UNIVERSITÄT LEIPZIG

Toolbox CSS
– Transformers // GPT, BERT, NLI, BERTopic

NSG SR 423, 10/12/2024

Felix Lennert, M.Sc.

OUTLINE

- Motivation
- Transformers
- How they work and what they can do
  - GPT
  - BERT
  - NLI
  - BERTopic

BOW HYPOTHESIS

Sentence 1: “This is a hell of a movie”

Sentence 2: “This movie is hell”

Preprocessing (stop word removal, word order)

Sentence/Token	movie	hell		
1	1	1	⇨	Negative
2	1	1	⇨	Negative

BERT

Sentence 1: “This is hell of a movie”

Sentence 2: “This movie is hell”

bert-base-uncased; trained on ~2,000 labeled IMDb reviews (~4min on MacBook Pro (2021, 16GB RAM, M1 Pro)

```
>>> predict(model, "this is a hell of a movie", tokenizer)
1
>>> predict(model, "this movie is hell", tokenizer)
0
```

Positive

Negative

BERTOPIC

Survey conducted in UK, Germany, Sweden

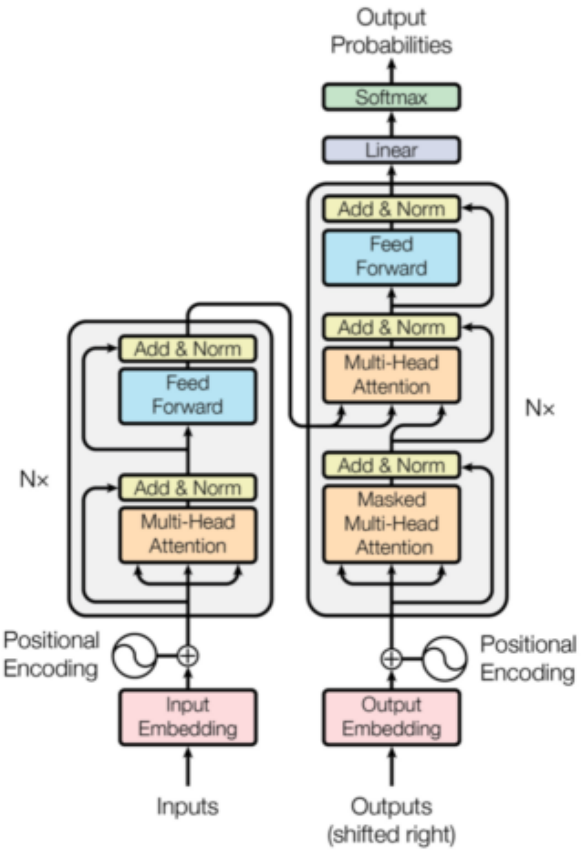
Question: how do people compare themselves to their parents?

Q17 str	topic fct(12)
Meine Kinder haben mehr als ich damals	Family, Partnership, Children
Jag är sjukpensionär,det var inte någon utav dem.	Uncertainty, Stability
Sicher die Lebensqualität	Wellbeing, Health, Quality of Life
Mehr Geld	Money, Income, Wealth
Mer utbildning, bättre avlönat jobb	Education
Mitt liv präglas hela tiden av oro och rädsla för framt…	Money, Income, Wealth
Meine Eltern konnten ihr Dasein relativ genießen	Family, Partnership, Children
Färdig med utbildning tidigare.	Education
Lebensstandard, Wohnung, Reisen, Freizeit	Money, Income, Wealth
Min psykiska ohälsa och höga krav på mig själv	Wellbeing, Health, Quality of Life

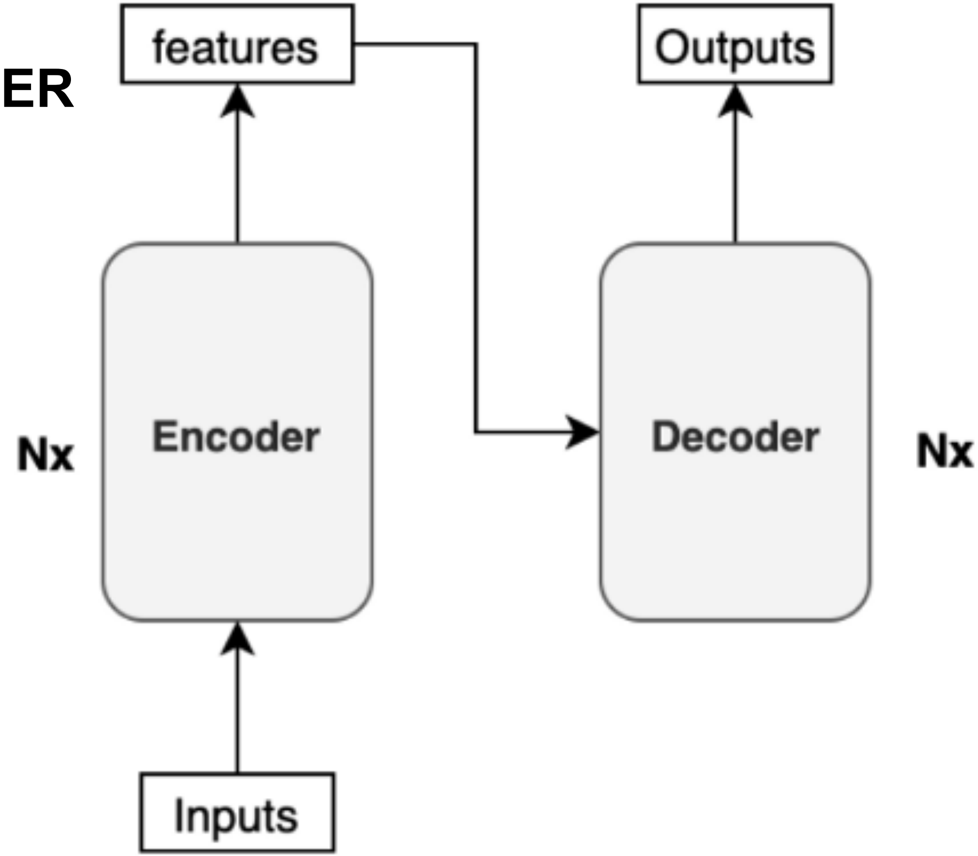
Problem: short answers; different languages

