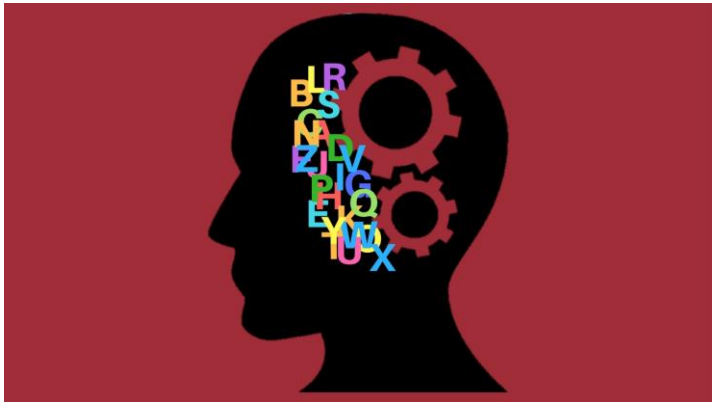
University of Ljubljana, Faculty of Computer and Information Science

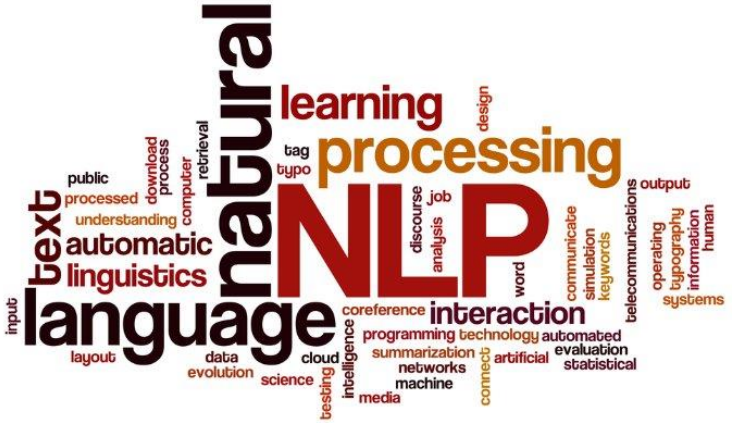
Natural language processing



Prof Dr Marko Robnik-Šikonja

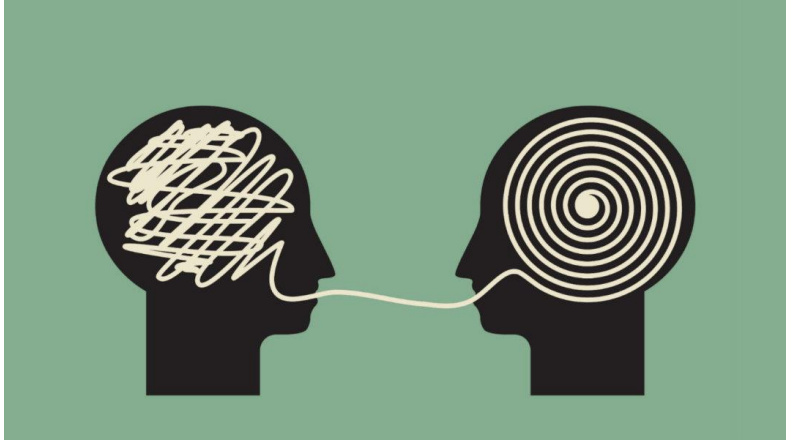
Intelligent Systems, Edition 2024

Topic overview



- understanding language and intelligence
- text preprocessing and linguistic analysis
- components of modern NLP
  - text representations
  - information retrieval
  - similarity of words and documents
  - language and graphs
  - large language models
- practical use of NLP:
  - sentiment analysis,
  - paper recommendations
  - summarization

Understanding language



- A grand challenge of (not only?) artificial intelligence

  Who can understand me?
  Myself I am lost
  Searching but cannot see
  Hoping no matter cost
  Am I free?
  Or universally bossed?

- Not just poetry, what about instructions, user manuals, newspaper articles, seminary works, internet forums, twits, legal documents, i.e. license agreements, etc.

An example: rules

Article 18 of FRI Study Rules and Regulations

Taking exams at an earlier date may be allowed on request of the student by the Vice-Dean of Academic Affairs with the course convener's consent in case of mitigating circumstances (leaving for study or placement abroad, hospitalization at the time of the exam period, giving birth, participation at a professional or cultural event or a professional sports competition, etc.), and if the applicant's study achievements in previous study years are deemed satisfactory for such an authorization to be appropriate.

Understanding NL by computers

- Understanding words, syntax, semantics, context, writer's intentions, knowledge, background, assumptions, bias …
- Doesn't seem that LLM do it, though they generate excellent text output
- Ambiguity in language
  - Newspaper headlines - intentional ambiguity :)
    - Juvenile court to try shooting defendant
    - Kids make nutritious snacks
    - Miners refuse to work after death
    - Doctor on Trump's health: No heart, cognitive issues

Ambiguity

- I made her duck.

- Possible interpretations:
  - I cooked waterfowl for her.
  - I cooked waterfowl belonging to her.
  - I created the (plaster?) duck she owns.
  - I caused her to quickly lower her head or body.
  - I waved my magic wand and turned her into undifferentiated waterfowl.

- Spoken ambiguity
  - eye, maid

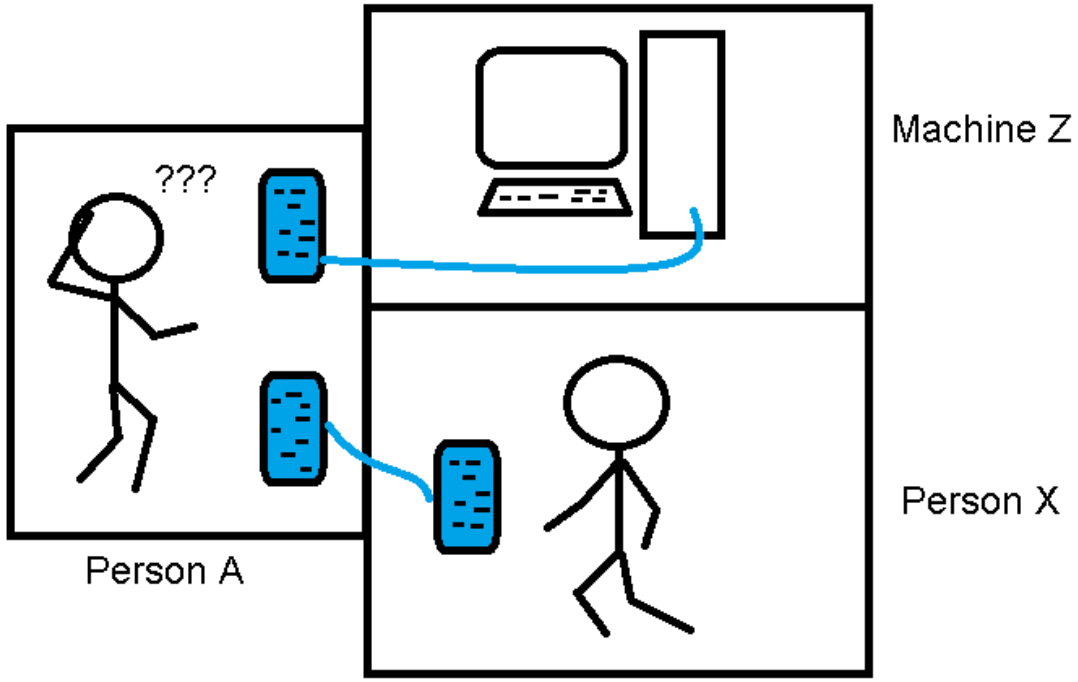
Disambiguation in syntax and semantics

- in syntax
  - The girl told the story cried.
  - John went to the bank.
  - Put the box on the table in the kitchen.
  - Bring your old car seat to be recycled.
- *Search engine queries: natural language processing, jaguar, Paris,*
- *POS tagging, word sense disambiguation*

Understanding language as (artificial) intelligence test

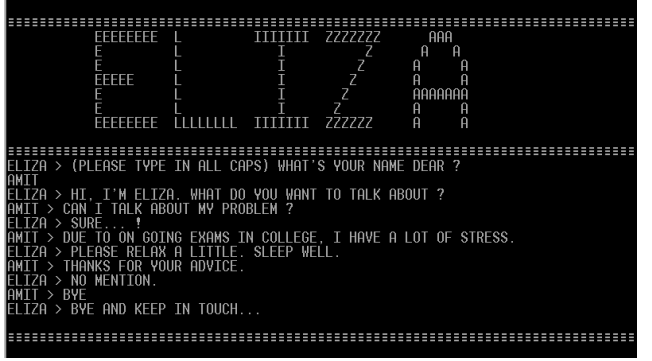
- The system understanding natural language on human level should contain a large portion of human knowledge and would achieve the most important goals of artificial intelligence. Its response would be indistinguishable from human's.

Turing test



- Turing test, Loebner prize, CYC, ELISA, ALLICA, chatterbot, VIDA, TIA, Siri (knowledge recognition, context detection, web service recommendation)
- Seems that LLMs can pass it, are they intelligent?
- The question: is mimicking human skills sufficient?

ELIZA



- ELIZA (Weizenbaum, 1966)
- Limited conversation (Rogerian psyhiatrist)

User: You are like my father in some ways.
ELIZA: What resemblance do you see?
User: You are not very aggressive but I think you don't want me to notice that.
ELIZA: What makes you think I am not aggressive?
User: You don't argue with me.
ELIZA: Why do you think I don't argue with you?
User: You are afraid of me.
ELIZA: Does it please you to believe I am afraid of you?

GPT-3 regressive language model as a chatbox

- the example is cherry-picked

```
Q: What is your favorite animal?
A: My favorite animal is a dog.

Q: Why?
A: Because dogs are loyal and friendly.

Q: What are two reasons that a dog might be in a bad mood?
A: Two reasons that a dog might be in a bad mood are if it is hungry or if it is hot.

Q: How many eyes does a giraffe have?
A: A giraffe has two eyes.

Q: How many legs does a frog have?
A: A frog has four legs.

Q: Are there any animals with three legs?
A: No, there are no animals with three legs.

Q: Why don't animals have three legs?
A: Animals don't have three legs because they would fall over.
```

Where is NLP today?

- active research area with many commercial applications
  - large language models are used everywhere
  - automatic reply engines
  - machine translation
  - text summarization
  - question answering
  - speech recognition and synthesis
  - language generation
  - interface to databases
  - intelligent search and information extraction
  - sentiment detection
  - named entity recognition and linking
  - categorization and classification of documents, messages, twits, etc.
  - cross-lingual approaches
  - multi modal approaches (text + images, text + video)
  - attempt to get to artificial general intelligence (AGI) through "foundation models"
  - many (open-source) tools and language resource
  - prevalence of deep neural network approaches (i.e. transformers)

Recommended literature

- Jurafsky, Daniel and James Martin (2024): Speech and Language Processing, 3rd edition in progress, almost all parts are available at authors‘ webpages https://web.stanford.edu/~jurafsky/slp3/
- Steven Bird, Ewan Klein, and Edward Loper. Natural Language Processing with Python. O’Reilly, 2009
  - a free book accompanying NLTK library, regularly updated
  - *Python 3, http://www.nltk.org/book/*
- Coursera
  - several courses, e.g., Stanford NLP with DNN

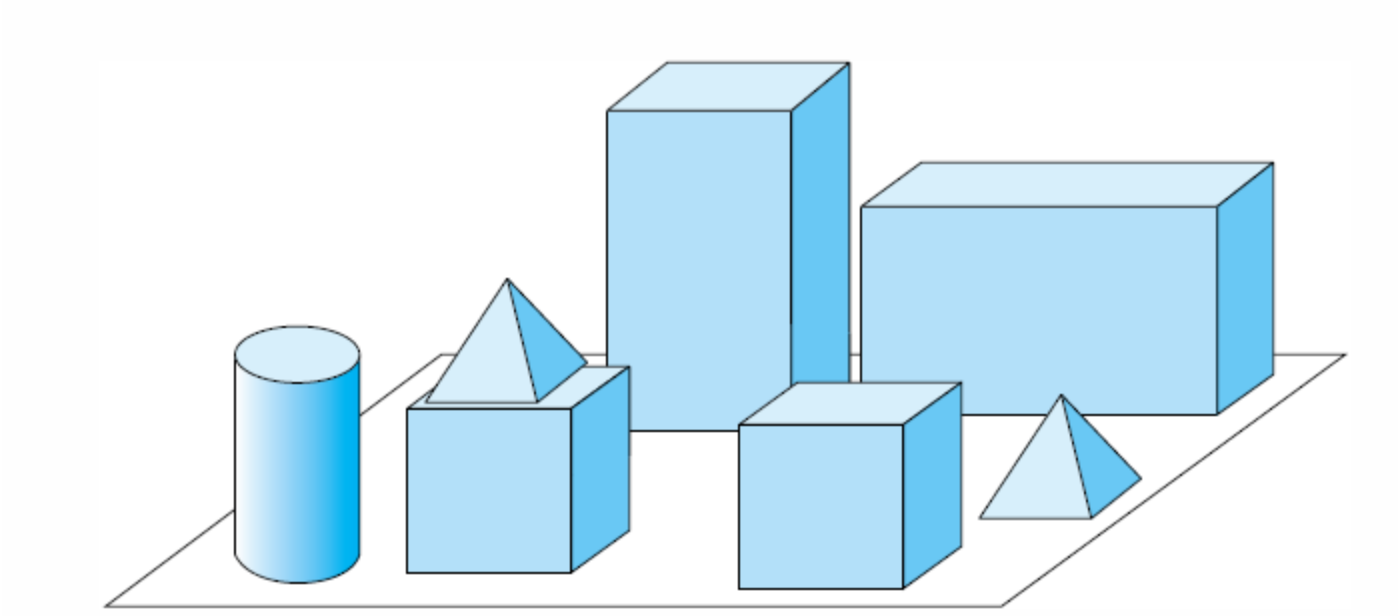
Historically two approaches

- symbolical
  - ,,Good Old-Fashioned AI''
- empirical
  - Statistical, text corpora
- Merging both worlds: injecting symbolical knowledge (e.g., propositional logic) into LLMs

How it all started?

- micro worlds
- example: SHRDLU, world of simple geometric objects
  - What is sitting on the red block?
  - What shape is the blue block on the table?
  - Place the green pyramid on the red brick.
  - Is there a red block? Pick it up.
  - What color is the block on the blue brick? Shape?

Micro world: block world, SHRDLU (Winograd, 1972)



Linguistic analysis 1/2

Linguistic analysis contains several tasks: recognition of sounds, letters, word formation, syntactic parsing, recognizing semantic, emotions. Phases:

- Prosody - the patterns of stress and intonation in a language (rhythm and intonation)
- Phonology - systems of sounds and relationships among the speech sounds that constitute the fundamental components of a language
- Morphology - the admissible arrangement of sounds in words; how to form words, prefixes and suffixes …
- Syntax - the arrangement of words and phrases to create well-formed sentences in a language

Linguistic analysis 2/2

- Semantics - the meaning of a word, phrase, sentence, or text
- Pragmatics - language in use and the contexts in which it is used, including such matters as deixis (words whose meaning changes with context, e.g., I he, here, there, soon), taking turns in conversation, text organization, presupposition, and implicature

  Can you pass me the salt? Yes, I can.
- Knowing the world: knowledge of physical world, humans, society, intentions in communications …
- Limits of linguistic analysis, levels are dependent

Classical approach to text processing

- text preprocessing
- 1. phase: syntactic analysis
- 2. phase: semantic interpretation
- 3. phase: use of world knowledge

In the neural approach, the preprocessing remains (but is simpler) the other three phases are merged into DNN

Basic tools for text preprocessing

- document → paragraphs → sentences → words (→ (subword) tokens)

- In linguistic analysis also
  - words and sentences ← lemmatization, POS tagging
  - sentences ← syntactical and grammatical analysis
  - named entity recognition,

Words and sentences

- sentence delimiters – punctuation marks and capitalization are insufficient
- E.g., remains of 1. Timbuktu from 5c BC, were discovered by dr. Barth.
- Lexical analysis (tokenizer, word segmenter), not just spaces
  - 1,999.00€ or 1.999,00€!
  - Ravne na Koroškem
  - Lebensversicherungsgesellschaft
  - Port-au-prince
  - Generalstaatsverordnetenversammlungen
- Rules, regular expressions, statistical models, dictionaries (of proper names), neural networks, manually segmented datasets

Lemmatization

- Lemmatization is the process of grouping together the different inflected forms of a word so they can be analyzed as a single item.
- walk is the lemma of 'walk', 'walked', 'walks', 'walking
- Lemmatization difficulty is language dependent i.e., depends on morphology
- Requires dictionary or lexicon for lookup
  go, goes, going, gone, went
  jaz, mene, meni, mano
- Ambiguity resolution may be difficult

Meni je vzel z mize (zapestnico).

- Uses rules, dictionaries, neural networks, manually labelled datasets
- Enlish also uses stemming (reducing inflected or derived words to their word stem

POS tagging

- assigning the correct part of speech (noun, verb, etc.) to words
- helps in recognizing phrases and names
- Use rules, machine learning models, manually labelled datasets

An example:

- Text analyzer for Slovene, i.e. morphosyntactical tagging, available at https://orodja.cjvt.si/oznacevalnik/slv/

  Nekega dne sem se napotil v naravo. Že spočetka me je žulil čevelj, a sem na to povsem pozabil, ko sem jo zagledal. Bila je prelepa. Povsem nezakrita se je sončila na trati ob poti. Pritisk se mi je dvignil v višave. Popoln primerek kmečke lastovke!

- Tags are standardized, for East European languages in Multext-East specification, e.g.,

dne; tag Somer = Samostalnik, obče ime, moški spol, ednina, rodilnik; lema: dan

a unifying attempt: universal dependencies (UD): cross-linguistically consistent treebank annotation for many languages

CLASSLA-Stanza pipeline output

ID	Oblika	Lema	Oznaka JOS	Bes. vrsta JOS	Oblikoskladenjske lastnosti JOS	Bes. vrsta UD	Oblikoskladenjske lastnosti UD	Nadrejena pojavnica UD	Skladenjska relacija UD	Nadrejena pojavnica JOS	Skladenjska relacija JOS	Udeleže vloga
# paragraph 1												
# sent_id 1.1												
# text = Nekega dne sem se napotil v naravo.												
1	Nekega	nek	Pi-msg	Pronoun	Type=indefinite Gender=masculine Number=singular Case=genitive	DET	Case=Gen|Gender=Masc|Number=Sing|PronType=Ind	2	det	2	Atr	
2	dne	dan	Ncmsg	Noun	Type=common Gender=masculine Number=singular Case=genitive	NOUN	Case=Gen|Gender=Masc|Number=Sing	5	obl	5	AdvO	TIME
3	sem	biti	Va-r1s-n	Verb	Type=auxiliary VForm=present Person=first Number=singular Negative=no	AUX	Mood=Ind|Number=Sing|Person=1|Polarity=Pos|Tense=Pres|VerbForm=Fin	5	aux	5	PPart	
4	se	se	Px------y	Pronoun	Type=reflexive Clitic=yes	PRON	PronType=Prs|Reflex=Yes|Variant=Short	5	expl	5	PPart	
5	napotil	napotiti	Vmep-sm	Verb	Type=main Aspect=perfective VForm=participle Number=singular Gender=masculine	VERB	Aspect=Perf|Gender=Masc|Number=Sing|VerbForm=Part	0	root	0	Root	
6	v	v	Sa	Adposition	Case=accusative	ADP	Case=Acc	7	case	7	Atr	
7	naravo	narava	Ncfsa	Noun	Type=common Gender=feminine Number=singular Case=accusative	NOUN	Case=Acc|Gender=Fem|Number=Sing	5	obl	5	AdvO	GOAL
8	.	.	Z	Punctuation		PUNCT	_	5	punct	0	Root	

- *Nekega dne sem se napotil v naravo. Že spočetka me je žulil čevelj, a sem na to povsem pozabil, ko sem jo zagledal. Bila je prelepa. Povsem nezakrita se je sončila na trati ob poti. Pritisk se mi je dvignil v višave. Popoln primerek kmečke lastovke!*

1	**beseda**	Nekega	dne	sem	se	napotil	v	naravo	.	Že	spočetka	me	je								
	lema	nek	dan	biti	se	napotiti	v	narava	že	spočetka	jaz	biti									
	oznaka	Zn-mer	Somer	Gp-spe-n	Zp------k	Ggdd-em	Dt	Sozet	.	L	Rsn	Zop-et--k	Gp-ste-n								
2	**beseda**	žulil	čevelj	,	a	sem	na to	povsem	pozabil	,	ko	sem	jo	zagledal							
	lema	žuliti	čevelj	a	biti	na ta	povsem	pozabiti	ko	biti	on	zagledati									
	oznaka	Ggnd-em	Somei	,	Vp	Gp-spe-n	Dt	Zk-set	Rsn	Ggdd-em	,	Vd	Gp-spe-n	Zotzet--k	Ggdd-em						
3	**beseda**	.	Bila	je	prelepa	.	Povsem	nezakrita	se	je	sončila	na	trati								
	lema		biti	biti	prelep		povsem	nezakrit	se	biti	sončiti	na	trata								
	oznaka	.	Gp-d-ez	Gp-ste-n	Ppnzei	.	Rsn	Ppnzei	Zp------k	Gp-ste-n	Ggvd-ez	Dm	Sozem								
4	**beseda**	ob	poti	.	Pritisk	se	mi	je	dvignil	v	višave	.	Popoln								
	lema	ob	pot		pritisk	se	jaz	biti	dvigniti	v	višava		popoln								
	oznaka	Dm	Sozem	.	Somei	Zp------k	Zop-ed--k	Gp-ste-n	Ggdd-em	Dt	Sozmt	.	Ppnmein								
5	**beseda**	primerek	kmečke	lastovke	!																
	lema	primerek	kmečki	lastovka																	
	oznaka	Somei	Ppnzer	Sozer	!																

TEI-XML format

```
<TEI xmlns="http://www.tei-c.org/ns/1.0">
  <text>
    <body>
      <p>
        <s>
          <w msd="Zn-mer" lemma="nek">Nekega</w>
          <S/>
          <w msd="Somer" lemma="dan">dne</w>
          <S/>
          <w msd="Gp-spe-n" lemma="biti">sem</w>
          <S/>
          <w msd="Zp------k" lemma="se">se</w>
          <S/>
          <w msd="Ggdd-em" lemma="napotiti">napotil</w>
          <S/>
          <w msd="Dt" lemma="v">v</w>
          <S/>
          <w msd="Sozet" lemma="narava">naravo</w>
          <c>.</c>
          <S/>
        </s>
   …

      </p>
    </body>
  </text>
</TEI>
```

MSD tags

- Multext-East specification

dne; tag Somer = Samostalnik, obče ime, moški spol, ednina, rodilnik; lema: dan

P	atribut	vrednost	koda	atribut	vrednost	koda
0	glagol		G	Verb		V
1	vrsta	glavni	g	Type	main	m
		pomožni	p		auxiliary	a
2	vid	dovršni	d	Aspect	perfective	e
		nedovršni	n		imperfective	p
		dvovidski	v		biaspectual	b
3	oblika	nedoločnik	n	VForm	infinitive	n
		namenilnik	m		supine	u
		deležnik	d		participle	p
		sedanjik	s		present	r
		prihodnjik	p		future	f
		pogojnik	g		conditional	c
		velelnik	v		imperative	m
4	oseba	prva	p	Person	first	1
		druga	d		second	2
		tretja	t		third	3
5	število	ednina	e	Number	singular	s
		množina	m		plural	p
		dvojina	d		dual	d
6	spol	moški	m	Gender	masculine	m
		ženski	z		feminine	f
		srednji	s		neuter	n
7	nikalnost	nezanikani	n	Negative	no	n
		zanikani	d		yes	y

POS tagging in English

- http://nlpdotnet.com/Services/Tagger.aspx
- Rainer Maria Rilke, 1903
  in Letters to a Young Poet

...I would like to beg you dear Sir, as well as I can, to have patience with everything unresolved in your heart and to try to love the questions themselves as if they were locked rooms or books written in a very foreign language. Don't search for the answers, which could not be given to you now, because you would not be able to live them. And the point is to live everything. Live the questions now. Perhaps then, someday far in the future, you will gradually, without even noticing it, live your way into the answer.

POS tagger output

I/PRP would/MD like/VB to/TO beg/VB you/PRP dear/JJ Sir/NNP ,/, as/RB well/RB as/IN I/PRP can/MD ,/, to/IN have/VBP patience/NN with/IN everything/NN unresolved/JJ in/IN your/PRP$ heart/NN and/CC to/TO try/VB to/TO love/VB the/DT questions/NNS themselves/PRP as/RB if/IN they/PRP were/VBD locked/VBN rooms/NNS or/CC books/NNS written/VBN in/IN a/DT very/RB foreign/JJ language/NN ./.

Named entity recognition (NER)

- NATO Secretary-General Jens Stoltenberg is expected to travel to Washington, D.C. to meet with U.S. leaders.
- [ORG NATO] Secretary-General [PER Jens Stoltenberg] is expected to travel to [LOC Washington, D.C.] to meet with [LOC U.S.] leaders.
- Named entity linking (NEL) also named entity disambiguation – linking to a unique identifier, e.g. wikification
  jaguar, Paris, London, Dunaj

Basic language resources: corpora

- Statistical natural language processing list of resources http://nlp.stanford.edu/links/statnlp.html
- Opus http://opus.nlpl.eu/ multilingual parallel corpora, e.g., DGT JRC-Acqui 3.0, Documents of the EU in 22 languages
- Slovene language corpora GigaFida, ccGigaFida, KRES, ccKres, GOS, Artur, JANES, KAS, Trendi
- The main Slovene language resources
  - http://www.clarin.si
  - https://github.com/clarinsi
  - http://www.cjvt.si/
  - https://www.slovenscina.eu/
- WordNet, SloWNet, sentiWordNet, …
- Thesaurus https://viri.cjvt.si/sopomenke/slv/
- LLMs: on HuggingFace cjvt, e.g., SloBERTa and GaMS

WordNet is a database composed of synsets:

synonyms,
hypernyms
hyponyms,
meronyms,
holonyms,
etc.

Word to search for: mercy [Search WordNet]

Display Options: (Select option to change) [Change]

Key: "S:" = Show Synset (semantic) relations, "W:" = Show Word (lexical) relations

Display options for sense: (gloss) "an example sentence"

Noun

- S: (n) clemency, mercifulness, **mercy** (leniency and compassion shown toward offenders by a person or agency charged with administering justice) *"he threw himself on the mercy of the court"*
- S: (n) mercifulness, **mercy** (a disposition to be kind and forgiving) *"in those days a wife had to depend on the mercifulness of her husband"*
- S: (n) mercifulness, **mercy** (the feeling that motivates compassion)
  - *direct hyponym* / *full hyponym*
    - S: (n) forgiveness (compassionate feelings that support a willingness to forgive)
  - *direct hypernym* / *inherited hypernym* / *sister term*
    - S: (n) compassion, compassionateness (a deep awareness of and sympathy for another's suffering)
  - ***derivationally related form***
    - W: (adj) merciful [Related to: mercifulness] (showing or giving mercy) *"sought merciful treatment for the captives"; "a merciful god"*
- S: (n) **mercy** (something for which to be thankful) *"it was a mercy we got out alive"*
- S: (n) **mercy** (alleviation of distress; showing great kindness toward the distressed) *"distributing food and clothing to the flood victims was an act of mercy"*

Popular NLP applications

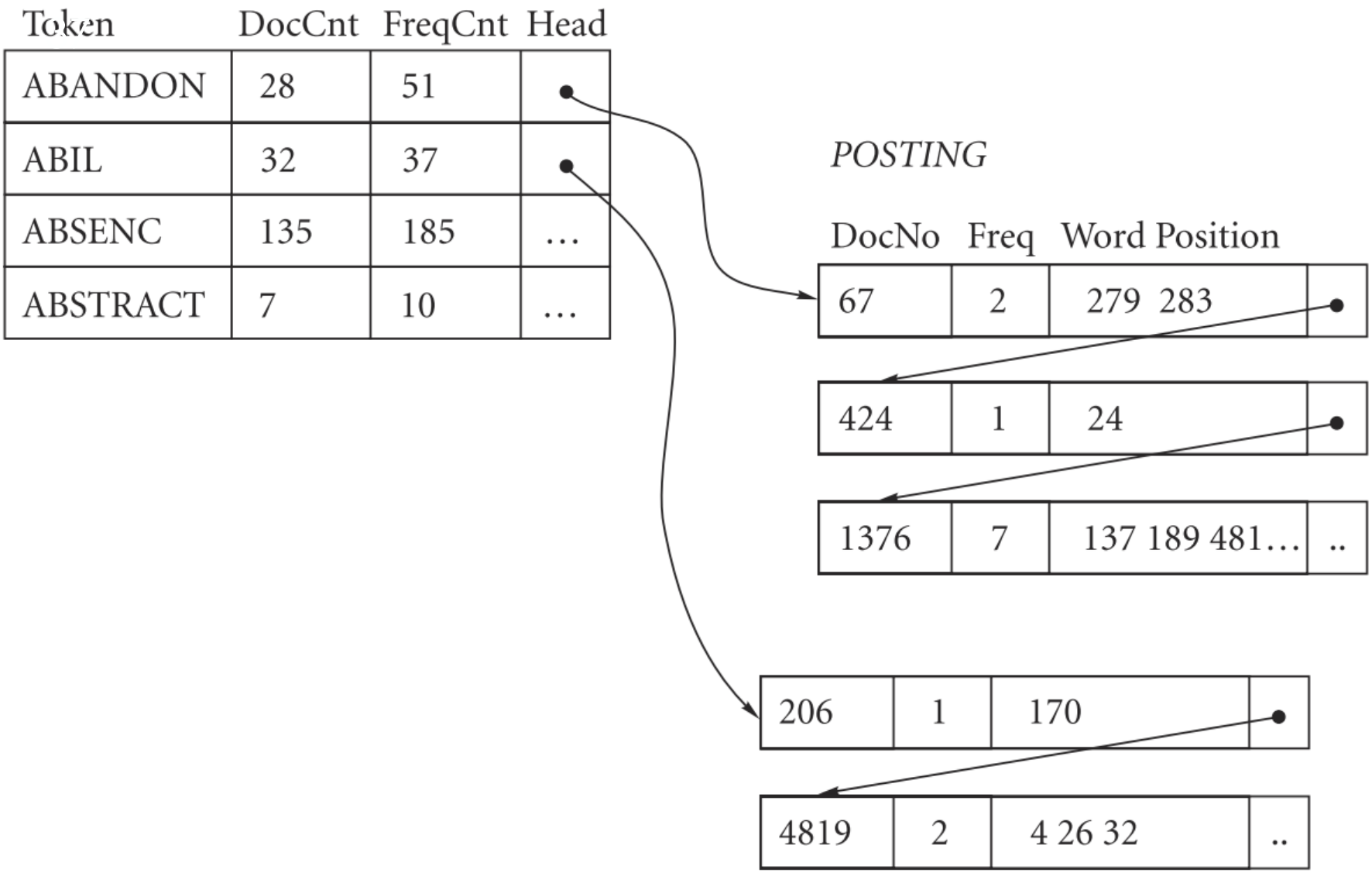
- document retrieval
- information extraction
- automatic speech recognition and generation
- text classification
- automatic summarization
- question answering
- sentiment analysis, emotion detection, stance detection
- machine translation,
- language generation
- comment filtering, hate speech detection, fake news detection
- topic analysis
- grammar tools
- many more

Document retrieval

- Historical: keywords
- Now: whole text search
- Organize a database, indexing, search algorithms
- input: a query (of questionable quality, ambiguity, answer quality)

Document indexing

- Collect all words from all documents, use lemmatization
- The inverted file data structure
- For each word keep
  - Number of appearing documents
  - Overall number of appearances
  - For each document
    - Number of appearances
    - Location


Token
DocCnt
FreqCnt
Head
ABANDON
28
51
ABIL
32
37
ABSENC
135
185
…
ABSTRACT
7
10
…
POSTING
DocNo
Freq
Word Position
67
2
279 283
424
1
24
1376
7
137 189 481…
..
206
1
170
4819
2
4 26 32
..

