Neural Approaches to Natural Language Generation

Claire Gardent

Loria
Laboratoire lorrain de recherche en informatique et ses applications

cnrs

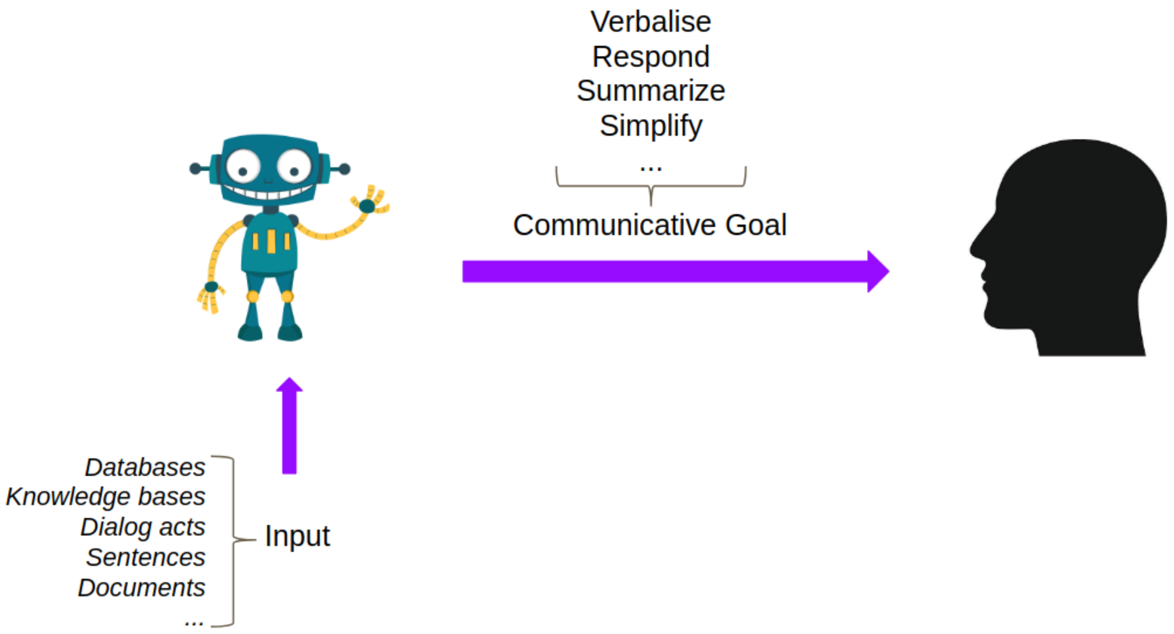
UNIVERSITÉ DE LORRAINE

ENS Lyon, 25 May 2021

Outline

- What is NLG ?
- A brief introduction to Neural Networks
- Neural NLG, the Encoder-Decoder Framework
- Some examples
  - Summarising multiple documents
  - Generating dialog responses using external knowledge
  - Converting Meaning Representations into 21 languages

What is NLG ?



Natural Language Generation (NLG) generates text from some ***input*** to satisfy a given ***communicative goal***

Input and goal define different types of text production tasks/applications e.g., summarisation, verbalisation of knowledge bases, ...

Input

Meaning Representations

- Abstract Meaning Representations (AMR), Logical Formulae (First Order Logic, Description Logic, SQL)

Data

- Knowledge Bases, Data Bases, Numerical data from signal processing, Images, ...

Text

- Multiple or single document, Dialog or Discourse

Communicative Goals

Describing, Verbalising

- a KB fragment, an entity in a DB, an image, a video

Summarising

- A text, Several texts, The content of a KB

Simplifying

- For children, foreigners, disabled people

Paraphrasing, Reformulating

- Expert/non expert

NLG Applications

Data-to-Text

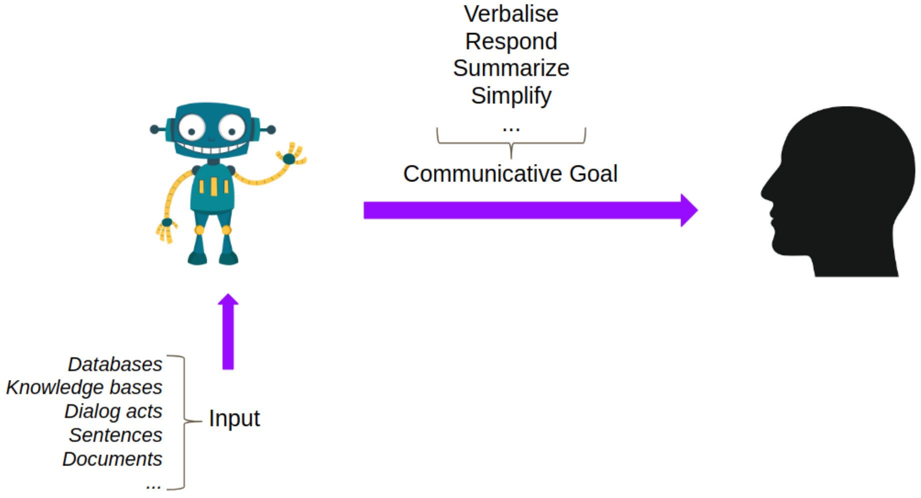
- Verbalise, summarise KB, DB, Numerical Data

Text-to-Text

- Summarise, simplify, paraphrase , (translate) one or more document

MR-to-Text

- Convert a meaning representation to its natural language equivalent



Neural Networks

Neuron

A neuron computes an ***activation value*** by applying an activation function to the weighted sum of its inputs.

$$y = g(\sum_{i=1}^{n} w_i \, x_i)$$

where $x_1, \ldots, x_n$ are the input values, $w_0, w_1, \ldots, w_n$, the weights and g is an activation function.

Activation Functions

Logistic, sigmoid, $f(x) \in [0, 1]$

$$f(x) = \frac{1}{1 + e^{-x}}$$

Tanh (Hyperbolic Tangent), $f(x) \in [-1, 1]$

$$tanh(x) = \frac{2}{1 + e^{-2x}} - 1$$

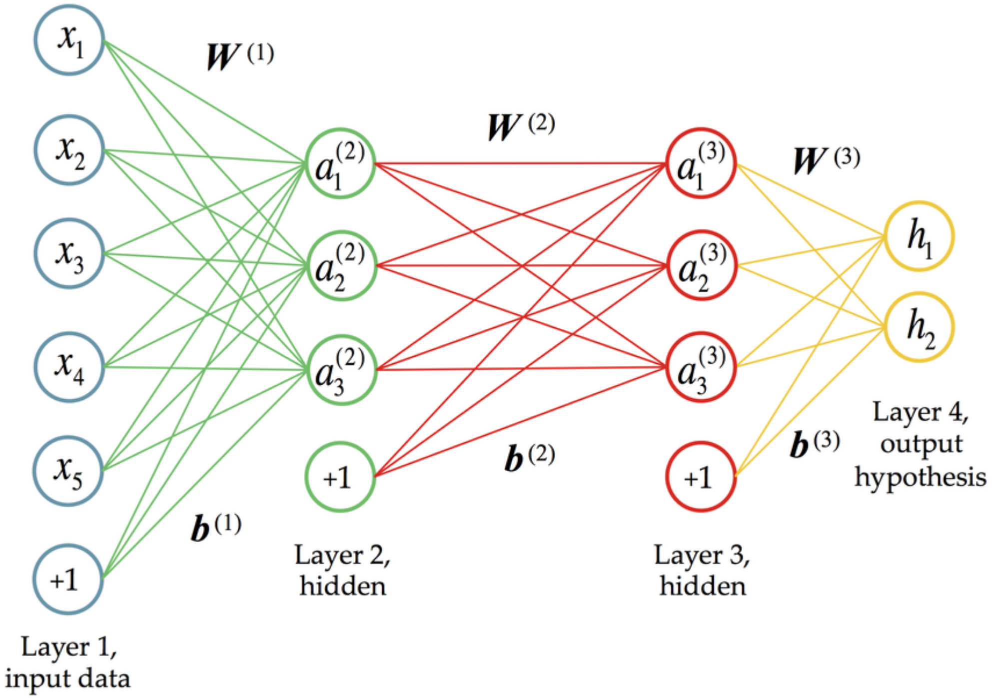
ReLU: Rectified Linear Unit, $f(x) \in [0, \infty]$

$$ReLu(x) = 0 \; if \; x < 0 \; else \; x$$

Blog: Complete Guide of Activation Functions . Another one

Neural Network Architecture

- Neurons are organized in **layers**
- The layers between input and output are referred to as ***hidden layers***
- Each neuron produces an activation value which is passed on as input to other neurons.
- The weights are ***learned*** using back propagation and stochastic gradient descent



Types of Neural Networks

Convolutional Neural Networks (CNN). Convolutions compute activations values over different, possibly overlapping, part of the input. Deep networks with higher level layers capturing higher level features of the input. Common in computer vision.

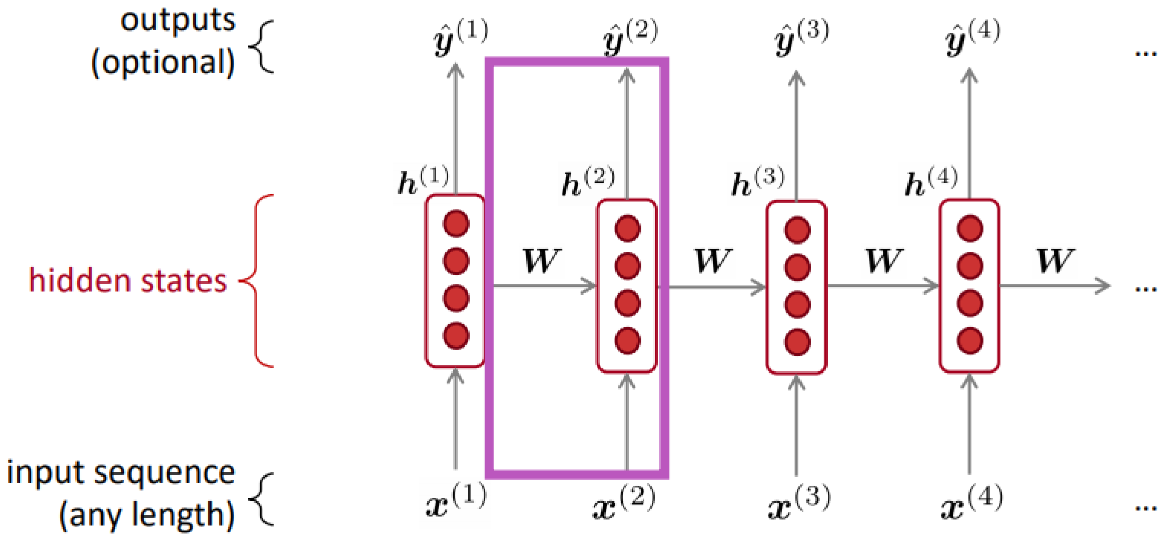
Feed-Forward Networks (FFN, MLP). Fully connected. All the neurons of layer L connected to all neurons of layer L + 1. Commonly used for classification

Recurrent Neural Networks (RNN). Recurs over the input processing it sequentially, one input token at a time. Common in Natural Language Processing

Transformers. Deep + Attention. Currently, the standard in NLP

Recurrent Neural Networks

Process the input sequentially, one token at a time



RNN Hidden State

At each time step, an RNN computes a new ***hidden state*** based on the previous hidden state and the current input.

$$h_t = tanh(W_h \; h_{t-1} + W_e \; x_t)$$

- h_{t-1}, the hidden state is a vector of real values.
- W_h, W_e are matrices of weights between layers
- x_t, the input is represented by a vector of real values.

