Hochschule für
Wirtschaft und Recht Berlin
Berlin School of Economics and Law

Text Representation Part 2 Embedding-based approaches

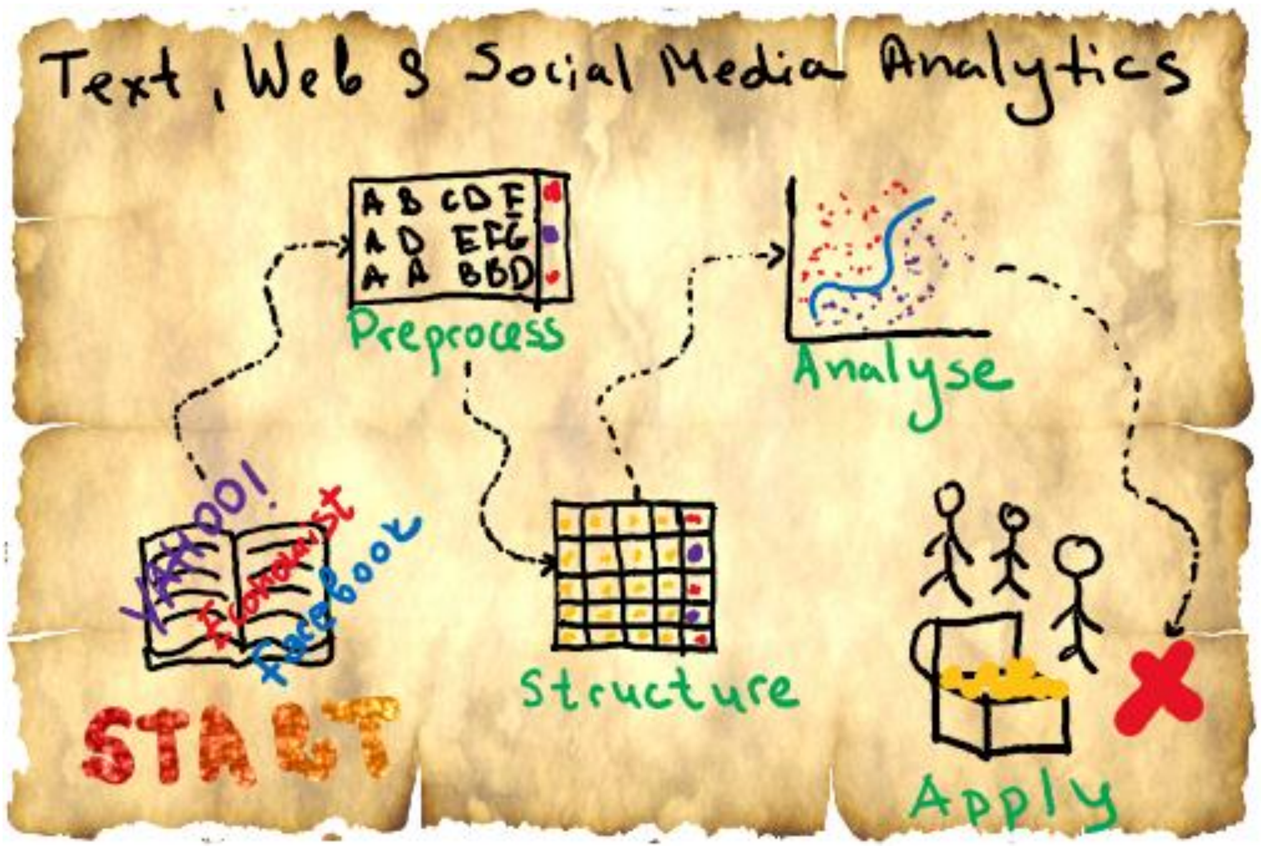
Text, Web and Social Media Analytics Lab

Prof. Dr. Diana Hristova

Exercise 3: Representation (1)

Can one group present please?

Treasury map: Text Representation



Course structure

Hochschule für Wirtschaft und Recht Berlin
Berlin School of Economics and Law

Date	Lecture	Exercise
12.04.2021	Introduction	Technical Installation
19.04.2021	Text Preprocessing	Projects kick-off
26.04.2021	Text Representation	Preprocessing Newsgroups
03.05.2021	**Text Representation (2)**	**Text Representation Newsgroups**
10.05.2021	Text Classification	**Text Representation Newsgroups (2)**
17.05.2021	Text Clustering	Newsgroups Topic Classification
31.05.2021	Text Mining in Social Media	Newsgroups Topic Clustering
07.06.2021	Mining Social Graphs	Sentiment Analysis and Time Series in Twitter
14.06.2021	Projects Status Update	Projects Status Update
21.06.2021	Web Analytics	Mining Social Graphs in Twitter
28.06.2021	Mock Exam	Web Analytics in E-commerce
05.07.2021	Final Presentation	Final Presentation
19.07.2021	Submit Code & Written report	
t.b.a.	Exam	

What will we learn today?

At the end of this lecture, you will:

1. Know the main idea behind using embeddings as document representation.
2. Understand the Word2vec approach for document representation.
3. Be able to compute document similarity.
4. Understand how transformer-(based) models work.

Motivation: Embeddings

Hochschule für Wirtschaft und Recht Berlin
Berlin School of Economics and Law

Reminder

- The Bag-of-words approach uses a vector where each feature represents one word in the dictionary.
- The feature values are the word importances
- Example: let the preprocessed corpus consist of the following sentences:
  - Doc1: “I like apples, but don’t like oranges.” ➔ “like appl like orang”
  - Doc2: “I like oranges.” ➔ “like orang”
  - Doc3: “I prefer apples to oranges.” ➔ “prefer appl orang”

Bag-of-words ➔

Doc	like	appl	orang	prefer
Doc1	2	1	1	0
Doc2	1	0	1	0
Doc3	0	1	1	1

Motivation: Embeddings

What is the Bag-of-words representation of those two documents with absolute frequencies?

- Doc1: “like appl”
- Doc2: “adore orang”

Motivation: Embeddings (2)

- Example: let the preprocessed corpus consist of the following sentences:
  - Doc1: “like appl”
  - Doc2: “adore orang”

Bag-of-words →

Doc	like	appl	orang	adore
Doc1	1	1	0	0
Doc2	0	0	1	1

? Can you compare those documents based on their representation?

Motivation: Embeddings (2)

- Example: let the preprocessed corpus consist of the following sentences:
  - Doc1: “like appl”
  - Doc2: “adore orang”

Bag-of-words ➡

Doc	like	appl	orang	adore
Doc1	1	1	0	0
Doc2	0	0	1	1

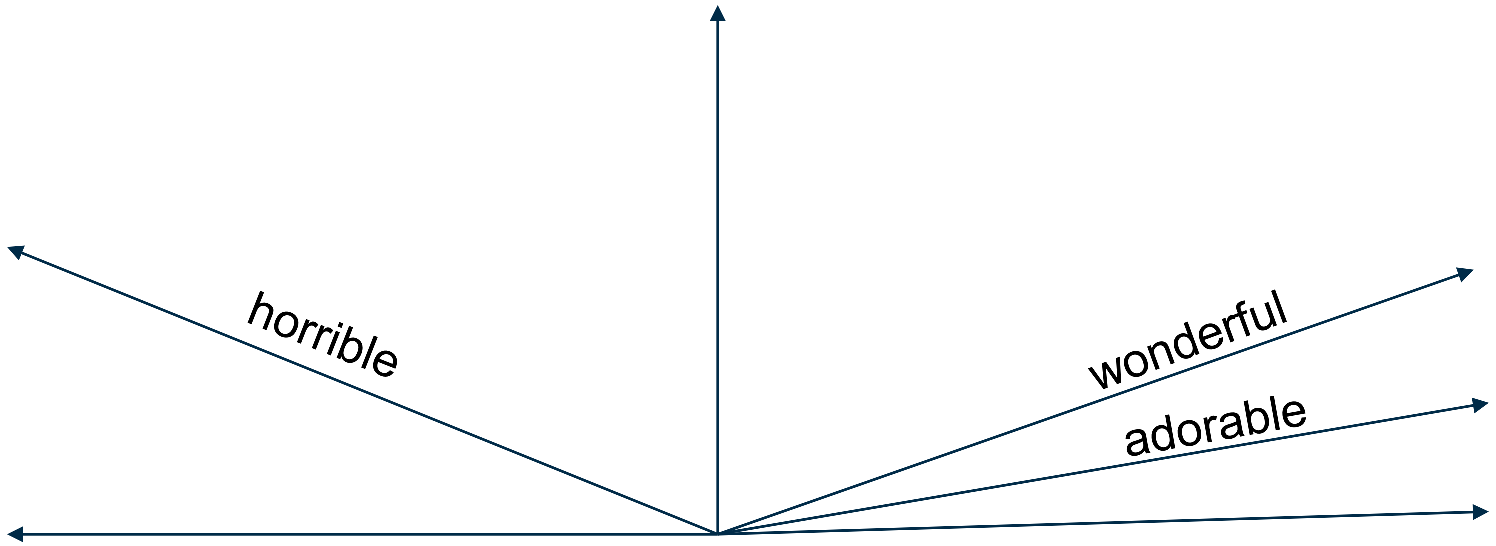
? Can you compare those documents based on their representation?

- Bag-of-words approaches do not consider the general meaning of the word in **relation to others** (e.g. ‚like‘ is similar to ‚adore‘) and thus do not allow for the comparison of documents that do not share words.