Solution: BERTopic with *distiluse-base-multilingual-cased-v2* – “understands” 100+ languages, can handle short answers

TRANSFORMER



TRANSFORMER

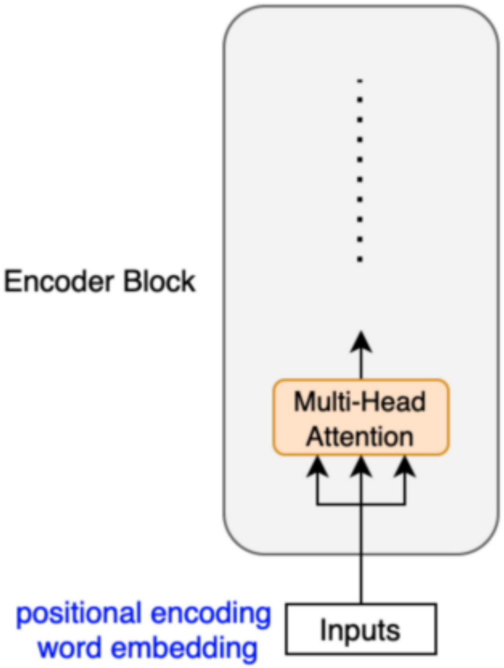


STEP 1: TOKENIZATION

- Here, each text is of fixed length – it needs to get padded (e.g., by including <pad>, <pad>, …, <pad>)
- "." might become <EOS>
- Also, there are length limits – e.g., BERT takes up to 512 tokens
- tokens also look a bit different, they break up the words a bit
- finally, tokens are replaced by their vectors (including their position)

```
>>> tokenizer = BertTokenizer.from_pretrained('bert-base-uncased')
>>> tokens = tokenizer.tokenize('This is what tokenization in BERT looks like.')
>>> print(tokens)
['this', 'is', 'what', 'token', '##ization', 'in', 'bert', 'looks', 'like', '.']
```

STEP 2: ATTENTION (IS ALL YOU NEED)



Attention Is All You Need

Ashish Vaswani* Google Brain avaswani@google.com
Noam Shazeer* Google Brain noam@google.com
Niki Parmar* Google Research nikip@google.com
Jakob Uszkoreit* Google Research usz@google.com
Llion Jones* Google Research llion@google.com
Aidan N. Gomez* † University of Toronto aidan@cs.toronto.edu
Łukasz Kaiser* Google Brain lukaszkaiser@google.com
Illia Polosukhin* ‡ illia.polosukhin@gmail.com

Goal: relate words to each other, impute context

STEP 2: ATTENTION (IS ALL YOU NEED)

Imagine you're at a crowded cocktail party and you have trouble hearing your friend. When you're talking to someone, you're not just listening to their words in isolation – you're…

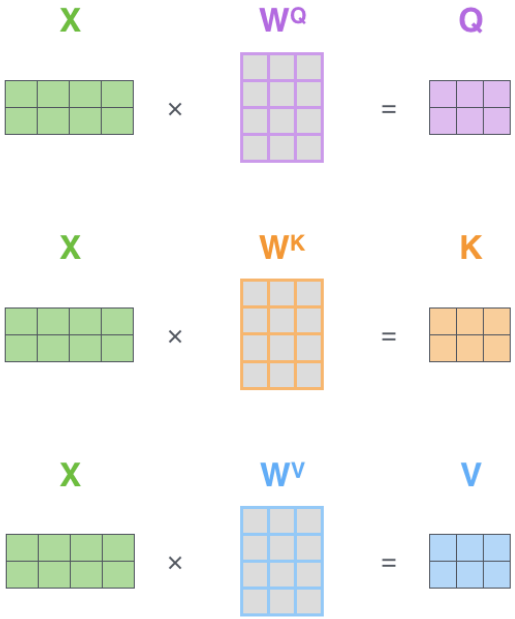
- Paying attention to their tone
- Watching their gestures
- Connecting their current sentence to what they said earlier
- Relating it to the ongoing conversation context
- …

STEP 2: ATTENTION (IS ALL YOU NEED)

$$Attention(Q, K, V) = softmax(\frac{QK^T}{\sqrt{d}})V$$

- Going through each word and creating three distinct versions of its vector: Query, Key, Value (Q,K,V):
  - Q: Query – the vector that gets compared – W_q * vector (W_q is learned in the model training process)
  - K: Key – the vector it gets compared to W_k * vector (W_k is learned in the model training process)
  - V: Value – containing “information” on the original vector – W_v * vector (W_v is learned in the model training process)
  - d = dimensions of vectors, helps with stability

STEP 2: ATTENTION (IS ALL YOU NEED)



STEP 2: ATTENTION (IS ALL YOU NEED)

$$Attention(Q, K, V) = softmax(\frac{QK^T}{\sqrt{d}})V$$

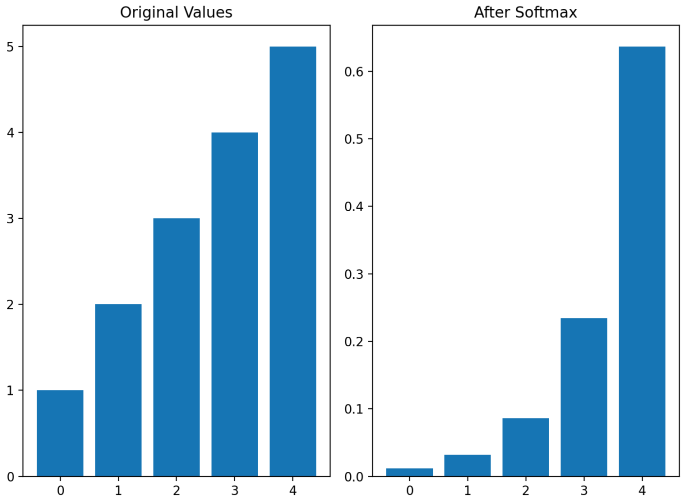
- QK^T: comparing the Query against the Key – “raw” similarity score
- $\sqrt{d}$: square root of the dimensions of vectors, helps with stability – numerator can take quite high values