Encoding Input Tokens as Vectors

Each token is mapped to an ***index*** and each index to a ***1-hot vector*** of size the size of the vocabulary. The ***vocabulary*** is the set of distinct tokens in the input data.

this $\to 1 \to \langle 1, 0, 0, \ldots, 0\rangle$

cat $\to 2 \to \langle 0, 1, 0, \ldots, 0\rangle$

dog $\to 3 \to \langle 0, 0, 1, \ldots, 0\rangle$

RNN Output

Optionally, at each time step, an RNN can compute an ***output*** based on the newly created hidden state. E.g.,

$$y_t = softmax(W_y \, h_t)$$

with

$$softmax(\overrightarrow{z}) = \frac{e^{z_i}}{\sum_{j=1}^{k} e^{z_j}}$$

By applying the softmax function to the hidden state, the RNN can output at each time step, a ***probability distribution*** .

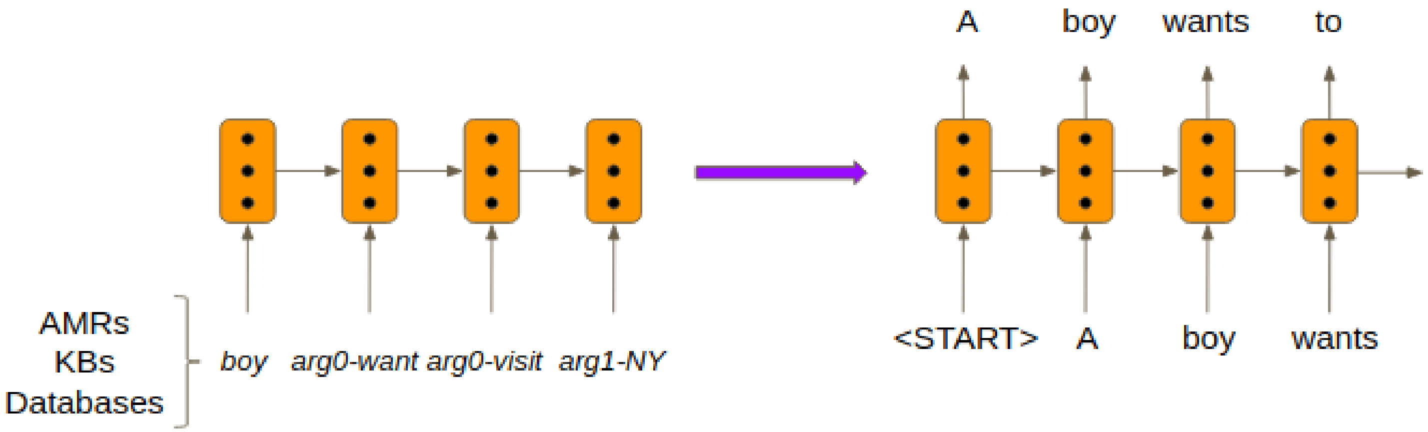
Typically, this is used to compute a probability distribution over a set of pre-defined classes (e.g., the syntactic category of a word) or over the vocabulary (predicting the most likely word given the previously processed tokens).

Neural NLG

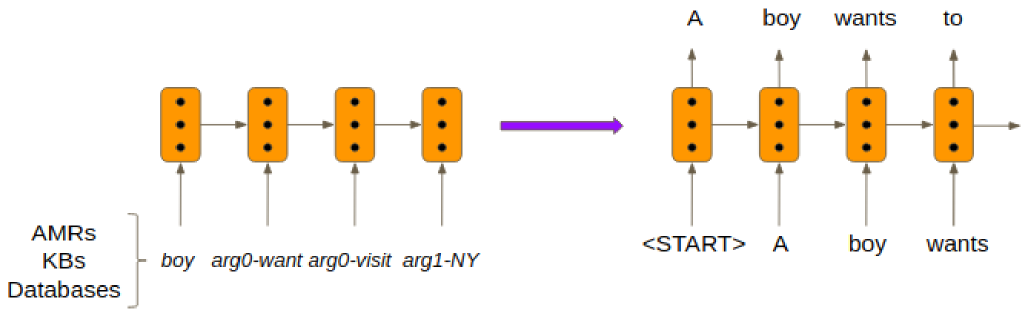
The Encoder-Decoder Framework

The encoder-decoder model provides a uniform framework for all NLG tasks (Data-to-Text, Text-to-Text, MR-to-Text)

- The ***encoder*** is a network which is applied to the input and outputs a vector representing this input. It can be a CNN, an RNN or a Transformer
- The ***decoder*** is a network which outputs a sequence of word one token at a time. It is either an RNN or a Transformer.



RNN Encoder-Decoder



RNN-Encoder

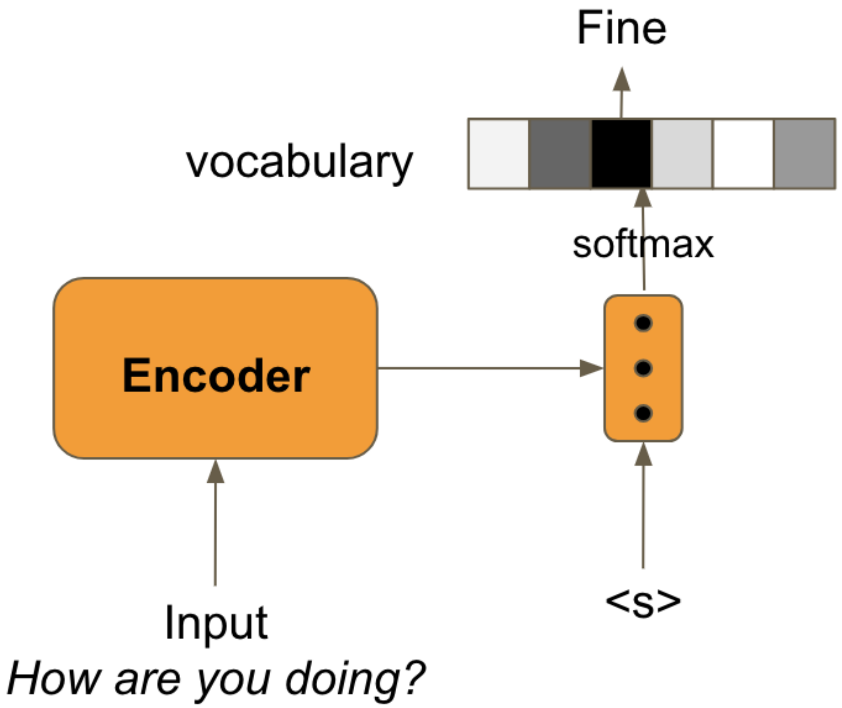
The vector representation of the input is that ***last hidden state*** output after processing the input.

RNN-Decoder

At each time step,

- the RNN outputs a ***probability distribution*** over the vocabulary
- a word w is output by sampling from this probability distribution
- w is input to the next recurrence/time step

