Overview of Natural Language Processing
Part II & ACS L390

Lecture 11: Natural Language Generation

Yulong Chen and Weiwei Sun

Department of Computer Science and Technology
University of Cambridge

Michaelmas 2024/25

Recall the tasks we have introduced so far:

- **PoS-tagging**: {input: sentence} $\rightarrow$ {output: PoS tags}
- **Syntactic Parsing**: {input: sentence} $\rightarrow$ {output: tree structures}
- **Grammar Induction**: {input: sentence} $\rightarrow$ {output: tree structures}
- **Text Classification**: {input: sentence} $\rightarrow$ {output: label}

Natural Language Understanding (NLU):
Process unstructured text into structured information

Recall the tasks we have introduced so far:

- **PoS-tagging**: {input: sentence} $\rightarrow$ {output: PoS tags}
- **Syntactic Parsing**: {input: sentence} $\rightarrow$ {output: tree structures}
- **Grammar Induction**: {input: sentence} $\rightarrow$ {output: tree structures}
- **Text Classification**: {input: sentence} $\rightarrow$ {output: label}

Natural Language Understanding (NLU):
Process unstructured text into structured information

Structured information is more amenable to machine understanding, but not to humans. We need machines to generate human-like languages!

I have a question about whether you've been attempted to look at generation? [...] That is a rich rich area which so few people address [...]

Well, I find generation completely terrifying [...] I am very interested in the problem [...] That's an important question.

Mark Steedman
FBA, FRSE

ACL lifetime archievement award lecture (vimeo.com/288152682)

equally (maybe even more) important to NLU

Lecture 11: Natural Language Generation

1. What is NLG?
2. Text-to-Text Generation Modeling

What is NLG?

Task Overview

Broadly, NLG is the task that asks machines to **generate human-like text**. Mathematically, NLG is a mapping problem:

$$NLG : X \rightarrow Y, \tag{1}$$

where X is the input, and Y is the output.

- Output: text (which humans can interact with)
- Input: structured representation (semantic representations, syntactic trees, etc)

Task Overview

Broadly, NLG is the task that asks machines to **generate human-like text**. Mathematically, NLG is a mapping problem:

$$NLG : X \rightarrow Y, \tag{1}$$

where X is the input, and Y is the output.

- Output: text (which humans can interact with)
- Input: structured representation (semantic representations, syntactic trees, etc)

NLG is more than the inverse of NLU!

Generation from What?!

- **Structured/Semi-Structured Representation**:
  - **Realization**: syntactic/semantic representations (AMR, SRL, logical form, etc)
  - **Data-to-text generation**: tables, databases, knowledge bases, etc
  - ...
- **Text**:
  - **Machine Translation**: from one language to another
  - **Text Summarisation**: from long/multiple texts to short text
  - **Chatbot/QA**: from user input (text/questions) to meaningful responses in dialogue form
  - ...
- **Other Modalities**:
  - **Image Caption Generation/QA**: from visual to text
  - **Video Summarisation**: from visual and audio to text
  - ...

NLG vs NLU

	NLU Task	**NLG Task**
Goal	Extract and understand syntactic/semantic/discourse/pragmatic information from texts	Generate human-like natural language expression based on task requirements
Input	natural language expression	(semi-) structured data, natural language expression, image, etc
Output	structured data	natural language expression
Typical Tasks	CCG parsing, discourse analysis, sentiment analysis	text summarisation, MT, dialogue response generation
Main Challenge	Context-dependent, ambiguity, et (that impact the understanding)	fluency, coherence, consistency, faithfulness, and in particular factuality (in the era of LLM)

The NLG/NLU **tasks** are stated from the input/output perspective.

Text-to-Text Generation Modeling: Text Summarisation as Example

Task Introduction

Take single-document (e.g., newspaper, dialogue) summarisation for example:

- **Task**: Given an input document $X = x_1, x_2, ..., x_{|X|}$, the goal is to generate a concise version or summary $Y = y_1, y_2, ..., y_{|Y|}$,[1] where the summary should be much shorter than the document, i.e. $|X| >> |Y|$.
- **Goal**: reduce the length of the input: remove redundant information; keep most salient information.
- **Application**: News abstract generation, SemanticScholar (TL;DR), even the potential title/topic recommendation (like in TikTok), etc.
- **Variant**: multi-doc summarisation, query-based summarisation, video summarisation, etc.

[1] Each x_i represents the i-th token in the document and y_j represents the j-th token in the summary. $|X|$ and $|Y|$ refer to the lengths of the document and summary, respectively, in terms of tokens.

Modeling

1. Statistic and heuristic method
    - Word frequency (Luhn, 1958)
2. Machine-learning methods
    - Bayes classifier (Kupiec et al., 1995)
3. Deep-learning based method
    - Sequence-to-Sequence (Rush et al., 2015)

But before that, let's get a more direct sense of the task

Given a **news** article, what is the laziest and easiest way to get its main information (summary)?

Adele announces 'random' Munich residency

31 January

Adele's residency in Las Vegas runs until June

By Yasmin Rufo & Ian Youngs
BBC News

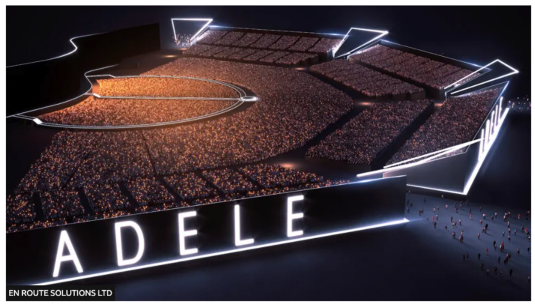
Adele has announced four concerts in a specially-built stadium in Munich this summer - an idea she described as "a bit random, but still fabulous!"

The shows will take place in an 80,000-capacity open-air venue in the German city on 2, 3, 9 and 10 August.

It will be the first time the singer has performed in mainland Europe since her last tour in 2016.

She said she hadn't been planning any more shows after her current Las Vegas residency and two London shows in 2022.

But the star said she had been tempted by the offer of "a one off, bespoke pop-up stadium designed around whatever show I want to put on", which was "pretty much slap bang in the middle of Europe".



An artist's impression of the Munich Messe, the venue for Adele's shows

"I couldn't think of a more wonderful way to spend my summer and end this beautiful phase of my life and career with shows closer to home during such an exciting summer," she wrote on social media.

She ended her message: "Guten Tag babes x"

Ticket registration is open until 5 February, with the pre-sale due to start on 7 February.