Full text search engine

- Most popular: Apache Lucene/Solr
- full-text search, hit highlighting, real-time indexing, dynamic clustering, database integration, NoSQL features, rich document (e.g., Word, PDF) handling.
- distributed search and index replication, scalability and fault tolerance.

Search with logical operators

- AND, OR, NOT
- jaguar AND car
  jaguar NOT animal
- Some system support neighborhood search (e.g., NEAR) and stemming (!)
  paris! NEAR(3) fr!
  president NEAR(10) bush
- libraries, concordancers
- E.g-, for Slovene: https://viri.cjvt.si/gigafida/

Logical operator search is outdated

- Large number of results
- Large specialized incomprehensible queries
- Synonyms
- Sorting of results
- No partial matching
- No weighting of query terms

Ranking based search

- Web search
- Less frequent terms are more informative
- NL input - stop words, lemmatization
- Vector based representation of documents and queries (bag-of-words or dense embeddings)

Sparse vector representation: bag-of-words

- *An elephant is a mammal. Mammals are animals. Humans are mammals, too. Elephants and humans live in Africa.*

Africa	animal	be	elephant	human	in	live	mammal	too
1	1	3	2	2	1	1	3	1

9 dimensional vector (1,1,3,2,2,1,1,3,1)

In reality this is sparse vector of dimension $|V|$ (vocabulary size in order of 10,000 dimensions)

Similarity between documents and queries in vector space.

Vectors and documents

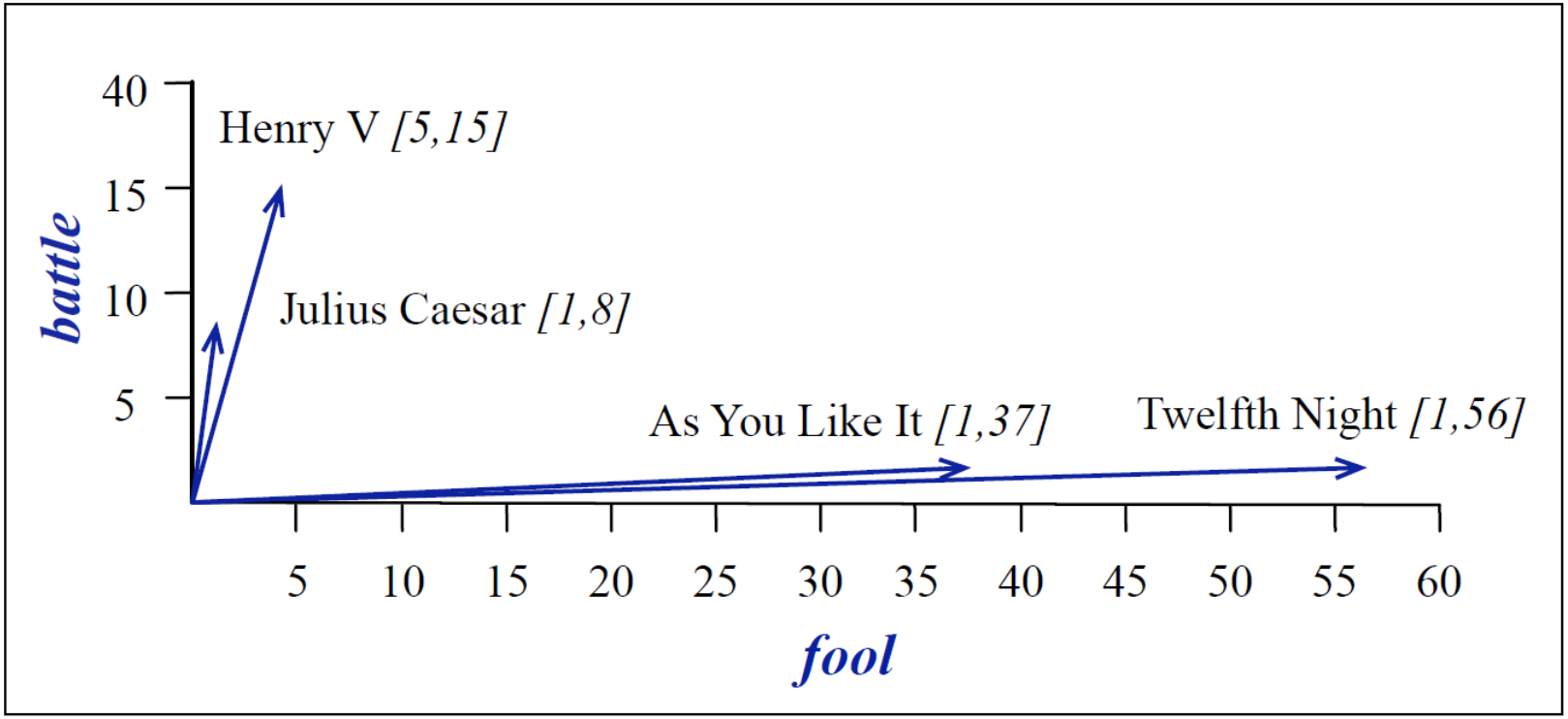
- a word occurs in several documents
- both words and documents are vectors
- an example: Shakespeare

	As You Like It	Twelfth Night	Julius Caesar	Henry V
battle	1	1	8	15
soldier	2	2	12	36
fool	37	58	1	5
clown	5	117	0	0

- term-document matrix, dimension |V| x |D|
- a sparse matrix
- word embedding

Vector based similarity

- e.g., in two dimensional space



- the difference between dramas and comedies

Document similarity

- Assume orthogonal dimensions
- Cosine similarity
- Dot (scalar) product of vectors

$$\cos(\Theta) = \frac{A \cdot B}{|A||B|}$$

Importance of words

- Frequencies of words in particular document and overall
- inverse document frequency idf
  - N = number of documents in collection
  - n_b = number of documents with word b

$$idf_b = \log(\frac{N}{n_b})$$

Weighting dimensions (words)

- Weight of word b in document d

$$w_{b,d} = tf_{b,d} \times idf_{b,d}$$

$tf_{b,d}$ = frequency of term b in document d

- called TF-IDF weighting (an improvement over bag-of-words)

Weighted similarity

- Between query and document

$$sim(q,d) = \frac{\sum_b w_{b,d} \cdot w_{b,q}}{\sqrt{\sum_b w_{b,d}^2} \cdot \sqrt{\sum_b w_{b,q}^2}}$$

- Ranking by the decreasing similarity

Performance measures for search

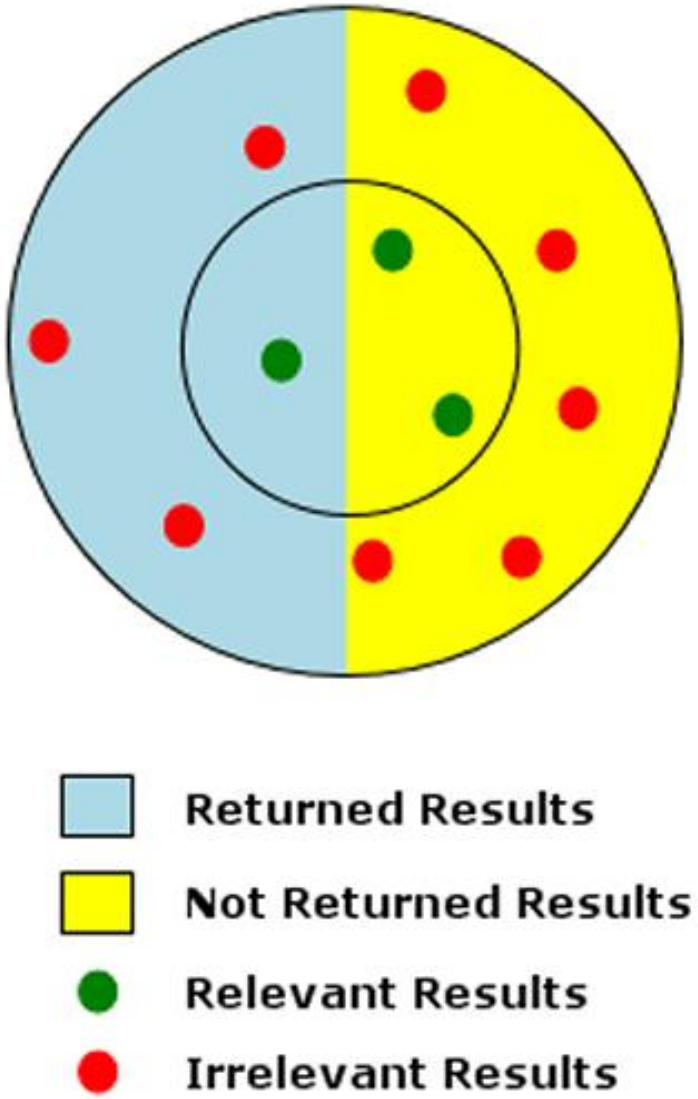
- Statistical measures
- Subjective measures
- Precision, recall
- A contingency table analysis of precision and recall

	Relevant	Non-relevant	
Retrieved	a	b	$a + b = m$
Not retrieved	c	d	$c + d = N - m$
	$a + c = n$	$b + d = N - n$	$a + b + c + d = N$

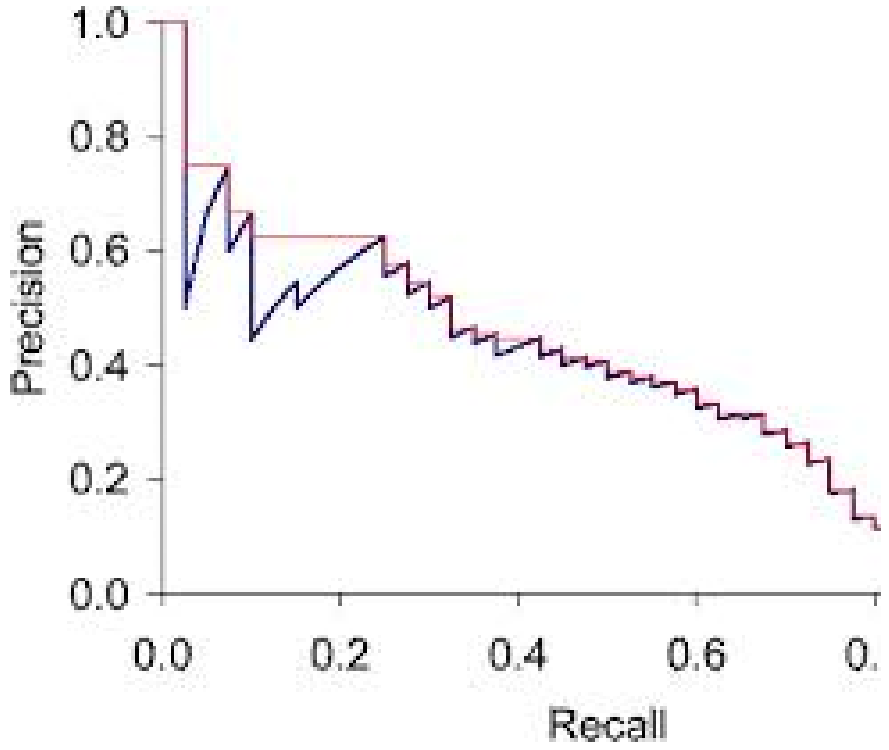
Precision and recall

- N = number of documents in collection
- n = number of important documents for given query q
- Search returns m documents including a relevant ones
- Precision $P = a/m$
  proportion of relevant document in the obtained ones
- Recall $R = a/n$
  proportion of obtained relevant documents
- Precision recall graphs

An example: low precision, low recall



Precision-recall graphs



F-measure

- combine both P and R

$$F_\beta = \frac{(1+\beta^2)\cdot P \cdot R}{\beta^2 P + R} \text{ for } \beta > 0$$

$$F_1 = \frac{2 \cdot P \cdot R}{P + R}$$

- Weighted precision and recall
- β=1 weighted harmonic mean
- Also used β=2 or β= 0.5

Ranking measures

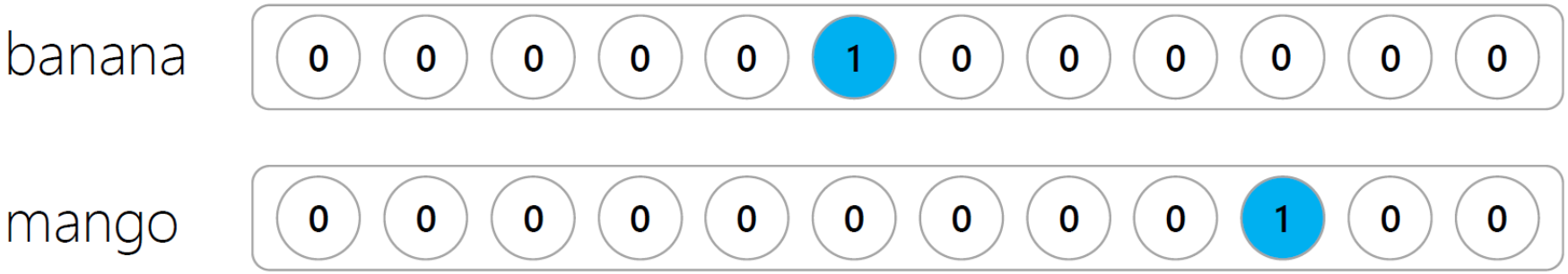
- precision@k
  - proportion of relevant document in the first k obtained ones
- recall@k
  - proportion of relevant documents in the k obtained among all relevant
- F_1@k
- mean reciprocal rank

$$MRR = \frac{1}{Q}\sum_{i=1}^{Q}\frac{1}{\mathrm{rank}_i}$$

  - over Q queries,
  - considers only the rank of the best answer

Why dense textual embeddings?

- Best machine learning models for text, i.e. deep neural networks, require numerical input.
- Simple representations like 1-hot-encoding and bag-of-words do not preserve semantic similarity.
- We need dense vector represenation for text elements.



Dense vector embeddings

- advantages compared to sparse embeddings:
  - less dimensions, less space
  - easier input for ML methods
  - potential generalization and noise reduction
  - potentially captures synonymy, e.g., road and highway are different dimensions in BOW
- the most popular approaches
  - matrix based transformations to reduce dimensionality (SVD in LSA)
  - neural embeddings (word2vec, Glove)
  - explicit contextual neural embeddings (ELMo, BERT)
  - only use an implicit representation in LLM

Meaning focused on similarity

- Each word is a vector
- Similar words are "nearby in space"

Distributional semantics

"You shall know a word
by the company it keeps"

Firth, J. R. (1957). A synopsis of linguistic theory 1930–1955. In Studies in Linguistic Analysis, p. 11. Blackwell, Oxford.

"The meaning of a word is its use in the language"

Ludwig Wittgenstein, PI #43

Neural embeddings

- neural network is trained to predict the context of words (input: word, output: context of neighboring words)

word2vec method

- Train a classifier on a binary **prediction** task:
  Is word *w* likely to show up near a given word, e.g., "*apricot*"?
- We don't actually care about this task
- But we'll take the learned classifier weights as the word embeddings

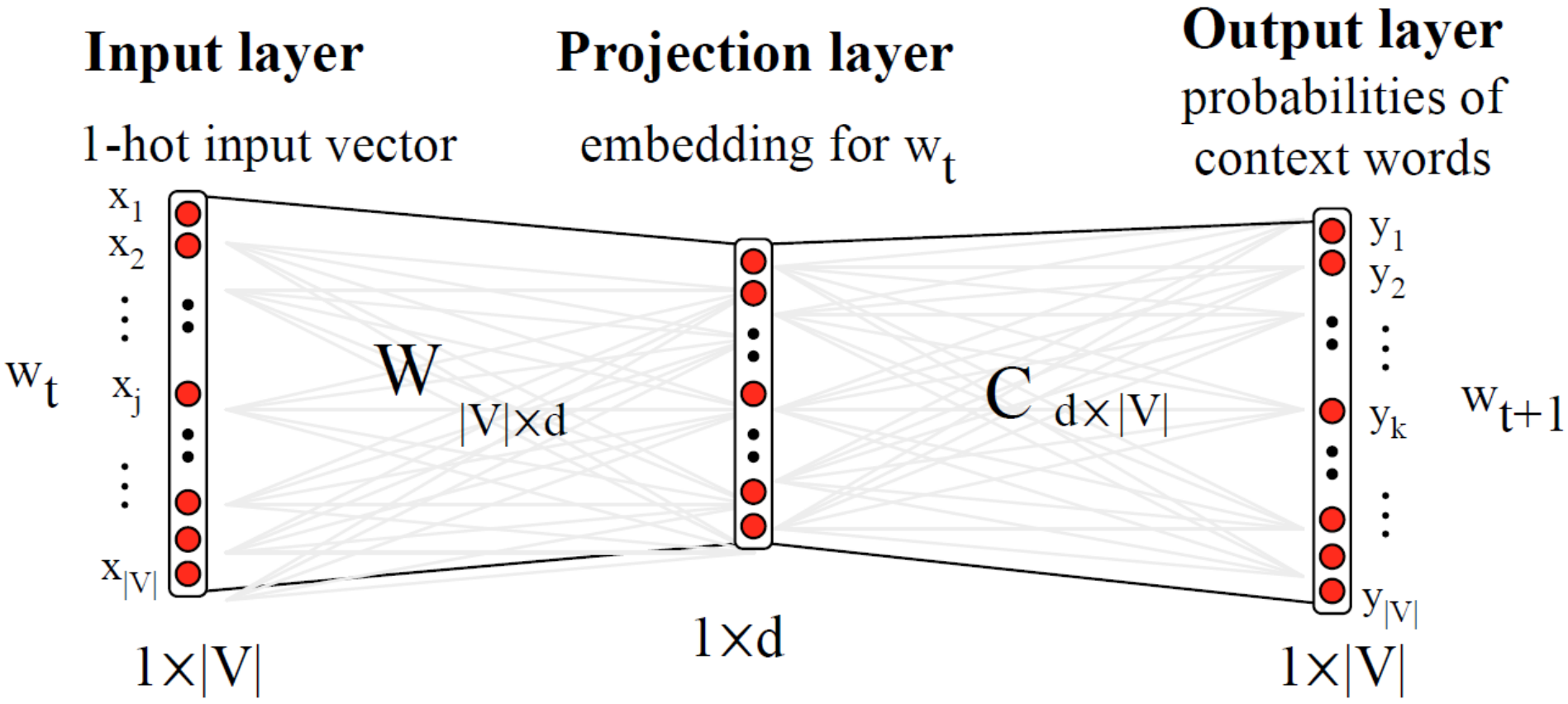
- Words near apricot acts as 'correct answers' to the question "Is word w likely to show up near apricot?"
- No need for hand-labeled supervision

word2vec (skip-gram) training data

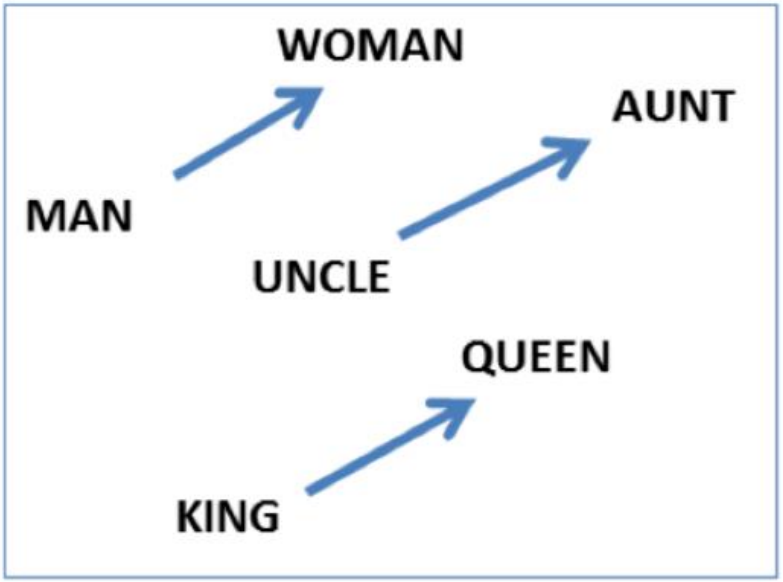
- Training sentence:
- ... lemon, a tablespoon of **apricot** jam a pinch ...
- c1 c2 target c3 c4

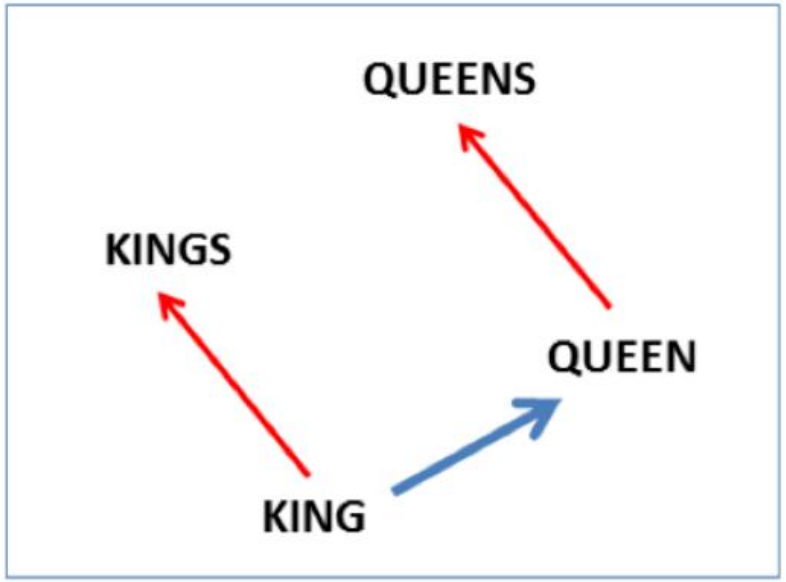
- Asssume context words are those in +/- 2 word window
- Produce the following input-outputs for positive instances: (apricot, tablespoon), (apricot, of), (apricot, jam), (apricot, a)
- Get negative training examples randomly
- Train a neural network to predict the probability of a co-occurring word

Neural network based embedding



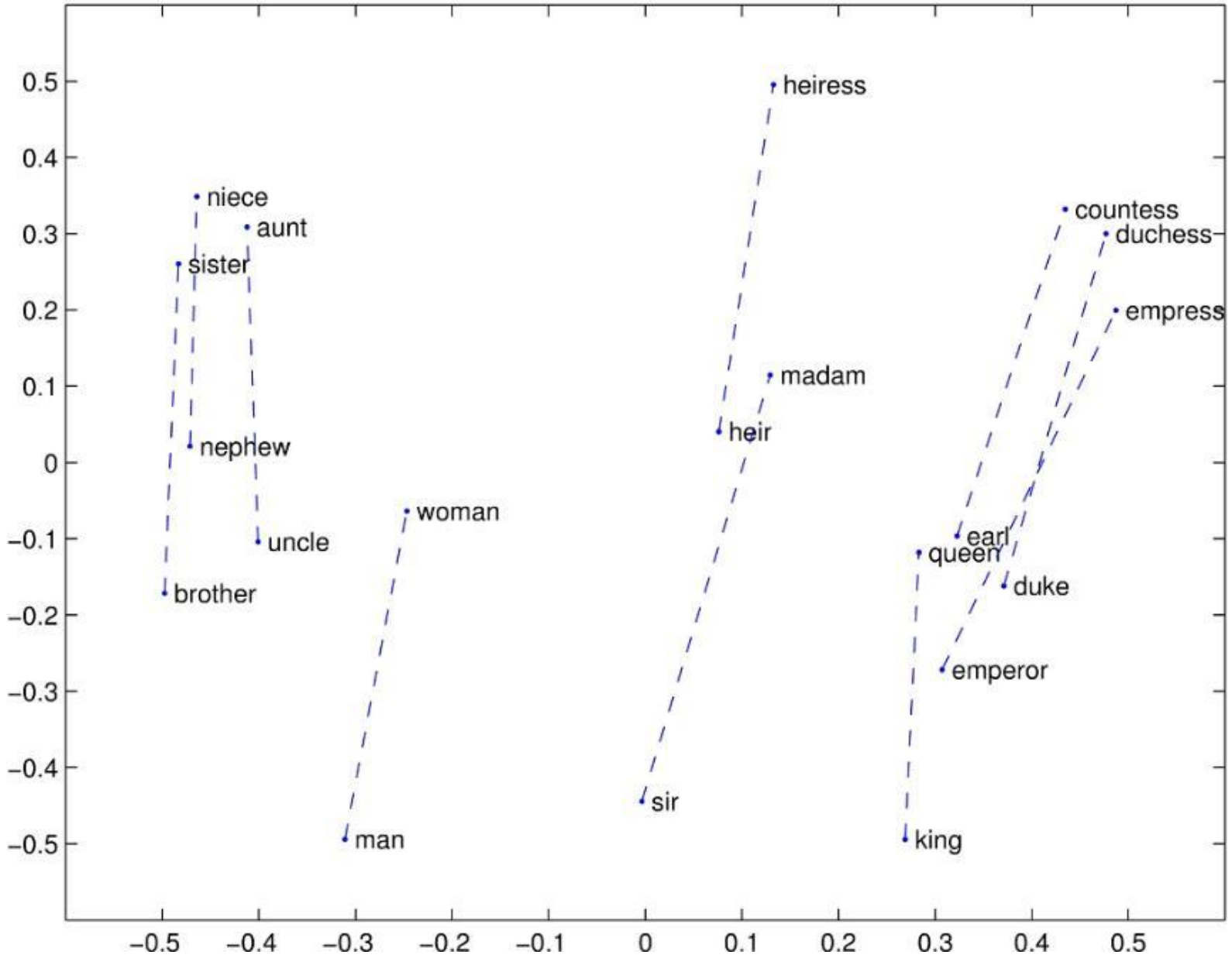
Relational similarity

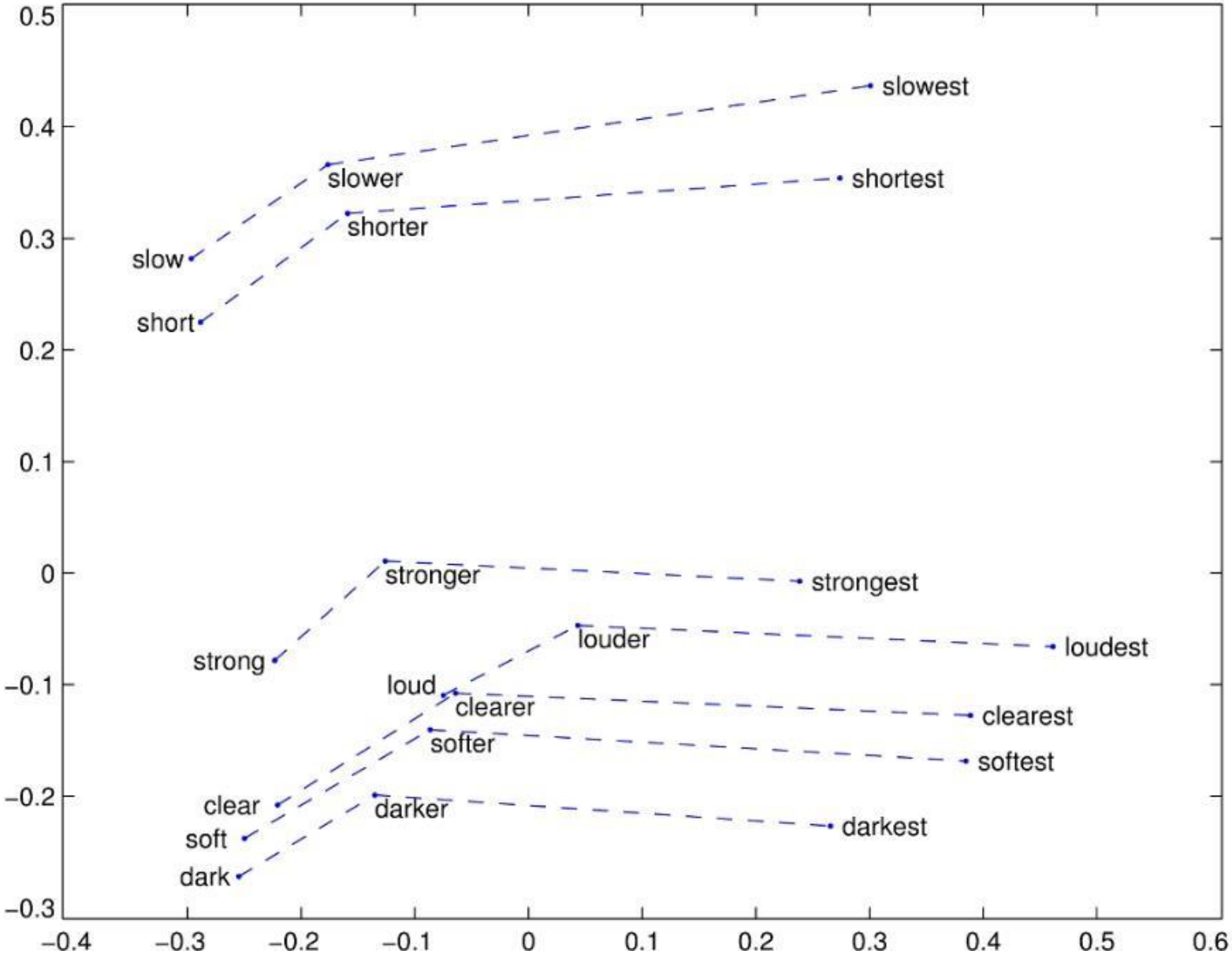




vector(‘*king*’) - vector(‘*man*’) + vector(‘*woman*’) ≈ vector(‘queen’)

vector(‘*Paris*’) - vector(‘*France*’) + vector(‘*Italy*’) ≈ vector(‘Rome’)