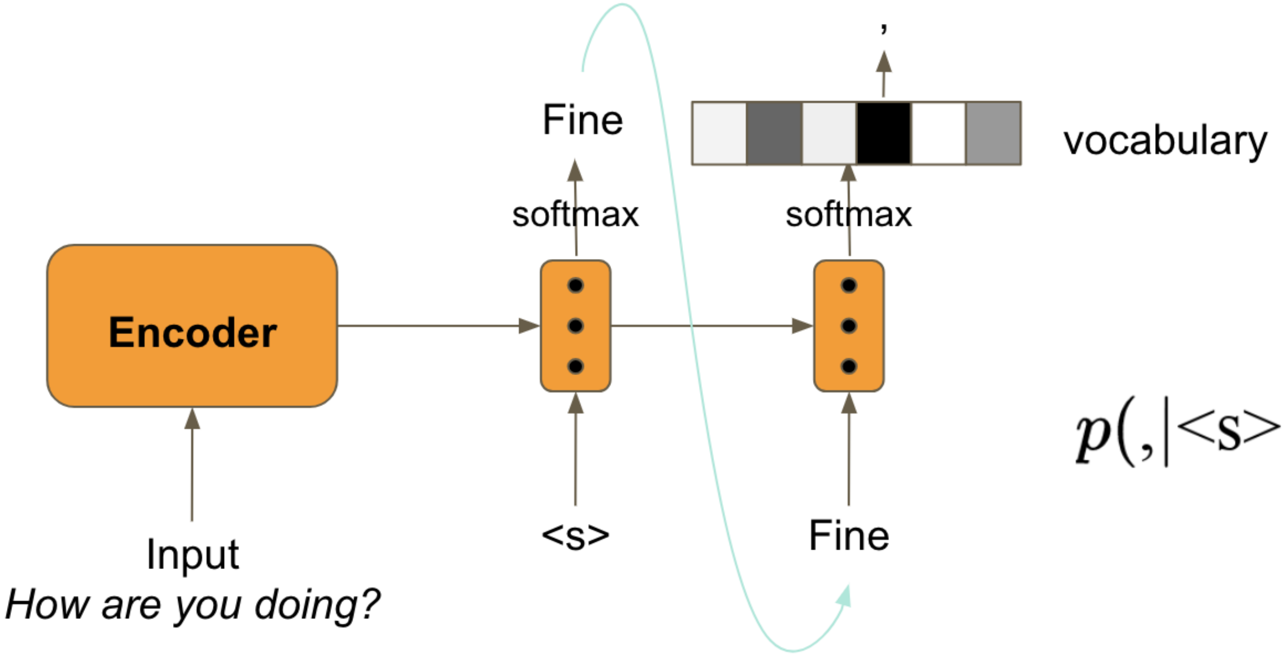
Generating Text using an RNN



$$p(\mathrm{Fine} \mid <\mathrm{s}>, \text{How are you doing?})$$

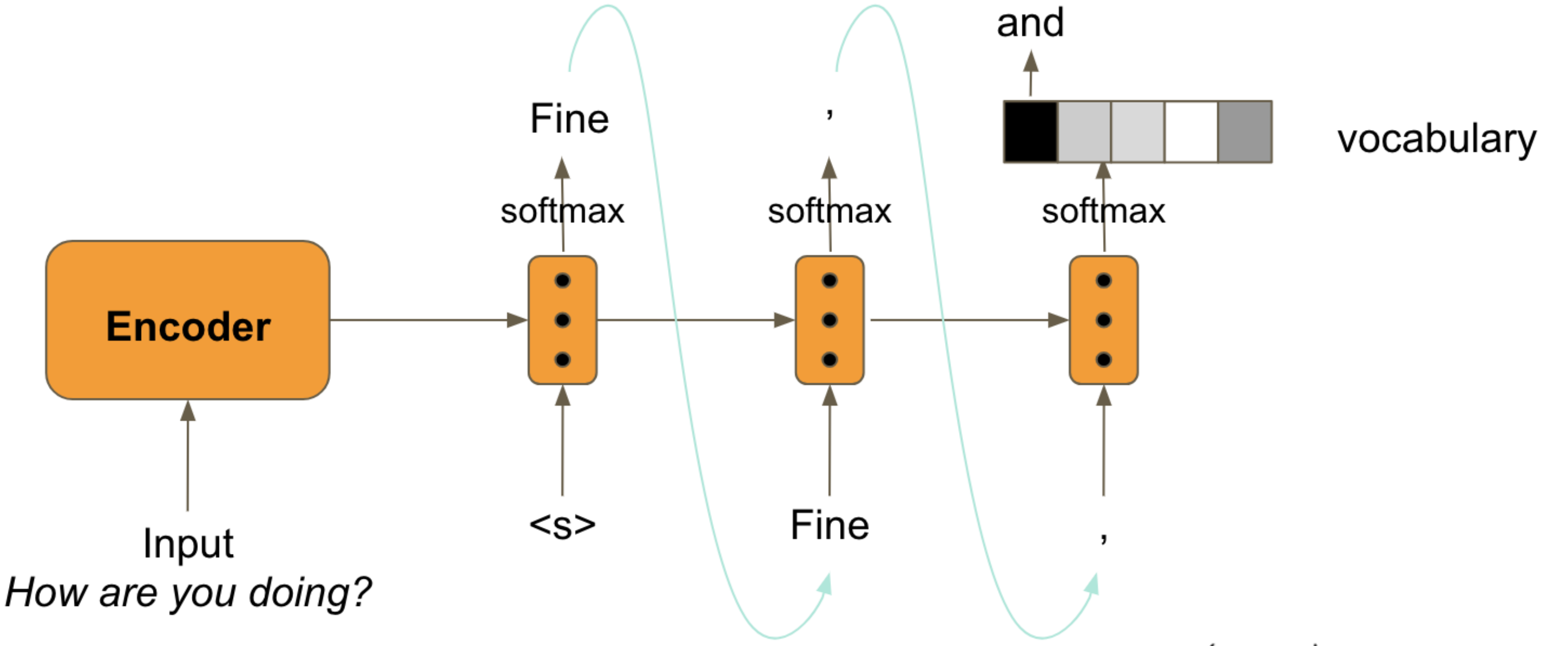
Conditional Generation

Generating Text using an RNN



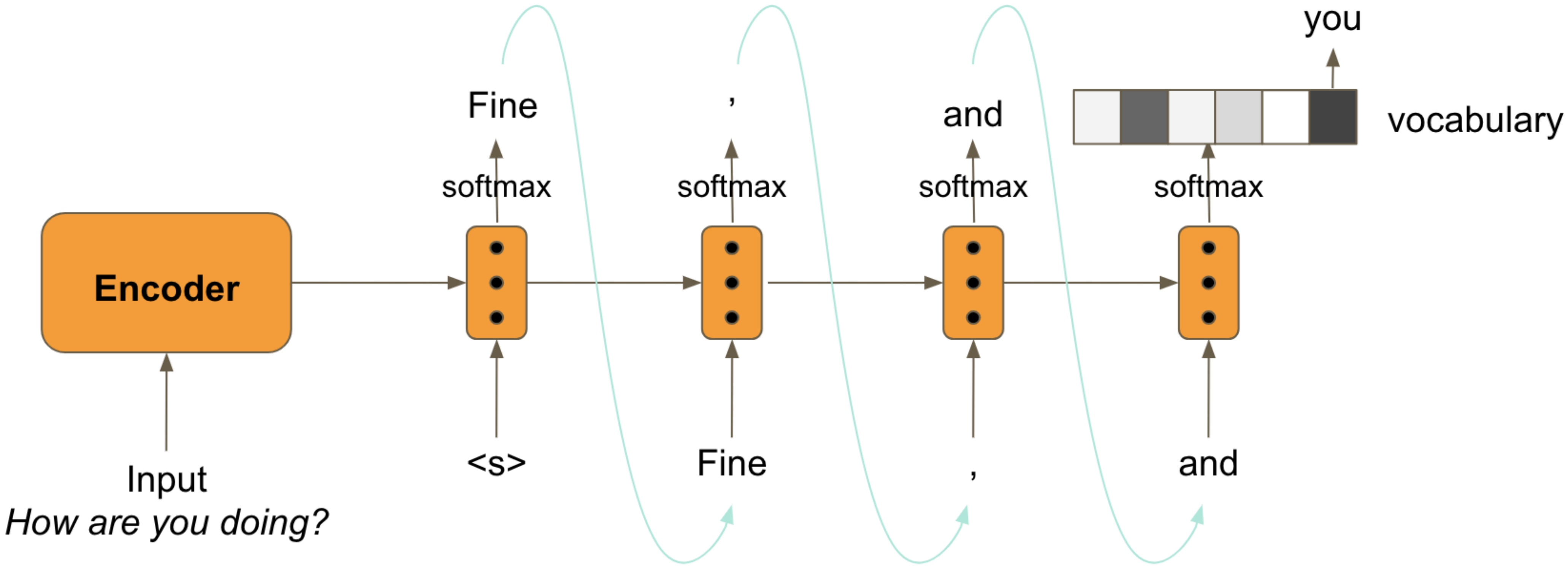
$$p(,|\text{<s> Fine, How are you doing?})$$

Generating Text using an RNN

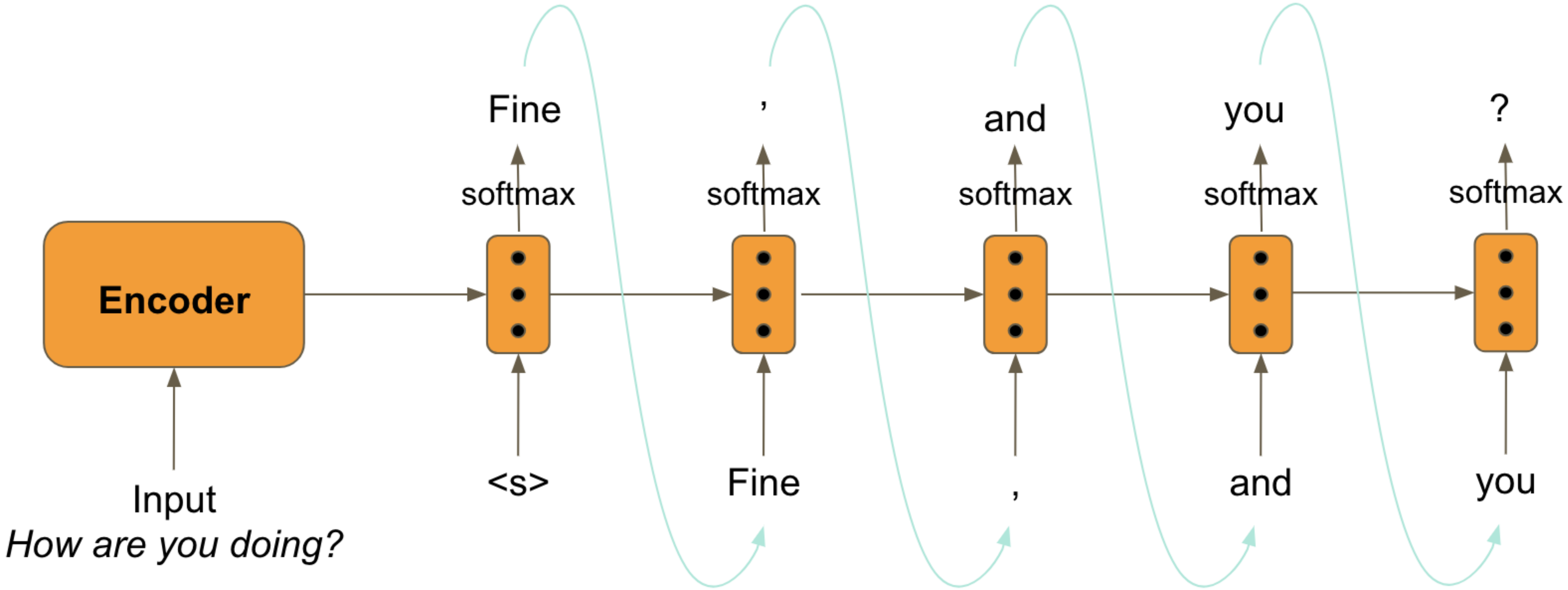


$$p(\text{and} \mid <\text{s}> \text{Fine,; How are you doing?})$$

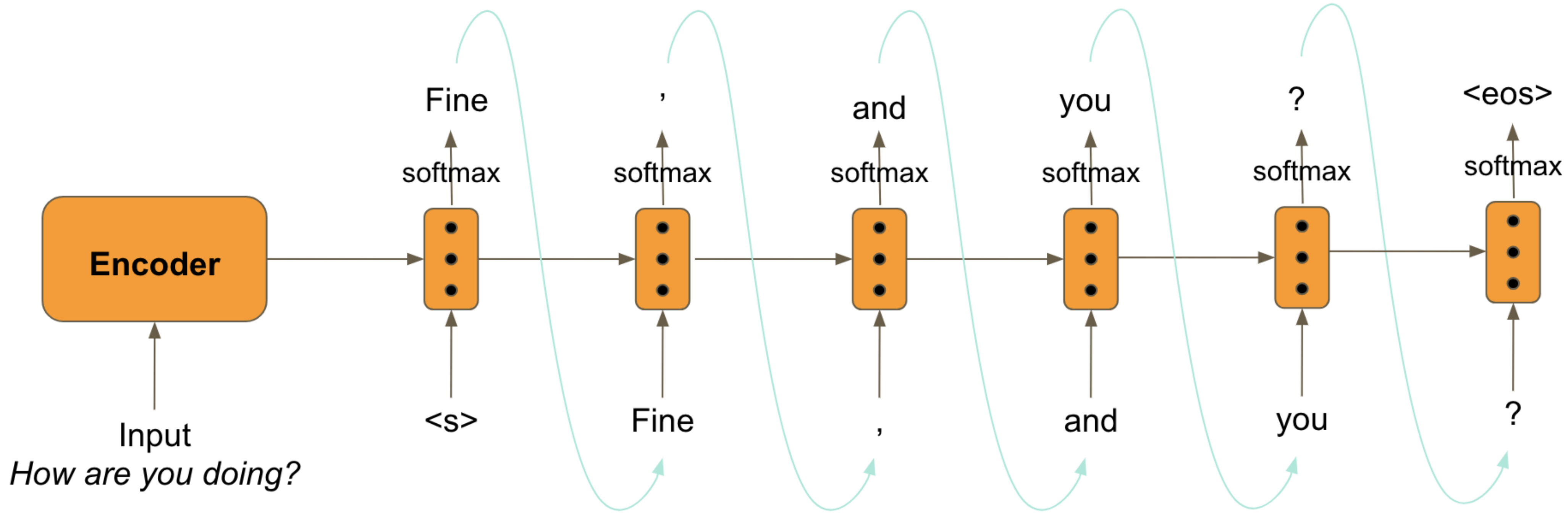
Generating Text using an RNN



Generating Text using an RNN

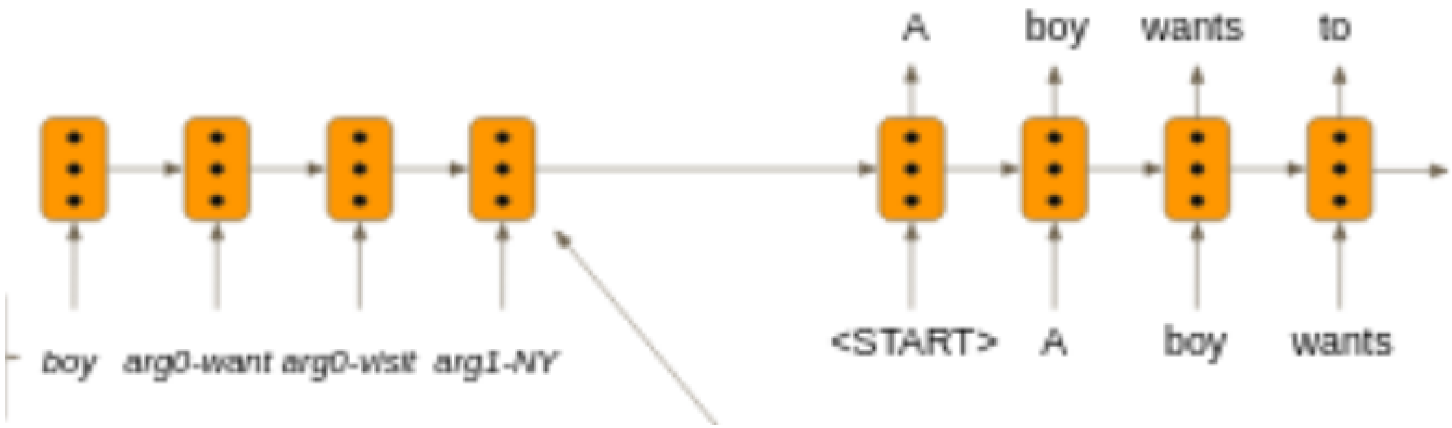


Generating Text using an RNN



Attention and Transformers

Standard RNN Decoding



- The input is compressed into a ***fixed-length vector***
- Performance decreases with the length of the input

