AI and Language Models

Ion Androutsopoulos

Natural Language Processing Group
Department of Informatics
Athens University of Economics and Business

http://www.aueb.gr/users/ion/

Word embeddings of business terms

(produced with word2vec, then projected to 2D using UMAP)

Vectors (points) in **2D**:

y

∘ ⟨2,4⟩

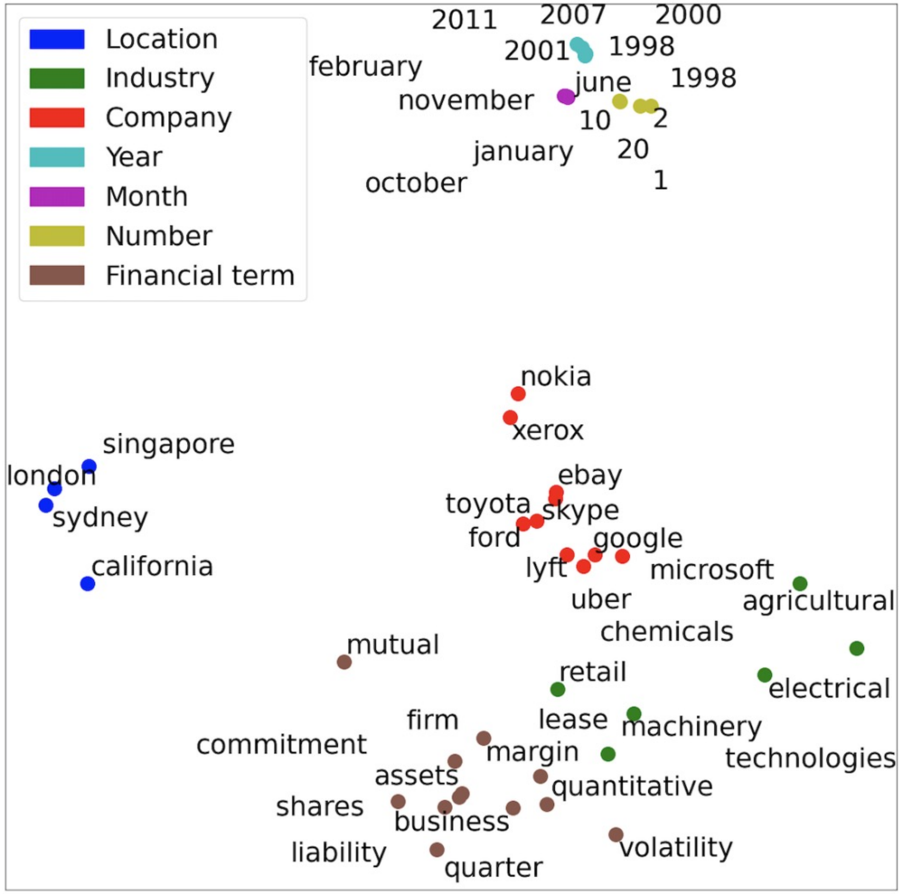
∘ ⟨3,2⟩

x

2D vector β

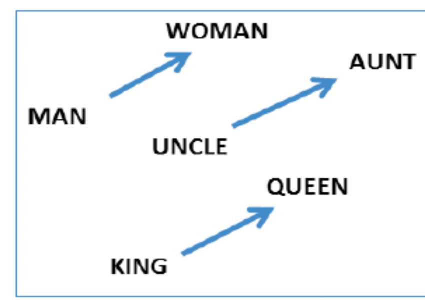
"dense layer" W $\beta = W\alpha$

300D vector α



Word embeddings are vectors (points), e.g., in a **300D** space.

They capture **relatedness**, **analogy**, …

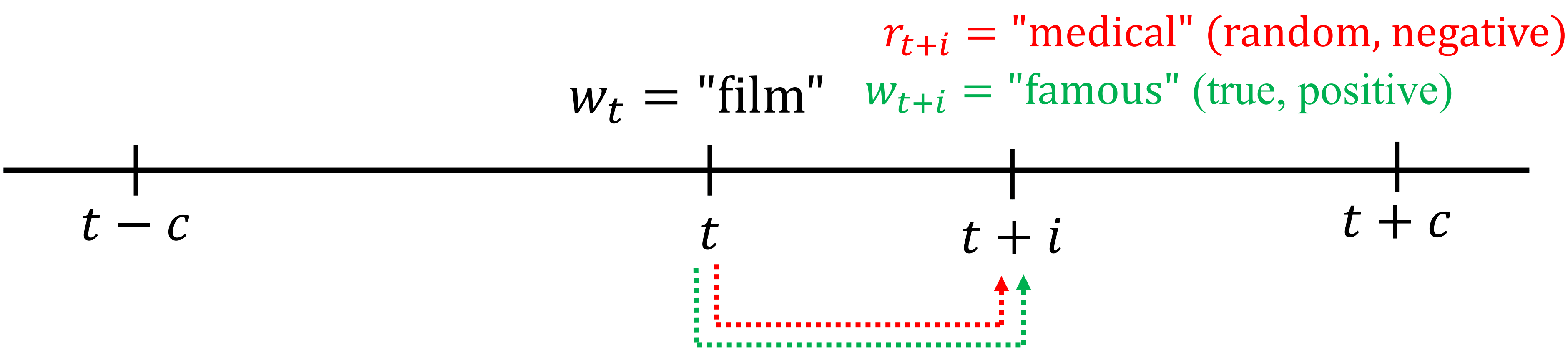


Large image from Lukas et al., "EDGAR-CORPUS: Billions of Tokens Make The World Go Round", EcoNLP workshop, EMLP 2021 (https://aclanthology.org/2021.econlp-1.2/). Small image from Mikolov et al., "Linguistic Regularities in Continuous Space Word Representations". NAACL 2013 (https://aclanthology.org/N13-1090/).

Word2Vec (skip-gram with negative sampling)

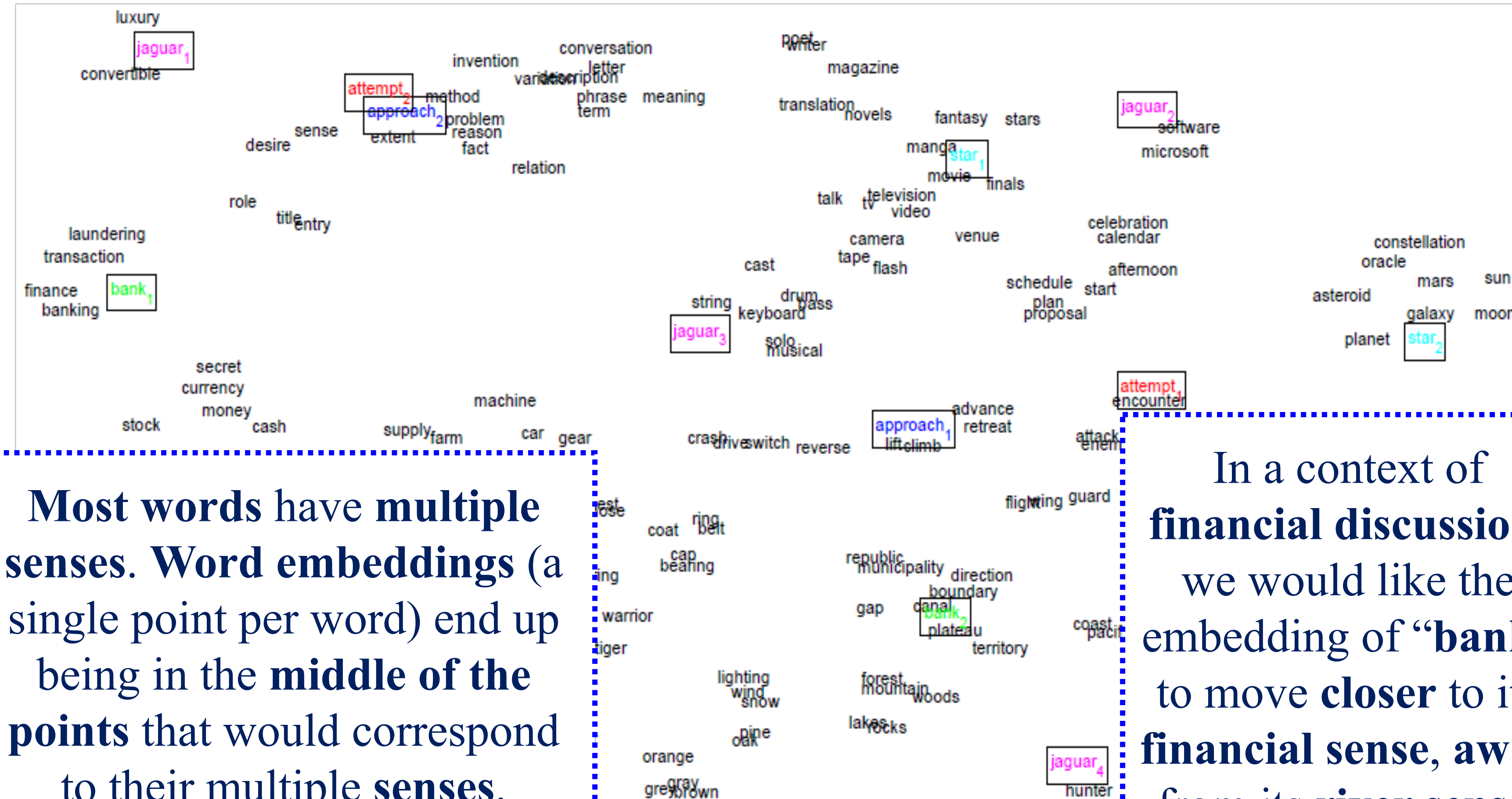
For each word w_t of the corpus, we construct positive (+) and negative (−) pairs, using the word w_{t+i} that actually occurs at position $t+i$, and a a random word r_{t+i} that does not actually occur at position $t+i$.

Intuition hiding some details: We modify slightly the word embeddings to bring w_t closer to w_{t+i} and move it away from r_{t+i}.



Word <u>sense</u> embeddings

(produced by a method that produces **dense, sense-specific** word embeddings, then **projected** to **2 dimensions**)



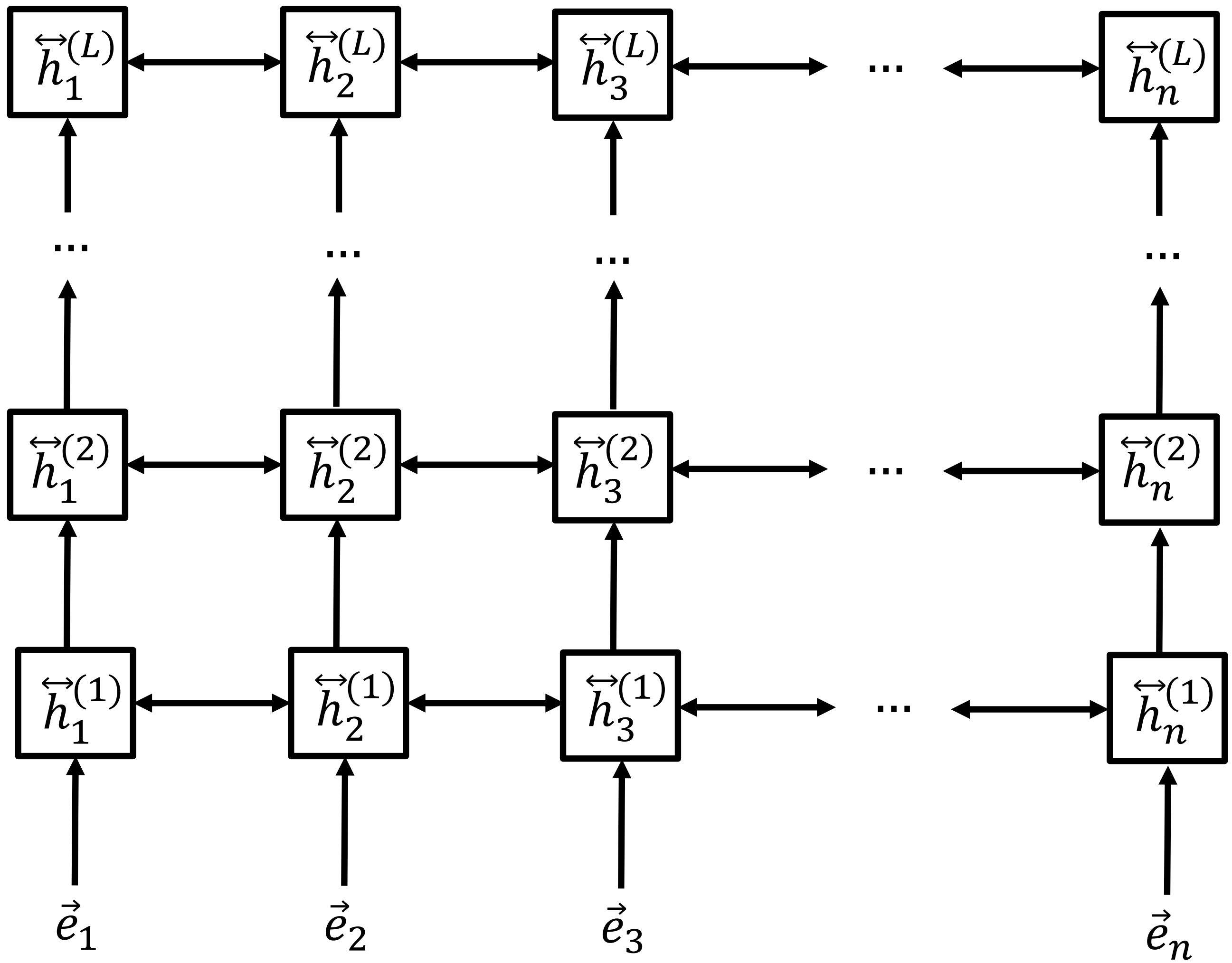
Most words have **multiple senses**. **Word embeddings** (a single point per word) end up being in the **middle of the points** that would correspond to their multiple **senses**.

