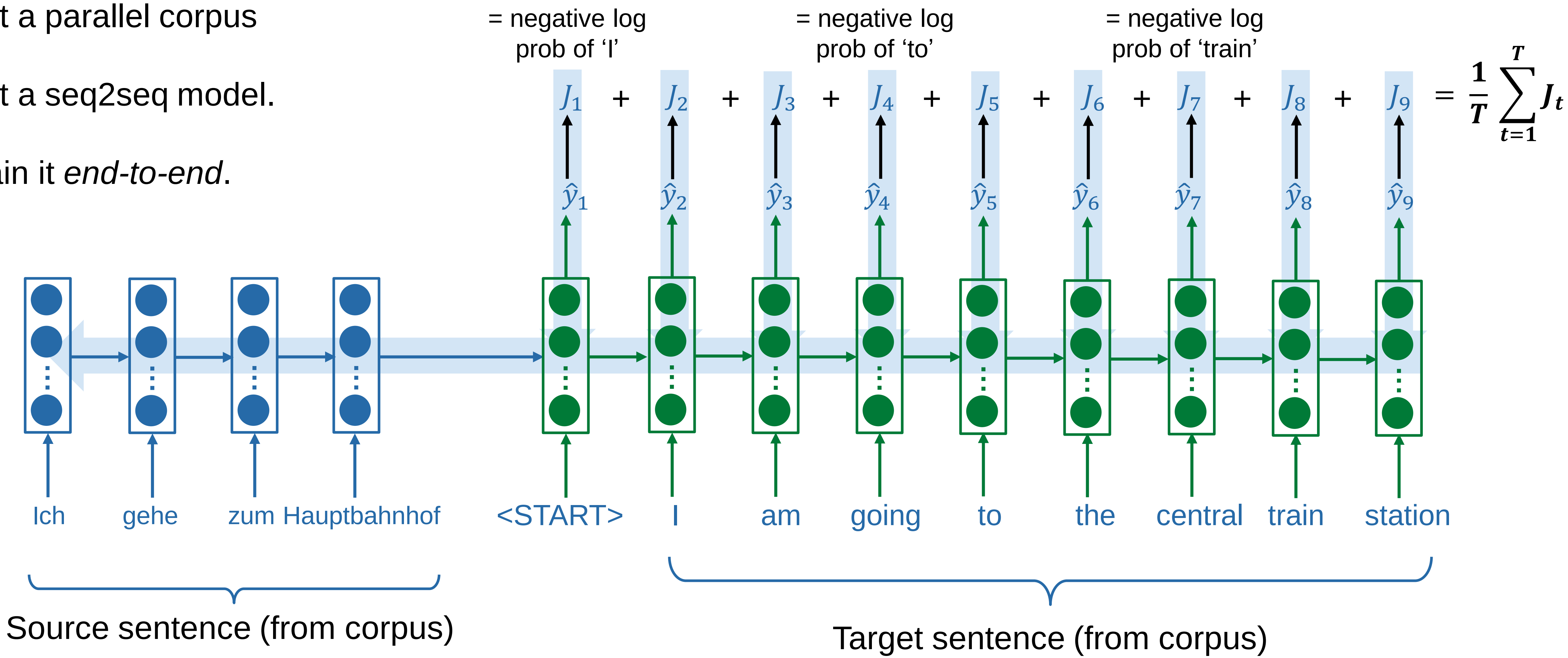
- Get a parallel corpus.

- Get a seq2seq model.



How to train your NMT model?

- Get a parallel corpus
- Get a seq2seq model.
- Train it *end-to-end*.



Greedy decoding picks the next word based on the highest probability

- The use of $argmax$ ensures that only the words with the highest probability are chosen as expected output $\hat{y}_t$ at each time step.

Greedy decoding picks the next word based on the highest probability

- The use of $argmax$ ensures that only the words with the highest probability are chosen as expected output $\hat{y}_t$ at each time step.

- This greedy approach is not always desirable. Why?

Greedy decoding picks the next word based on the highest probability

- The use of $argmax$ ensures that only the words with the highest probability are chosen as expected output $\hat{y}_t$ at each time step.

- This greedy approach is not always desirable. Why?

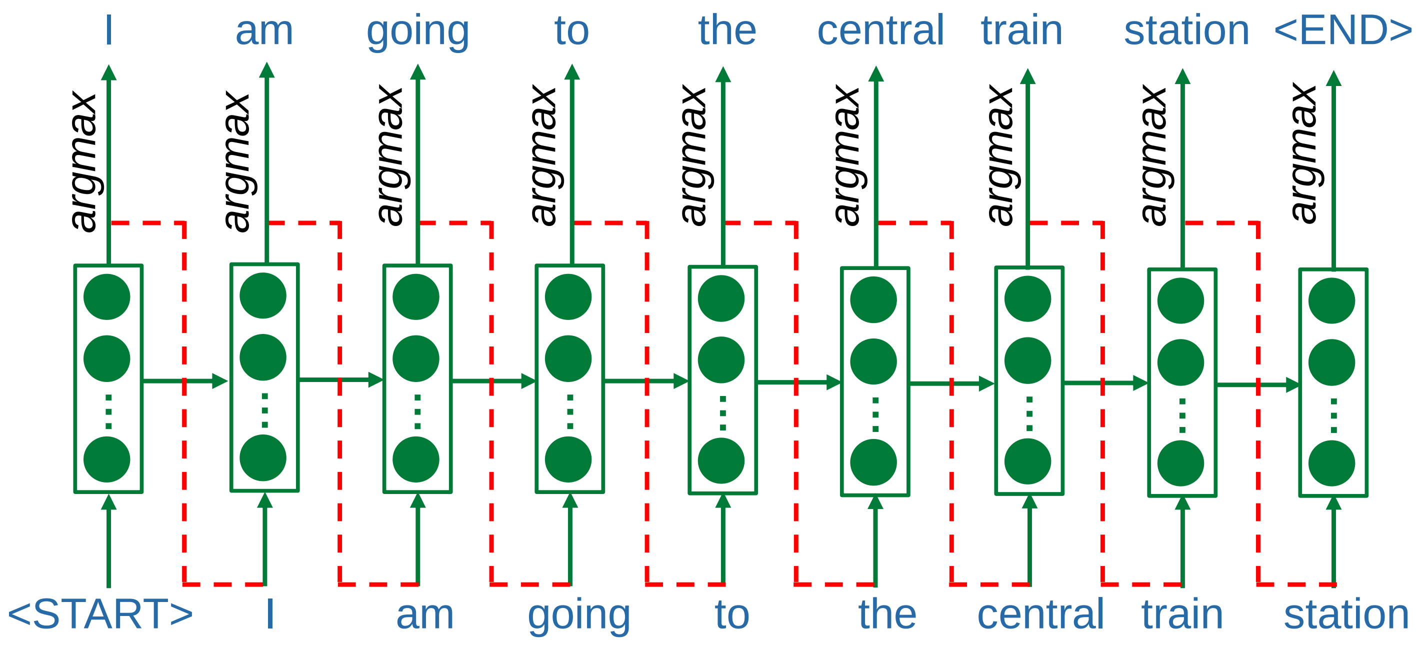
$$P(g|e) = P(g_1|e)P(g_2|g_1, e)P(g_3|g_1, g_2, e) \dots P(g_T|g_1, g_2, \dots g_{T-1}, e)$$

Greedy decoding picks the next word based on the highest probability

- The use of $argmax$ ensures that only the words with the highest probability are chosen as expected output $\hat{y}_t$ at each time step.

- This greedy approach is not always desirable. Why?

$$P(g|e) = P(g_1|e)P(g_2|g_1, e)P(g_3|g_1, g_2, e) \ldots P(g_T|g_1, g_2, \ldots g_{T-1}, e)$$



Greedy decoding has some serious problems

- We cannot undo a choice that is made at a previous time step.

Greedy decoding has some serious problems

- We cannot undo a choice that is made at a previous time step.
- Suppose the input is ***Ich gehe zum Hauptbahnhof*** (***I am going to the central train station***).

Greedy decoding has some serious problems

- We cannot undo a choice that is made at a previous time step.

- Suppose the input is ***Ich gehe zum Hauptbahnhof*** (***I am going to the central train station***).

    I ___

Greedy decoding has some serious problems

- We cannot undo a choice that is made at a previous time step.

- Suppose the input is ***Ich gehe zum Hauptbahnhof*** (***I am going to the central train station***).

I ___ I am ___

Greedy decoding has some serious problems

- We cannot undo a choice that is made at a previous time step.

- Suppose the input is ***Ich gehe zum Hauptbahnhof*** (***I am going to the central train station***).

I ___ I am ___ I am going ___

NUST Defining futures
School of Electrical Engineering
& Computer Science

Greedy decoding has some serious problems

- We cannot undo a choice that is made at a previous time step.

- Suppose the input is ***Ich gehe zum Hauptbahnhof*** (***I am going to the central train station***).

I ___ I am ___ I am going ___ I am going back (made a mistake here)

Greedy decoding has some serious problems

- We cannot undo a choice that is made at a previous time step.

- Suppose the input is ***Ich gehe zum Hauptbahnhof*** (***I am going to the central train station***).

  I ___ I am ___ I am going ___ I am going back (made a mistake here)

- How can we fix this?

Greedy decoding has some serious problems

- We cannot undo a choice that is made at a previous time step.
- Suppose the input is ***Ich gehe zum Hauptbahnhof*** (***I am going to the central train station***).

  I ___ I am ___ I am going ___ I am going back (made a mistake here)

- How can we fix this?
  - Exhaustive Search? Compute all possible sequences.

Greedy decoding has some serious problems

- We cannot undo a choice that is made at a previous time step.
- Suppose the input is ***Ich gehe zum Hauptbahnhof*** (***I am going to the central train station***).

  I ___ I am ___ I am going ___ I am going back (made a mistake here)

- How can we fix this?
  - Exhaustive Search? Compute all possible sequences.
    - At each time step t of decoder, V^t partial translations are tracked, where V is vocabulary size.
    - This results in $O\ (V^T)$ complexity which is far too expensive.

Beam search decoding provides inexpensive way to find suitable translation

- At each time step, keep track of k most likely partial translations (hypotheses).
  - k is the beam size, a hyperparameter whose value may be determined heuristically.
  - k determines the size of the search space.