Sutskever et al. 2014

RNN Decoder with Attention

Input

- the previous state s_{t-1}, the previously generated token y_{t-1} and ***a context vector*** c_t

Context vector

- depends on the previous hidden state and therefore ***changes at each step***
- indicates ***which part of the input is most relevant*** to the decoding step

Attention

Attention is a way to obtain a fixed-size representation

- of an arbitrary set of representations (the ***values***)
- dependent on some other representation (the ***query***)

Decoder with attention

- Query = previous decoder state
- Values = encoder hidden states

*The **context vector** creates a representation of the **input** which is dependent on the previous **decoder state***

Encoder-Decoder Attention

Scores are computed between each encoder hidden state and the previous hidden state which are then turned into a probability distribution

$$\overrightarrow{e^t} = [s_{t-1}{}^\top h_1 \dots s_{t-1}{}^\top h_n]$$

$$\alpha^t = softmax(\overrightarrow{e^t})$$

The ***context vector*** is the weighted sum of the encoder hidden states

$$c^t = \sum_{i=1}^{n} \alpha_i^t h_i$$

The ***new decoder state*** is computed taking into account this context vector.

$$s^t = f(s^{t-1}, y^{t-1}, c^t)$$

Transformers

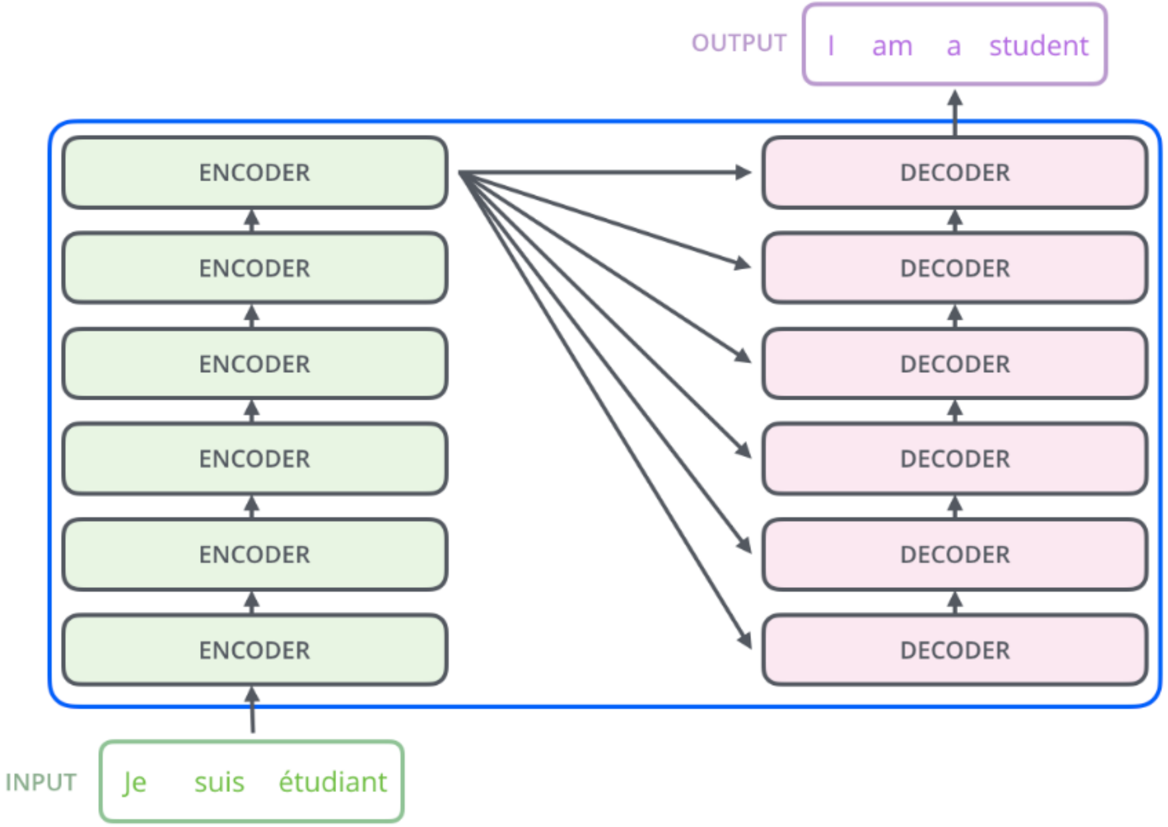
The Transformer

- ***Deep*** model
  Stack of Transformer Blocks
- ***Parallel*** processing of each token
  No recurrence
- Use ***(self-)attention*** to create word representations
- ***Multiple self-attention heads*** to create multiple views (features)

Vaswani et al. NIPS 2017

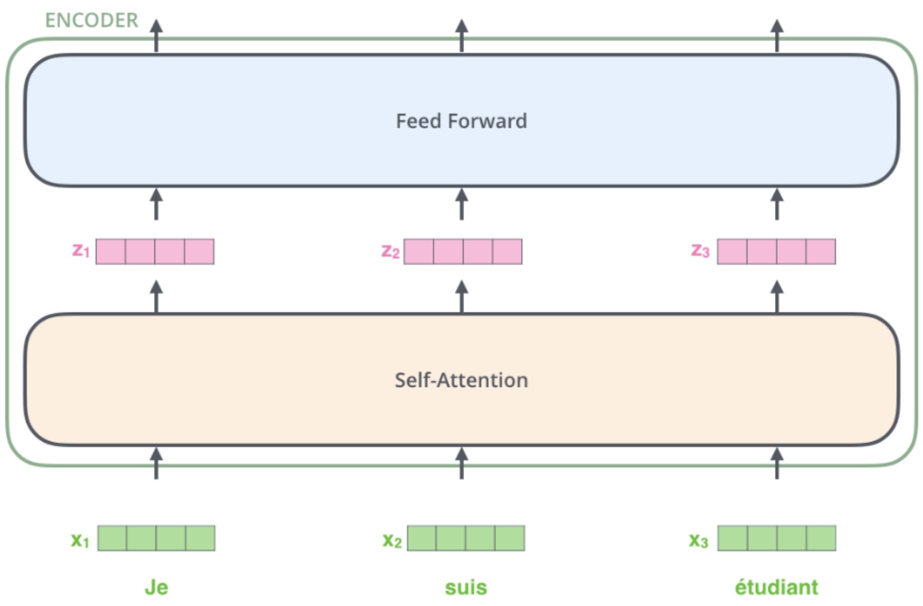
Deep Model

A Transformer Encoder-Decoder stacks multiple encoders / decoders



Transformer Encoder

- Tokens are processed in a parallel manner
- Positional embeddings are added to the input and summed with the word embeddings.
- Each encoder consists of two sub-layers Self-Attention and Feed-Forward



Self-Attention Layer

- ***Score each input word*** q_i ***against each other input words*** k_j
  Dot product(***Query, Key***)

- Divide the scores by the square root of the dimension of the key vectors (to have more stable gradients)

- Apply a Softmax layer

- Sum up the weighted value vectors. This produces the output the S-A layer for the input token at position j.

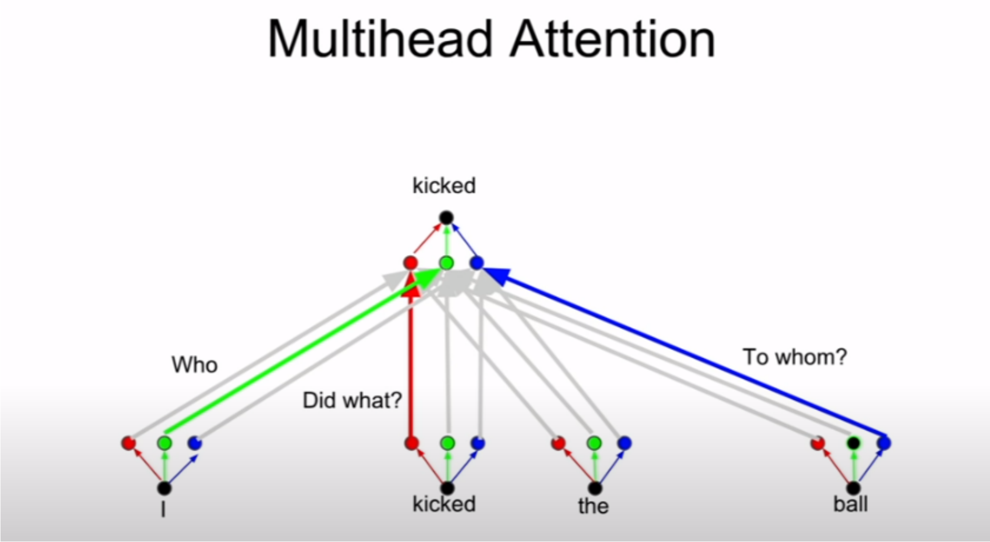
Computing the embedding of input word q_i

$$S = \langle q_i \bullet k1, q_i \bullet k2, \ldots, q_i \bullet kn \rangle$$

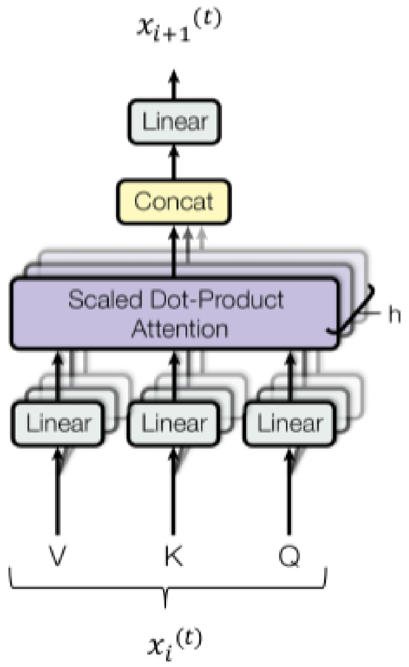
$$S' = \langle \frac{q_i \bullet k1}{\sqrt{Dk}}, \frac{q_i \bullet k2}{\sqrt{Dk}}, \ldots, \frac{q_i \bullet kn}{\sqrt{Dk}} \rangle$$

$$C = softmax(S') = \langle \alpha_i^1, \alpha_i^2, \ldots, \alpha_i^n \rangle$$

$$z_i = \sum_{j=1}^{n} \alpha_i^j \bullet v_j$$



Multiple heads help captures differents aspects (***features***) of the input tokens



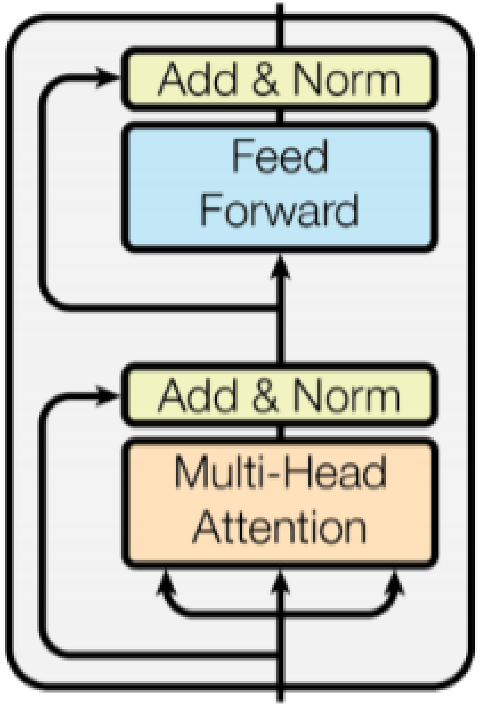
The output of the self-attention heads are concatenated and projected into a matrix of word vectors for the next layer

Residual Connections, Layer Normalization, and Feed Forward Network

Residual Connection The vector output by the self-attention layer is added to the original input embedding.

