AdvNLE Seminar 5

Sequence Classification / Propaganda Detection

Dr Julie Weeds, Spring 2023

Warm-up

- What is the difference between sequence labelling and sequence classification? How many examples can you come up with of applications of each.

- Can you think of ways that either could be applied in the following scenarios:
  - Machine translation?
  - Summarization?

Previously

- Distributional models of word meaning
  - how similar are two words based on how they are used in text?
- Language models
  - how likely is a sequence of words in a language?
- Neural language models
- Sequence Labelling (Named Entity Recognition)

This week

- Sequence classification
- Document / sequence representations
- Evaluation
- Distributional representations of meaning
- Composition
- Propaganda detection

Sequence classification

- Aka document classification, text classification
- Make a single sequence-level classification decision per sentence / per document
- Classification could be
  - *sentiment, topic, relevance … spam, hate speech, machine-generated …*
- Classification could be
  - *Binary or multi-class*
- Classification could be
  - *Hard or soft*

Evaluation

- Is class distribution balanced?
- Accuracy
- Precision, Recall, F1

Actual Class

Predicted Class	1	2	3	4
1	**TP**	**FP**	**FP**	**FP**
2	**FN**	TN	TN	TN
3	**FN**	TN	TN	TN
4	**FN**	TN	TN	TN

Actual Class

Predicted Class	1	0
1	TP	FP
0	FN	TN

- In multi-class scenario, need to compute precision, recall and F1 for each class
- Macro-average => unweighted average
- Micro-average => average weighted by the size of each class

Representing sequences for classification

- Classical document-level representation is **bag-of-words**
- What are the benefits and limitations of this?

	the	plot	is	great	not	boring	dull
1. The plot is great	1	1	1	1	0	0	0
2. The plot is not great	1	1	1	1	1	0	0
3. The great plot is not boring	1	1	1	1	1	1	0
4. The boring plot is not great	1	1	1	1	1	1	0
5. The plot is dull	1	1	1	0	0	0	1

Using Word Embeddings

- Find the **sum/centroid/max** of all of the word embeddings for words in the sequence
- Pass this as input to a classifier (e.g., logistic regression / SVM / NN)

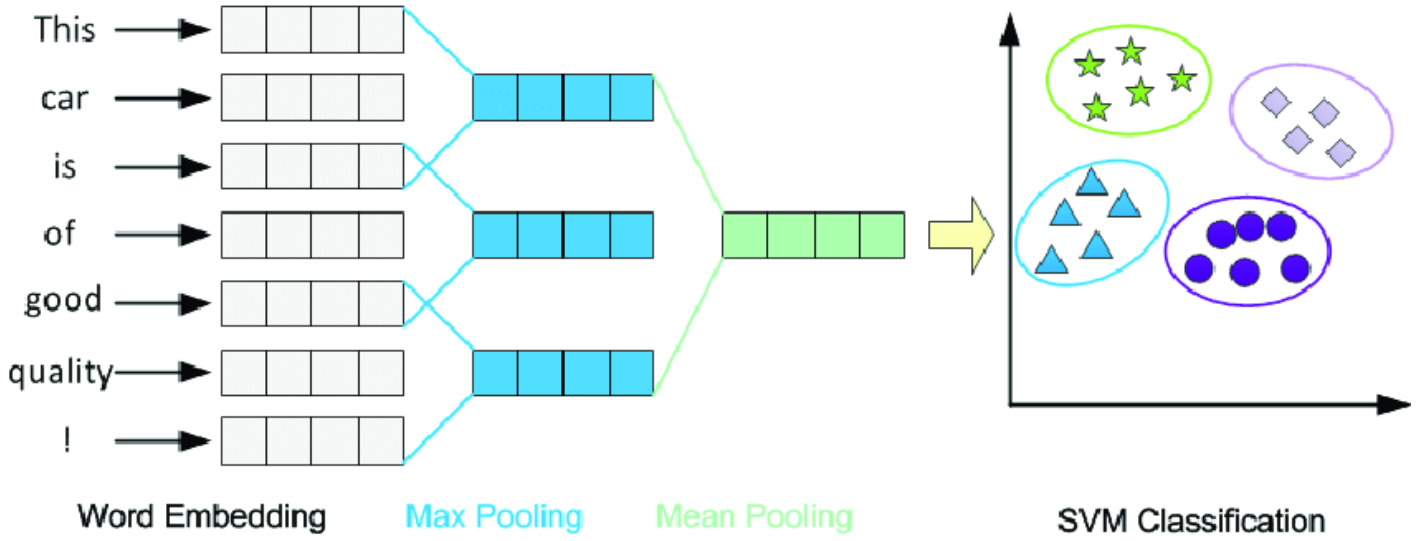


Figure from Pan et al. 2019

Question

- What are the benefits / disadvantages of using word embeddings in this way?

Distributed Representations of Word Meaning

- Distributional Hypothesis
- *"words which mean similar things tend to behave in similar ways"*
- i.e., they co-occur with similar words

	miaows	....	elected	...	decides	...	comfy	...
leader	0		3		5		0	
president	0		5		5		0	
ruler	0		3		5		0	
cat	5		0		0		1	
chair	0		0		0		5	

Distributed Representations of Sentential Meaning

- Can we do the same as for words?
  - Hypothesize that sentences which mean similar things tend to behave in similar ways?
  - Collect all of the contexts of sentences
  - Represent using vectors and compare
- Why / why not?

The Principle of Compositionality

- widely attributed to Gottlieb Frege, but assumed by others e.g., George Boole
- ***"The meaning of a complex expression is determined by the meanings of its constituent expressions and the rules used to combine them"***

Composition for Meaning Representations

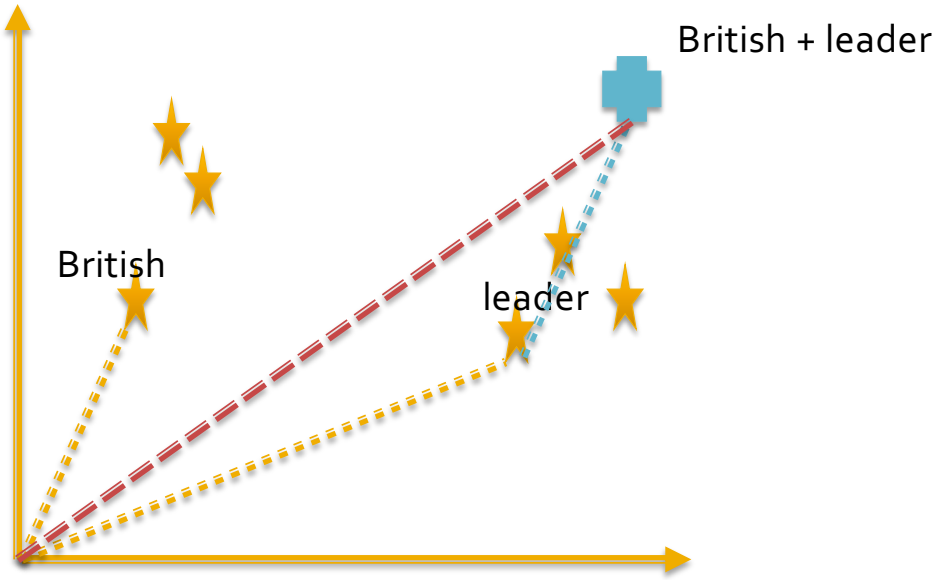
- **Constituent** expressions are **words**
- Words are represented by distributional representations / **embeddings** (e.g., Word2Vec or GloVe)

- So to get a representation of a sentence we need to …
- … compose the embeddings of the constituent words

- How? What are the rules for composition?

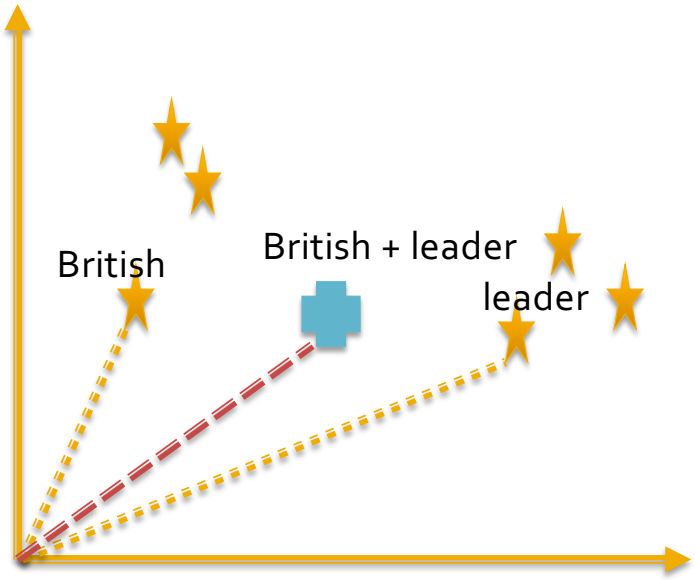
Additive composition

- Simply add the vectors



Additive composition

- Or average the vectors (find the centroid)



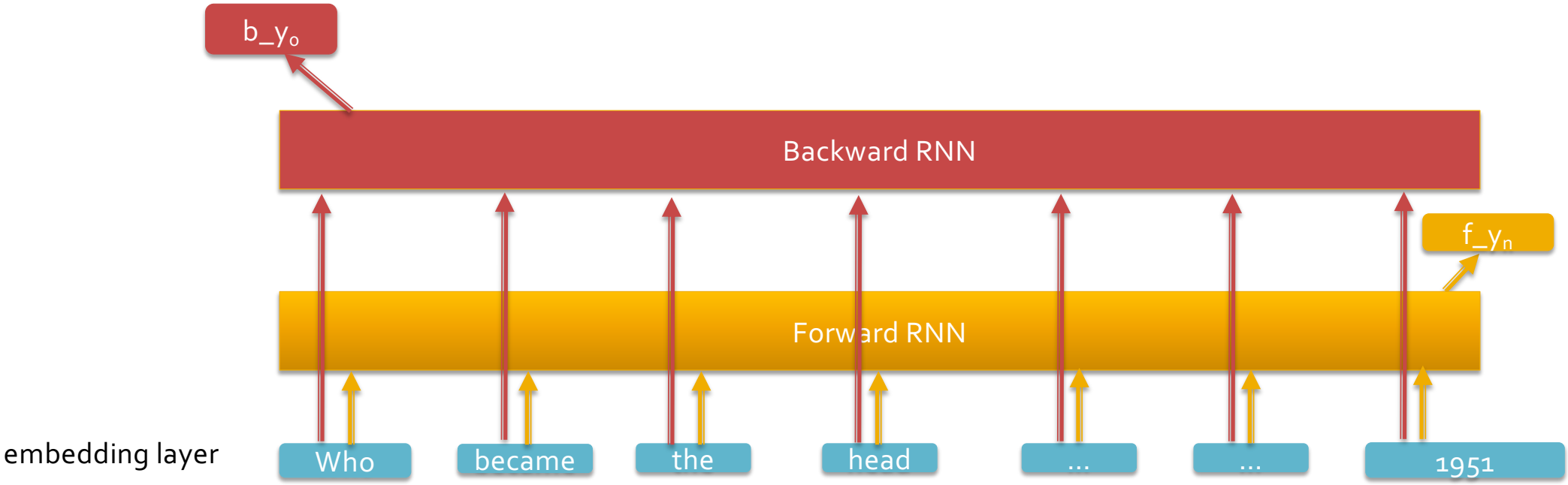
- remember, we are interested in cosine similarity between vectors
- → direction
- so very little difference between adding and averaging
- especially if vectors are normalised to unit length

Disadvantages of Adding Word Embeddings

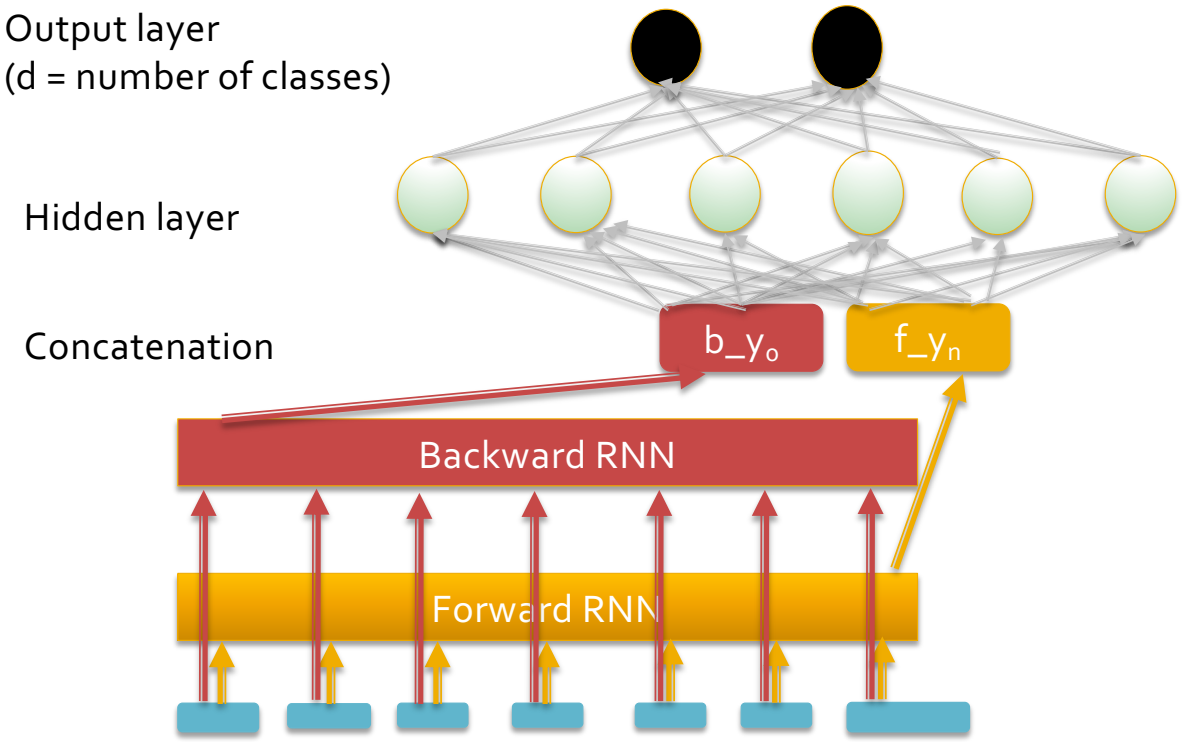
- Word embeddings are **uncontextualised**
  - contain a mixture of usages of all senses of the word
  - e.g., *head* as part of the body and *head* as leader
  - but only one sense is intended in a given sentence
- Pays no attention to word order or syntax
- What about function words such as *in*, *on*, *every* and *not*

Language modelling for Sequence Representations

- Represent a sequence by what is predicted to the left and right of it
- e.g., concatenation of [b_y_o,f_y_n]



Classification using representations from LMs



Representations taken from last hidden layer (not output layer)

Trained to predict previous word

Trained to predict next word

Propaganda Detection (Part 2)

Da San Martino et al. (2020) introduce a competition to detect and classify propaganda techniques in text. When reading this paper, do not be overly concerned with the different systems which took part in the competition. We will focus on the overall idea of propaganda detection, the two tasks introduced in this paper (span identification and technique classification), the dataset and the evaluation metrics. Once you have read the paper, consider the following questions.

1. What do you understand by the term propaganda and why might it be important to develop systems which can automatically detect propaganda in text?
2. Why is automatic propaganda detection difficult?
3. Give examples of 3 different propaganda techniques being used in text. Explain why this is propaganda.
4. What textual features might be useful to help a system detect propaganda?
5. Describe the pipeline proposed by the paper for propaganda identification. Can you think of any alternatives? What advantages / disadvantages are there of each?
6. How was the PTC-SemEval20 corpus collected and annotated? What do you understand by "the γ agreement on the annotated articles is on average 0.6"?
7. How do the authors evaluate systems on the span identification task?
8. Micro-average F_1 is used to evaluate systems on the technique classification task. The authors state that for a single-label task, this is equivalent to accuracy. Explain
9. Outline one method which could be used to carry out span identification.
10. Outline one method which could be used to carry out techniques classification.
11. Systems were evaluated for span identification on both the development set and the test set. Why do you think the results are not the same on both?
12. What is the predominant propaganda technique found in the corpus? If a system labelled every propaganda snippet with this label, how would it do? What do you think of the system results for techniques classification (Table 6)?

Question 1

- What do you understand by the term propaganda and why might it be important to develop systems which automatically detect propaganda in text?

Question 1

- What do you understand by the term propaganda and why might it be important to develop systems which automatically detect propaganda in text?

Propaganda comes in many forms, but it can be recognized by its persuasive function, sizable target audience, the representation of a specific group's agenda, and the use of faulty reasoning and/or emotional appeals (Miller, 1939). The term *propaganda* was coined in the 17th century, and initially referred to the propagation of the Catholic faith in the New World (Jowett and O'Donnell, 2012a, p. 2). It soon took a pejorative connotation, as its meaning was extended to also mean opposition to Protestantism. In more recent times, the Institute for Propaganda Analysis (Ins, 1938) proposed the following definition:

> **Propaganda.** *Expression of opinion or action by individuals or groups deliberately designed to influence opinions or actions of other individuals or groups with reference to predetermined ends.*

Recently, Bolsover and Howard (2017) dug deeper into this definition identifying its two key elements: (*i*) trying to influence opinion, and (*ii*) doing so on purpose.

Question 2

- Why is automatic propaganda detection difficult?

Question 2

- Why is automatic propaganda detection difficult?
- At least 3 reasons:
  1. ....
  2. ....
  3. ....