Beam search decoding provides inexpensive way to find suitable translation

- At each time step, keep track of k most likely partial translations (hypotheses).
  - k is the beam size, a hyperparameter whose value may be determined heuristically.
  - k determines the size of the search space.
- Formally, a hypothesis is a sequence of predicted words, $\hat{y}^{<1>}, \hat{y}^{<2>}, \dots, \hat{y}^{<t>}$.
  - Each hypothesis has a suitability score defined as

$$score(\hat{y}^{<1>}, \hat{y}^{<2>}, \dots, \hat{y}^{<t>}) = \sum_{i=0}^{t} \log P_{LM}\,(\hat{y}^{i} | \hat{y}^{<1>}, \hat{y}^{<2>}, \dots, \hat{y}^{<i-1>}, x)$$

Beam search decoding provides inexpensive way to find suitable translation

- At each time step, keep track of k most likely partial translations (hypotheses).
  - k is the beam size, a hyperparameter whose value may be determined heuristically.
  - k determines the size of the search space.
- Formally, a hypothesis is a sequence of predicted words, $\hat{y}^{<1>}, \hat{y}^{<2>}, \dots, \hat{y}^{<t>}$.
  - Each hypothesis has a suitability score defined as

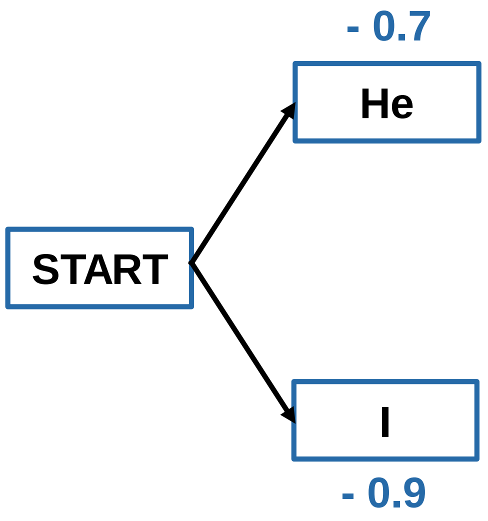
$$score(\hat{y}^{<1>}, \hat{y}^{<2>}, \dots, \hat{y}^{<t>}) = \sum_{i=0}^{t} \log P_{LM}\,(\hat{y}^{i} | \hat{y}^{<1>}, \hat{y}^{<2>}, \dots, \hat{y}^{<i-1>}, x)$$

- **Note:** Scores of all hypotheses will be negative. High scores means more suitable hypothesis.
- **Caution:** Beam search is not guaranteed to find the optimal solution. But it’s efficient.

Let’s take an example of beam search

- For $k = 2$, and an example target sentence “***He hit me with a stick***”.

$score(He) = log\ P_{LM}(He| < START >)$



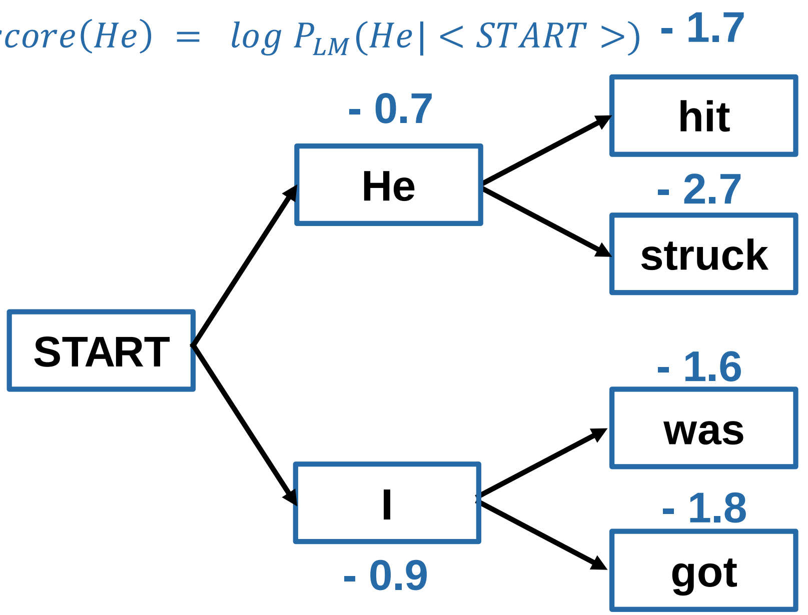
$score(I) = log\ P_{LM}(I| < START >)$

Let's take an example of beam search

- For $k = 2$, and an example target sentence "***He hit me with a stick***".

$$score(He\ hit) = log\ P_{LM}(He| < START >) + log\ P_{LM}(hit| < START > He)$$

$$score(He) = log\ P_{LM}(He| < START >)$$

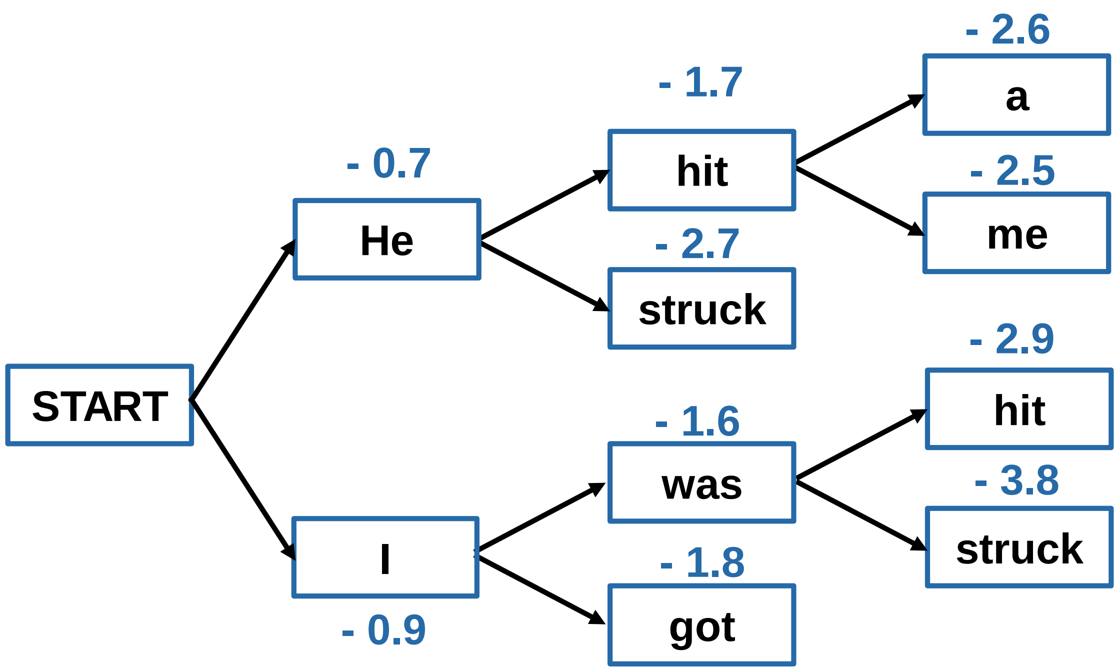


$$score(I) = log\ P_{LM}(I| < START >)$$

$$score(I\ got) = log\ P_{LM}(I| < START >) + log\ P_{LM}(got| < START > I)$$

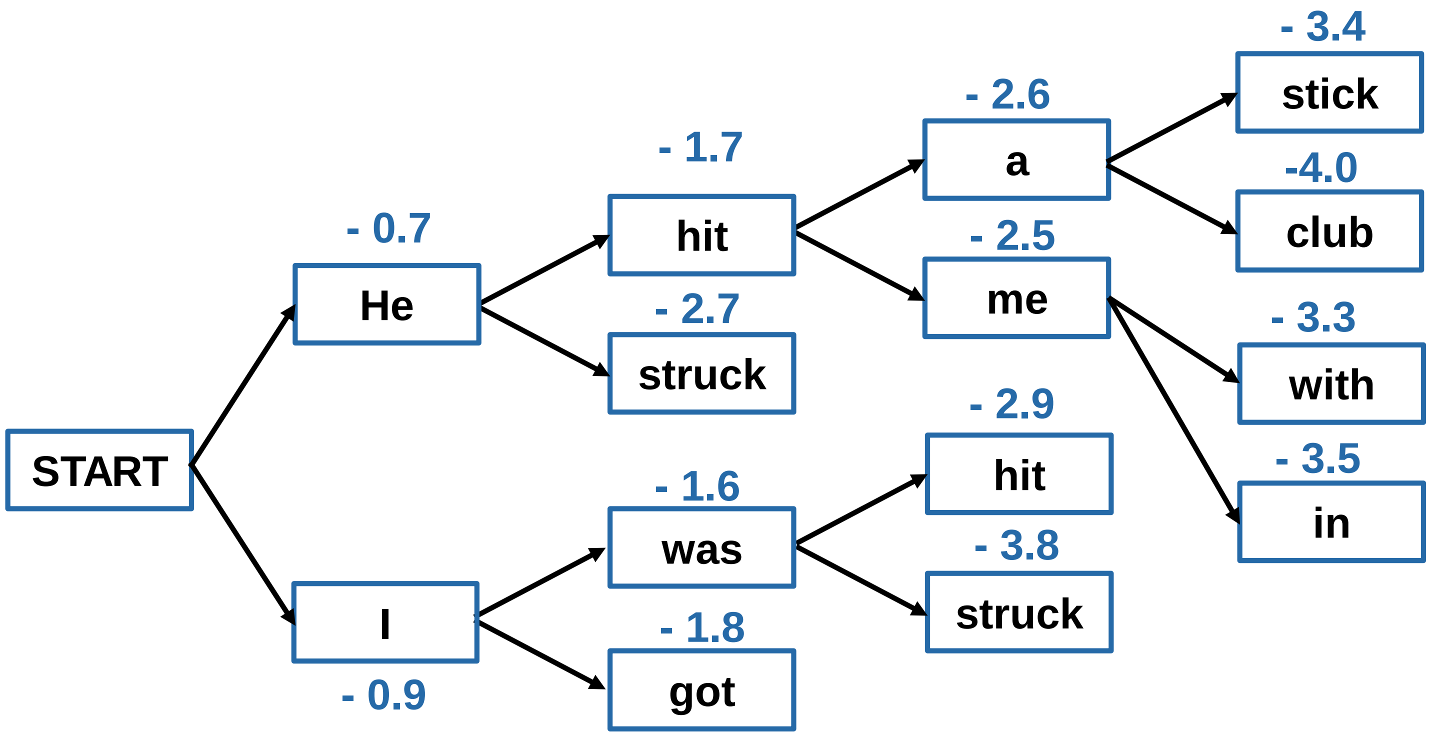
Let's take an example of beam search

- For $k = 2$, and an example target sentence "***He hit me with a stick***".