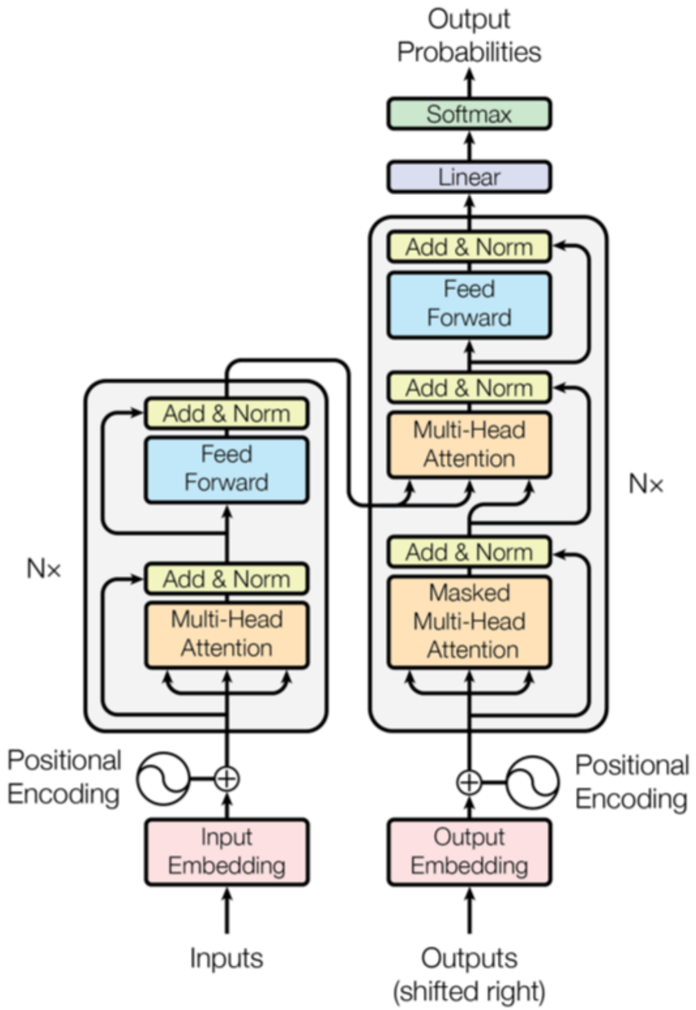
Layer Normalisation The output of the residual connection goes through a layer normalization.

Feed Forward Network The feed-forward network is a couple of linear layers with a ReLU activation in between.
The output of the FFN is added to the input of the pointwise feed-forward network and further normalized.



Decoding

- Each decoder consists of three sub-layers.
- The ***encoder-decoder attention layer*** helps the decoder focus on relevant parts of the input sentence (similar to what attention does in RNN models).
- Masked ***Multi-Head Attention***: attends over the words decoded so far
- The ***linear layer*** acts as a classifier.
- The ***softmax layer*** produces a probability distribution over the target vocabulary



Example NLG Models

Example NLG Models

- Dealing with very long input
- Generating text using external knowledge
- Multilingual NLG

Generating from Large Input

Generating from Large Input

Given a query Q:

- retrieve web documents D that satisfy that query.
- Generate text conditioning on Q and D