▶ Embeddings

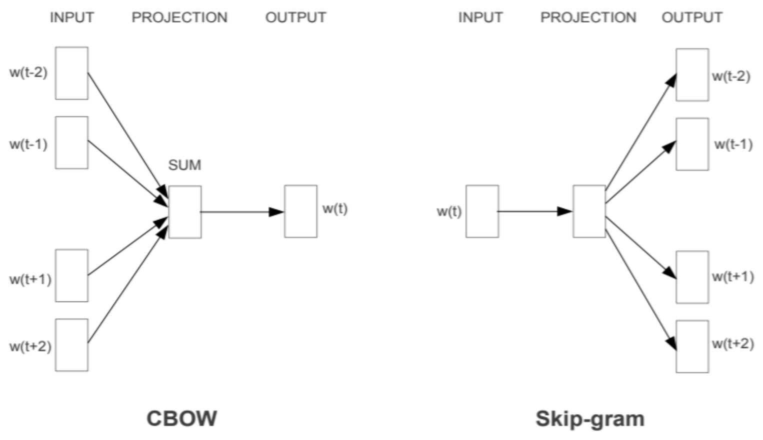
Vectorization: Embeddings

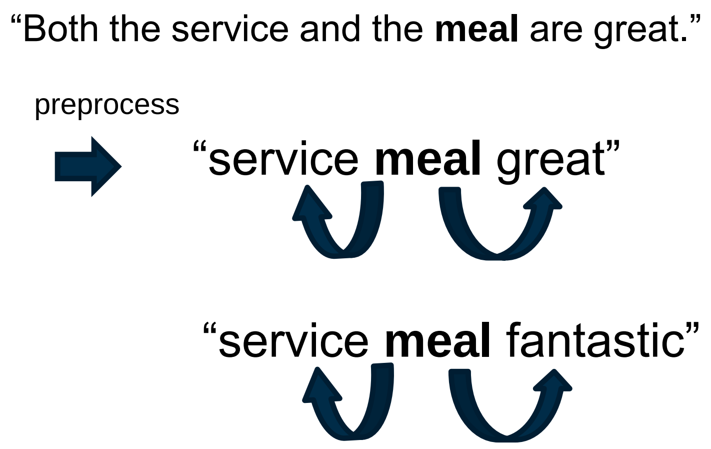
- **Embeddings (distributional representation)** represent words in a continuous vector space such that similar words are close together.
- Allow for negative values to facilitate the representation of similarity.



Embeddings: Word2vec

- Word2Vec is an embeddings‘ model created by Google (Mikolov et al. 2013).
- **Main idea:** words that appear in similar contexts must have similar embedding representation.
- It is based on a two-layer neural network.
- **Two approaches:** continues-bag-of-words and skip-gram



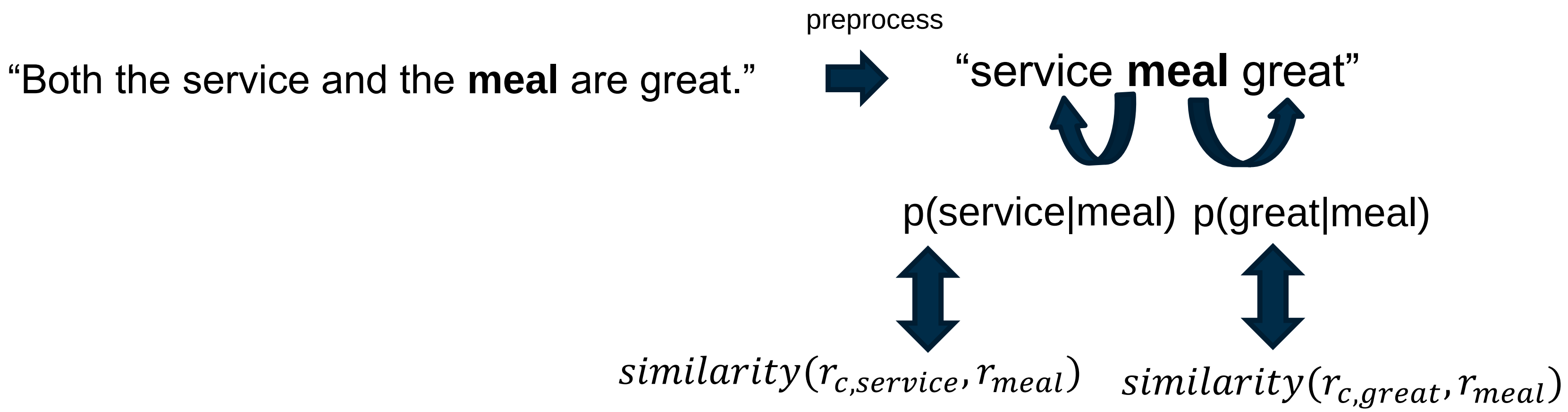

“Both the service and the **meal** are great.”
preprocess
“service **meal** great”
“service **meal** fantastic”

Mikolov, Tomas; Sutskever, Ilya; Chen, Kai; Corrado, Greg; Dean, Jeffrey (2013): Distributed Representations of Words and Phrases and their Compositionality.
Mikolov, T., Le, Q. V., & Sutskever, I. (2013). Exploiting similarities among languages for machine translation. *arXiv preprint arXiv:1309.4168*

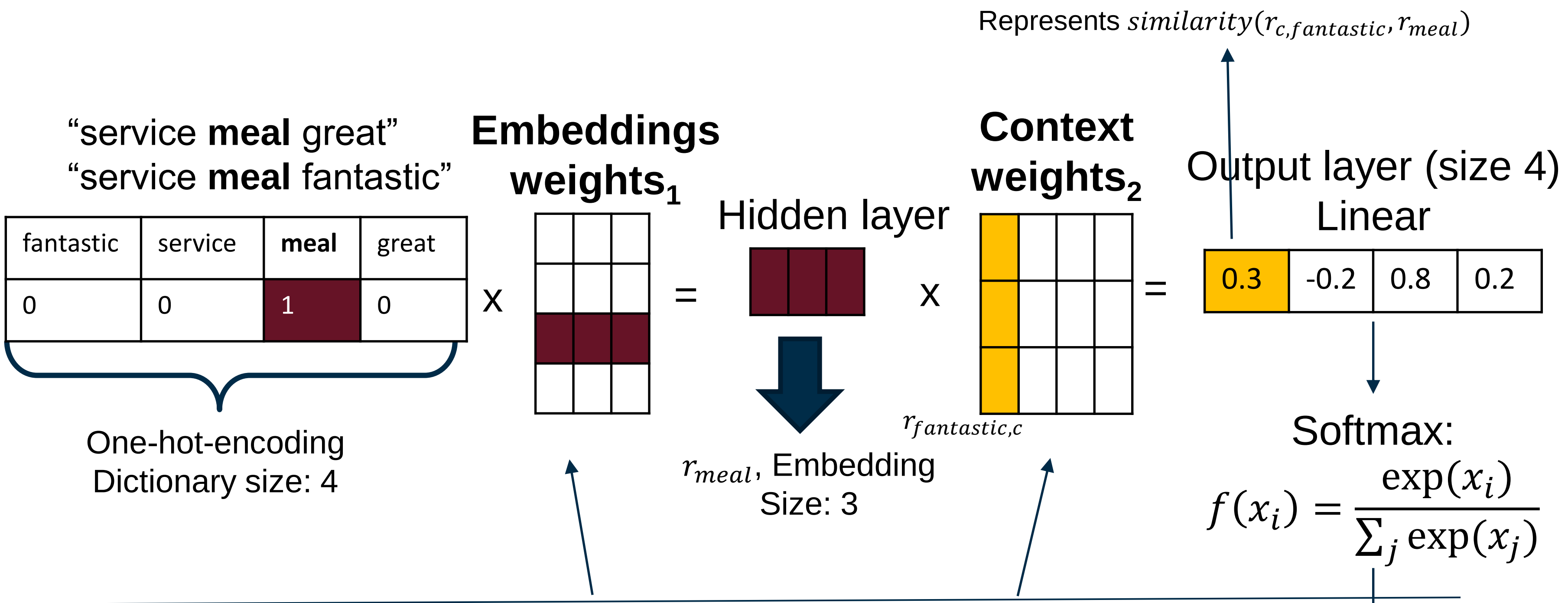
Embeddings: Word2vec (2)

Training (Skip-gram): predict words that appear in the neighbourhood of the word *w*

1. Start with an initial embedding and context representation for each word in the dictionary
2. For the words other than *w*, calculate the probability that they appear in the neighbourhood of *w* based on the similarity of their embeddings and context representation
3. Adjust the representations, so that the calculated probability corresponds as well as possible to the true probability in the text.


“Both the service and the **meal** are great.”
preprocess
“service **meal** great”
p(service|meal)
p(great|meal)
$similarity(r_{c,service}, r_{meal})$
$similarity(r_{c,great}, r_{meal})$

Embeddings: Word2vec (3)



fantastic	service	**meal**	great
0	0	1	0

$$f(x_i) = \frac{\exp(x_i)}{\sum_j \exp(x_j)}$$

Real context of meal (window=2)

Ground truth

fantastic	service	great
25%	50%	25%

vs.

fantastic	service	**meal**	great
20%	50%	**5%**	25%

Embeddings: Word2vec (4)

The resulting embeddings:

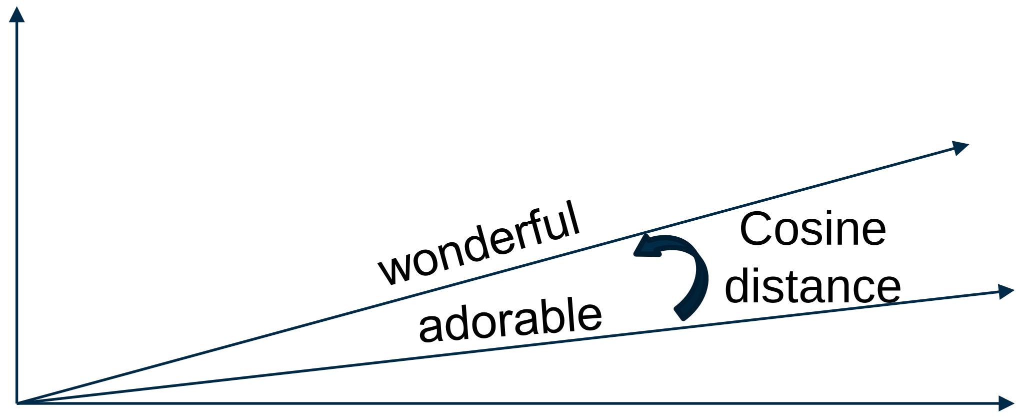
- are derived based on the similarity between words,
- allow for vector arithmetic that mimic the semantic meaning of a word (e.g. King-Men+Woman=Queen),
- are the **same** for all usages of the word.