Let's take an example of beam search

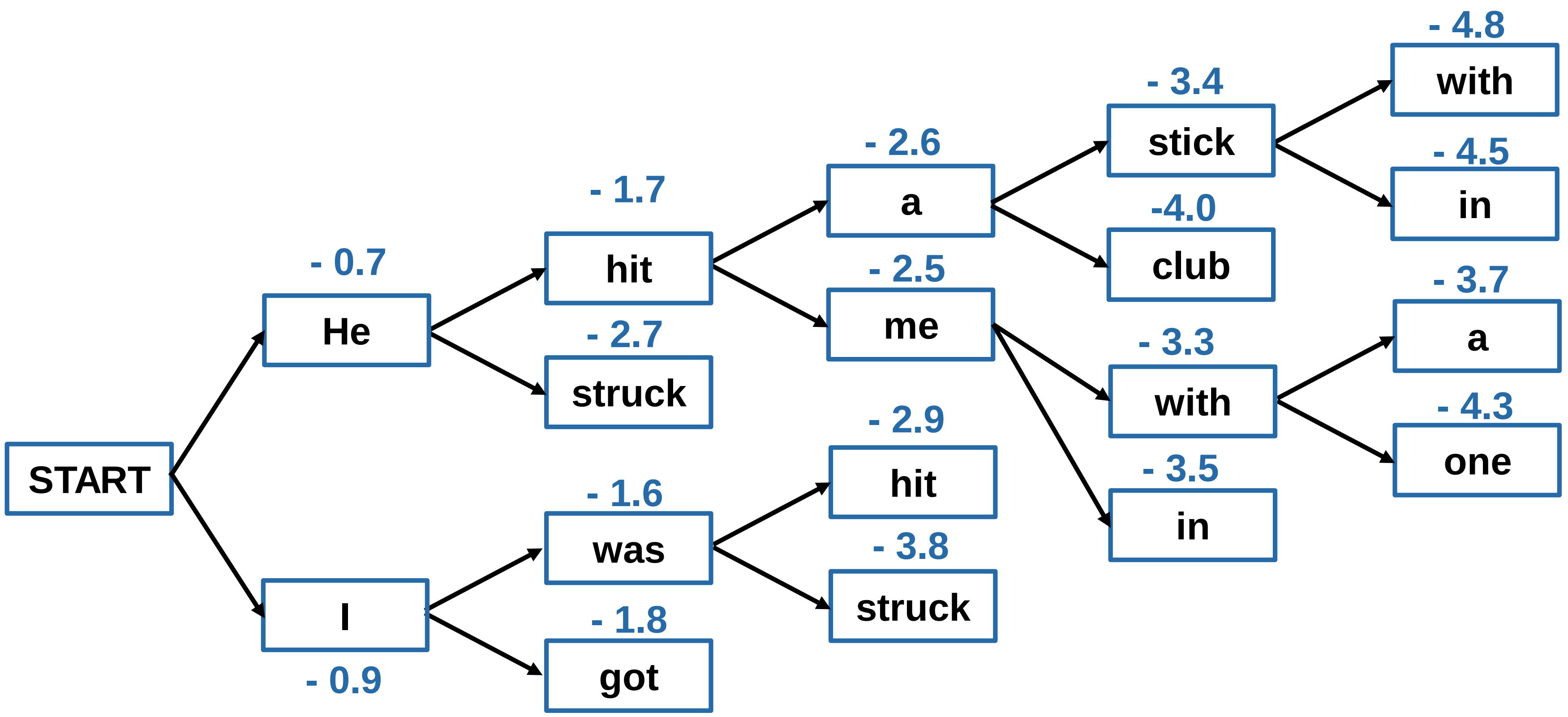
- For $k = 2$, and an example target sentence "***He hit me with a stick***".



NUST
Defining futures
School of Electrical Engineering
& Computer Science

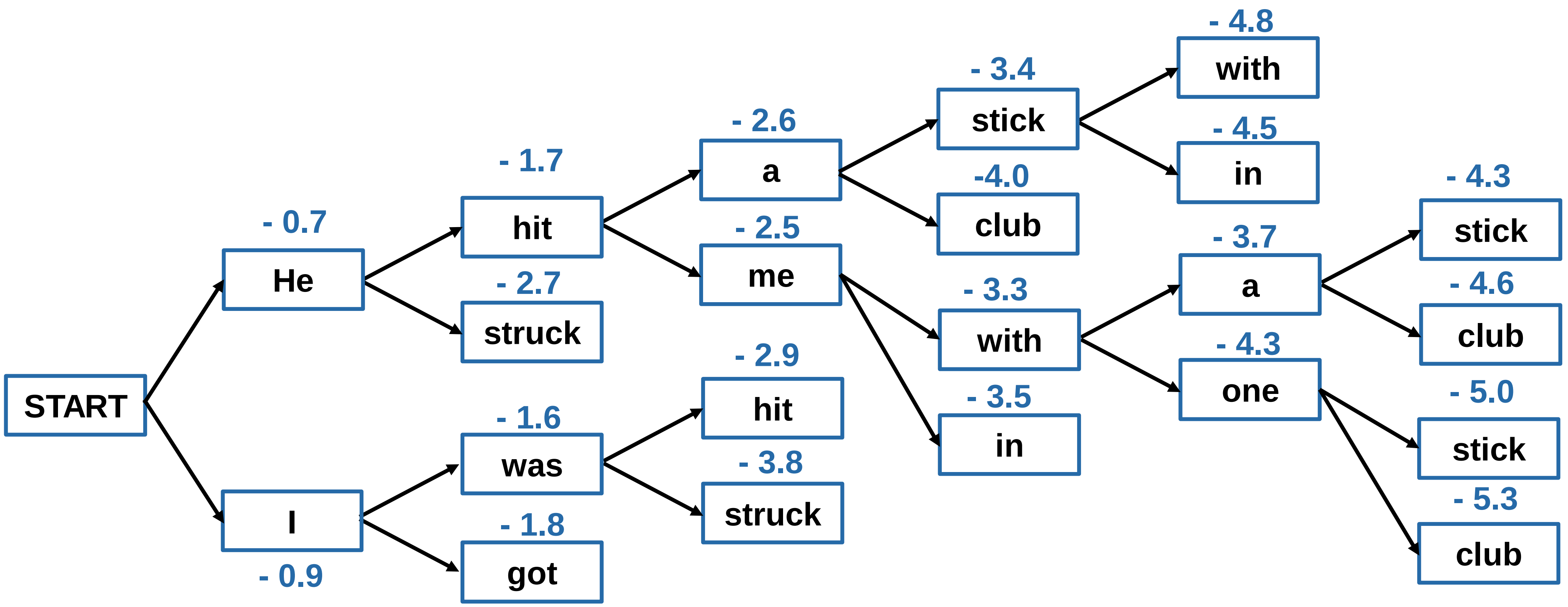
Let’s take an example of beam search

- For $k = 2$, and an example target sentence “***He hit me with a stick***”.



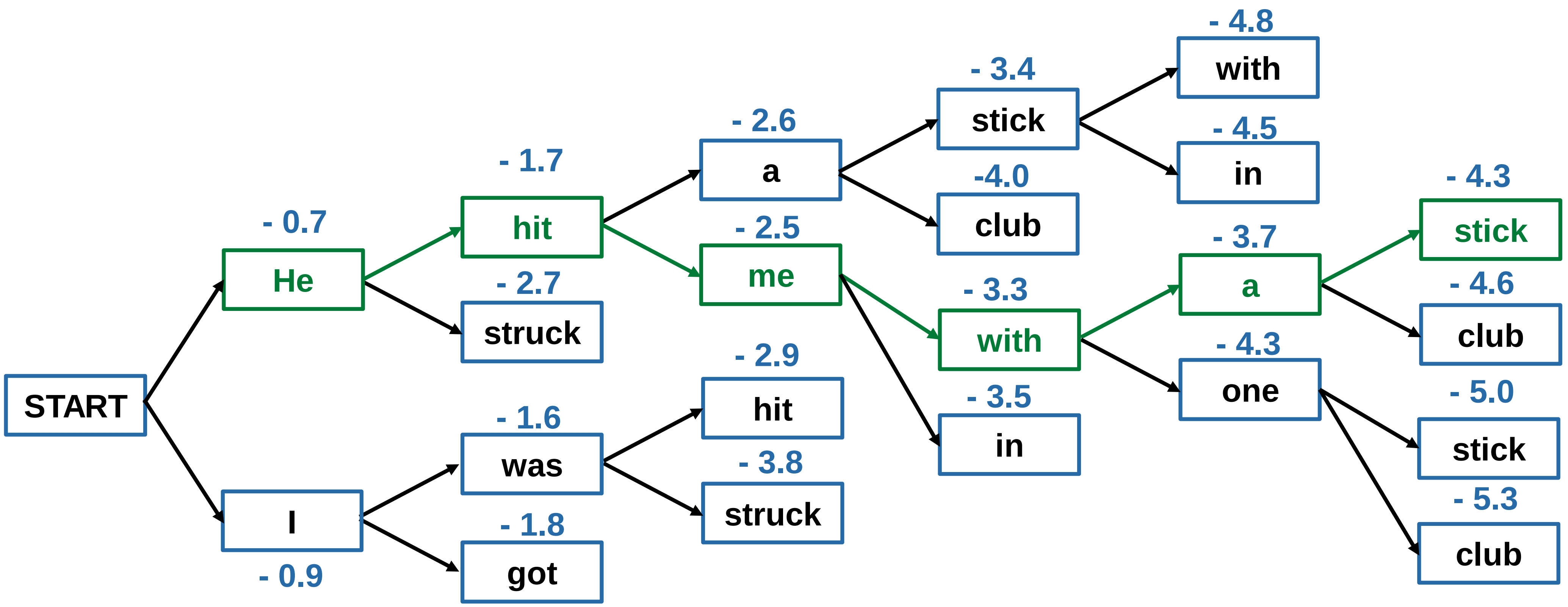
Let's take an example of beam search

- For $k = 2$, and an example target sentence "***He hit me with a stick***".



Let's take an example of beam search

- For $k = 2$, and an example target sentence "***He hit me with a stick***".



When to stop beam search and how to pick the most suitable hypothesis?

- In greedy search, target sentence is ended when $< END >$ token is generated.

- In beam search, different hypotheses may generate $< END >$ token at different time steps.

When to stop beam search and how to pick the most suitable hypothesis?