STEP 2: ATTENTION (IS ALL YOU NEED)

$$Attention(Q,K,V) = softmax(\frac{QK^T}{\sqrt{d}})V$$

$$\text{softmax}(x_i) = \frac{e^{x_i}}{\sum_{j=1}^{n} e^{x_j}}$$

=> gives more weight to higher values, decreases weight for lower values; values sum up to 1



STEP 2: ATTENTION (IS ALL YOU NEED)

- “The bank is by the river”
- Q: “bank” looking for context; K: comparing with all other words
- Raw scores might be:
  “the”: 0.1; “is”: 0.2; “by”: 0.3; “river”: 0.8 (highest score because it helps clarify the meaning)
- After division by $\sqrt{d_k}$: Scales these scores down to reasonable numbers
- softmax: conversion to attention probabilities:
  "the": 5%; ”is": 10%; ”by": 15%; ”river": 70%

=> Now we know to pay most attention to "river" when understanding "bank"

STEP 2: ATTENTION (IS ALL YOU NEED)

$$Attention(Q, K, V) = softmax(\frac{QK^T}{\sqrt{d}})V$$

Finally: introduce V, inserts “meaning”

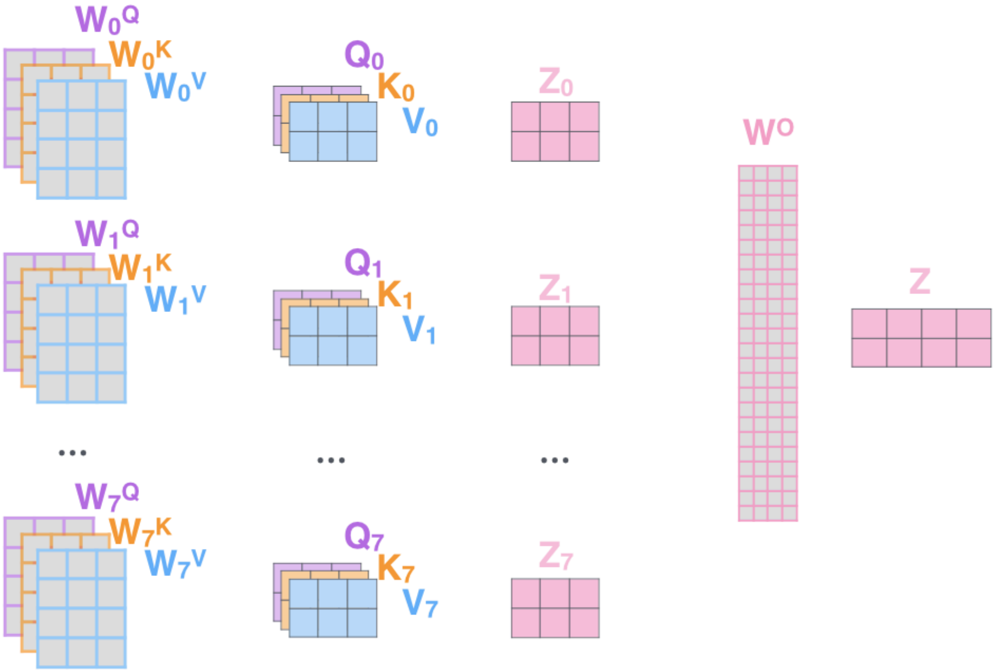
STEP 2: ATTENTION (IS ALL YOU NEED)

Imagine you're at a crowded cocktail party and you have trouble hearing your friend. When you're talking to someone, you're not just listening to their words in isolation – you're…

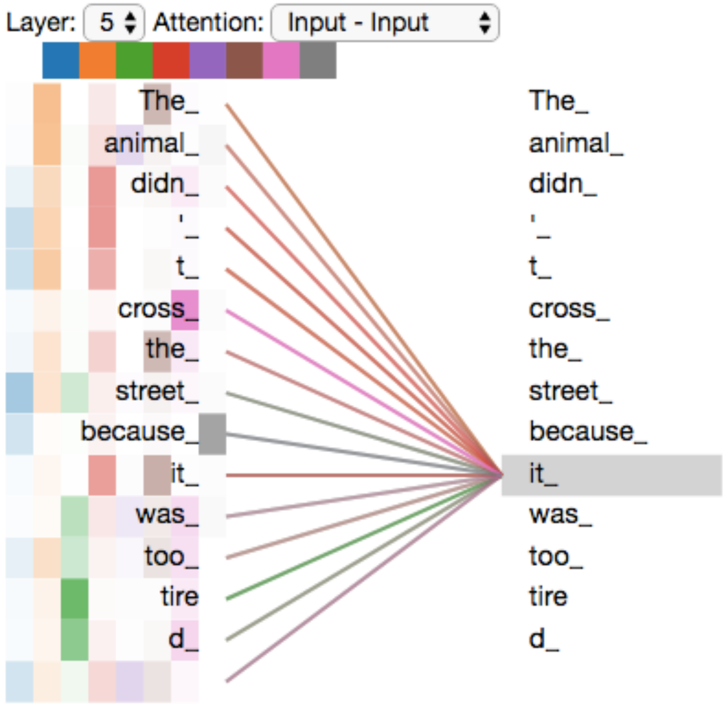
- Paying attention to their tone
- Watching their gestures
- Connecting their current sentence to what they said earlier
- Relating it to the ongoing conversation context
- …