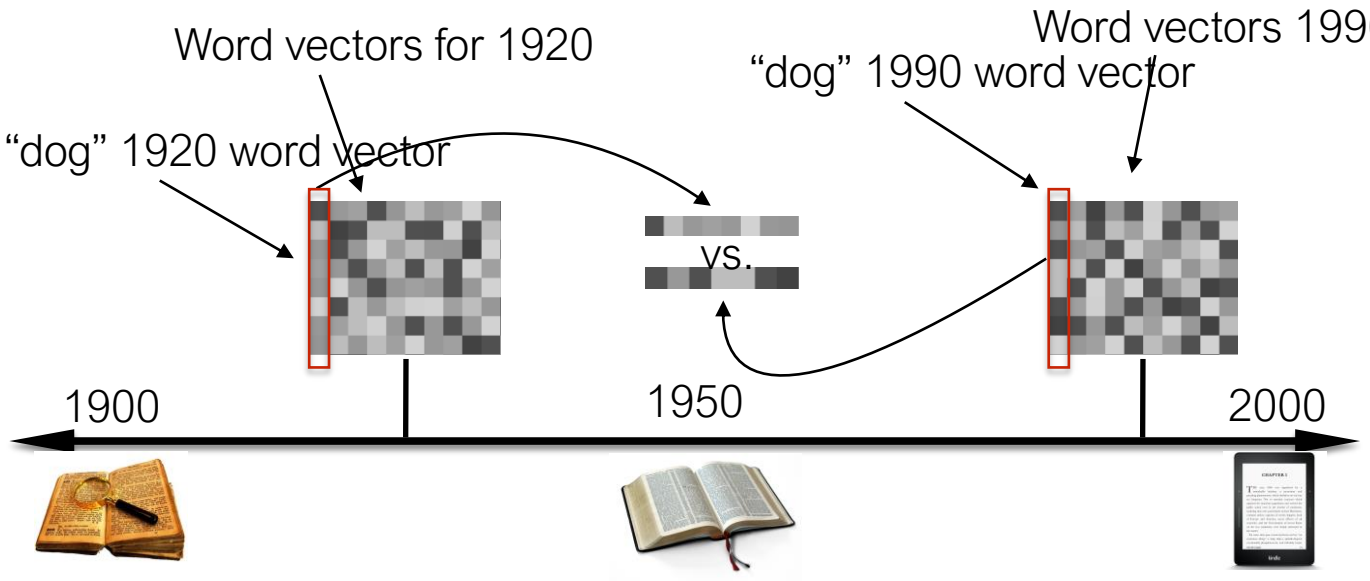

0.5
0.4
0.3
0.2
0.1
0
−0.1
−0.2
−0.3
−0.4
−0.5
−0.5
−0.4
−0.3
−0.2
−0.1
0
0.1
0.2
0.3
0.4
0.5
niece
aunt
sister
nephew
uncle
brother
woman
man
heiress
madam
heir
sir
countess
duchess
empress
earl
queen
duke
emperor
king


slowest
slower
shortest
shorter
slow
short
stronger
strongest
strong
louder
loudest
loud
clearer
clearest
softer
softest
clear
darker
darkest
soft
dark
0.5
0.4
0.3
0.2
0.1
0
−0.1
−0.2
−0.3
−0.4
−0.3
−0.2
−0.1
0
0.1
0.2
0.3
0.4
0.5
0.6

Embeddings can help study word history

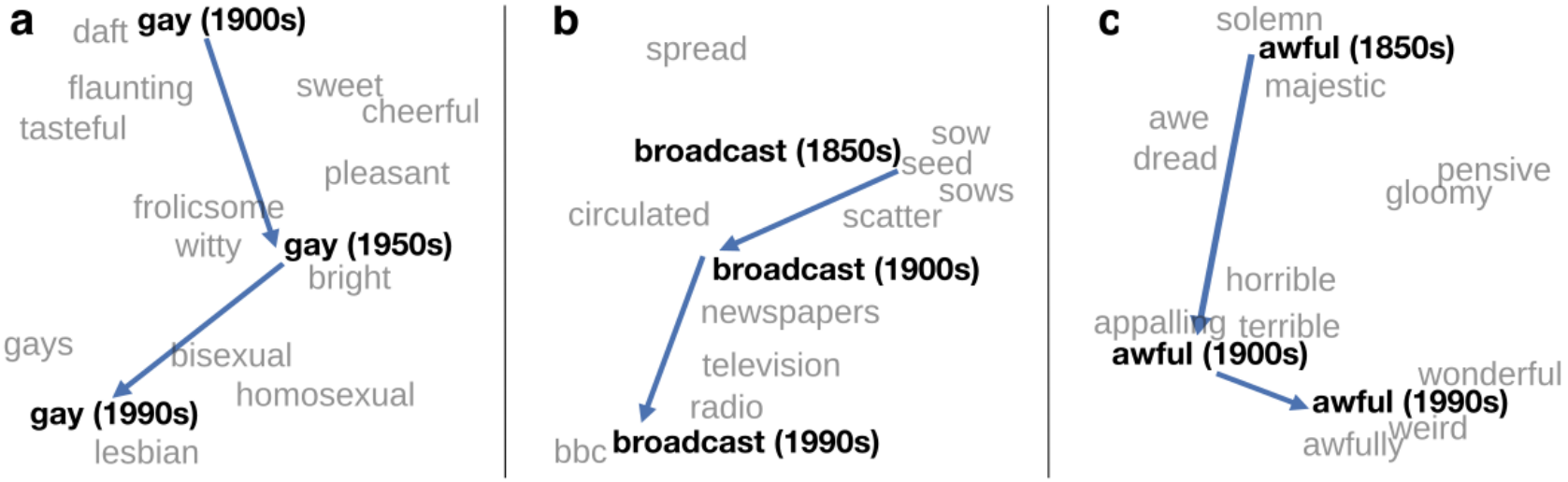
- Train embeddings on old books to study changes in word meaning!!

Diachronic word embeddings for studying language change



Visualizing changes

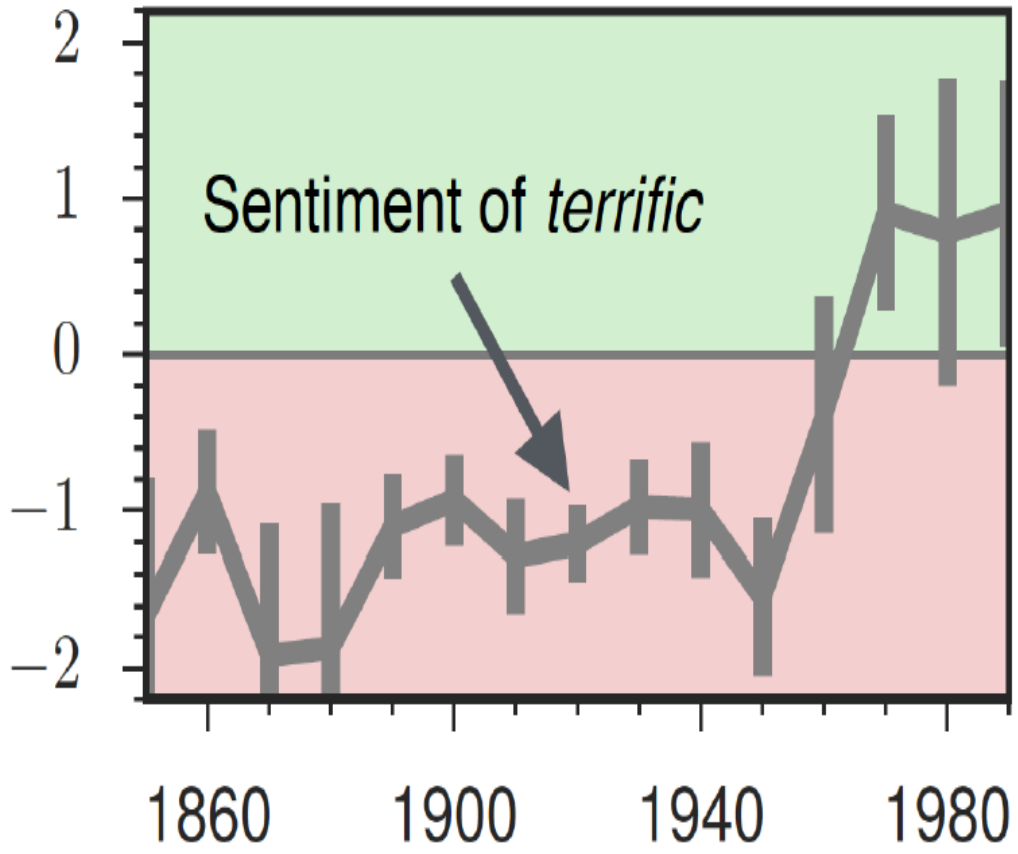
Project 300 dimensions down into 2



~30 million books, 1850-1990, Google Books data

The evolution of sentiment words

Negative words change faster than positive words



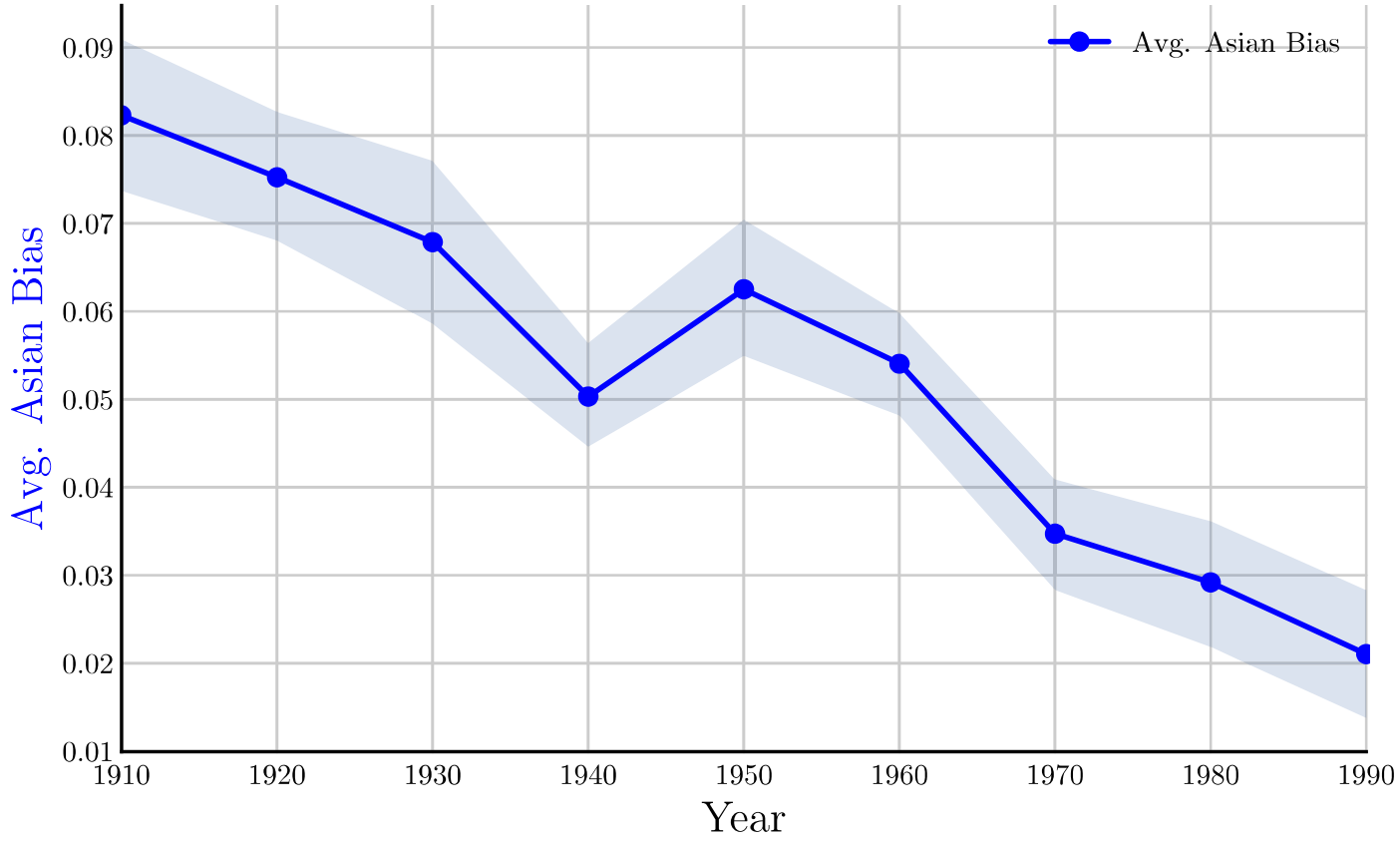
Embeddings reflect cultural bias

- Ask “Paris : France :: Tokyo : x”
  - x = Japan
- Ask “father : doctor :: mother : x”
  - x = nurse
- Ask “man : computer programmer :: woman : x”
  - x = homemaker

Bolukbasi, Tolga, Kai-Wei Chang, James Y. Zou, Venkatesh Saligrama, and Adam T. Kalai. "Man is to computer programmer as woman is to homemaker? debiasing word embeddings." In *Advances in Neural Information Processing Systems*, pp. 4349-4357. 2016.

Change in linguistic framing 1910-1990

Change in association of Chinese names with adjectives framed as "othering" (*barbaric*, *monstrous*, *bizarre*)



Garg, Nikhil, Schiebinger, Londa, Jurafsky, Dan, and Zou, James (2018). Word embeddings quantify 100 years of gender and ethnic stereotypes. *Proceedings of the National Academy of Sciences*, *115*(16), E3635–E3644

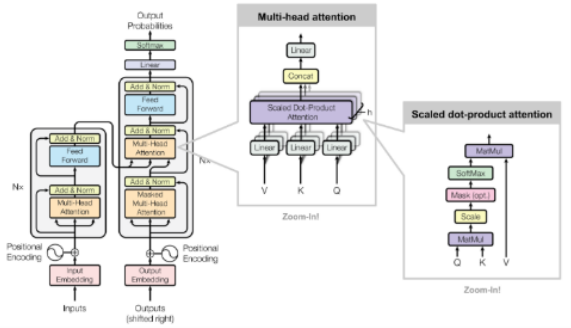
Contextual embeddings / Large language models

- word2vec produces the same vector for a word like bank irrespective of its meaning and context
- recent embeddings take the context into account
- already established as a standard
- e.g., BERT for contextual word embeddings

University of Ljubljana
Faculty of Computer and Information Science

Large language models (LLMs)

- pretrained neural large language models
- trained on large text corpora to capture relations in language
- finetuned to specific tasks
- many are publicly available
- on HuggingFace

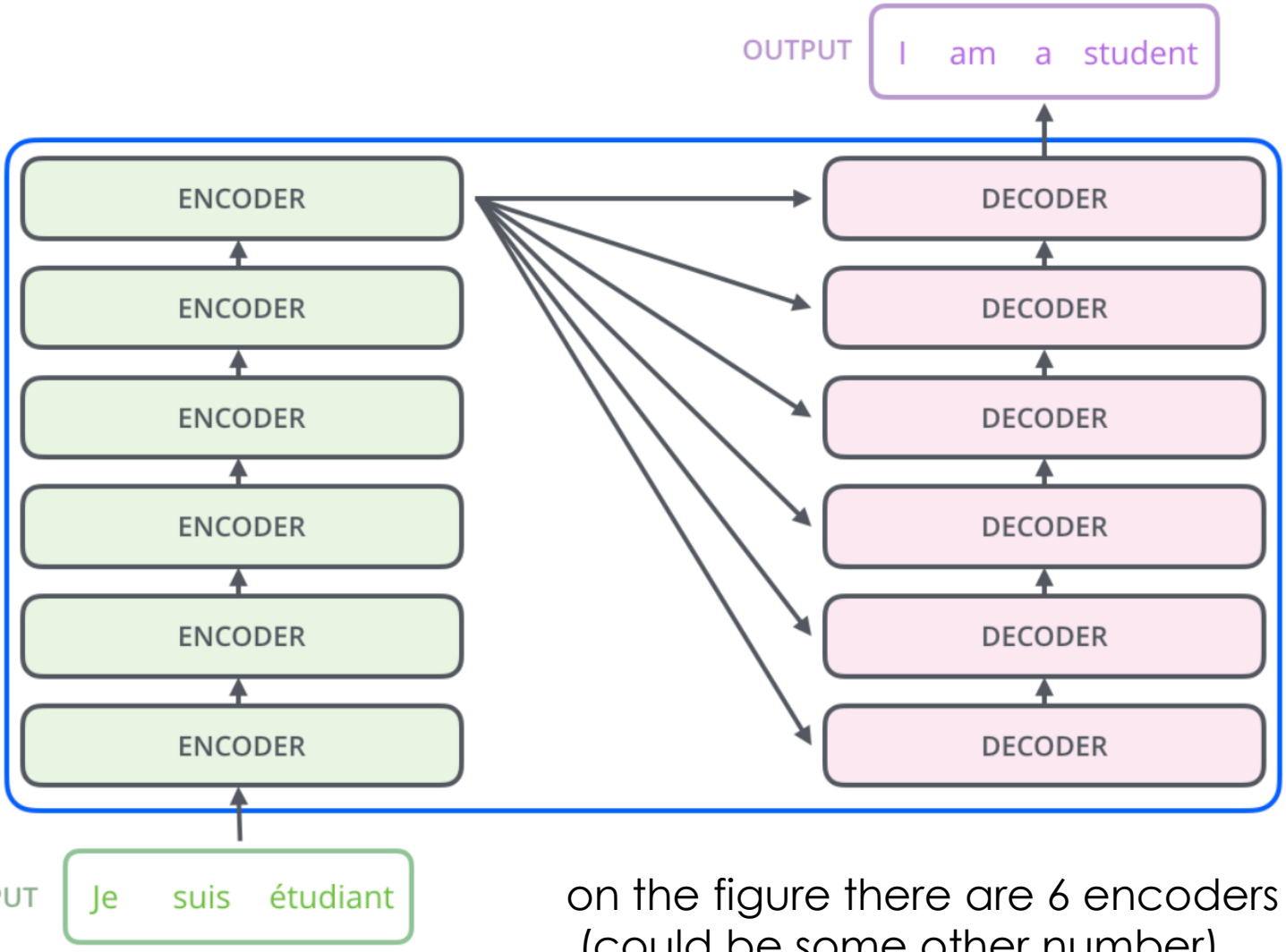


Transformer architecture of NNs

- currently the most successful DNN
- non-recurrent
- architecturally it is an encoder-decoder model
- fixed input length
- can be parallelized
- adapted for GPU (TPU) processing
- based on extreme use of attention

- https://github.com/dair-ai/Transformers-Recipe

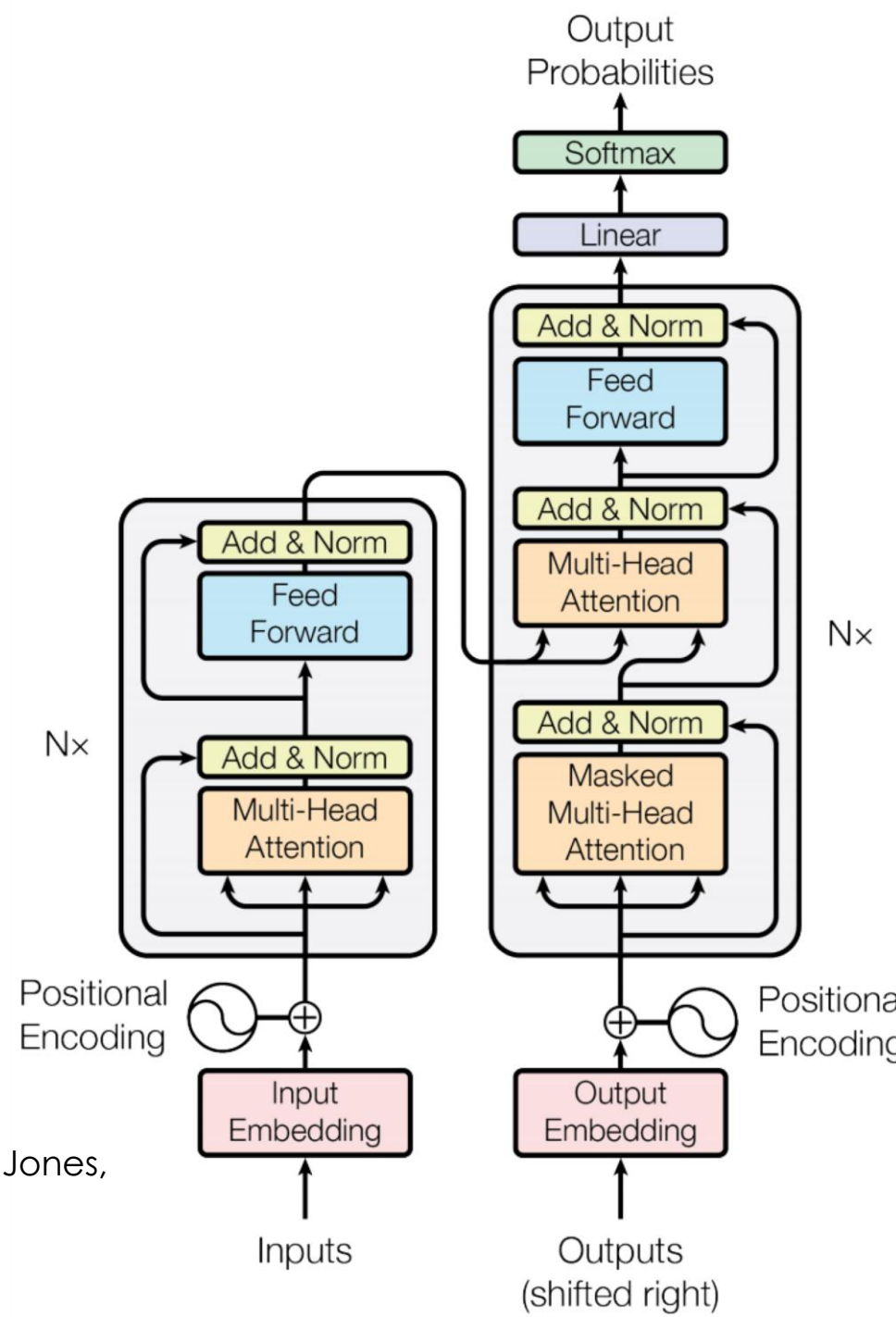
Transformer is an encoder-decoder model



on the figure there are 6 encoders and 6 decoders
(could be some other number)

Transformer overview

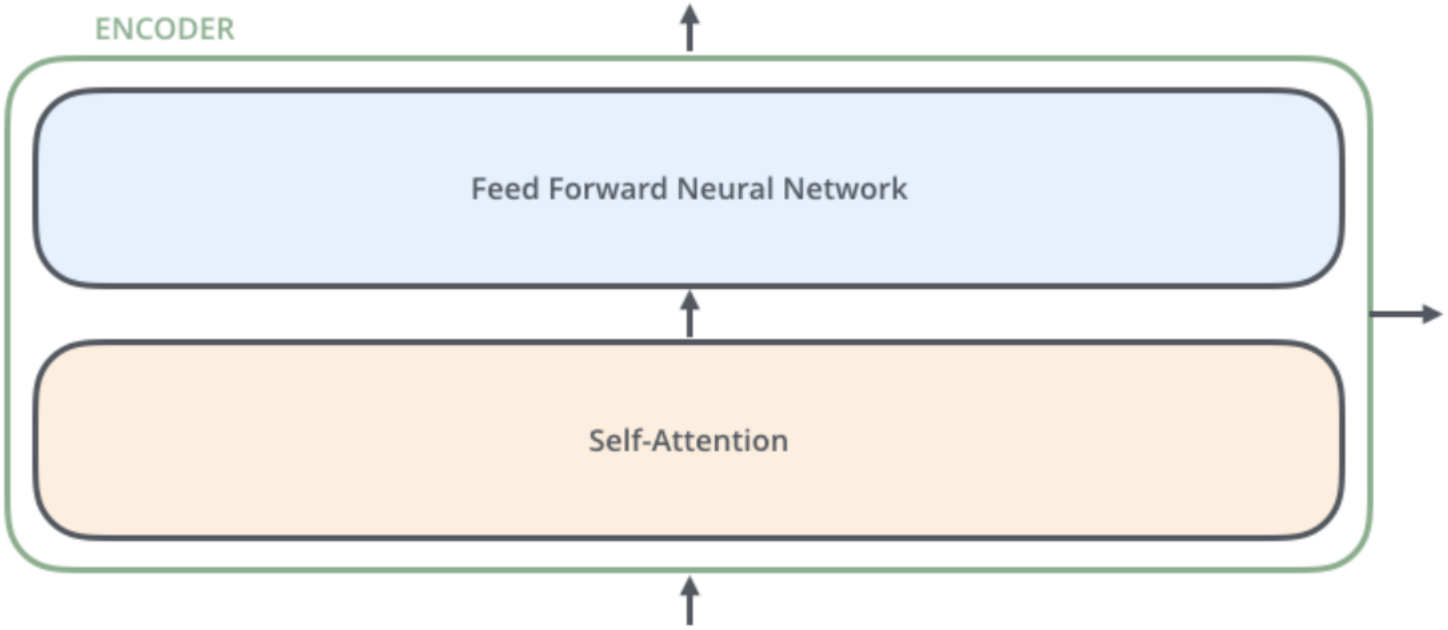
- Initial task: machine translation with parallel corpus
- Predict each translated word
- Final cost/loss/error function is standard cross-entropy error on top of a softmax classifier



Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł. and Polosukhin, I., 2017. Attention is all you need. In *Advances in neural information processing systems* (pp. 5998-6008).

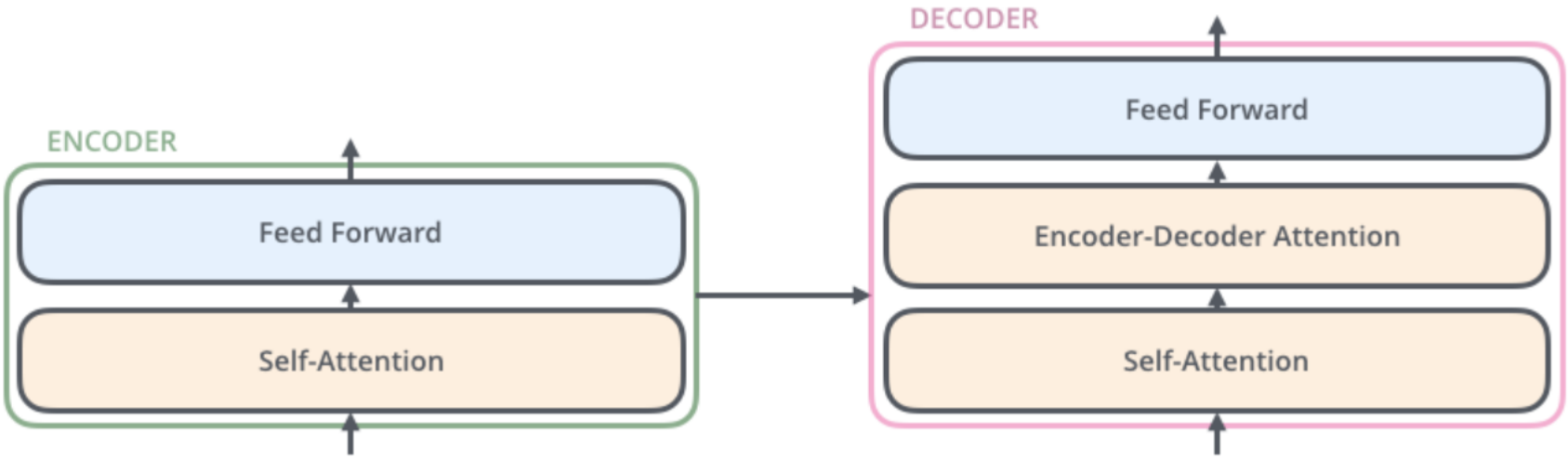
Transformer: encoder

- two layers
- no weight sharing between different encoders
- self-attention helps to focus on relevant part of input

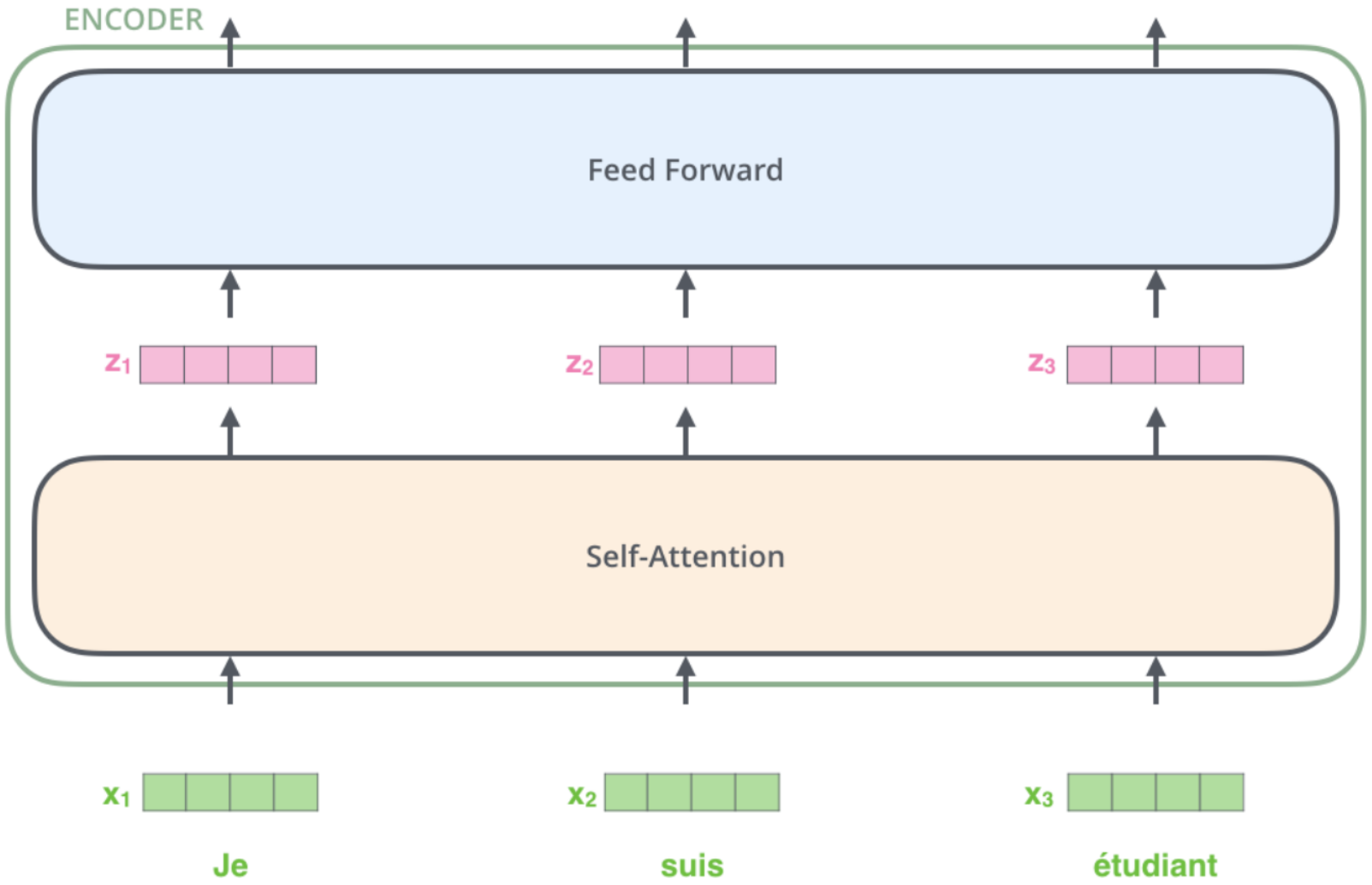


Transformer: decoder

- the same as encoder but with an additional attention layer in between, receiving input fom encoder