- In greedy search, target sentence is ended when $< END >$ token is generated.
- In beam search, different hypotheses may generate $< END >$ token at different time steps.
  - Set aside a completed hypothesis that has generated $< END >$ token and continue exploring others.
  - Beam search is stopped when either
    - T (predefined) time steps have arrived.
    - N (predefined) number of hypotheses have been completed.

When to stop beam search and how to pick the most suitable hypothesis?

- In greedy search, target sentence is ended when $< END >$ token is generated.
- In beam search, different hypotheses may generate $< END >$ token at different time steps.
  - Set aside a completed hypothesis that has generated $< END >$ token and continue exploring others.
  - Beam search is stopped when either
    - T (predefined) time steps have arrived.
    - N (predefined) number of hypotheses have been completed.
- Absolute hypotheses scores can be deceiving.

$$score(\hat{y}^{<1>}, \hat{y}^{<2>}, \dots, \hat{y}^{<t>}) = \sum_{i=0}^{t} \log P_{LM}\,(\hat{y}^{i} | \hat{y}^{<1>}, \hat{y}^{<2>}, \dots, \hat{y}^{<i-1>}, x)$$

  - Normalised scores are better estimates of suitability.

The NMT has some merits and demerits

- Provides better performance.
  - More fluent translation
  - Better use of context
- Single end-to-end system can be efficiently and conveniently optimised.
  - No subcomponents to optimise individually.

The NMT has some merits and demerits

- Provides better performance.
  - More fluent translation
  - Better use of context
- Single end-to-end system can be efficiently and conveniently optimised.
  - No subcomponents to optimise individually.
- Requires less human effort.
  - No feature engineering needed.
- Reusable
  - Same model different language pairs.
  - Requires bilingual data of course.

The NMT has some merits and demerits

- Provides better performance.
  - More fluent translation
  - Better use of context
- Single end-to-end system can be efficiently and conveniently optimised.
  - No subcomponents to optimise individually.
- Requires less human effort.
  - No feature engineering needed.
- Reusable
  - Same model different language pairs.
  - Requires bilingual data of course.

- Less interpretable
  - Difficult to track errors
- Hard to exert control
  - Cannot specify rules
- Safety concerns
  - Model can say whatever it wants.

MT models are evaluated using BLEU metric

- It compares machine translation with one or more human translations and computes a similarity score based on n-gram precision and brevity penalty.
    - Checks how many n-grams generated by MT are actually present in human translation.

Papineni, Kishore, et al. "BLEU: a method for automatic evaluation of machine translation." Proceedings of the 40th annual meeting of the Association for Computational Linguistics. 2002.

MT models are evaluated using BLEU metric

- It compares machine translation with one or more human translations and computes a similarity score based on n-gram precision and brevity penalty.
  - Checks how many n-grams generated by MT are actually present in human translation.
  - Also evaluates if MT is significantly shorter than human translation. If c is length of candidate translation and r is the length of reference translation.

$$Bravety\ Panelty = \begin{cases} 1, & if\ c > r \\ e^{\left(1-\frac{r}{c}\right)}, & if\ c \leq r \end{cases}$$

Papineni, Kishore, et al. "BLEU: a method for automatic evaluation of machine translation." Proceedings of the 40th annual meeting of the Association for Computational Linguistics. 2002.

MT models are evaluated using BLEU metric

- It compares machine translation with one or more human translations and computes a similarity score based on n-gram precision and brevity penalty.

  - Checks how many n-grams generated by MT are actually present in human translation.

  - Also evaluates if MT is significantly shorter than human translation. If c is length of candidate translation and r is the length of reference translation.

$$Bravety\ Panelty = \begin{cases} 1, & if\ c > r \\ e^{\left(1-\frac{r}{c}\right)}, & if\ c \leq r \end{cases}$$

- BLEU is useful but imperfect.

  - There can be many valid translations. It does not consider semantic similarity between words, or inclusion of all reference information in candidate.

Papineni, Kishore, et al. "BLEU: a method for automatic evaluation of machine translation." Proceedings of the 40th annual meeting of the Association for Computational Linguistics. 2002.

METEOR is another automatic evaluation metric for machine translation

- METEOR (Metric for Evaluation of Translation with Explicit ORdering) combines n-gram precision and recall.

$$MeteorScore = \frac{Prec \times Recall}{\alpha . Prec + (1-\alpha)\, Recall}\left(1 - \gamma\left(\frac{chunks}{u_m}\right)^{\beta}\right)$$

$chunks$ = bigram/trigram matches, u_m = unigrams in candidate
$\alpha, \beta,$ and γ are hyperparameters.

Denkowski, Michael, and Alon Lavie. "Meteor 1.3: Automatic metric for reliable optimization and evaluation of machine translation systems." Proceedings of the sixth workshop on statistical machine translation. 2011.

METEOR is another automatic evaluation metric for machine translation

- METEOR (Metric for Evaluation of Translation with Explicit ORdering) combines n-gram precision and recall.

$$MeteorScore = \frac{Prec \times Recall}{\alpha . Prec + (1-\alpha)\, Recall}\left(1 - \gamma \left(\frac{chunks}{u_m}\right)^{\beta}\right)$$

$chunks$ = bigram/trigram matches, u_m = unigrams in candidate
$\alpha, \beta,$ and γ are hyperparameters.

Source: Auf der Matte saß die Katze

Candidate: On the mat sat the cat

Reference: The cat sat on the mat

$$Prec = \frac{n_m}{n_c}$$

$$Recall = \frac{n_m}{n_r}$$

Denkowski, Michael, and Alon Lavie. "Meteor 1.3: Automatic metric for reliable optimization and evaluation of machine translation systems." Proceedings of the sixth workshop on statistical machine translation. 2011.

METEOR is another automatic evaluation metric for machine translation

- METEOR (Metric for Evaluation of Translation with Explicit ORdering) combines n-gram precision and recall.

$$MeteorScore = \frac{Prec \times Recall}{\alpha . Prec + (1-\alpha)\ Recall}\left(1 - \gamma\left(\frac{chunks}{u_m}\right)^{\beta}\right)$$

$chunks$ = bigram/trigram matches, u_m = unigrams in candidate
$\alpha, \beta,$ and γ are hyperparameters.

- Considers semantic similarity for matching.

- Correlates better with human judgement.

Source: Auf der Matte saß die Katze

Candidate: On the mat sat the cat

Reference: The cat sat on the mat

$$Prec = \frac{n_m}{n_c}$$

$$Recall = \frac{n_m}{n_r}$$

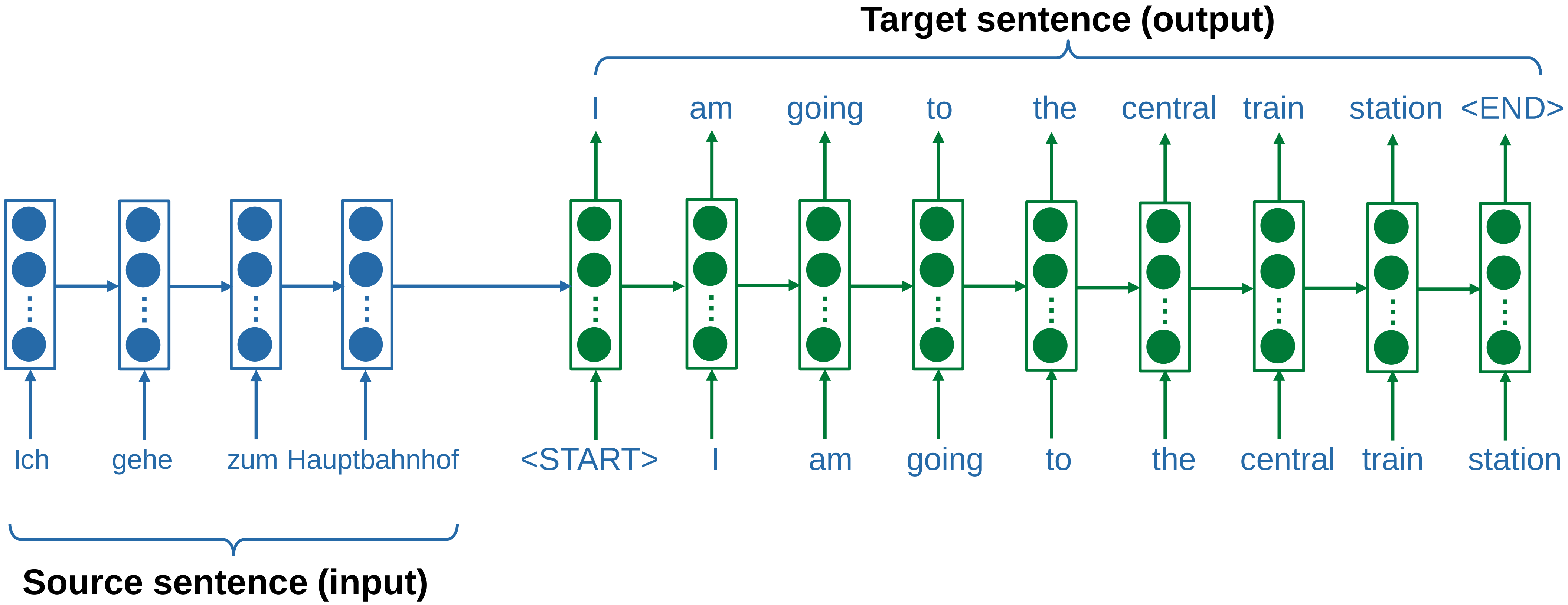
Denkowski, Michael, and Alon Lavie. "Meteor 1.3: Automatic metric for reliable optimization and evaluation of machine translation systems." Proceedings of the sixth workshop on statistical machine translation. 2011.