=> Attention here is "multi-head attention" – different heads look at different aspects – tap into different embedding spaces

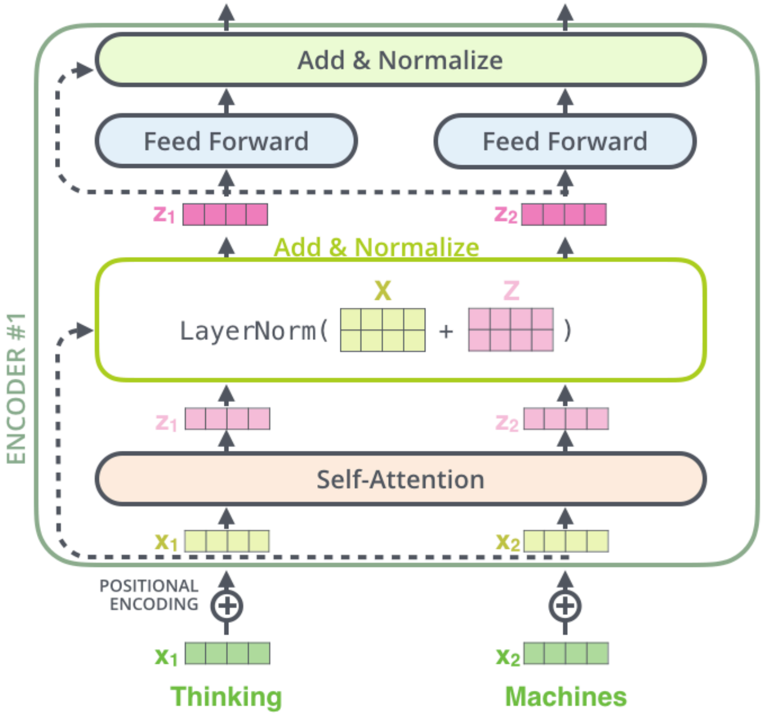
STEP 2: ATTENTION (IS ALL YOU NEED)



STEP 2: ATTENTION (IS ALL YOU NEED)



STEP 3: POSITION-WISE FEED FORWARD



STEP 3: POSITION-WISE FEED FORWARD

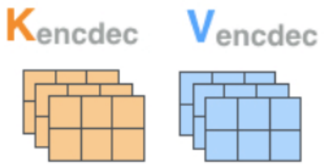
Two linear transformations with Rectified Linear Unit (ReLU) in between

- Linear transformation #1: expands each individual vector – e.g., from 512 dimensions to 2048
- ReLU: sets negative values to 0, leaves positive values as is
  => ifelse(x >= 0){x}else{0}
- Linear transformation #2: back to original dimensionality

ENCODER OUTPUT

- Embeddings with context – the model has “read” the input text
- This can be used for different tasks:
  - sequence classification head (BERT) => feeds these vectors into a “linear layer” (assigning probability to each class)
  - also: regression head (BERT) => for continuous values
  - token classification head (BERT) => assigns one label to each token (e.g., named-entity recognition)
  - …
  - translation => feed forward to **decoder to generate new text**
- Note that GPT does not use the encoder in the first place, only the decoder