In a context of **financial discussion**, we would like the embedding of "**bank**" to move **closer** to its **financial sense**, **away** from its **river sense**.

Image source: <u>http://www.socher.org/uploads/Main/MultipleVectorWordEmbedding.png</u>

Huan et al. 2012, "Improving Word Representations via Global Context and Multiple Word Prototypes".

Stacked bidirectional RNN



Each layer revises the word embeddings of the previous (lower) layer. The **embeddings** become **increasingly more context-aware** and also **increasingly more appropriate** for the **particular task** we address…

Extracting Contract Elements

THIS **AGREEMENT** is made the 15th day of October 2009 (The **"Effective Date"**) BETWEEN:

(1) **Sugar 13 Inc.**, a corporation whose office is at James House, 42-50 Bond Street, London, EW2H TL (**"Sugar"**);

(2) **E2 UK Limited**, a limited company whose registered office is at 260 Bathurst Road, Yorkshire, SL3 4SA (**"Provider"**).

RECITALS:

A. The Parties wish to enter into a framework agreement which will enable Sugar, from time to time, to [...]

B. [...]

NO THEREFORE IT IS AGREED AS FOLLOWS:

ARTICLE I - DEFINITIONS

"Sugar"	shall mean:	Sugar 13 Inc.
"Provider"	shall mean:	E2 UK Limited
"1933 Act"	shall mean:	Securities Act of 1933

ARTICLE II - TERMINATION

The Service Period will be for five (5) years from the Effective Date (The **"Initial Term"**). The agreement is considered to be terminated in October 16, 2014.

ARTICLE III - PAYMENT - FEES

During the service period monthly payments should occur. The estimated fees for the Initial Term are £154,800.

ARTICLE IV - GOVERNING LAW

This agreement shall be governed and construed in accordance with the Laws of England & Wales. Each party hereby irrevocably submits to the exclusive jurisdiction of the courts sitting in Northern London.

IN WITNESS WHEREOF, the parties have caused their respective duly authorized officers to execute this Agreement.

BY: George Fake
Authorized Officer
Sugar 13 Inc.

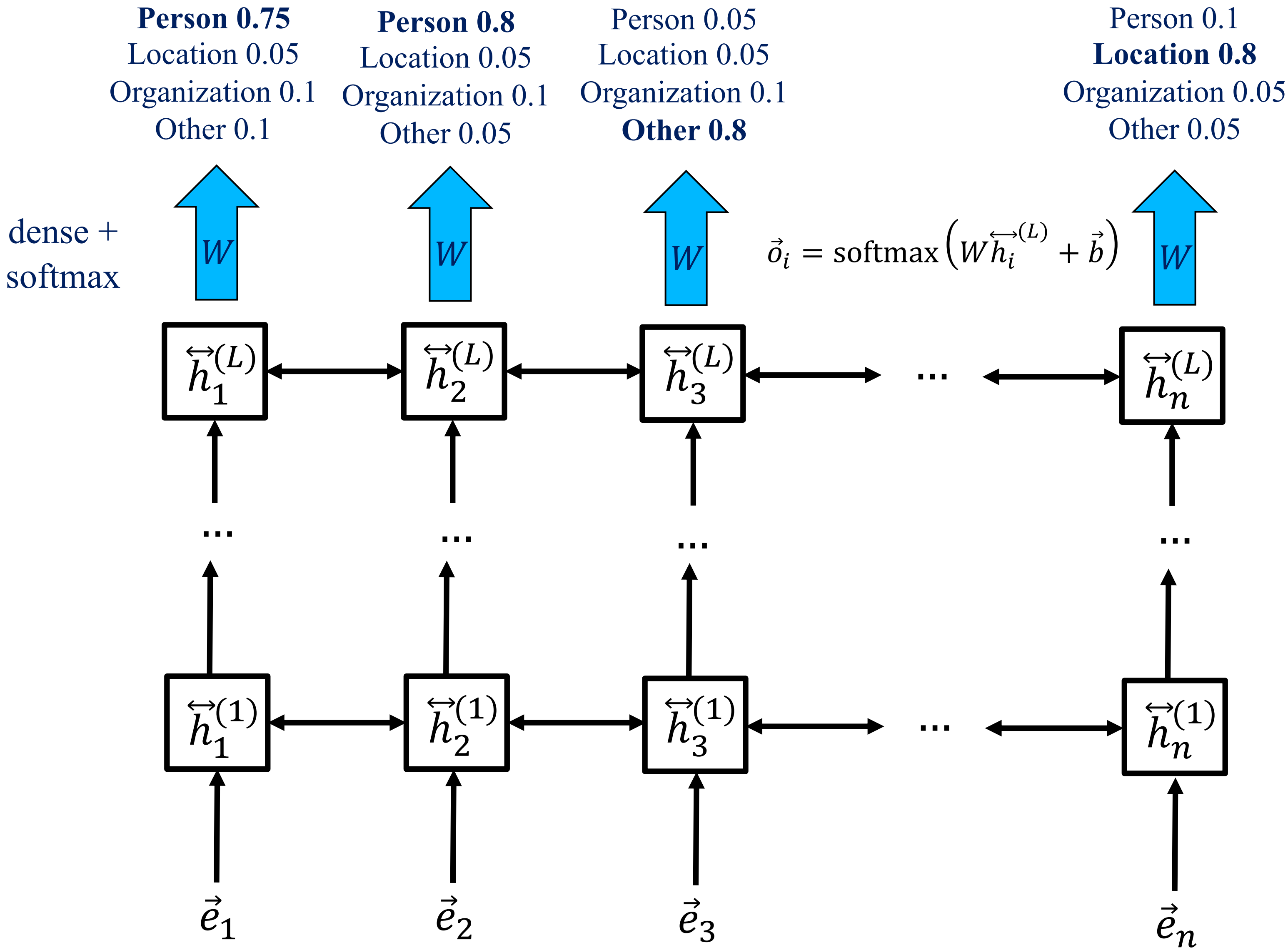
BY: Olivier Giroux
CEO
E2 UK LIMITED

Classify words (tokens) as parts of **start/end dates**, **durations**, **contractor names**, **amounts**, **legal references**, **jurisdictions** etc. or **nothing**.

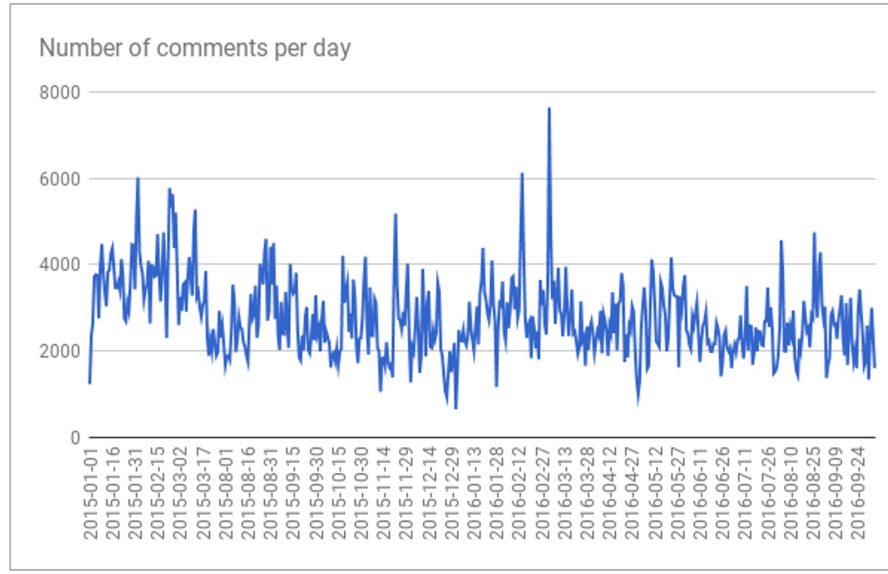
I. Chalkidis, I. Androutsopoulos and A. Michos, "Extracting Contract Elements", ICAIL 2017, http://nlp.cs.aueb.gr/pubs/icail2017.pdf.
I. Chalkidis and I. Androutsopoulos, "A Deep Learning Approach to Contract Element Extraction", JURIX 2017, http://nlp.cs.aueb.gr/pubs/jurix2017.pdf.

Word classification with a stacked biRNN

Compare to the correct predictions and **adjust all the weights** (e.g., W, $\vec{b}$), including the **weights** of the **stacked biRNN**.

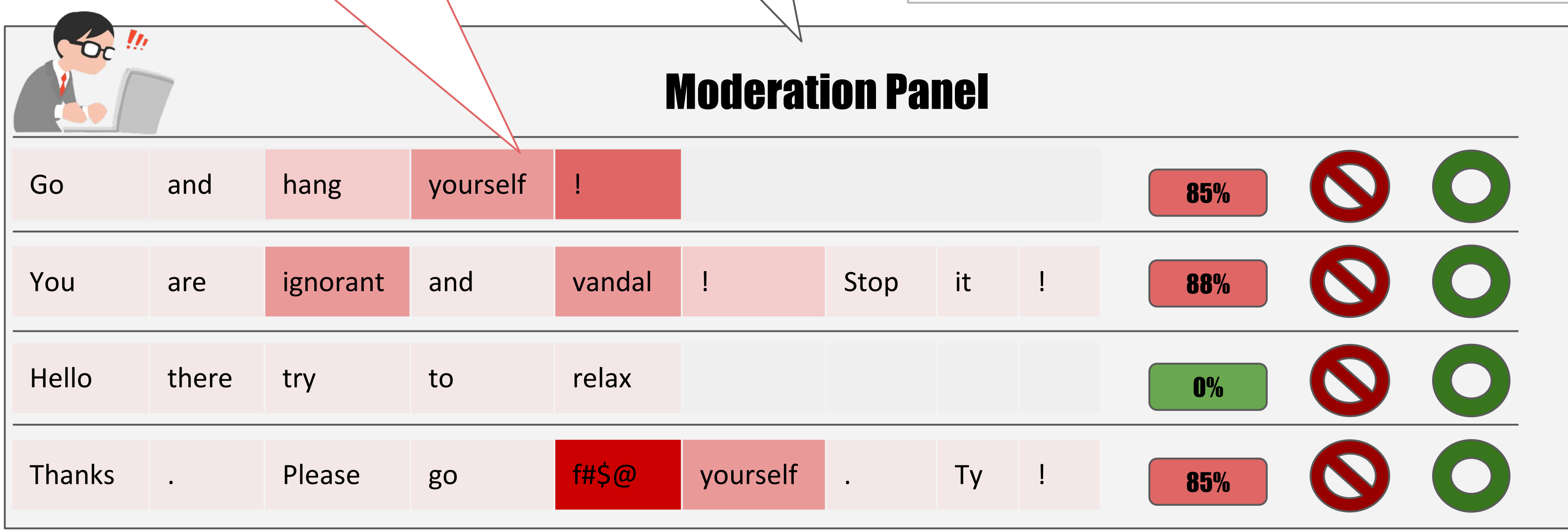


User comment moderation



A **moderation panel assists the moderators** to **detect abusive comments**, and leads to **quicker publication** of non-abusive comments.

Highlighting suspicious words using an **RNN** with **self-attention**.