The problem of machine translation is far from solved

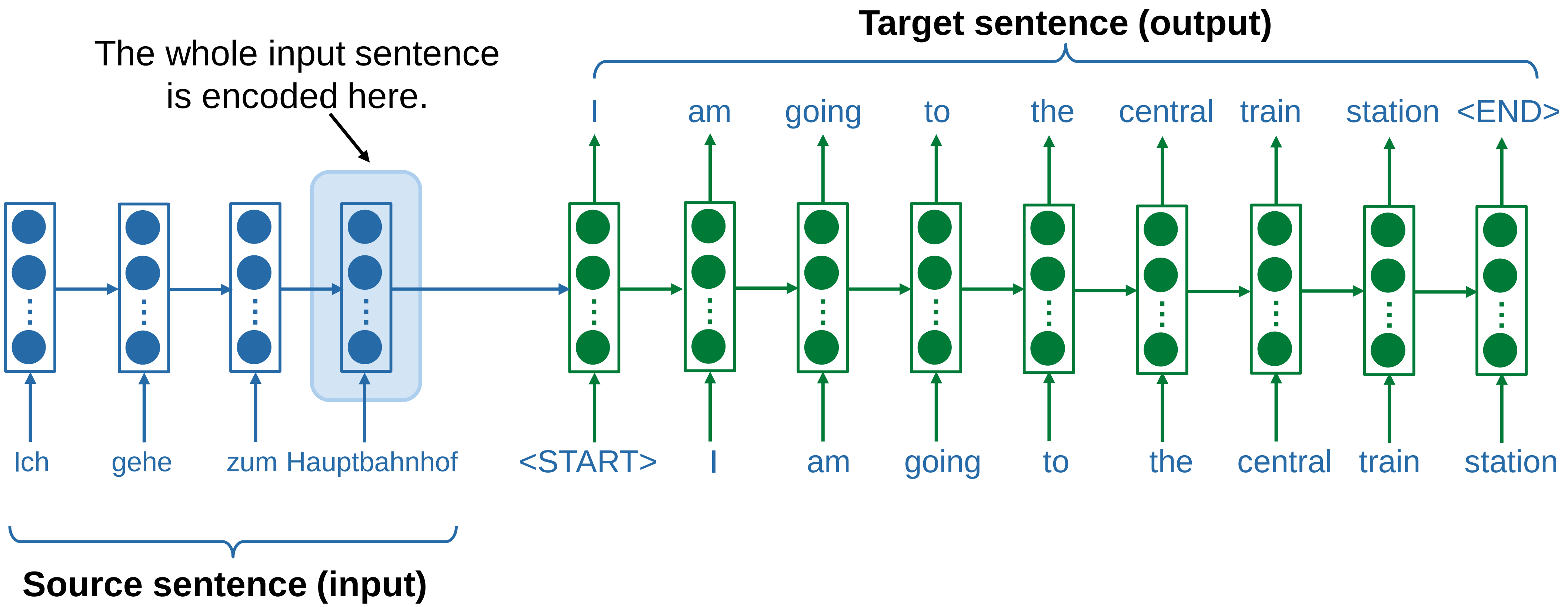
- Machine translation has achieved a lot but many challenges still remain.
  - Out of vocabulary words.
  - Domain mismatch.
  - Maintaining wider context.
  - Low-resource language pairs.

Interesting Read: *https://www.skynettoday.com/editorials/state_of_nmt*

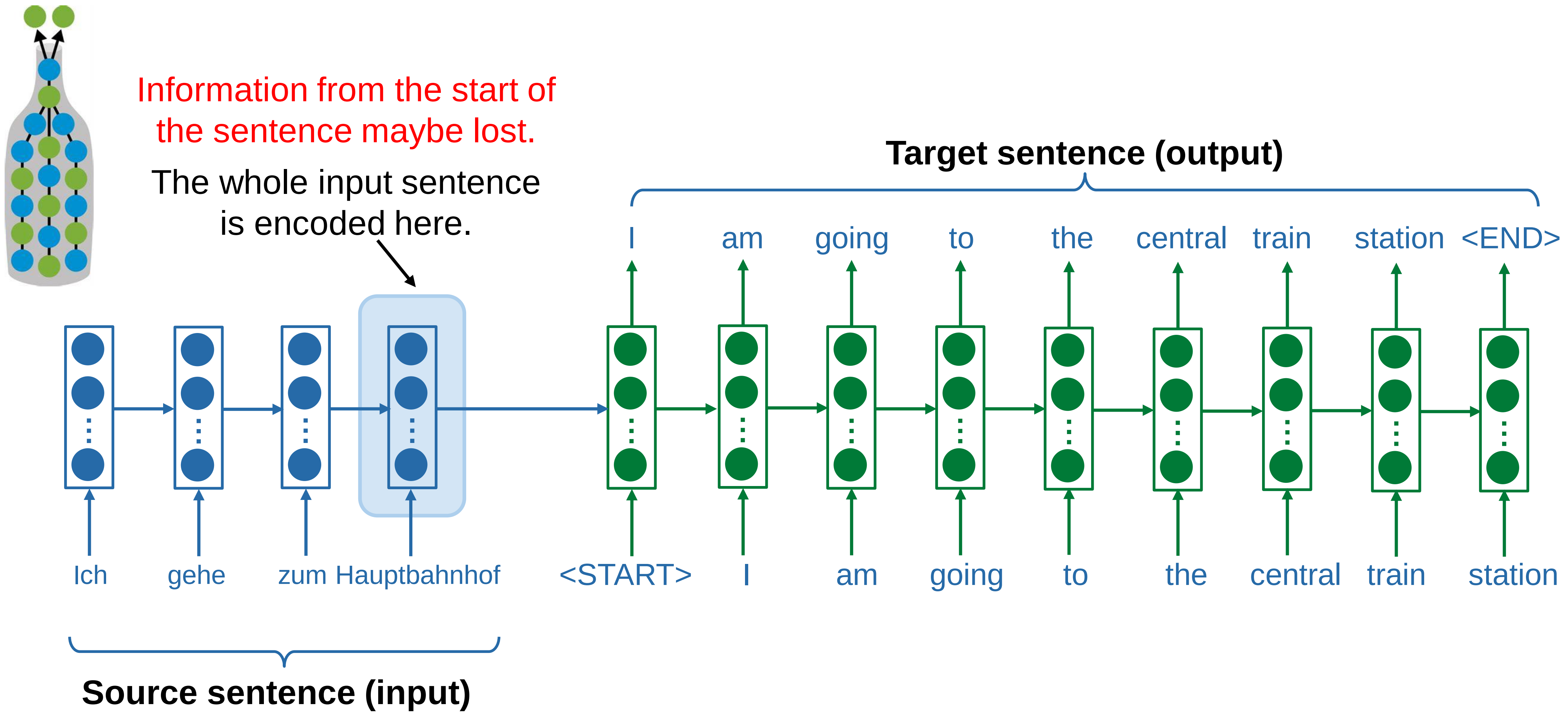
Classical seq2seq model has a few shortcomings



Classical seq2seq model has a few shortcomings



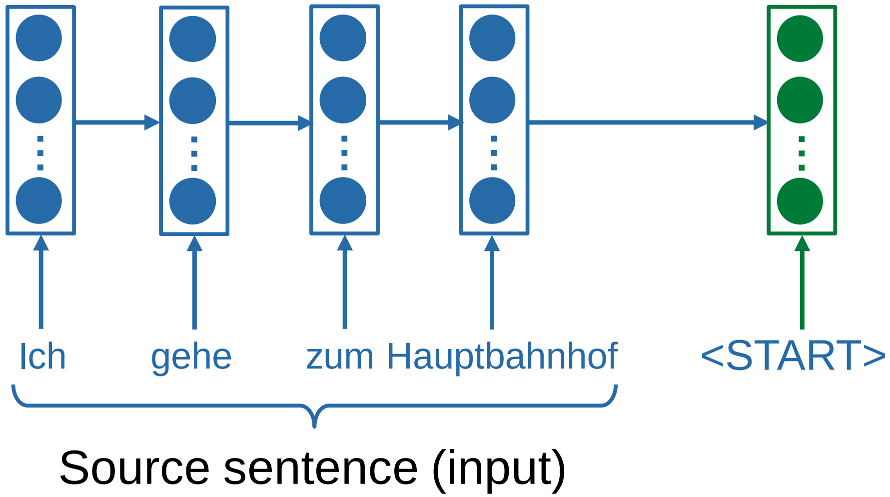
Classical seq2seq model has a few shortcomings



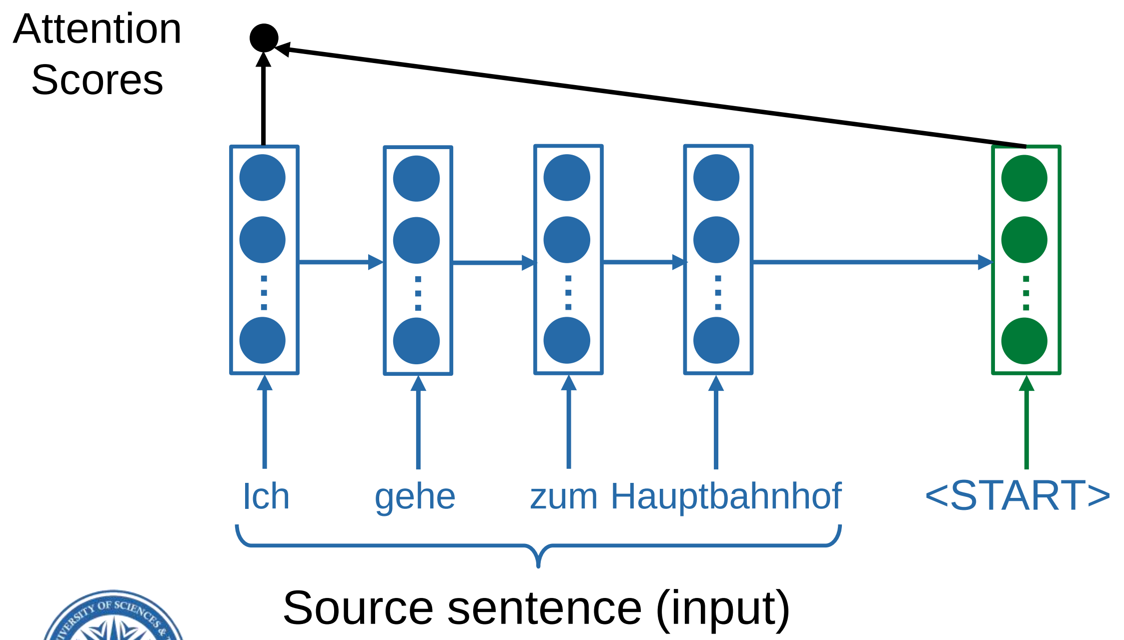
Information bottleneck in seq2sec models can be removed using Attention

Information bottleneck in seq2sec models can be removed using Attention

Information bottleneck in seq2sec models can be removed using Attention



At each decoder time step t, take dot product of decoder activation $da^{<t>}$ and encoder activations $ea^{<1>}, ea^{<2>}, \ldots, ea^{<T_x>}$ to get attention scores.