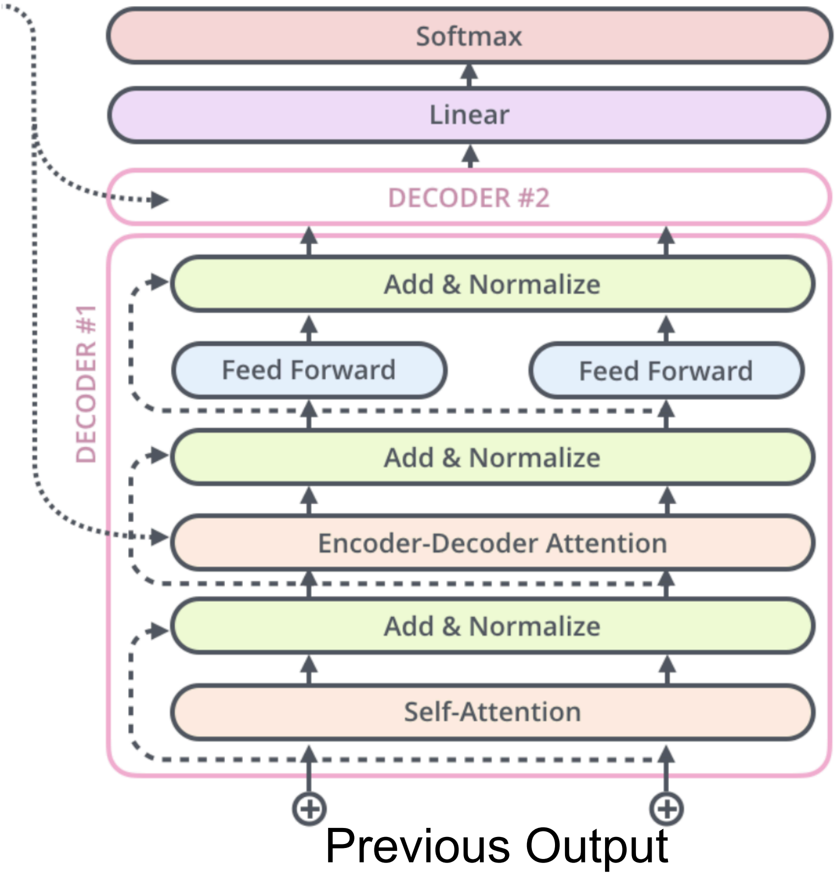
DECODER



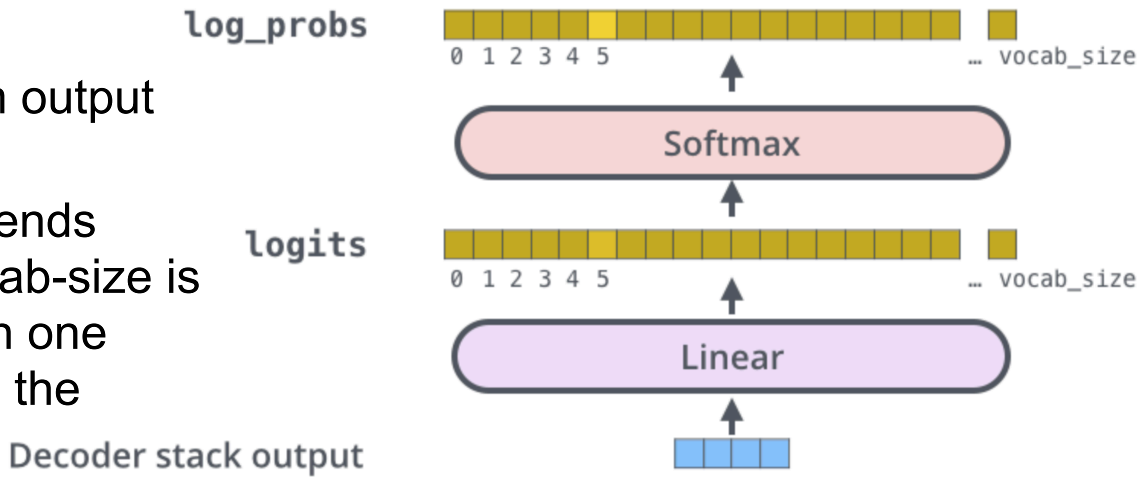
Works similarly to encoder, but:

- **Encoder** sees each word in input – **before and after the focal word**
- **Decoder** only sees the words that have been generated **before the focal word**

=> goal for decoder: predict next word



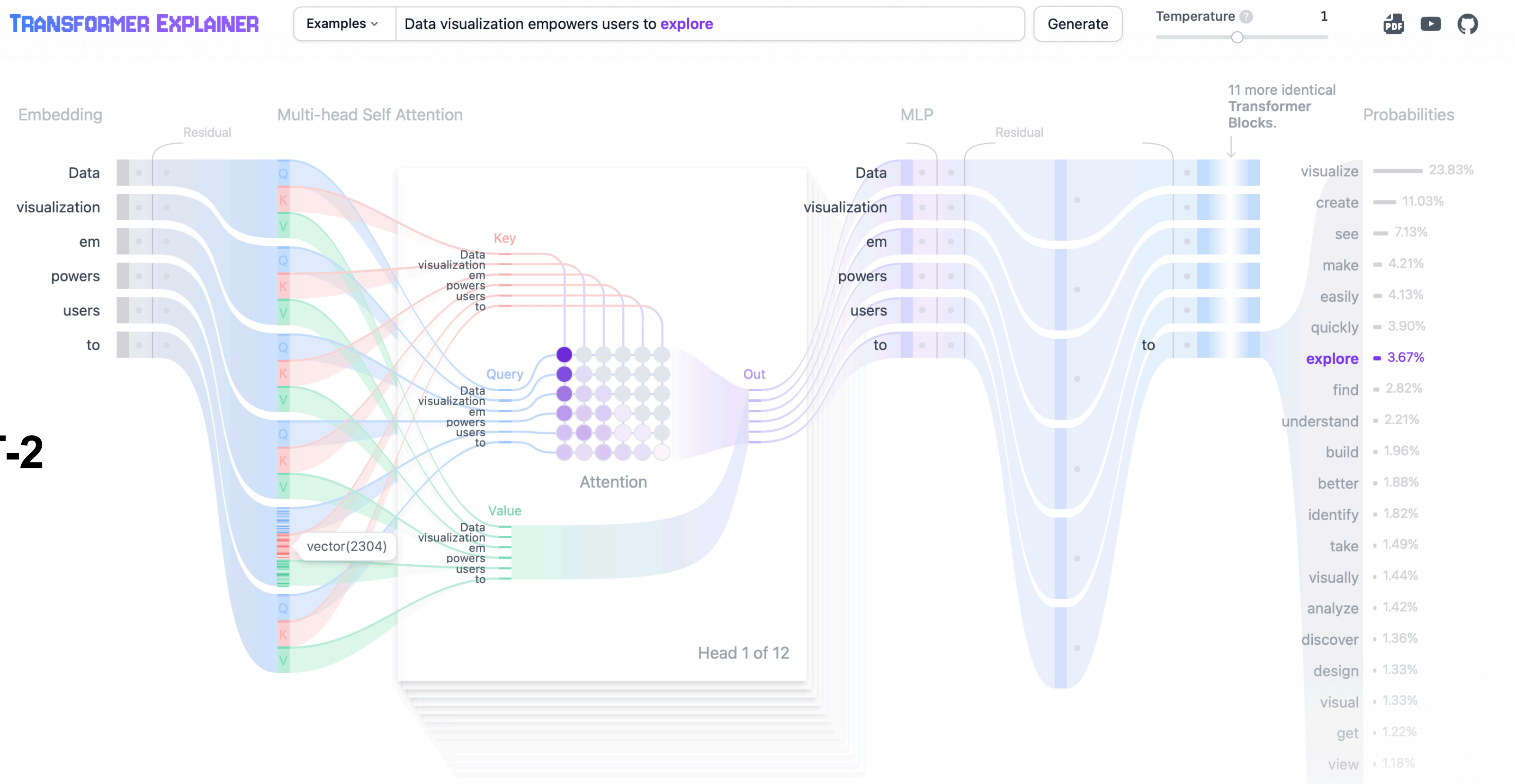
DECODER



For word-prediction: transform output into probabilities

- *linear layer* (neural net) extends dimensionality – e.g., if vocab-size is 10k, we get 10k logits, each one corresponding to a token in the vocabulary
- *softmax*: turns this into probabilities – token with highest probability is chosen

New Horizons – Transformer models | Decoder



GPT-2

Note: GPT =/= ChatGPT – GPT powers ChatGPT, but

CHATBOTS

They are powered by GPTs, but modified:

- Data Format: Conversations are formatted as alternating prompts and responses
  - Special tokens mark different speakers/roles
  - Example format:
    <HUMAN>: How do I make pasta?
    <ASSISTANT>: First, boil water...
    <HUMAN>: How long should I boil it?
    <ASSISTANT>: Typically 8-12 minutes...