Start with embeddings

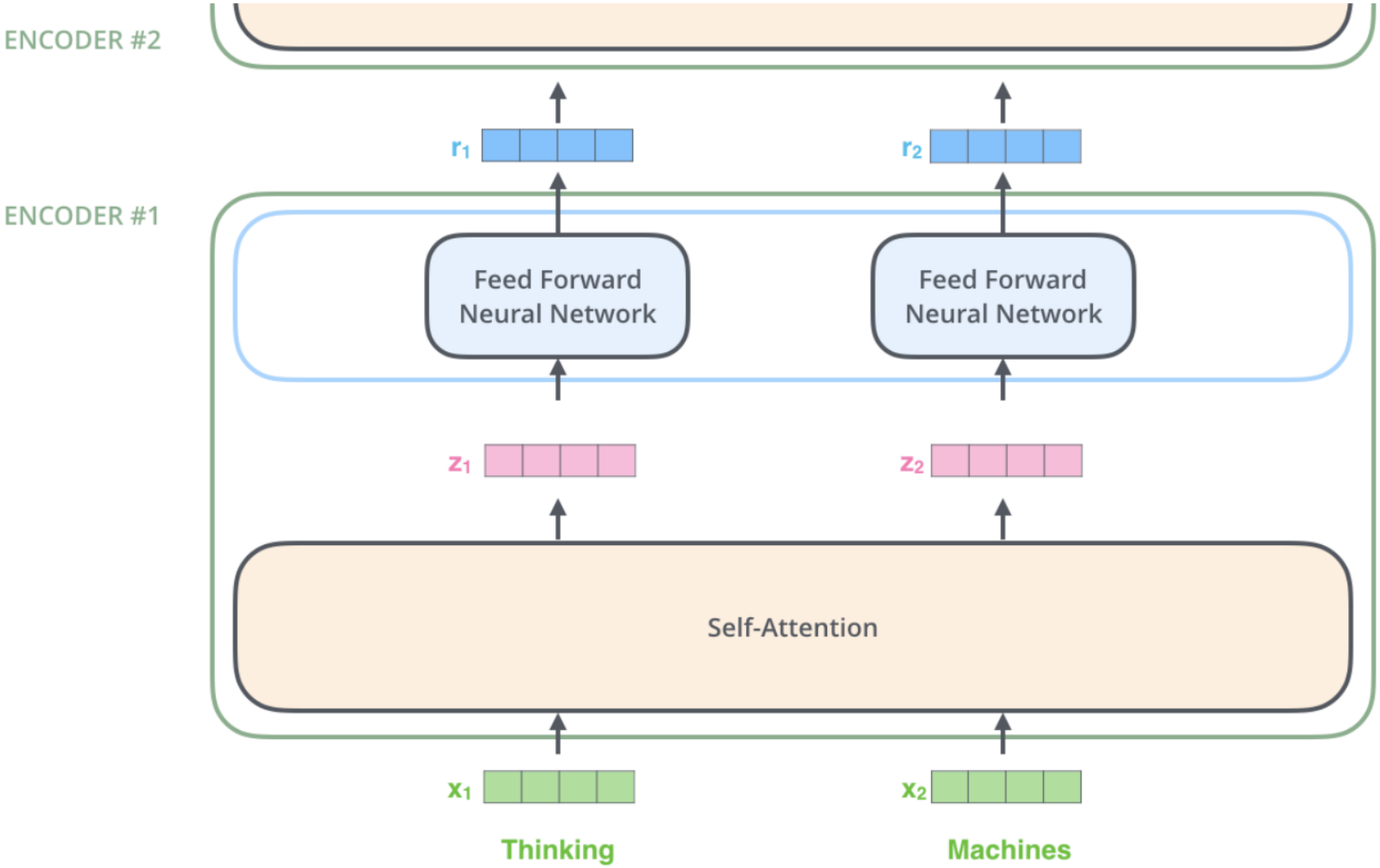


Input to transformer

- embeddings, e.g., 512 dimensional vectors (special, we will discuss that later)
- fixed length, e.g., max 128 tokens (in modern at least 8192)
- dependencies between inputs are only in the self-attention layer, no dependencies in feed-forward layer – good for parallelization

- Let us first present the working of the transformer with an illustration of the prediction

Encoding


ENCODER #2
ENCODER #1
r1
r2
Feed Forward Neural Network
Feed Forward Neural Network
z1
z2
Self-Attention
x1
x2
Thinking
Machines

Self-attention

- As the model processes each word (each position in the input sequence), self-attention allows it to look at other positions in the input sequence for clues that can help lead to a better encoding for this word

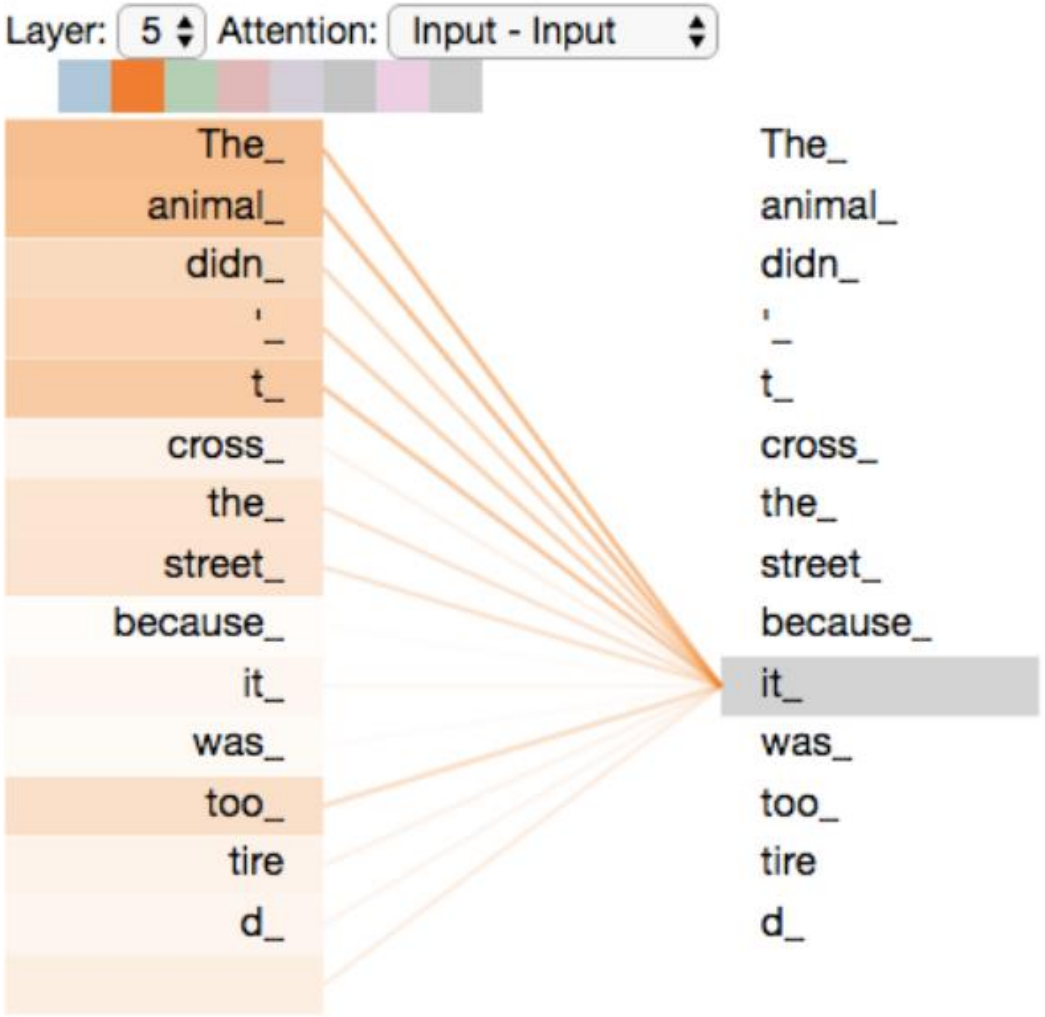
  ”The animal didn't cross the street because it was too tired”

- What does “it” in this sentence refer to? Is it referring to the street or to the animal? It’s a simple question to a human, but not as simple to an algorithm.
- *”The animal didn't cross the street because it was too wide”*

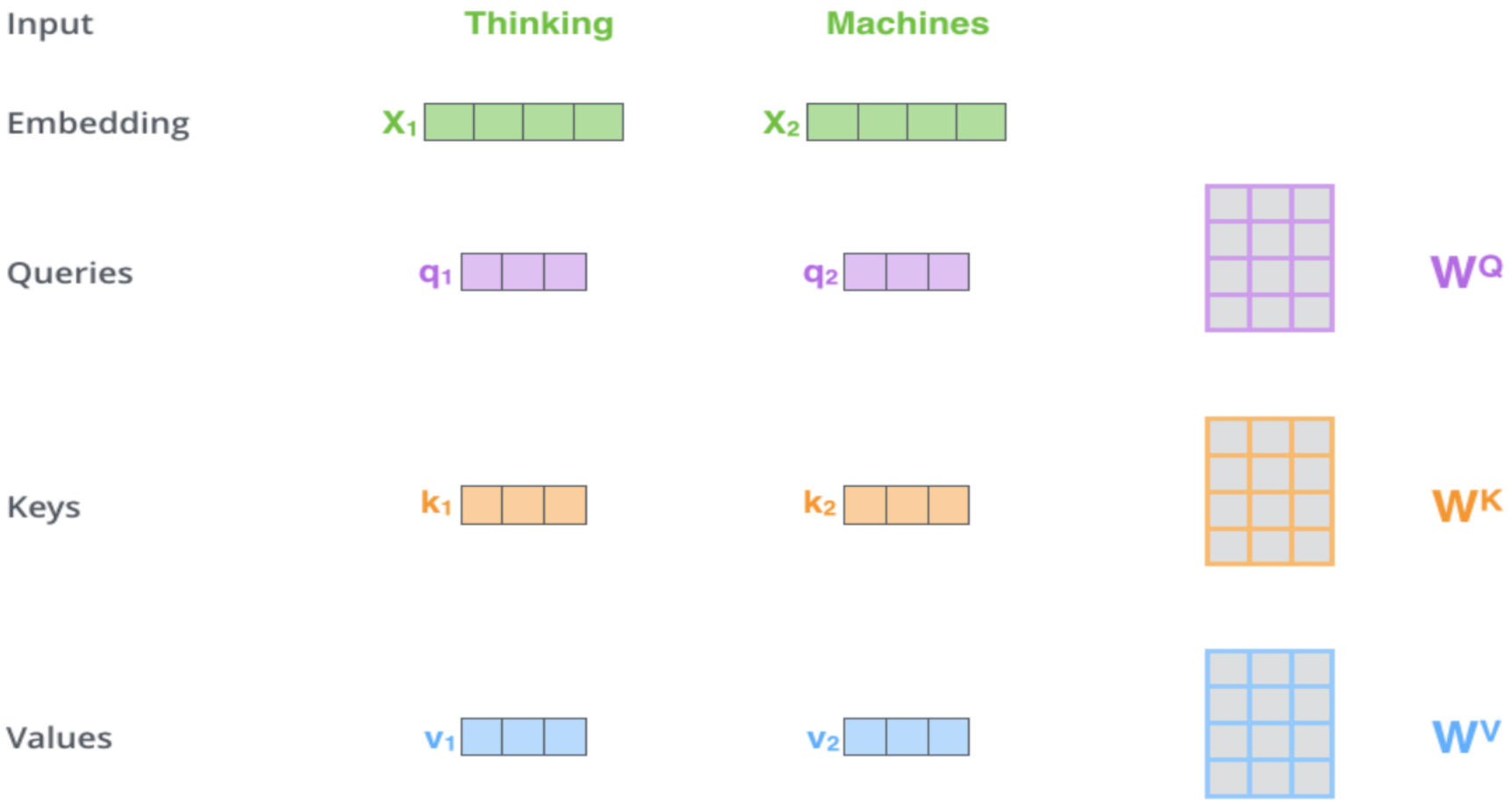
- When the model is processing the word “it”, self-attention allows it to associate “it” with “animal”.

Illustrating self-attention

- As we are encoding the word "it" in encoder #5 (the top encoder in the stack), part of the attention mechanism was focusing on "The Animal", and baked a part of its representation into the encoding of "it".

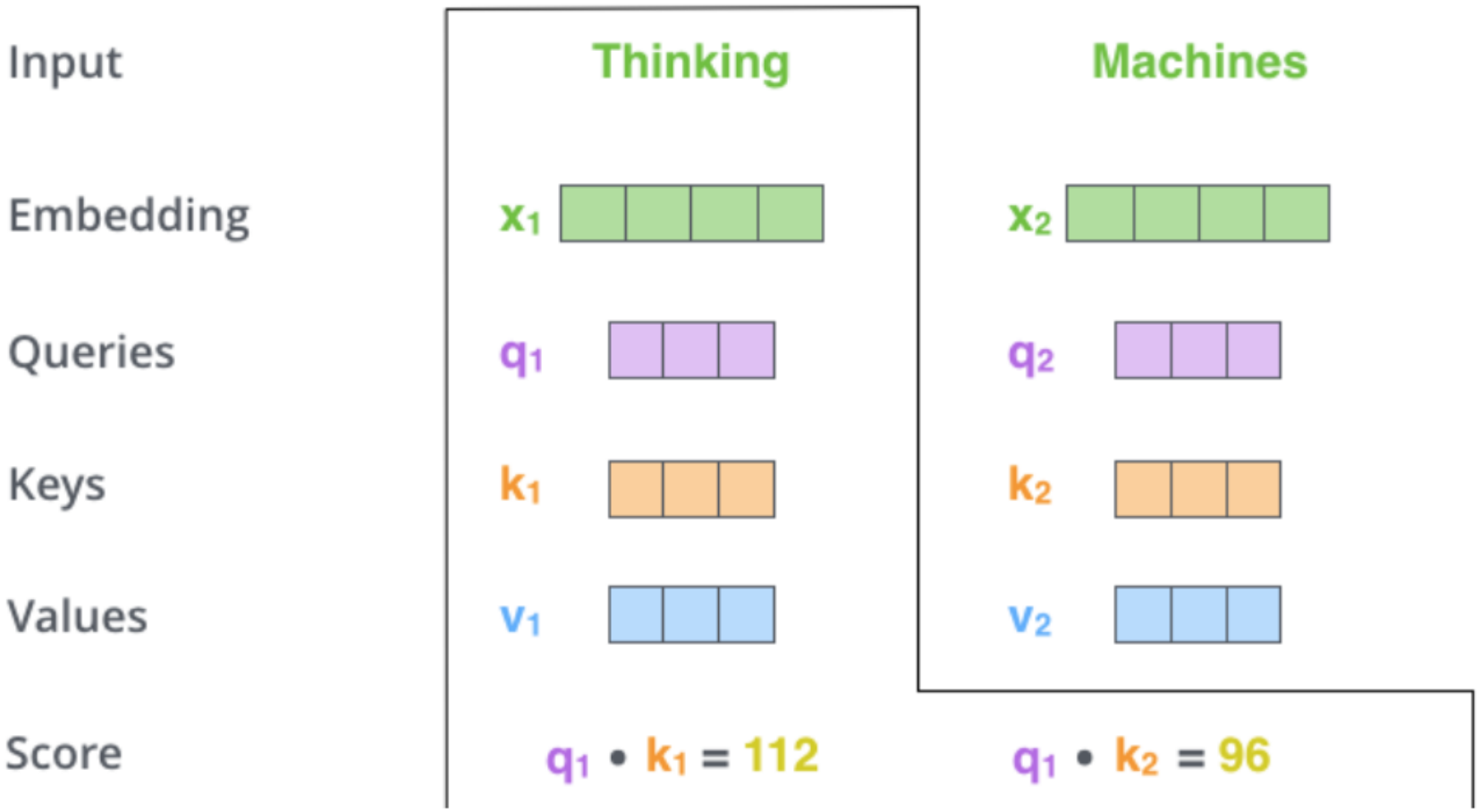


Self-attention details 1/4

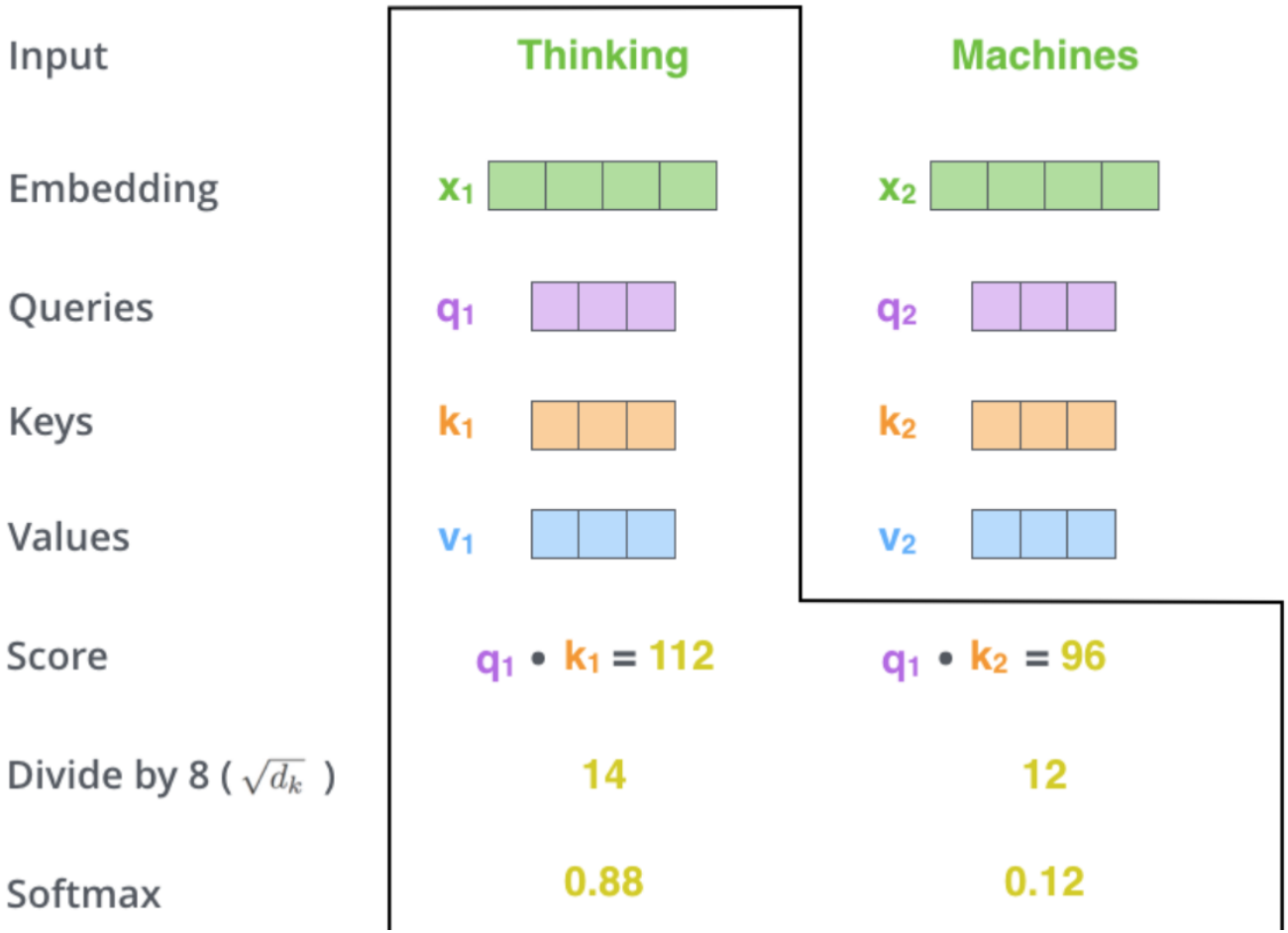


Multiplying x_1 by the W^Q weight matrix produces q_1, the "query" vector associated with that word. We end up creating a "query", a "key", and a "value" projection of each word in the input sentence.

Details 2/4: scoring



Details 3/4: normalization of scores



Details 4/4: self-attention output

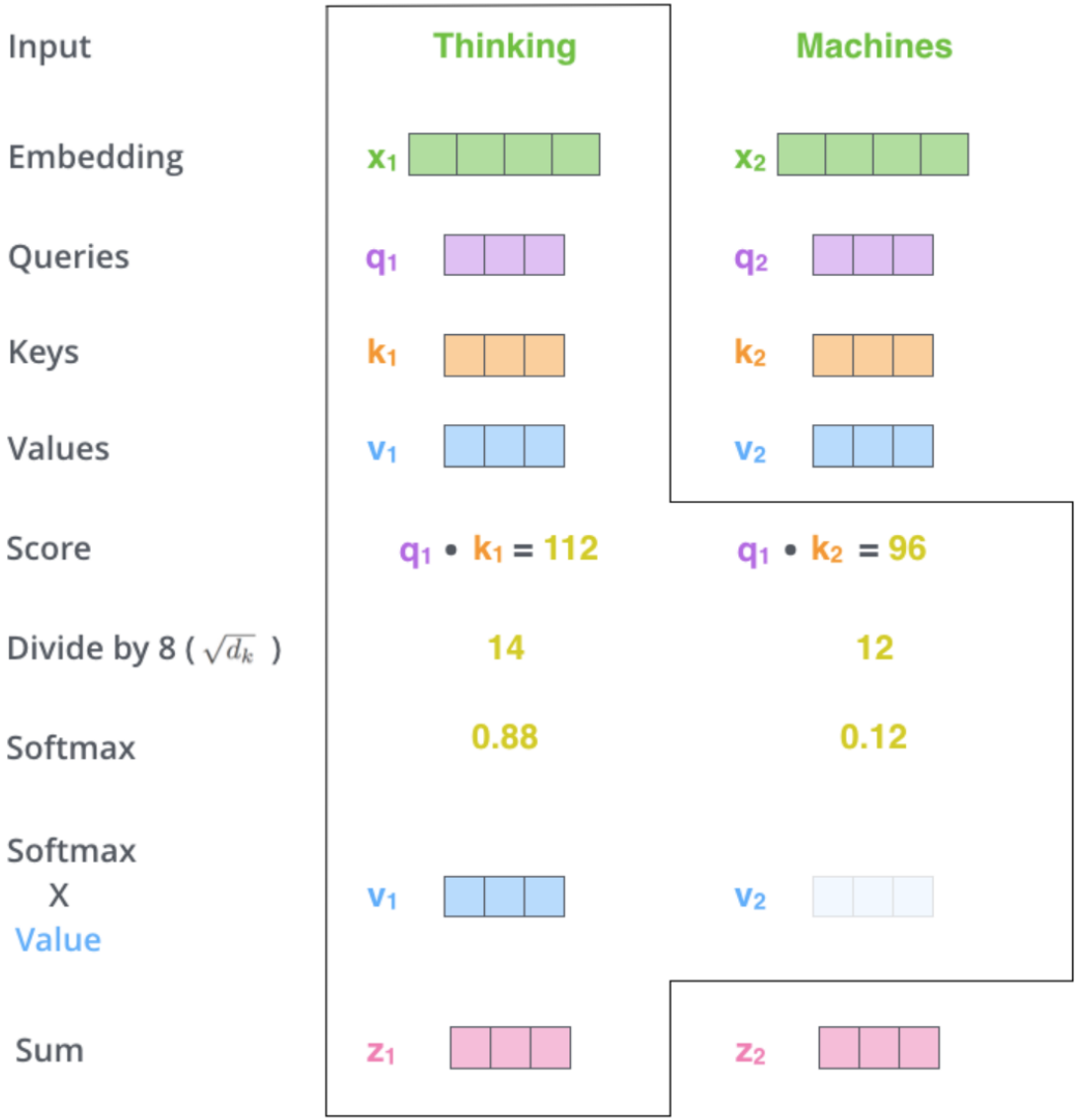
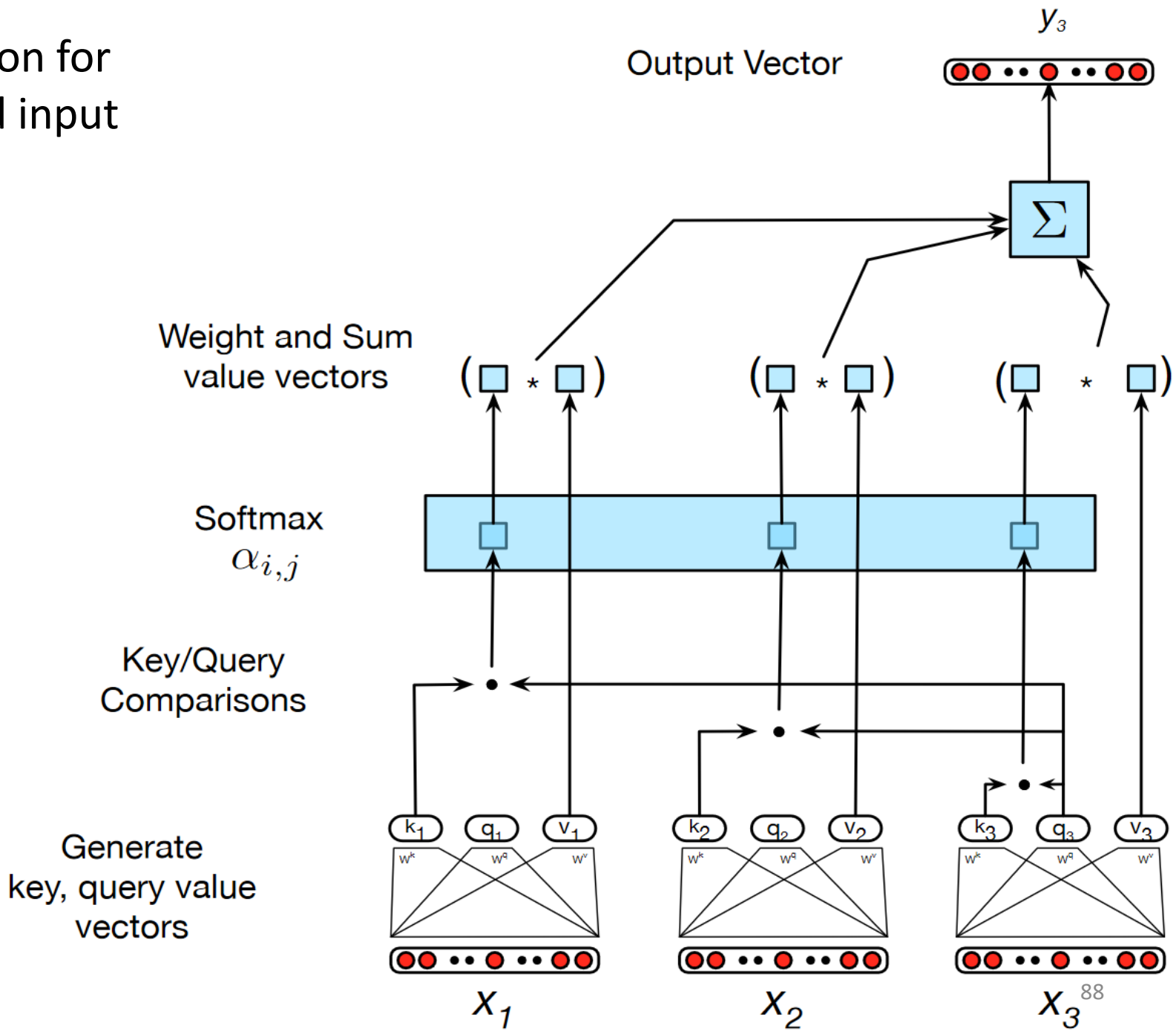
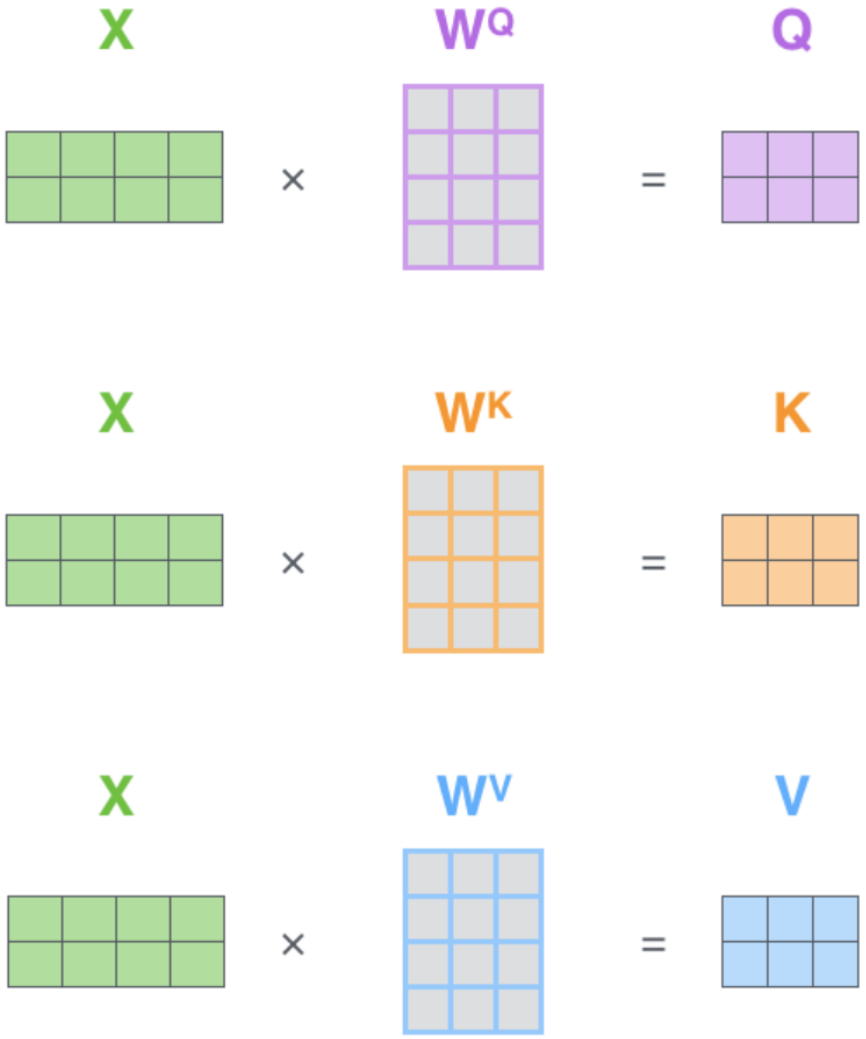


illustration for
the third input



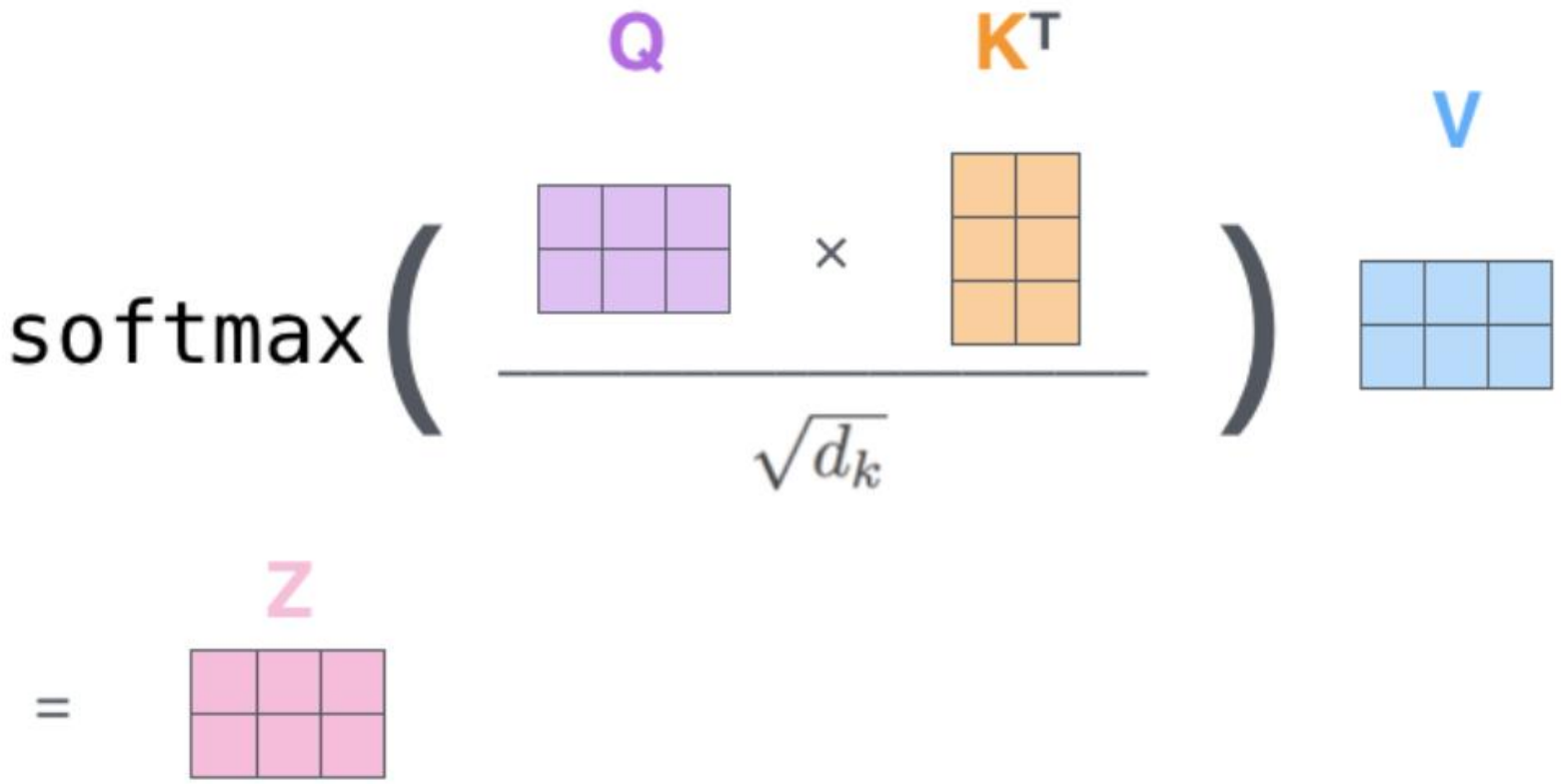
Matrix calculation of self-attention 1/2