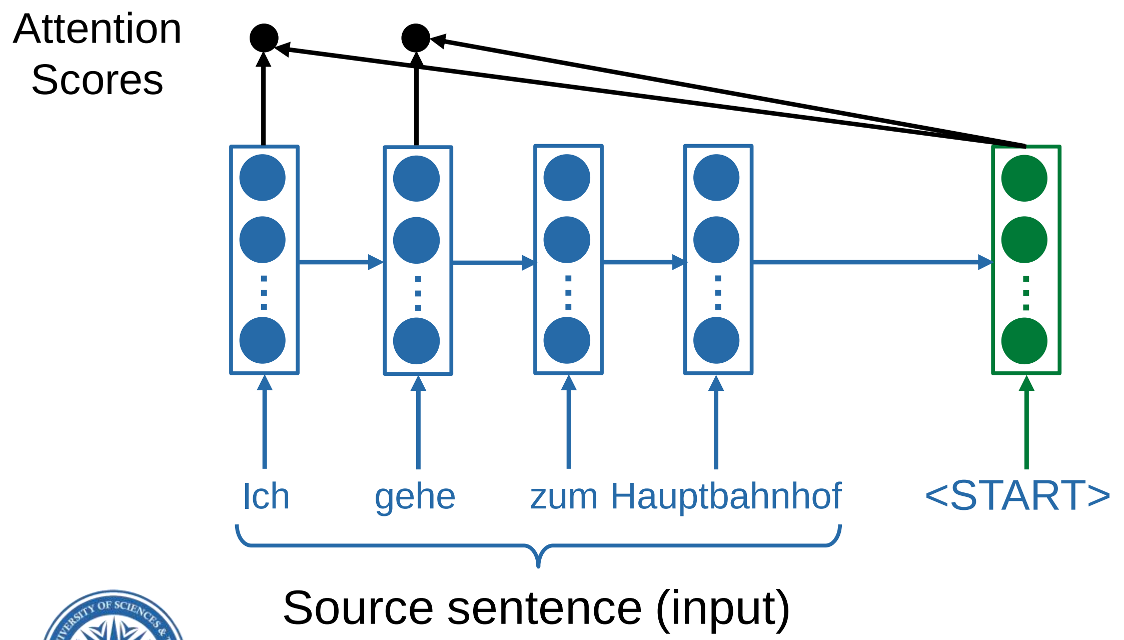


Mathematically,

$$e_1^{<t>} = [\boldsymbol{da}^{<t>}]^T . \boldsymbol{ea}^{<1>}$$

Information bottleneck in seq2sec models can be removed using Attention

At each decoder time step t, take dot product of decoder activation $da^{<t>}$ and encoder activations $ea^{<1>}, ea^{<2>}, \dots, ea^{<T_x>}$ to get attention scores.

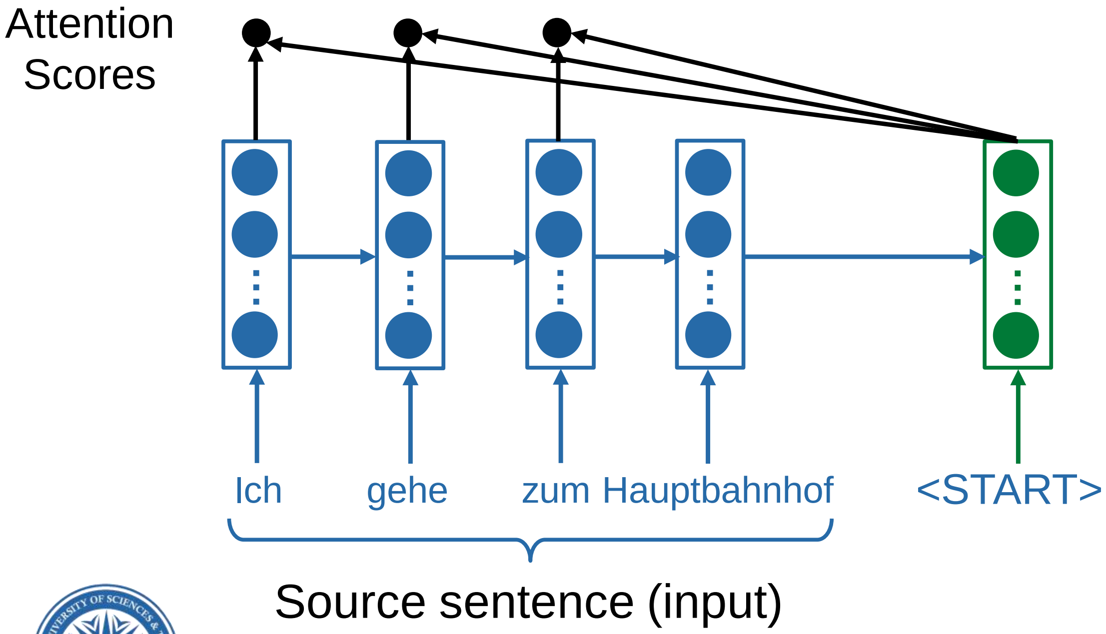


Mathematically,

$$e_2^{<t>} = [\boldsymbol{da}^{<t>}]^T . \boldsymbol{ea}^{<2>}$$

Information bottleneck in seq2sec models can be removed using Attention

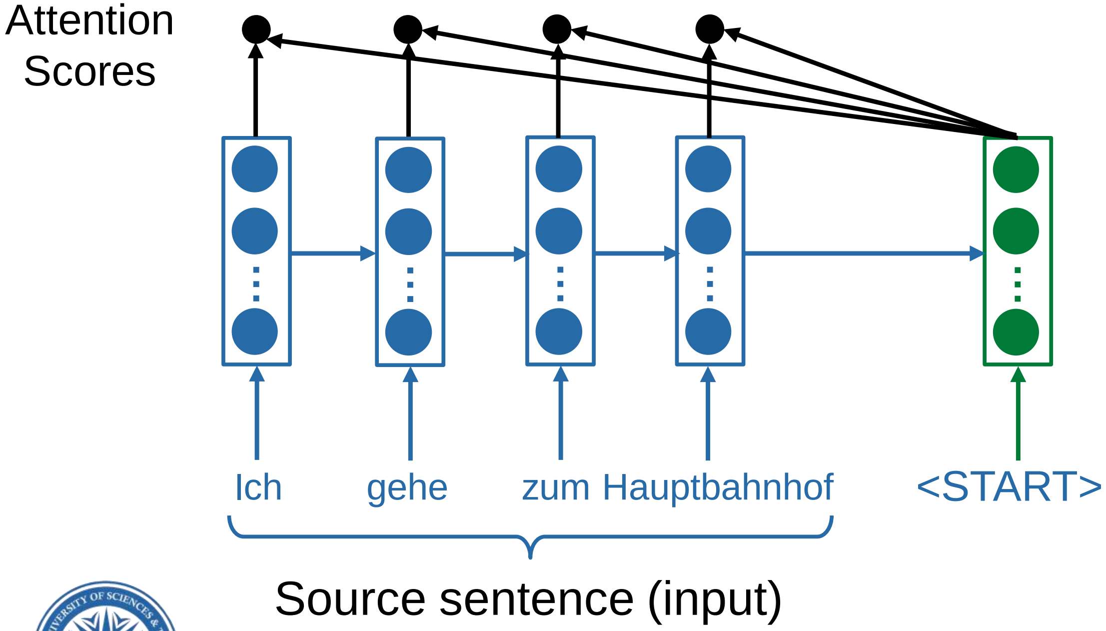
At each decoder time step t, take dot product of decoder activation $da^{<t>}$ and encoder activations $ea^{<1>}, ea^{<2>}, \dots, ea^{<T_x>}$ to get attention scores.



Mathematically,

$$e_3^{<t>} = [\boldsymbol{da}^{<t>}]^T . \boldsymbol{ea}^{<3>}$$

At each decoder time step t, take dot product of decoder activation $da^{<t>}$ and encoder activations $ea^{<1>}, ea^{<2>}, \ldots, ea^{<T_x>}$ to get attention scores.