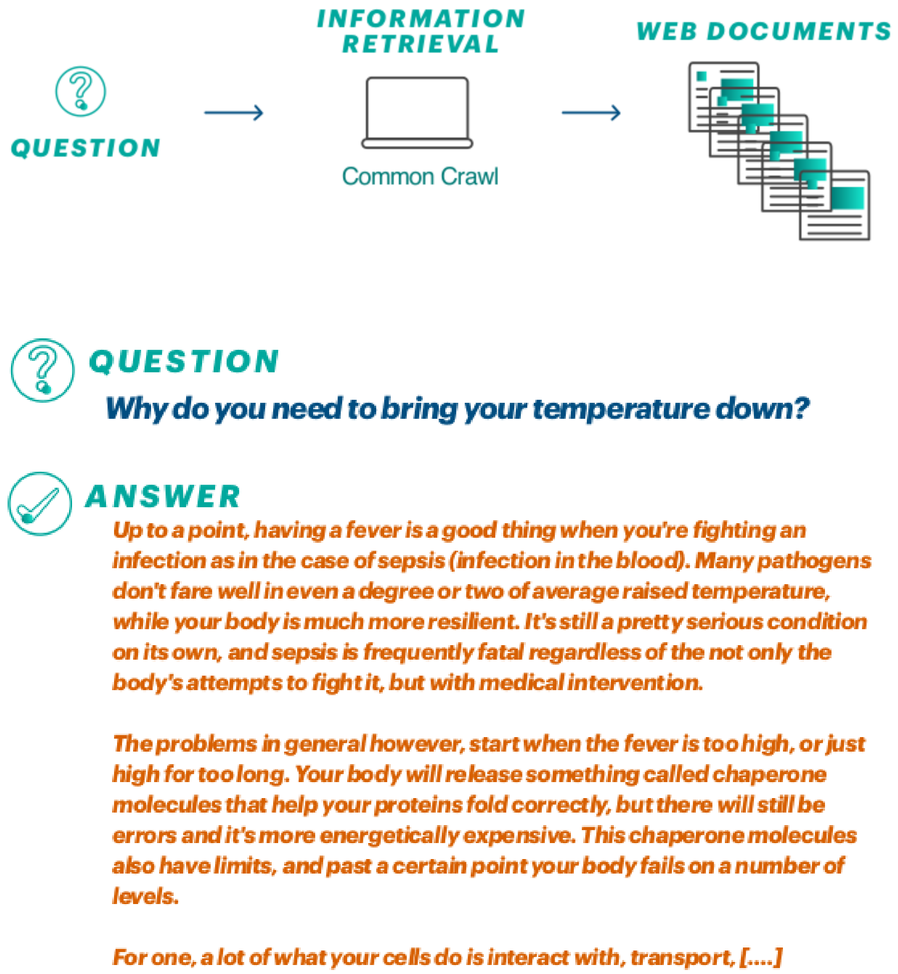
Question Answering

- Generate the answer to a question

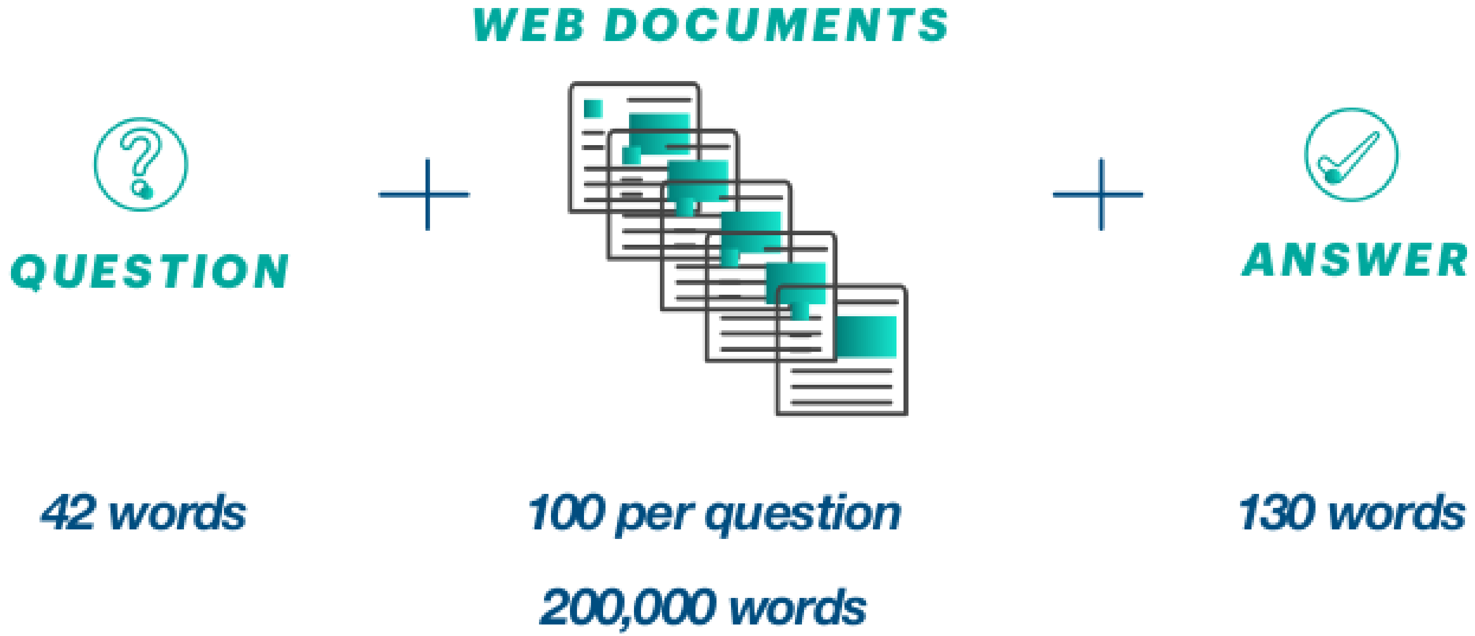
Multi-Document Summarisation

- Generate a biography

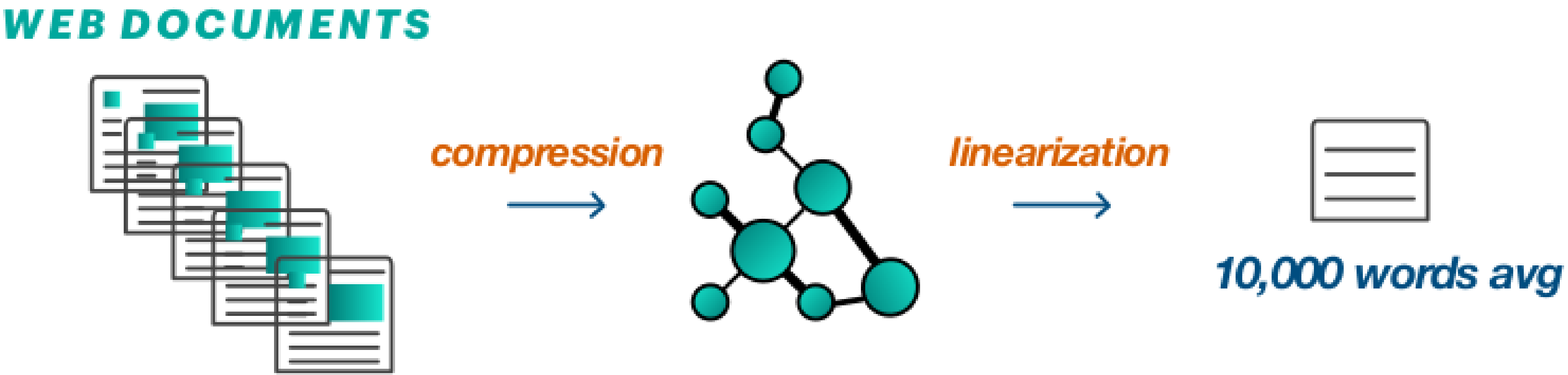
Fan et al. 2019



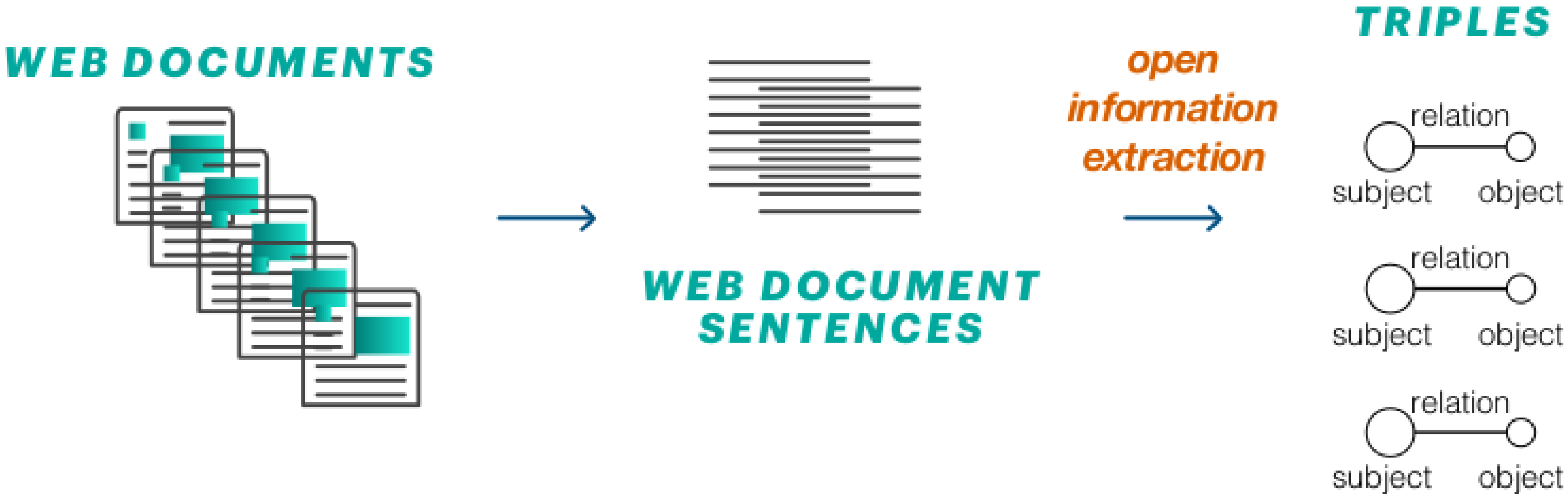
Long Form Input

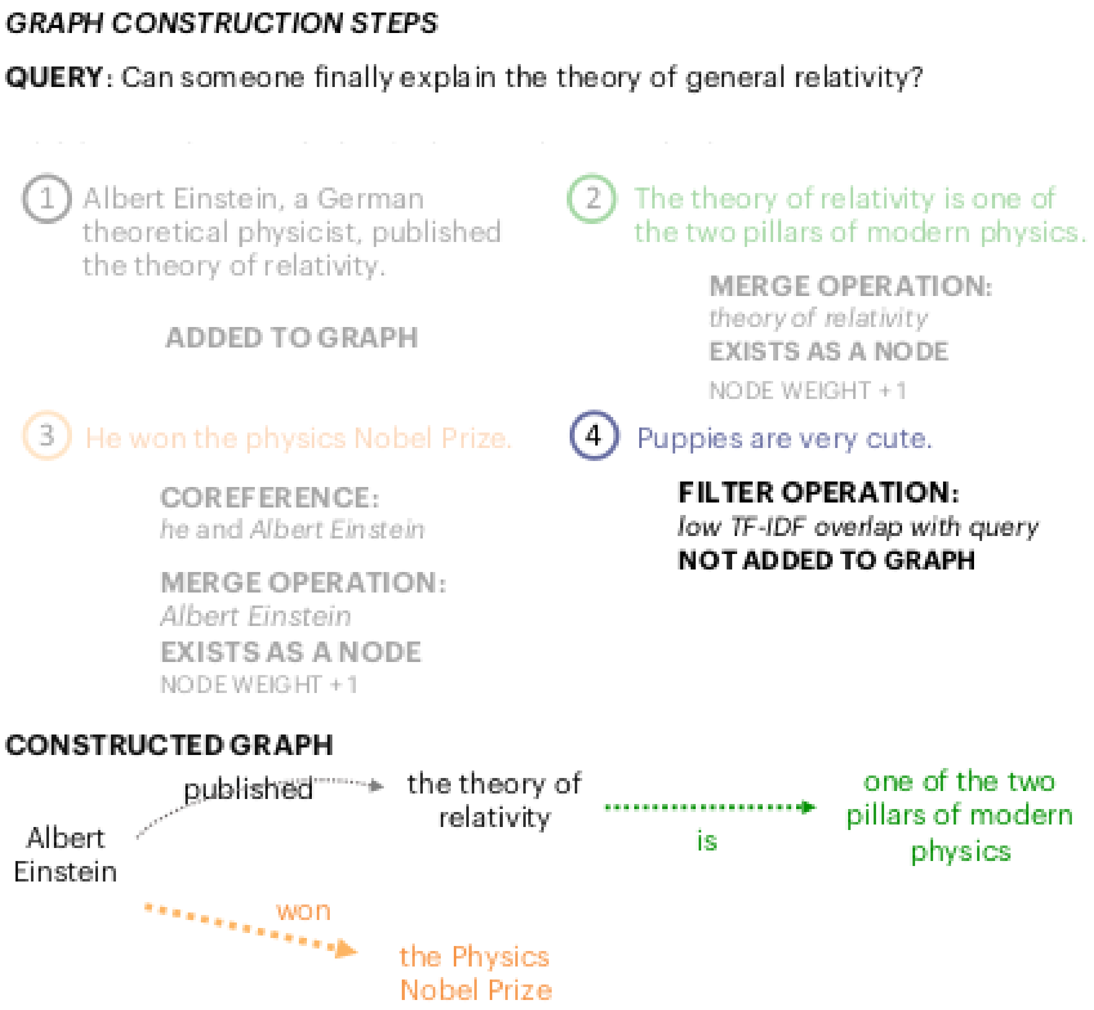


Dealing with Long Form Input



Converting text to a graph

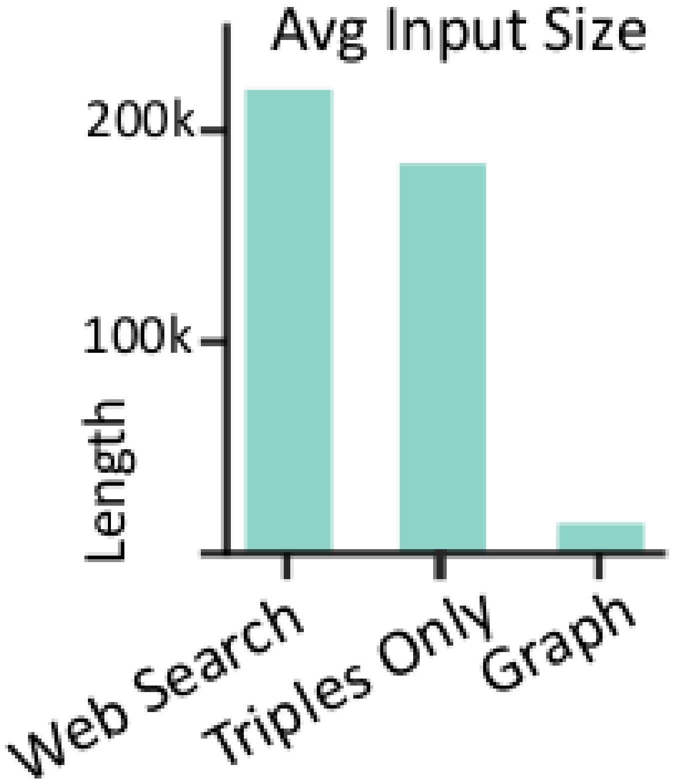



GRAPH CONSTRUCTION STEPS
QUERY: Can someone finally explain the theory of general relativity?
1 Albert Einstein, a German theoretical physicist, published the theory of relativity.
ADDED TO GRAPH
2 The theory of relativity is one of the two pillars of modern physics.
MERGE OPERATION:
theory of relativity
EXISTS AS A NODE
NODE WEIGHT +1
3 He won the physics Nobel Prize.
COREFERENCE:
he and Albert Einstein
MERGE OPERATION:
Albert Einstein
EXISTS AS A NODE
NODE WEIGHT +1
4 Puppies are very cute.
FILTER OPERATION:
low TF-IDF overlap with query
NOT ADDED TO GRAPH
CONSTRUCTED GRAPH
Albert Einstein
published
the theory of relativity
is
one of the two pillars of modern physics
won
the Physics Nobel Prize

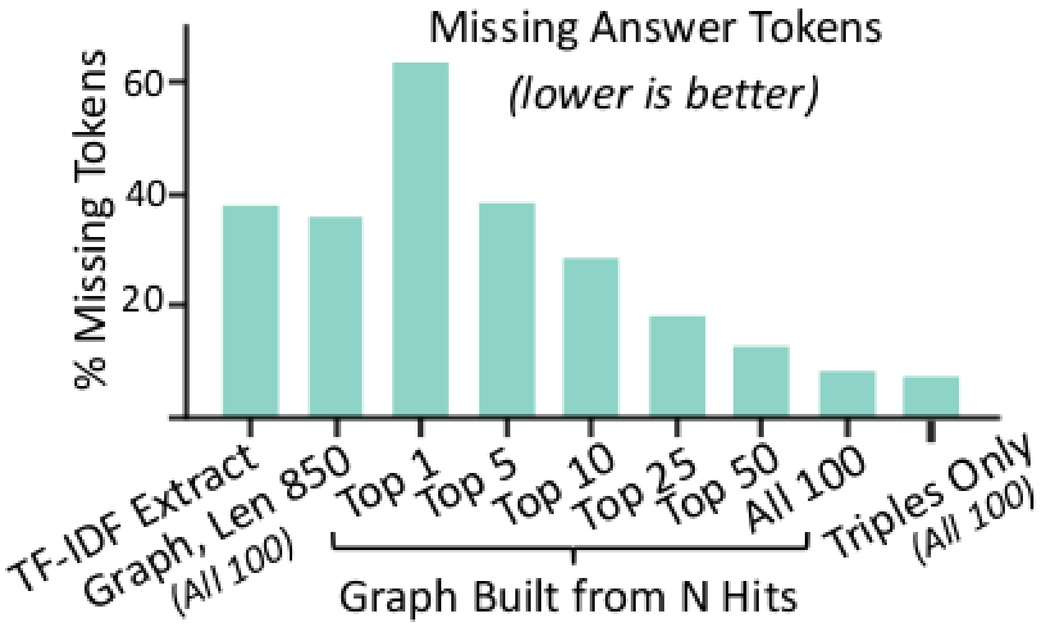
How much does the graph compress the input ?