? How can we calculate similarity between two representations?

Embeddings: Cosine Similarity

Hochschule für Wirtschaft und Recht Berlin
Berlin School of Economics and Law

- **Cosine similarity (based on the cosine distance)**



- $word1 = (a_1, a_2)$ and $word2 = (b_1, b_2)$

$$CosSim(word1, word2) = \frac{a_1 b_1 + a_2 b_2}{\sqrt{{a_1}^2 + {a_2}^2}\sqrt{{b_1}^2 + {b_2}^2}}$$

Embeddings: Cosine Similarity (2)

- $word1 = (a_1, a_2)$ and $word2 = (b_1, b_2)$

$$CosSim(word1, word2) = \frac{a_1 b_1 + a_2 b_2}{\sqrt{{a_1}^2 + {a_2}^2}\sqrt{{b_1}^2 + {b_2}^2}}$$

- $wonderful = (5,1), adorable = (5,4), horrible = (-5,3)$

$$CosSim(wonderful, adorable) = \frac{5 * 5 + 1 * 4}{\sqrt{5^2 + 1^2}\sqrt{5^2 + 4^2}} = 0.89$$

$$CosSim(wonderful, horrible) = ?$$

Embeddings: Cosine Similarity (3)

Hochschule für Wirtschaft und Recht Berlin
Berlin School of Economics and Law

- $word1 = (a_1, a_2)$ and $word2 = (b_1, b_2)$

$$CosSim(word1, word2) = \frac{a_1 b_1 + a_2 b_2}{\sqrt{{a_1}^2 + {a_2}^2}\sqrt{{b_1}^2 + {b_2}^2}}$$

- $wonderful = (5,1), adorable = (5,4), horrible = (-5,3)$

$$CosSim(wonderful, adorable) = \frac{5 * 5 + 1 * 4}{\sqrt{5^2 + 1^2}\sqrt{5^2 + 4^2}} = 0.89$$

$$CosSim(wonderful, horrible) = \frac{5 * (-5) + 1 * 3}{\sqrt{5^2 + 1^2}\sqrt{(-5)^2 + 3^2}} = -0.74$$

Embeddings: Word2vec (5)

- **Advantages:** Considers the relationship of the word meaning to other words, interpretable.
- **Disadvantage:** vector representation is always the same regardless of the particular use.

Why is this a disadvantage?

Embeddings: Word2Vec (3)

- **Advantages:** Considers the relationship of the word meaning to other words, interpretable.
- **Disadvantage:** vector representation is always the same regardless of the particular use.
  - ‘bank office’
  - ‘river bank’
  - ‘blood bank’

▶ Contextual Embeddings

Hochschule für Wirtschaft und Recht Berlin
Berlin School of Economics and Law

Contextual Embeddings: Embedding for Language Models (ELMo)

- ELMo was developed by AI2 (Peters et al. 2018)
- As opposed to Word2vec, ELMo embeddings are not fixed, but calculated every time depending on the given text.
- It uses a deep learning model that considers long context dependencies e.g. sentences and not just the two words before and after.
- ELMo is derived at a character level ➔ it can predict words it has not seen
- There are readily trained models which you can fine-tune for your problem.

Peters, Matthew E.; Neumann, Mark; Iyyer, Mohit; Gardner, Matt; Clark, Christopher; Lee, Kenton; Zettlemoyer, Luke (2018): Deep contextualized word representations.

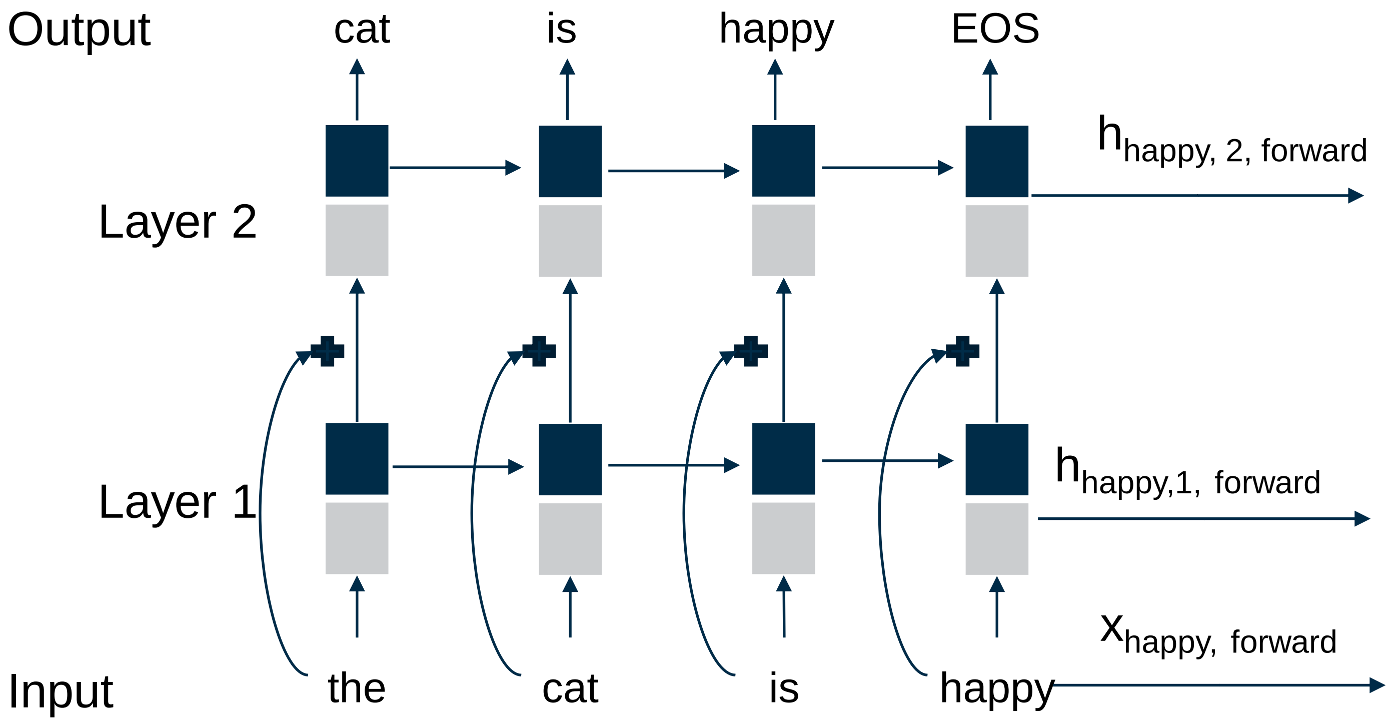
https://www.flickr.com/photos/jeepersmedia/12704477613

Contextual Embeddings: Embedding for Language Models (ELMo) (2)

- ELMo was trained to predict the next word given the sequence before and after it.
- It uses a 2-layer-Long Short-Term Memory Neural Network (LSTM): a special type of Recurrent Neural Network capable of modelling long-term dependencies, which:
  - Sequentially transforms the information coming from the previous layer and
  - Each layer produces a hidden state representing the current information.
- The final embeddings are derived by combining (concatenation and weighted summation) the hidden state values with the initial embedding.
- The network is bi-directional such that both a forward and a backward language model (i.e., next words and previous words) are trained.
- The results from the last layer are fed into a linear+ softmax layer to produce prediction probabilities.

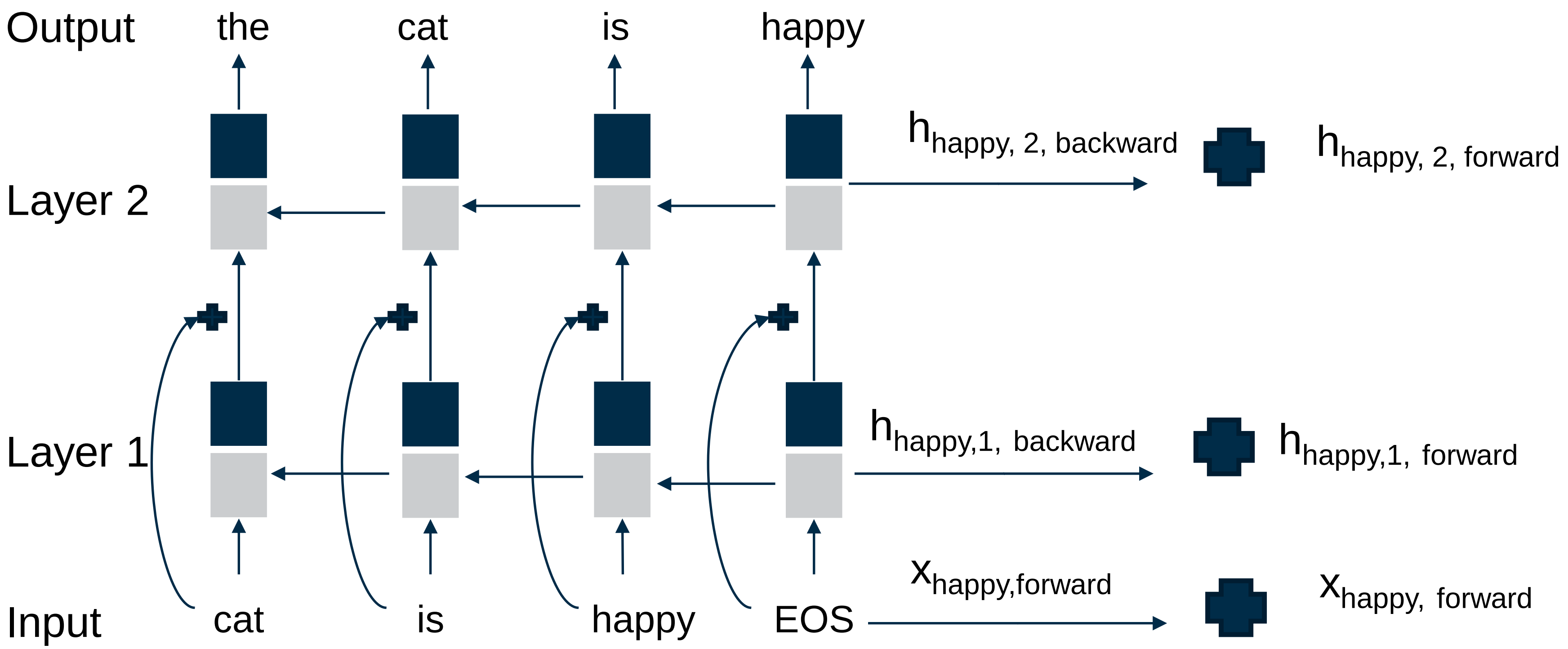
Hochschule für Wirtschaft und Recht Berlin
Berlin School of Economics and Law