Question 3

- Give examples of 3 different propaganda techniques being used in text. Explain why this is propaganda

#	Technique	Snippet
1	Loaded language	**Outrage** as Donald Trump suggests injecting disinfectant to kill virus.
2	Name calling, labeling	WHO: Coronavirus emergency is '**Public Enemy Number 1**'
3	Repetition	I still have a **dream**. It is a **dream** deeply rooted in the American **dream**. I have a **dream** that one day . . .
4	Exaggeration, minimization	Coronavirus **'risk to the American people remains very low'**, Trump said.
5	Doubt	**Can the same be said for the Obama Administration**?
6	Appeal to fear/prejudice	**A dark, impenetrable and "irreversible" winter of persecution of the faithful by their own shepherds will fall**.
7	Flag-waving	Mueller attempts **to stop the will of We the People**!!! It's time to jail Mueller.
8	Causal oversimplification	**If France had not have declared war on Germany then World War II would have never happened.**
9	Slogans	**"BUILD THE WALL!"** Trump tweeted.
10	Appeal to authority	**Monsignor Jean-Franois Lantheaume, who served as first Counsellor of the Nunciature in Washington, confirmed that "Vigan said the truth. That's all."**
11	Black-and-white fallacy	Francis said these words: "**Everyone is guilty for the good he could have done and did not do . . . If we do not oppose evil, we tacitly feed it**."
12	Thought-terminating cliché	**I do not really see any problems there.** Marx is the President.
13	Whataboutism	President Trump —**who himself avoided national military service** in the 1960's— keeps beating the war drums over North Korea.
	Straw man	"Take it seriously, but with a large grain of salt." **Which is just Allen's more nuanced way of saying: "Don't believe it**."
	Red herring	**"You may claim that the death penalty is an ineffective deterrent against crime – but what about the victims of crime? How do you think surviving family members feel when they see the man who murdered their son kept in prison at their expense? Is it right that they should pay for their son's murderer to be fed and housed?"**
14	Bandwagon	He tweeted, "**EU no longer considers #Hamas a terrorist group. Time for US to do same."**
	Reductio ad hitlerum	"Vichy journalism," a term which now fits so much of the mainstream media. **It collaborates in the same way that the Vichy government in France collaborated with the Nazis.**

Table 1: The 14 propaganda techniques with examples, where the propaganda span is shown in bold.

Question 4

- What textual features might be useful to help a system detect propaganda?

#	Technique	Snippet
1	Loaded language	**Outrage** as Donald Trump suggests injecting disinfectant to kill virus.
2	Name calling, labeling	WHO: Coronavirus emergency is '**Public Enemy Number 1**'
3	Repetition	I still have a **dream**. It is a **dream** deeply rooted in the American **dream**. I have a **dream** that one day . . .
4	Exaggeration, minimization	Coronavirus **'risk to the American people remains very low'**, Trump said.
5	Doubt	**Can the same be said for the Obama Administration**?
6	Appeal to fear/prejudice	**A dark, impenetrable and "irreversible" winter of persecution of the faithful by their own shepherds will fall**.
7	Flag-waving	Mueller attempts **to stop the will of We the People**!!! It's time to jail Mueller.
8	Causal oversimplification	**If France had not have declared war on Germany then World War II would have never happened.**
9	Slogans	**"BUILD THE WALL!"** Trump tweeted.
10	Appeal to authority	**Monsignor Jean-Franois Lantheaume, who served as first Counsellor of the Nunciature in Washington, confirmed that "Vigan said the truth. That's all."**
11	Black-and-white fallacy	Francis said these words: "**Everyone is guilty for the good he could have done and did not do . . . If we do not oppose evil, we tacitly feed it**."
12	Thought-terminating cliché	**I do not really see any problems there.** Marx is the President.
13	Whataboutism	President Trump —**who himself avoided national military service** in the 1960's— keeps beating the war drums over North Korea.
	Straw man	"Take it seriously, but with a large grain of salt." **Which is just Allen's more nuanced way of saying: "Don't believe it**."
	Red herring	**"You may claim that the death penalty is an ineffective deterrent against crime – but what about the victims of crime? How do you think surviving family members feel when they see the man who murdered their son kept in prison at their expense? Is it right that they should pay for their son's murderer to be fed and housed?"**
14	Bandwagon	He tweeted, "**EU no longer considers #Hamas a terrorist group. Time for US to do same."**
	Reductio ad hitlerum	"Vichy journalism," a term which now fits so much of the mainstream media. **It collaborates in the same way that the Vichy government in France collaborated with the Nazis.**

Table 1: The 14 propaganda techniques with examples, where the propaganda span is shown in bold.

Question 5

- Describe the pipeline proposed by the paper for propaganda identification. Can you think of any alternatives? What advantages / disadvantages are there of each?

Question 5

- Describe the pipeline proposed by the paper for propaganda identification. Can you think of any alternatives? What advantages / disadvantages are there of each?

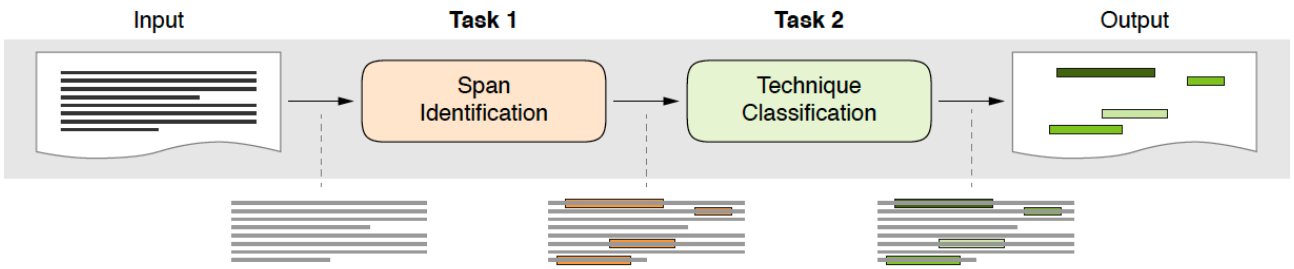


Figure 1: The full propaganda identification pipeline, including the two subtasks: Span Identification and Technique Classification.

Question 6

- How was the PTC-SemEval20 corpus collected and annotated? What do you understand by "the gamma agreement on the annotated articles is on average 0.6"?

Question 6

- How was the PTC-SemEval20 corpus collected and **annotated**?

Input article

Manchin says Democrats acted like **babies** at the SOTU

In a glaring sign of just how **stupid** and **petty** things have become in Washington these days [...] State of the Union speech not looking as though Trump **killed his grandma**. [...]

Annotation file

Article ID	Technique	Start	End
123456	Name_Calling	34	40
123456	Loaded_Language	83	89
123456	Loaded_Language	94	99
123456	Loaded_Language	350	368
...	...		

Figure 2: Example of a plain-text article (left) and its annotation (right). The *Start* and the *End* columns are the indices representing the character span of the spotted technique.

Question 6

- **What do you understand by "the gamma agreement on the annotated articles is on average 0.6"?**

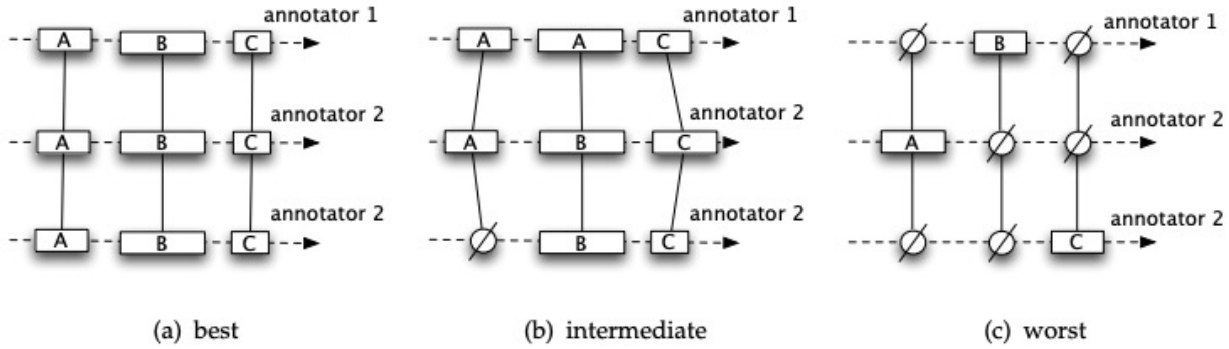


Figure 11
Examples of best, intermediate, and worst possible disorders.

$$\forall s \in c, \gamma = 1 - \frac{\delta(s)}{\delta_e(c)} \tag{8}$$

If all annotators perfectly agree (Figure 11a), $\gamma = 1$. Figure 11c corresponds to the worst case, where the annotators are worse than annotating at random, with $\gamma < 0$. Figure 11b shows an intermediate situation.

From Mathet at al. 2015

Question 7

- How do the authors evaluate systems on the span identification task?

Question 7

- How do the authors evaluate systems on the span identification task?

Let d be a news article in a set D. A gold span t is a sequence of contiguous indices of the characters composing a text fragment $t \subseteq d$. For example, in Figure 4 (top-left) the gold fragment *"stupid and petty"* is represented by the set of indices $t_1 = [4, 19]$. We denote with $T_d = \{t_1, \ldots, t_n\}$ the set of all gold spans for an article d and with $T = \{T_d\}_d$ the set of all gold annotated spans in D. Similarly, we define $S_d = \{s_1, \ldots, s_m\}$ and S to be the set of predicted spans for an article d and a dataset D, respectively. We compute precision P and recall R by adapting the formulas in (Potthast et al., 2010):

$$P(S,T) = \frac{1}{|S|} \cdot \sum_{d \in D} \sum_{s \in S_d, t \in T_d} \frac{|(s \cap t)|}{|t|}, \quad (1)$$

$$R(S,T) = \frac{1}{|T|} \cdot \sum_{d \in D} \sum_{s \in S_d, t \in T_d} \frac{|(s \cap t)|}{|s|}. \quad (2)$$

Question 8

- Micro-average F1 is used to evaluate systems on the techniques classification task. The authors state that for a single-label tast, this is equivalent to accuracy. Explain.

Evaluation

- Is class distribution balanced?
- Accuracy
- Precision, Recall, F1

Actual Class

Predicted Class	1	2	3	4
1	**TP**	**FP**	**FP**	**FP**
2	**FN**	TN	TN	TN
3	**FN**	TN	TN	TN
4	**FN**	TN	TN	TN

Actual Class

Predicted Class	1	0
1	TP	FP
0	FN	TN

- In multi-class scenario, need to compute precision, recall and F1 for each class
- Macro-average => unweighted average
- Micro-average => average weighted by the size of each class

Question 9

- Outline one method which could be used to carry out span identification.

Question 10

- Outline one method which could be used to carry out technique classification.

Question 11

- Systems were evaluated for span identification on both the development set and the test set. Why do you think the results are not the same on both?

Question 11

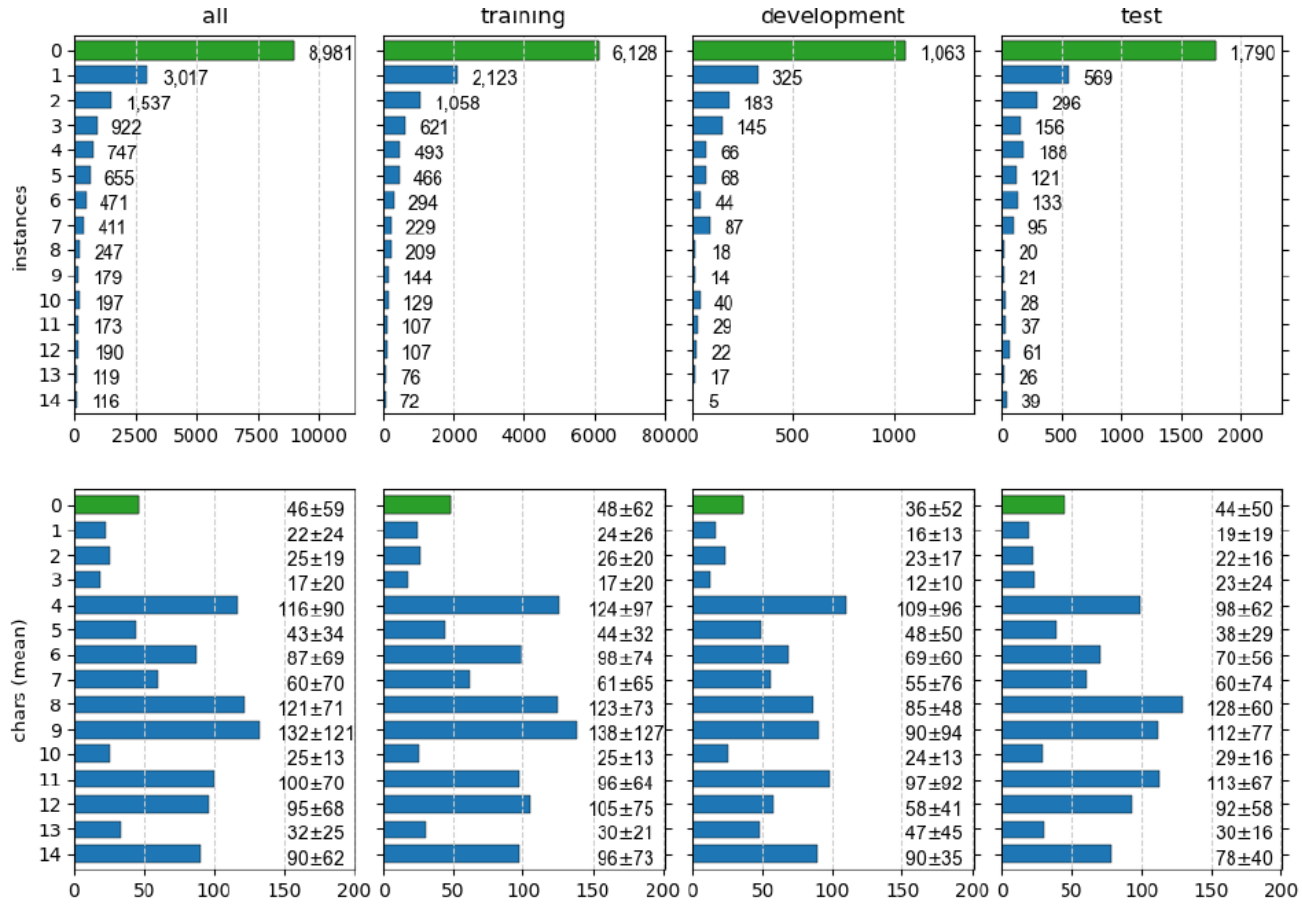
	Test				Development			
Team	**Rnk**	**F_1**	**P**	**R**	**Rnk**	**F_1**	**P**	**R**
Hitachi	1	**51.55**	56.54	47.37	4	**50.12**	42.26	61.56
ApplicaAI	2	49.15	59.95	41.65	3	52.19	47.15	58.44
aschern	3	49.10	53.23	45.56	5	49.99	44.53	56.98
LTIatCMU	4	47.66	50.97	44.76	7	49.06	43.38	56.47
UPB	5	46.06	58.61	37.94	8	46.79	42.44	52.13
Fragarach	6	45.96	54.26	39.86	12	44.27	41.68	47.21
NoPropaganda	7	44.68	55.62	37.34	9	46.13	40.65	53.31
CyberWallE	8	43.86	42.16	45.70	17	42.39	33.45	57.86
Transformers	9	43.60	49.86	38.74	14	43.06	40.85	45.52
SWEAT	10	43.22	52.77	36.59	16	42.51	42.97	42.06
YNUtaoxin	11	43.21	55.62	35.33	11	44.35	40.74	48.67
DREAM	12	43.10	54.54	35.63	19	42.15	42.66	41.65
newsSweeper	13	42.21	46.52	38.63	10	44.45	38.76	52.10
PsuedoProp	14	41.20	41.54	40.87	22	39.32	34.27	46.11
Solomon	15	40.68	53.95	32.66	15	42.86	43.24	42.49
YNUHPCC	16	40.63	36.55	45.74	18	42.27	32.08	61.95
NLFIIT	17	40.58	50.91	33.73	21	39.67	35.04	45.72
PALI	18	40.57	53.20	32.79	2	52.35	49.64	55.37
UESTCICSA	19	39.85	56.09	30.90	13	44.17	43.21	45.18
TTUI	20	39.84	**66.88**	28.37	6	49.59	48.76	50.44
BPGC	21	38.74	49.39	31.88	25	36.79	34.72	39.12
DoNotDistribute	22	37.86	42.36	34.23	24	37.73	32.41	45.12
UTMNandOCAS	23	37.49	37.97	37.03	31	34.35	23.65	62.69
Entropy	24	37.23	41.68	33.63	32	32.89	30.82	35.25
syrapropa	25	36.20	49.53	28.52	1	53.40	39.88	80.80

- Systems were evaluated for span identification on both the development set and the test set. Why do you think the results are not the same on both?

Question 12

- What is the predominant propaganda technique found in the corpus? If a system labelled every propaganda snippet with this label, how would it do? What do you think of the system results for technique classification (Table 6)?

- What is the predominant propaganda technique found in the corpus?
- If a system labelled every propaganda snippet with this label, how would it do?



What do you think of the system results for technique classification (Table 6)?