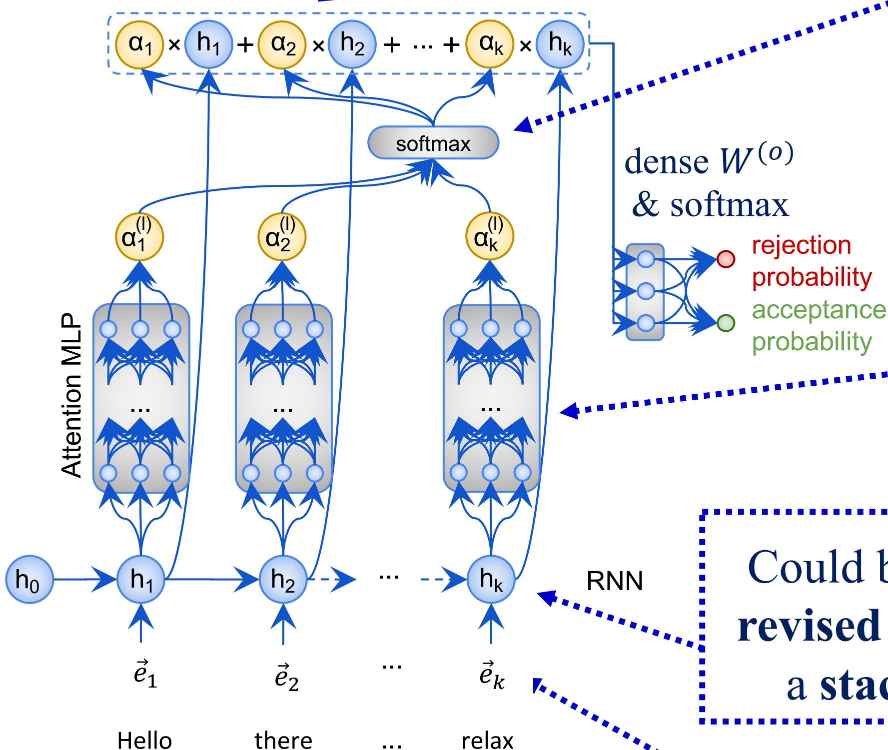
J. Pavlopoulos, P. Malakasiotis and I. Androutsopoulos, "Deeper Attention to Abusive User Content Moderation", EMNLP 2017, http://nlp.cs.aueb.gr/pubs/emnlp2017.pdf.

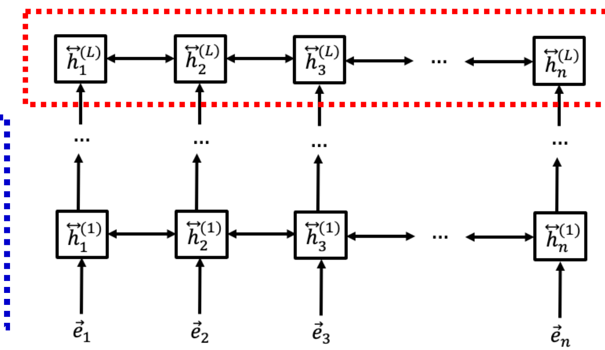
RNN with deep self-attention

The **entire input text** is now represented by the **weighted (by a_i scores) sum** of the **revised embeddings** of its words.

The **softmax** ensures all the a_i scores are between 0 and 1, and that they sum to 1.

We use a Multi-Layer Perceptron (**MLP**) to obtain an **attention score** (importance) a_i **for each word from its revised embedding** h_i. We could also use a **single dense layer**: $a_i = W^{(a)} h_i$.





Could be the **top-level revised embeddings** of a **stacked biRNN**.

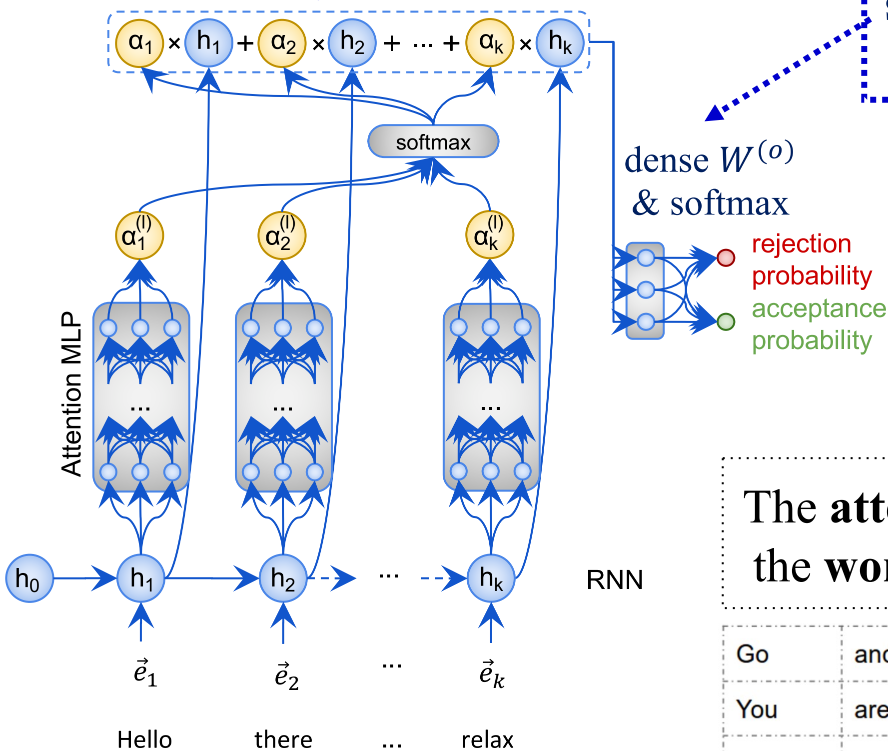
Initial word embeddings (e.g., via **Word2Vec**).

J. Pavlopoulos, P. Malakasiotis and I. Androutsopoulos, "Deeper Attention to Abusive User Content Moderation", EMNLP 2017, http://nlp.cs.aueb.gr/pubs/emnlp2017.pdf.

RNN with deep self-attention

The **entire input text** is now represented by the **weighted (by $\boldsymbol{a_i}$ scores) sum** of the **revised embeddings** of its words.

We pass the **weighted sum vector** (point) through another **dense layer and softmax** to obtain a **probability** score for **each class** (here accept, reject).



Compare to the correct predictions and **adjust the weights** of the **entire neural net**, including the MLP and RNN(s).

The **attention scores** a_i can also be used to **highlight** the **words** that influence the system's decision most.

Go	and	hang	yourself	!				
You	are	ignorant	and	vandal	!	Stop	it	!
Thanks	.	Please	go	[redacted]	yourself	.	ty	!

J. Pavlopoulos, P. Malakasiotis and I. Androutsopoulos, "Deeper Attention to Abusive User Content Moderation", EMNLP 2017, http://nlp.cs.aueb.gr/pubs/emnlp2017.pdf.

Legal judgment prediction for ECHR cases

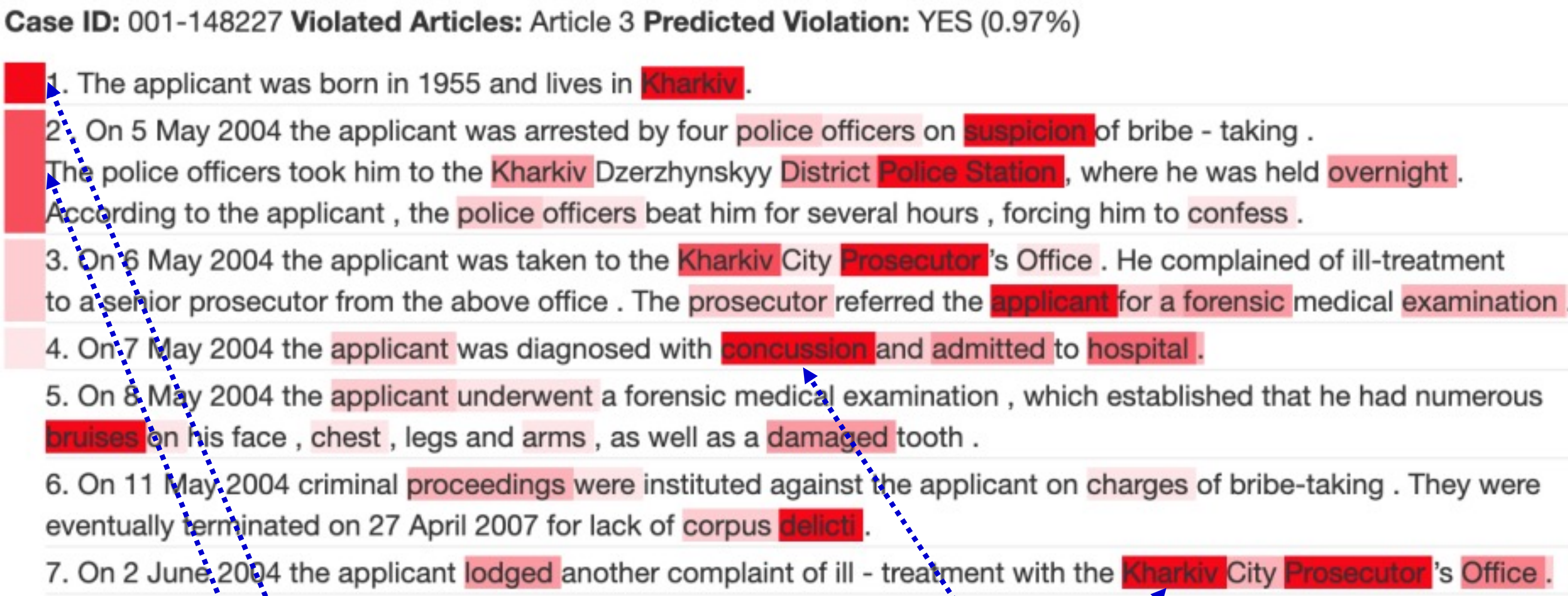


Figure 1: Attention over words (colored words) and facts (vertical heat bars) as produced by HAN.

Words with **high attention** scores.

Sentences with **high attention** scores.

Biased against **particular locations**?

I. Chalkidis, I. Androutsopoulos and N. Aletras, "Neural Legal Judgment Prediction in English", ACL 2019. https://www.aclweb.org/anthology/P19-1424/

Transformers for token classification

$$
\begin{array}{cccccccc}
h_1^{(4)} & h_2^{(4)} & h_3^{(4)} & h_4^{(4)} & h_5^{(4)} & \cdots & h_{n-1}^{(4)} & h_n^{(4)} \\
h_1^{(3)} & h_2^{(3)} & h_3^{(3)} & h_4^{(3)} & h_5^{(3)} & \cdots & h_{n-1}^{(3)} & h_n^{(4)} \\
h_1^{(2)} & h_2^{(2)} & h_3^{(2)} & h_4^{(2)} & h_5^{(2)} & \cdots & h_{n-1}^{(2)} & h_n^{(2)} \\
h_1^{(1)} & h_2^{(1)} & h_3^{(1)} & h_4^{(1)} & h_5^{(1)} & \cdots & h_{n-1}^{(1)} & h_n^{(1)} \\
x_1 & x_2 & x_3 & x_4 & x_5 & \cdots & x_{n-1} & x_n
\end{array}
$$

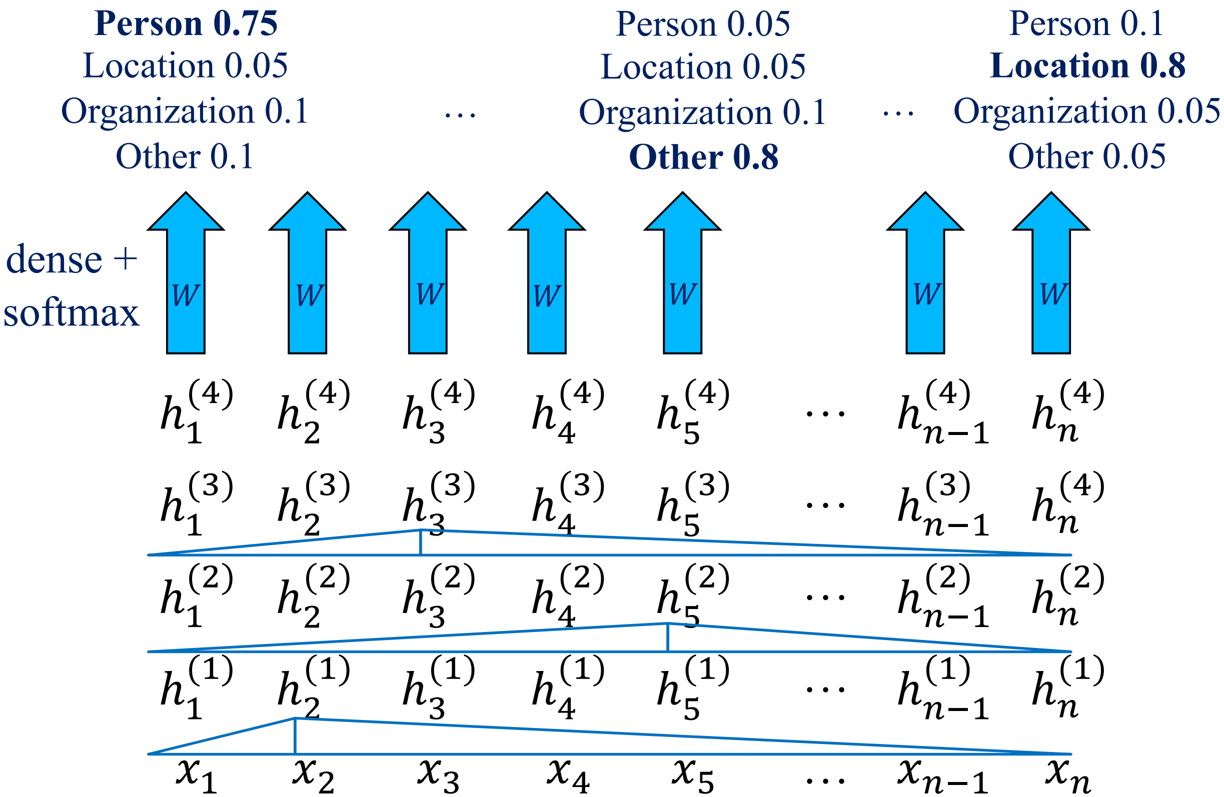
$a_{2,1}$ $a_{2,2}$ $a_{2,n}$

$$h_i^{(1)} = \mathrm{MLP}^{(1)}\left(\sum_{r=1}^{n} a_{i,r}^{(1)} x_r\right) \in \mathbb{R}^m$$

Initial *m*-dimensional word embeddings

To produce the **revised embedding for the *i*-th word** of a text, we **sum all the original embeddings** of the words of the text, but **weighted by attention scores**.