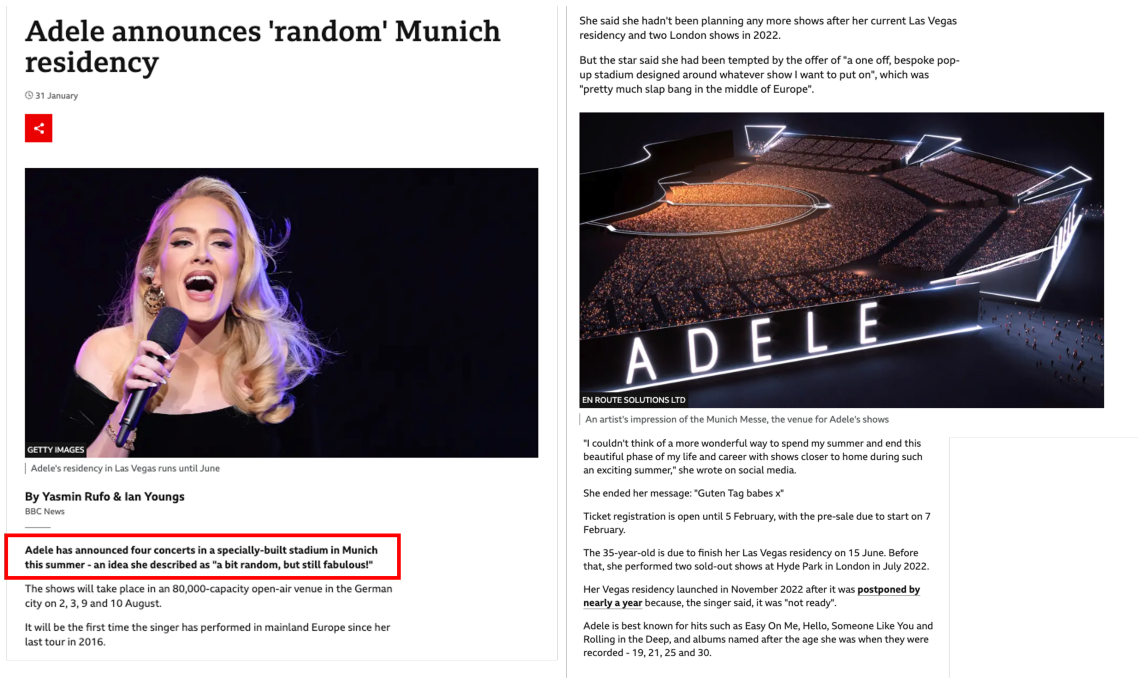
The 35-year-old is due to finish her Las Vegas residency on 15 June. Before that, she performed two sold-out shows at Hyde Park in London in July 2022.

Her Vegas residency launched in November 2022 after it was **postponed by nearly a year** because, the singer said, it was "not ready".

Adele is best known for hits such as Easy On Me, Hello, Someone Like You and Rolling in the Deep, and albums named after the age she was when they were recorded - 19, 21, 25 and 30.

But before that, let's get a more direct sense of the task

Given a **news** article, what is the laziest and easiest way to get its main information (summary)?



Adele announces 'random' Munich residency

31 January

Adele's residency in Las Vegas runs until June

By Yasmin Rufo & Ian Youngs
BBC News

Adele has announced four concerts in a specially-built stadium in Munich this summer - an idea she described as "a bit random, but still fabulous!"

The shows will take place in an 80,000-capacity open-air venue in the German city on 2, 3, 9 and 10 August.

It will be the first time the singer has performed in mainland Europe since her last tour in 2016.

She said she hadn't been planning any more shows after her current Las Vegas residency and two London shows in 2022.

But the star said she had been tempted by the offer of "a one off, bespoke pop-up stadium designed around whatever show I want to put on", which was "pretty much slap bang in the middle of Europe".

An artist's impression of the Munich Messe, the venue for Adele's shows

"I couldn't think of a more wonderful way to spend my summer and end this beautiful phase of my life and career with shows closer to home during such an exciting summer," she wrote on social media.

She ended her message: "Guten Tag babes x"

Ticket registration is open until 5 February, with the pre-sale due to start on 7 February.

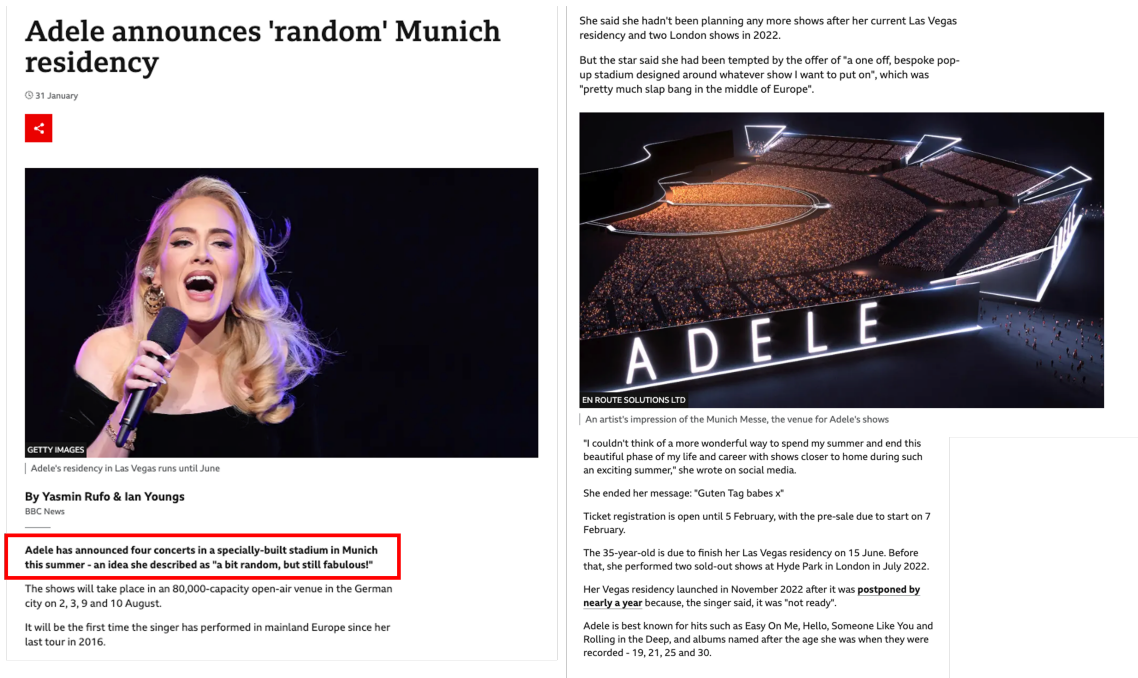
The 35-year-old is due to finish her Las Vegas residency on 15 June. Before that, she performed two sold-out shows at Hyde Park in London in July 2022.

Her Vegas residency launched in November 2022 after it was **postponed by nearly a year** because, the singer said, it was "not ready".

Adele is best known for hits such as Easy On Me, Hello, Someone Like You and Rolling in the Deep, and albums named after the age she was when they were recorded - 19, 21, 25 and 30.

But before that, let's get a more direct sense of the task

Given a **news** article, what is the laziest and easiest way to get its main information (summary)?



Adele announces 'random' Munich residency

31 January

GETTY IMAGES

Adele's residency in Las Vegas runs until June

By Yasmin Rufo & Ian Youngs

BBC News

Adele has announced four concerts in a specially-built stadium in Munich this summer - an idea she described as "a bit random, but still fabulous!"

The shows will take place in an 80,000-capacity open-air venue in the German city on 2, 3, 9 and 10 August.

It will be the first time the singer has performed in mainland Europe since her last tour in 2016.

She said she hadn't been planning any more shows after her current Las Vegas residency and two London shows in 2022.

But the star said she had been tempted by the offer of "a one off, bespoke pop-up stadium designed around whatever show I want to put on", which was "pretty much slap bang in the middle of Europe".

An artist's impression of the Munich Messe, the venue for Adele's shows

"I couldn't think of a more wonderful way to spend my summer and end this beautiful phase of my life and career with shows closer to home during such an exciting summer," she wrote on social media.

She ended her message: "Guten Tag babes x"

Ticket registration is open until 5 February, with the pre-sale due to start on 7 February.

The 35-year-old is due to finish her Las Vegas residency on 15 June. Before that, she performed two sold-out shows at Hyde Park in London in July 2022.

Her Vegas residency launched in November 2022 after it was **postponed by nearly a year** because, the singer said, it was "not ready".