Contextual Embeddings: Embedding for Language Models (ELMo) (3)



Hochschule für Wirtschaft und Recht Berlin
Berlin School of Economics and Law

Contextual Embeddings: Embedding for Language Models (ELMo) (3)

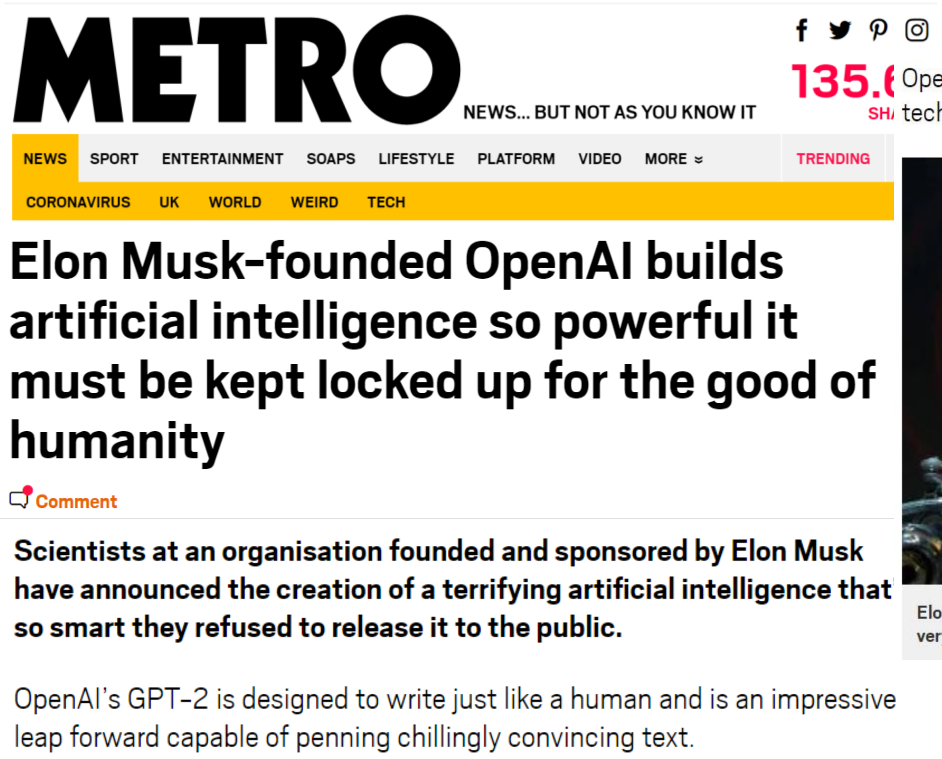


- **Disadvantage:** Training is difficult to parallelise due to the recurrent nature.

➔ Transformers

Hochschule für Wirtschaft und Recht Berlin
Berlin School of Economics and Law

Contextual Embeddings: The Transformer



METRO NEWS... BUT NOT AS YOU KNOW IT

135.

SH

NEWS SPORT ENTERTAINMENT SOAPS LIFESTYLE PLATFORM VIDEO MORE TRENDING

CORONAVIRUS UK WORLD WEIRD TECH

Elon Musk-founded OpenAI builds artificial intelligence so powerful it must be kept locked up for the good of humanity

Comment

Scientists at an organisation founded and sponsored by Elon Musk have announced the creation of a terrifying artificial intelligence that so smart they refused to release it to the public.

OpenAI's GPT-2 is designed to write just like a human and is an impressive leap forward capable of penning chillingly convincing text.

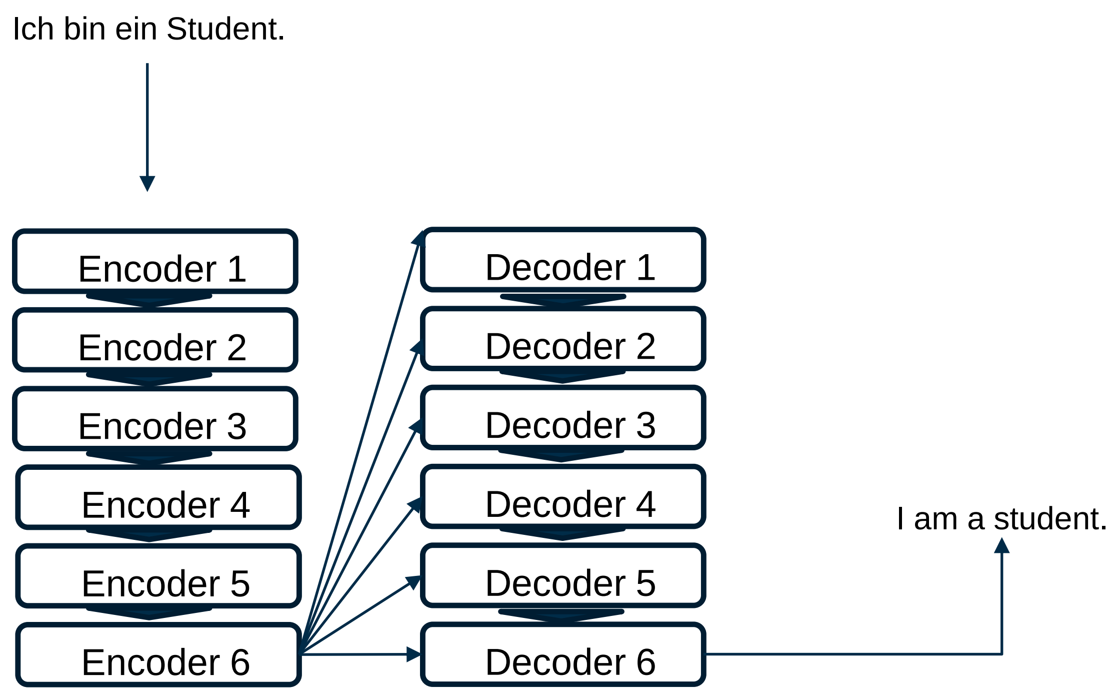
OpenAI wrote: 'Due to our concerns about malicious applications of the technology, we are not releasing the trained model.

Elon is one of the world's most famous doom-mongers and fears the rise of the machines will end very badly for humans Pictures: REX/Joe Rogan Experience

https://metro.co.uk/2019/02/15/elon-musks-openai-builds-artificial-intelligence-powerful-must-kept-locked-good-humanity-8634379/

Transformers: Main Architecture

- Transformers follow an encoder-decoder transfomer architecture stemming from translation systems.
- The original transformer paper uses six encoders and six decoders.



Vaswani, Ashish; Shazeer, Noam; Parmar, Niki; Uszkoreit, Jakob; Jones, Llion; Gomez, Aidan N. et al. (2017): Attention Is All You Need. Online available at http://arxiv.org/pdf/1706.03762v5.

Transformers: Inputs

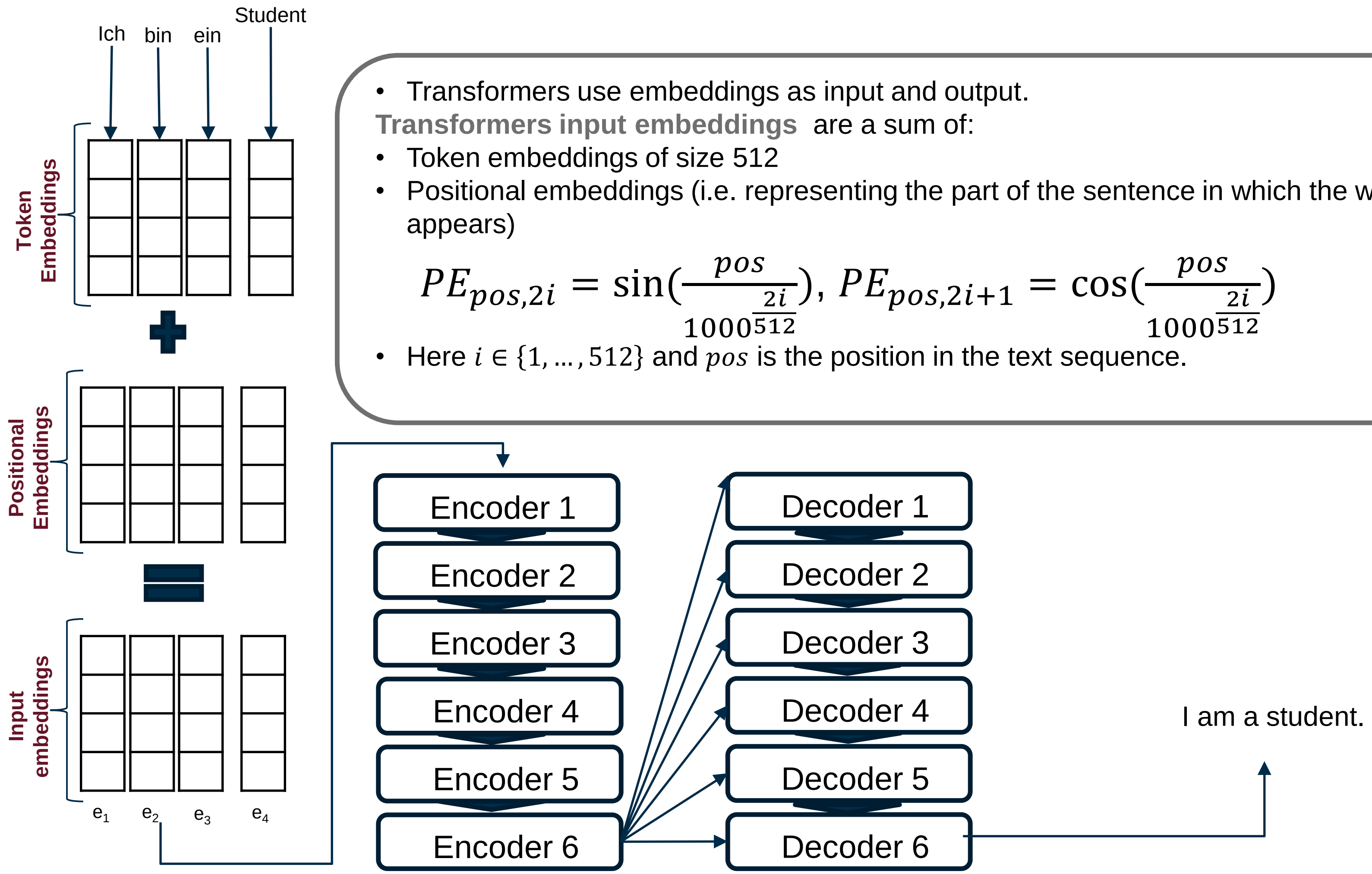
- Transformers use embeddings as input and output.