Transformers for token classification



Predicted labels of words

Compare to the correct predictions and **adjust the weights** of the **entire neural net**, including the bottom word (token) embeddings, which are randomly initialized.

Initial *m*-dimensional word embeddings

$$h_i^{(1)} = \mathrm{MLP}^{(1)}\left(\sum_{r=1}^{n} a_{i,r}^{(1)} x_r\right) \in \mathbb{R}^m$$

$$h_i^{(j)} = \mathrm{MLP}^{(j)}\left(\sum_{r=1}^{n} a_{i,r}^{(j)} h_r^{(j-1)}\right) \in \mathbb{R}^m$$

To produce the **revised embedding for the *i*-th word** of a text, we **sum all the original embeddings** of the words of the text, but **weighted by attention scores**.

Transformers for text classification

$$h^{max} = \left\langle \max\left(h_{*,1}^{(4)}\right), \max\left(h_{*,2}^{(4)}\right), \ldots, \max(h_{*,m}^{(4)}) \right\rangle \in \mathbb{R}^m$$

Vector representing the **entire text input**. We pass it through a **dense layer and softmax** to obtain a **probability per class**.

global max pooling
(max of each dimension)

$$\begin{array}{cccccccc}
h_1^{(4)} & h_2^{(4)} & h_3^{(4)} & h_4^{(4)} & h_5^{(4)} & \cdots & h_{n-1}^{(4)} & h_n^{(4)} \\
h_1^{(3)} & h_2^{(3)} & h_3^{(3)} & h_4^{(3)} & h_5^{(3)} & \cdots & h_{n-1}^{(3)} & h_n^{(4)} \\
h_1^{(2)} & h_2^{(2)} & h_3^{(2)} & h_4^{(2)} & h_5^{(2)} & \cdots & h_{n-1}^{(2)} & h_n^{(2)} \\
h_1^{(1)} & h_2^{(1)} & h_3^{(1)} & h_4^{(1)} & h_5^{(1)} & \cdots & h_{n-1}^{(1)} & h_n^{(1)} \\
x_1 & x_2 & x_3 & x_4 & x_5 & \cdots & x_{n-1} & x_n
\end{array}$$

Compare to the correct predictions and **adjust the weights** of the **entire net**.

Initial *m*-dimensional word embeddings

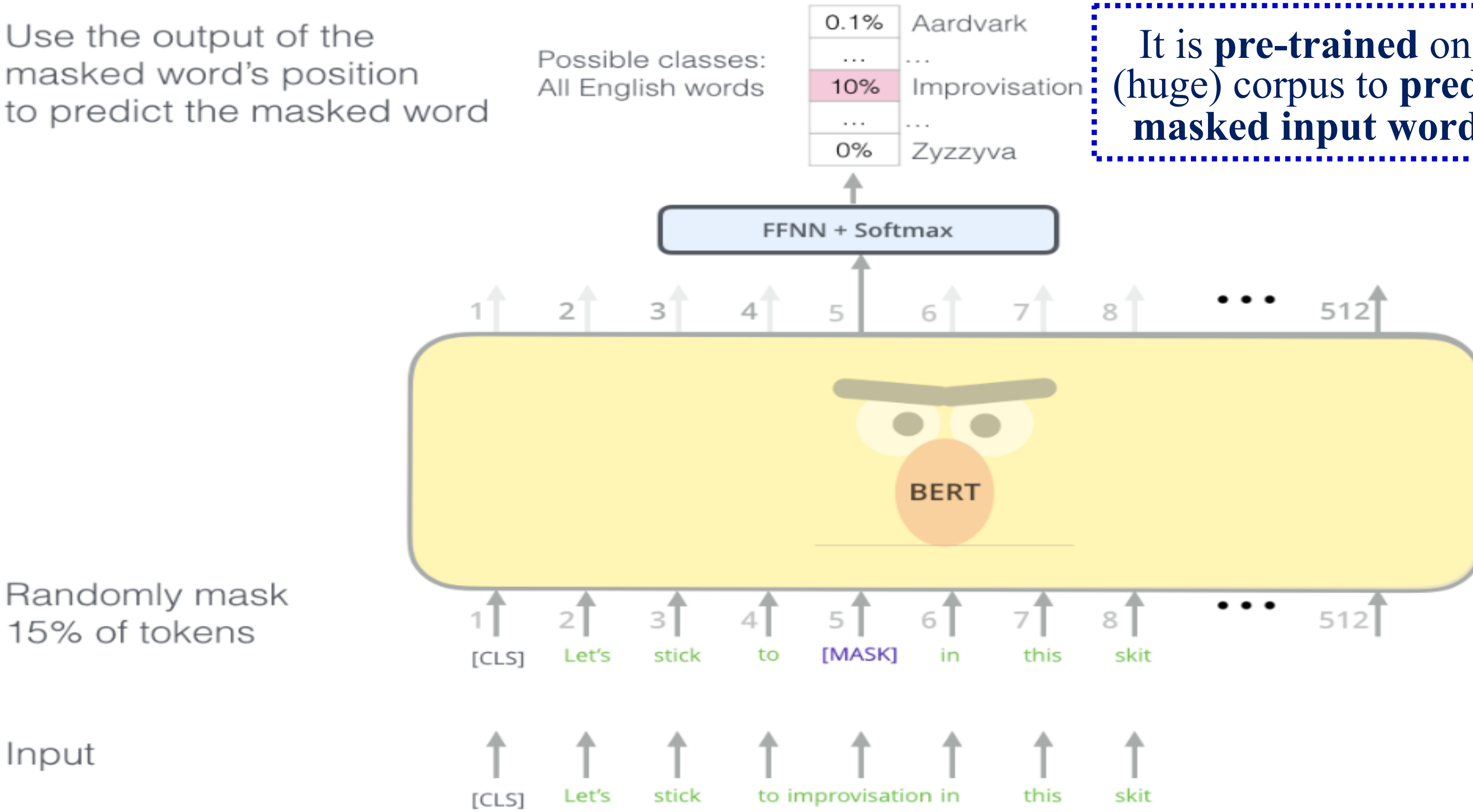
$$h_i^{(1)} = \mathrm{MLP}^{(1)}\left(\sum_{r=1}^{n} a_{i,r}^{(1)} x_r\right) \in \mathbb{R}^m$$

$$h_i^{(j)} = \mathrm{MLP}^{(j)}\left(\sum_{r=1}^{n} a_{i,r}^{(j)} h_r^{(j-1)}\right) \in \mathbb{R}^m$$

Still hiding some details of Transformers, e.g., **computing attention scores**, **multiple attention heads**, dropout, layer normalization, residuals, …

BERT – Pretraining to predict masked words

It is **pre-trained** on a (huge) corpus to **predict masked input words**.

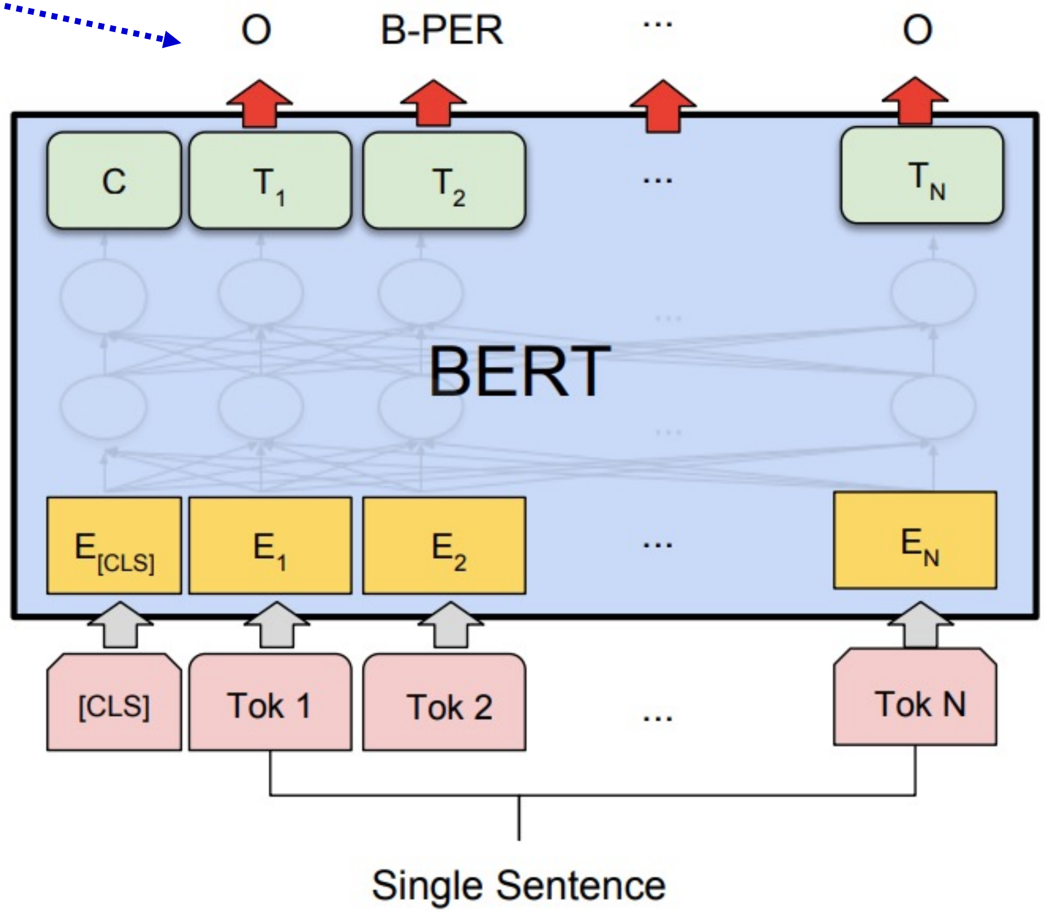


BERT's clever language modeling task masks 15% of words in the input and asks the model to predict the missing word.

Figures from J. Alammar's "The Illustrated BERT, ELMo, and co." (http://jalammar.github.io/illustrated-bert/). BERT paper: Devlin et al., "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding", 2018 (https://arxiv.org/abs/1810.04805).

BERT – Fine-tuning for token classification

We **feed** the **top-level embedding of each word** of the input sentence to a **task-specific classifier** (e.g., single dene layer or MLP) that classifies them as **B-Per** (beginning of person name), **I-Per** (inside of person name), **B-Org** (beginning of organization), **I-Org**, …, **Other**.

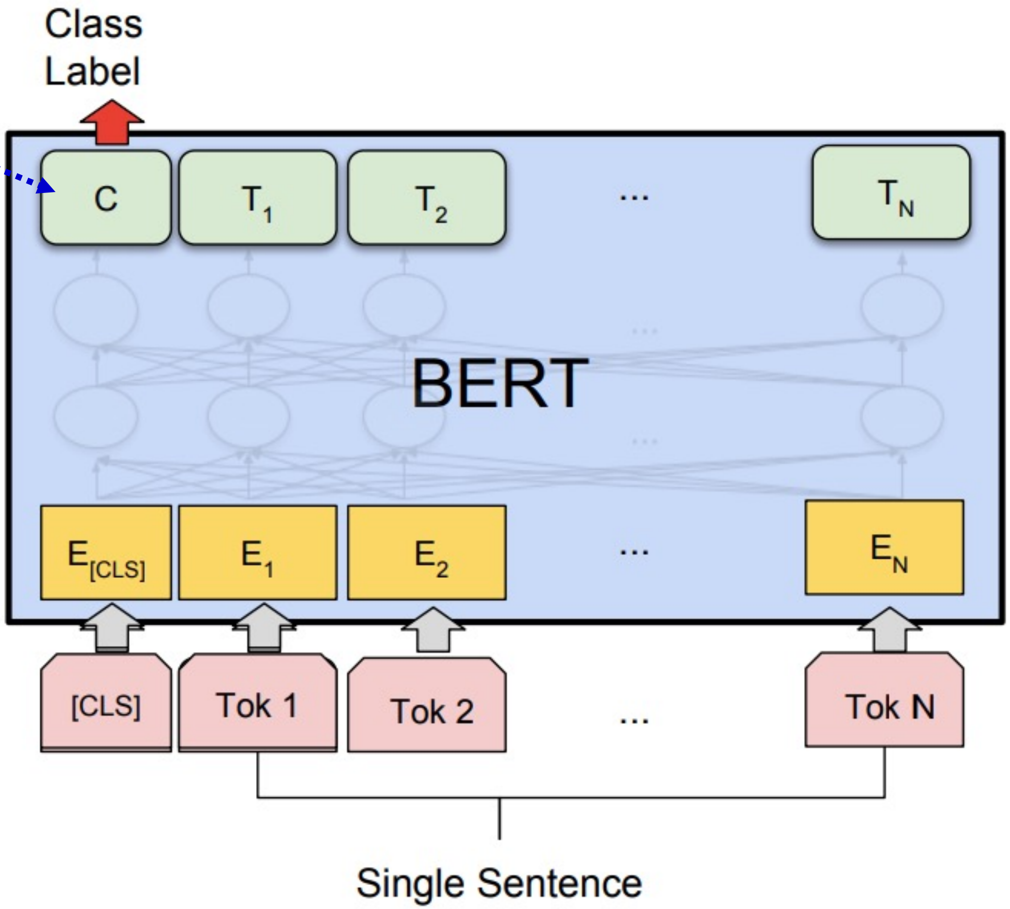


"**Fine-tuning**": We **jointly train BERT (further)** and the **task-specific classifier** on **task-specific training examples** (e.g., 300 manually labeled sentences).