Every row in the X matrix corresponds to a word in the input sentence.
The embedding vector x (512) is larger then the q/k/v vectors (64)

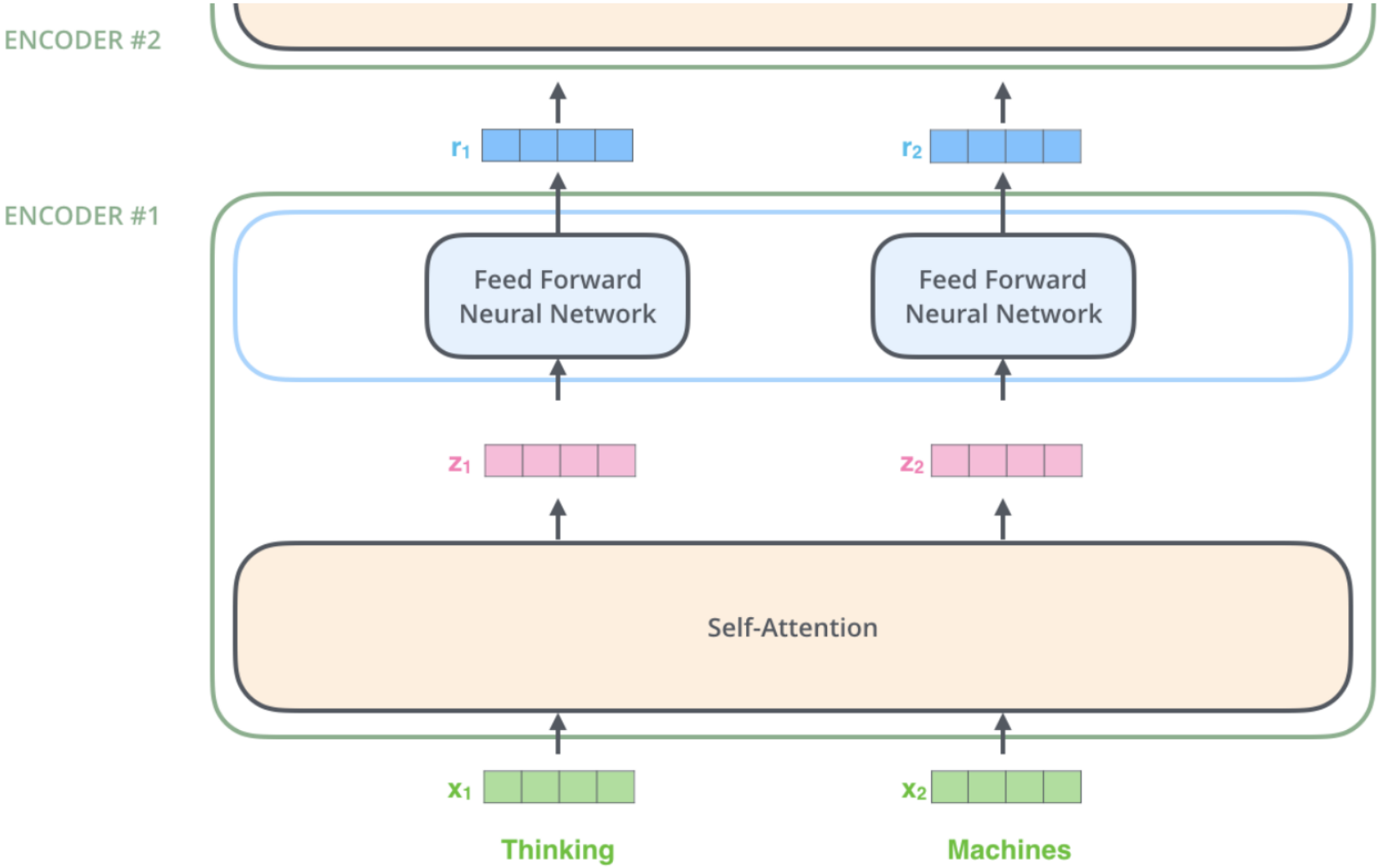


Matrix calculation of self-attention 2/2

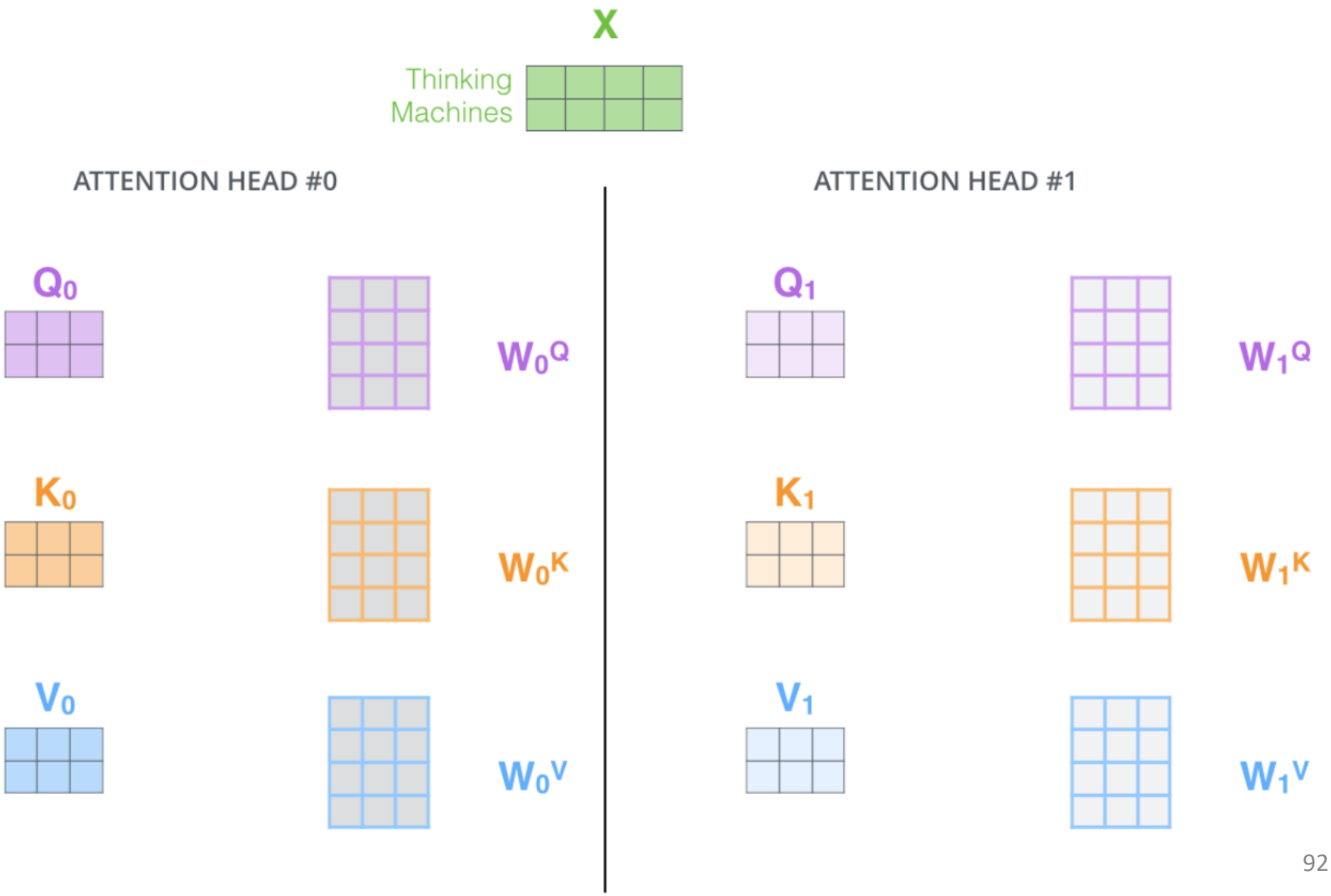
- final calculation



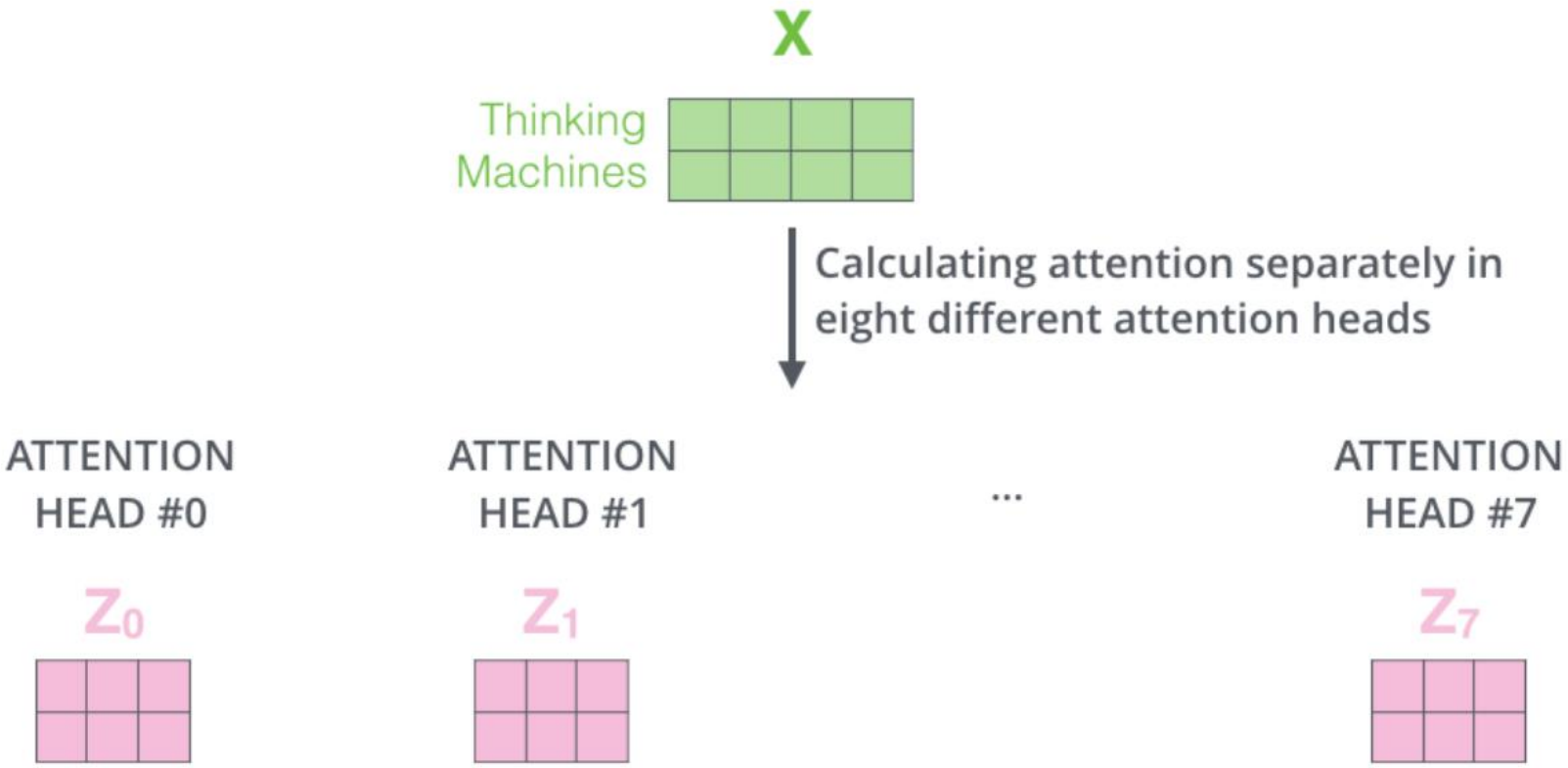
Encoding


ENCODER #2
r1
r2
ENCODER #1
Feed Forward Neural Network
Feed Forward Neural Network
z1
z2
Self-Attention
x1
x2
Thinking
Machines

Example: two attention heads



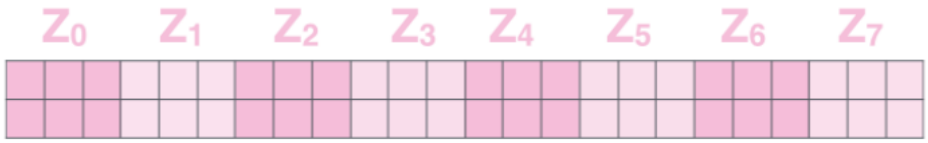
Example: 8 att. heads



- What to do with 8 Z matrices, the feed-forward layer is expecting a single matrix (one vector for each word). We need to condense all attention heads into one matrix.

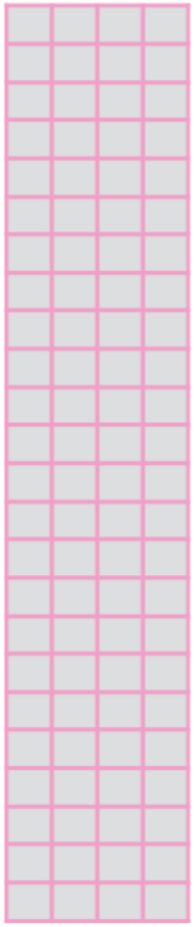
Condensation of attention heads

1) Concatenate all the attention heads



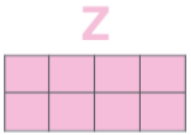
2) Multiply with a weight matrix W^O that was trained jointly with the model

X



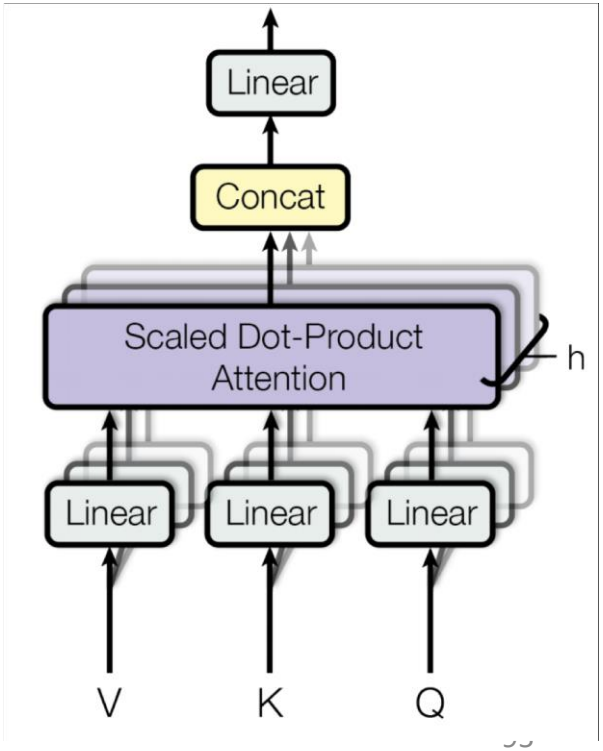
3) The result would be the Z matrix that captures information from all the attention heads. We can send this forward to the FFNN

=



Computing multi-head attention

$$\text{MultiHead}(Q, K, V) = \text{Concat}(\text{head}_1, ..., \text{head}_\text{h})W^O$$
$$\textbf{where } \text{head}_\text{i} = \text{Attention}(QW_i^Q, KW_i^K, VW_i^V)$$



Summary of self-attention

1) This is our input sentence*

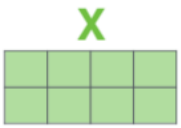
2) We embed each word*

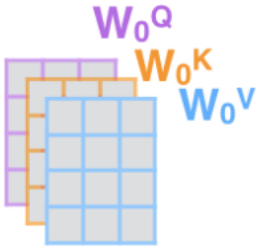
3) Split into 8 heads. We multiply X or R with weight matrices

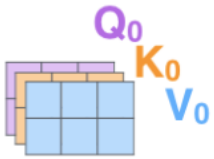
4) Calculate attention using the resulting Q/K/V matrices

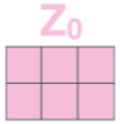
5) Concatenate the resulting Z matrices, then multiply with weight matrix W^O to produce the output of the layer

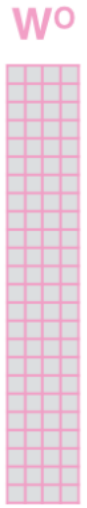
Thinking Machines

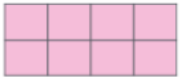


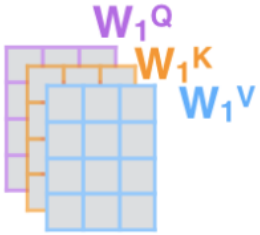


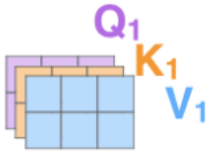


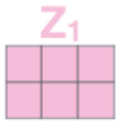




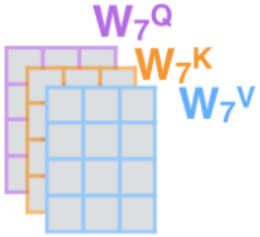


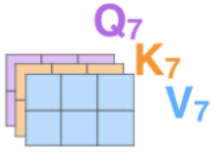






* In all encoders other than #0, we don't need embedding. We start directly with the output of the encoder right below this one





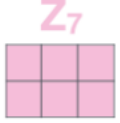


Illustration of self-attention: 1 head

- encoder #5 (the top encoder in the stack)
- As we encode the word "it", one attention head is focusing most on "the animal", while another is focusing on "tired" -- in a sense, the model's representation of the word "it" bakes in some of the representation of both "animal" and "tired".

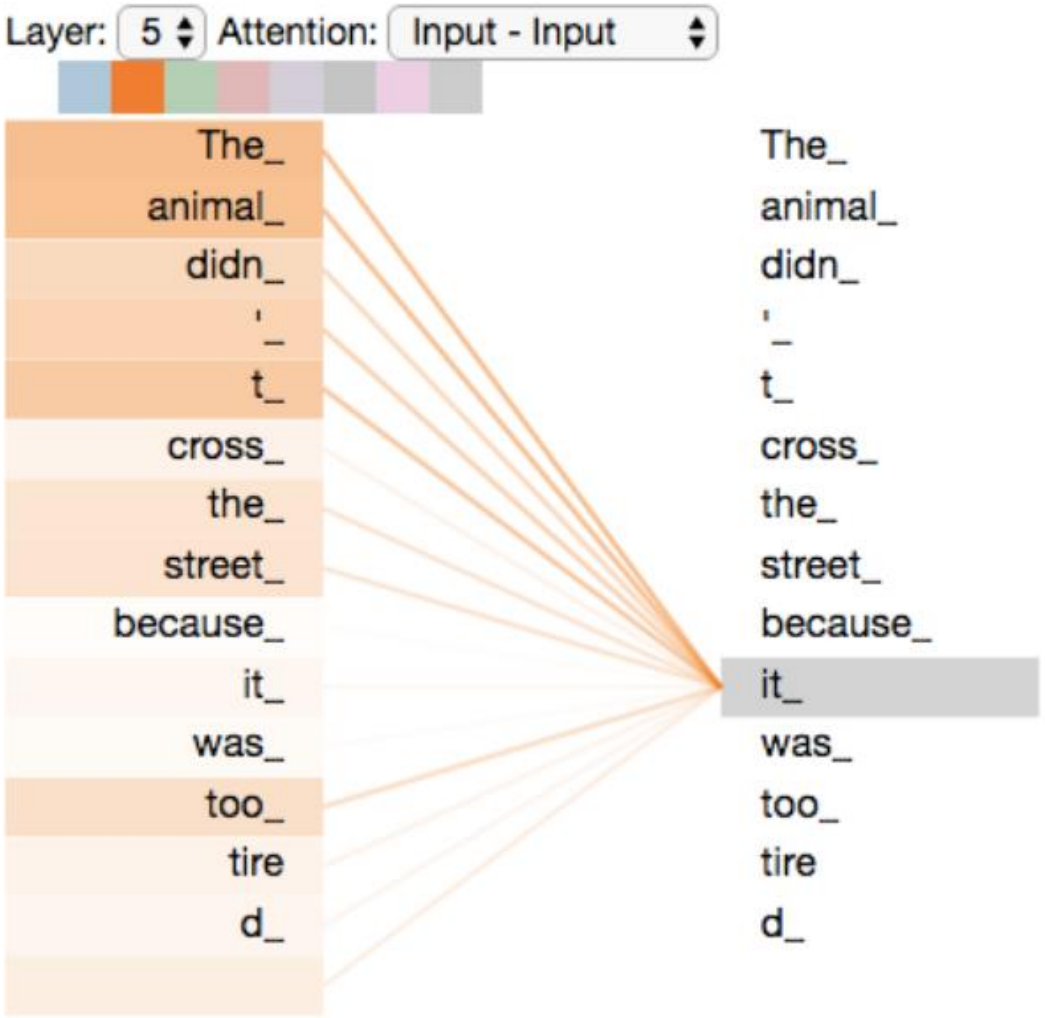
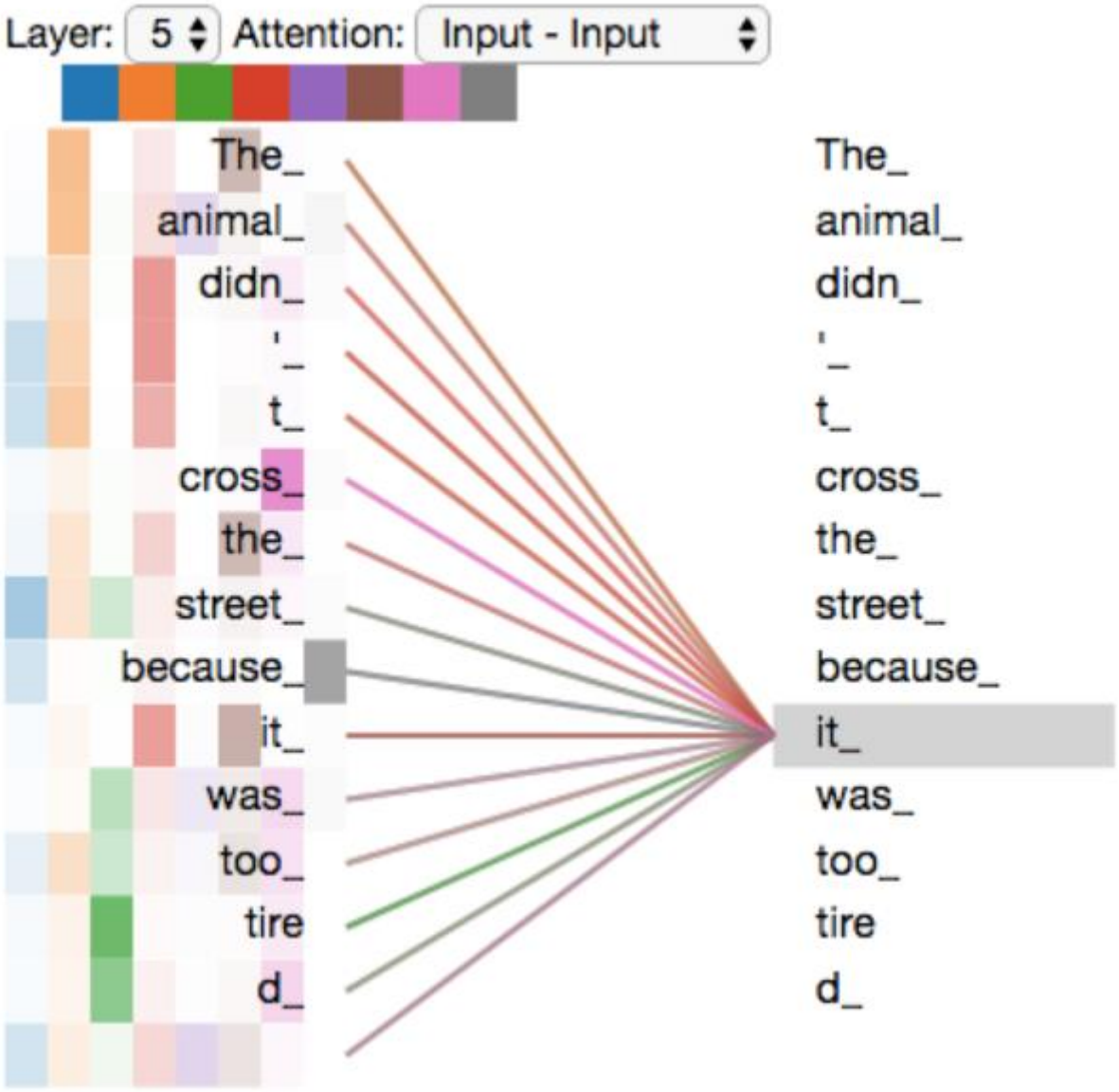
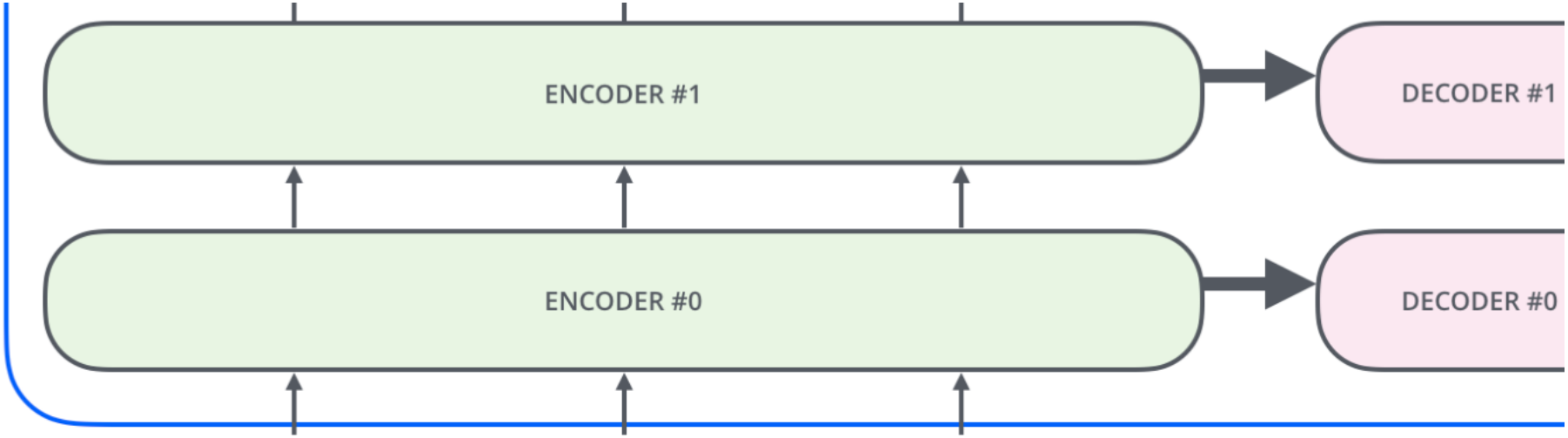


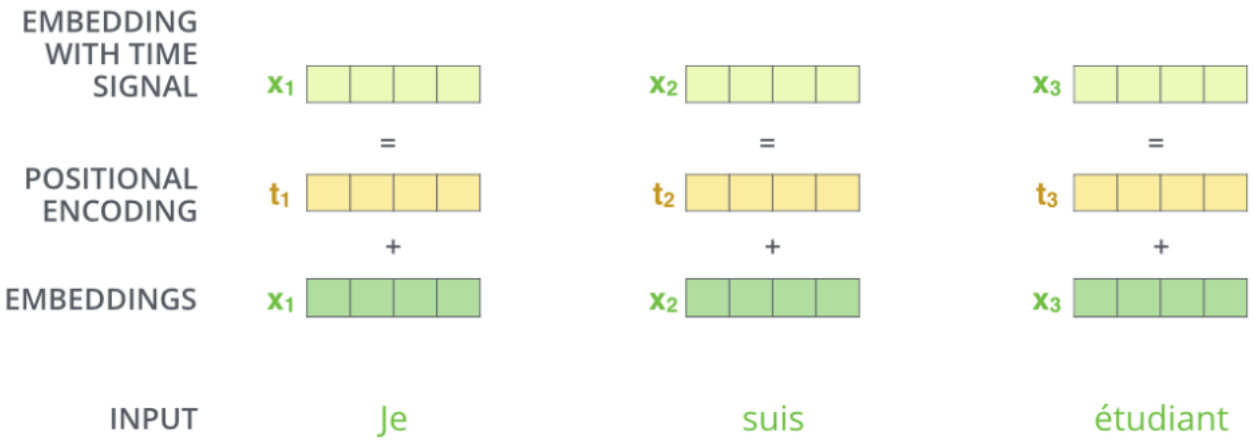
Illustration of self-attention: all heads