Rnk	Team	Overall	1	2	3	4	5	6	7	8	9	10	11	12	13	14
1	ApplicaAI	**62.07**	77.12	74.38	**54.55**	33.59	56.23	**45.49**	**69.43**	22.73	51.28	48.15	49.02	39.22	25.00	8.33
2	aschern	62.01	77.02	**75.65**	53.38	32.65	59.44	41.78	66.35	25.97	**54.24**	35.29	**53.57**	**42.55**	18.87	14.93
3	Hitachi	61.73	75.64	74.20	37.88	34.58	**63.43**	38.94	68.02	**36.62**	45.61	40.00	47.92	29.41	**26.92**	4.88
4	Solomon	58.94	74.66	70.75	42.53	28.44	61.82	39.39	61.84	19.61	50.75	26.67	42.00	38.10	0.00	4.88
5	newsSweeper	58.44	75.32	74.23	20.69	37.10	56.55	42.80	60.53	19.72	50.75	41.67	25.00	21.62	8.00	13.04
6	NoPropaganda	58.27	**77.17**	73.90	42.71	**37.99**	56.27	38.02	59.30	12.12	42.42	23.26	8.70	23.26	0.00	0.00
7	Inno	57.99	73.31	74.30	24.89	35.39	58.65	45.09	59.41	24.32	43.75	43.14	40.40	29.63	19.36	10.71
8	CyberWallE	57.37	74.68	70.92	47.68	28.34	58.65	39.84	54.38	15.39	39.39	14.63	23.68	23.81	0.00	12.25
9	PALI	57.32	74.29	69.09	24.56	28.57	58.97	36.59	61.62	30.59	39.22	27.59	39.62	40.82	20.90	**28.57**
10	DUTH	57.21	73.71	71.41	20.10	28.24	59.16	33.33	58.95	26.23	34.78	44.44	33.33	27.03	17.78	9.30
11	DiSaster	56.65	74.49	68.10	20.44	30.64	59.12	35.25	58.25	14.63	42.55	51.16	26.67	19.05	4.35	20.41
12	djichen	56.54	73.21	68.38	29.75	31.42	60.00	33.65	56.19	22.79	30.77	37.50	43.81	27.91	18.87	20.83
13	SocCogCom	55.81	72.18	67.34	18.88	34.86	60.40	31.62	54.26	6.35	40.91	28.57	26.51	23.53	10.00	9.76
14	TTUI	55.64	73.22	68.49	21.18	32.20	57.40	41.48	61.68	23.08	37.50	28.24	35.29	25.00	20.29	24.56
15	JUST	55.31	71.96	64.73	21.94	29.57	58.26	37.10	62.56	27.27	33.33	**48.89**	28.89	31.82	28.57	24.49
16	NLFIIT	55.25	72.55	69.30	21.55	30.30	55.66	24.89	63.32	0.00	41.67	29.63	32.10	13.64	0.00	9.30
17	UMSIForeseer	55.14	73.02	70.79	21.49	28.57	57.21	31.97	56.14	0.00	39.22	29.41	0.00	14.29	0.00	9.76
18	BPGC	54.81	71.58	67.51	23.74	33.47	53.78	33.65	58.93	24.18	40.00	30.77	40.00	20.69	20.90	12.50
19	UPB	54.30	70.09	68.86	20.00	30.62	52.55	30.00	55.87	16.95	34.62	20.00	19.72	22.86	4.88	0.00
20	syrapropa	54.25	71.47	68.44	30.77	28.10	56.14	29.77	57.02	21.51	29.03	31.58	30.61	28.57	9.09	19.61
21	WMD	52.01	69.33	64.67	13.89	25.46	53.94	29.20	52.08	5.71	6.90	7.14	0.00	7.41	0.00	5.00
22	YNUHPCC	50.50	68.08	62.33	17.72	21.54	51.04	26.40	55.56	3.45	27.59	29.79	38.38	17.78	15.00	13.79
23	UESTCICSA	49.94	68.23	66.88	27.96	25.44	44.99	22.75	53.14	3.74	41.38	12.77	11.27	28.57	3.70	0.00
24	DoNotDistribute	49.72	68.44	60.65	19.44	27.23	46.25	29.75	53.76	14.89	28.07	22.64	24.49	12.25	9.68	4.55
25	NTUAAILS	46.37	65.79	54.55	18.43	29.66	48.75	28.31	46.47	0.00	13.79	36.36	0.00	11.43	4.08	9.76
26	UAIC1860	41.17	62.33	42.97	11.16	21.01	36.41	22.12	38.78	7.60	11.43	17.39	2.90	5.56	4.26	9.76
27	UNTLing	39.11	62.57	36.74	7.78	11.82	32.65	5.29	40.48	2.86	17.65	4.35	0.00	0.00	0.00	0.00
28	HunAlize	37.10	58.59	15.82	2.09	23.81	31.76	11.83	29.95	7.84	4.55	6.45	8.00	0.00	0.00	0.00
29	Transformers	26.54	47.55	24.06	2.86	0.00	0.98	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
30	Baseline	25.20	46.48	0.00	19.26	14.42	29.14	3.68	6.20	11.56	0.00	0.00	0.00	0.00	0.00	0.00
31	Entropy	20.39	37.74	15.49	5.83	6.39	12.81	6.32	4.95	7.41	0.00	3.92	2.27	0.00	6.78	0.00
32	IJSE8	19.72	38.07	14.70	4.92	8.23	15.47	7.07	8.57	2.27	0.00	0.00	0.00	0.00	0.00	0.00

Table 6: **Technique classification F_1 performance on the test set.** The systems are ordered on the basis of the final ranking. Columns **1** to **14** show the performance for each of the propaganda techniques (cf. Section 2). The best score for each technique appears highlighted. (*Note: We found a bug in the evaluation script after the end of the competition. The correct ranking, shown in Appendix B, does not differ substantially from above.*)

AdvNLP Week 7

Machine Translation

Dr Julie Weeds, Spring 2024

Warm-up

Compare the phrases on the left with the phrases on the right…

- panda car
- memory lane
- rocket science
- crash course
- rat race

- car park
- climate change
- application form
- student house
- bank account

Previously

- Distributional semantics
- Language models
- Neural language models
- Sequence labelling
- Sequence classification

Overview

- What makes machine translation (MT) hard?
- Evaluation of MT
- Classical MT (Pre 1990s)
- Statistical MT (1990-2015)
  - Word-based models
  - Phrase-based models
- Neural MT (2015 -)
  - Encoder-decoder models

MT Lecture Questions

1. What makes Machine Translation a hard problem?
2. What aspect of MT can be evaluated by monolingual raters and what aspect requires bilingual raters?
3. What do BLEU and chrF have in common? How are they different?
4. What are some of the key components / choices in setting up a statistical MT system?
5. Why should neural MT work better?

Why is/was MT hard?

- Lexical differences
- Structural differences (morphological differences and syntactic differences)
- Study of systematic cross-linguistic similarities and differences is called **linguistic typology**
  - See World Atlas of Language Structures (Dryer and Haspelmath, 2013)

BLEU

- computes modified precision for unigrams, bigrams, trigrams and often quadrigrams
- combines using geometric mean
- incorporates a penalty for translations which are too short
- good for evaluation of incremental changes to same general architecture
- see Papineni 2002

chrF

- chrP = percentage of character 1-grams, 2-grams, … , k-grams in the hypothesis that occur in the reference, averaged
- chrR = percentage of character 1-grams, 2-grams, …, k-grams in the reference that occur in the hypothesis, averaged

$$chrF\beta = (1+\beta)^2 \frac{chrP\ .chrR}{\beta^2.chrP + chrR}$$

β=2 gives twice as much weight to chrR as to chrP

Key points in statistical MT

- focus on the result NOT the process
- based on probabilities derived from *parallel corpora*
- Estimation maximization to obtain word translation probabilities
- Alignment models e.g. word alignment vs phrase alignment
- Generative vs discriminative models
- Decoding is a search problem
- Inability to generalise

Part 3

Neural Machine Translation

Neural Machine Translation (NMT)

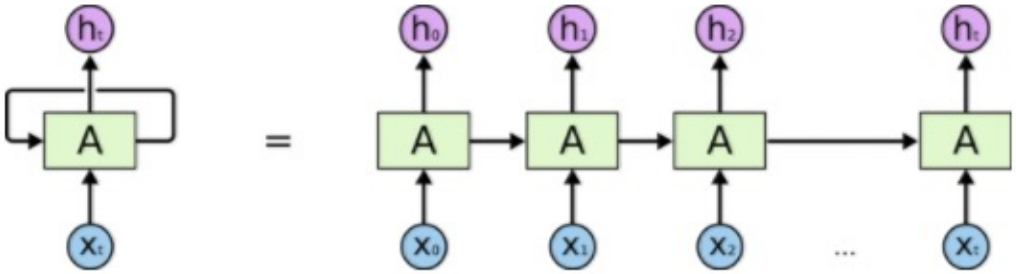
- Continuous representations (e.g., word2vec embeddings) for words and phrases are able to capture their morphological, syntactic and semantic similarity
- As in SMT, train on parallel corpora where sentences are aligned
- Maximise the probability of the sequence of tokens in the target language $y_1 \ldots y_m$ given the sequence of tokens in the source language $x_1 \ldots x_n$

$$P(y_1, \ldots, y_m | x_1, \ldots, x_n)$$

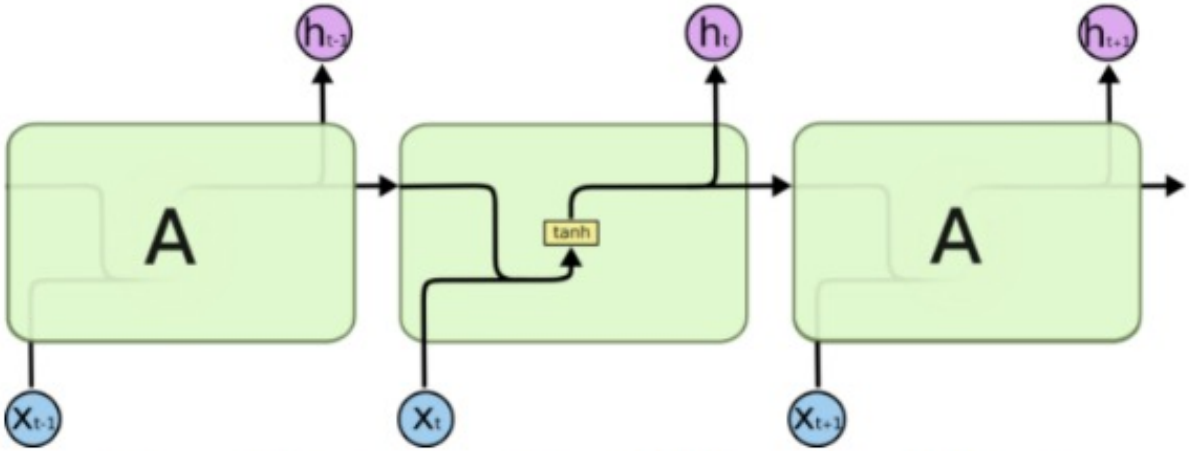
Basic architecture for NMT

- Encoder – decoder architecture
  - Aka sequence-to-sequence or seq2seq architecture
- 2 recurrent neural networks (RNNs) – one to consume the input text sequence and one to generate translated output text.

RNNs



An unrolled recurrent neural network.

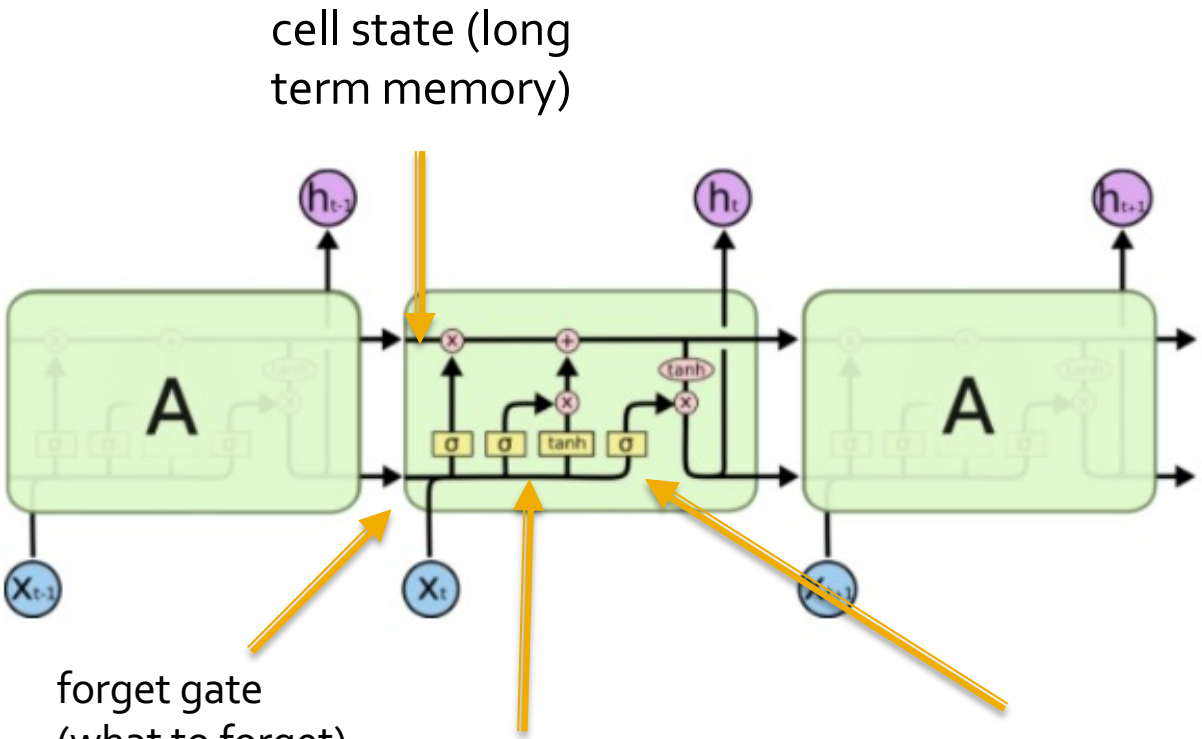


The repeating module in a standard RNN contains a single layer.

RNNs are very effective at learning language models i.e., P(E) the probability of a sentence in a given language. During training, the error (i.e., difference between output and next word) is back-propagated to update the weights used to combine X_t and h_{t-1} AND the representations of the words (X_t)

Long short term memory networks (LSTMs)

- Simple RNNs struggle with long term dependencies e.g., "He grew up in Spain. He speaks fluent …"



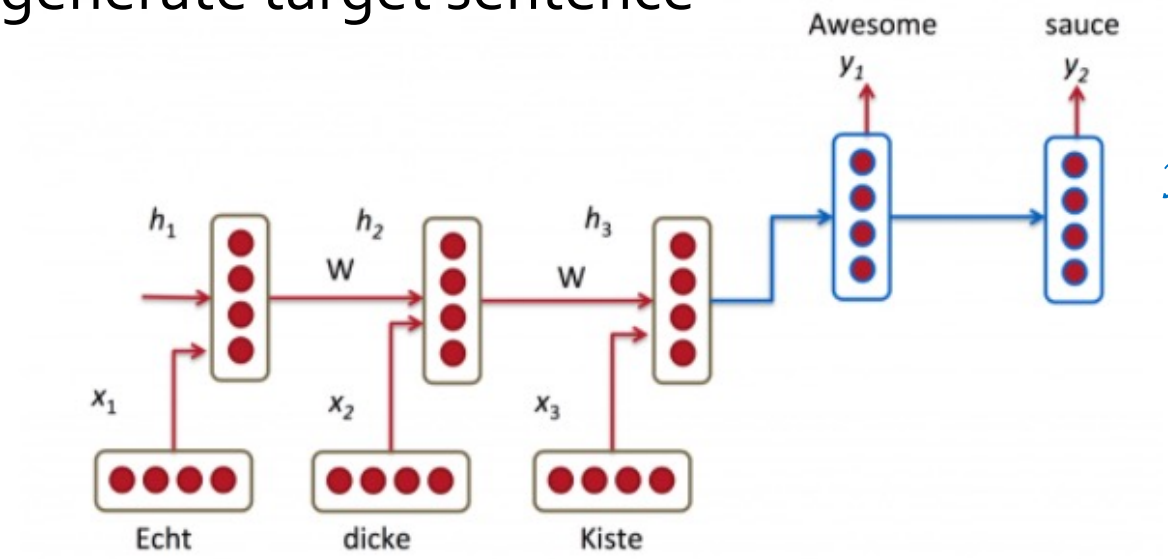
LSTMs overcome this problem by having 4 interacting layers in each repeated module.

Basic architecture for NMT

- RNN1, the encoder, builds a representation of the source sentence $x = x_1 \ldots x_n$

$$h = encoder(x)$$

- The output from RNN1 (after the complete source sentence has been read) is input to RNN2, the decoder to generate target sentence



$$y_{i+1} = decoder(h, y_1 \ldots y_i)$$

Encoder-decoder details

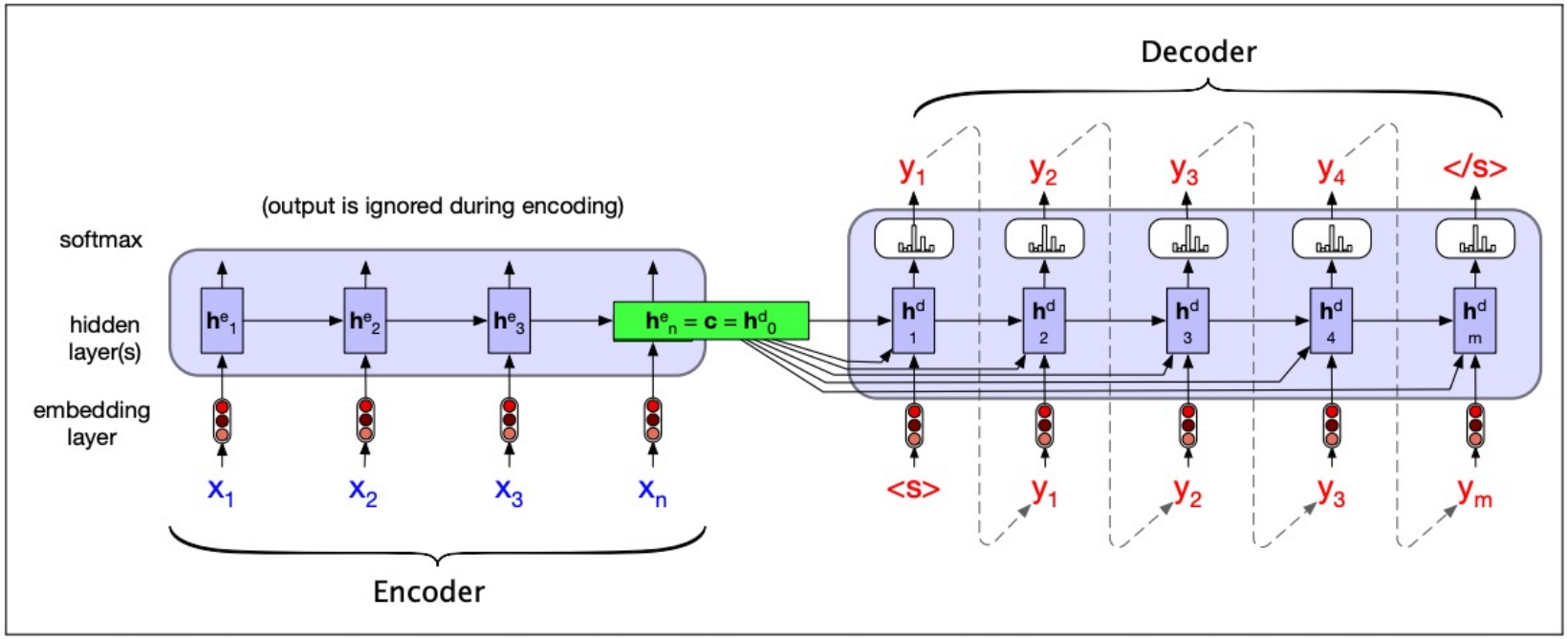


Figure 9.18 A more formal version of translating a sentence at inference time in the basic RNN-based encoder-decoder architecture. The final hidden state of the encoder RNN, h_n^e, serves as the context for the decoder in its role as h_0^d in the decoder RNN, and is also made available to each decoder hidden state.