Adele is best known for hits such as Easy On Me, Hello, Someone Like You and Rolling in the Deep, and albums named after the age she was when they were recorded - 19, 21, 25 and 30.

Find its leading sentence(s)!

Statistic and heuristic method

The main idea of lead-n method:

the key information is concentrated in its leading text

Method:

- Use the leading n sentences as summary

Comments:

- Very simple (the simplest) but **surprisingly effective** on texts that have **an inverted pyramid structure** (e.g., news).
- Easily extended to other variants: longest-n, last-n, etc.

Statistic and heuristic method

The main idea of lead-n method:

the key information is concentrated in its leading text

Method:

- Use the leading n sentences as summary

Comments:

- Very simple (the simplest) but **surprisingly effective** on texts that have **an inverted pyramid structure** (e.g., news).
- Easily extended to other variants: longest-n, last-n, etc.

Hint here: always look into your data and be familiar with what task you are doing!

Statistic and heuristic method

The main idea of lead-n method:

the key information is concentrated in its leading text

Method:

- Use the leading n sentences as summary

Comments:

- Very simple (the simplest) but surprisingly effective on texts that have an inverted pyramid structure (e.g., news).
- Easily extended to other variants: longest-n, last-n, etc.
- But it is biased towards positions. Perform poorly on other texts (e.g., academic papers).

Statistic and heuristic method

In addition to positions, what features can tell the importance of a sentence?

Statistic and heuristic method

The main idea in (Luhn, 1958):

"... frequency of word occurrence in an article furnishes a useful measurement of word significance" (Luhn, 1958)

Statistic and heuristic method

The main idea in (Luhn, 1958):

"... frequency of word occurrence in an article furnishes a useful measurement of word significance" (Luhn, 1958)

Method:

1. Given an article, calculate the frequencies of words in it.
2. Filter out very high-frequent and low-frequent words.
3. Calculate *significance factor* of a sentence based on the word frequency.
4. Use sentences with the highest significance factor scores as summaries.

Statistic and heuristic method

The main idea in (Luhn, 1958):

"... frequency of word occurrence in an article furnishes a useful measurement of word significance" (Luhn, 1958)

Method:

1. Given an article, calculate the frequencies of words in it.
2. Filter out very high-frequent and low-frequent words.
3. Calculate *significance factor* of a sentence based on the word frequency.
4. Use sentences with the highest significance factor scores as summaries.

Comments:

1. Introduce word frequency as statistical feature into summarisation.
2. Introduce the concept of selecting sentences by scores.
3. However, the feature is shallow and local (no semantic info, sentence relation, etc).

Both lead-n and Luhn (1958) select important sentences from the document using shallow features in a heuristic way, and both are free of training (why?)

Both lead-n and Luhn (1958) select important sentences from the document using shallow features in a heuristic way, and both are free of training (why?)

- At that time, there was no annotated data for training. The development of NLP/AI/ML is highly related to the development of data.
- No (good) ML methods.

Both lead-n and Luhn (1958) select important sentences from the document using shallow features in a heuristic way, and both are free of training (why?)

- At that time, there was no annotated data for training. The development of NLP/AI/ML is highly related to the development of data.
- No (good) ML methods.

With the availability of data, and development traditional machine-learning:

- Can you come up with an idea to model summarisation with the knowledge that you have learnt in this course so far?

Both lead-n and Luhn (1958) select important sentences from the document using shallow features in a heuristic way, and both are free of training (why?)

- At that time, there was no annotated data for training. The development of NLP/AI/ML is highly related to the development of data.
- No (good) ML methods.

With the availability of data, and development traditional machine-learning:

- Can you come up with an idea to model summarisation with the knowledge that you have learnt in this course so far?

Use the NLU method to model text summarisation:

- For each sentence, we can **classify**: *whether it should be included in the summary or not.*
- With more data, we can train our classifier instead of heuristic scoring.

Machine-learning methods

A Trainable Document Summarizer. Kupiec et al., 1995.

The main idea of Kupiec et al., (1995):

Use text classification methods to extract sentences from the document.

Machine-learning methods

A Trainable Document Summarizer. Kupiec et al., 1995.

The main idea of Kupiec et al., (1995):

Use text classification methods to extract sentences from the document.

Estimate the probability:

- Input: a sentence x from a document X, i.e., $x \in X$.
- Output: a binary label c whether sentence x should be included in a summary Y.
- Training corpus: $D = \{(x_i, c_i)\}_{i=1}^{n}$.
- Goal of modeling: $P(c|x)$

Machine-learning methods

Estimate the probability:

- Direct parameterisation from training corpus:

$$P(c|x) = \frac{\text{count}(c, x) \in D}{\text{count}(x) \in D} \tag{2}$$

x is a sentence, which is too sparse. For an unseen x^*, we cannot estimate its probability using MLE!

Machine-learning methods

Estimate the probability:

- Direct parameterisation from training corpus:

$$P(c|x) = \frac{\text{count}(c, x) \in D}{\text{count}(x) \in D} \tag{2}$$

  x is a sentence, which is too sparse. For an unseen x^*, we cannot estimate its probability using MLE!

- Instead, we use generative method for parameterization (e.g., Naïve Bayes classifier):

$$P(c|x) = \frac{P(x|c)P(c)}{P(x)} \tag{3}$$

  note: $P(x)$ is a constant and can be ignored (because it is the same for all sentences). $P(c)$ is easy to estimate, which is the frequency of label c in training data, and can also be ignored in this task.

Machine-learning methods

Kupiec et al., (1995)'s Naïve Bayes classifier:

- Thus, we can estimate the probability by:

$$P(c|x) \propto P(x|c) \tag{4}$$

- Assume x can be represented by independent features (e.g., word, etc):

$$P(x|c) = P(f_1, f_2, ..f_n|c) = P(f_1|c)P(f_2|c)...P(f_n|c) \tag{5}$$

- Then we can estimate $P(f_i|c)$ by its frequency in the training data, i.e.:

$$P(f_i|c) = \frac{\text{count}(f_i, c) \in D}{\text{count}(c) \in D} \tag{6}$$

- Finally:

$$P(c|x) \propto P(x|c) \approx P(f_1, f_2, ..f_n|c) \approx \prod_{i=1}^{n} P(f_i|c) \tag{7}$$

Machine-learning methods

Kupiec et al., (1995)'s Naïve Bayes classifier:

- We estimate the probability by:

$$P(c|x) \propto \prod_{i=1}^{n} P(f_i|c) \tag{8}$$

- Now, we can select summary sentences by their probabilities, or we can learn a threshold from the training corpus.

Machine-learning methods

Kupiec et al., (1995)'s Naïve Bayes classifier:

- We estimate the probability by:

$$P(c|x) \propto \prod_{i=1}^{n} P(f_i|c) \tag{8}$$

- Now, we can select summary sentences by their probabilities, or we can learn a threshold from the training corpus.

Kupiec et al., (1995) consider rich features than simple word frequency:

- Sentence Length Cut-off Feature: sentence longer than 5 words or not.
- Fixed-Phrase Feature: sentence contains specific phrases or not ("*in conclusion*", "in summary", etc).
- ...