Graph conversion

- drastically reduces the size of the input.
- allows for the full input to be encoded

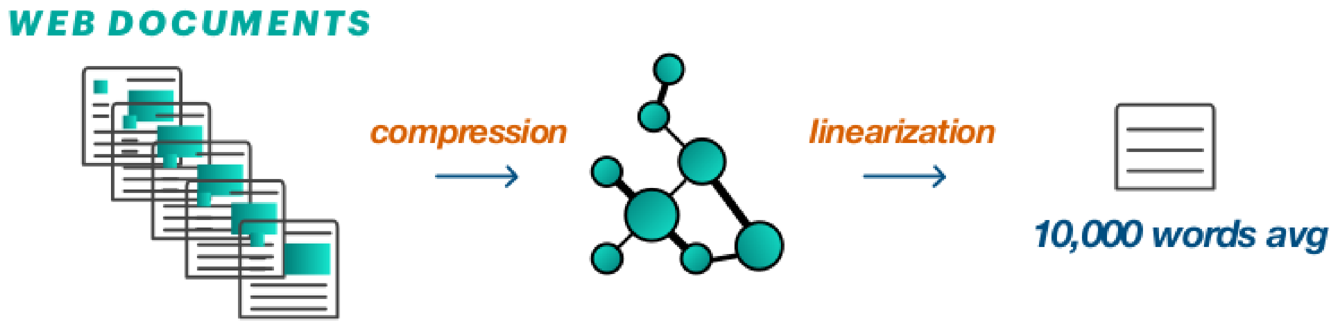


How much does the graph preserve answer relevant information ?



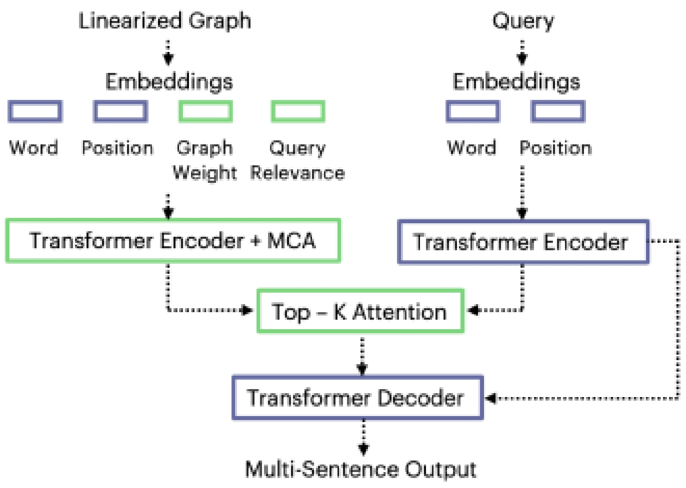
The graph for the full input is missing only 8.7% of the answer tokens .

Encoding the graph



WORD EMBEDDING	<sub>	Albert	Einstein	<obj>	the	theory	of	relativity	<pred>	published	<s>	developed	<obj>	the	Physics	Nobel	Prize	<s>	won
POSITION EMBEDDING	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
GRAPH WEIGHT EMBEDDING	0	4	4	0	2	2	2	2	0	1	0	1	0	3	3	3	3	0	2
QUERY RELEVANCE EMBEDDING	0	1	1	0	1	1	1	1	0	1	0	2	0	1	1	1	1	0	1

Encoding 10K tokens in a Transformer



- MCA in Encoder
  Memory Compressed Attention

- Hierarchical Top-k Attention

Example Answer Output (QA)

Question: Why is touching microfiber towels such an uncomfortable feeling?

True Answer: Do you mean the kind of cloths used to clean glasses and lenses? I've never noticed any uncomfortable feeling myself, but I do find touching certain cleaning cloths can be quite uncomfortable. There's a brand called "e - cloth" which market themselves as not needing any cleaning supplies. Basically the threads have tiny little hooks and barbs in them, so dragging them over skin makes you feel like it 's catching against you.

Generation: The microfiber is made up of a bunch of tiny fibers that are attached to them. When you touch them, the fibers that make up the microfiber are attracted to each other. When they touch, they're attracted to the other end of the fiber, which is what makes them feel uncomfortable.

Fluent and factually correct. Some repetition.

When you touch ...other/ When they touch, ... fiber.

Example Biography Output (Summarisation)

Title: wylie rogers

True Paragraph: wylie rogers (born march 16, 1985) is an american ice hockey goaltender. he played with the rio grande valley killer bees of the central hockey league. prior to turning professional, rogers attended the university of alaska fairbanks where he played four seasons of ncaa men's division i ice hockey with the alaska nanooks men's ice hockey team. rogers played the 2010-11 season in the netherlands with the heerenveen flyers.

Generation: wylie rogers is an american football coach and former player. he is currently the head football coach at the university of north carolina at chapel hill. rogers played college football at north carolina state university. he was a member of the north carolina tar heels men's football team that won the ncaa division i football championship in 2012. he also played for the university's men's basketball team. after his playing career ended, rogers was hired as the head coach at unc chapel hill in 2013. he led the tar heels to their first ncaa division ii championship in 2015.

Key challenge: factuality

Wylie Rogers is an ice hockey player not a football coach.
He is at University of Alaska not North Carolina.

Generating Text using External Knowledge

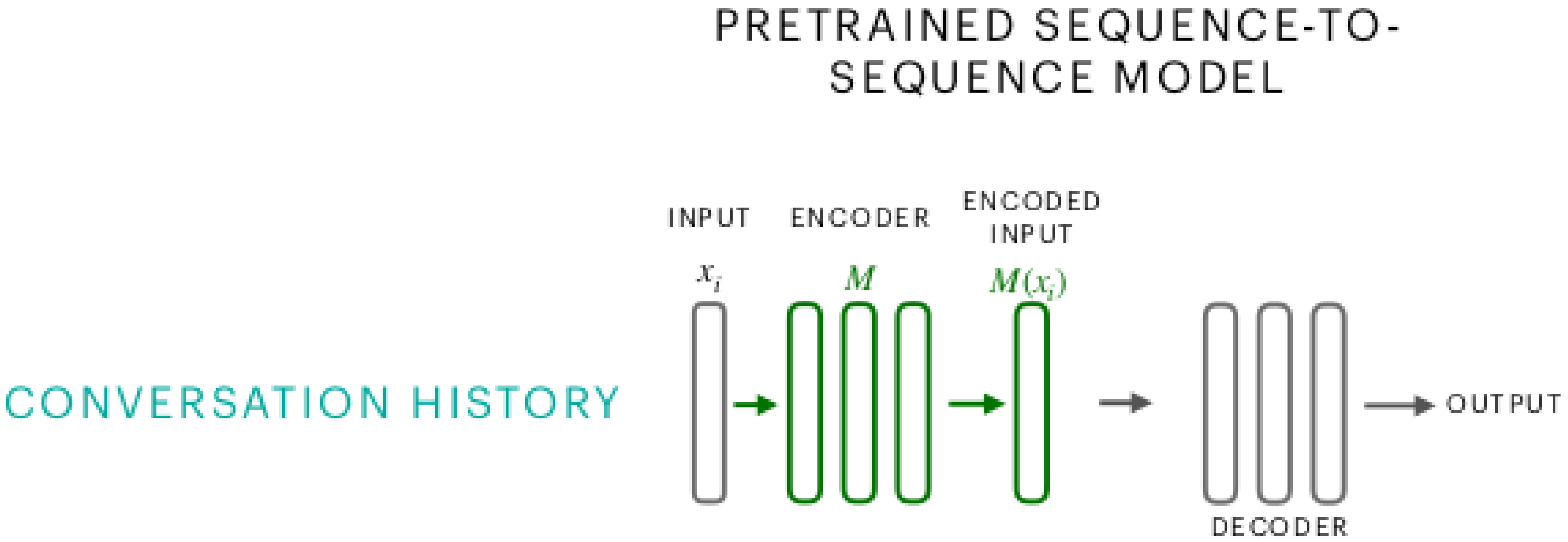
Knowledge in Human-Machine Dialog

Human: What's your favorite disney movie? There are so many good ones.
Model: I love the incredibles, they are my favorite disney movie.
Fetched Knowledge: *disney announced intentions to develop additional superhero films after the success of the incredibles*
Fetched Training Utterance: *i love kiteboarding, it is one of my favorite activities on the water.*

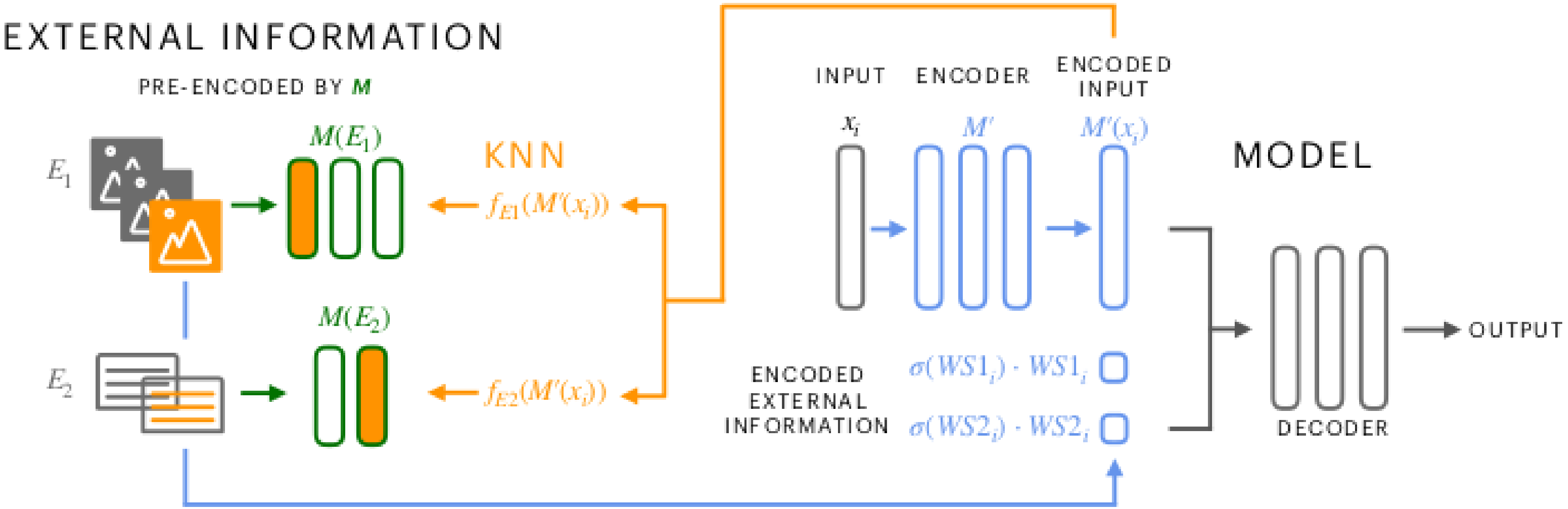
- World knowledge helps give content to the dialog
- Similar training utterances provide a template for the response (Linguistic knowledge)