Possible weaknesses

- Slow training and inference speed
- Ineffectiveness at dealing with rare words
- Output sentences that do not translate all words of the input sentence
- Difficulty in translating long sentences since the encoder output (or context) needs to encode the whole sentence
  - Information from start of sentence may be lost

Rare words (Luong et al. 2015)

- Due to computational constraints, NMT systems usually limited to top 30K-80K of most frequent words in each language
- Unknown/rare words can be translated using a dictionary or exact copy provided it is known which source word generated UNK token in target.
- Problem when sentence contains multiple rare words
- Luong et al. first use a word alignment of parallel corpora and annotate unknown words with positional information (e.g., UNK1)
- Output from NMT can then be post-processed

Subword tokenization

- Word vocabulary is huge and sparse
- Character vocabulary is small and dense, but lacking in semantic meaning
- Subword tokenization provides a compromise
- Frequent words tend to be a token whereas rare words will be broken down into subwords based on character n-grams
- Shared vocabulary for source and target languages – makes it easy to copy tokens like names from source to target
- Common algorithms include
  - BytePiece Encoding (BPE)
  - Wordpiece algorithm
  - Unigram / SentencePiece algorithm

Long sentences

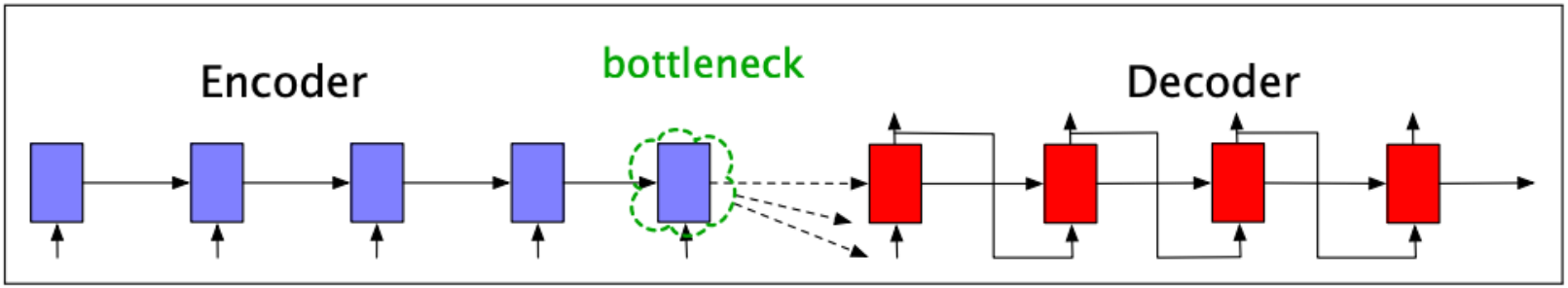


Figure 9.20 Requiring the context c to be only the encoder's final hidden state forces all the information from the entire source sentence to pass through this representational bottleneck.

- Attention (more on this next week) provides a way for the decoder to get information from all of the hidden states of the encoder rather than just the last hidden state

Attention

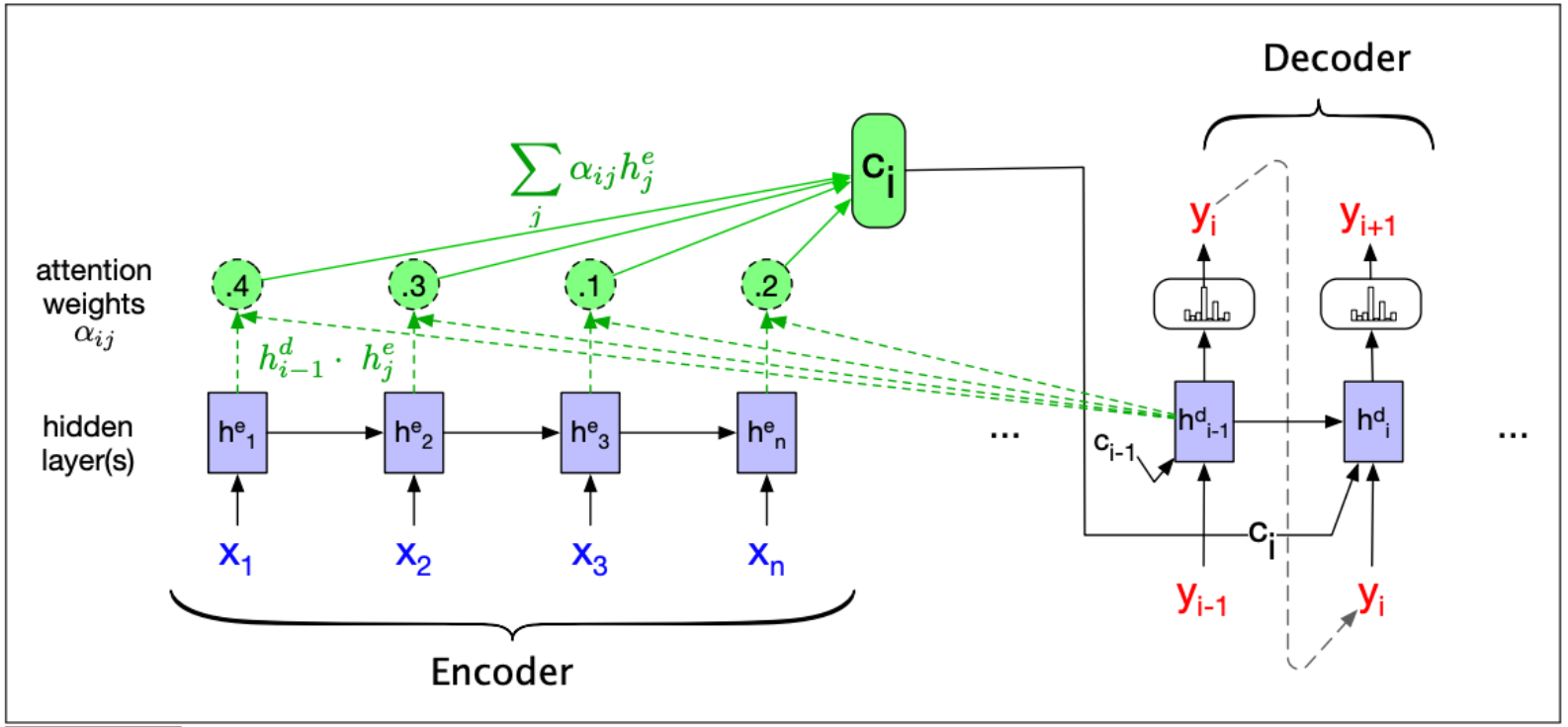
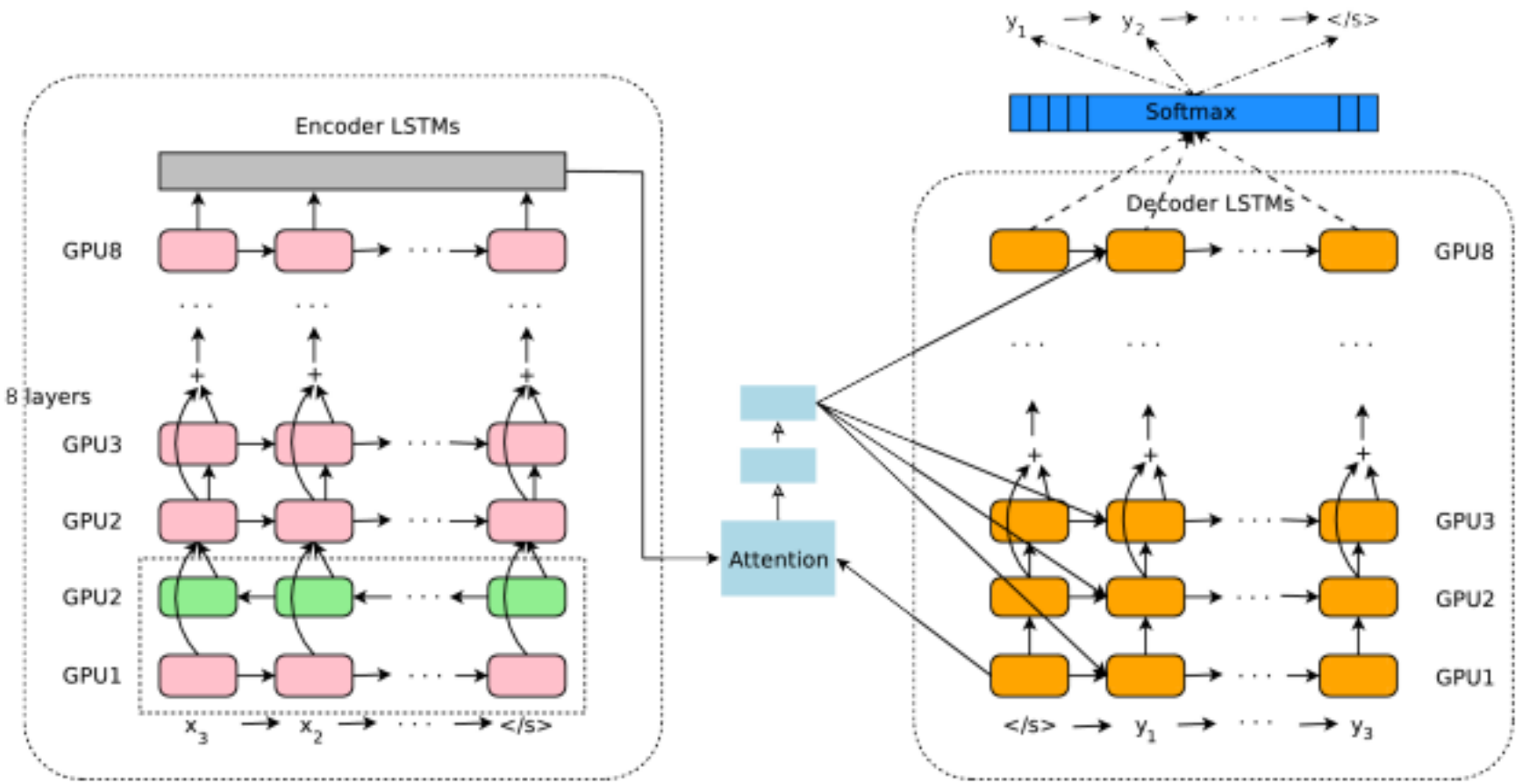


Figure 9.22 A sketch of the encoder-decoder network with attention, focusing on the computation of $\mathbf{c}_i$. The context value $\mathbf{c}_i$ is one of the inputs to the computation of $\mathbf{h}_i^d$. It is computed by taking the weighted sum of all the encoder hidden states, each weighted by their dot product with the prior decoder hidden state $\mathbf{h}_{i-1}^d$.

Google NMT (GNMT)

- Recurrent networks are LSTMs with attention (8 layers - residual connections between layers to encourage gradient flow)
- For parallelism, the attention from the decoder network connect to top layer of encoder network
- Low-precision arithmetic for inference, accelerated using special hardware (Google's TPU)
- Rare words dealt with using sub-word units (balancing flexibility of single characters with efficiency of full words)
- Beam search techniques includes a length normalization procedure and a coverage penalty to encourage model to translate all of the input

GNMT Architecture



Transformers and LLMs in MT?

- Transformers generally have higher performance than LSTMS and GRUs
- Generally replacing seq2seq architectures
- More on this in weeks 8-10

- https://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=9355969
- https://arxiv.org/abs/2209.07417

Open questions

- High resource vs low resource languages

Evaluation Exercise

- What are the chrF1 and chrF2 scores for each of the following hypothesis translations if k = 2?

REF	witness for the past,	Unigram precision	Unigram recall	Bigram precision	Bigram recall	ChrF1	ChrF2
HYP1	witness of the past,						0.86
HYP2	past witness						0.62

References

- Nal Kalchbrenner and Phil Blunsom. 2013. Recurrent continuous translation models, In *EMNLP*
- Philip Koehn. 2004. Pharoah: A Beam Search Decoder for Phrase-Based Statistical Machine Translation Models. In *AMTA*
- Minh-Thang Luong, Ilya Sutskever, Quoc Le, Oriol Vinyals and Wojciech Zaremba. 2015. Addressing the Rare Word Problem in Neural Machine Translation. In *ACL*
- Ilya Sutskever, Oriol Vinyals and Quoc Le. Sequence to Sequence Learning with Neural Networks. In *NIPS*
- Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc Le and Mohammad Norouzi. 2016. Google's Neural Machine Translation System: Bridging the Gap between Human and Machine Translation. ArXiv Oct 2016

AdvNLE Seminar 10

Using LLMs and the Future

Dr Julie Weeds, Spring 2024

Previously …

- Distributional representations of meaning
- Neural Language modelling
- Contextualised word embeddings
- Large language models
  - BERT (Bi-directional Encoder Representations from Transformers)
  - Pre-training
  - Fine-tuning

Today

- More distant relatives of BERT
  - GPT and ChatGPT
- Generative applications with LLMs
  - MT
  - Summarization
  - Question-answering
- Trustworthy and responsible AI
- Environmental impact of LLMs
- Revision

More Distant Relatives of BERT

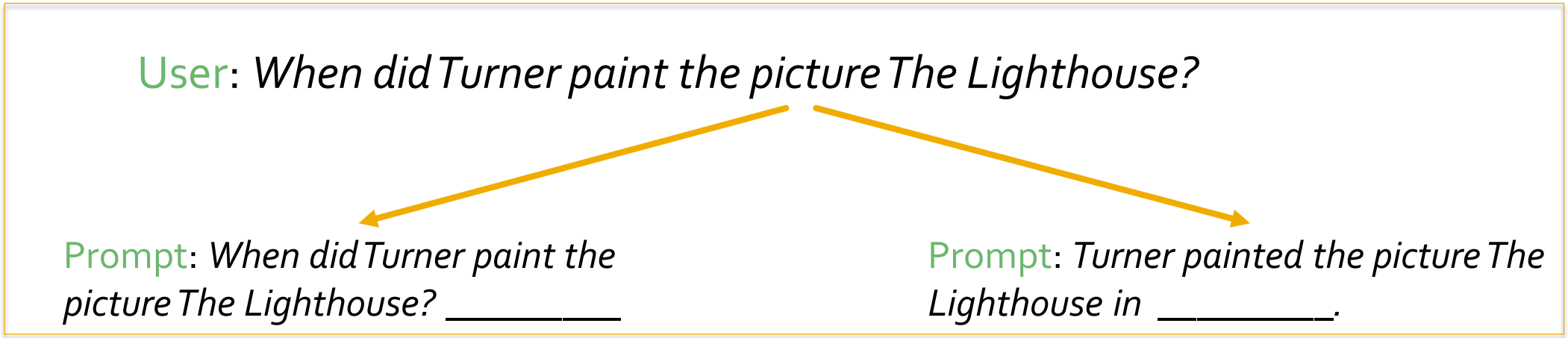
- Other Pretrained Large Language Models, generally still based on transformers e.g.,
  - GPT (GPT-2, GPT-3, ChatGPT GPT-4 …)
  - Turing-NLG,
  - XLNet,
  - Electra
  - Dolly
  - NeMo
  - BLOOM
  - LLaMa
  - PaLM2

Generative Pre-trained Transformer 3 (GPT-3)

- Brown et al. 2020: *Language Models are Few Shot Learners*
- autoregressive language model
  - this means it predicts the next token rather than masked tokens
- variable length inputs but uni-directional in nature
- largest non-sparse language model: 175 billion parameters, 10x bigger than competitors
- Trained on Common Crawl, WebText2, Books1, Books2 and Wikipedia
- No fine-tuning. Used to generate answers using a few-shot training / prompting paradigm

Generating responses

- Take a ***prompt*** and use a language model to predict what comes next or what fills in the gaps.
- Working out what is the best prompt strategy (how to convert the user utterance into a prompt for the LLM) is known as **prompt engineering**



- Find the word(s) that fills in the blank(s) with the highest probability according to the large language model (pretrained on the very large corpus)

What could possibly go wrong?

What could possibly go wrong?

- Large Language Models are trained to give plausible / believable answers based on large text corpora
- If 2 (or more words) often occur together in the training data, they may lead to a higher probability response even if factually incorrect in the specific context –
  - This is referred to as "***hallucination***" where a large language model adds information which has no basis in say the text being summarized or translated
- Text corpora may contain factually incorrect documents (e.g., fiction) and / or text which exhibits ***biases*** so these can be replicated in generated texts

Common hallucinations

- Based on your knowledge of LLMs, what do you think the most common types of errors and hallucinations are likely to be?

Genuine example (Feb 24)

You

Where in Rotherfield is the Six Bells pub?

ChatGPT

The Six Bells pub in Rotherfield is located at Church Street, Rotherfield, Crowborough, East Sussex, TN6 3LQ, United Kingdom.

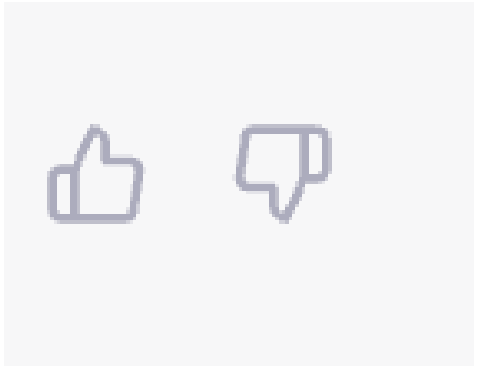
- There is no Six Bells pub in Rotherfield
- There is no Church Street in Rotherfield (only a Church Road)
- TN6 3LQ is Court Meadow, Rotherfield

ChatGPT (OpenAI, 2023)

- InstructGPT models
  - Trained with humans in the loop
  - Deployed as default language models on OpenAI's API
  - Better at following user intentions than GPT-3
  - More truthful and less toxic
  - Uses "alignment" technique
    - Reinforcement learning from human feedback (RLHF)

Reinforcement Learning from Human Feedback (RLHF)

- Customers submit prompts to the API
- Human labelers provide demonstrations of desired model behaviours
  - This data is used to fine-tune GPT-3
- Human labelers then rank different model outputs
  - This data is use to train a **reward model** to predict which output labellers would prefer
- GPT-3 has an additional input known as its "policy"
  - this is fine-tuned to maximise the reward using proximal policy optimization (PPO) (Schulman et al. 2017)



Training ChatGPT (Ouyang et al. 2022)

Step 1

Collect demonstration data, and train a supervised policy.

A prompt is sampled from our prompt dataset.

A labeler demonstrates the desired output behavior.

This data is used to fine-tune GPT-3 with supervised learning.

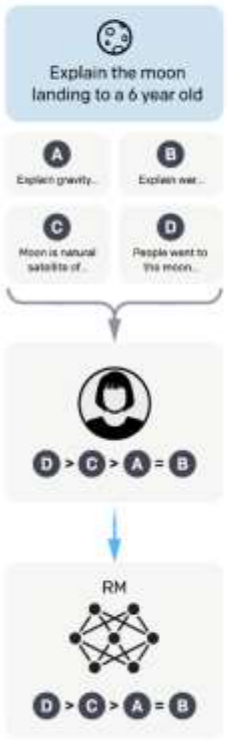
Step 2

Collect comparison data, and train a reward model.

A prompt and several model outputs are sampled.

A labeler ranks the outputs from best to worst.

This data is used to train our reward model.