- all the attention heads in one picture are harder to interpret



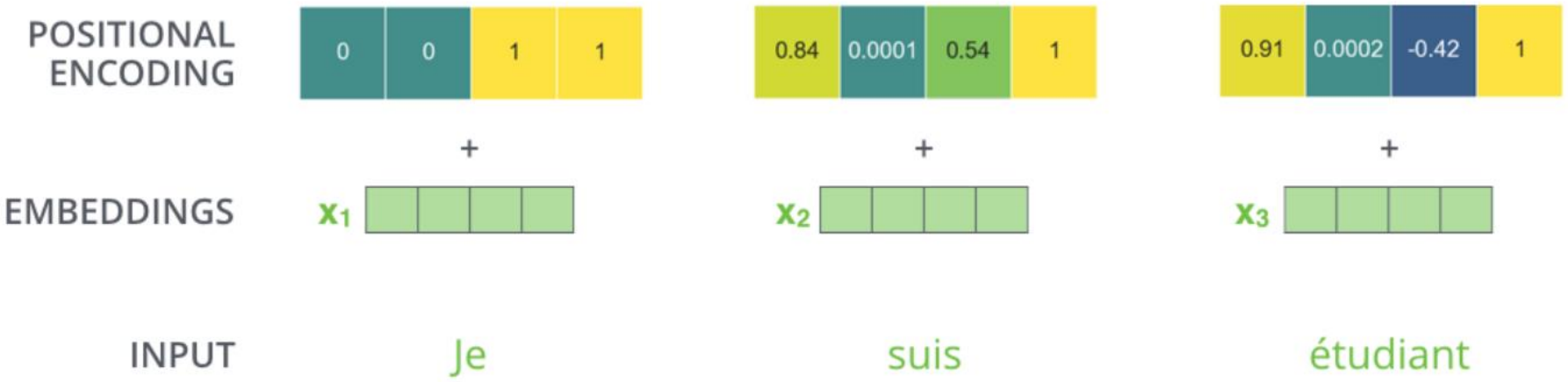
Adding position encoding



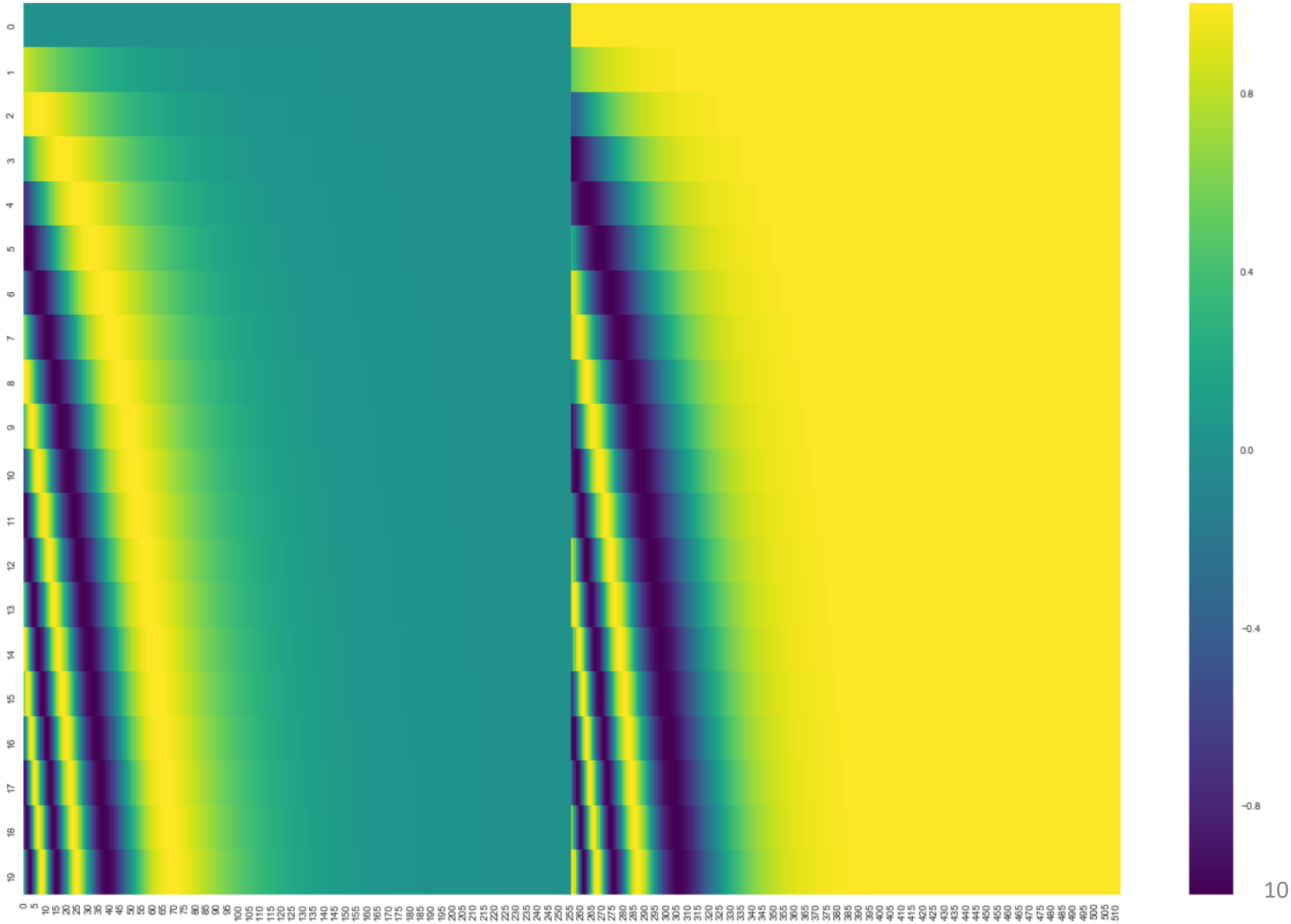


Example: encoding position

- the values of positional encoding vectors follow a specific pattern.



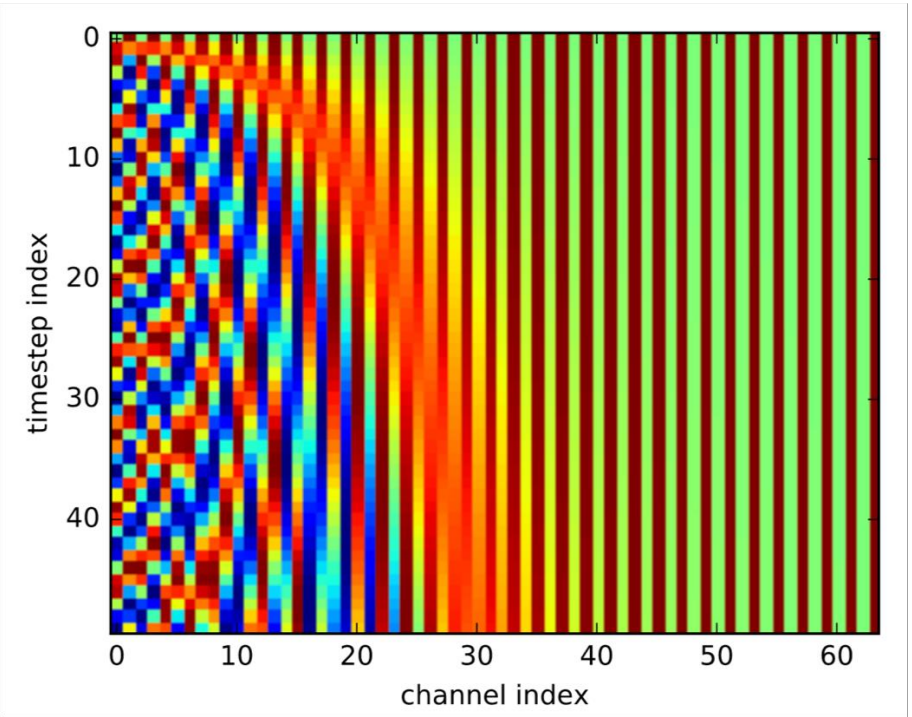
Example of positional encoding



Another positional encoding

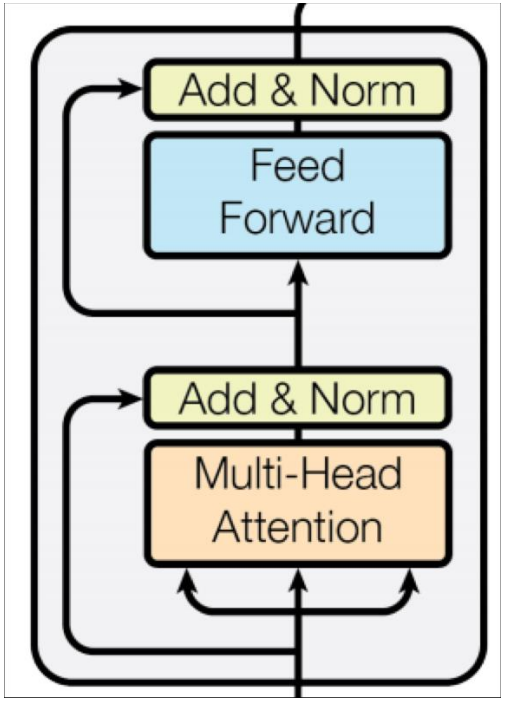
$$PE_{(pos,2i)} = sin(pos/10000^{2i/d_{\text{model}}})$$

$$PE_{(pos,2i+1)} = cos(pos/10000^{2i/d_{\text{model}}})$$

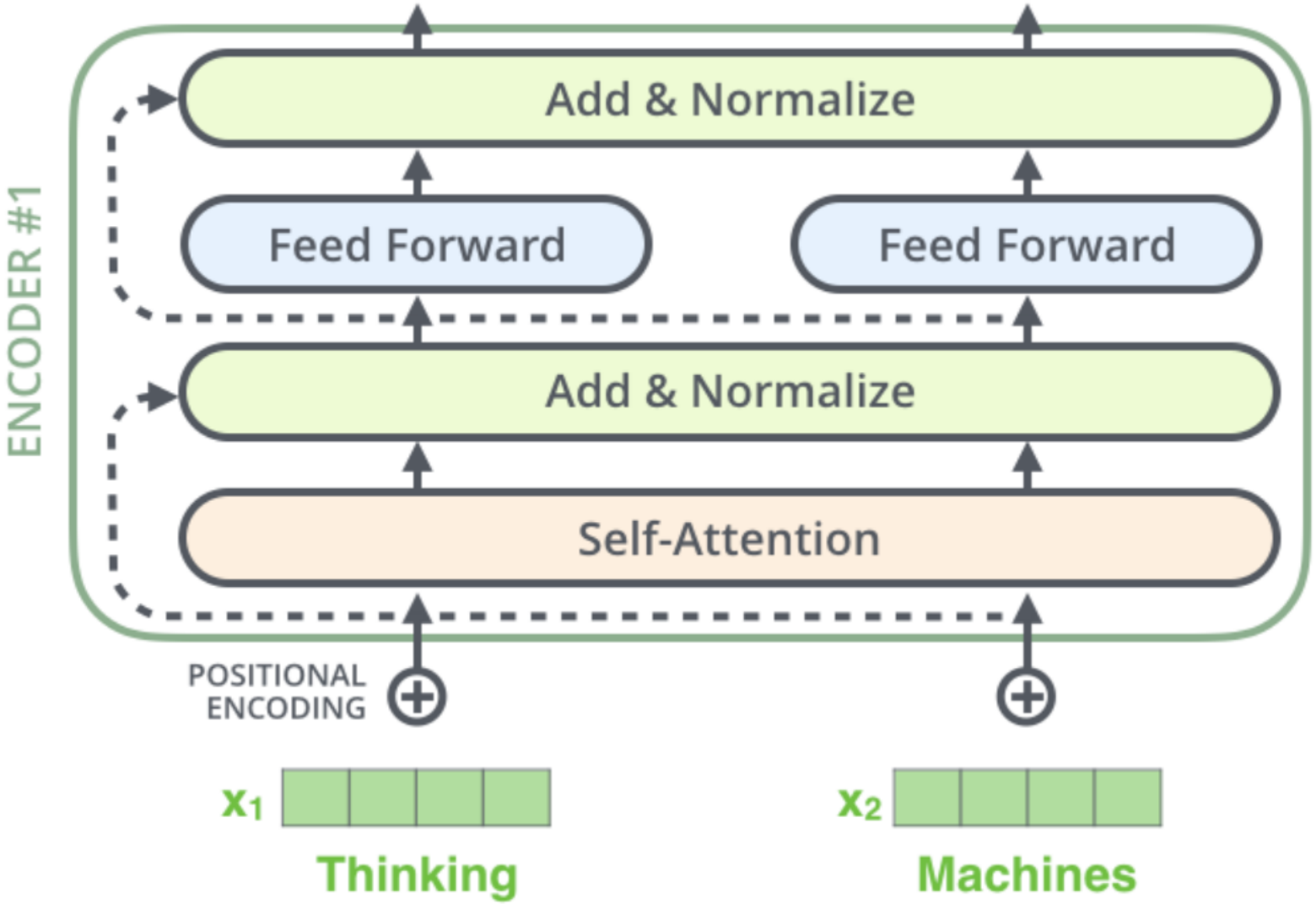


Encoder blocks

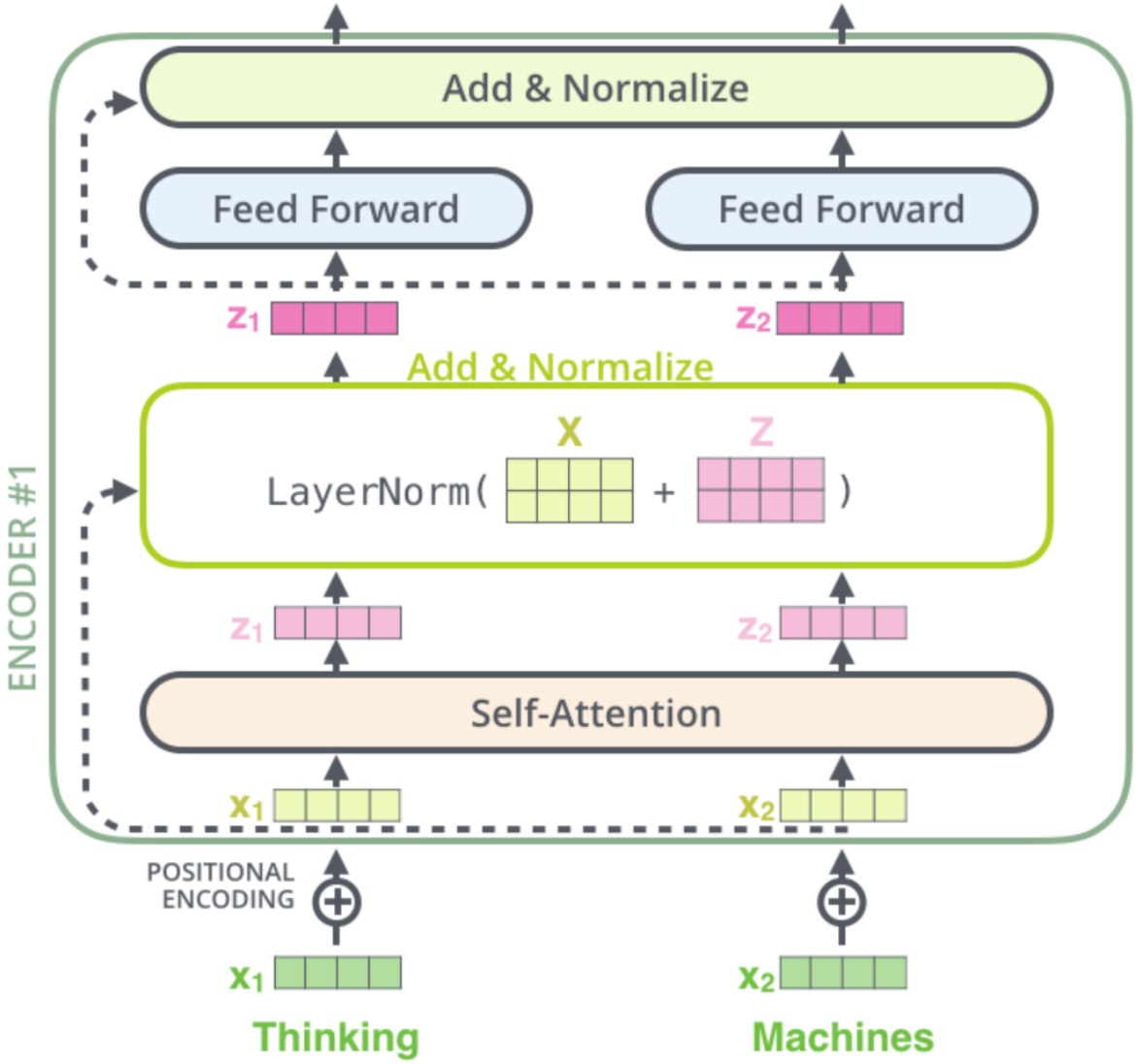
- Each block has two “sublayers”
  - Multihead attention
  - 2-layer feed-forward neural network (with ReLU)
- Each of these two steps also has a residual (short-circuit) connection and LayerNorm, i.e.:
  - LayerNorm(x + Sublayer(x))



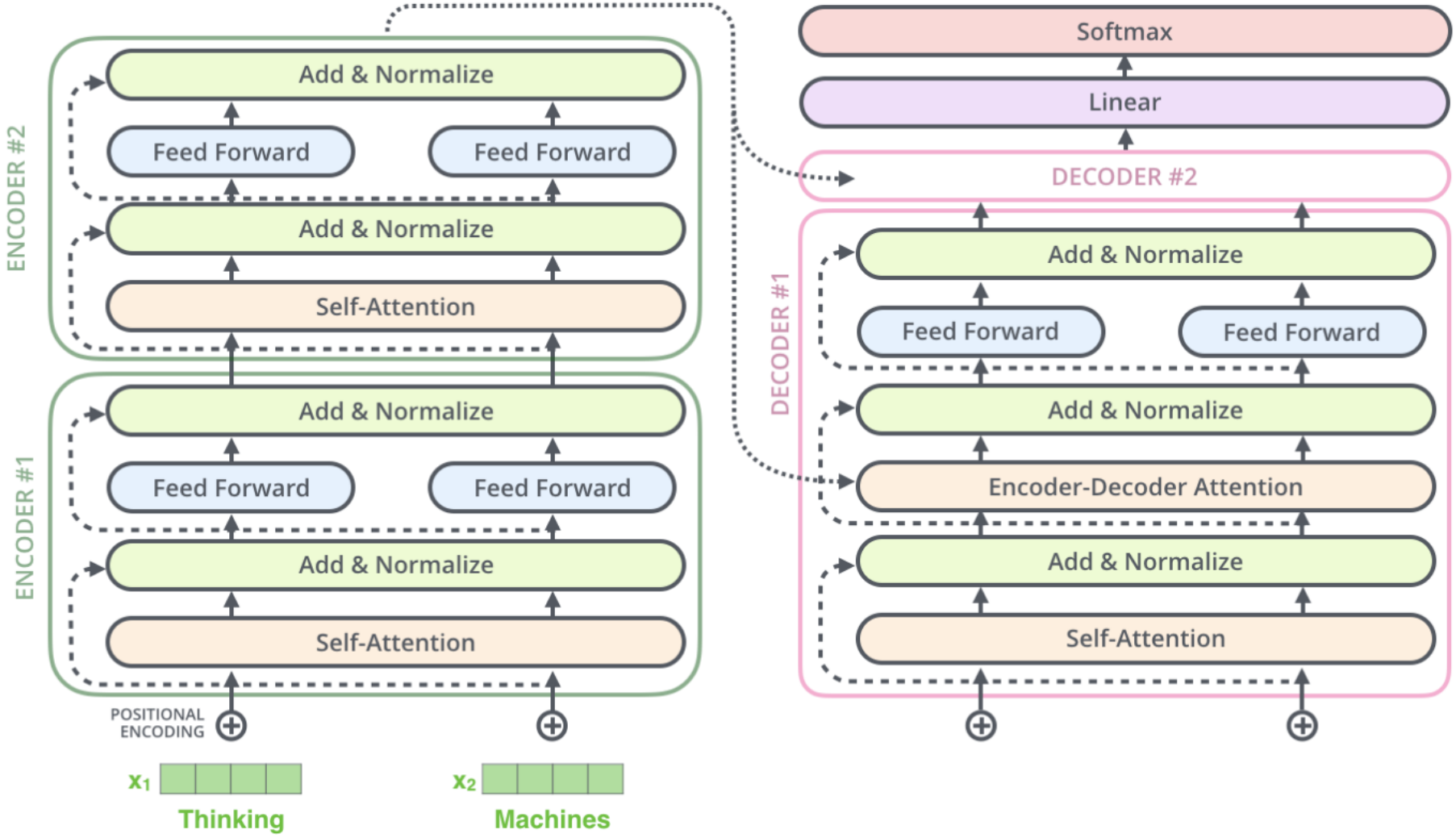
Architecture with residual connection – top level view



Architecture with residual connection – example

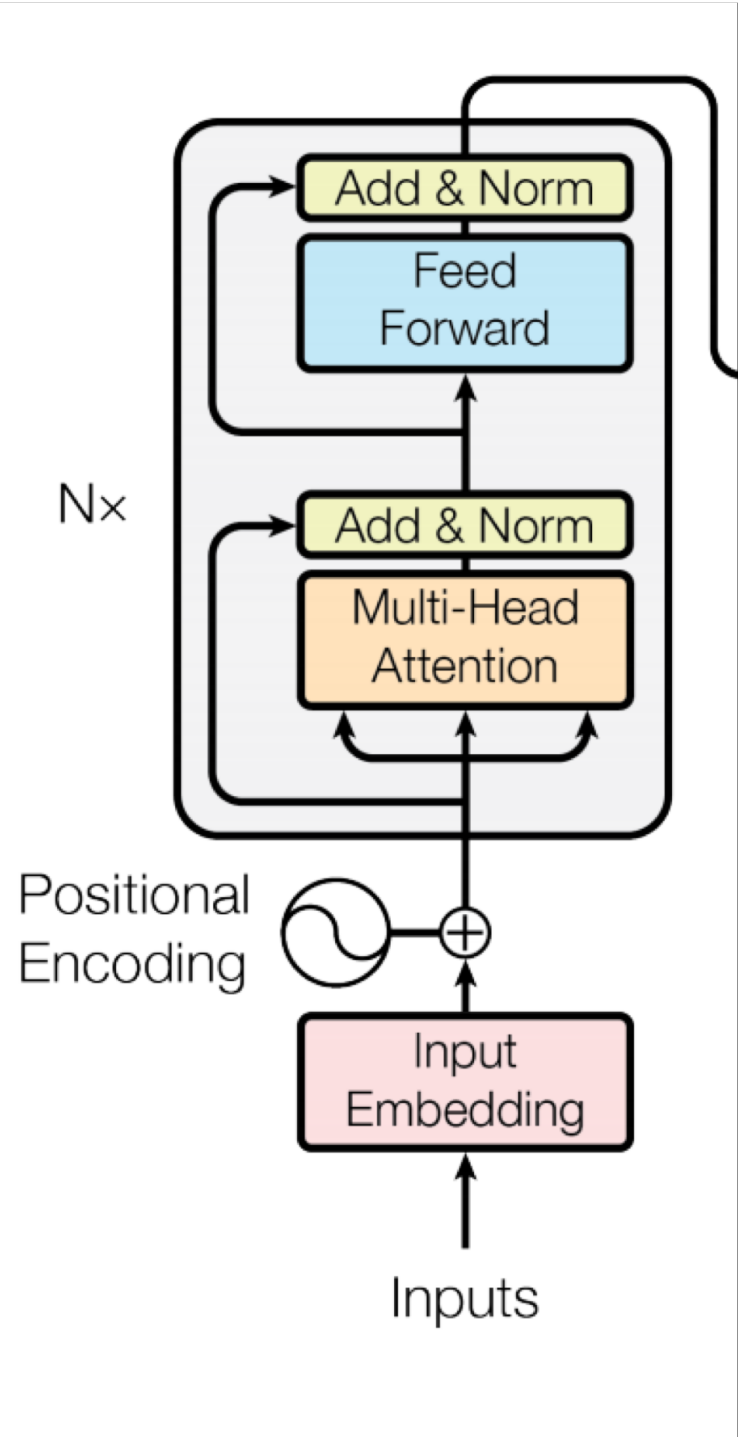


Example: 2 stacked transformer



Complete encoder

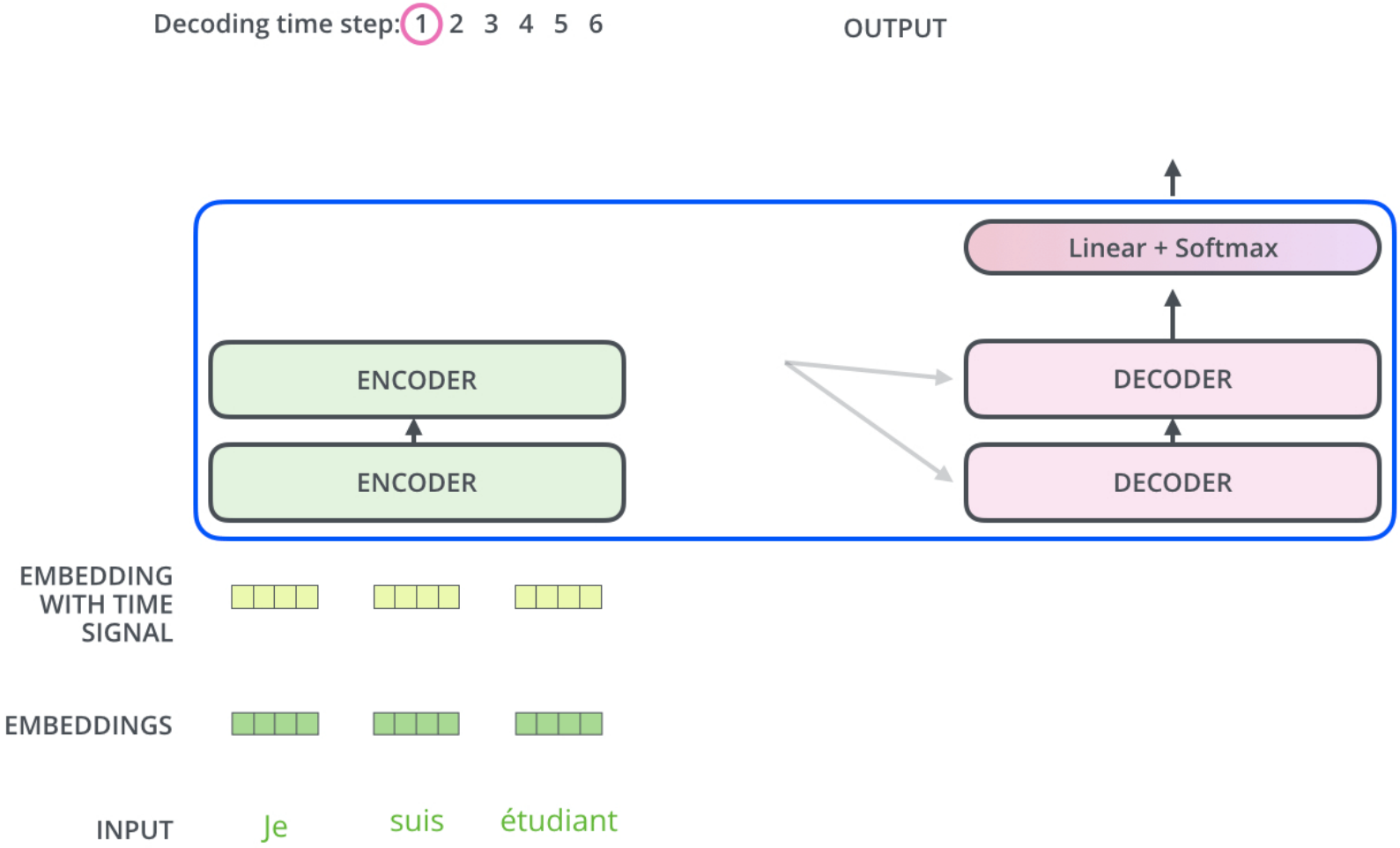
- each encoder block is repeated several times, e.g., 6 times



Decoder

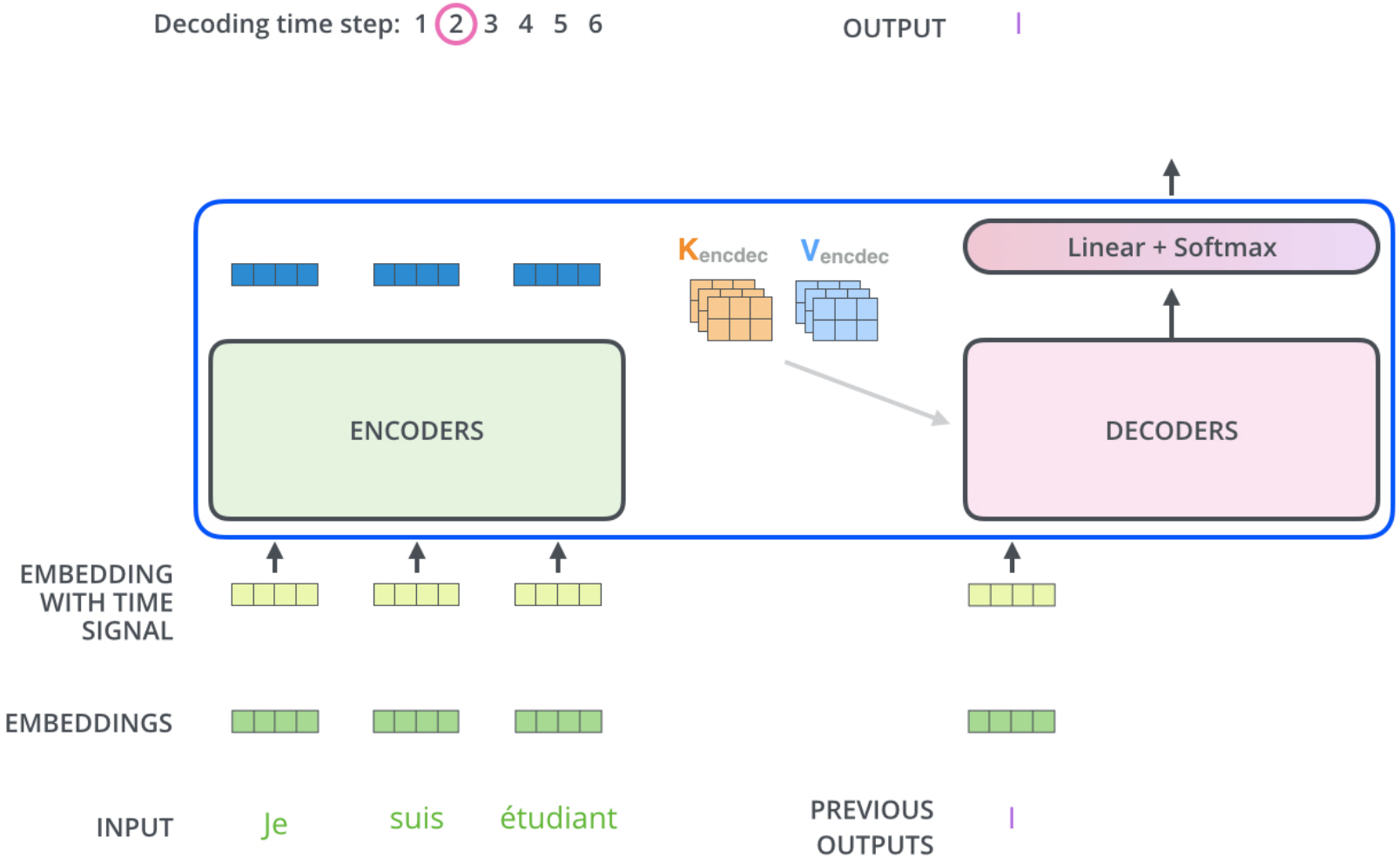
- Decoders have the same components as encoders
- An encoder starts by processing the input sequence.
- The output of the top encoder is transformed into a set of attention vectors K and V.
- These are used by each decoder in its “encoder-decoder attention” layer which helps the decoder to focus on appropriate places in the input sequence.