Fan et al. 2021

Human-Machine Dialog

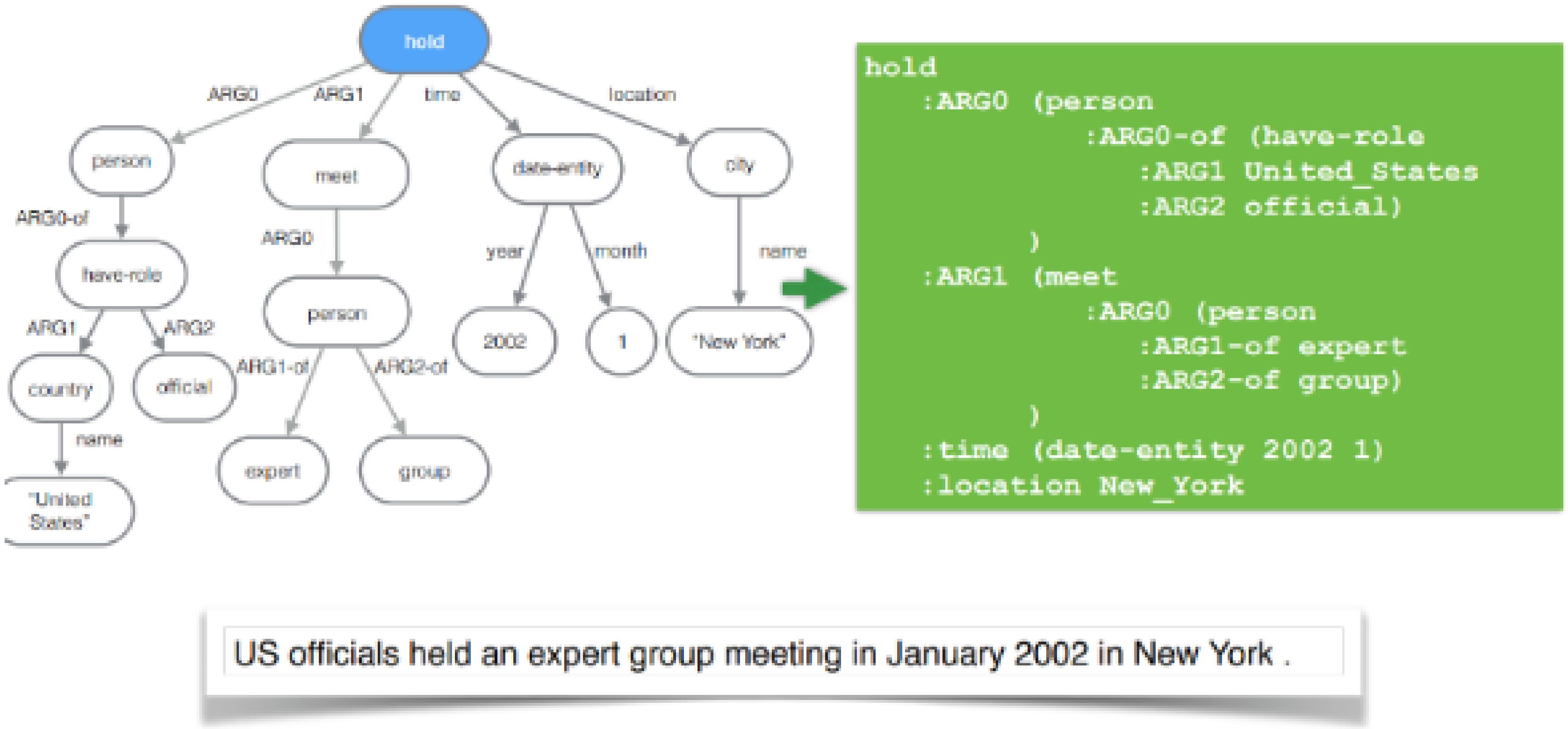


Extending Human-Machine Dialog with External Knowledge Retrieval



Generating Text into 21 EU Languages

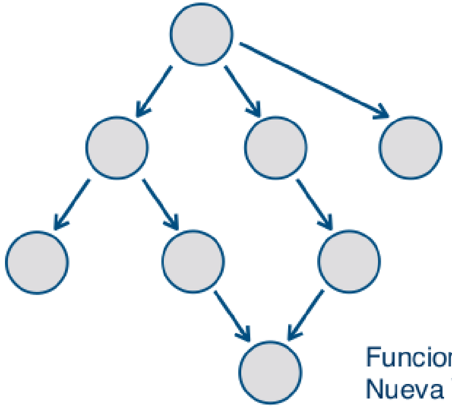
Generating Text from Abstract Meaning Representations (AMR)



Fan et al. 2020

Graph $\rightarrow$ 21 Languages

Amerikanska tjänstemän höll ett expertgruppsmöte i januari 2002 i New York.



Americkí predstavitelia usporiadali stretnutie expertnej skupiny v januári 2002 v New Yorku.

US officials held an expert group meeting in January 2002 in New York.

Des responsables américains ont tenu une réunion d'un groupe d'experts en janvier 2002 à Ne York.

Funcionarios estadounidenses celebraron una reunión de un grupo de expertos en enero de 2002 en Nueva York.

Romance, Germanic, Slavic, Uralic

Encoding AMRs

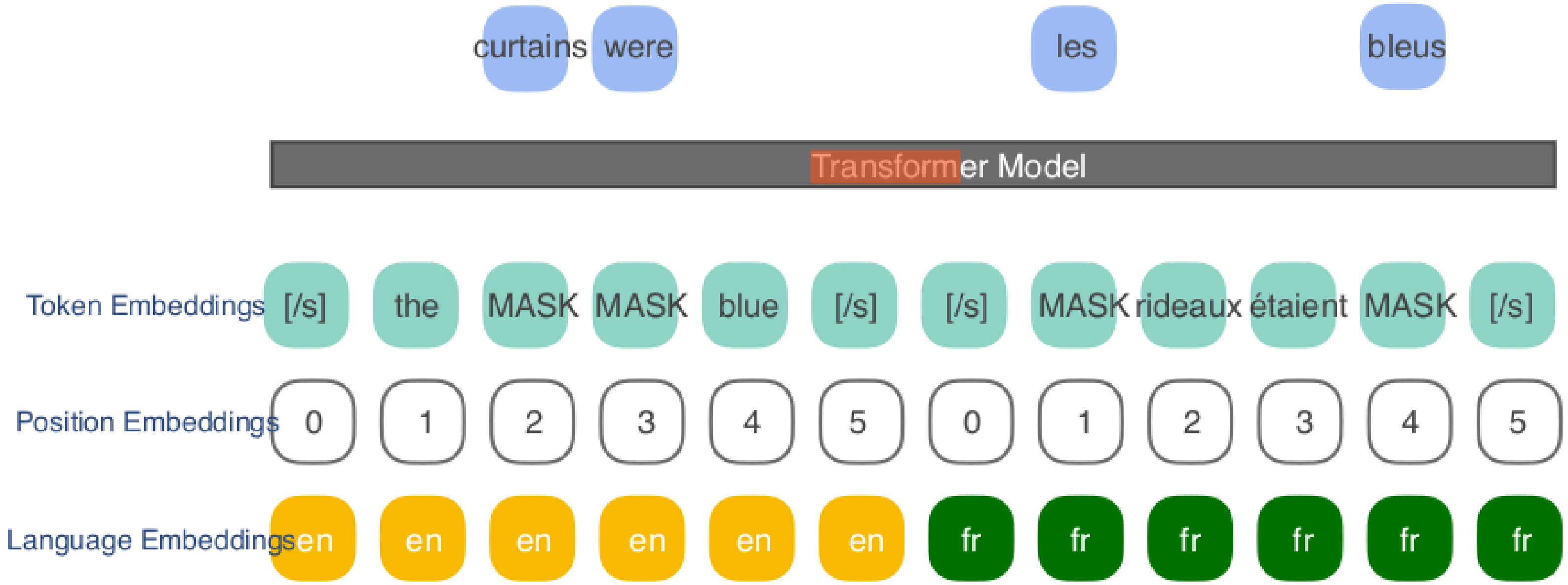
- Transformer encoder
- Linearise (and simplify) AMRs
- Graph structure
  Node: token + distance from root + subgraph identifier
- Pretrain encoder on 30M silver AMRs derived from text using JAMR

Multilingual Decoding

- Crosslingual embeddings
  Shared vector space for words from different languages

- Multilingual decoding
  Prefix each training instance with a control token
  Train on multilingual data

Cross-lingual Embeddings



Lample and Conneau (2019)

Multilingual Decoding

Decoding into Slovak

sv

hold
:ARG0 person : ARG0-of have-org-role :ARG1 :op1
United :op2 States :ARG2 official
:ARG1 meet :ARG0 person :ARG1-of expert :ARG2-
of group
:time date-entity :year 2002 :month 1
:location city :op1 New :op2 York

Amerikanska tjänstemän höll ett
expertgruppsmöte i januari 2002 i New York.

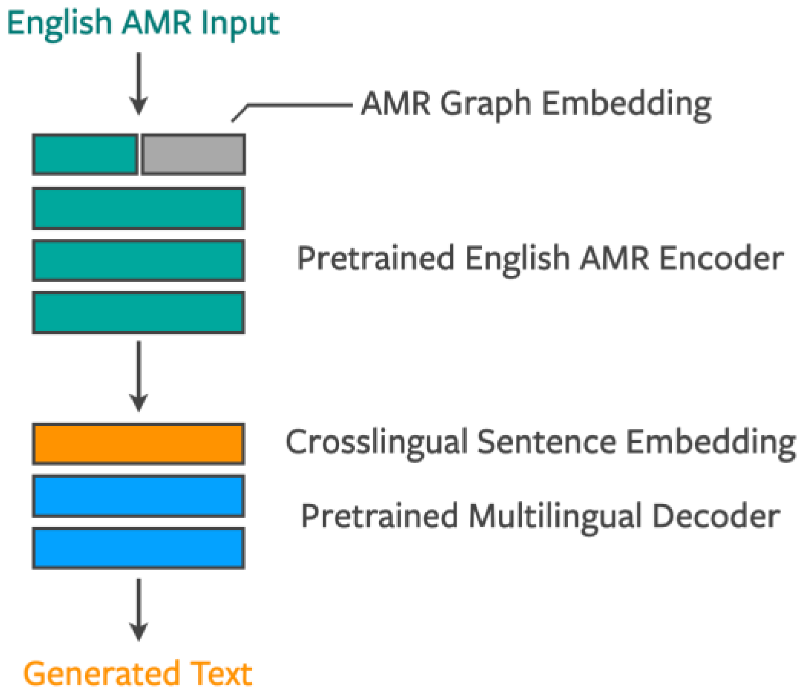
Decoding into French

fr

hold
:ARG0 person : ARG0-of have-org-role :ARG1 :op1
United :op2 States :ARG2 official
:ARG1 meet :ARG0 person :ARG1-of expert :ARG2-
of group
:time date-entity :year 2002 :month 1
:location city :op1 New :op2 York

Des responsables américains ont tenu une
réunion d'un groupe d'experts en janvier 2002 à
New York.

Multilingual AMR-to-NL Model



- Encoder: pretraining on Silver AMRs
- Decoder: language model pretraining

Open Questions

Open Questions

- Factuality, Faithfulness to the input
- Multilingual Input/Output
- Long form Input/Output
- Multi-modal Input

The end