Transformers input embeddings are a sum of:

- Token embeddings of size 512
- Positional embeddings (i.e. representing the part of the sentence in which the word appears)

$$PE_{pos,2i} = \sin\left(\frac{pos}{1000^{\frac{2i}{512}}}\right),\ PE_{pos,2i+1} = \cos\left(\frac{pos}{1000^{\frac{2i}{512}}}\right)$$

- Here $i \in \{1, \dots, 512\}$ and pos is the position in the text sequence.



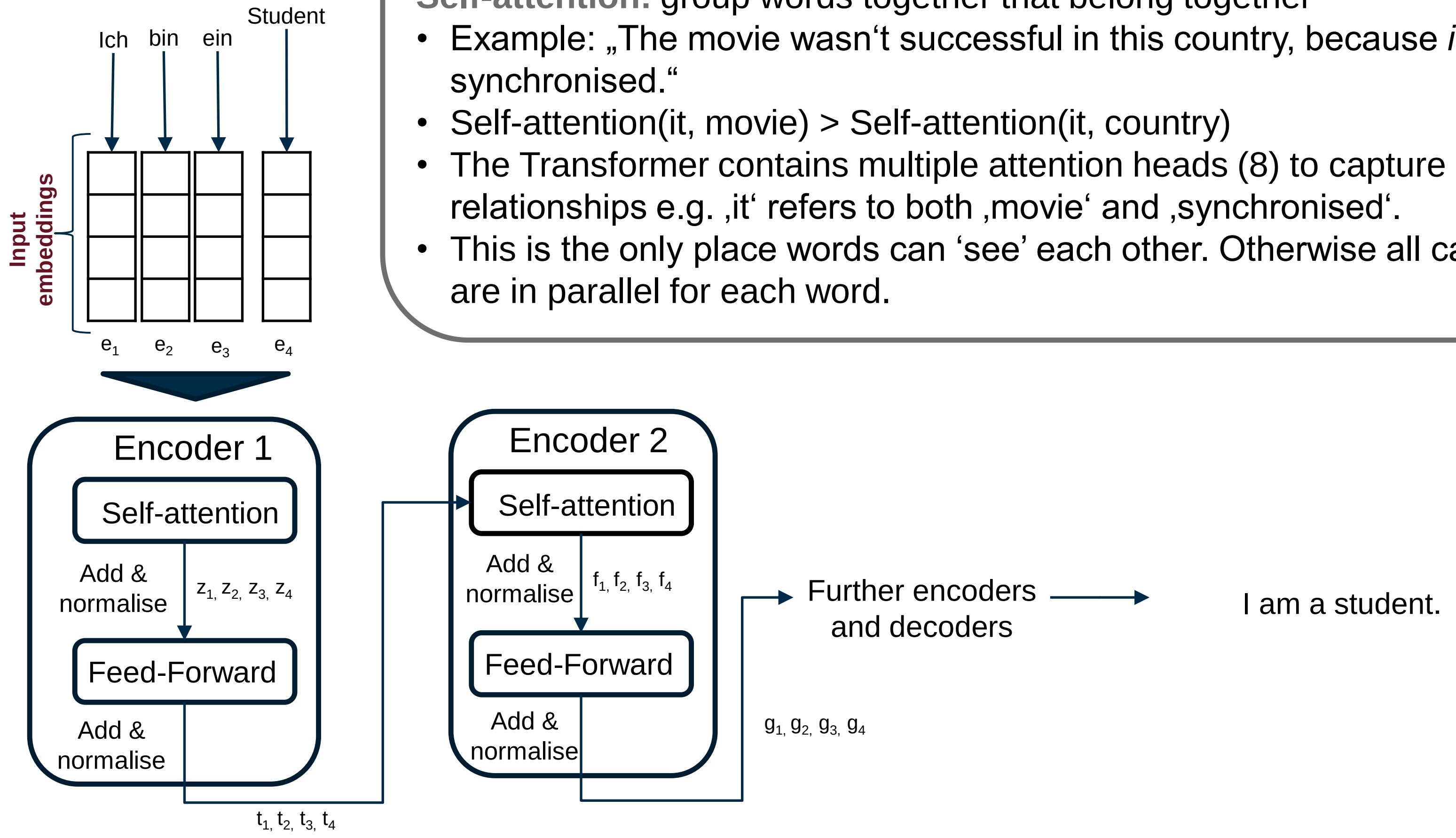
Transformers: Encoders

How does the encoder work?

- It consists of **Self-attention**, **Add &Normalize** and a **Feed-forward** element

Self-attention: group words together that belong together

- Example: „The movie wasn‘t successful in this country, because *it* was not synchronised.“
- Self-attention(it, movie) > Self-attention(it, country)
- The Transformer contains multiple attention heads (8) to capture multiple relationships e.g. ‚it‘ refers to both ‚movie‘ and ‚synchronised‘.
- This is the only place words can ‘see’ each other. Otherwise all calculations are in parallel for each word.



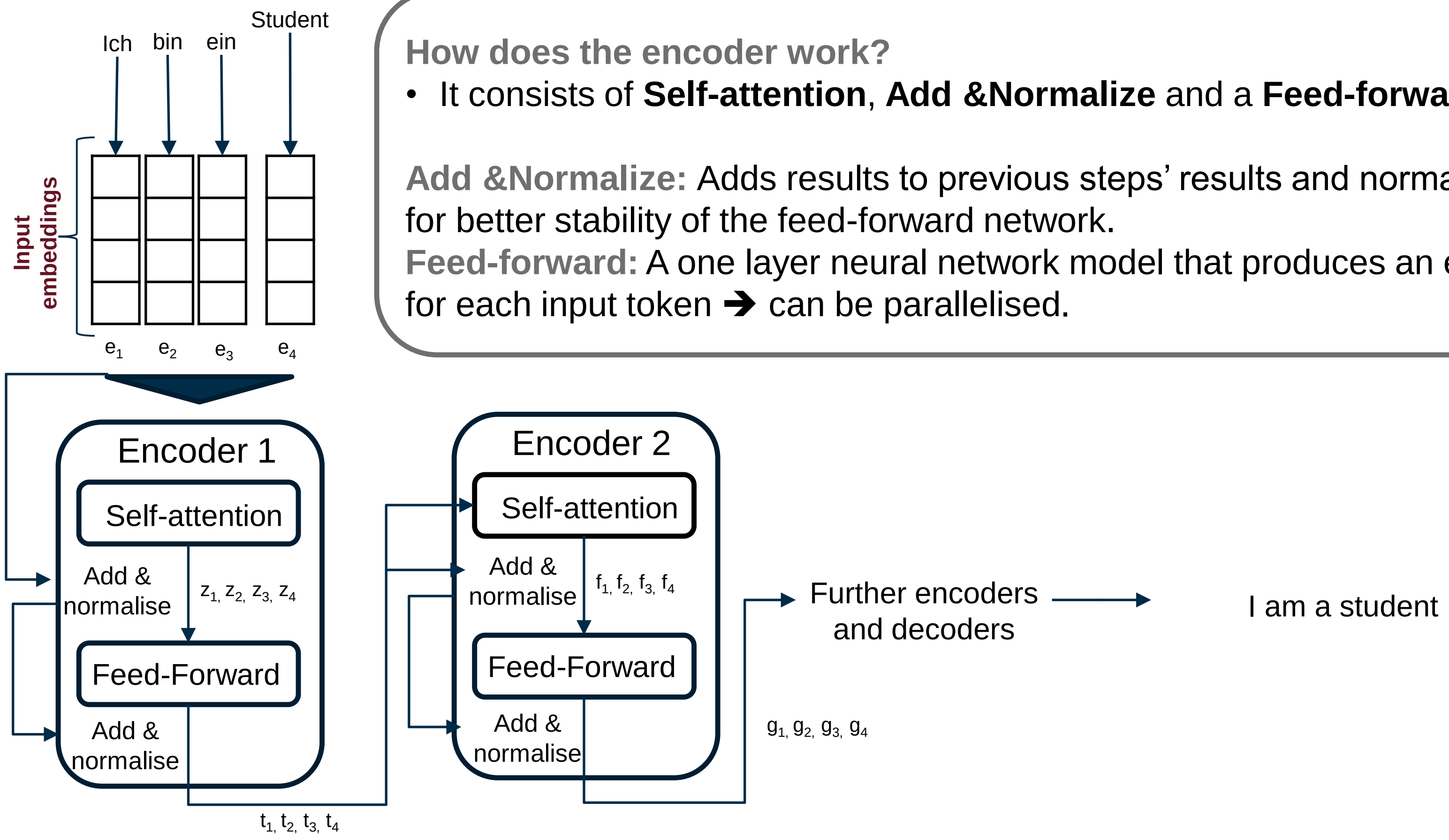
Transformers: Encoders (2)

Hochschule für Wirtschaft und Recht Berlin
Berlin School of Economics and Law

How does the encoder work?