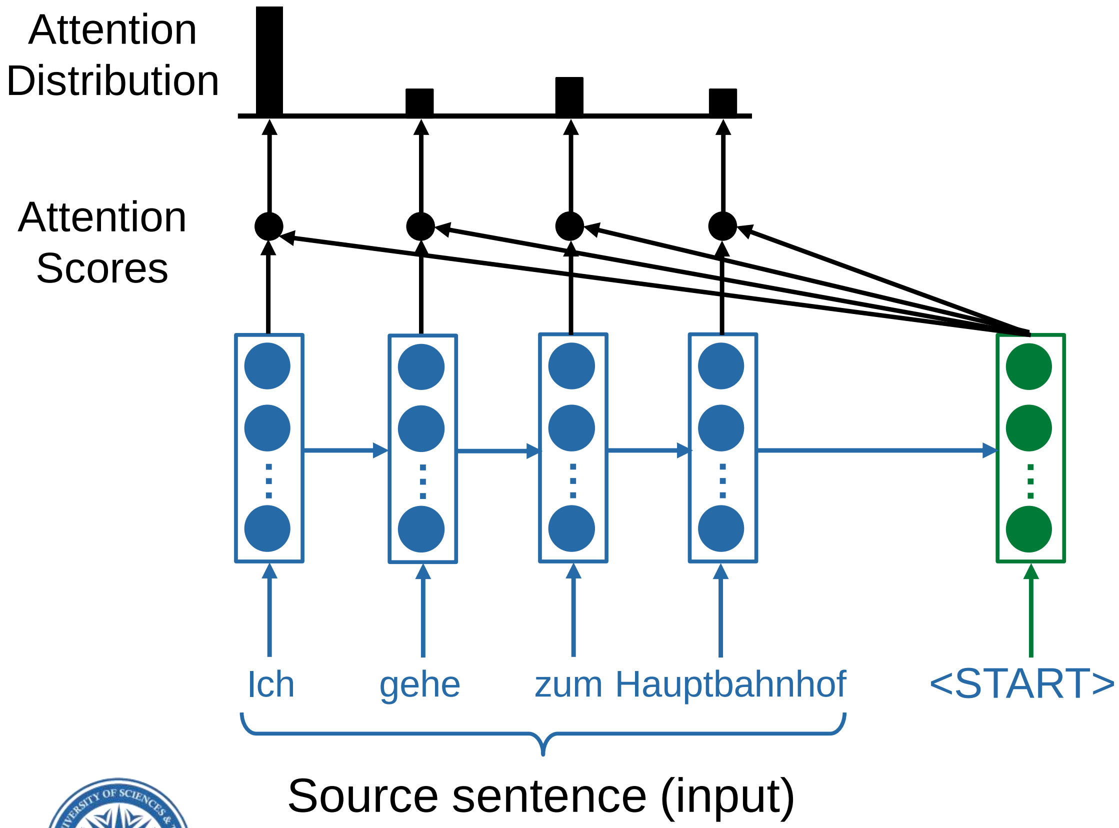
Mathematically,

$$e_4^{<t>} = [\boldsymbol{da}^{<t>}]^T . \boldsymbol{ea}^{<4>}$$

$$\boldsymbol{e}^t = [e_1^{<t>}, e_2^{<t>}, \ldots, e_{T_x}^{<t>}]$$

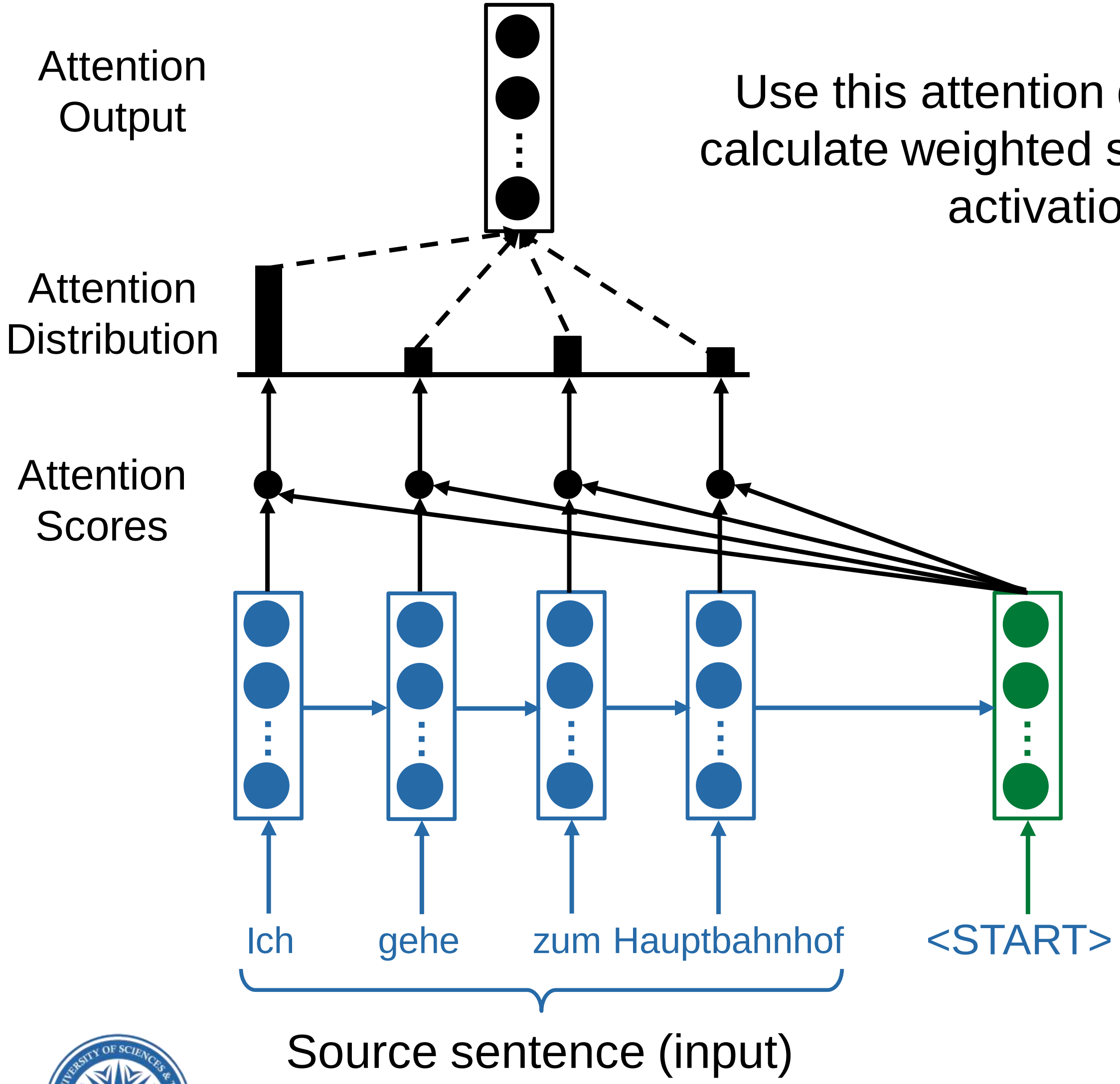
Information bottleneck in seq2sec models can be removed using Attention

Take $softmax$ of attention score to turn these values into probability distribution



Mathematically,

$$\boldsymbol{\alpha}^{<t>} = softmax\,(\boldsymbol{e}^{t})$$

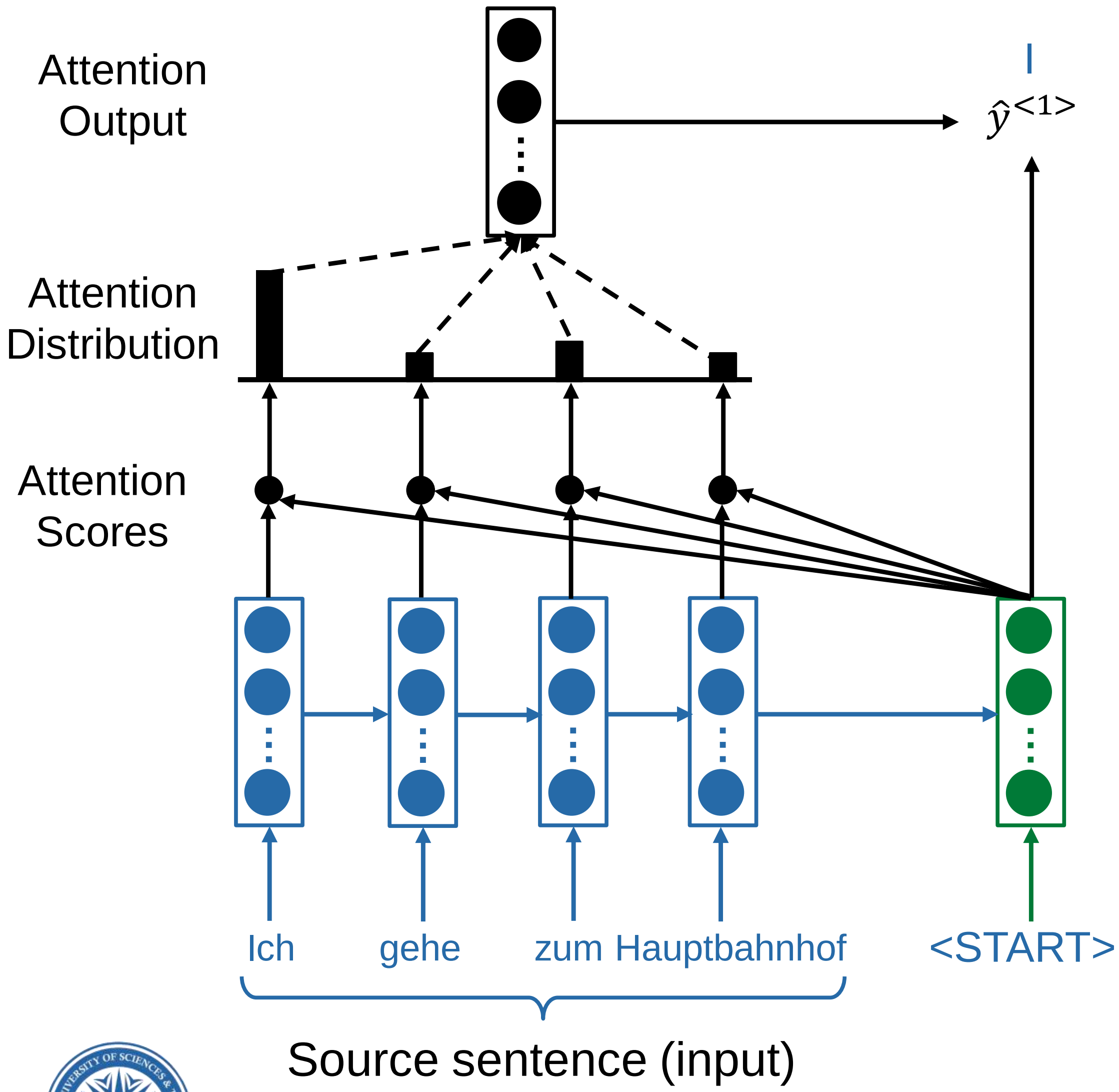


Use this attention distribution to calculate weighted sum of encoder activations.

Mathematically,

$$\boldsymbol{a}^{<t>} = \sum_{i=1}^{T_x} \boldsymbol{\alpha}_i^{<t>} \boldsymbol{ea}^{<i>}$$