All summarisation methods we have introduced so far:

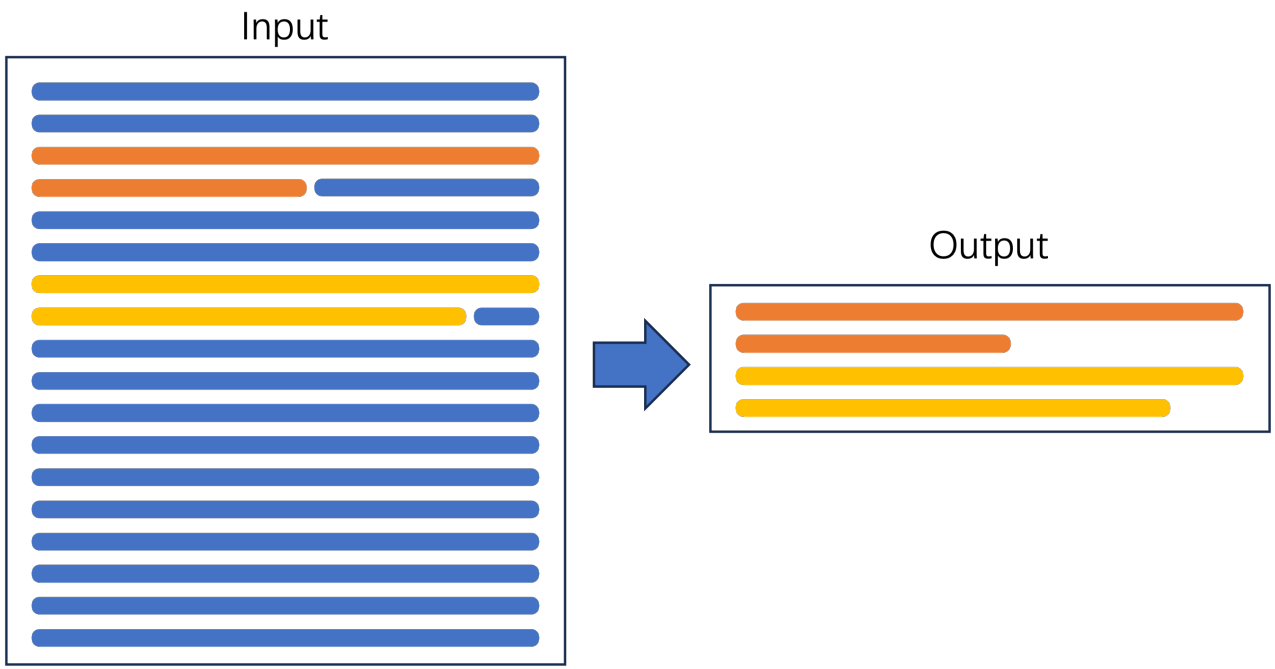
- lead-n (longest-n, last-n)
- Luhn's (1958) word frequency
- Kupiec et al.'s (1995) naïve bayes classifer

They select sentences from the input text to construct the output summary.

Extractive summarisation:

- Methods that generate summaries by selecting and extracting important sentences from the input text, i.e.:

$$Y = \{y_1, y_2, \ldots, y_k\} \quad \text{where} \quad y_i \in X \quad \text{for each } i = 1, 2, \ldots, k \tag{9}$$



Consider the below case of dialogue summarisation: Which summary is better?

Input

#Person_1#: Good morning. I wonder whether you have got an answer from your superior.
#Person_2#: Yes, we had a meting about it yesterday afternoon.
#Person_1#: What's the answer?
#Person_2#: We decided that we could agree to your price, but we are a bit worried about the slow delivery.
#Person_1#: Let me see. I quoted your delivery in three months, didn't I?
#Person_2#: Yes, but we hope that the wool could reach us as soon as possible.
#Person_1#: I thought you would. So I rang Auckland last night. As you are our biggest customer, they agreed to ship the order on the first vessel available that will leave Auckland next month.
#Person_2#: Good, if you agree we'll draft the agreement right away and sign it then.
#Person_1#: By all means.

Summary 1

#Person_1#: I thought you would. So I rang Auckland last night. As you are our biggest customer, they agreed to ship the order on the first vessel available that will leave Auckland next month.

Summary 2

#Person_1# and #Person_2# agree to sign an agreement since #Person_1# could speed up the delivery as #Person_2# hopes.

Consider the below case of dialogue summarisation: Which summary is better?

Input

#Person_1#: Good morning. I wonder whether you have got an answer from your superior.
#Person_2#: Yes, we had a meting about it yesterday afternoon.
#Person_1#: What's the answer?
#Person_2#: We decided that we could agree to your price, but we are a bit worried about the slow delivery.
#Person_1#: Let me see. I quoted your delivery in three months, didn't I?
#Person_2#: Yes, but we hope that the wool could reach us as soon as possible.
#Person_1#: I thought you would. So I rang Auckland last night. As you are our biggest customer, they agreed to ship the order on the first vessel available that will leave Auckland next month.
#Person_2#: Good, if you agree we'll draft the agreement right away and sign it then.
#Person_1#: By all means.

Summary 1

#Person_1#: I thought you would. So I rang Auckland last night. As you are our biggest customer, they agreed to ship the order on the first vessel available that will leave Auckland next month.

Summary 2

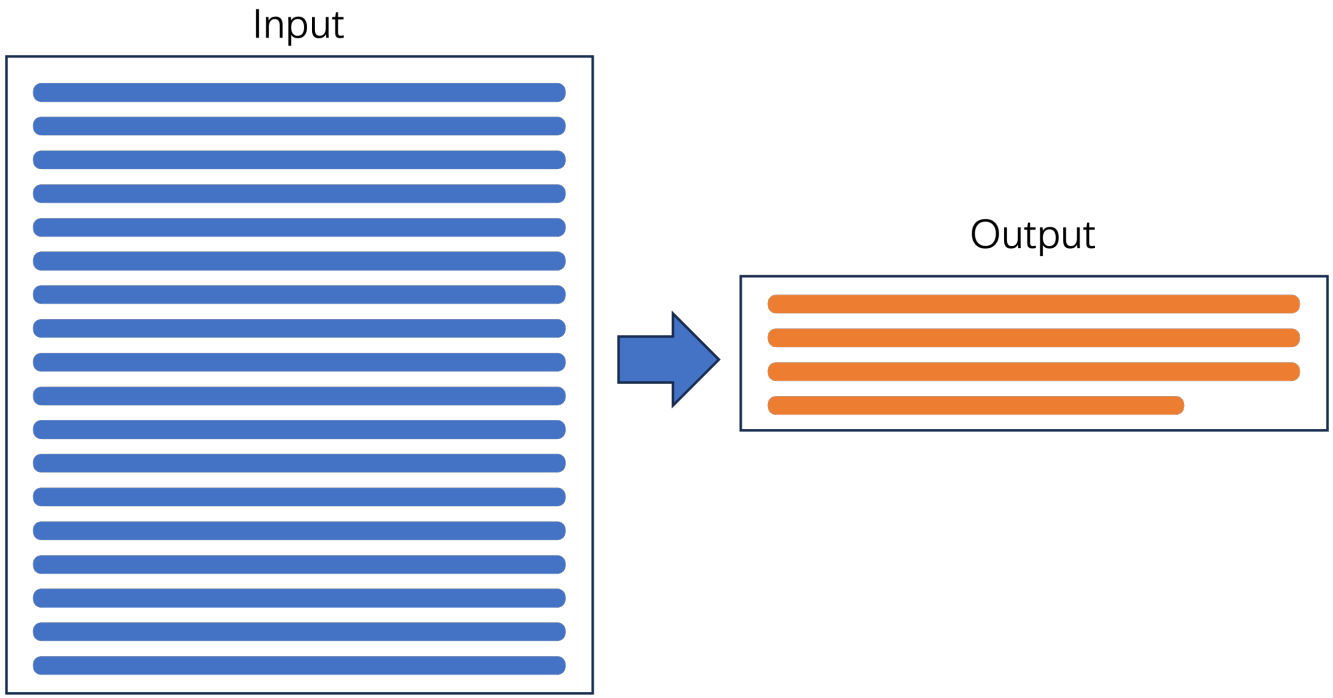
#Person_1# and #Person_2# agree to sign an agreement since #Person_1# could speed up the delivery as #Person_2# hopes.