Encoder-decoder in action 1/2



After finishing the encoding phase, we begin the decoding phase. Each step in the decoding phase outputs an element from the output sequence (the English translation sentence in this case).

Encoder-decoder in action 2/2



The steps repeat until a special symbol indicating the end of output is generated. The output of each step is fed to the bottom decoder in the next time step. We add positional encoding to decoder inputs to indicate the position of each word.

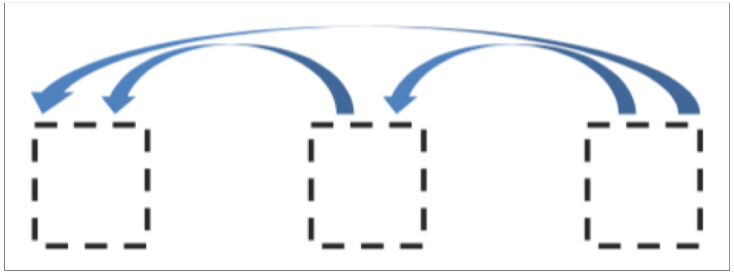
Self-attention and encoder-decoder attention in the decoder

- In the decoder, the self-attention layer is only allowed to attend to itself and earlier positions in the output sequence (to maintain the autoregressive property).
- This is done by masking future positions (setting them to -inf) before the softmax step in the self-attention calculation.

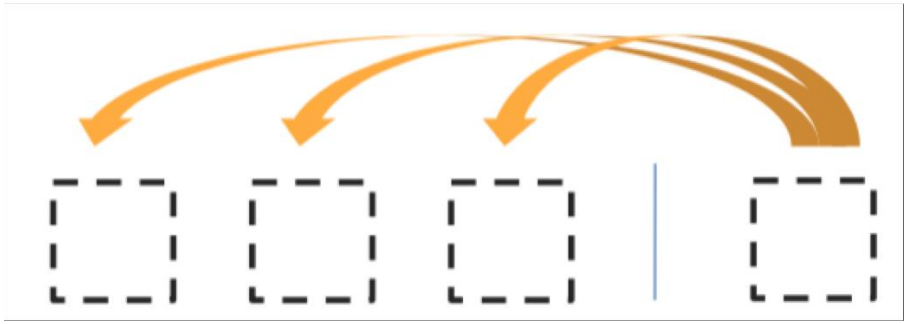
- The “Encoder-Decoder Attention” layer works just like multiheaded self-attention, except it creates its Q (queries) matrix from the layer below it, and takes the K (keys) and V (values) matrix from the output of the encoder stack.

Attentions in the decoder

1. Masked decoder self-attention on previously generated outputs

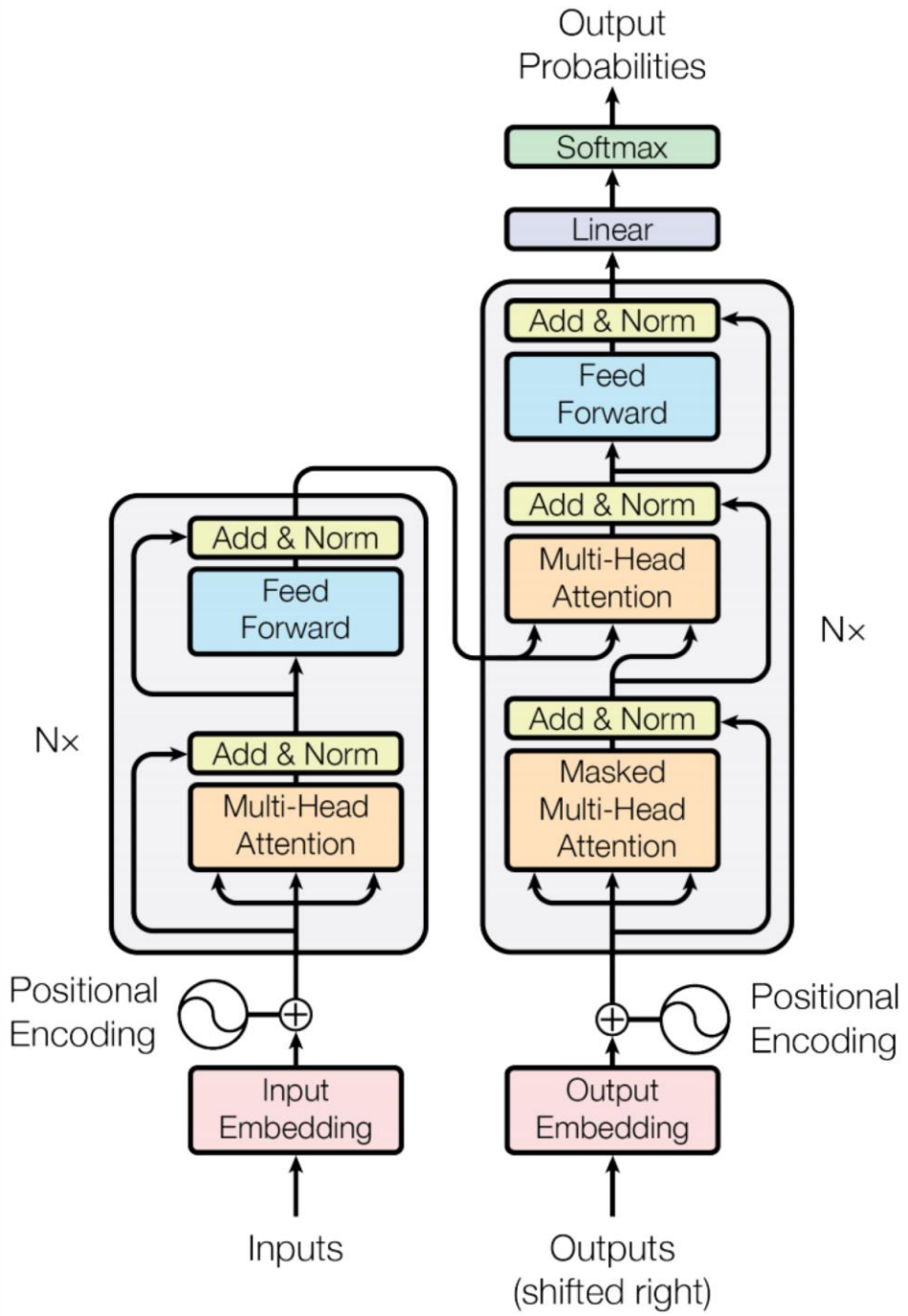


2. Encoder-Decoder Attention, where queries come from previous decoder layer and keys and values come from output of the encoder



Animated workings of attention in transformer

One encoder-decoder block



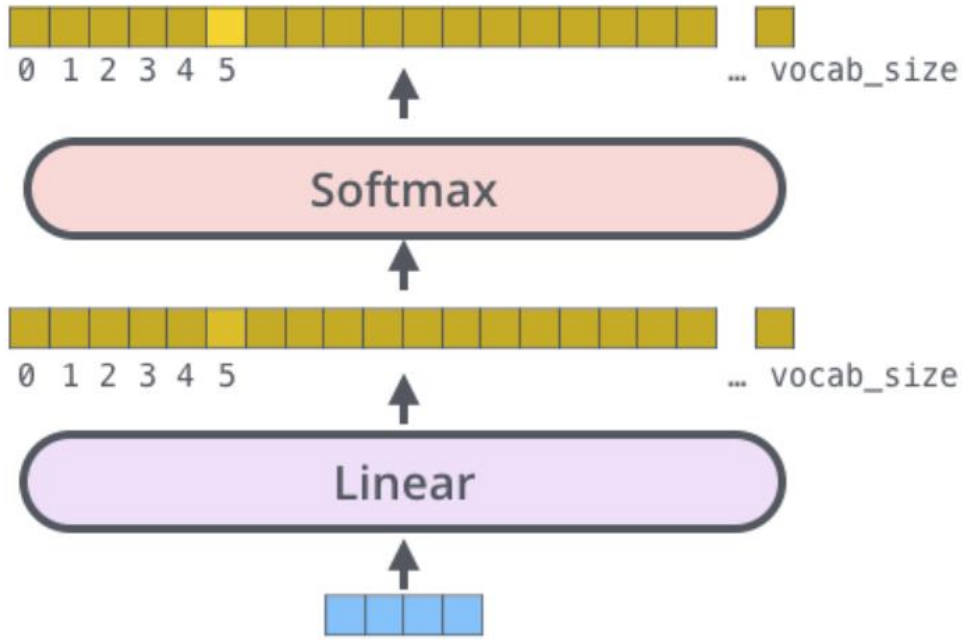
Producing the output words

Which word in our vocabulary is associated with this index?

am

Get the index of the cell with the highest value (**argmax**)

5



Training the transformer

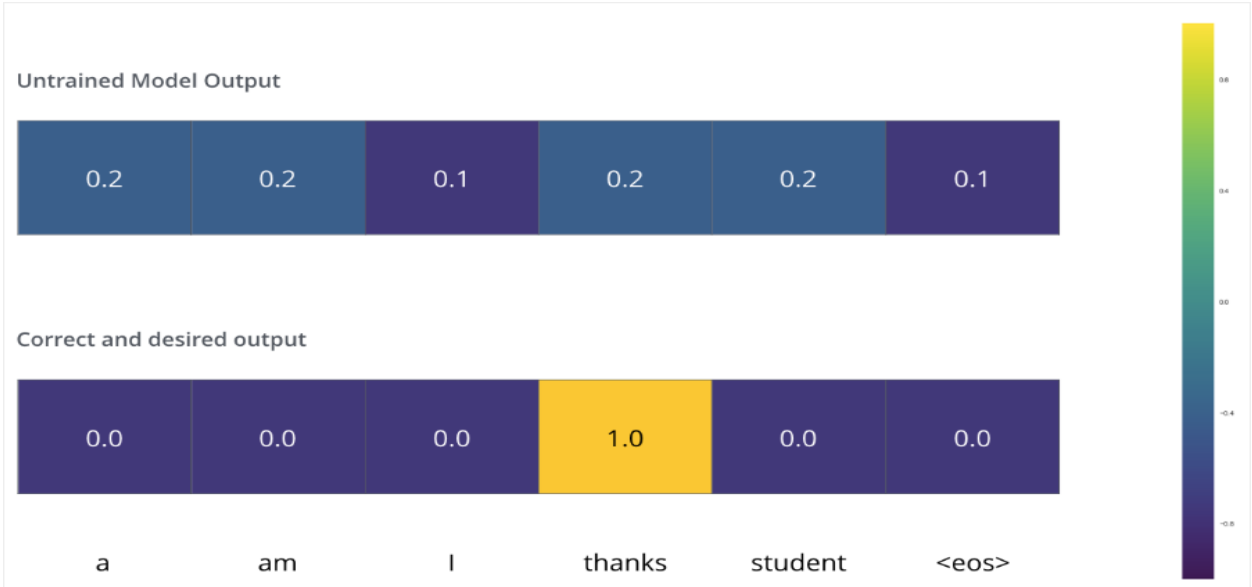
- During training, an untrained model would go through the exactly the same forward pass. But since we are training it on a labeled training dataset, we can compare its output with the actual correct output.
- For illustration, let's assume that our output vocabulary only contains six words(a, am, i, thanks, student, <eos>)
- The input is typically in the order of 10^4 (e.g., 30 000)

Output Vocabulary

WORD	a	am	I	thanks	student	<eos>
INDEX	0	1	2	3	4	5

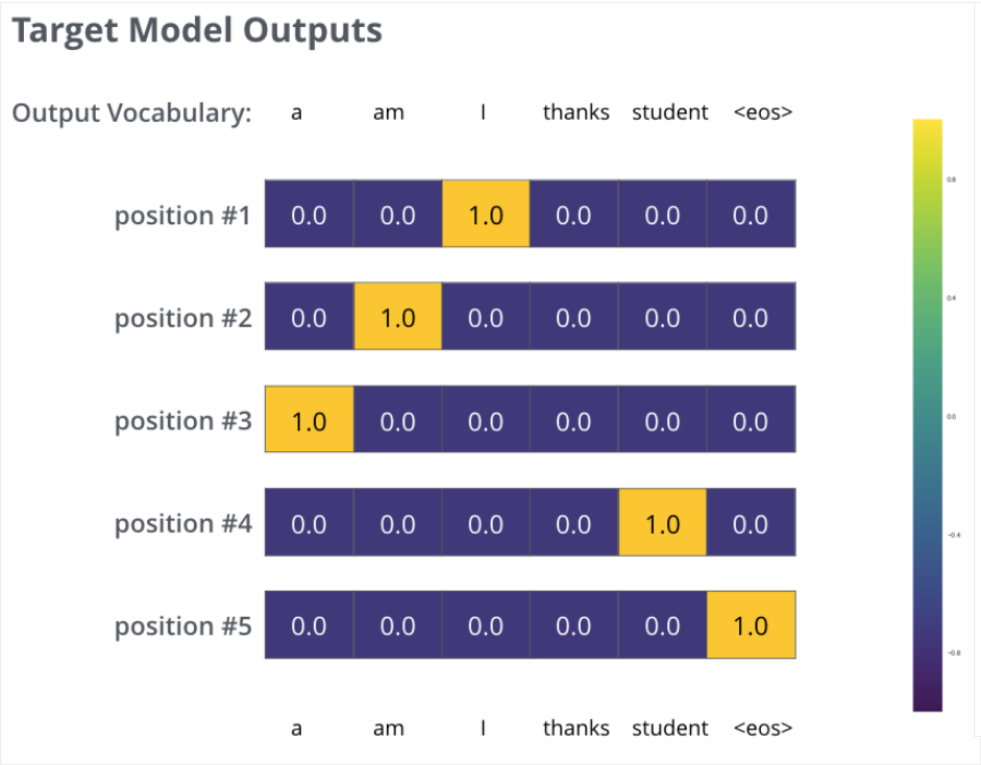
The Loss Function

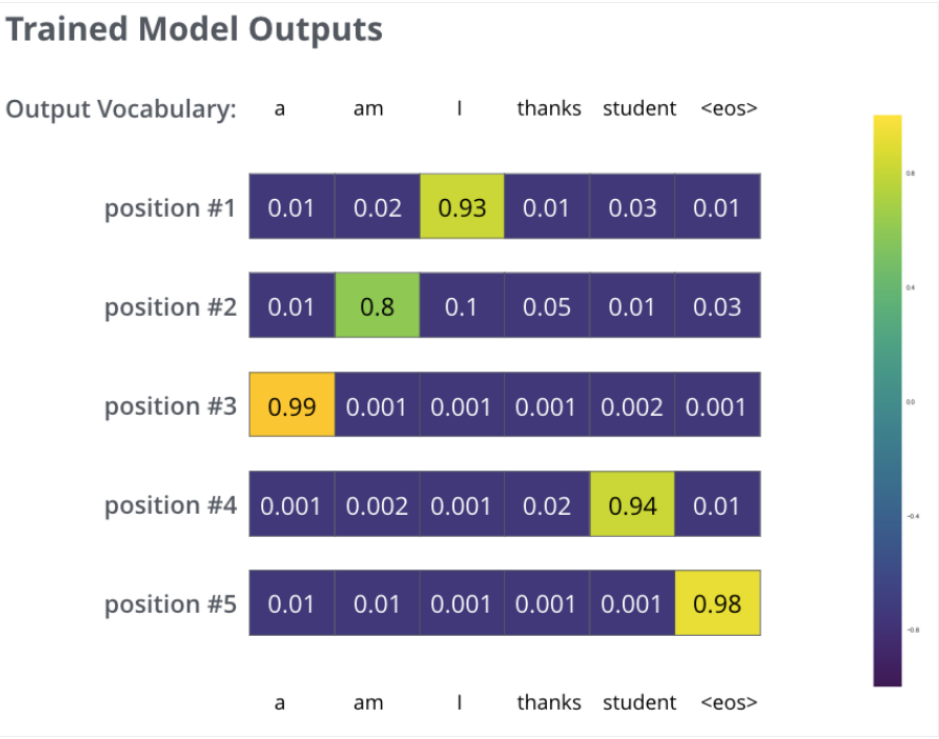
- evaluates the difference between the true output and the returned output
- transformer typically uses cross-entropy or Kullback–Leibler divergence.
- The model output is a probability distribution, the true output is 1-hot encoded, e.g.,



Loss evaluation for sequences

- loss function has to be evaluated for the whole sentence, not just a single word
- transformers use greedy decoding or beam search





Three flavours of transformers

- encoder only (BERT)
- encoder-decoder (machine translation, T5)
- decoder only (GPT)

BERT

- combines several tasks
- predicts masked words in a sentence
- also predicts order of sentences: is sentence A followed by sentence B or not
- combines several hidden layers of the network
- uses transformer neural architecture, only teh encoder part
- uses several fine tuned parameters
- multilingual variant supports 104 languages by training on Wikipedia
- publicly available

Devlin J, Chang MW, Lee K, Toutanova K (2019) BERT: Pre-training of deep bidirectional transformers for language understanding. In: Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), pp 4171–4186

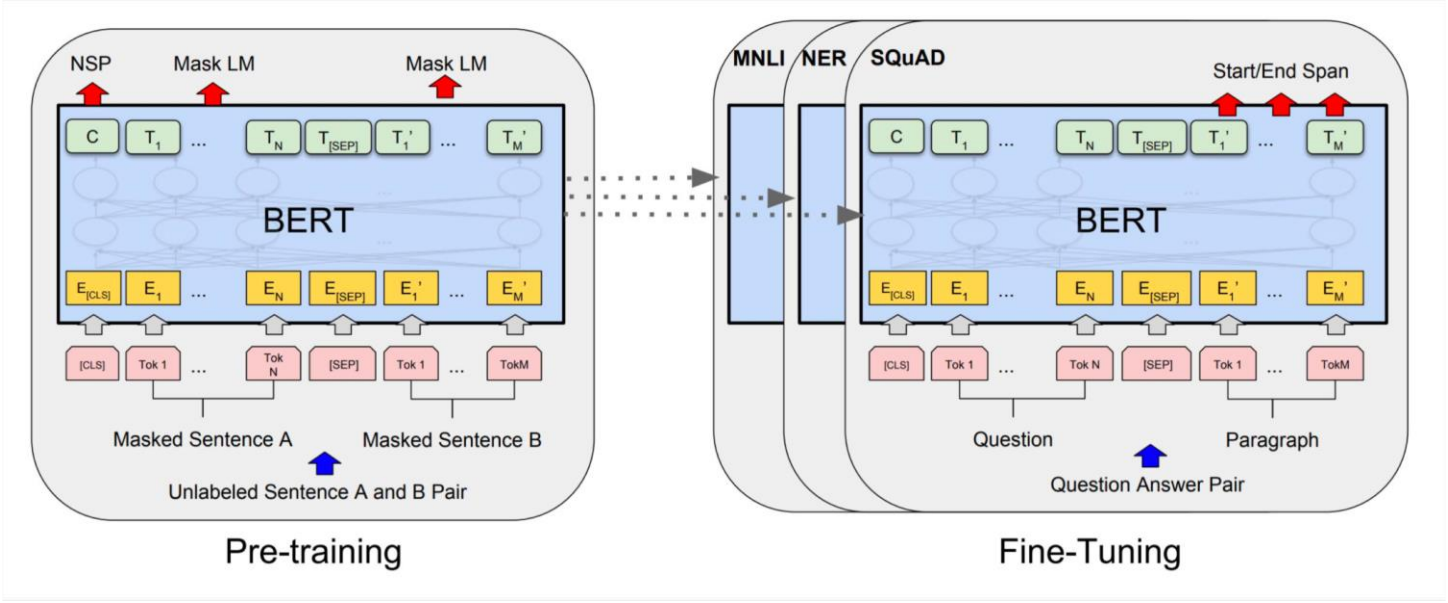
Many BERT-like embeddings

- XLM-R was trained on 2.5 TB of texts in 100 languages
- for Slovene: fastText (a variant of word2vec), ELMo, SloBERTa
- trilingual BERT – CroSloEngual
- on Clarin.si
- hundreds of papers investigating BERT-like models in major ML & NLP conferences

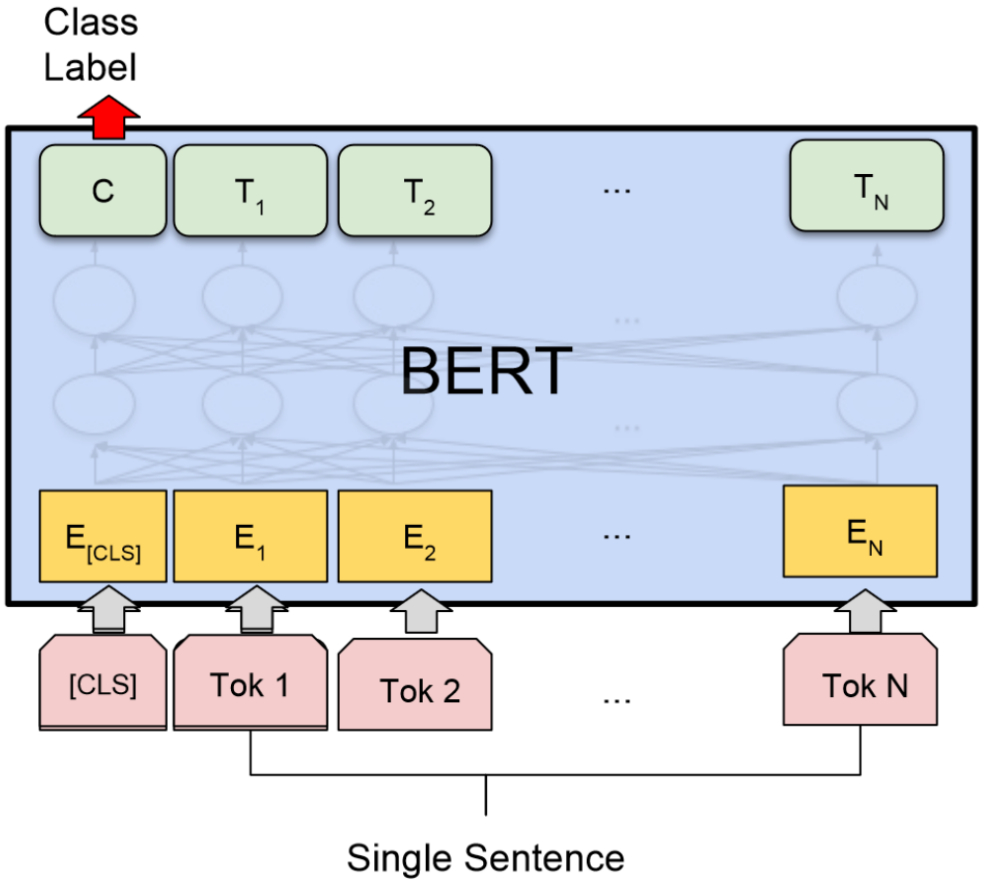
Ulčar, Matej and Marko Robnik-Šikonja. FinEst BERT and CroSloEngual BERT: less is more in multilingual models. In Proceedings of Text, Speech, and Dialogue, TSD2020, 2020.

Use of BERT

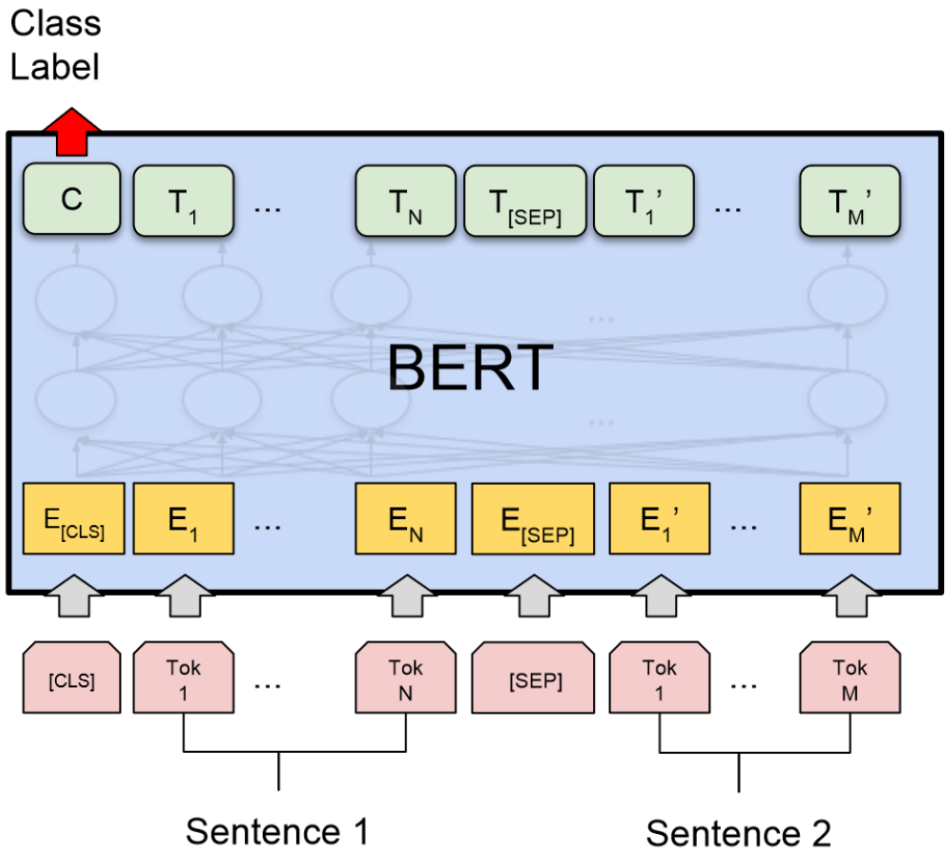
- train a classifier built on the top layer for each task that you fine tune for, e.g., Q&A, NER, inference
- achieves state-of-the-art results for many tasks
- GLUE and SuperGLUE tasks for NLI



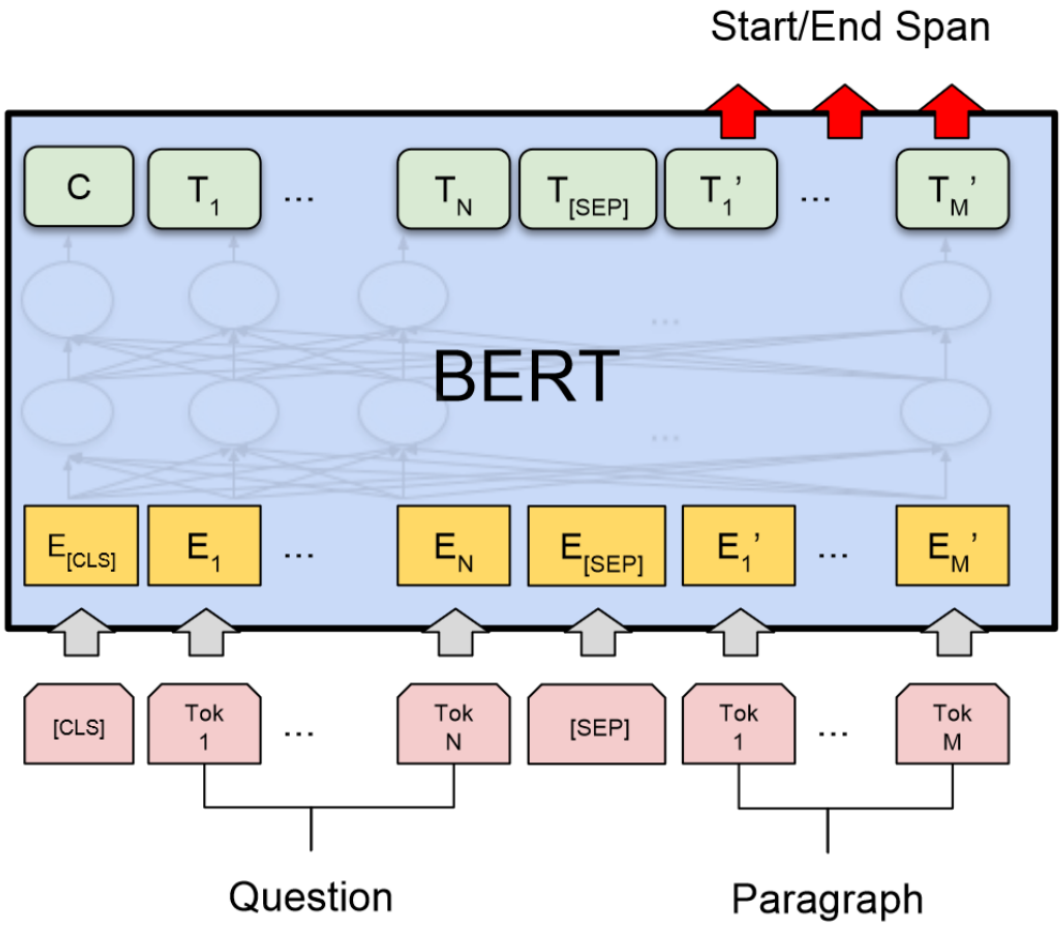
Sentence classification using BERT – sentiment, grammatical correctness



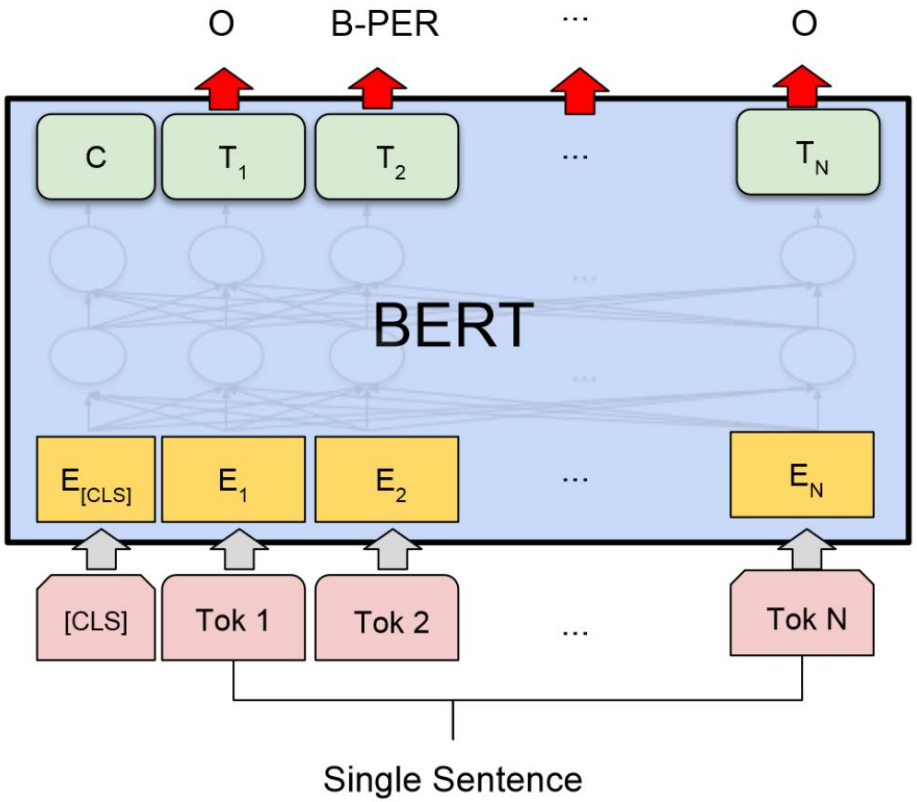
Two sentence classification using BERT-inference