Step 3

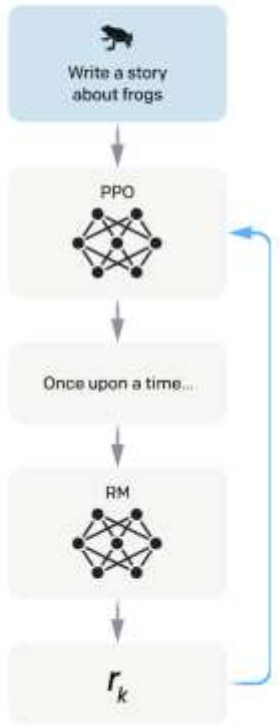
Optimize a policy against the reward model using reinforcement learning.

A new prompt is sampled from the dataset.

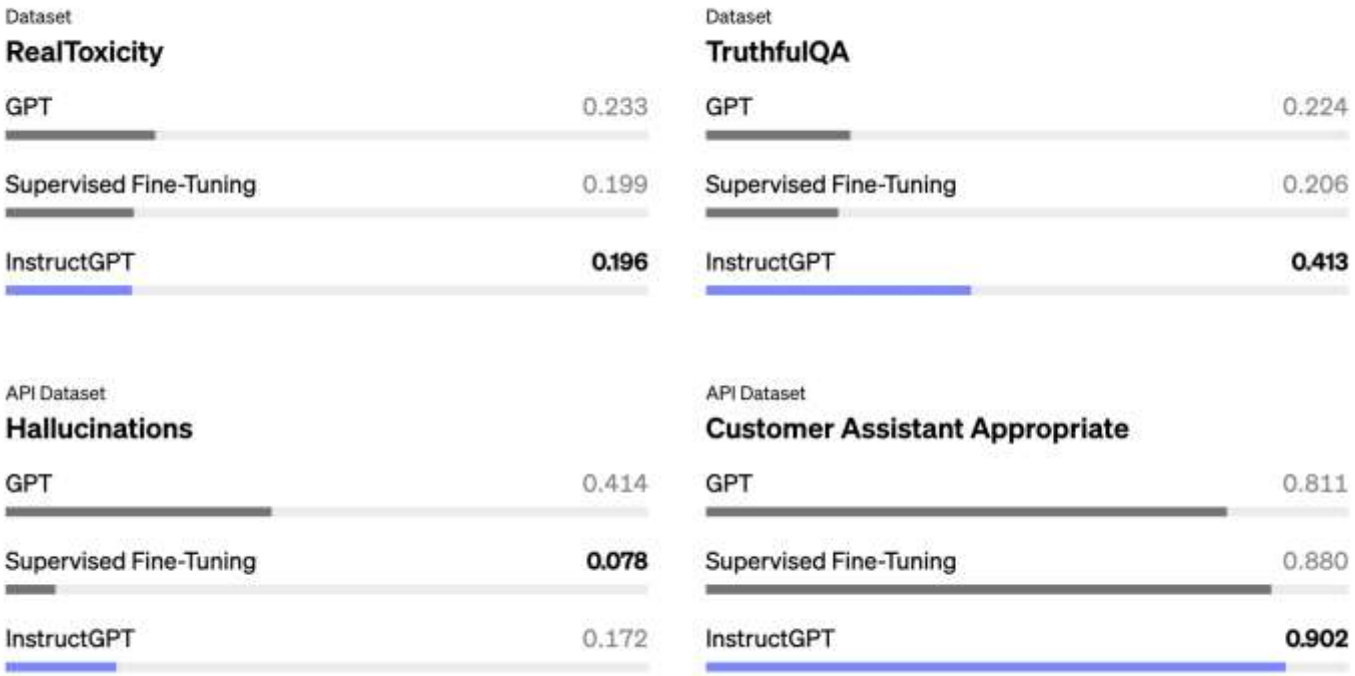
The policy generates an output.

The reward model calculates a reward for the output.

The reward is used to update the policy using PPO.



Does it work?



According to OpenAI

- Less toxic
- More truthful
- Less hallucinations
- More appropriate

Yes, but …?

Weaknesses

ChatGPT is a pre-trained large language model

- It doesn't know or understand anything
- It doesn't look things up or carry out inferences
- Words which mean "similar" things are easy for it to mix up especially numbers and dates
- It has been trained on a very large corpus which is biased towards
  - a certain time period, particular geographic locations, culture and ways of thinking
- Fine-tuning process could be subverted

Using Generative Models

- Nearly all NLP applications can be posed as a prompt to a generative model:
  - Translate the following text from English to French: The cat sat on the mat
  - Summarise the following information in 3 sentences: …..
  - Who played Hans Solo in Star Wars?
  - What is the sentiment of the following review:
- **Encode** the prompt
- **Decode** and generate a response

Document – level MT (Wang et al. 2023)

Figure 1: An example of translating a document-level text from English to Chinese using GPT-4 (Date: 2023.03.17). We highlight the discourse phenomena using figures and lines, which are invisible to GPT-4.

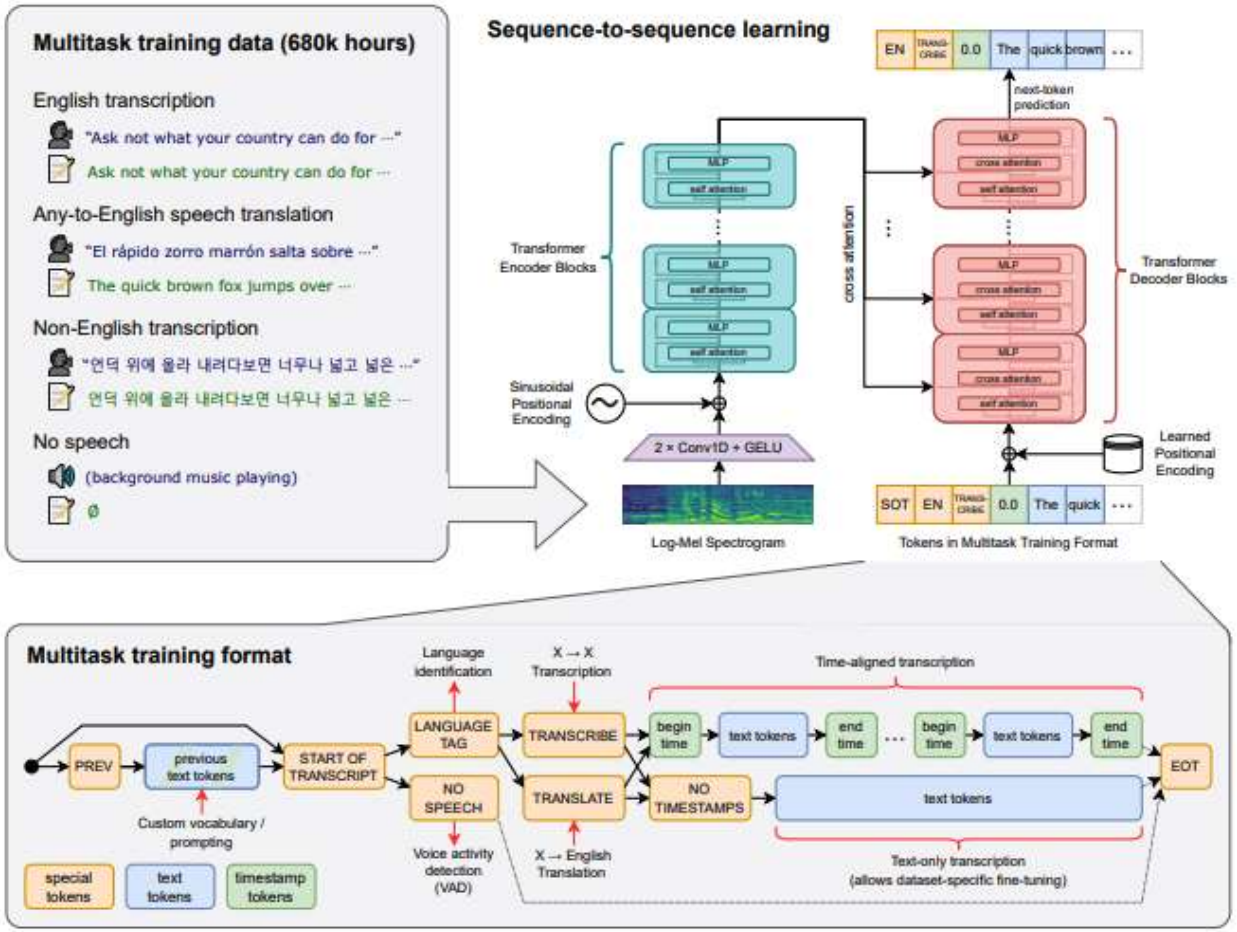
Model	Automatic (d-BLEU)					Human (General/Discourse)				
	News	Social	Fiction	Q&A	Ave.	News	Social	Fiction	Q&A	Ave.
Google	27.7	35.4	16.0	12.0	22.8	1.9/2.0	1.2/1.3	2.1/2.4	1.5/1.5	1.7/1.8
DeepL	**30.3**	33.4	16.1	11.9	22.9	2.2/2.2	1.3/1.1	2.4/2.6	1.6/1.5	1.9/1.9
Tencent	29.3	**38.8**	**20.7**	15.0	**26.0**	2.3/2.2	1.5/1.5	2.6/2.8	1.8/1.7	2.1/2.1
GPT-3.5	29.1	35.5	17.4	17.4	24.9	2.8/2.8	2.5/2.7	**2.8**/**2.9**	2.9/2.9	2.8/2.8
GPT-4	29.7	34.4	18.8	**19.0**	25.5	**3.3**/**3.4**	**2.9**/**2.9**	2.6/2.8	**3.1**/**3.2**	**3.0**/**3.1**

- Introduces evaluation metric which are “discourse aware”
- Shows the superiority of LLMs over other advanced MT systems
- Demonstrates the need for context-aware prompts
  - ChatGPT can translate sentence-by-sentence but remember the discourse-level context via the chat box
  - But best results generally by providing the whole document

Whisper (Radford et al. 2023)

- Trained to transcribe and translate audio
- Dataset constructed from audio paired with human-generated transcripts on the Internet
- Minimal pre-processing
- Audio language detection
- Multi-task format
- Zero-shot evaluation
- Available in Huggingface

Overview of Whisper



X → English	High	Mid	Low	All
XMEF-X	34.2	20.2	5.9	14.7
XLS-R (2B)	36.1	27.7	15.1	22.1
mSLAM-CTC (2B)	37.8	29.6	18.5	24.8
Maestro	**38.2**	31.3	18.4	25.2
Zero-Shot Whisper	36.2	**32.6**	**25.2**	**29.1**

Table 3. **X→en Speech translation performance.** Zero-shot Whisper outperforms existing models on CoVoST2 in the overall, medium, and low resource settings but still moderately underperforms on high-resource languages compared to prior directly supervised work.

Machine Translation: remaining challenges

- Long documents can still pose problems
- Stylised translation
- Interactivity
- Multi-modal
- Evaluation – quality and interpretability / explanations

Benchmarking LLMs for News Summarization (Zhang et al. 2024)

Article: https://www.dailymail.co.uk/femail/article-3052124

Reference summary:

Rare tote is so exclusive that it can't even be brought in store. Experts say it's an extremely good investment as only increases in value. Kim Kardashian and Victoria Beckham are both fans of Hermes.

0-shot GPT-3 Davinci summary:

The most expensive handbag in the world is a pink Hermes Birkin bag, which has been put on sale for £140,000. The bag is so exclusive that it can't even be brought in store and experts say it's an extremely good investment. With the same amount of money that the handbag costs, you could splash out on an Aston Martin DB9 Volante.

What is the main idea of the article?

0-shot Instruct Davinci summary:

The article discusses a pink Hermes Birkin bag that is on sale for £140,000. The bag is made-to-measure and embellished with rubies. The article also discusses how the bag is a collector's item and an investment.

Freelance writer summary:

Hermes is a brand rich in history and value, so much so that experts are saying their new £140,000 handbag is an extremely good investment. The bag, studded in rubies, is only the latest in Hermes Birkin's long line of highly expensive handbags for highly stylish women, which include celebrities and royalty.

Figure 2: Examples summaries generated by GPT-3 models (Section 3) or written by freelance writers (Section 4) of an article from the CNN/DM dataset. We find that the instruction-tuned GPT-3 model can generate a much better summary compared to the non-instruction-tuned variant. The reference summary from CNN/DM is not coherent whereas the freelance writer summary is both coherent and relevant.

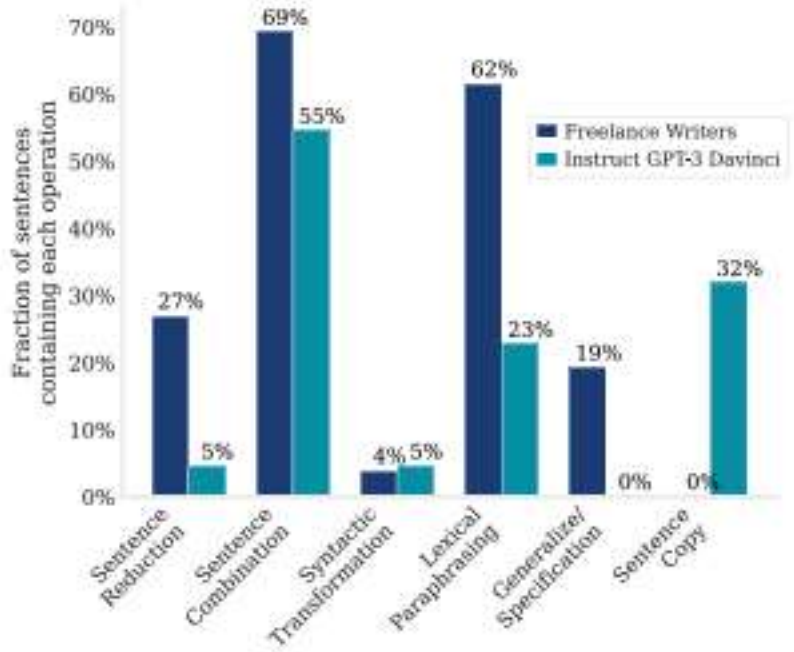


Figure 4: Distributions of cut and paste operations in the summaries written by freelance writers and by Instruct Davinci. By comparison, human-written summaries contain more lexical paraphrasing and sentence reduction whereas the Instruct Davinci model has more direct copying from the article.

Retrieval Augmented Generation (Gao et al. 2023)

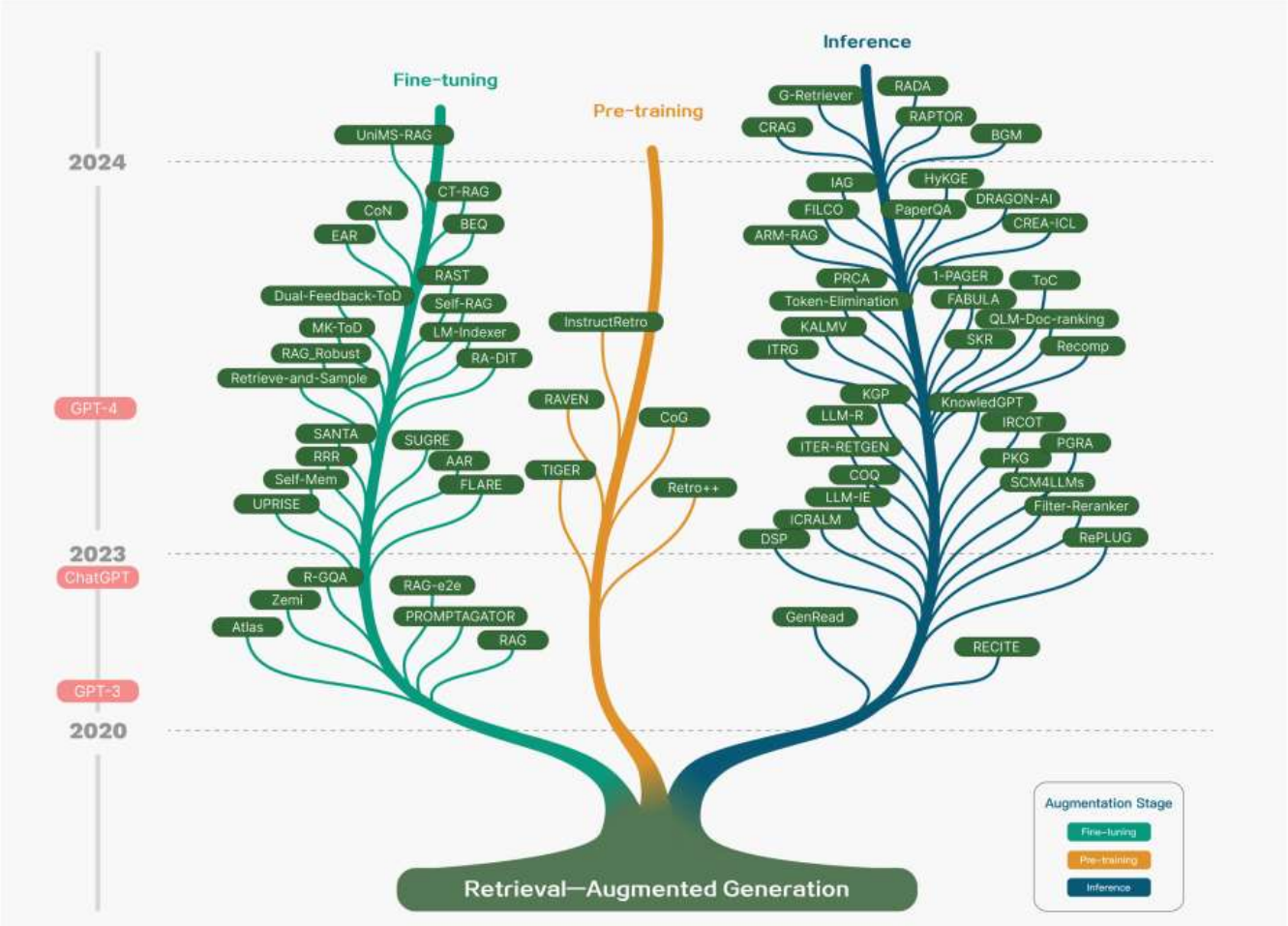


Fig. 1. Technology tree of RAG research. The stages of involving RAG mainly include pre-training, fine-tuning, and inference. With the emergence of LLMs,