Extractive summarisation:

- Lack coherence between sentences.
- Redundant information.
- Can be difficult to understand with the original context.

Abstractive summarisation:

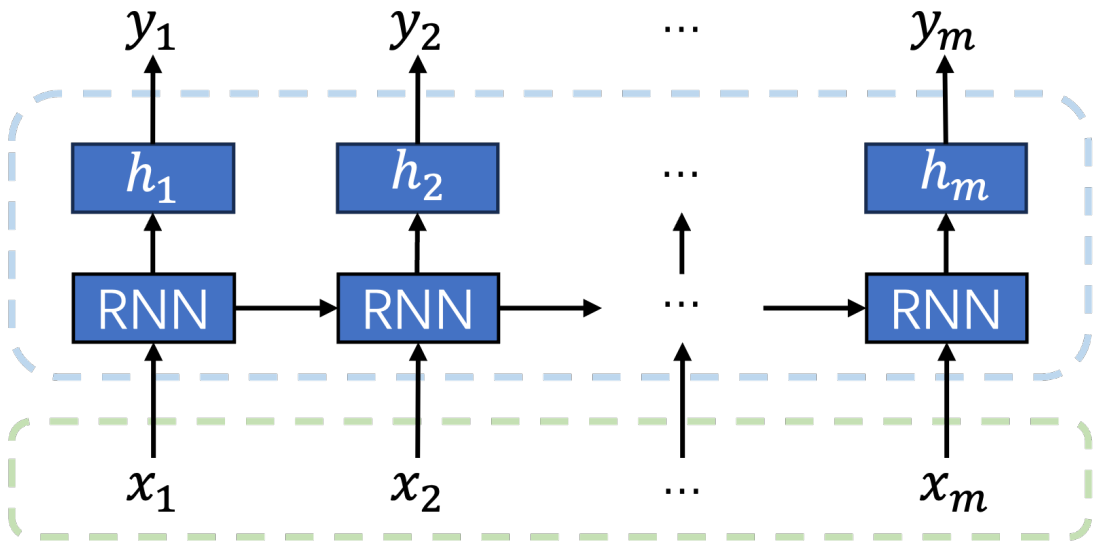
With the development of **deep-learning technologies** and the availability of **large corpora**, generative models can deeply understand input text and generate **new sentences**.



Deep-learning based method

Since we are dealing with two texts, can we model it directly?

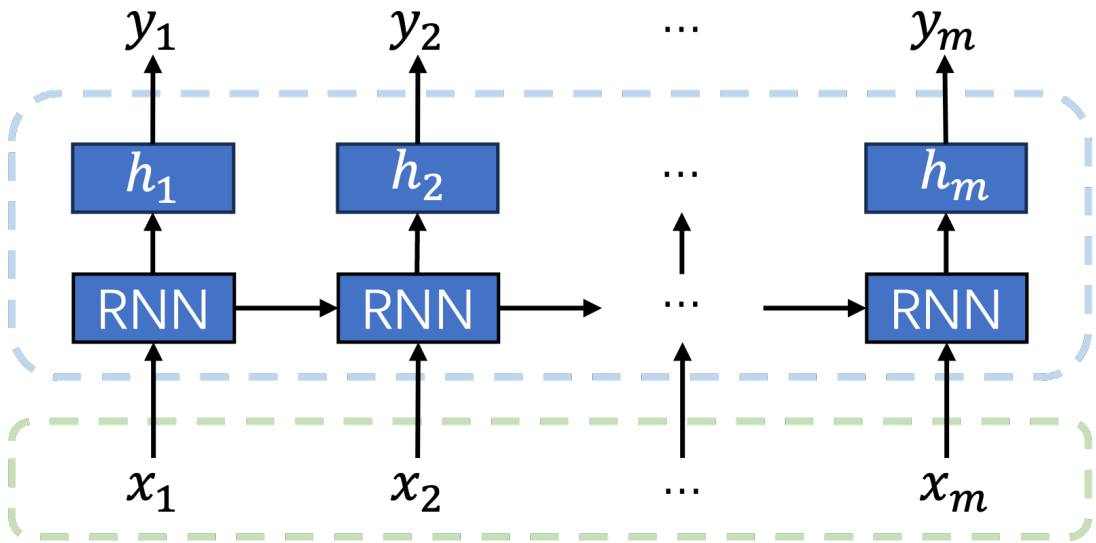
- Directly map one sequence into another sequence.
- Actually, an RNN can be potentially used to generate a sequence:



Deep-learning based method

Since we are dealing with two texts, can we model it directly?

- Directly map one sequence into another sequence.
- Actually, an RNN can be potentially used to generate a sequence:



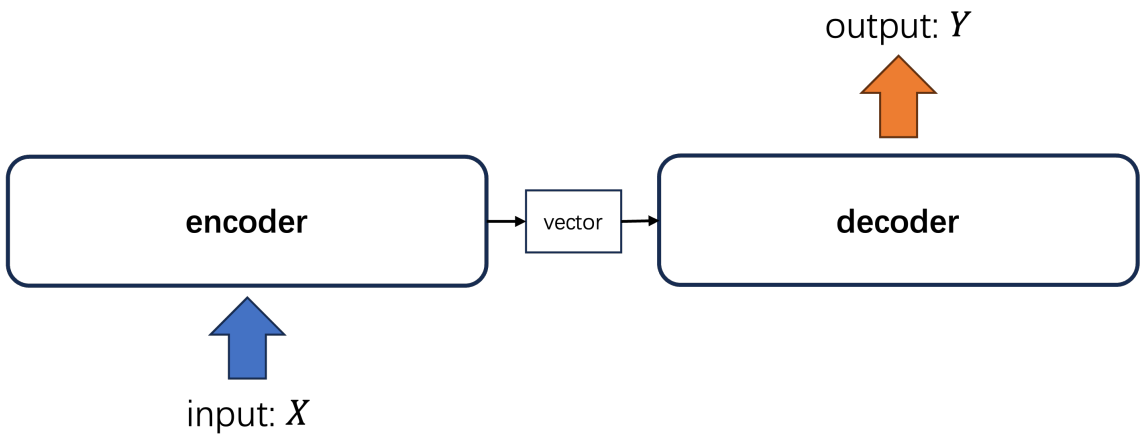
- Note:
  - Here, the alignment between X and Y is mostly explicit, e.g., a one-to-one correspondence (x_i is the current stock information, and y_i is the predicted stock prices of tomorrow).
  - X and Y are of the same lenth.

Deep-learning based method

Sequence to Sequence Learning with Neural Networks. Sutskever et al., 2014.