  => Model learns to predict the next tokens given the conversation history

  => Particular focus on generating appropriate responses after human prompts
- Hence, it must learn:
  - Appropriate tone/style
  - Staying in character/role
  - Maintaining conversation context
  - Following instructions

HOW IS IT USED IN THE SOCIAL SCIENCES? – GPT

MACHINE BIAS

How do Generative Language Models Answer Opinion Polls?

Julien Boelaert[1], Samuel Coavoux[2], Étienne Ollion[2], Ivaylo Petev[2], and Patrick Präg[2]

“Our results i) confirm that to date, **models cannot replace research subjects for opinion or attitudinal research**; ii) that **they display a strong bias on each question** (reaching only a small region of social space); and iii) that this bias varies randomly from one question to the other (reaching a different region every time).”

Leveraging AI for democratic discourse: Chat interventions can improve online political conversations at scale

Lisa P. Argyle, Christopher A. Bail, Ethan C. Busby, +4, and David Wingate Authors Info & Affiliations

We develop an AI chat assistant that makes real-time, evidence-based suggestions for messages in divisive online political conversations. In a randomized controlled trial, we show that when one participant in a conversation had access to this assistant, **it increased their partner's reported quality of conversation and both participants' willingness to grant political opponents space to express and advocate their views in the public sphere**. Participants had the ability to accept, modify, or ignore the AI chat assistant's recommendations. Notably, participants' policy positions were unchanged by the intervention.

Large language models empowered agent-based modeling and simulation: a survey and perspectives

Chen Gao, Xiaochong Lan, Nian Li, Yuan Yuan, Jingtao Ding, Zhilun Zhou, Fengli Xu & Yong Li

Promises:
- Human-like behavior simulation through natural language understanding
- Rich agent-to-agent and agent-environment interactions
- Potential for more sophisticated economic and social simulations

Shortcomings:
- High computational costs
- Reliability issues: inconsistent responses; hallucination
- Hard to control and validate agent behaviors
- Also: no standardized evaluation methods and benchmarks
- Safety and ethical concerns regarding biased or harmful outputs

Large Language Models Outperform Expert Coders and Supervised Classifiers at Annotating Political Social Media Messages

Petter Törnberg[1,2]

"these models are capable of zero-shot annotation based on instructions written in natural language, they obviate the need of large sets of training data"

"the task used is to identify the political affiliation of politicians based on a single X/ Twitter messages"

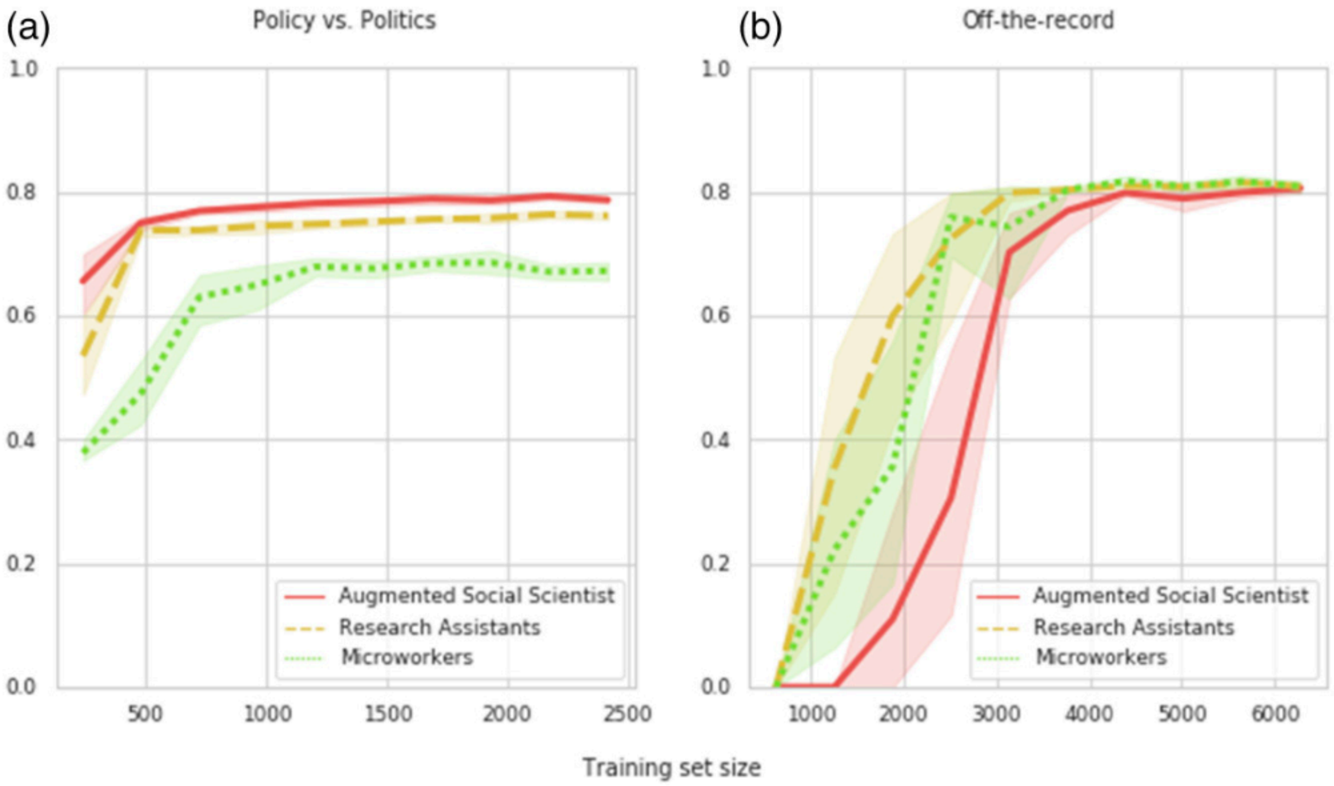
"The paper finds that GPT-4 achieves higher accuracy than both supervised models and human coders across all languages and country contexts. In the US context, it achieves an accuracy of 0.934 and an inter-coder reliability of 0.982."