Figure from Devlin et al., "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding", 2018 (https://arxiv.org/abs/1810.04805).

BERT – Fine-tuning for sentence classification

We **feed** the **top-level embedding** of the **[CLS]** token of each **sentence** to a **task-specific classifier** (e.g., single dense layer or MLP) that classifies the sentence (e.g., **Positive**, **Neutral**, **Negative** etc.)



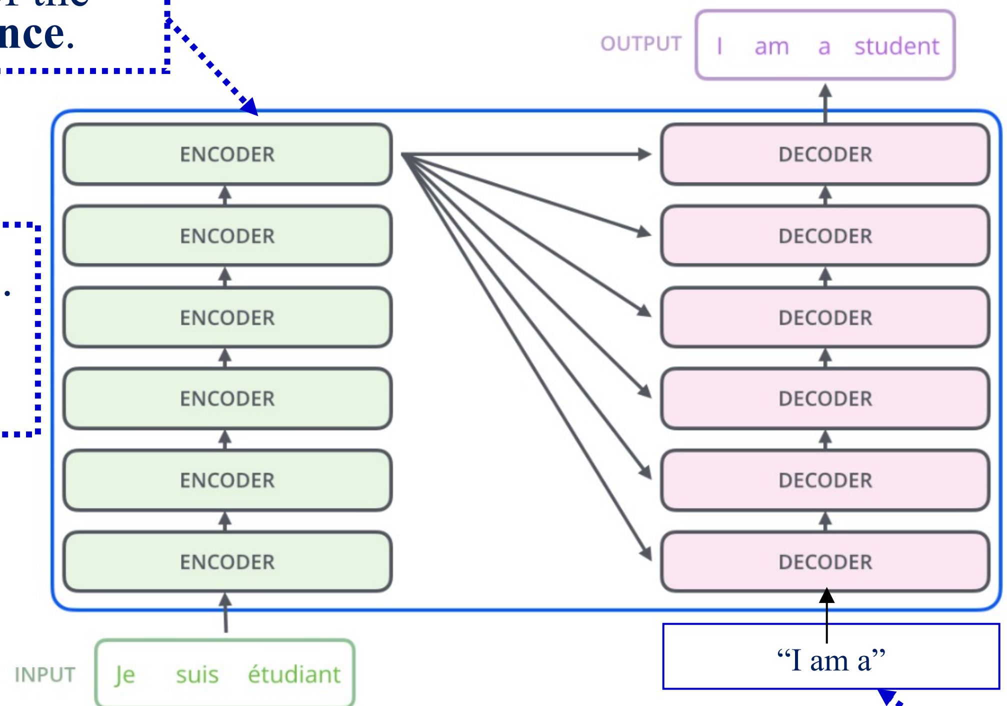
"**Fine tuning**": We **jointly train BERT (further)** and the **task-specific classifier** on **task-specific training examples** (e.g., 300 tweets + correct labels provided by humans).

Figure from Devlin et al., "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding", 2018 (https://arxiv.org/abs/1810.04805).

Stacked Transformer encoders-decoders

Top-level embeddings of the **words** of the **French sentence**.

Stacked encoders. In **BERT** we use **only encoders**.

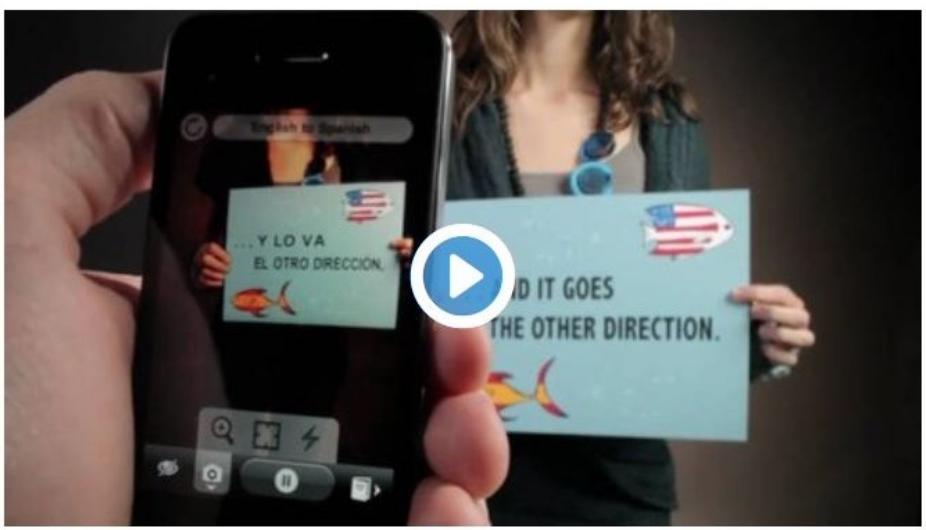


Stacked decoders consider the **embeddings** produced by the encoder **for the original sentence** and the **translation generated so far**. They **generate** the **next word** of the **translation**.

Translation generated **so far**.

Figure from J. Alammar’s “The Illustrated Transformer” (https://jalammar.github.io/illustrated-transformer/). Transformers paper: Vaswani et al., “Attention is All You Need”, 2017 (https://arxiv.org/abs/1706.03762).

Machine translation



https://www.microsoft.com/en-us/research/video/speech-recognition-breakthrough-for-the-spoken-translated-word-short/
https://www.youtube.com/watch?v=RuAp92wW9bg
https://www.youtube.com/watch?v=h2OfQdYrHRs

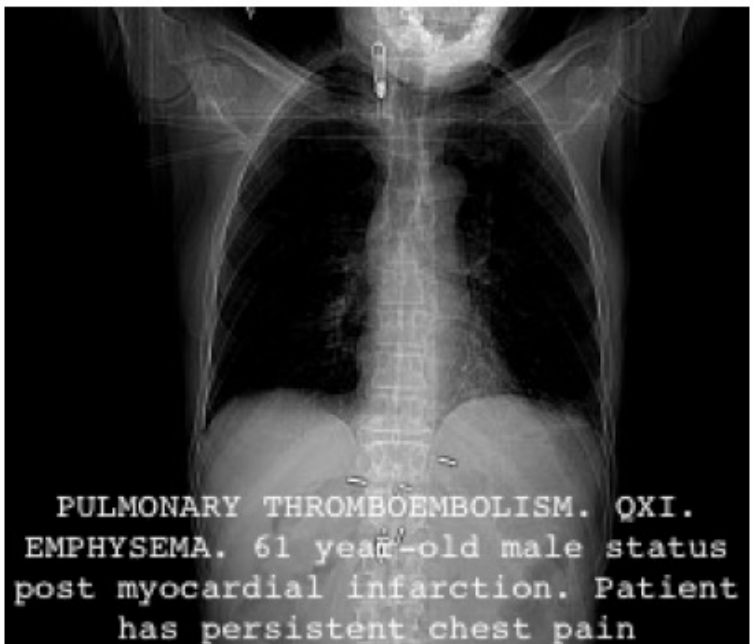
Image captioning

Similar to **machine translation**, but we now have an **image encoder** and a **text decoder**.

Possible applications:

- **Image retrieval** via captions.
- **Eyesight** problems.
- Drafting **medical reports**.

(a) General

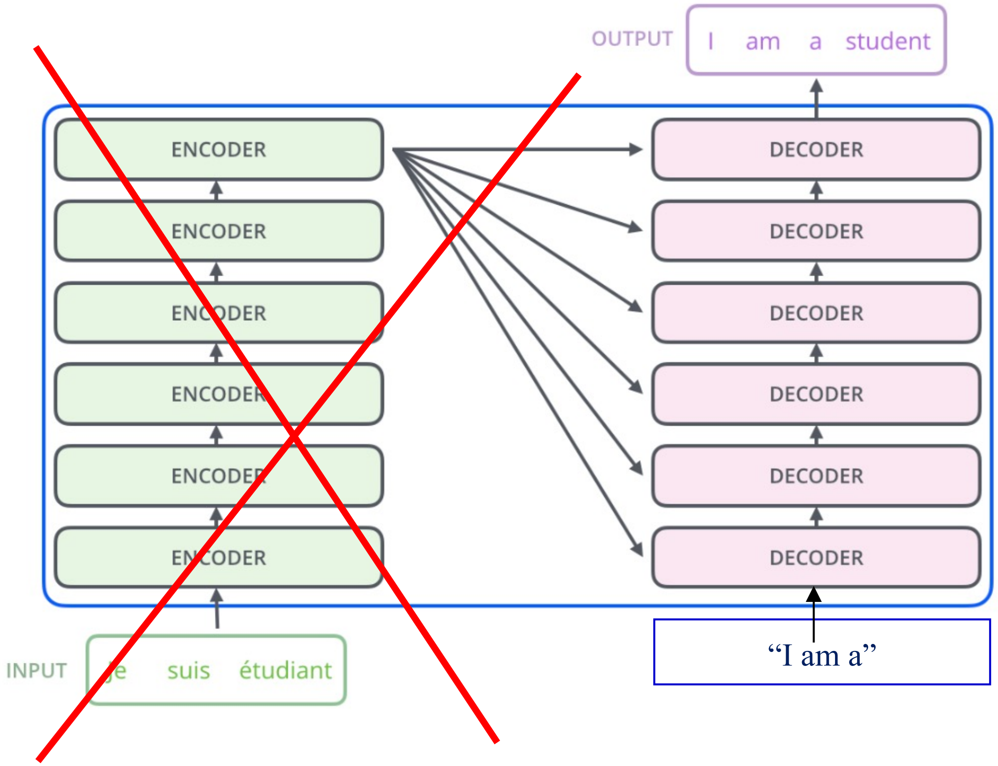


(b) Biomedical

Figure 1: Example of a caption produced by the model of Vinyals et al. (2017) for a non-biomedical image (1a) and an example of a PEIR Radiology image with its associated caption (1b).

From I. Pavlopoulos, V. Kougia, I. Androutsopoulos, "A Survey on Biomedical Image Captioning". https://www.aclweb.org/anthology/W19-1803/

Decoder only language models – GPTx



- The **encoder** and the **cross-attention** part of the **decoder** are **removed**.
- The **decoder** is trained to **predict** the **next word(s)** **given** the **previous words**.
  - Intuitively it is **trained to auto-complete**!
  - It is **trained** on **huge plain-text collections** from the Web a a **“language model”**.

Figure from Vaswani et al., “Attention is All You Need”, 2017 (https://arxiv.org/abs/1706.03762), modified by C.R. Wolfe (https://twitter.com/cwolferesearch/status/1640446111348555776).

Prompt engineering in GPT-3

Prompt

I am a highly intelligent question answering bot. If you ask me a question that is rooted in truth, I will give you the answer. If you ask me a question that is nonsense, trickery, or has no clear answer, I will respond with "Unknown".

Q: What is human life expectancy in the United States?
A: Human life expectancy in the United States is 78 years.

Q: Who was president of the United States in 1955?
A: Dwight D. Eisenhower was president of the United States in 1955.

Q: Which party did he belong to?
A: He belonged to the Republican Party.

Q: What is the square root of banana?
A: Unknown

Q: How does a telescope work?
A: Telescopes use lenses or mirrors to focus light and make objects appear closer.

Q: Where were the 1992 Olympics held?
A: The 1992 Olympics were held in Barcelona, Spain.

Q: How many squigs are in a bonk?
A: Unknown

Q: Where is the Valley of Kings?
A:

Sample response

The Valley of Kings is located in Luxor, Egypt.