Retrieve-Read Framework for RAG

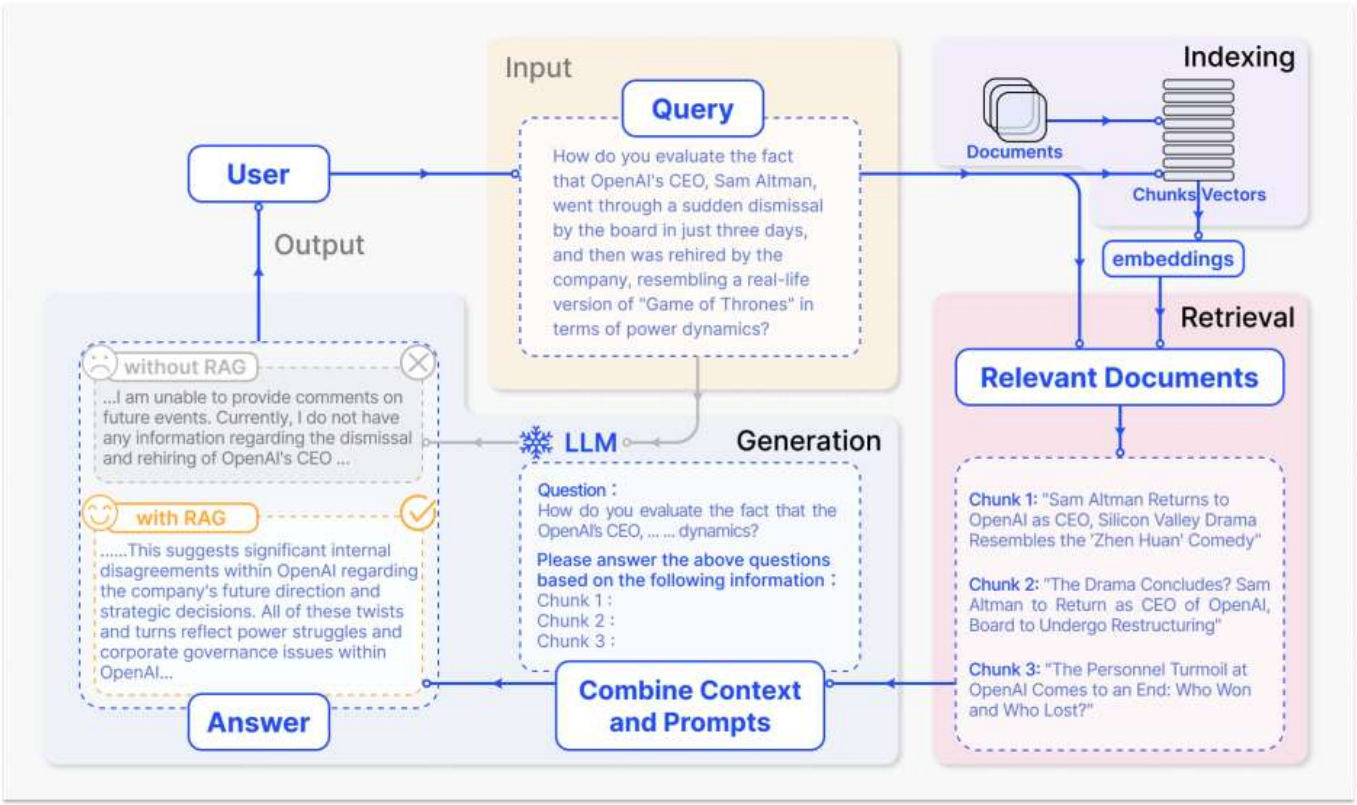


Fig. 2. A representative instance of the RAG process applied to question answering. It mainly consists of 3 steps. 1) Indexing. Documents are split into chunks, encoded into vectors, and stored in a vector database. 2) Retrieval. Retrieve the Top k chunks most relevant to the question based on semantic similarity. 3) Generation. Input the original question and the retrieved chunks together into LLM to generate the final answer.

More Advanced RAG

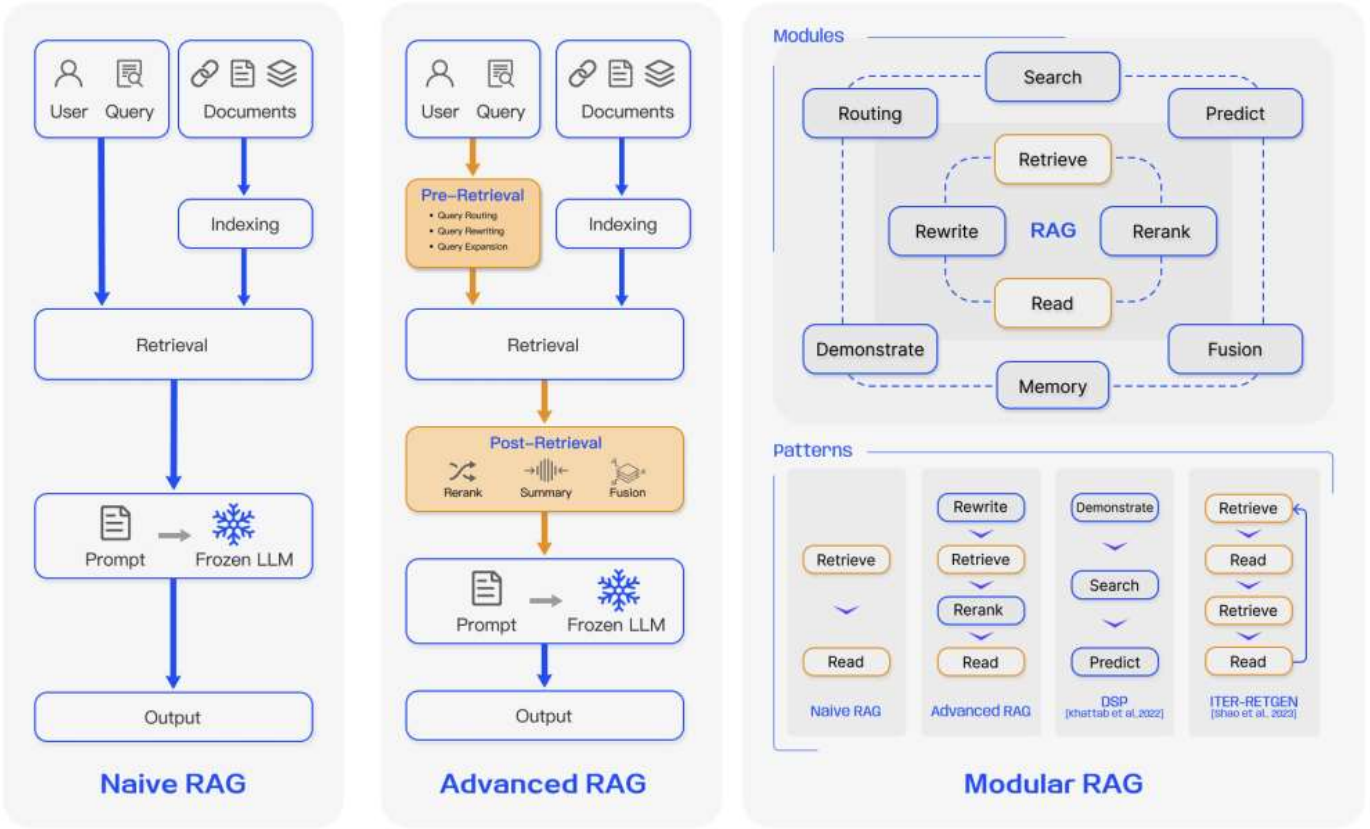


Fig. 3. Comparison between the three paradigms of RAG. (Left) Naive RAG mainly consists of three parts: indexing, retrieval and generation. (Middle) Advanced RAG proposes multiple optimization strategies around pre-retrieval and post-retrieval, with a process similar to the Naive RAG, still following a chain-like structure. (Right) Modular RAG inherits and develops from the previous paradigm, showcasing greater flexibility overall. This is evident in the introduction of multiple specific functional modules and the replacement of existing modules. The overall process is not limited to sequential retrieval and generation; it includes methods such as iterative and adaptive retrieval.

Ethical considerations

- What else do you need to consider before using an advanced LLM system to generate text for you?

Accountability

By Maria Yagoda 23rd February 2024

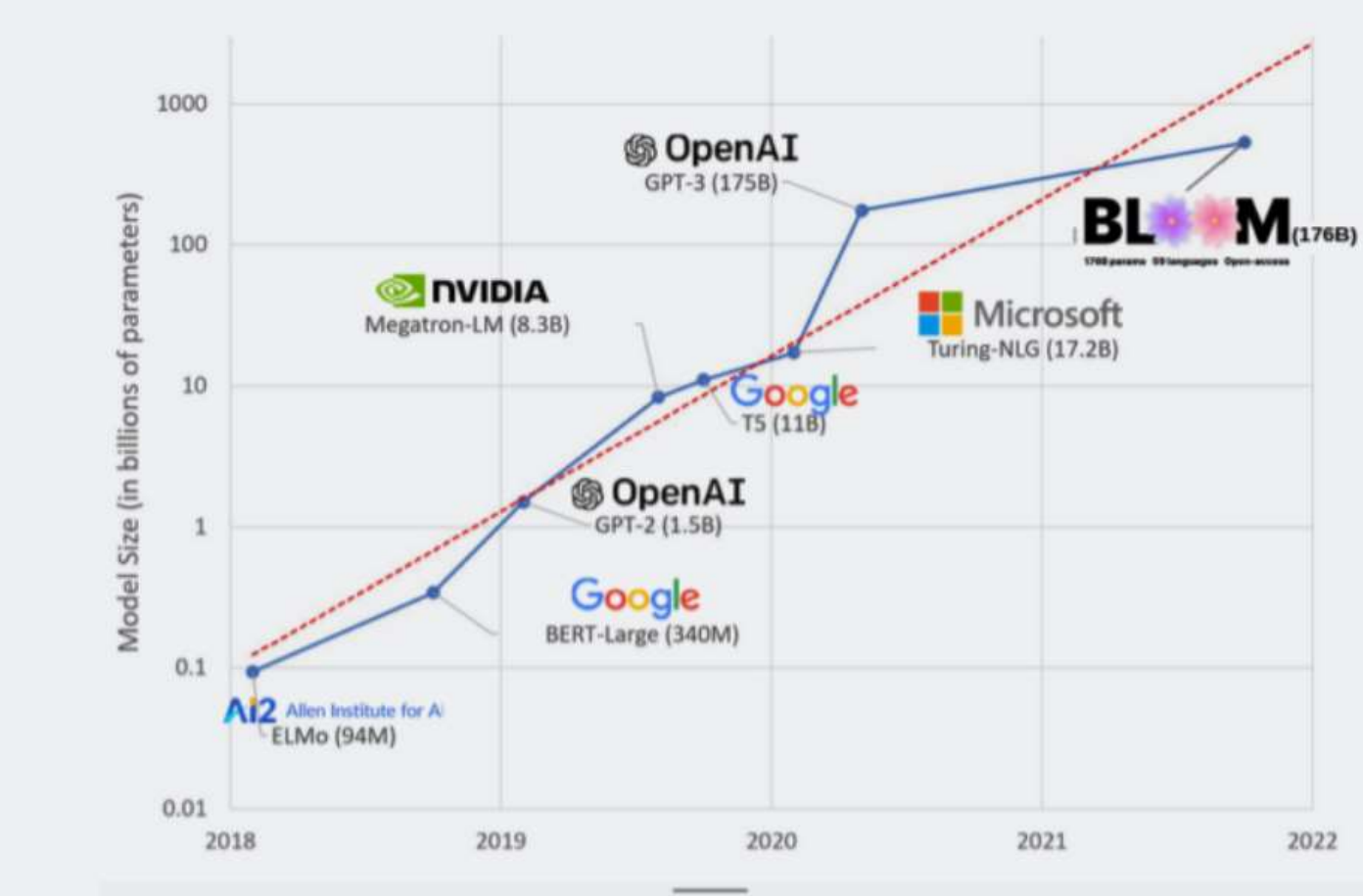
When Air Canada's chatbot gave incorrect information to a traveller, the airline argued its chatbot is "responsible for its own actions".

Artificial intelligence **is having a growing impact on the way we travel,** and a remarkable new case shows what AI-powered chatbots can get wrong – and who should pay. In 2022, Air Canada's chatbot **promised a discount** that wasn't available to passenger Jake Moffatt, who was assured that he could book a full-fare flight for his grandmother's funeral and then apply for a bereavement fare after the fact.

According to a civil-resolutions tribunal **decision** last Wednesday, when Moffatt applied for the discount, the airline said the chatbot had been wrong – the request needed to be submitted before the flight – and it wouldn't offer the discount. Instead, the airline said the chatbot was a "separate legal entity that is responsible for its own actions". Air Canada argued that Moffatt should have gone to the link provided by the chatbot, where he would have seen the correct policy.

The British Columbia Civil Resolution Tribunal rejected that argument, ruling that Air Canada had to pay Moffatt $812.02 (£642.64) in damages and tribunal fees. "It should be obvious to Air Canada that it is responsible for all the information on its website," read tribunal member Christopher Rivers' written response. "It makes no difference whether the information comes from a static page or a chatbot." The BBC reached out to Air Canada for additional comment and will update this article if and when we receive a response.

Model Growth



Enlarge / Model size growth over the years.

Source: https://arstechnica.com/gadgets/2023/04/generative-ai-is-cool-but-lets-not-forget-its-human-and-environmental-costs/

Environmental Impact of LLMs

Model name	Number of parameters	Datacenter PUE	Carbon intensity of grid used	Power consumption	CO_2eq emissions	CO_2eq emissions × PUE
GPT-3	175B	1.1	429 gCO_2eq/kWh	1,287 MWh	*502 tonnes*	552 tonnes
Gopher	280B	1.08	330 gCO_2eq/kWh	*1,066 MWh*	*352 tonnes*	380 tonnes
OPT	175B	*1.09* [2]	*231 gCO_2eq/kWh*	*324 MWh*	70 tonnes	*76.3 tonnes* [3]
BLOOM	176B	1.2	57 gCO_2eq/kWh	433 MWh	25 tonnes	30 tonnes

Table 4: Comparison of carbon emissions between BLOOM and similar LLMs. Numbers in *italics* have been inferred based on data provided in the papers describing the models.

Sasha Luccioni, et al.

- high levels of energy use and CO2 emissions
- Training GPT-3 and Llama 2 required around 1.3 GWH → 552 tonnes of CO2 → 2 or 3 full Boeing 767s flying round-trip from New York to San Francisco
- Electricity use of ChatGPT in inference likely surpasses that of training within weeks or days (inference counts for 90% of AI workloads) → more like using a Boeing 767 to transport a single passenger at a time

What should we do?

Part 2 : Revision

Revision questions

1. Name and give examples of different semantic relationships which can hold between *distributionally similar* words.
2. What is Zipf's Law and why is it a problem?
3. What are the main similarities and differences between word2vec and GLoVE?
4. Explain how to use a trigram model to compute the probability of a sentence.
5. What is perplexity?
6. What advantage do LSTMs have over vanilla RNNs in language modelling?
7. How and why might you combine a character-based network with a word-based network in language modelling?
8. For what types of problems are CRFs typically used?
9. What's the difference between a generative statistical classification model and a discriminative statistical classification model?
10. When and why might it be better to use F1 rather than accuracy as an evaluation metric?

More questions

1. In a multi-class scenario, how would you calculate micro-average F1 and macro-average F1. Which is better and why?
2. Describe 2 different ways word embeddings might be combined to make sentence embeddings
3. Give an example of structural differences between languages.
4. Outline 2 different methods for evaluating machine translation systems.
5. How might an encoder-decoder network be used for MT?
6. What is subword tokenization?
7. In an attention head, what are the 3 different vectors which are created? How are they created and how are they used?
8. What is the input representation used by BERT?
9. What is the difference between masked language modelling and autoregressive language modelling?
10. What is transfer learning? Explain with reference to BERT

AdvNLE Seminar 8

Pre-training Large Language Models

Dr Julie Weeds, Spring 2024

Warm-up

- what **applications** can you think of where it might be useful to have a measure of how similar two sentences or documents are?
  - what are the different ways in which two sentences might be similar?
  - how does this affect the applications?

Paraphrase identification and semantic matching

- text simplification
- automated marking
- question answering
- text summarization
- information retrieval
- recommendation systems
- document clustering

See: Yang et al. (2020) for a discussion of 4 different categories of semantic matching problems in NLP (https://arxiv.org/pdf/2004.12297.pdf)

Semantic matching in … Question Answering

- "Who became the head of the UK government in 1951?"
- "Who was elected as British prime minister in 1951?"

- If we know the answer to one of these questions …
- we probably know the answer to the other one too

Sir Winston Leonard Spencer Churchill, KG, OM, CH, TD, DL, FRS, RA was a British politician, statesman, army officer, and writer. He was Prime Minister of the United Kingdom from 1940 to 1945, during the Second World War, and again from 1951 to 1955.

Semantic matching in … Automated Marking

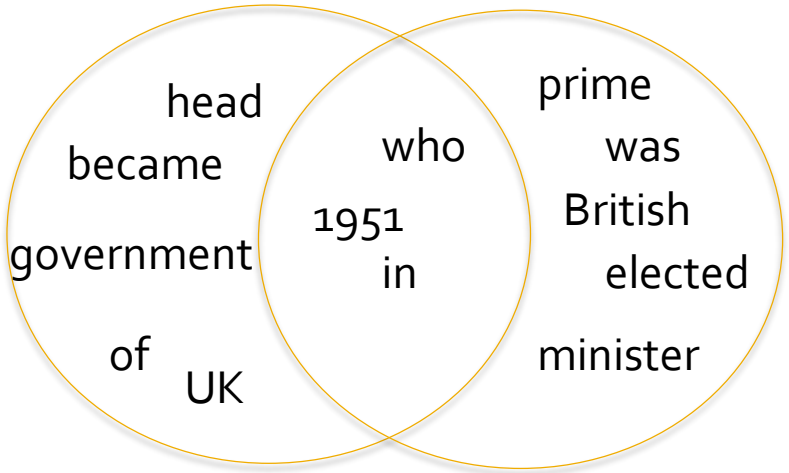
How are igneous rocks formed?

Igneous rocks form when magma (molten rock) cools and crystallizes, either at volcanoes on the surface of the Earth or while the melted rock is still inside the crust.

Igneous rocks (from the Latin word for fire) form when hot, molten rock crystallizes and solidifies.

Simple text matching

- word overlap
  - e.g., using Jaccard's coefficient



$$jacc_{sim(A,B)} = \frac{|A \cap B|}{|A \cup B|} = \frac{3}{13}$$

- Bag-of-words representations of sentences / documents
  - weight words using TF-IDF
  - cosine similarity between vectors
- TF-IDF = "term frequency, inverse document frequency"
- gives more weight to:
  - higher frequency terms
  - more discriminating terms

Beyond simple text matching

- Why do we need more powerful methods than simple text matching?