- It consists of **Self-attention**, **Add &Normalize** and a **Feed-forward** element

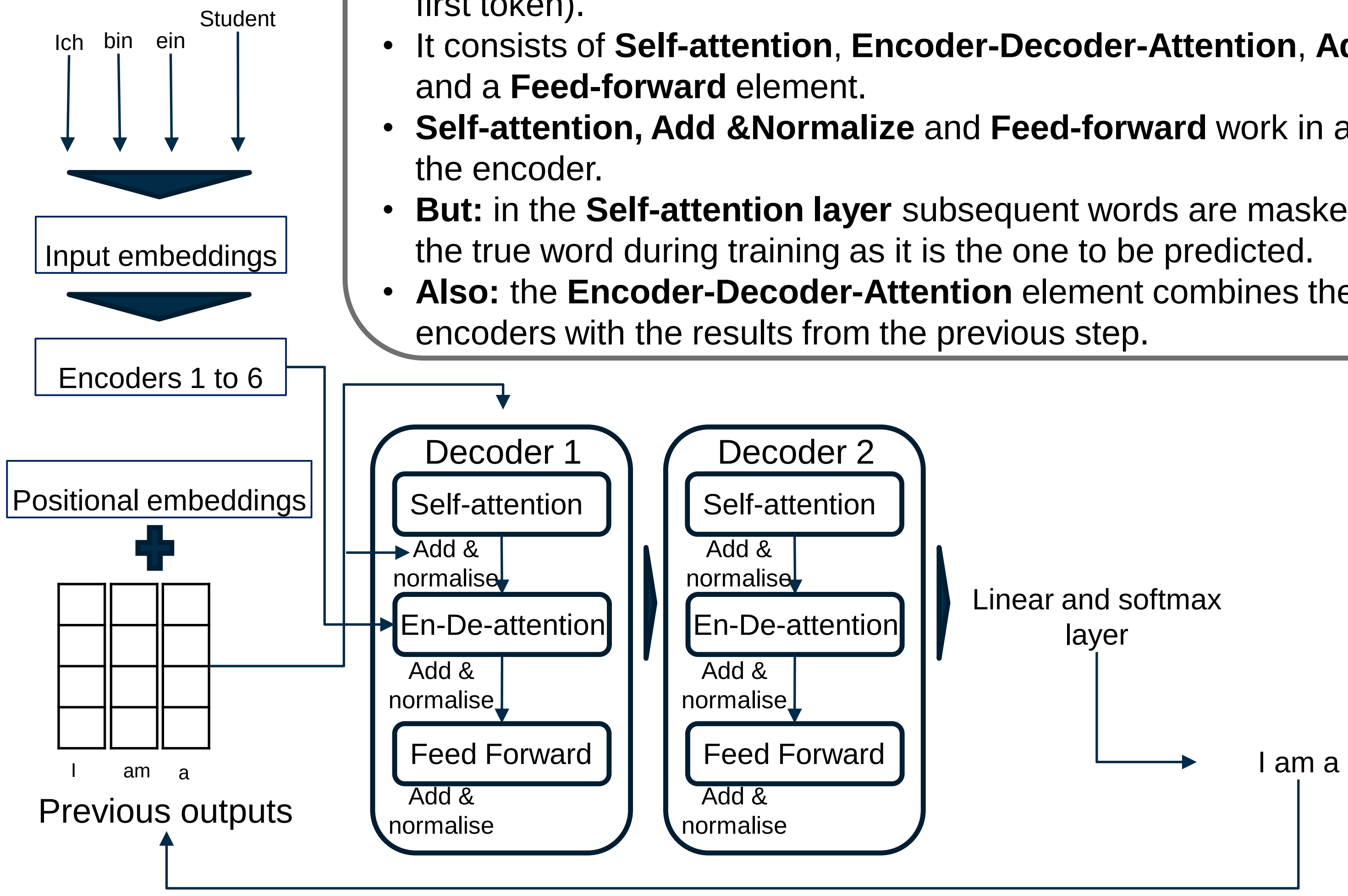
Add &Normalize: Adds results to previous steps' results and normalises them for better stability of the feed-forward network.
Feed-forward: A one layer neural network model that produces an embedding for each input token ➔ can be parallelised.



Transformers: Decoders

How does the decoder work?

- Its input consists of the already determined output tokens (special case for the first token).
- It consists of **Self-attention**, **Encoder-Decoder-Attention**, **Add &Normalize** and a **Feed-forward** element.
- **Self-attention, Add &Normalize** and **Feed-forward** work in a similar way as the encoder.
- **But:** in the **Self-attention layer** subsequent words are masked to avoid seeing the true word during training as it is the one to be predicted.
- **Also:** the **Encoder-Decoder-Attention** element combines the output from the encoders with the results from the previous step.



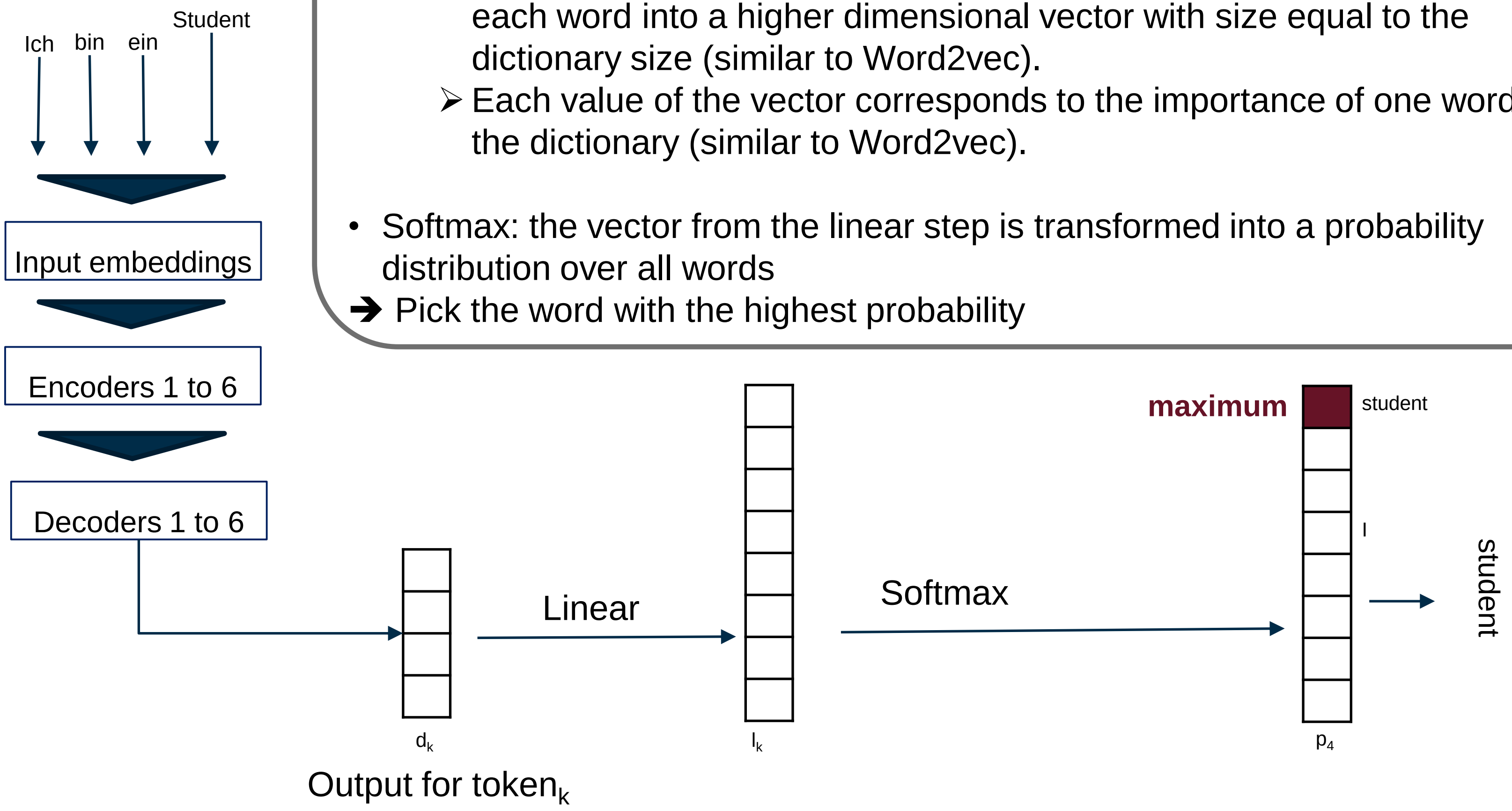
Transformers: Linear and Softmax layer

How do we map the output from the final decoder to a word?

- Linear step:
  - ➢ A fully connected neural network that maps the decoder embedding for each word into a higher dimensional vector with size equal to the dictionary size (similar to Word2vec).
  - ➢ Each value of the vector corresponds to the importance of one word in the dictionary (similar to Word2vec).

- Softmax: the vector from the linear step is transformed into a probability distribution over all words

➔ Pick the word with the highest probability



Transformers: Final thoughts

Hochschule für Wirtschaft und Recht Berlin
Berlin School of Economics and Law

- The transformer architecture is very suitable for translation tasks.
- **But:** many NLP tasks aim at classification and require embeddings (not translated ones) as inputs to a classification model.