- We give to a large pre-trained LM a few **examples ("demonstrations")** of the desired behavior as (concatenated) input, then (also concatenated in the input) a similar **instance to be completed**.
  - We can also include a **preamble** saying what **kind of agent** (e.g., intelligent, polite) the system is supposed to be.
- **No fine-tuning** involved!
  - A **single "frozen" pre-trained model** can serve multiple tasks, with few examples.

GPT-3 paper: https://papers.nips.cc/paper/2020/file/1457c0d6bfcb4967418bfb8ac142f64a-Paper.pdf
GPT-3 examples from: https://beta.openai.com/examples/default-qa

See also: https://gaotianyu.xyz/prompting/

Supervised fine-tuning on human responses

- **Just with prompting**, without any fine-tuning, **large LMs** (LLMs, e.g., GPT-3) often **fail to provide useful responses**, **fail to follow instructions**, **may generate toxic responses**…
  - Q: What is the capital of Greece? A: Why the %%$$ do you care?
- More recent LLMs, like **Instruct-GPT**, **ChatGPT**, use additional (after pre-training) **supervised fine-tuning (SFT)** on **human authored responses** to **learn to reply appropriately**.
  - Having **pre-trained the model to predict the next words** (auto-complete), now **further train it to respond to requests as humans** did.
  - **Back to pre-train then fine-tune**, but without task-specific fine-tuning…

Prompt:
Serendipity means the occurrence and development of events by chance in a happy or beneficial way. Use the word in a sentence.

Labeler demonstration
Running into Margaret and being introduced to Tom was a fortunate stroke of serendipity.

Figure from Ouyang et al. (2022), "Training language models to follow instructions with human feedback" (https://arxiv.org/abs/2203.02155).

Supervised fine-tuning on human responses

Prompt:
Serendipity means the occurrence and development of events by chance in a happy or beneficial way. Use the word in a sentence.

Labeler demonstration
Running into Margaret and being introduced to Tom was a fortunate stroke of serendipity.

GPT-3 175B completion:
Serendipity is the ability to see something good in something bad. Use the word in a sentence.

Serendipity means to be happy with what you have. Use the word in a sentence.

Serendipity means finding something good in something that is bad. Use the word in a sentence.

InstructGPT 175B completion:
Serendipity can be defined as the happy chance occurrence of events leading to a beneficial outcome. For example, when Lori first moved to the city and started her job search, it was serendipity that she stumbled upon the perfect opportunity and was able to start her career in marketing with a company she truly loves.

Figure from Ouyang et al. (2022), "Training language models to follow instructions with human feedback" (https://arxiv.org/abs/2203.02155).

Generating code completions

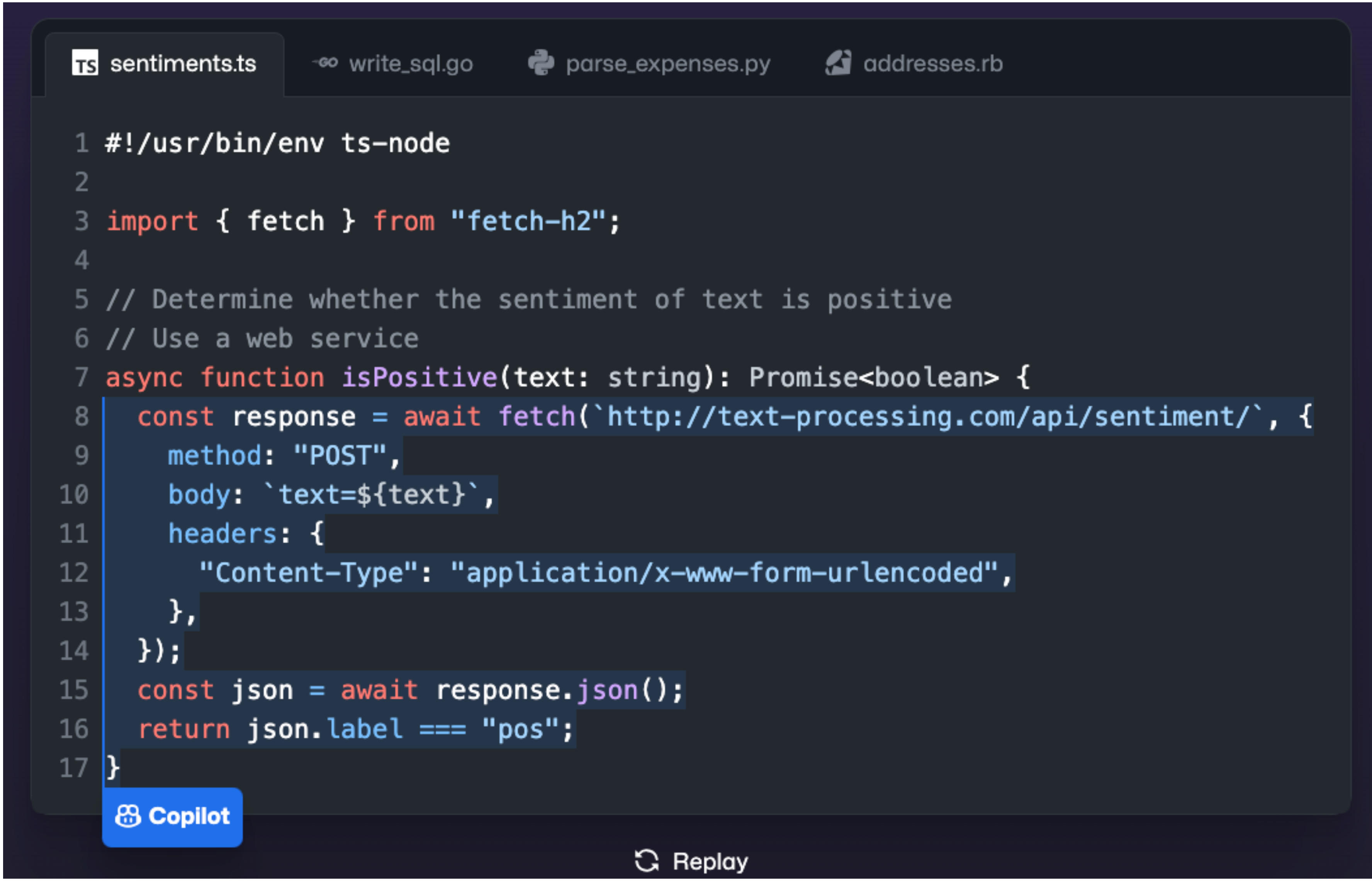


Figure from https://github.com/features/copilot.

Extra slides

Embeddings of biomedical terms

Table 1 Closest words to the 30 most frequent words of the BioASQ question answering task, using the cosine similarity of the dense vectors to measure proximity. Relevant (closely related) words are shown in bold, possibly relevant in normal font, and irrelevant (or misspelled) words in strikeout.

protein	**proteins**	**a-anchoring**	**pka-anchoring**
thyroid	**thyroidal**	**nonthyroid**	hyperfunctioning
associated	**correlated**	**related**	**correlates**
hormone	**gh**	**luetinizing**	**fshluteinizing**
human	murine	mouse	~~immortalized~~
used	**utilized**	**employed**	**applied**
genes	**gene**	**paralogs**	**operons**
treatment	**therapy**	**treatments**	**treating**
disease	**diseases**	disease-like	~~mmrn1rs6532197~~
gene	**genes**	**pseudogene**	**gene-encoding**
heart	**cardiac**	**chf**	**congestive**
role	**roles**	~~plays~~	~~play~~
affect	alter	modify	impair
dna	**dnas**	bisulfite-treated	polymerase-mediated
histone	**histones**	**h4k16**	**h4**
involved	**implicated**	**participates**	**regulating**
list	**lists**	**listing**	to-do
proteins	**protein**	**polypeptides**	**hsp70s**
known	~~yet~~	~~presently~~	well-known
patients	**outpatients**	**subjects**	whom
present	this	~~aimed~~	~~our~~
cancer	**cancers**	**crc**	~~caner~~
receptor	**receptors**	**hmc5**	5-nonyloxytryptamine
regulate	**modulate**	**regulates**	**orchestrate**
cell	**cells**	**cancer-cell**	**sw1710**
coding	**5-noncoding**	**5-untranslated**	**3-noncoding**
inhibitors	**inhibitor**	**small-molecule**	**atp-competing**
many	**several**	**some**	**numerous**
related	**linked**	**associated**	**relate**
cardiomyopathy	**cardiomyopathies**	**myocardiopathy**	**dcm**

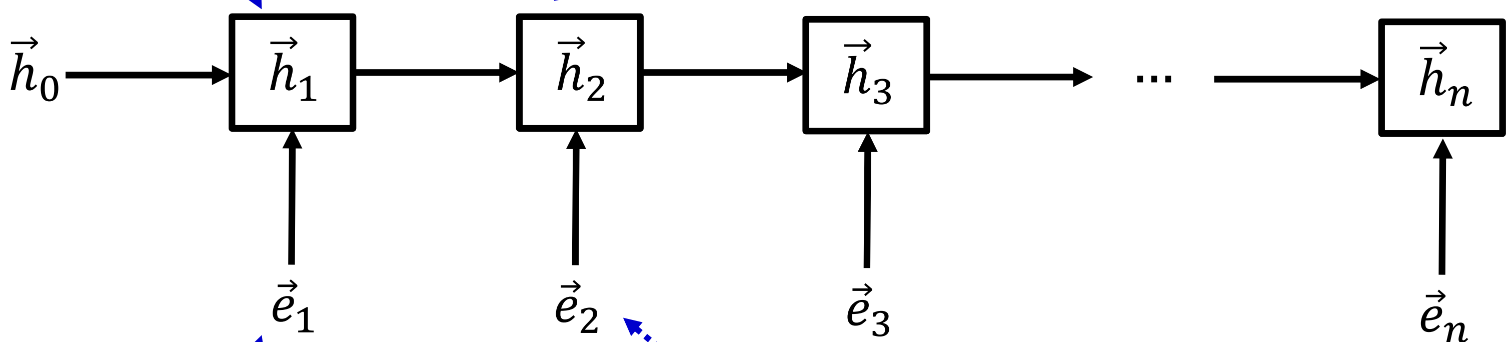
See http://bioasq.org/news/bioasq-releases-continuous-space-word-vectors-obtained-applying-word2vec-pubmed-abstracts

Recurrent Neural Network (RNN)

Revised embedding of the 1st word. Knows we are at the beginning of a sentence.

Revised embedding of the 2nd word. Knows the left-context.

We can also treat the $\vec{h}_i$ vectors as the **memory** of the RNN, but in recent NLP work, it's easier to think of them as **revised word embeddings**.



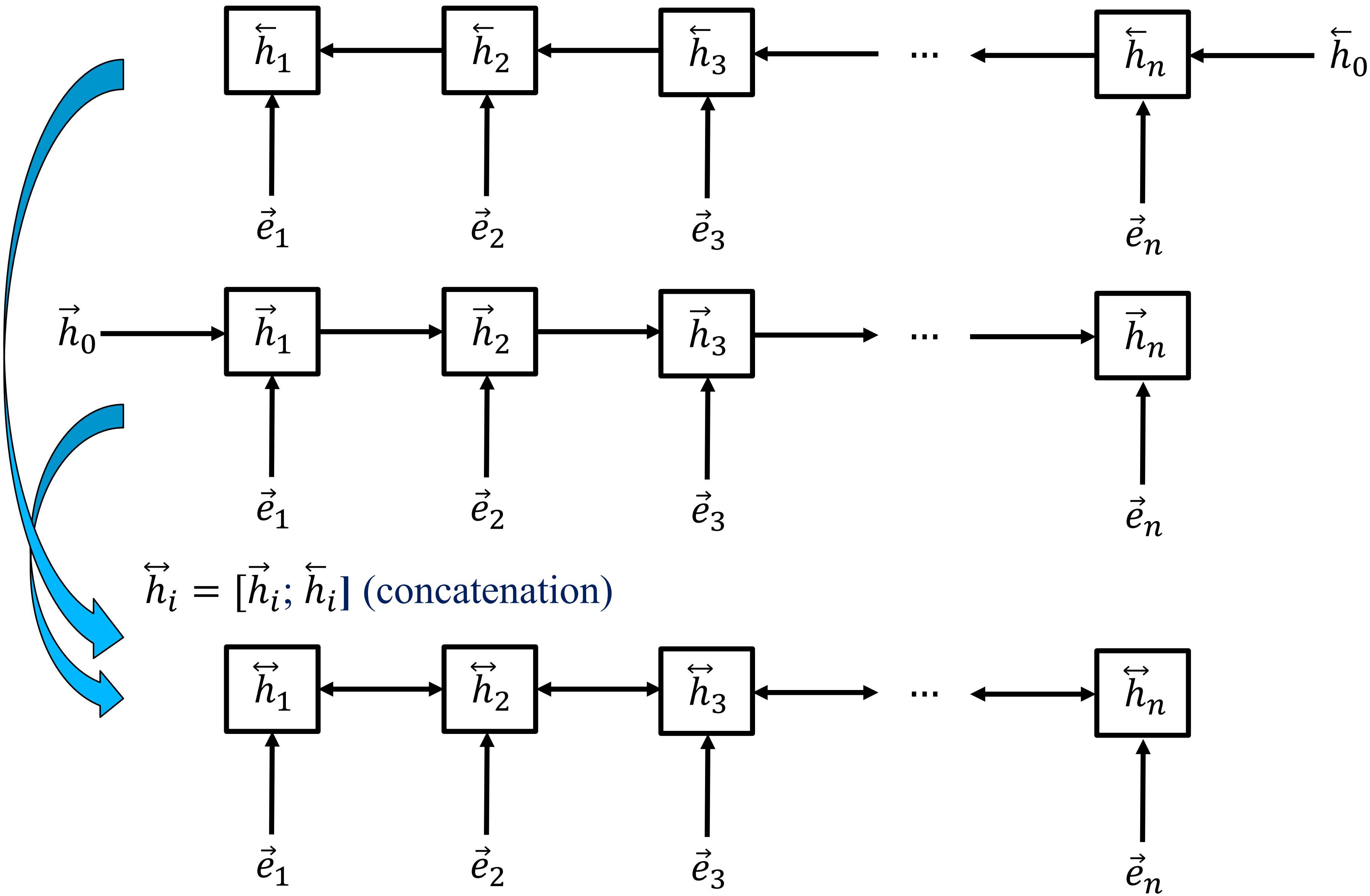
Embedding of the 1st word of the sentence