Previously

- Distributional models of word meaning
  - how similar are two words based on how they are used in text?
- Language models
  - how likely is a sequence of words in a language?
- Neural language models
- Tasks
  - Sequence labelling
  - Sequence classification
  - Sequence generation

This week

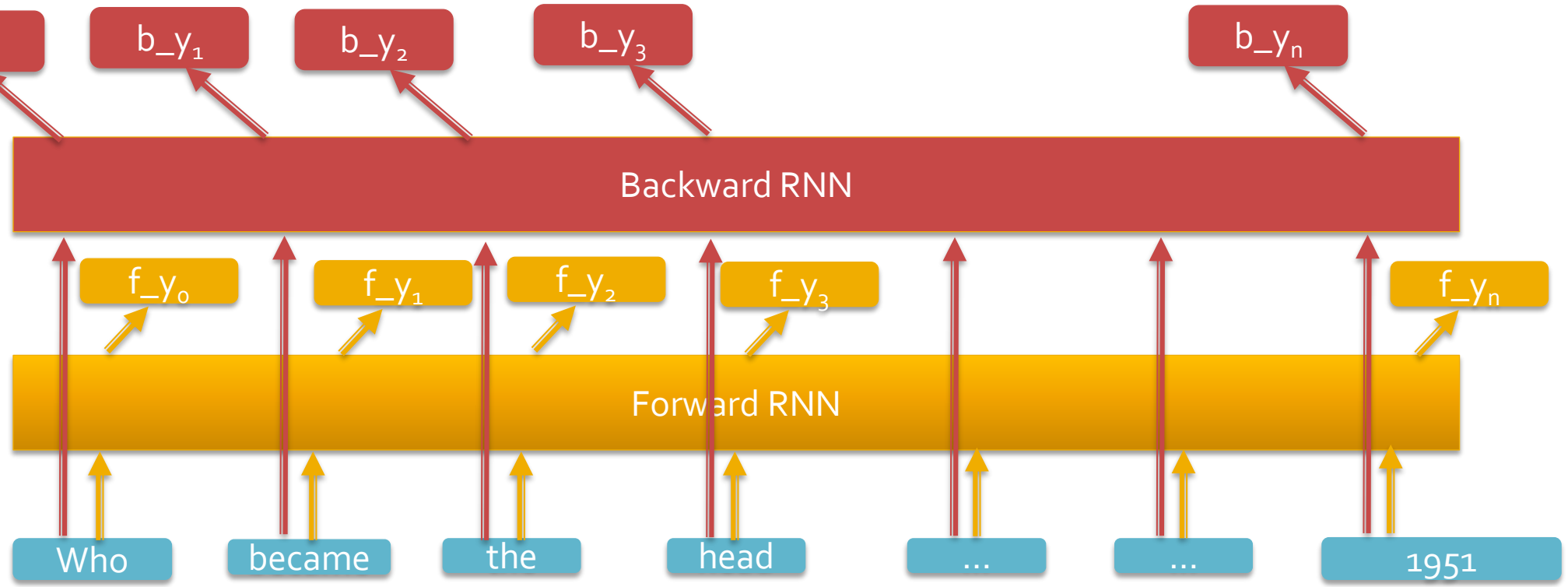
- Large language models
  - Contextualised word embeddings (ELMo)
  - Transformer architecture and attention
  - BERT (bidirectional encoder representation from transformers)
  - Pre-training:
    - Masked language modelling
    - Autoregressive language modelling
    - Next sentence prediction
  - **Sentence representations / paraphrasing (Seminar)**

Lecture questions

1. Imagine you have a 100M word corpus of news articles with a vocabulary of size 50K. Explain the difference between static word embeddings and contextualized word embeddings derived from this corpus.
2. Why are transformers now generally preferred to LSTMs in the NLP community?
3. In the sentence, "A few faint stars glimmered in the sky.", what words might need to pay attention to other words in the sentence, in order for a good contextualized word representation to be derived?
4. Explain how the output of an attention head is derived (for one of the words in the sentence above)
5. Will the encoder of a transformer produce the same representation of the sentences, "The dog bit the boy." and the "The boy bit the dog." Why/ why not?

Contextualised word embeddings

- Represent each word based on its context
- e.g., concatenation of [b_y_t,f_y_t]
- compose contextualised word embeddings as before



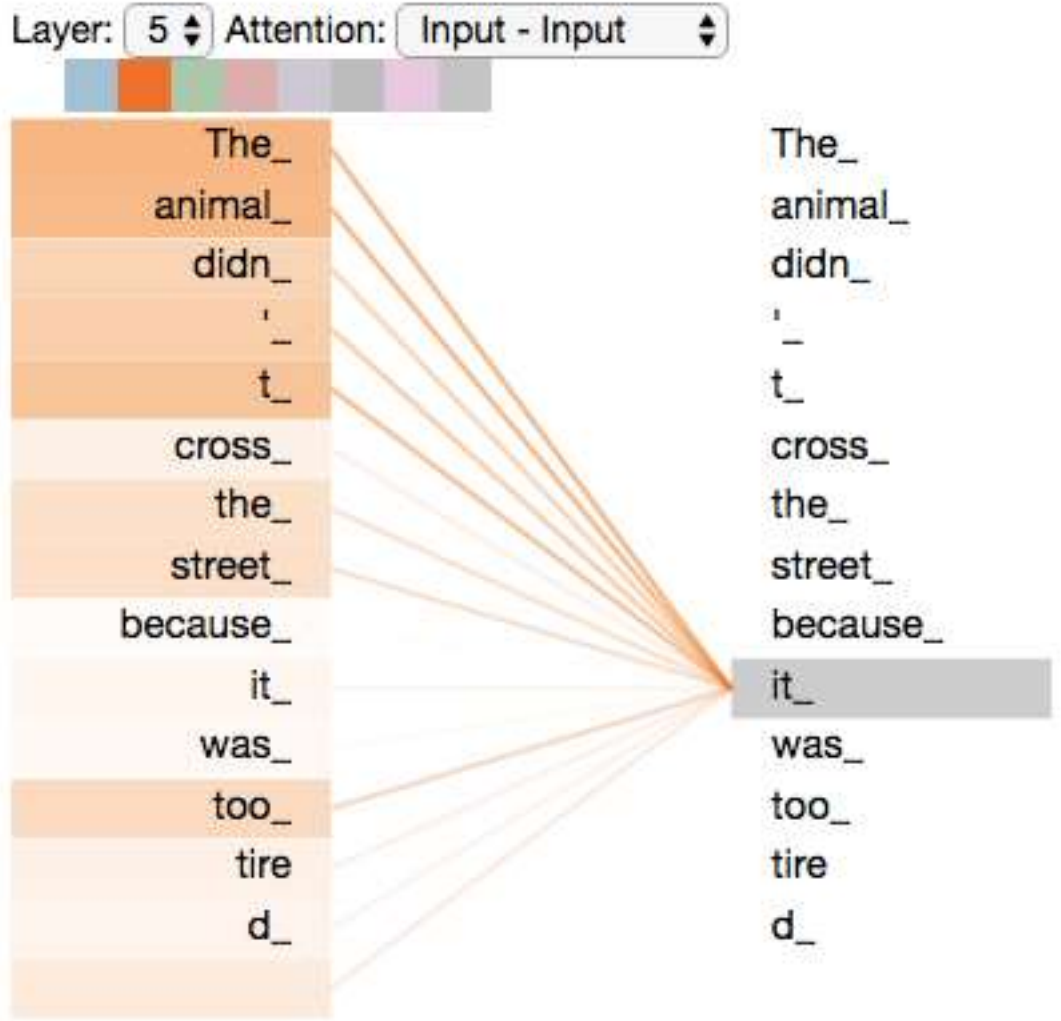
embedding layer

Transformers (Vashwani et al. 2017)

- Sequence transduction model using stacked self-attention
  - no convolutions or recurrence
  - easier to parallelize than RNNs
  - faster to train than RNNs
  - captures more long-range dependencies than CNNs with fewer parameters
- What's a sequence transduction model?
- What's stacked self-attention?

Self-Attention (High level)

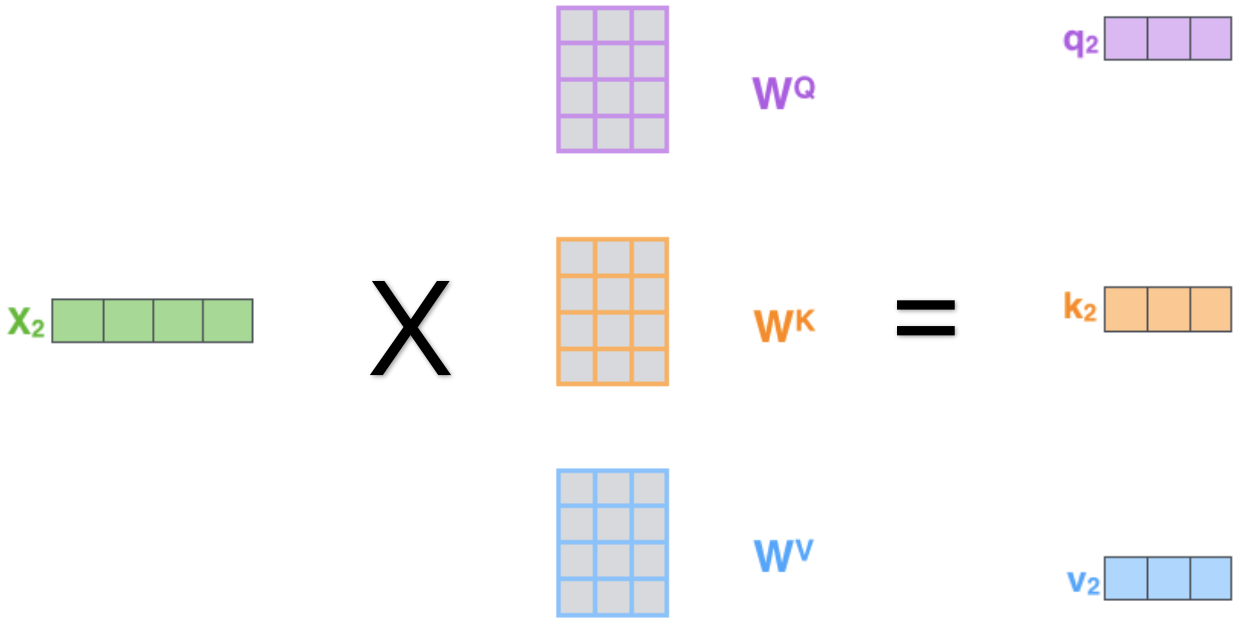
- What to pay attention to when encoding or decoding
  - **Attention**: in previous layer
  - **Self-attention**: in the same layer
- When encoding the word ***it*** in the sentence "*The animal didn't cross the street because* ***it*** *was too tired*", self-attention should allow the model to associate ***it*** with ***animal***



Example from http://jalammar.github.io/illustrated-transformer/

Self-attention - Step 1

- Step 1: from the encoder's input vector (e.g., word embedding), create 3 vectors: **Q**uery, **K**ey and **V**alue
- Q,K, and V embeddings usually smaller in number of dimensions e.g.,
  - input 512
  - output 64



input embedding

trainable weight matrices

output query, key and value embeddings

Self-attention score

- Step 2: To work out how much the word in position$_i$ (e.g., "*it*") should pay attention to a word in position$_j$ (e.g., "*animal*") words in the sentence, we take the **dot product** of the Q_i (the **query** vector for *it*) with the K_j (the **key** vector for *animal*)

$$SelfAtt(w_i, w_j) = Q_i . K_j$$

- Do this for every word in the sentence with every other word in the sentence

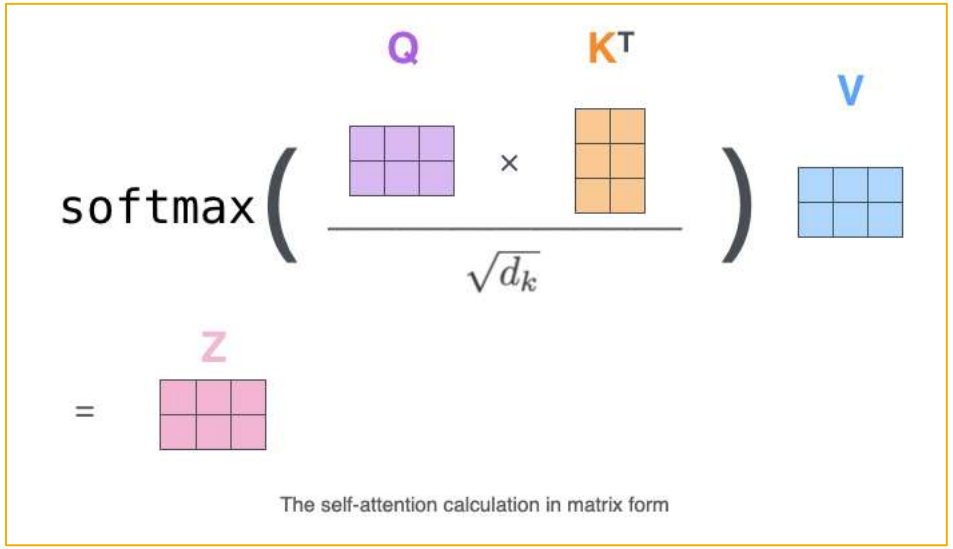
Normalise

- Step 3: Divide all of the scores by 8
  - the square root of the dimension of the key vectors which is 64 here
  - this is the default and seems to lead to more stable gradients
- Step 4: Apply a softmax to the scores
  - this results in them all being positive
  - and summing to 1
  - so can be thought of a as a probability / proportion – "what proportion of my attention should I pay to word j?"

Weight sum of the value vectors

- Step 5: For each word w_i, multiply the value vector (V_j) of each other word w_j by its softmax score with w_i
- Step 6: Sum to produce output embedding (at this layer) for w_i

$$Z_i = \sum_j softmax\_score(w_i, w_j) \times V_j$$



Self-attention summary

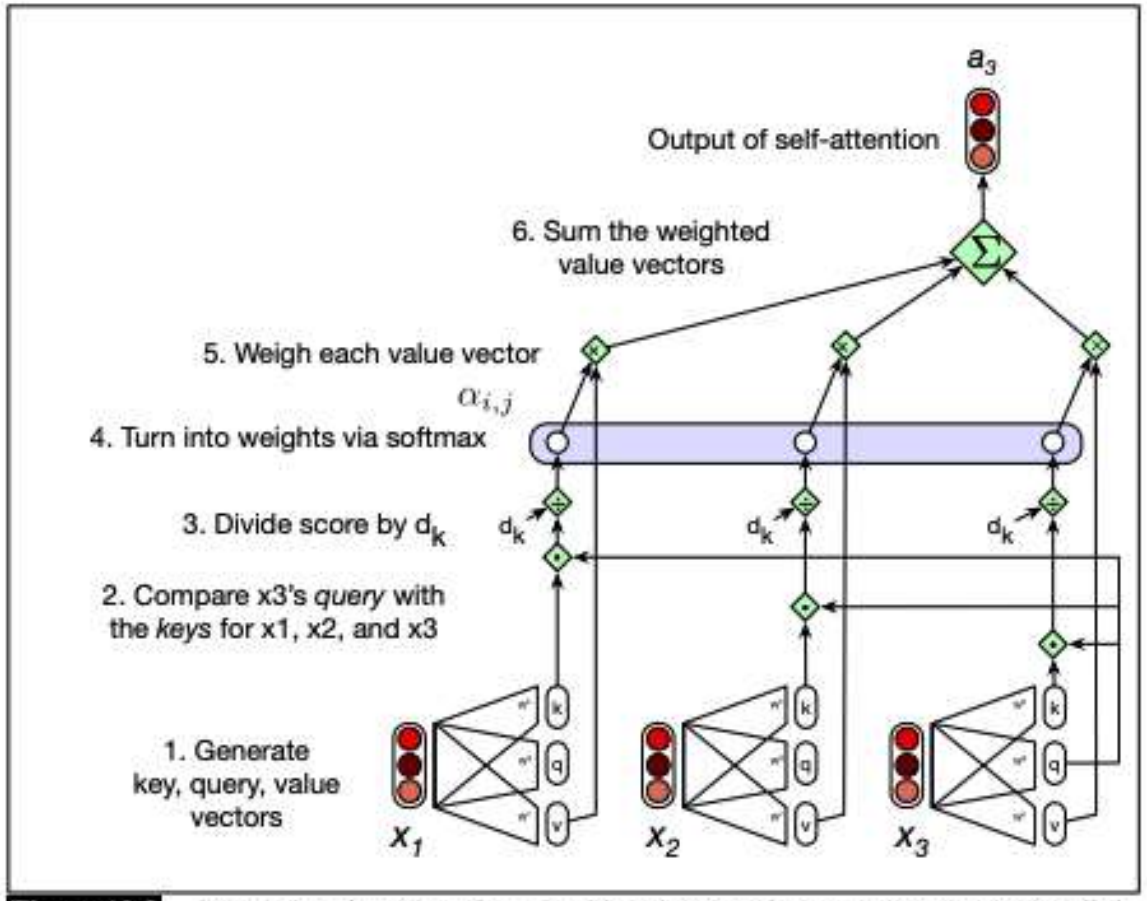
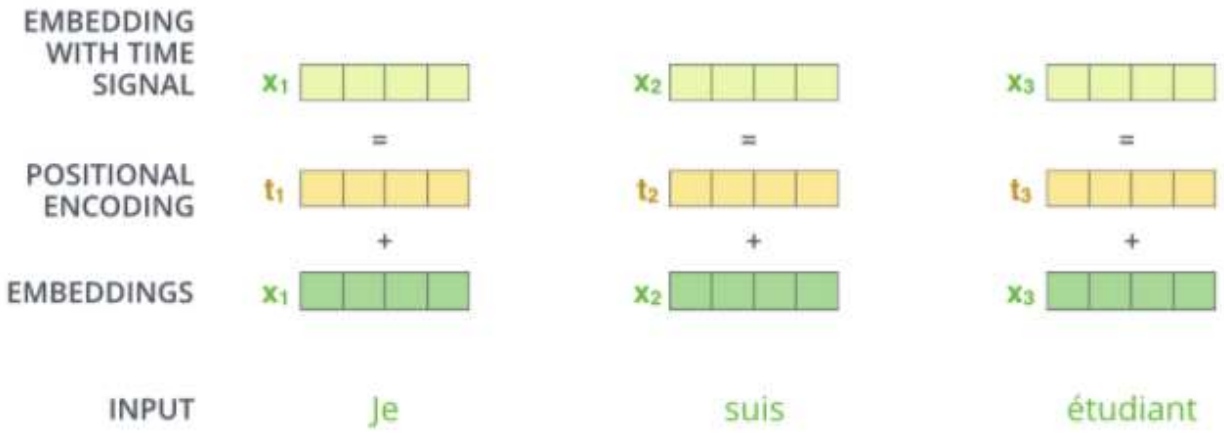


Figure 10.3 Calculating the value of $\mathbf{a}_3$, the third element of a sequence using causal (left-to-right) self-attention.