The main idea of Sutskever et al., (2014):

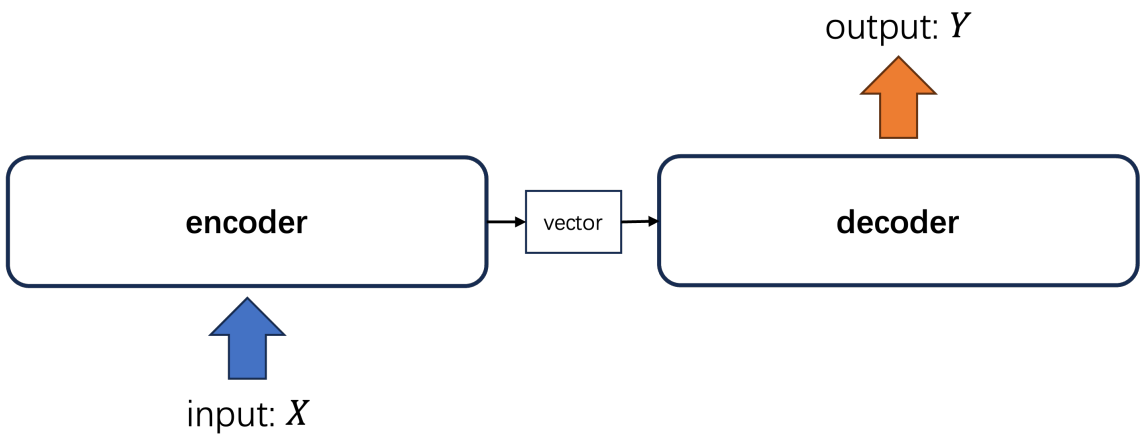
- Introduce the encoder-decoder architecture:
  - Encoder: convert the input sequence into a vector
  - Decoder: generate output sequence progressively from the vector.
- Aim to map two sequences of different lengths and without a fixed one-to-one alignment (summarisation, translation, etc).



Deep-learning based method

Method of Sutskever et al., (2014):

- Input: an input sequence (e.g., a document) $X = x_1, x_2, \ldots, x_m$. x_i is the i-th token in X and m is the length of X.
- Output: an output sequence (e.g., a summary) $Y = y_1, y_2, \ldots y_n$. y_i is the i-th token in Y and n is the length of Y.
- Training corpus: $D = \{(X_i, Y_i)\}_{i=1}^{l}$.
- Modeling: $P(Y|X)$.



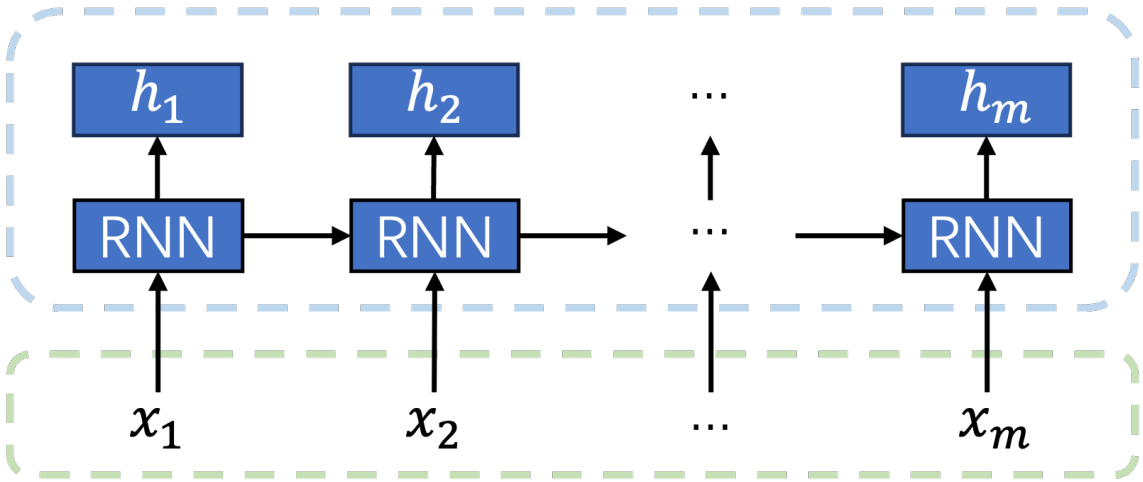
Deep-learning based method

Let's quickly recall how to encode text using RNN:

- Encode the text using a uni-directional RNN (LSTM):

$$h_{i+1}^{en} = \text{RNN}(h_i^{en}, \text{emb}^{en}(x_{i+1})) \tag{10}$$

  - emb is the embedding function, which maps the token x_i into an embedding vector. emb can be randomly initialized or using a pre-trained embedding, and can be trained together with RNN weights.
  - h_i^{en} is the hidden state at time step i. h_0^{en} is the initial state, which can be zero-initialized but usually initialized using $\sum_{i=1}^{m} emb(x_i)$.
  - We take the final hidden state as the sentence representation: $h^{en} = h_m^{en}$.



Deep-learning based method

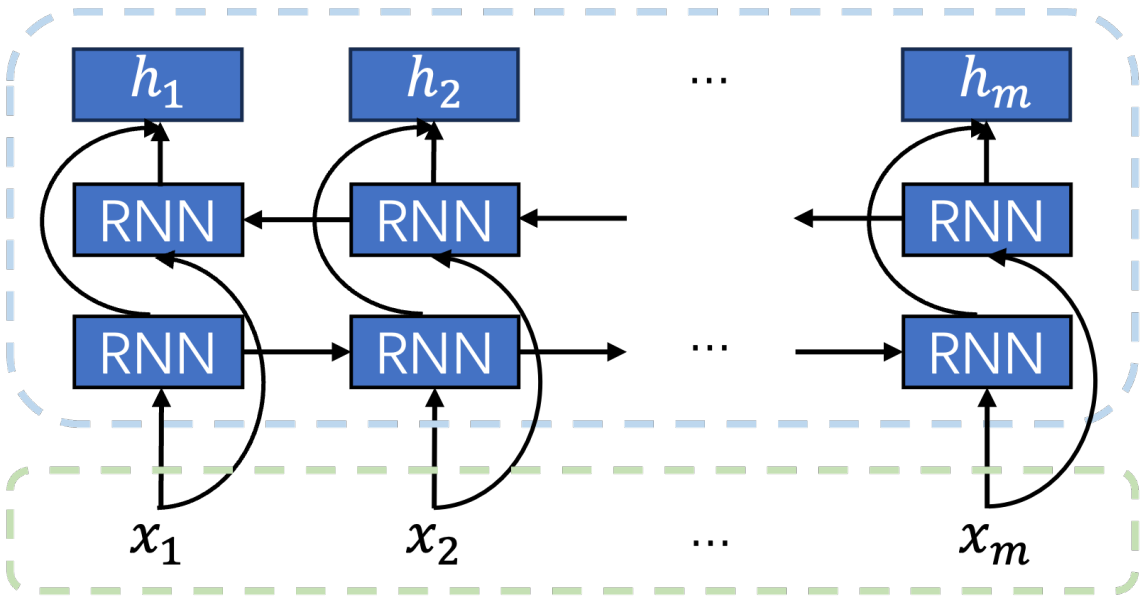
Let's quickly recall how to encode text using RNN:

- Or we can use other neural networks, such as Bi-RNN:

$$\overleftarrow{\mathbf{h}_i^{en}} = \text{RNN}(\overleftarrow{\mathbf{h}_{i+1}^{en}}, \text{emb}^{en}(x_i)) \tag{11}$$

$$\overrightarrow{\mathbf{h}_i^{en}} = \text{RNN}(\overrightarrow{\mathbf{h}_{i-1}^{en}}, \text{emb}^{en}(x_i)) \tag{12}$$

$$\mathbf{h}_m^{en} = [\overrightarrow{\mathbf{h}_m^{en}}; \overleftarrow{\mathbf{h}_1^{en}}] \tag{13}$$



Deep-learning based method

How to decode a text from a vector?