? How can we apply transformers to classification tasks?

Transformers: Final thoughts

Hochschule für Wirtschaft und Recht Berlin
Berlin School of Economics and Law

- The transformer architecture is very suitable for translation tasks.
- **But:** many NLP tasks aim at classification and require embeddings (not translated ones) as inputs to a classification model.

? How can we apply transformers to classification tasks?

- **One possibility:** use only decoders (OpenAI GPT-2, https://demo.allennlp.org/next-token-lm?text=AllenNLP%20is)
- **But:** then a word can only see the previous words and not the next one ➔ loss of information
- **Another possibility:** use only encoders
- **But:** then you can‘t train on next word prediction

➔ BERT

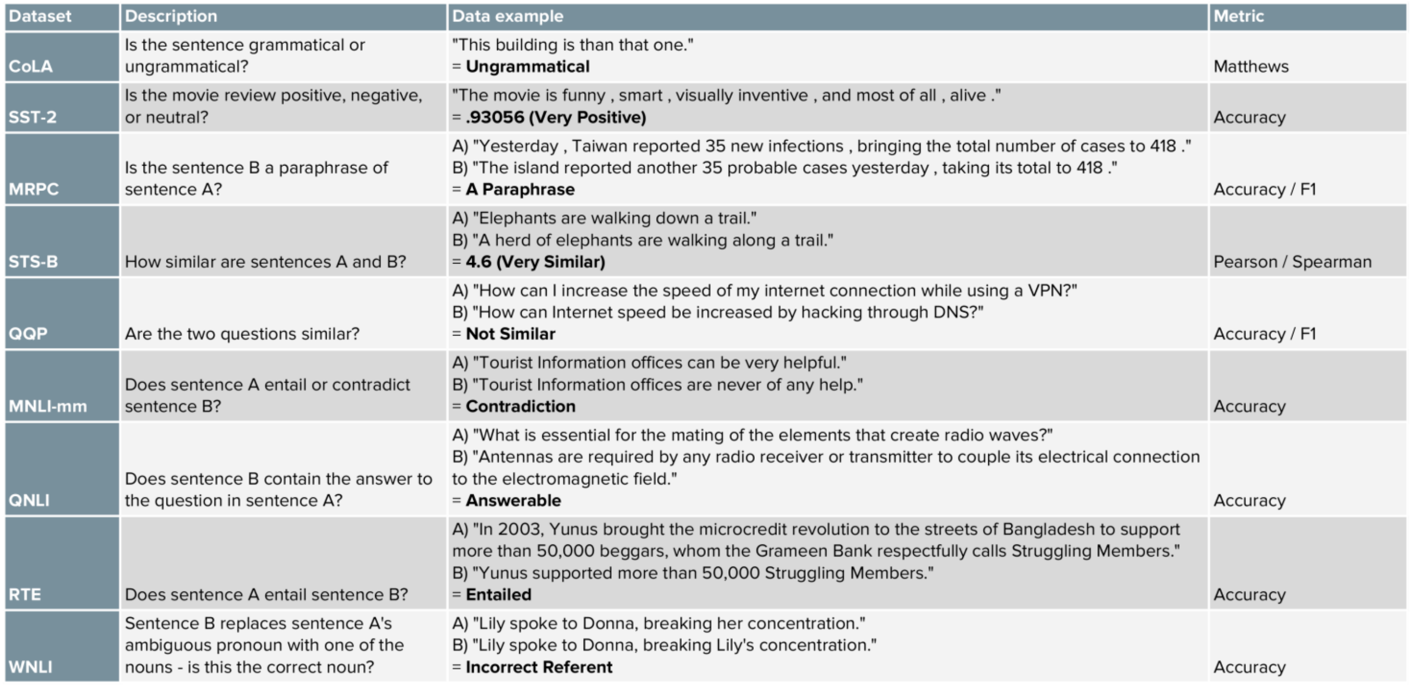
Transformers: Bidirectional Encoder Representations from Transformers (BERT)

- BERT was developed by Google in 2018 (Devlin et al. 2018).
- It was trained on unlabelled Wikipedia (2.5 Billion words) and BookCorpus (800 M words) data using TPUs for 4 days.
- BERT-Base (12-layer, 768-hidden) and BERT-Large (24-layer, 1024-hidden)
- The training was done by: 1) masking randomly words in the text and predicting them and 2) next sentence prediction.
- The pretrained models can be fine-tuned for downstream tasks such as sentiment analysis.
- Some of BERT variants achieve better aggregated performance than humans on the NLP GLUE tasks.

Devlin, Jacob; Chang, Ming-Wei; Lee, Kenton; Toutanova, Kristina (2018): BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. Available at http://arxiv.org/pdf/1810.04805v2.

GLUE

- **Main idea:** create a set of tasks so that a model can evaluated on all of them ➔ important to determine the genarl language understanding of the model

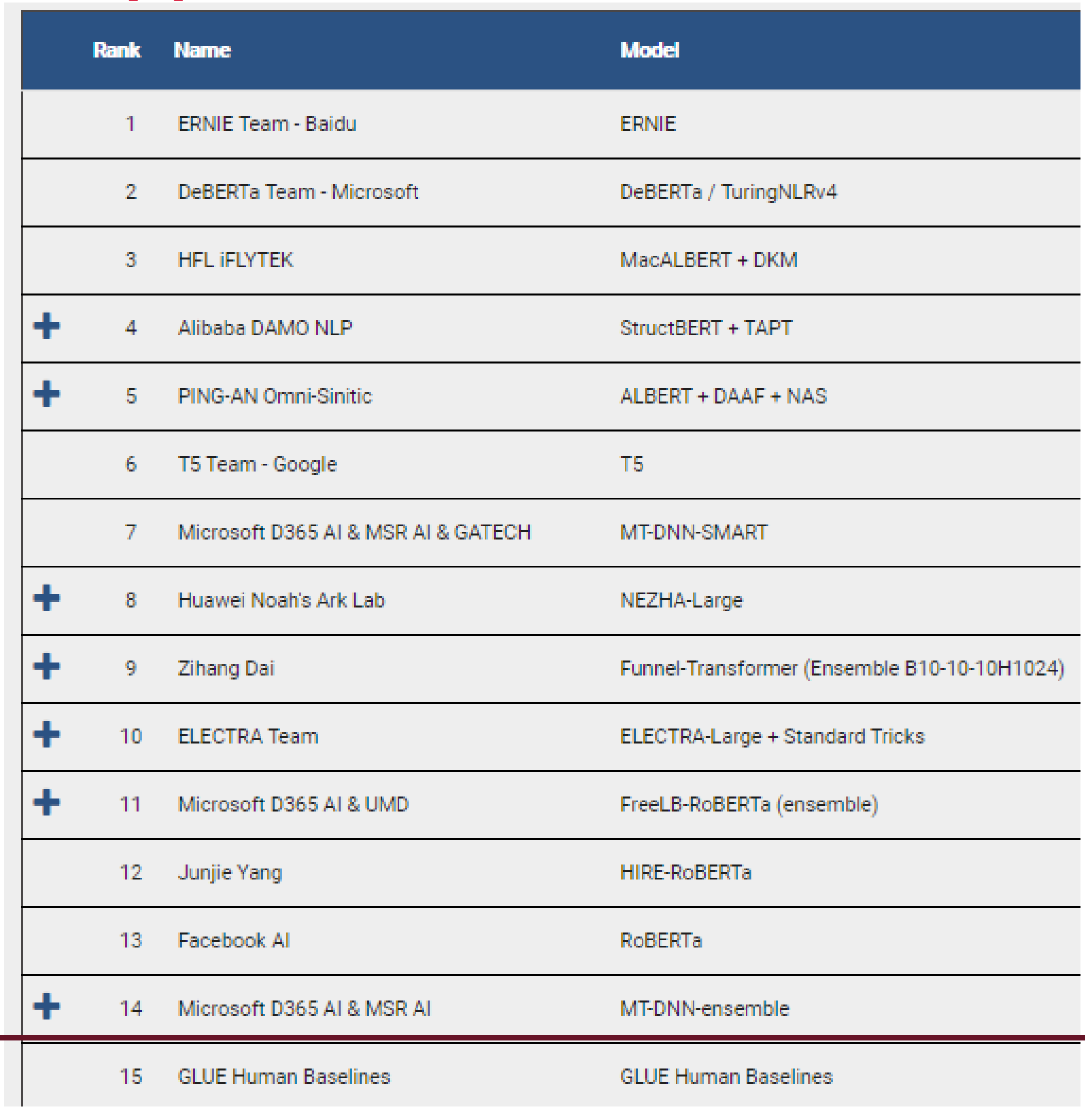


Dataset	Description	Data example	Metric
CoLA	Is the sentence grammatical or ungrammatical?	"This building is than that one." = **Ungrammatical**	Matthews
SST-2	Is the movie review positive, negative, or neutral?	"The movie is funny , smart , visually inventive , and most of all , alive ." = **.93056 (Very Positive)**	Accuracy
MRPC	Is the sentence B a paraphrase of sentence A?	A) "Yesterday , Taiwan reported 35 new infections , bringing the total number of cases to 418 ." B) "The island reported another 35 probable cases yesterday , taking its total to 418 ." = **A Paraphrase**	Accuracy / F1
STS-B	How similar are sentences A and B?	A) "Elephants are walking down a trail." B) "A herd of elephants are walking along a trail." = **4.6 (Very Similar)**	Pearson / Spearman
QQP	Are the two questions similar?	A) "How can I increase the speed of my internet connection while using a VPN?" B) "How can Internet speed be increased by hacking through DNS?" = **Not Similar**	Accuracy / F1
MNLI-mm	Does sentence A entail or contradict sentence B?	A) "Tourist Information offices can be very helpful." B) "Tourist Information offices are never of any help." = **Contradiction**	Accuracy
QNLI	Does sentence B contain the answer to the question in sentence A?	A) "What is essential for the mating of the elements that create radio waves?" B) "Antennas are required by any radio receiver or transmitter to couple its electrical connection to the electromagnetic field." = **Answerable**	Accuracy
RTE	Does sentence A entail sentence B?	A) "In 2003, Yunus brought the microcredit revolution to the streets of Bangladesh to support more than 50,000 beggars, whom the Grameen Bank respectfully calls Struggling Members." B) "Yunus supported more than 50,000 Struggling Members." = **Entailed**	Accuracy
WNLI	Sentence B replaces sentence A's ambiguous pronoun with one of the nouns - is this the correct noun?	A) "Lily spoke to Donna, breaking her concentration." B) "Lily spoke to Donna, breaking Lily's concentration." = **Incorrect Referent**	Accuracy

https://docs.google.com/spreadsheets/d/1BrOdjJgky7FfeiwC_VDURZuRPUFUAz_jfczPPT35P00/edit#gid=70328292