The Augmented Social Scientist: Using Sequential Transfer Learning to Annotate Millions of Texts with Human-Level Accuracy

Salomé Do[1,2], Étienne Ollion[3], and Rubing Shen[2,3]

- Claim: LLMs lower the cost of annotation – less training examples required, hence “experts” can annotate small samples
- How do fewer, but better (i.e., more accurate/valid) annotations by experts (the researchers) augmented by LLM hold up against more but potentially biased annotations (all annotations made by research assistants)?
- How do training data generated by researchers hold up against training data generated by research assistants/ microworkers?
- Sequence extraction

	F1 – Policy vs. Politics	F1 – Off the record
Human – Microworkers	0.65	0.70
Human – Research assistants	**0.80**	**0.86**
Model without pre-training	0.67 [0.671, 0.673]	0.41 [0.390, 0.437]
Augmented social scientist (model with pre-training)	0.78 [0.781, 0.792]	0.82 [0.816, 0.834]


(a)
Policy vs. Politics
1.0
0.8
0.6
0.4
0.2
0.0
500
1000
1500
2000
2500
Augmented Social Scientist
Research Assistants
Microworkers
(b)
Off-the-record
1.0
0.8
0.6
0.4
0.2
0.0
1000
2000
3000
4000
5000
6000
Augmented Social Scientist
Research Assistants
Microworkers
Training set size

Politics as Usual? Measuring Populism, Nationalism, and Authoritarianism in U.S. Presidential Campaigns (1952–2020) with Neural Language Models

Bart Bonikowski, Yuchen Luo, and Oscar Stuhler

- Populism: hard to capture and multi-faceted concept
- "The relatively rare, polysemic, and variable frames in our study had previously been difficult to capture at scale because of the inadequacy of traditional machine learning methods and the shortcomings of dictionary-based approaches"

IN A SIMILAR VEIN: NATURAL LANGUAGE INFERENCE (NLI) – HYPOTHESIS TESTING

- NLI: determining the logical relationship between two pieces of text – a premise and a hypothesis
- Does the hypothesis…
  - …entail (logically follow from) the premise,
  - contradict the premise,
  - or is it neutral (neither entails nor contradicts)?

Example:

- Premise: “The cat is sleeping on the couch” | Hypothesis: "There is a cat in the house"
  => Relationship: ENTAILMENT (couches are in houses; if cat on couch, cat in house)
- Premise: “The cat is sleeping on the couch” | Hypothesis: "The cat is playing outside”
  => Relationship: CONTRADICTION (cats can’t play while sleeping)
- Premise: “The cat is sleeping on the couch” | Hypothesis: “The cat is dreaming”
  => Relationship: NEUTRAL (might or might not be dreaming while sleeping)

IN A SIMILAR VEIN: NATURAL LANGUAGE INFERENCE (NLI) – HYPOTHESIS TESTING

- NLI: determining the logical relationship between two pieces of text – a premise and a hypothesis
- Does the hypothesis…
  - …entail (logically follow from) the premise,
  - contradict the premise,
  - or is it neutral (neither entails nor contradicts)?

Potential use:

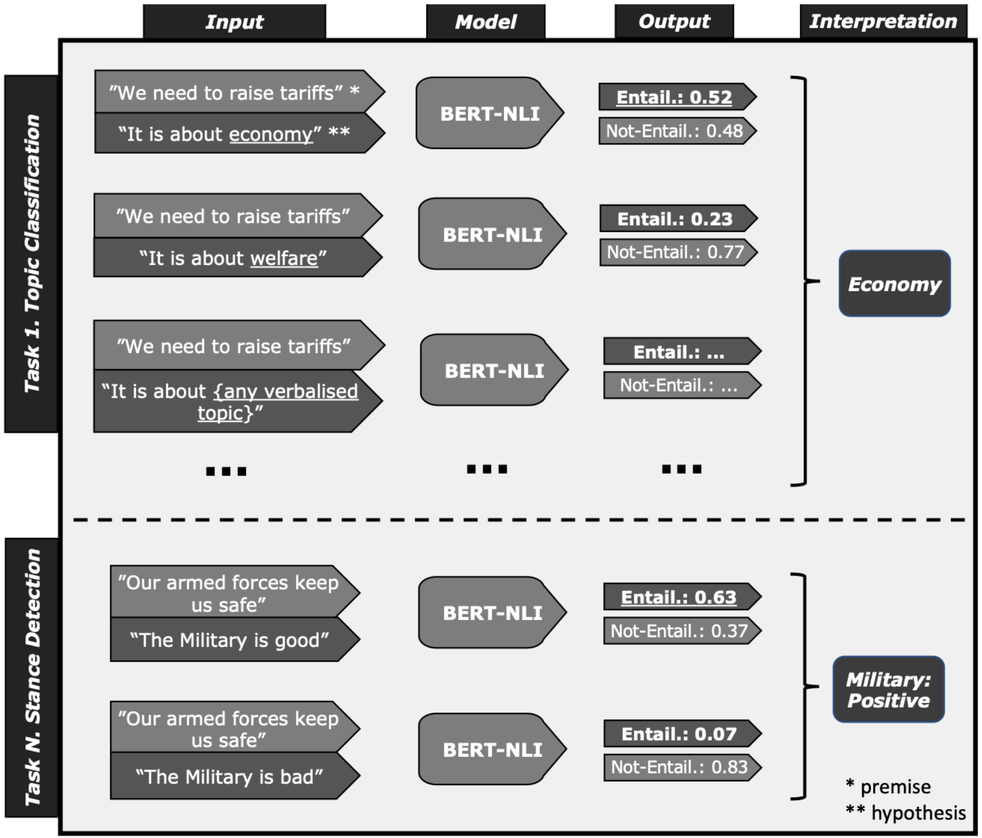
- Use it to classify text
- Example (current BA):
  - Premise: Titles of job ads
  - Hypothesis: The job ads contains gender-neutral language