- Now, we already have a hidden state vector $\mathbf{h}_m^{en}$ that can represent the input sequence, and our goal is to model $P(Y|X)$:

$$P(Y|X) = P(y_1, y_2, \ldots, y_n|x_1, x_2, \ldots, x_m) \tag{14}$$

Deep-learning based method

How to decode a text from a vector?

- Now, we already have a hidden state vector $\mathbf{h}_m^{en}$ that can represent the input sequence, and our goal is to model $P(Y|X)$:

$$P(Y|X) = P(y_1, y_2, \ldots, y_n | x_1, x_2, \ldots, x_m) \tag{14}$$

- Directly generating the entire Y at one time is difficult.
  - Y is sparse.
  - For sentences, there are dependencies between tokens.

Deep-learning based method

How to decode a text from a vector?

- Now, we already have a hidden state vector $\mathbf{h}_m^{en}$ that can represent the input sequence, and our goal is to model $P(Y|X)$:

$$P(Y|X) = P(y_1, y_2, \ldots, y_n | x_1, x_2, \ldots, x_m) \tag{14}$$

- Directly generating the entire Y at one time is difficult.
  - Y is sparse.
  - For sentences, there are dependencies between tokens.
- How to map an X of m token to a Y of n tokens?
  - Can we predict m first before generating Y?

Deep-learning based method

How to decode a text from a vector?

- Now, we already have a hidden state vector $\mathbf{h}_m^{en}$ that can represent the input sequence, and our goal is to model $P(Y|X)$:

$$P(Y|X) = P(y_1, y_2, \ldots, y_n | x_1, x_2, \ldots, x_m) \tag{14}$$

- Directly generating the entire Y at one time is difficult.
  - Y is sparse.
  - For sentences, there are dependencies between tokens.
- How to map an X of m token to a Y of n tokens?
  - Can we predict m first before generating Y? Even humans cannot do it accurately.

Deep-learning based method

How to decode a text from a vector?

- Now, we already have a hidden state vector $\mathbf{h}_m^{en}$ that can represent the input sequence, and our goal is to model $P(Y|X)$:

$$P(Y|X) = P(y_1, y_2, \ldots, y_n | x_1, x_2, \ldots, x_m) \tag{14}$$

- Directly generating the entire Y at one time is difficult.
  - Y is sparse.
  - For sentences, there are dependencies between tokens.
- How to map an X of m token to a Y of n tokens?
  - Can we predict m first before generating Y? Even humans cannot do it accurately.
  - *And do we really care about m?*

Deep-learning based method

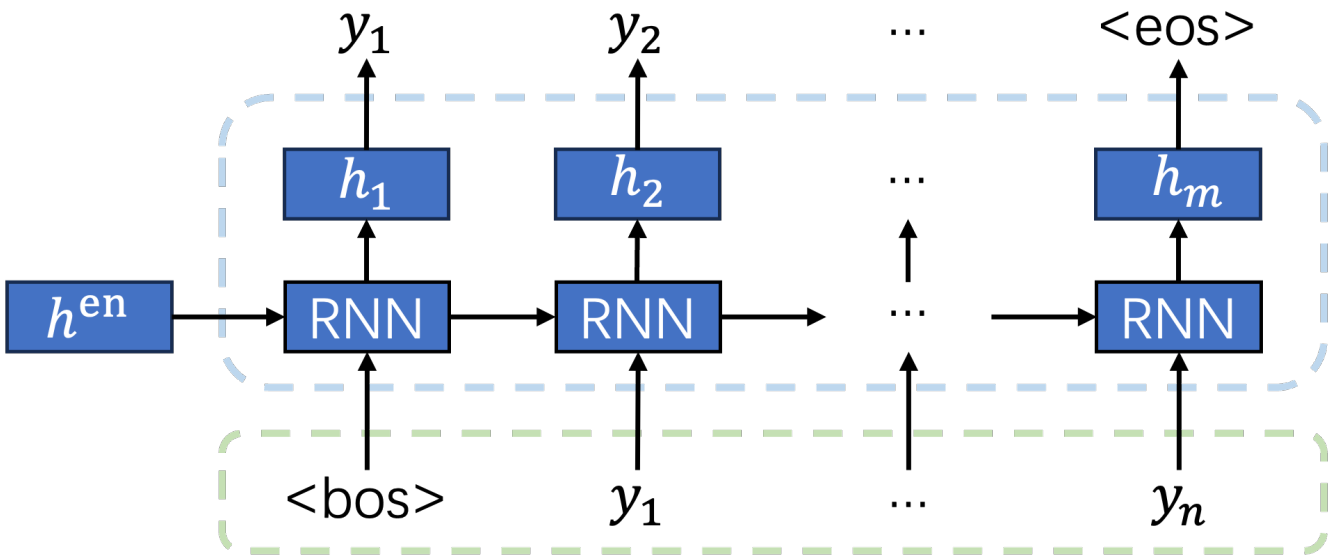
How to decode a text from a vector?

- Now, we already have a hidden state vector $\mathbf{h}_m^{en}$ that can represent the input sequence, and our goal is to model $P(Y|X)$:

$$P(Y|X) = P(y_1, y_2, \ldots, y_n|x_1, x_2, \ldots, x_m) \tag{14}$$

- Directly generating the entire Y at one time is difficult.
  - Y is sparse.
  - For sentences, there are dependencies between tokens.
- How to map an X of m token to a Y of n tokens?
  - Can we predict m first before generating Y? Even humans cannot do it accurately.
  - *And do we really care about m?* We only care when and how the decoder should stop decoding.

Deep-learning based method



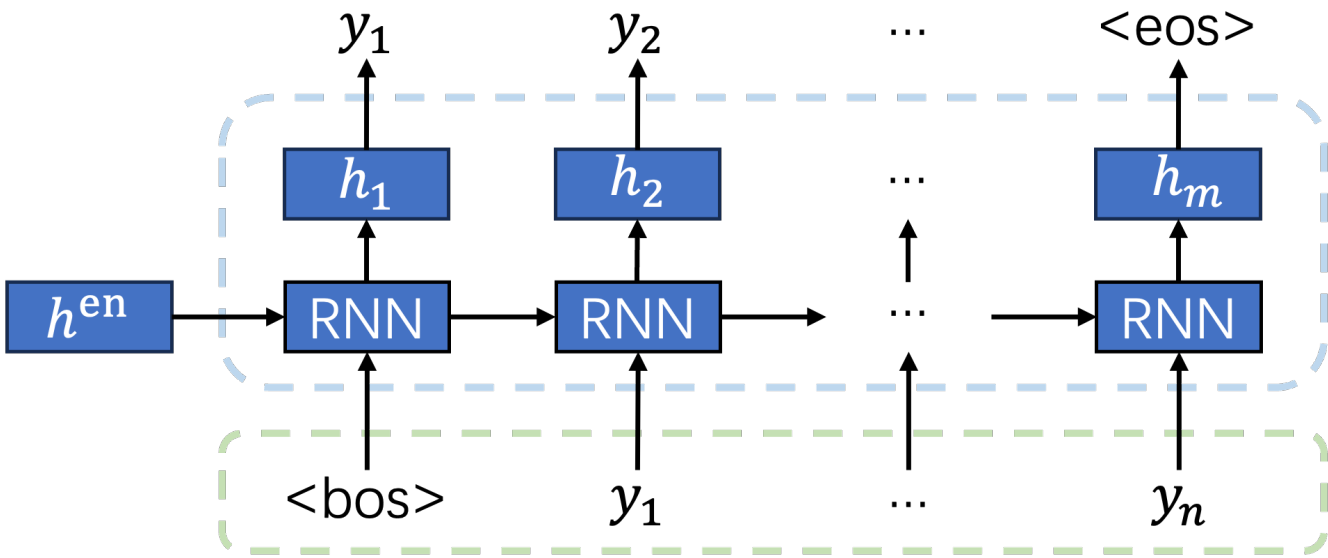
At decoding time step i:

- Compute the hidden state of the decoder:

$$\mathbf{h}_i^{de} = \text{RNN}(\mathbf{h}_{i-1}^{de}, \text{emb}^{de}(y_{i-1})) \tag{15}$$

- Can emb^{de} be same as emb^{en}?
- Note at the beginning:
  - y_0 is a special token <bos>.
  - Initialize the hidden state using the encoder hidden state, i.e., $\mathbf{h}_0^{de} = \mathbf{h}^{en}$.