Hochschule für Wirtschaft und Recht Berlin
Berlin School of Economics and Law

GLUE (2)

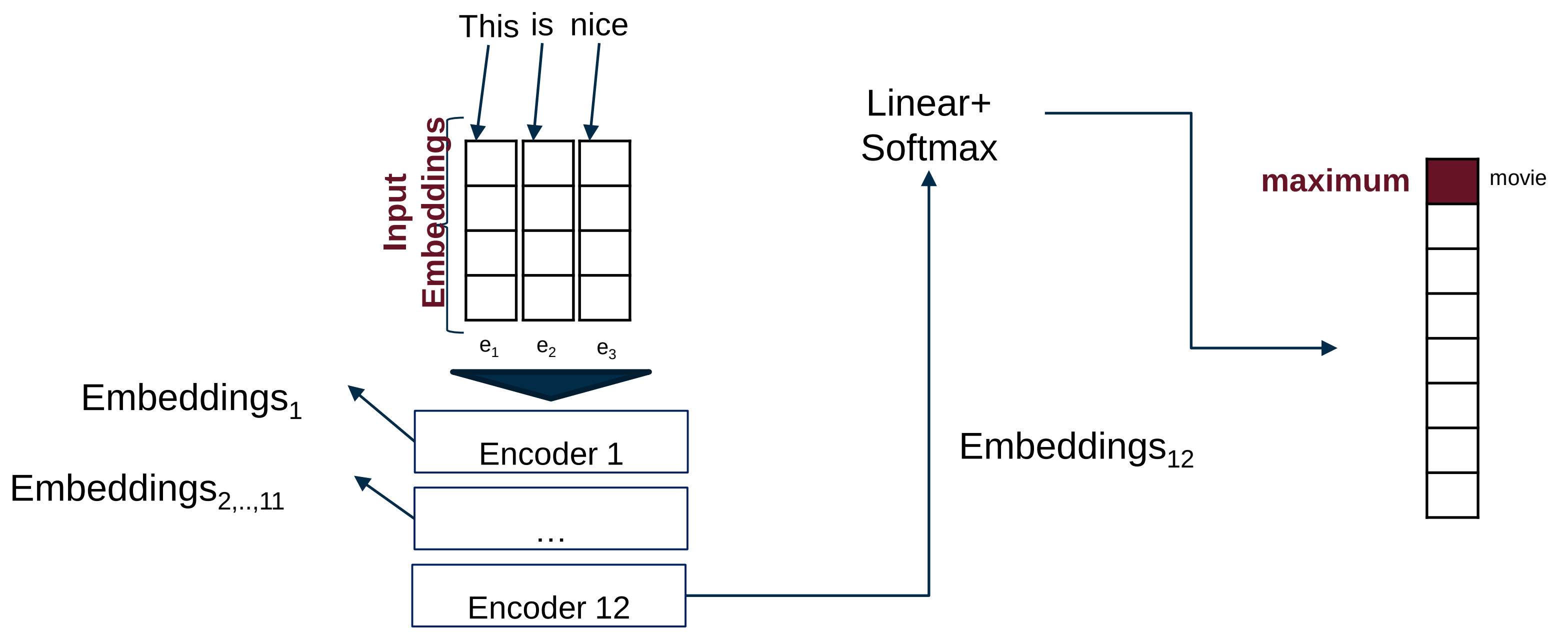


	Rank	Name	Model
	1	ERNIE Team - Baidu	ERNIE
	2	DeBERTa Team - Microsoft	DeBERTa / TuringNLRv4
	3	HFL iFLYTEK	MacALBERT + DKM
+	4	Alibaba DAMO NLP	StructBERT + TAPT
+	5	PING-AN Omni-Sinitic	ALBERT + DAAF + NAS
	6	T5 Team - Google	T5
	7	Microsoft D365 AI & MSR AI & GATECH	MT-DNN-SMART
+	8	Huawei Noah's Ark Lab	NEZHA-Large
+	9	Zihang Dai	Funnel-Transformer (Ensemble B10-10-10H1024)
+	10	ELECTRA Team	ELECTRA-Large + Standard Tricks
+	11	Microsoft D365 AI & UMD	FreeLB-RoBERTa (ensemble)
	12	Junjie Yang	HIRE-RoBERTa
	13	Facebook AI	RoBERTa
+	14	Microsoft D365 AI & MSR AI	MT-DNN-ensemble
	15	GLUE Human Baselines	GLUE Human Baselines

https://gluebenchmark.com/leaderboard

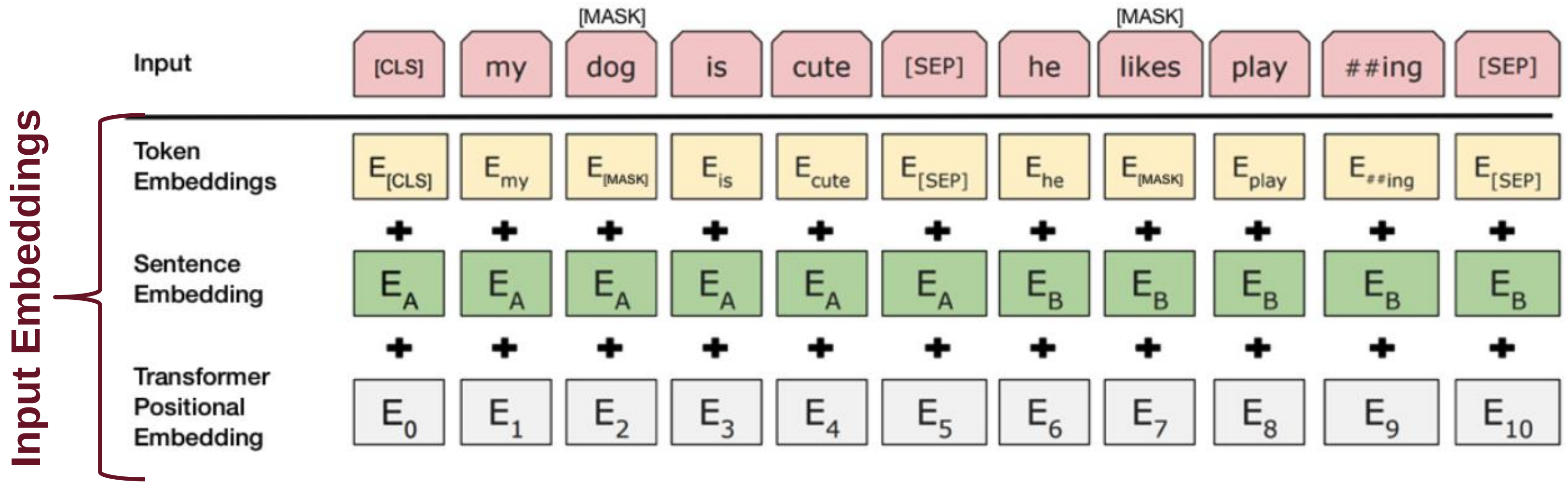
BERT (2)

- BERT consists of a trained Transformer encoder stack (12 encoders in BERT-base) i.e. no decoder.
- The output from the last encoder can then be used for downstream tasks such as classification.



BERT (3)

- BERT uses as input **WordPiece** embeddings, where words like “playing” are split into “play” and “#ing”.
- **Token embeddings** are the dictionary IDs for each token in a text e.g. “play” ➔ 103
- **Sentence embeddings** distinguish between sentences (relevant only for training on sentences).
- **Positional embeddings** indicate the position of each word



[MASK]
[MASK]
Input
[CLS] my dog is cute [SEP] he likes play ##ing [SEP]
Input Embeddings
Token Embeddings
$E_{[CLS]}$ E_{my} $E_{[MASK]}$ E_{is} E_{cute} $E_{[SEP]}$ E_{he} $E_{[MASK]}$ E_{play} $E_{\#\#ing}$ $E_{[SEP]}$
Sentence Embedding
E_A E_A E_A E_A E_A E_A E_B E_B E_B E_B E_B
Transformer Positional Embedding
E_0 E_1 E_2 E_3 E_4 E_5 E_6 E_7 E_8 E_9 E_{10}

Devlin, Jacob; Chang, Ming-Wei; Lee, Kenton; Toutanova, Kristina (2018): BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. Available at http://arxiv.org/pdf/1810.04805v2.

Summary and Outlook

Summary:

- Embeddings can be used to represent words in a continuous vector space.
- The Word2vec embeddings approach uses the context of the word to train its representation so that contextually similar words are close together in the vector space.
- Word (or document) similarity can be computed with the cosine similarity.
- Contextual embeddings such as ELMo are calculated separately for each document in order to incorporate words with multiple meanings.
- Transformer models have revolutionized the field of NLP modelling. Some of BERT variants achieve better aggregated performance than humans on the GLUE tasks.
- **Outlook:** Structured text representation can be used as input for classification and clustering algorithms.

Questions?

Exercise 4

In a minute, six break-out rooms will be created. Choose the room that corresponds to your group in Moodle e.g. Room 1= Group 1. In your project group discuss and document the solution for Exercise 4 (in Moodle).