Information bottleneck in seq2sec models can be removed using Attention



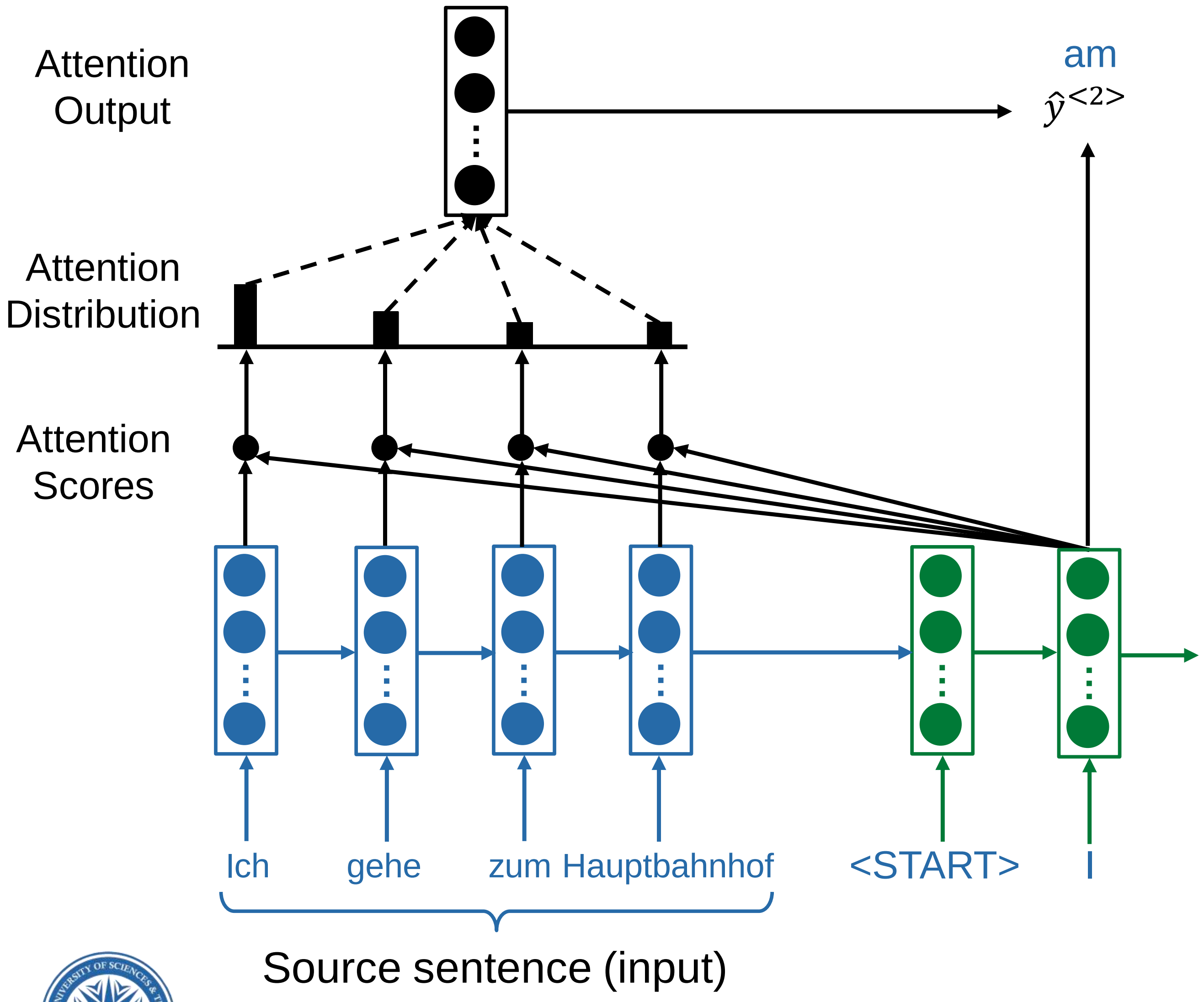
Attention output is used to influence the generation of word at this time step.

Concatenate attention output with decoder activation and calculate $\hat{y}^{<t>}$.

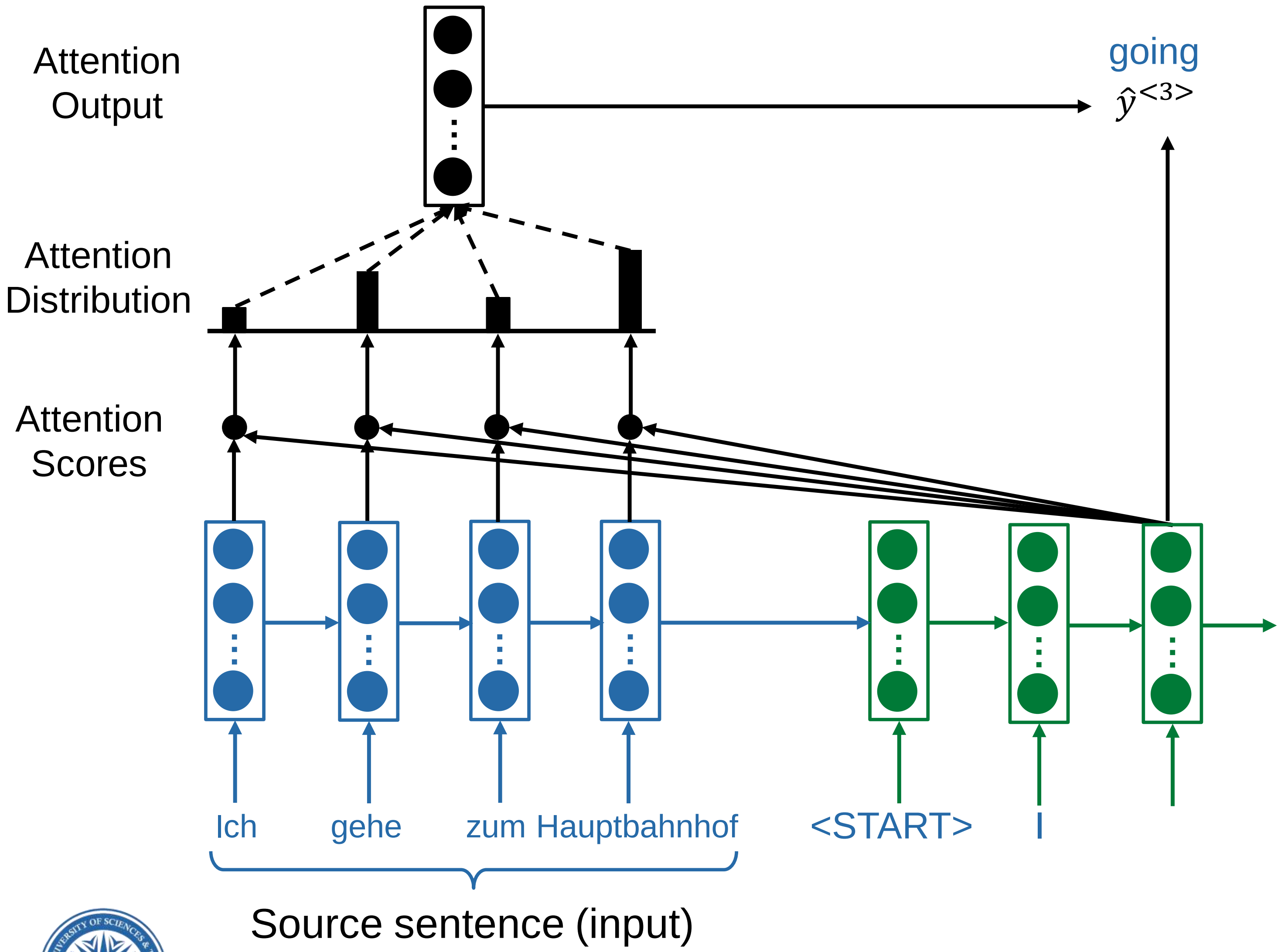
Mathematically,

$$\hat{y}^{<t>} = activation\ ([\boldsymbol{a}^{<t>} : \boldsymbol{da}^{<t>}])$$

Information bottleneck in seq2sec models can be removed using Attention



Information bottleneck in seq2sec models can be removed using Attention



Adding attention to seq2seq models has many advantages

- Attention helps decoder focus on relevant parts in the source sentence.

Adding attention to seq2seq models has many advantages

- Attention helps decoder focus on relevant parts in the source sentence.

- It resolves information bottleneck problem.

    - Instead of relying on a single vector to capture the whole source sentence, now decoder has two vectors for guidance.

Adding attention to seq2seq models has many advantages

- Attention helps decoder focus on relevant parts in the source sentence.

- It resolves information bottleneck problem.

    - Instead of relying on a single vector to capture the whole source sentence, now decoder has two vectors for guidance.

- It also helps with vanishing gradient.

    - Direct connections between encoder and decoder are helpful especially in longer sentences.

Adding attention to seq2seq models has many advantages

- Attention helps decoder focus on relevant parts in the source sentence.
- It resolves information bottleneck problem.
    - Instead of relying on a single vector to capture the whole source sentence, now decoder has two vectors for guidance.
- It also helps with vanishing gradient.
    - Direct connections between encoder and decoder are helpful especially in longer sentences.
- Attention may provide some interpretability.
    - Analysis of attention output can help understand what the decoder was fixating at while predicting a certain target word.
    - Soft alignment is achieved for free without even explicitly training for it.

Attention can be implemented in multiple ways

- To compute $\boldsymbol{e} \in \mathbb{R}^N$ from encoder activations $\boldsymbol{h_1}, \boldsymbol{h_2}, \dots, \boldsymbol{h_N} \in \mathbb{R}^{d_1}$ and decoder activations $\boldsymbol{s} \in \mathbb{R}^{d_2}$, we can use

  - Dot Product Attention.

$$e_i = \boldsymbol{s}^T \boldsymbol{h}_i \in \mathbb{R}$$

It assumes $d_1 = d_2$.

Attention can be implemented in multiple ways

- To compute $\boldsymbol{e} \in \mathbb{R}^N$ from encoder activations $\boldsymbol{h_1}, \boldsymbol{h_2}, \ldots, \boldsymbol{h_N} \in \mathbb{R}^{d_1}$ and decoder activations $\boldsymbol{s} \in \mathbb{R}^{d_2}$, we can use

  - Dot Product Attention.

$$e_i = \boldsymbol{s}^T \boldsymbol{h}_i \in \mathbb{R}$$

  It assumes $d_1 = d_2$.

  - Multiplicative Attention.

$$e_i = \boldsymbol{s}^T \boldsymbol{W} \boldsymbol{h}_i \in \mathbb{R}$$

  Here $\boldsymbol{W} \in \mathbb{R}^{d_2 \times d_1}$ is a learnable weight matrix.

Attention can be implemented in multiple ways

- To compute $\boldsymbol{e} \in \mathbb{R}^N$ from encoder activations $\boldsymbol{h_1}, \boldsymbol{h_2}, \ldots, \boldsymbol{h_N} \in \mathbb{R}^{d_1}$ and decoder activations $\boldsymbol{s} \in \mathbb{R}^{d_2}$, we can use

  - Dot Product Attention.

$$e_i = \boldsymbol{s}^T \boldsymbol{h}_i \in \mathbb{R}$$

  It assumes $d_1 = d_2$

  - Multiplicative Attention.

$$e_i = \boldsymbol{s}^T \boldsymbol{W} \boldsymbol{h}_i \in \mathbb{R}$$

  Here $\boldsymbol{W} \in \mathbb{R}^{d_2 \times d_1}$ is a learnable weight matrix.

  - Additive Attention.

$$e_i = \boldsymbol{v}^T \tanh(\boldsymbol{W}_1 \boldsymbol{h}_i + \boldsymbol{W}_2 \boldsymbol{s}) \in \mathbb{R}$$

  Here $\boldsymbol{W}_1 \in \mathbb{R}^{d_3 \times d_1}$ and $\boldsymbol{W}_2 \in \mathbb{R}^{d_3 \times d_2}$ are learnable weight metrices, $v \in \mathbb{R}^{d3}$ is a weight vector, and d_3 is a hyperparameter called attention dimensionality

Do you have any problem?

Some material (images, tables, text etc.) in this presentation has been borrowed from different books, lecture notes, and the web. The original contents solely belong to their owners, and are used in this presentation only for clarifying various educational concepts. Any copyright infringement is ***not at all*** intended.

NUST
Defining futures
School of Electrical Engineering
& Computer Science