Deep-learning based method



At decoding time step i:

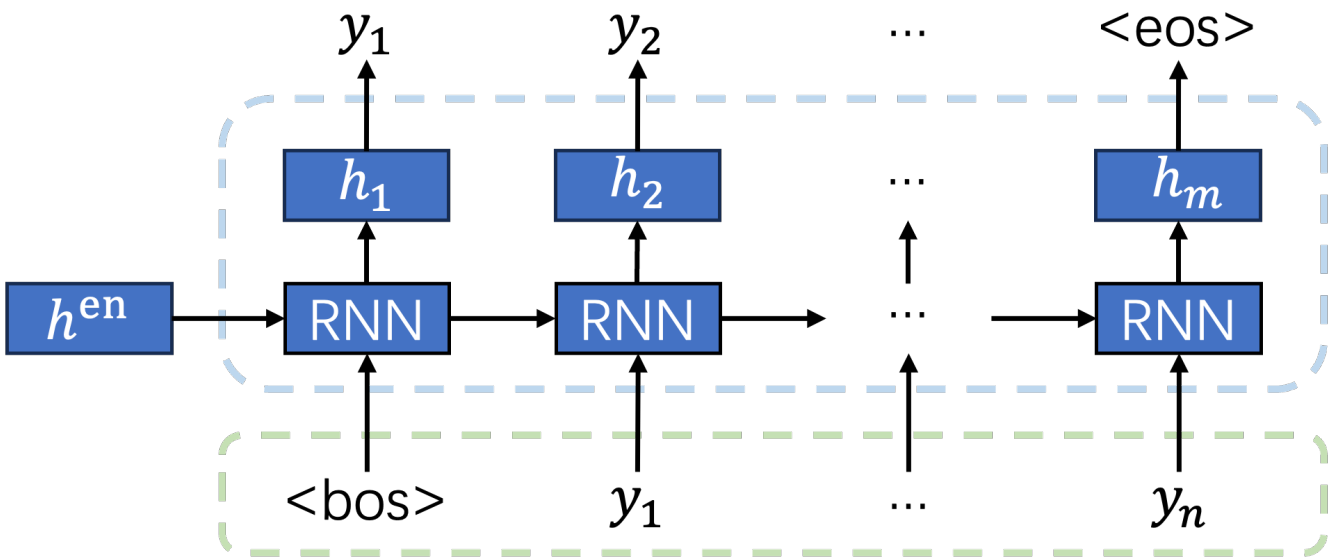
- Map the hidden state $\mathbf{h}_i^{de}$ to output probabilities of tokens:

$$o_i = W\mathbf{h}_i^{de} + b \tag{16}$$

$$p_i = \texttt{softmax}(o_i) \tag{17}$$

- Note:
  - o_i is the unnormalized score over the decoder vocabulary.
  - p_i is the normalized probability distribution over the vocabulary.

Deep-learning based method

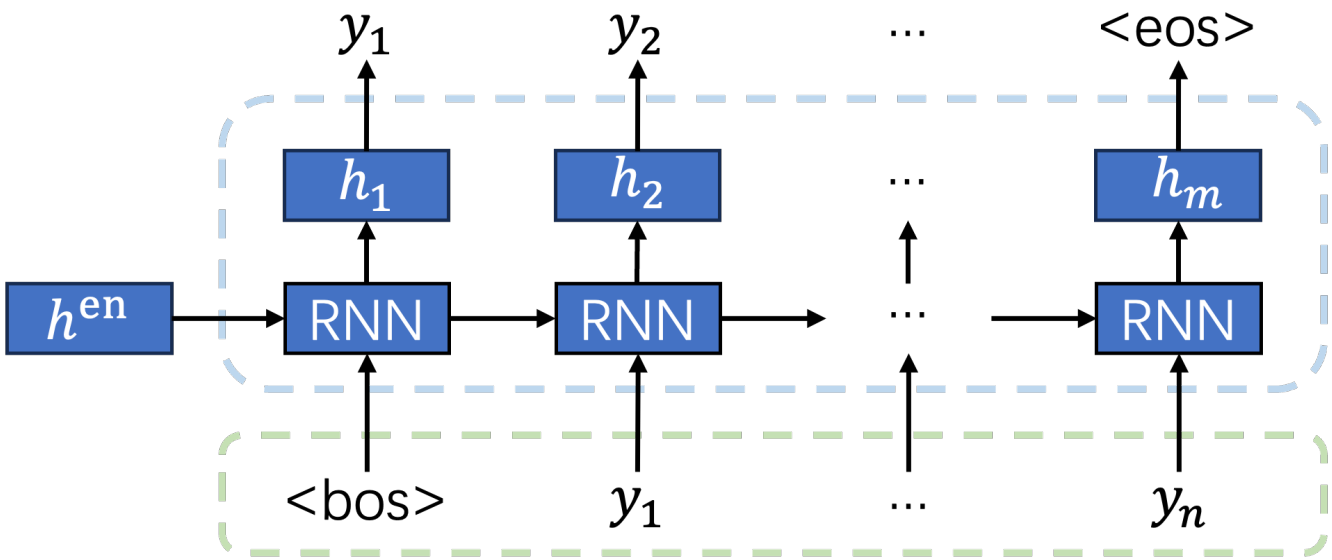


At decoding time step i:

- Finally we choose the output token of the highest probability:

$$y_i = \text{argmax}(p_i) \tag{18}$$

Deep-learning based method



At decoding time step i:

- Finally we choose the output token of the highest probability:

$$y_i = \text{argmax}(p_i) \tag{18}$$

Stop decoding when <eos> is predicted.

Deep-learning based method

Training:

- Given the input X and output Y, we can optimize the model in a **teacher-forcing** manner:

$$P(Y|X) = P(y_1, \ldots, y_n|x_1, \ldots, x_m) \tag{19}$$

$$= \prod_{i=1}^{n} P(y_i|X, y_1, \ldots, y_{i-1}) \tag{20}$$

  - At each time step, instead of using the model's prediction $\hat{y}_{i-1}$ from the previous step as input, we provide the actual ground truth y_{i-1}.
  - More stable and avoid errors propagation from previous predictions.

- For each data, we use the negative log-likelihood to calculate the loss:

$$\mathcal{L} = -\sum_{i=1}^{n} \log P(y_i|X, y_1, \ldots, y_{i-1}) \tag{21}$$

Reading

- Part II: NLP Applications. D Jurafsky and J Martin. *Speech and Language Processing*
  `web.stanford.edu/~jurafsky/slp3/3.pdf`
- Chapter 7: Neural Networks and Neural Language Models. D Jurafsky and J Martin. *Speech and Language Processing*
  `web.stanford.edu/~jurafsky/slp3/7.pdf`