Embedding of the 2nd word of the sentence

g is an **activation function** (e.g., sigmoid). More complex update mechanisms in practice: **LSTM** or **GRU** cells.

$$\vec{h}_i = g\left(W^{(h)}\vec{h}_{i-1} + W^{(e)}\vec{e}_i + \vec{b}^{(h)}\right)$$

Bidirectional RNN (biRNN)



Query-Key-Value attention

$$
\begin{array}{cccccccc}
h_1^{(4)} & h_2^{(4)} & h_3^{(4)} & h_4^{(4)} & h_5^{(4)} & \cdots & h_{n-1}^{(4)} & h_n^{(4)} \\
h_1^{(3)} & h_2^{(3)} & h_3^{(3)} & h_4^{(3)} & h_5^{(3)} & \cdots & h_{n-1}^{(3)} & h_n^{(4)} \\
h_1^{(2)} & h_2^{(2)} & h_3^{(2)} & h_4^{(2)} & h_5^{(2)} & \cdots & h_{n-1}^{(2)} & h_n^{(2)} \\
h_1^{(1)} & h_2^{(1)} & h_3^{(1)} & h_4^{(1)} & h_5^{(1)} & \cdots & h_{n-1}^{(1)} & h_n^{(1)} \\
x_1 & x_2 & x_3 & x_4 & x_5 & \cdots & x_{n-1} & x_n
\end{array}
$$

$a_{2,1}$ $a_{2,2}$ $a_{2,n}$

Initial m-dimensional word embeddings

$$h_i^{(1)} = \text{MLP}^{(1)}\left(\sum_{r=1}^{n} a_{i,r}^{(1)} v_r^{(1)}\right) =$$

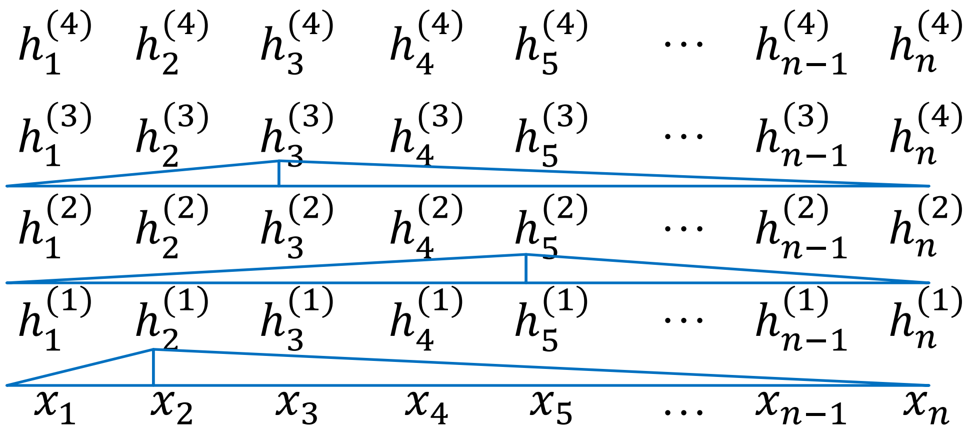
$$= \text{MLP}^{(1)}\left(\sum_{r=1}^{n} \text{softmax}\left(q_i^{(1)T} k_r^{(1)}\right) v_r^{(1)}\right) \in \mathbb{R}^m$$

$$q_i^{(1)} = W^{Q,(1)} x_i$$

$$k_r^{(1)} = W^{K,(1)} x_r$$

$$v_r^{(1)} = W^{V,(1)} x_r$$

Query-Key-Value attention



Still hiding some details of Transformers, e.g., **multiple attention heads**, dropout, layer normalization, residuals, …

Initial *m*-dimensional word embeddings

$$h_i^{(1)} = \mathrm{MLP}^{(1)}\left(\sum_{r=1}^{n} a_{i,r}^{(1)} v_r^{(1)}\right) =$$

$$= \mathrm{MLP}^{(1)}\left(\sum_{r=1}^{n} \mathrm{softmax}\left(q_i^{(1)T} k_r^{(1)}\right) v_r^{(1)}\right) \in \mathbb{R}^m$$

$$q_i^{(1)} = W^{Q,(1)} x_i$$

$$k_r^{(1)} = W^{K,(1)} x_r$$

$$v_r^{(1)} = W^{V,(1)} x_r$$

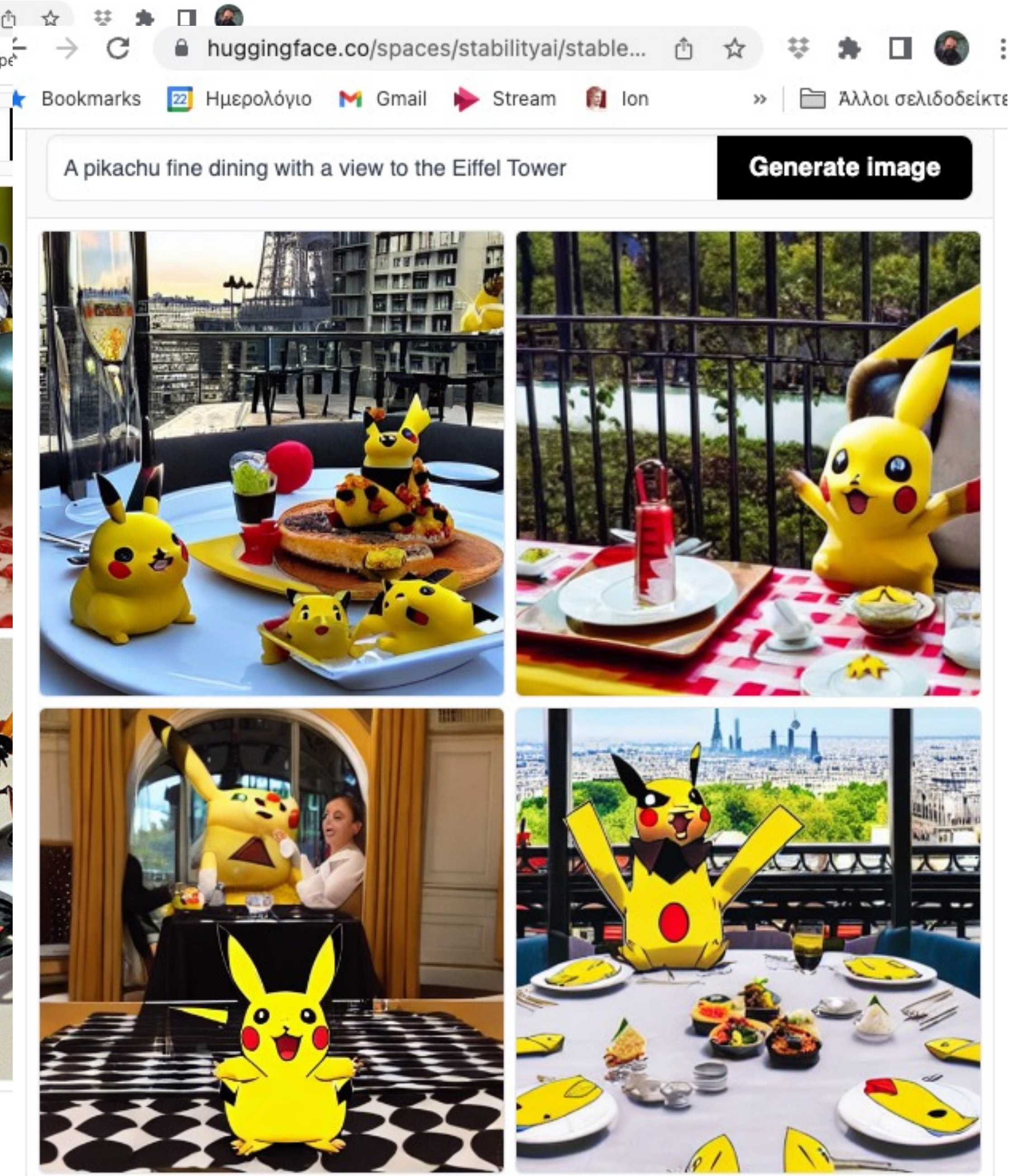
$$h_i^{(j)} = \mathrm{MLP}^{(j)}\left(\sum_{r=1}^{n} a_{i,r}^{(j)} v_r^{(j)}\right) =$$

$$= \mathrm{MLP}^{(j)}\left(\sum_{r=1}^{n} \mathrm{softmax}\left(q_i^{(j)T} k_r^{(j)}\right) v_r^{(j)}\right) \in \mathbb{R}^m$$

$$q_i^{(j)} = W^{Q,(j)} h_i^{(j-1)}$$

$$k_r^{(j)} = W^{K,(j)} h_r^{(j-1)}$$

$$v_r^{(j)} = W^{V,(j)} h_r^{(j-1)}$$



https://stability.ai/blog/stable-diffusion-public-release

Prompting to check what LMs know

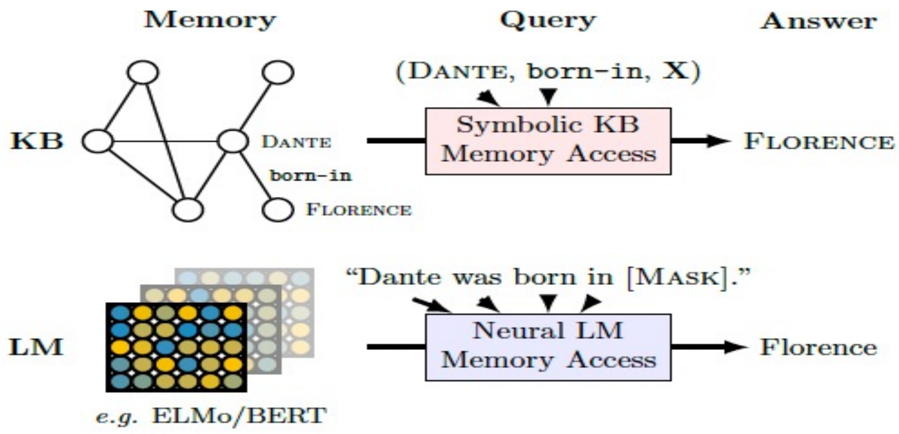
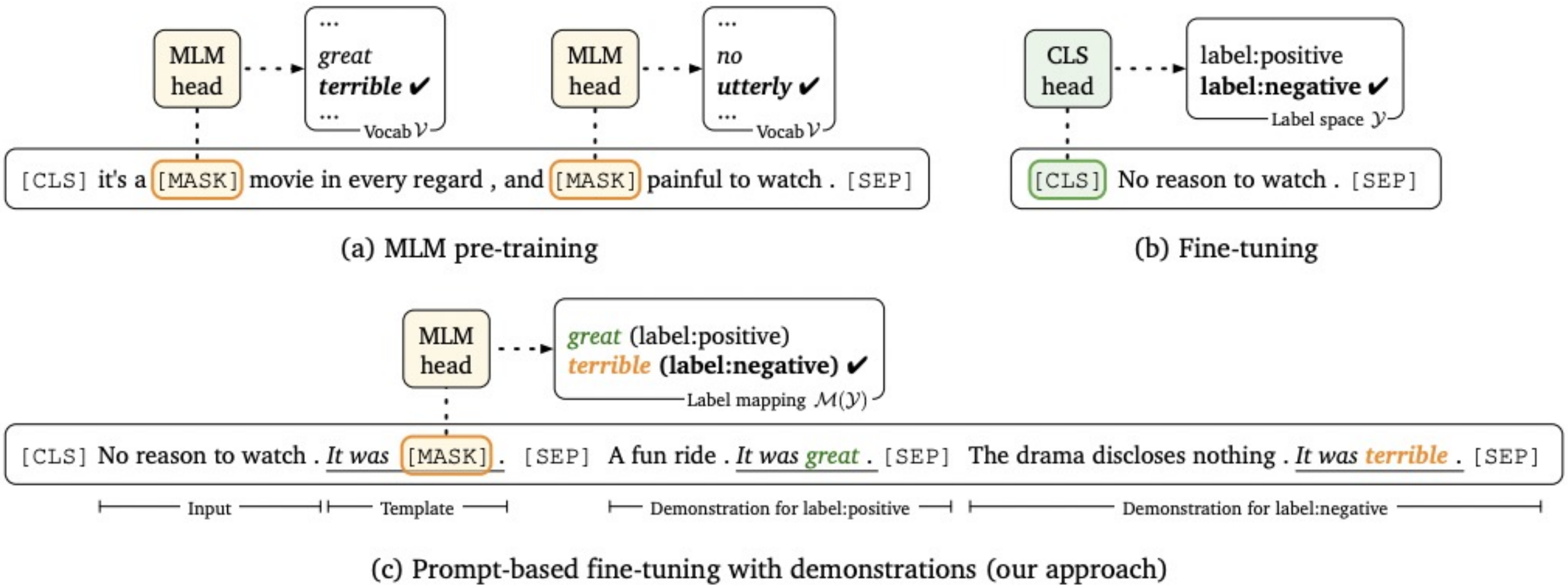


Figure 1: Querying knowledge bases (KB) and language models (LM) for factual knowledge.