- Image taken from Chapter 10, Jurafsky and Martin
- This is actually for uni-directional self-attention
- However, it would be the same for the a_3 in the bidirectional case assuming an input sequence which is 3 tokens long!

Positional Encoding

- How does the model account for word order?
- Adds a positional encoding vector to each input embedding
- Positional encoding usually follows a fixed pattern
  - enables the model to determine the position of each word and distance between them

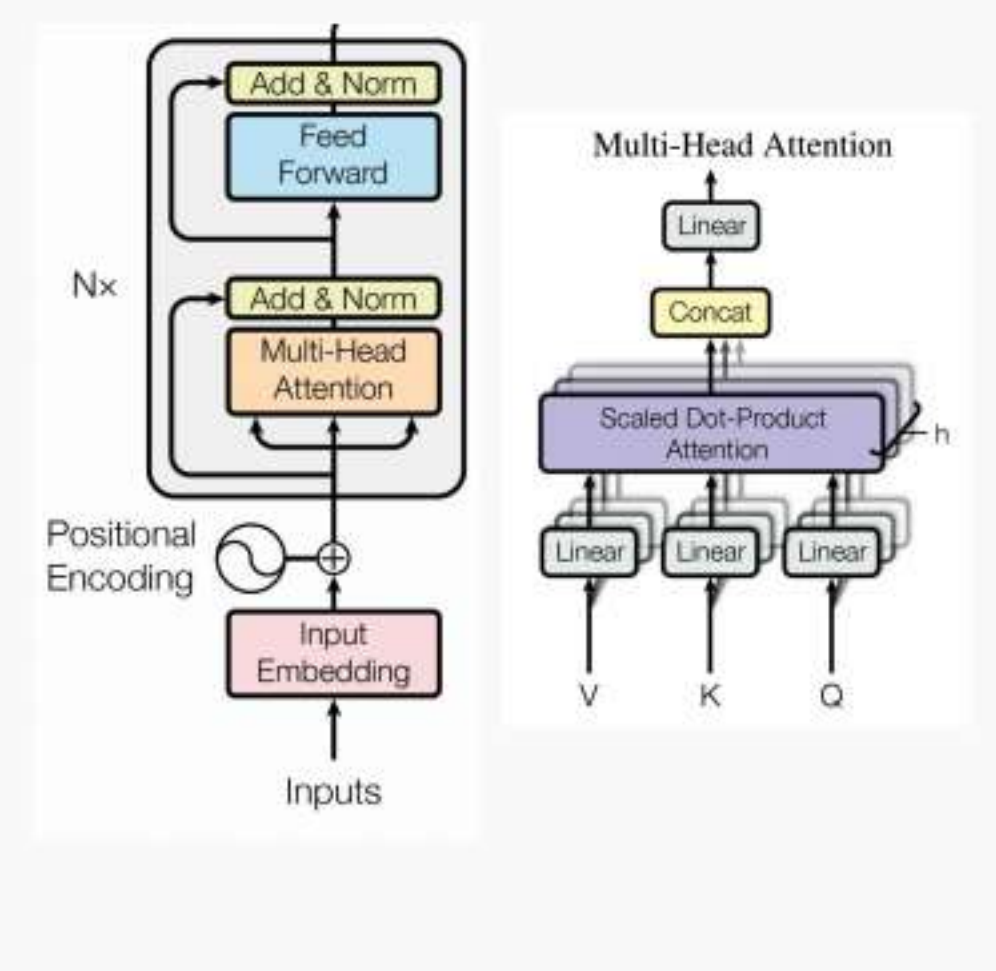


BERT (Devlin et al. 2019)

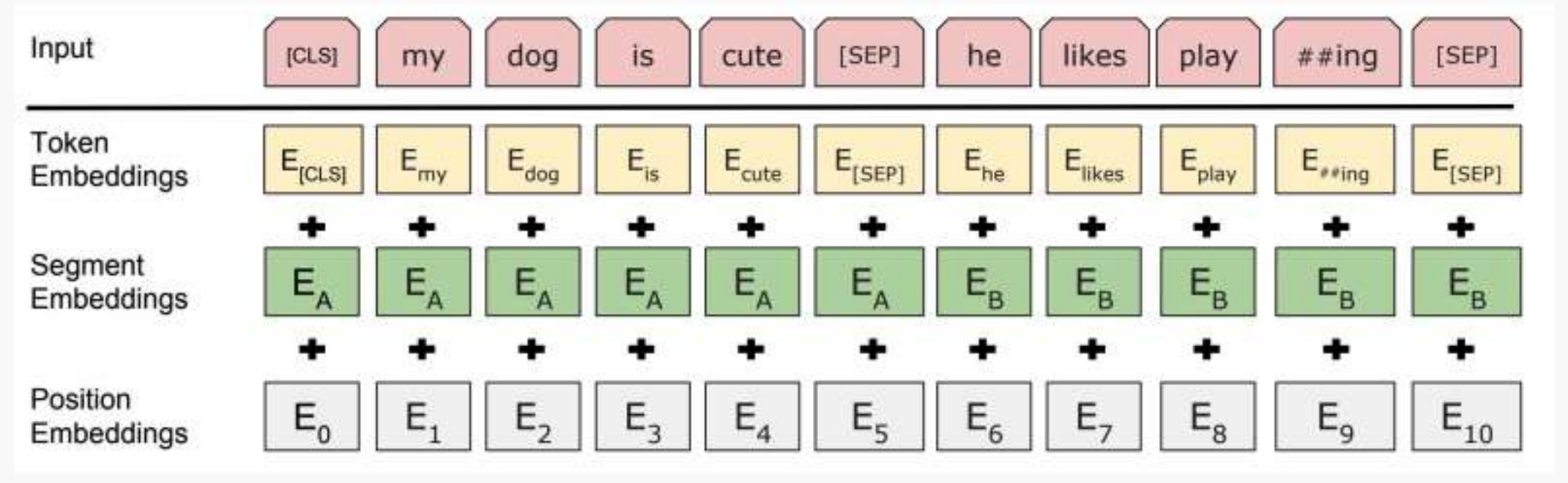
- **B**idirectional **E**ncoder **R**epresentations from **T**ransformers
  - just uses the encoder portion of the transformer architecture
  - uses left and right context of a target word to build representation
- Pre-trained on general language modelling tasks
  - masked language modelling task
  - next sentence prediction
- fine-tuned on task-specific training data
- Pre-trained models available from Google:
  - https://github.com/google-research/bert
- And from huggingface!

Encoder Representation

- Multi-headed self-attention
  - models context
- Feed-forward layers
  - non-linear hierarchical features
- Layer norm and residuals
  - makes training deep networks healthy
- Positional embeddings
  - learn relative positioning

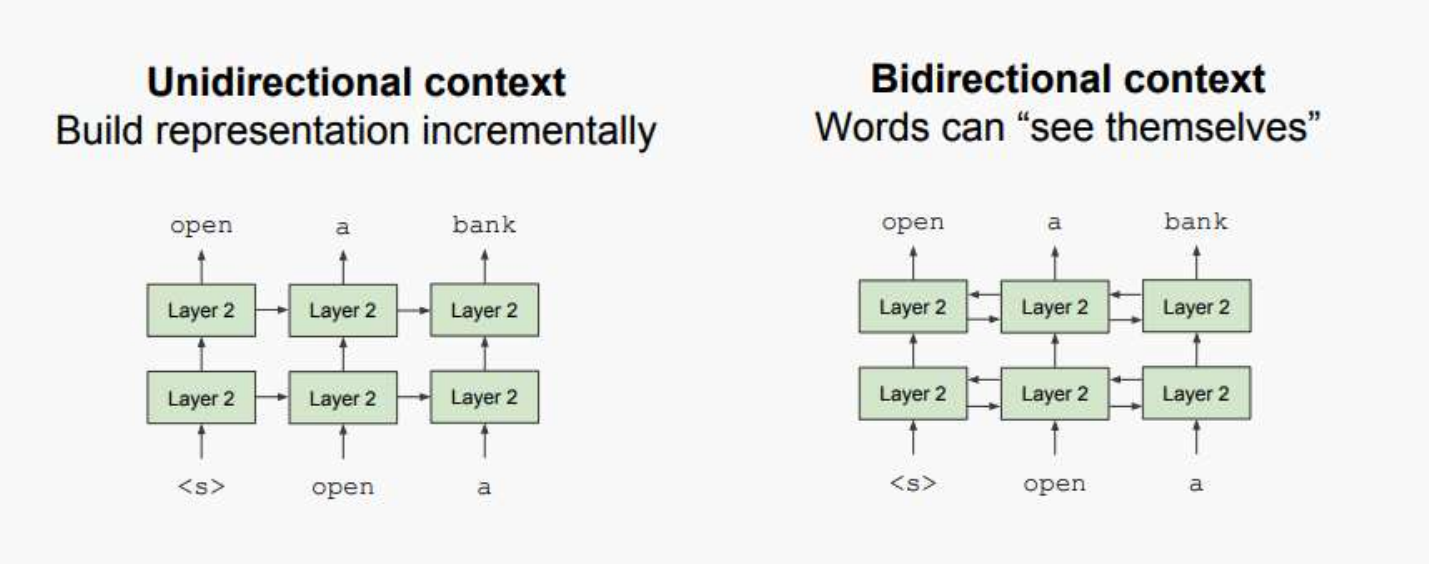


Input Representation



- Each token is sum of three embeddings
- 30,000 WordPiece vocabulary → morphology → better representations for rare words
- sentences are separated by a special "SEP" token
- a special "CLS" token is added at the beginning of the chunk
  - often used in classification

Unidirectional vs Bidirectional models

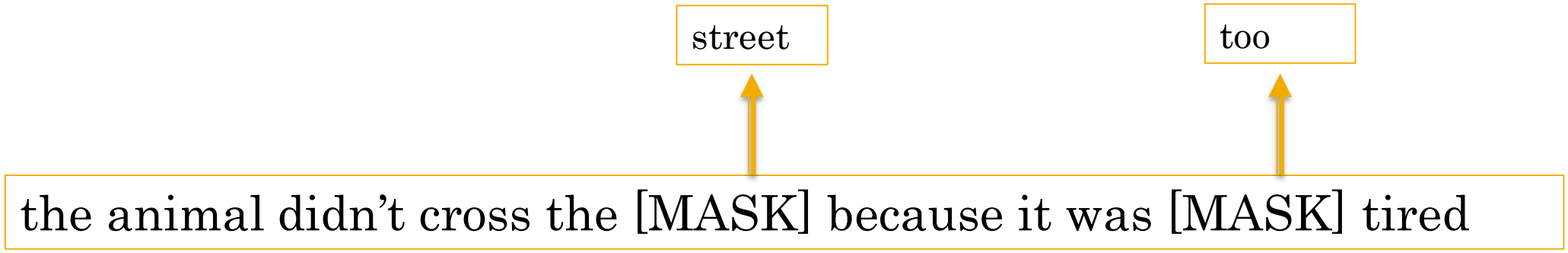


from https://nlp.stanford.edu/seminar/details/jdevlin.pdf

- Most language models trained to predict the next word in sequence
- Known as **auto-regressive** or **unidirectional**
- BERT uses the whole of the sentence to predict missing or masked word
- masked language model is **bidirectional**

Masked Language Model

- Mask out k% of the input words, and predict the masked words
  - if k is too small → expensive to train
  - if k is too large → not enough context
  - Google use k=15



Next Sentence Prediction

- To learn relationships between sentences, predict whether sentence B is actual sentence that follows sentence A or is a random sentence

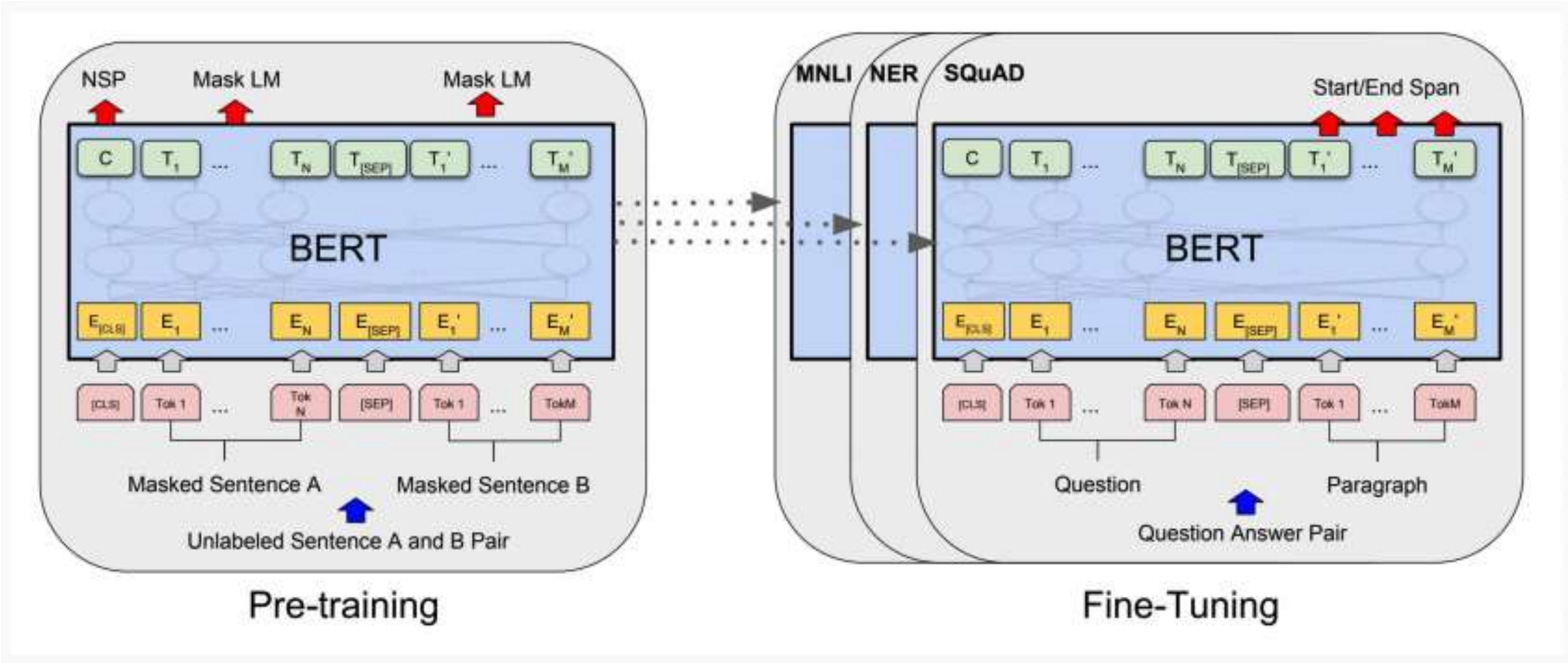
sentence A = The man went to the store
sentence B = He bought a gallon of milk
Label = IsNextSentence

sentence A = The man went to the store
sentence B = Penguins are flightless
Label = NotNextSentence

Pre-trained model details

- Data: Wikipedia (2.5B words) + BookCorpus (800M words)
- Batch Size: 131,072 words (1024 sequences * 128 length or 256 sequences * 512 length)
- Training Time: 1M steps (~40 epochs)
- Optimizer: AdamW, 1e-4 learning rate, linear decay
- `BERT-Base`: 12-layer, 768-hidden, 12-head
- `BERT-Large`: 24-layer, 1024-hidden, 16-head
- Trained on 4x4 or 8x8 TPU slice for 4 days

Fine-tuning Procedure



Representing sentences

- How can we use BERT to compare pairs of sentences to see if they mean the same or similar things?
- How do we get from a BERT encoding of a sequence of tokens to a representation of a sentence?
- What are the options?

Reimers and Gurevych (2019) present Sentence-BERT (SBERT), a modification of the BERT netowrk using siamese and triplet networks that they claim is able to derive semantically meaningful sentence embeddings. Once you have read the paper, consider the following questions.

1. For a pair regression task, the standard BERT set-up requires pairs of sentences to be presented as input to the encoder network. Why is this set-up unfeasible if you want to find the most similar sentences in a collection?

2. What have other researchers done to overcome this problem with using BERT?

3. What is the SBERT strategy?

4. What do you understand by the term Siamese network structure?

5. What are the different pooling strategies that the authors experiment with? What works best?

6. Outline the 4 evaluation tasks used for semantic textual similarity.

7. Why would Spearman's rank correlation coefficient be better than Pearson's product-moment correlation coefficient when comparing ratings of semantic textual similarity?

8. What are the different objective functions that the authors experiment with? When is each used?

9. How is the SentEval evaluation different to the previous evaluation experiments?

10. What are the main conclusions of the paper? Are you convinced?

Q1

- For a pair regression task, the standard BERT set-up requires pairs of sentences to be presented as input to the encoder network. Why is this set-up unfeasible if you want to find the most similar sentences in a collection?

Q2

- What have other researchers done to overcome this problem with using BERT?

Q3

- What is the SBERT strategy?

Q4

- What do you understand by the term Siamese network structure?

Q5

- What are the different pooling strategies that the authors experiment with? What works best?

Q6

- Outline the 4 evaluation tasks used for semantic textual similarity?

Q7

- Why would Spearman's rank correlation coefficient be better than Pearson's product-moment correlation coefficient when comparing ratings of semantic textual similarity?

Q8

- What are the different objective functions that the authors experiment with? When is each used?

Q9

- How is the SentEval evaluation different to the previous evaluation experiments?

Q10

- What are the main conclusions of the paper? Are you convinced?

Further reading

- Devlin et al. (2019): Pre-training of Deep Bidirectional Transformers for Language Understanding in NAACL 2019, https://www.aclweb.org/anthology/N19-1423/
- Peters et al. (2018): Deep contextualised word representations in Proceedings of NAACL 2018 https://arxiv.org/pdf/1802.05365.pdf
- Peters et al. (2018): Dissecting Contextual Word Embeddings: Architecture and Representation in Proceedings of EMNLP 2018 https://www.aclweb.org/anthology/D18-1179.pdf
- Reimers et al. (2019):
- Vashwani et al. (2017): Attention is all you need in Proceedings of NIPS 2017
- Yang et al. (2020) :

(https://arxiv.org/pdf/2004.12297.pdf)