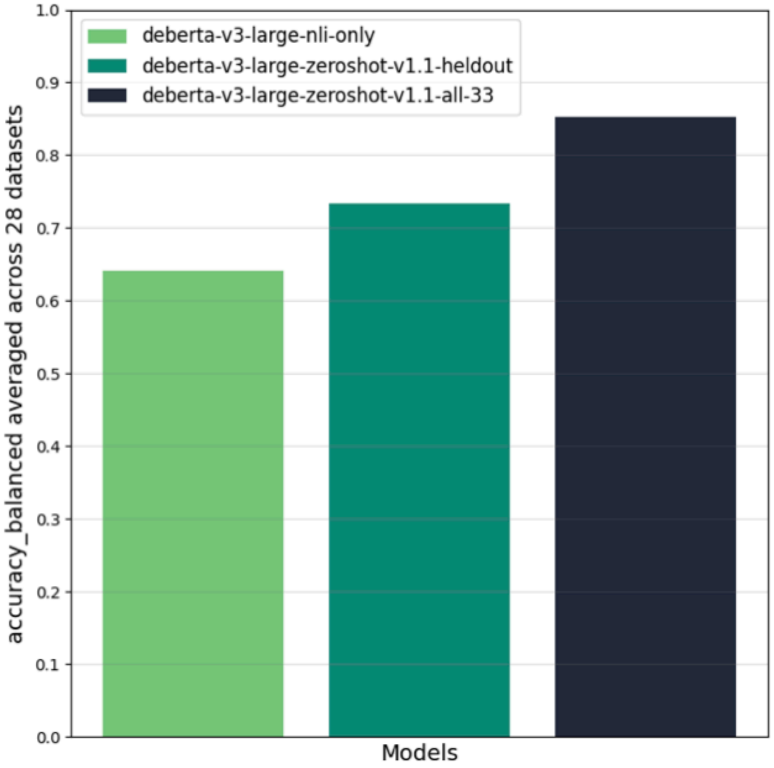
Building Efficient Universal Classifiers with Natural Language Inference

Moritz Laurer[‡‖], Wouter van Atteveldt[‡], Andreu Casas[†], Kasper Welbers[‡]
[‡] Vrije Universiteit Amsterdam
[†] University of London, Royal Holloway
[‖] Hugging Face
moritz@huggingface.co

Building Efficien ... age Inference

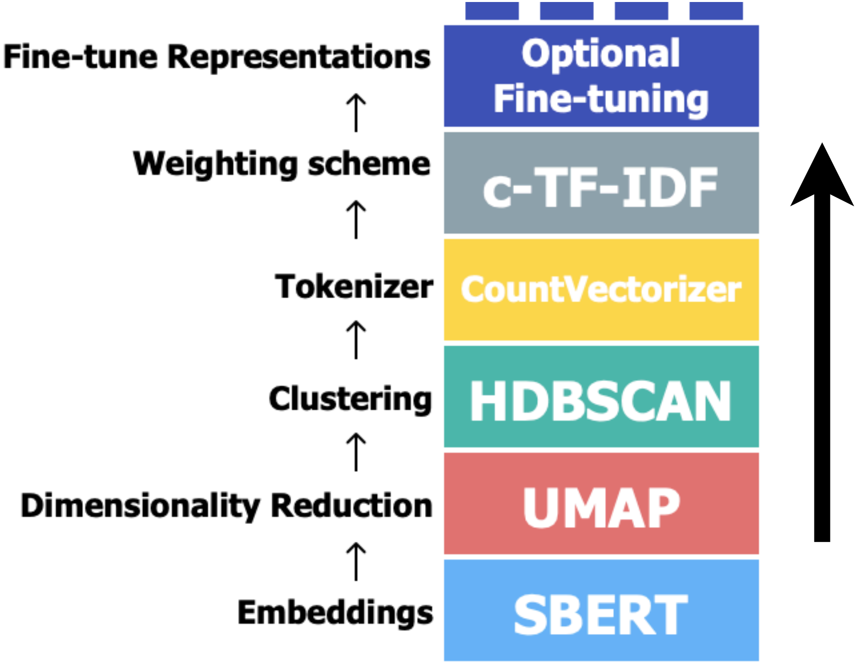
Moritz Laur ... Welbers‡





deberta-v3-zeroshot-v1.1-all-33: fine-tuned with up to 500 examples

HOW ABOUT UNSUPERVISED TASKS: BERTOPIC



Promise: flexible framework

- Can use different base models for, e.g., language understanding
- Possibilities:
  - Prime it with topics (seeded topic model)
  - Provide training examples (supervised topic model)
  - Not only use text, but, e.g., images (multi-modal topic modeling)
  - Model topics over time (dynamic topic modeling)

CONCLUSION

- Python!
- Computationally expensive (GPUs!)
  => environmental impact
  “the process of building and testing a final paper-worthy model required training 4,789 models over a six-month period. … it emitted more than 78,000 pounds and is likely representative of typical work in the field.”
  => flight to NYC and back: 5,000 pounds/person // FINETUNING IS LESS COSTLY
- Same principles as with “classic” methods apply (validate etc.)
- Usually: better performance though

MERCI

Felix Lennert
Institut für Soziologie

felix.lennert@uni-leipzig.de
www.uni-leipzig.de