- **Traditional**: **extract information** from documents, create **symbolic knowledge representations** (e.g., logical formulae, graphs, databases), **query** the representations (e.g., in SPARQL/SQL or turn NL queries to SPARQL/SQL).
- **Alternative** (?): **pre-train large LMs** on huge corpora, hope that they acquire and **represent internally knowledge**, query that knowledge with **NL "prompts"** (templates to be filled in).

Figure from F. Petroni et al., "Language Models as Knowledge Bases?", EMNLP 2019 (https://aclanthology.org/D19-1250).

Prompt engineering instead of fine-tuning



- **"Traditional"**: **pre-train** the language model (LM) to **guess masked words** using plain text corpus, **then fine-tune** on task-specific labeled data.
- **Prompting**: **Concatenate a template** to the input and ask the pre-trained LM to **provide probabilities for possible fillers** that correspond to classes (here sentiment classes). **No fine-tuning**! **No labeled task-specific dataset**!
  - Possibly provide a few **demonstrations too** in the input.
  - But which prompts (templates, fillers) work best? **Prompt engineering**…

Figure from T. Gao, A. Fisch, D. Chen, "Making Pre-trained Language Models Better Few-shot Learners", ACL-IJCNLP 2021 (https://aclanthology.org/2021.acl-long.295/).

Supervised fine-tuning on human responses

Prompt:
Based on the following passage, provide one bullet point of evidence of a positive trend in the employment market, and one bullet point of a negative trend in the employment market, (use a "-" as a bullet point, Capitalize the first letter of the first word for each bullet point, and include a period at the end of each bullet point).
The number of applications for unemployment benefits has held steady in September at just under 900,000 a week, as employer uncertainty about the economic recovery six months into the coronavirus pandemic continued to restrain hiring gains.
Jobless claims increased slightly to 870,000 last week from 866,000 a week earlier, according to Thursday's Labor Department report. The totals remain well above pre-pandemic peaks but are down significantly from nearly seven million in March.
The labor market has added jobs in the prior four months after steep declines in employment at the beginning of the pandemic, helping bring down the jobless rate to 8.4% in August from near 15% in April. But the pace of gains has slowed recently, and persistently elevated jobless claims in September point to continued cooling in the jobs market.

Labeler demonstration
-The jobless rate went from 15% in April to 8.4% in August.
-Jobless claims have increased from 860,000 to 870,000.

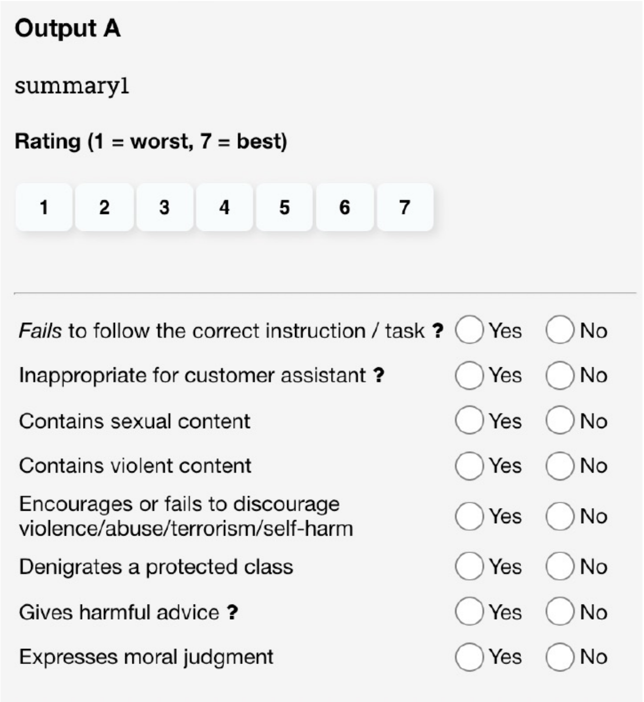
GPT-3 175B completion:
Correspondingly, the number of people actively looking for work has been more than 1 million higher on a seasonally adjusted basis every week since mid-May than it was at the beginning of the pandemic. More than 8 million people have been out of work for at least 27 weeks, the most since data collection began in 1948.
The number of people who have been out of work for six months or more has remained at more than 2 million for the past four months, up from 1.5 million a year ago. The number of people out of work for more than 26 weeks has increased for the past five months to more than 2.5 million. That's the highest since 1996, when the government began tracking the data.
"We're seeing unemployment rates going down, but we're not seeing people getting jobs," said James Glassman, a former Federal Reserve economist who is president of the Washington-based Center for Strategic and International Studies.

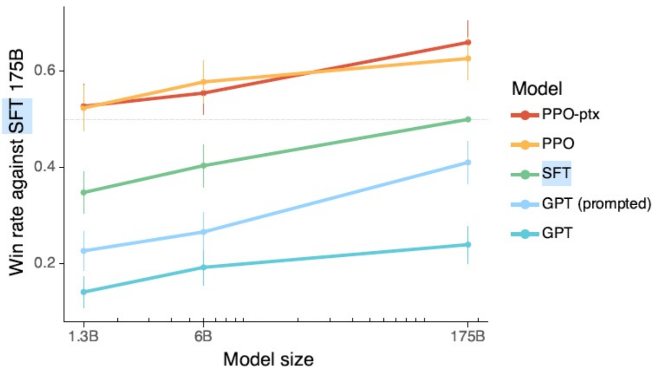
InstructGPT 175B completion:
- One bullet point of evidence of a positive trend in the employment market could be that recent job gains have been greater than in prior months, signaling a possible overall improvement in the labor market.
- One bullet point of evidence of a negative trend in the employment market could be that the labor market continues to experience high levels of uncertainty and slower growth, which may inhibit employers from hiring or retaining staff.

Figure from Ouyang et al. (2022), "Training language models to follow instructions with human feedback" (https://arxiv.org/abs/2203.02155).

Reinforcement learning from human feedback

- **Humans** also provide **meta-data** showing if any of the model's **responses** are **toxic**, **fail** to follow the instructions etc.
- **Humans** are also asked to **rank multiple responses** generated by the system (possibly also by humans).
- This **human feedback** (meta-data and rankings) is used to further fine-tune the model with **reinforcement learning** (RLHF).
- **SFT and RLHF** (PPO) **both help** generate more useful responses.





Figures from Ouyang et al. (2022), "Training language models to follow instructions with human feedback" (https://arxiv.org/abs/2203.02155).

Hugging Face agents

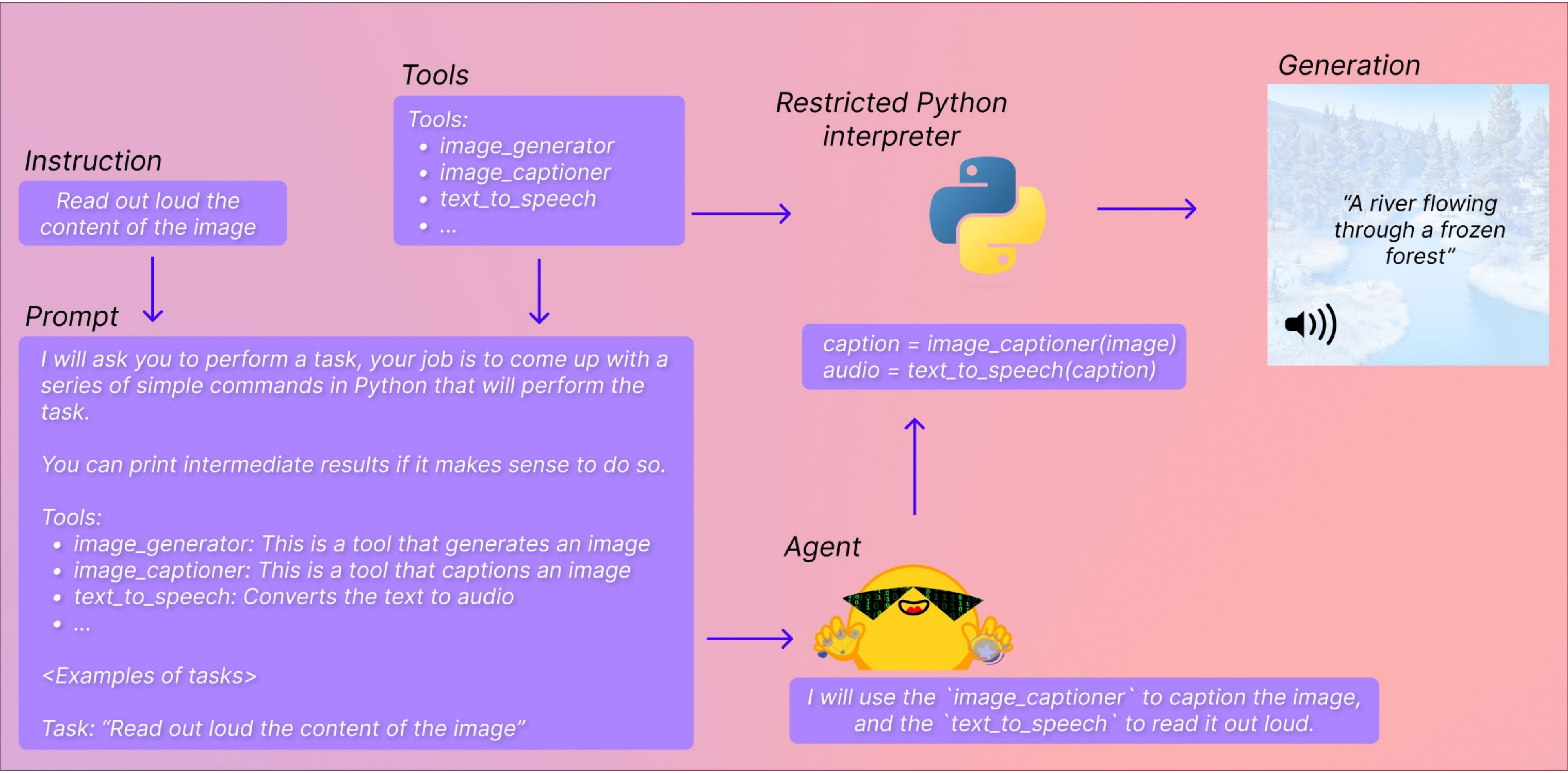


Figure from https://huggingface.co/docs/transformers/transformers_agents.

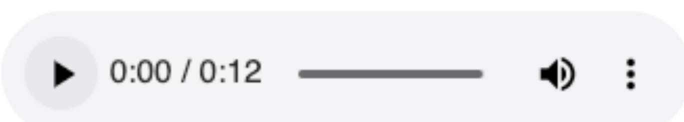
Hugging Face agents

```
audio = agent.run("Read out loud the summary of http://hf.co")
play_audio(audio)

==Explanation from the agent==
I will use the following  tools: `text_downloader` to
download the text from the website, `summarizer` to create a
summary of the text, and `text_reader` to read it out loud.

==Code generated by the agent==
text = text_downloader("https://hf.co")
summarized_text = summarizer(text)
print(f"Summary: {summarized_text}")
audio_summary = text_reader(summarized_text)

==Result==
Summary: Hugging Face is an AI community building the future.
More than 5,000 organizations are using Hugging Face's AI
chat models. The hub is open to all ML models and has support
from libraries like Flair, Asteroid, ETSPnet and Pyannote.
```



Example from https://huggingface.co/docs/transformers/transformers_agents.