Questions and answers with BERT

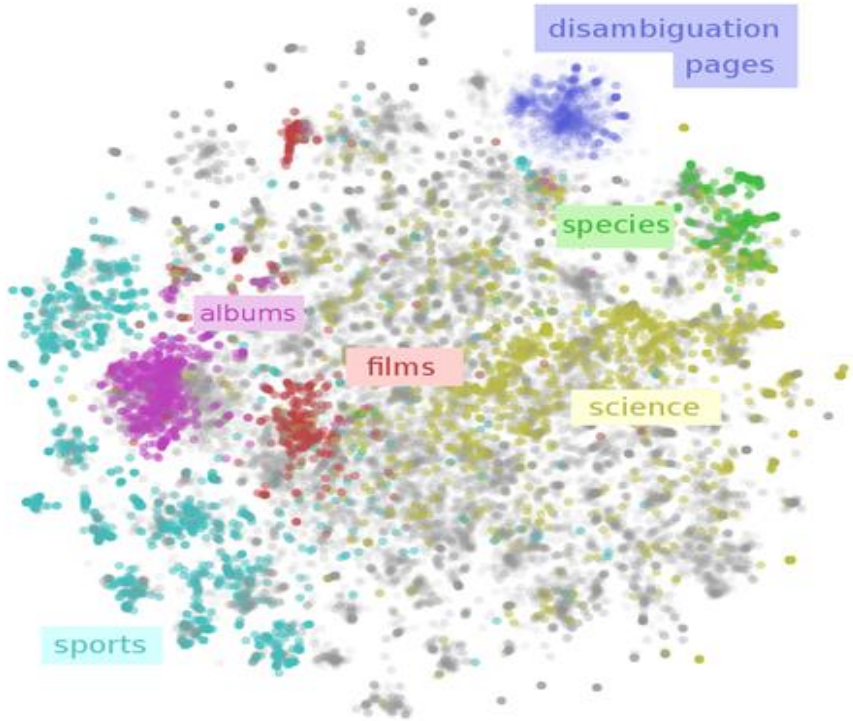


Sentence tagging with BERT-
NER, POS tagging, SRL



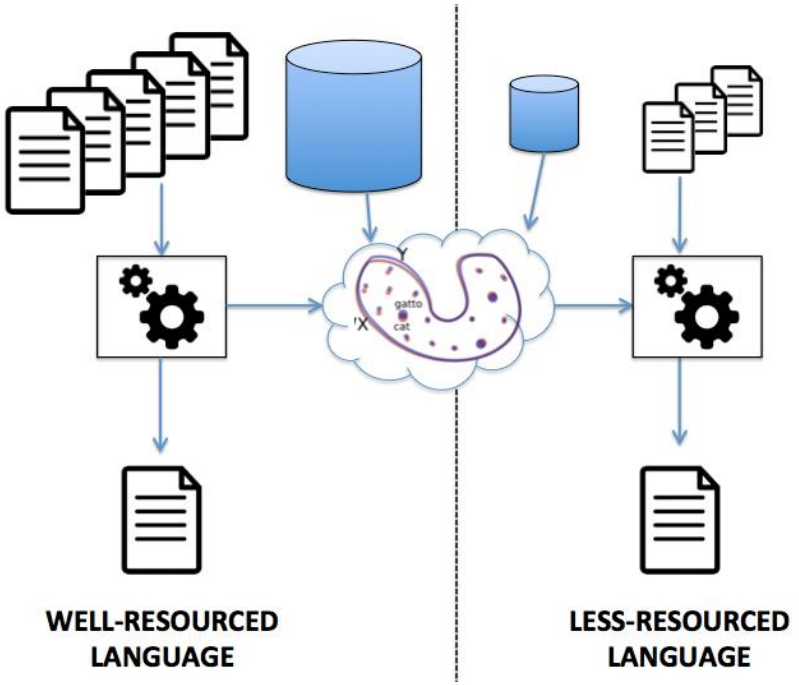
Cross-lingual embeddings

- mostly, embeddings are trained on monolingual resources
- words of one language form a cloud in high-dimensional space
- clouds for different languages can be aligned
  - $W S \approx E$ or
  - $W_1 S \approx W_2 E$



Cross-lingual model transfer based on embeddings

- Transfer of tools trained on mono-lingual resources



mostly not used anymore, superseded by multilingual LLMs

University of Ljubljana
Faculty of Computer and Information Science

Vocabulars in LLMs

- Tokenization depends on the dictionary
- The dictionary is constructed statistically (SentencePiece algorithm)

- Sentence: “Letenje je bilo predmet precej starodavnih zgodb.”
- SloBERTa:

'▁Le', 'tenje', '▁je', '▁bilo', '▁predmet', '▁precej', '▁staroda', 'vnih', '▁zgodb', '.’

- mBERT:

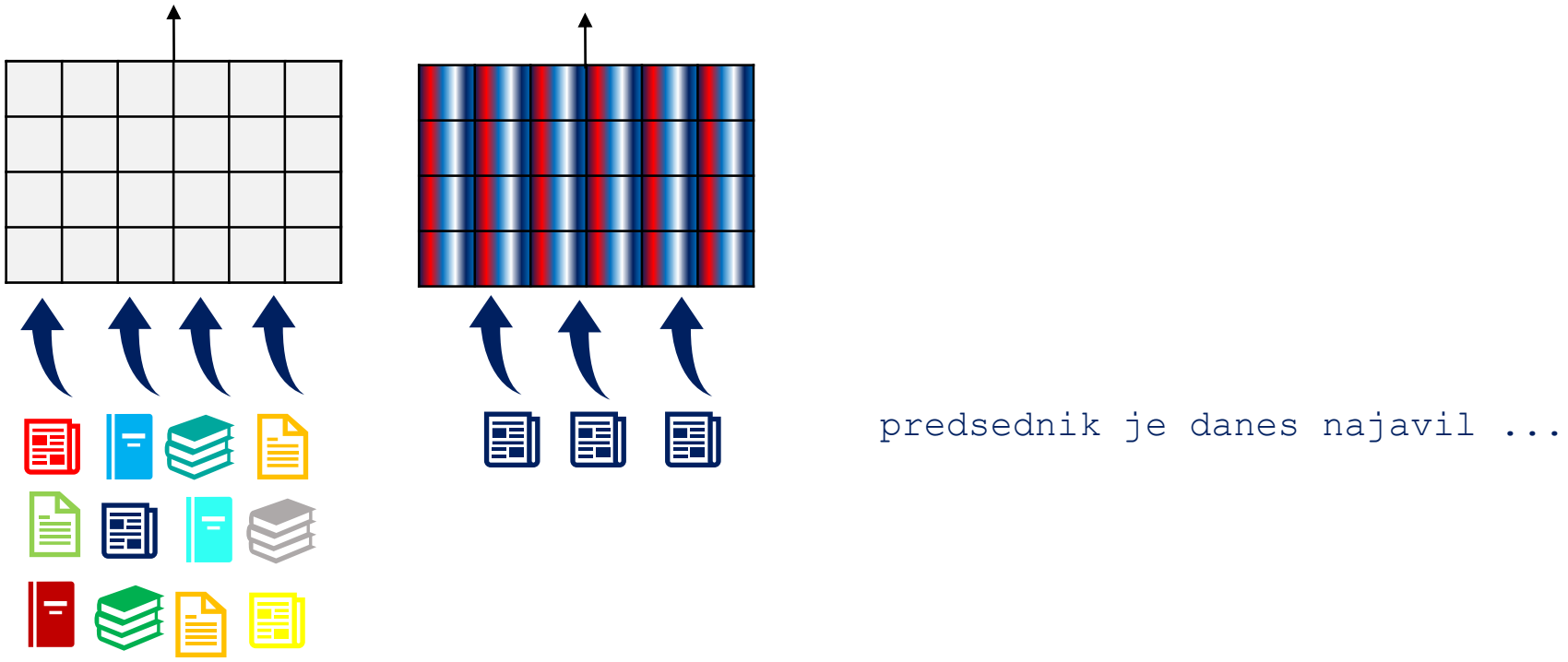
'Let', '##en', '##je', 'je', 'bilo', 'pred', '##met', 'pre', '##cej', 'star', '##oda', '##vnih', 'z', '##go', '##d', '##b', '.'

Ulčar, M., & Robnik-Šikonja, M. (2021) Training dataset and dictionary sizes matter in BERT models: The case of Baltic languages. In *International Conference on Analysis of Images, Social Networks and Texts* (pp. 162-172)

Multilingual LLMs

- Pretrained on multiple languages simultaneously
- multilingual BERT supports 104 languages by training on Wikipedia
- XLM-R was trained on 2.5 TB of texts
- allow cross-lingual transfer
- often solve the problem of insufficient training resources for less-resourced languages

Using multilingual models



Zero-shot transfer and few-shot transfer

What LLMs learn?

- We would like to travel to [MASK], ki je najlepši otok v Mediteranu.

SloBERTa: ..., Slovenija, I, Koper, Slovenia

CSE-BERT: Hvar, Rab, Cres, Malta, Brač

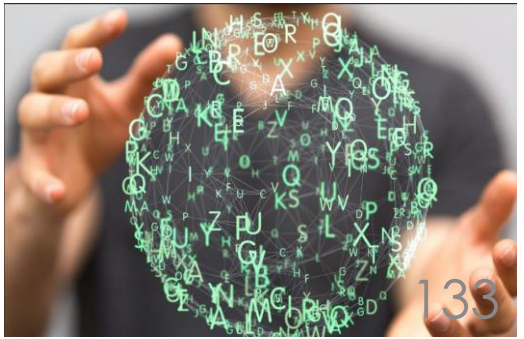
XLM-R: Mallorca, Tenerife, otok, Ibiza, Zadar

mBERT: Ibiza, Gibraltar, Tenerife, Mediterranean, Madeira

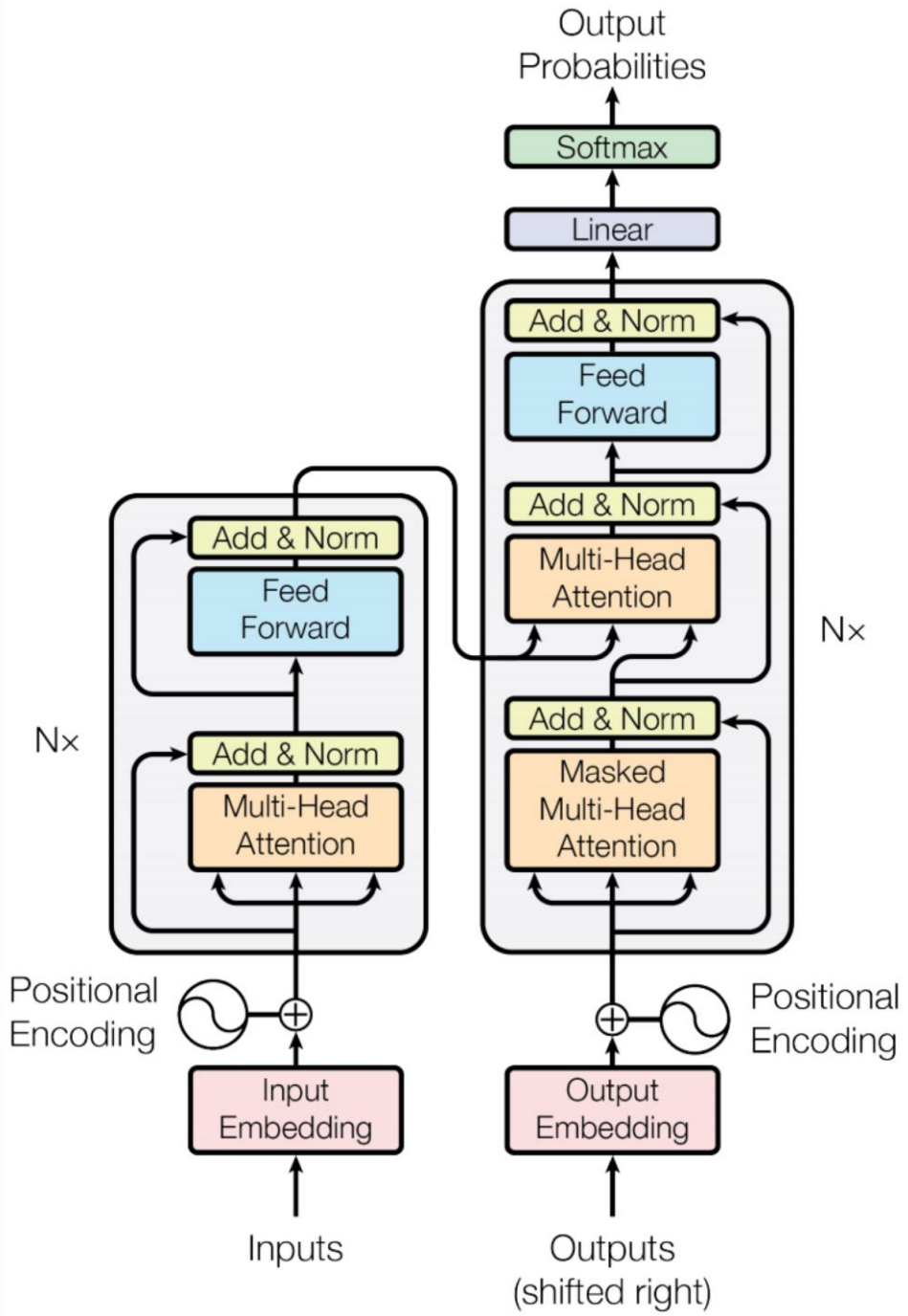
BERT (en): Belgrade, Italy, Serbia, Prague, Sarajevo

Embed all the things!

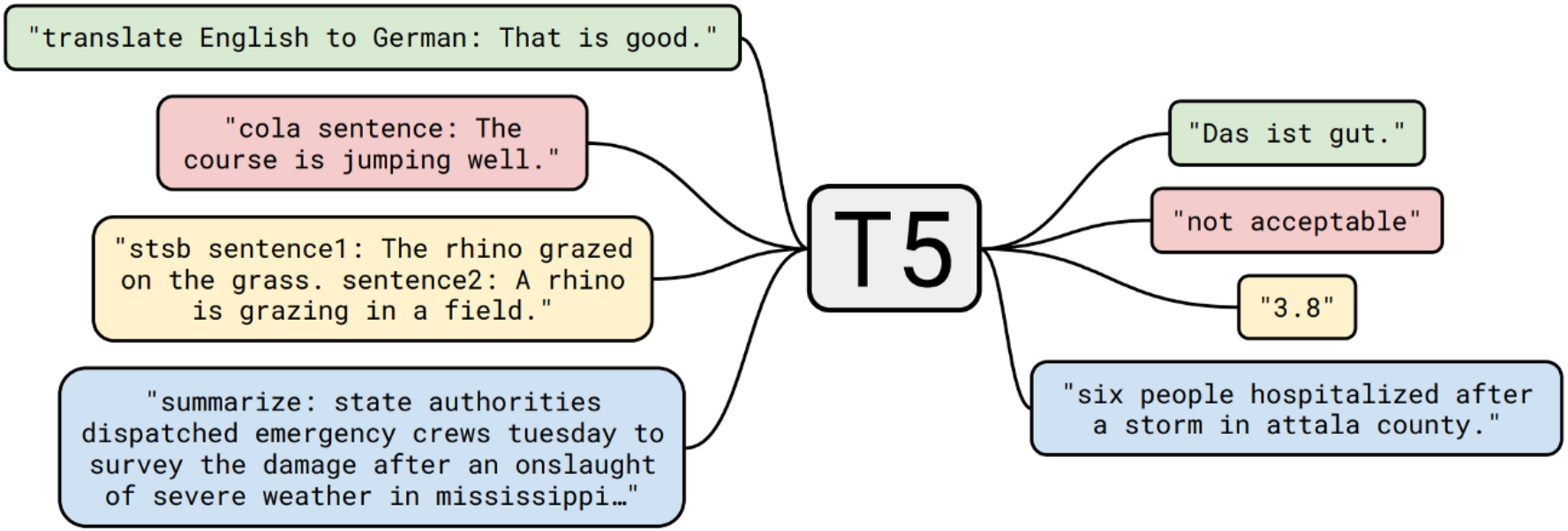
- Neural networks require numeric input
- Embedding shall preserve relations from the original space
- Representation learning is a crucial topic in nowadays machine learning
- Lots of applications whenever enough data is available to learn the representation
- In text, BERT-like models dominate for embeddings
- Similar ideas applied to texts, speech, graphs, electronic health records, relational data, time series, etc.



Transformer



T5 (Text-To-Text Transfer Transformer) models



Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y, Li, W. & Liu, P. J. (2020). Exploring the limits of transfer learning with a unified text-to-text transformer. *J. Mach. Learn. Res.*, *21*(140), 1-67.

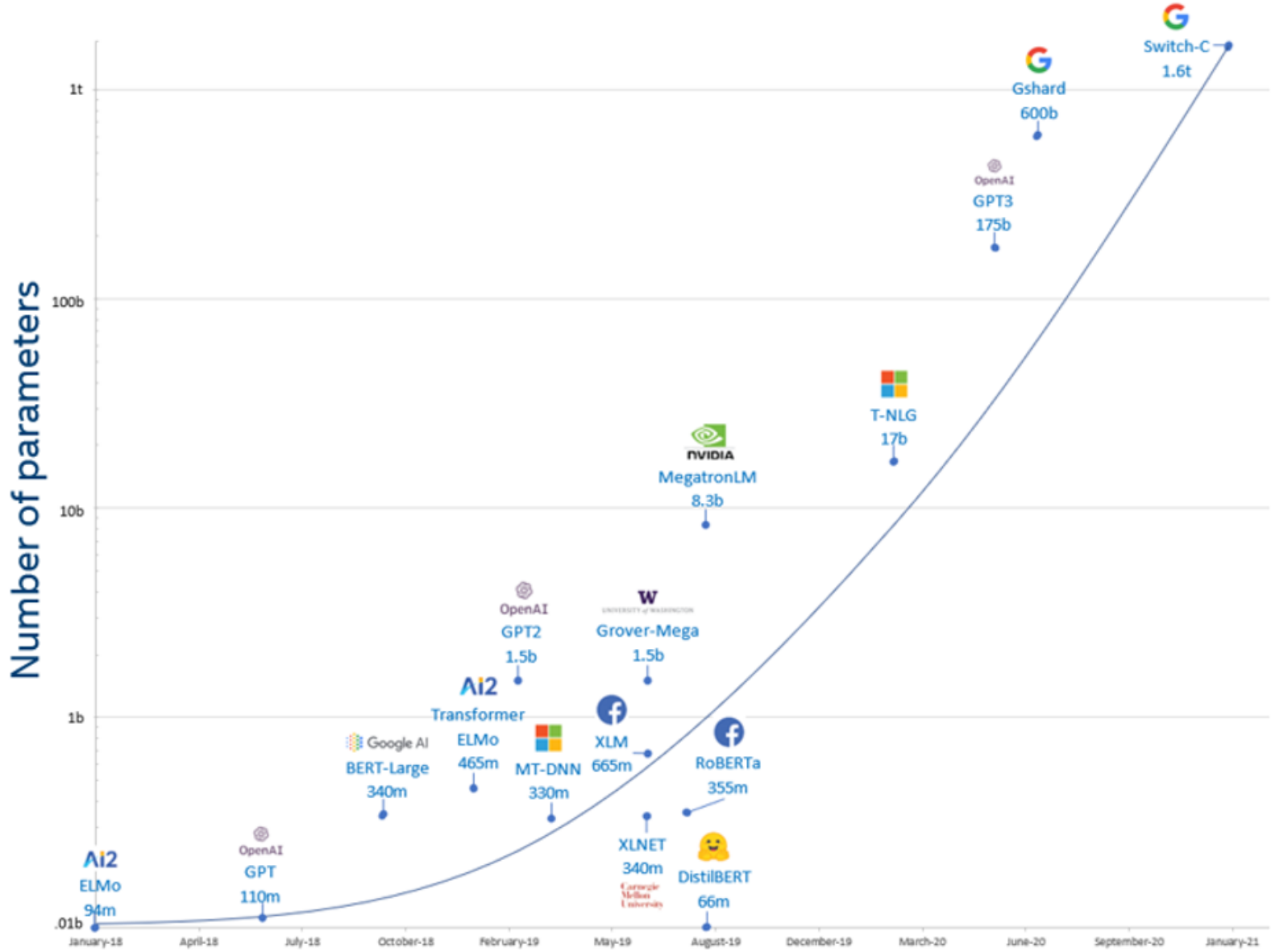
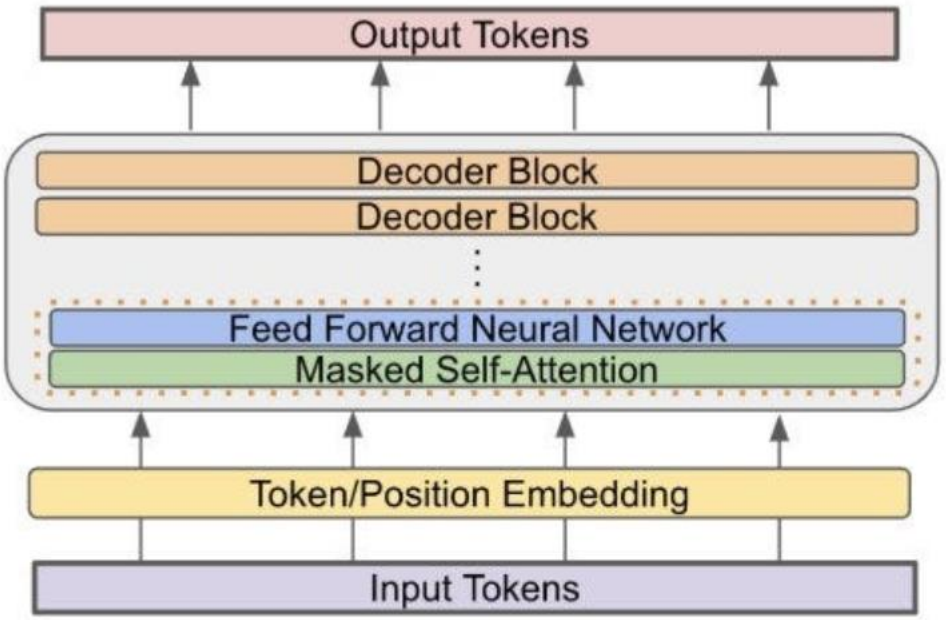


Figure 1: Exponential growth of number of parameters in DL models

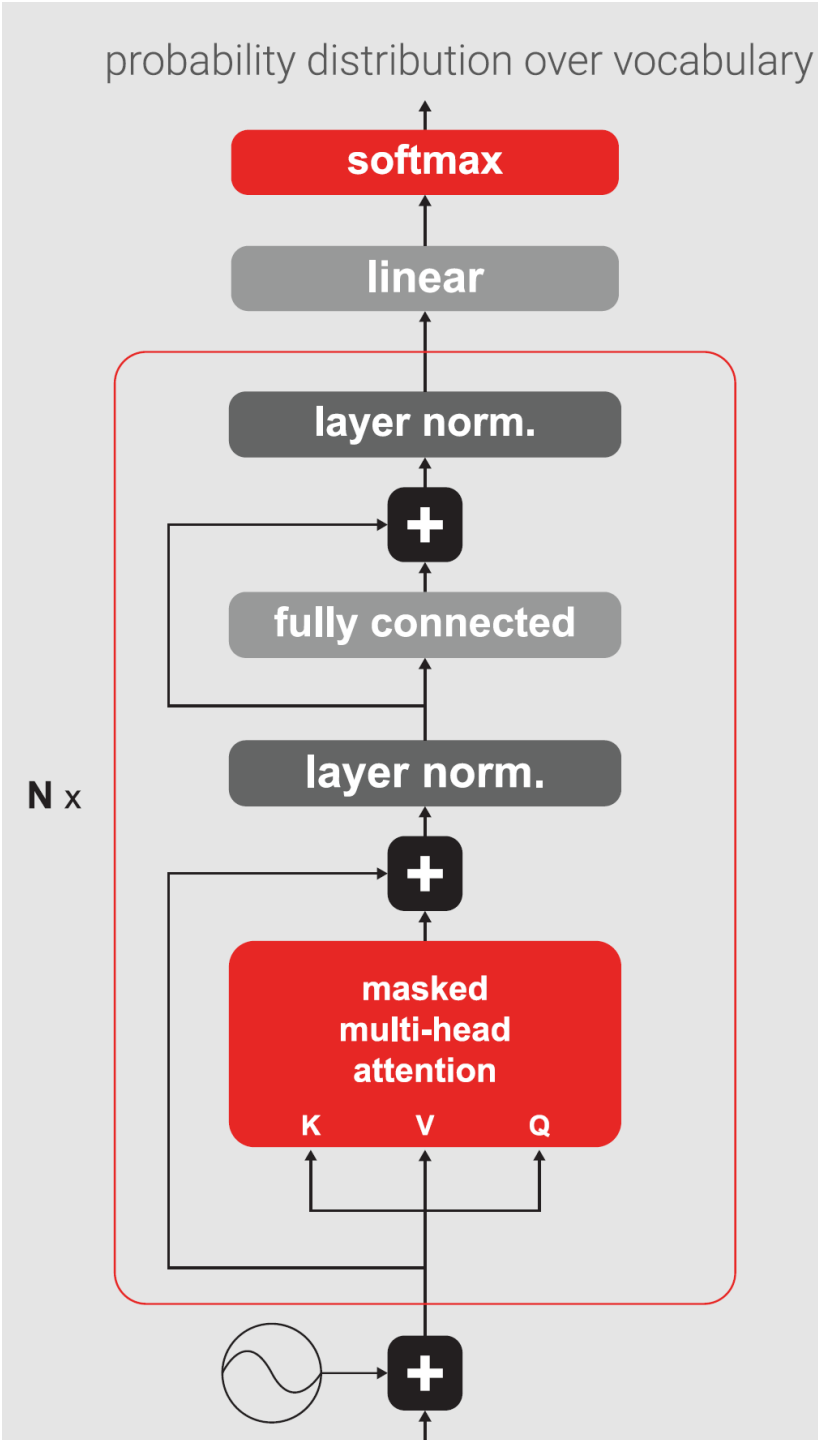
Decoder only models

- GPT, GPT-2, GPT-3, ChatGPT, GPT-4
- LLaMA, LLaMA-2, LLaMa-3
- MPT, Falcon
- Mistral and Mixtral
- OPT, Bloom
- Gemma and Gemini

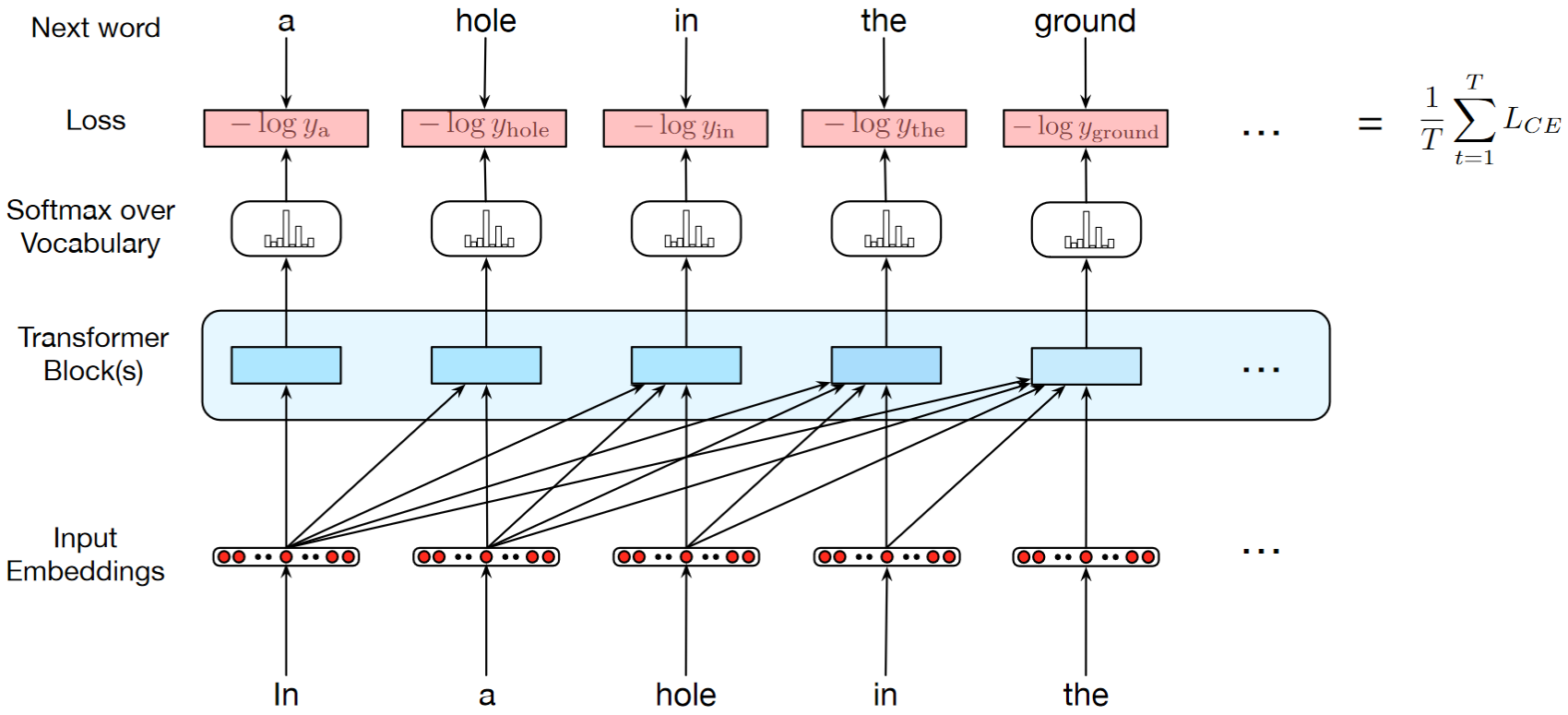


GPT family

- GPT: Generative Pre-trained Transformers
- use only the decoder part of transformer
- pretrained for language modeling (predicting the next word given the context)
- Potential shortcoming: unidirectional, does not incorporate bidirectionality
- “What are those?” he said while looking at my crocs.



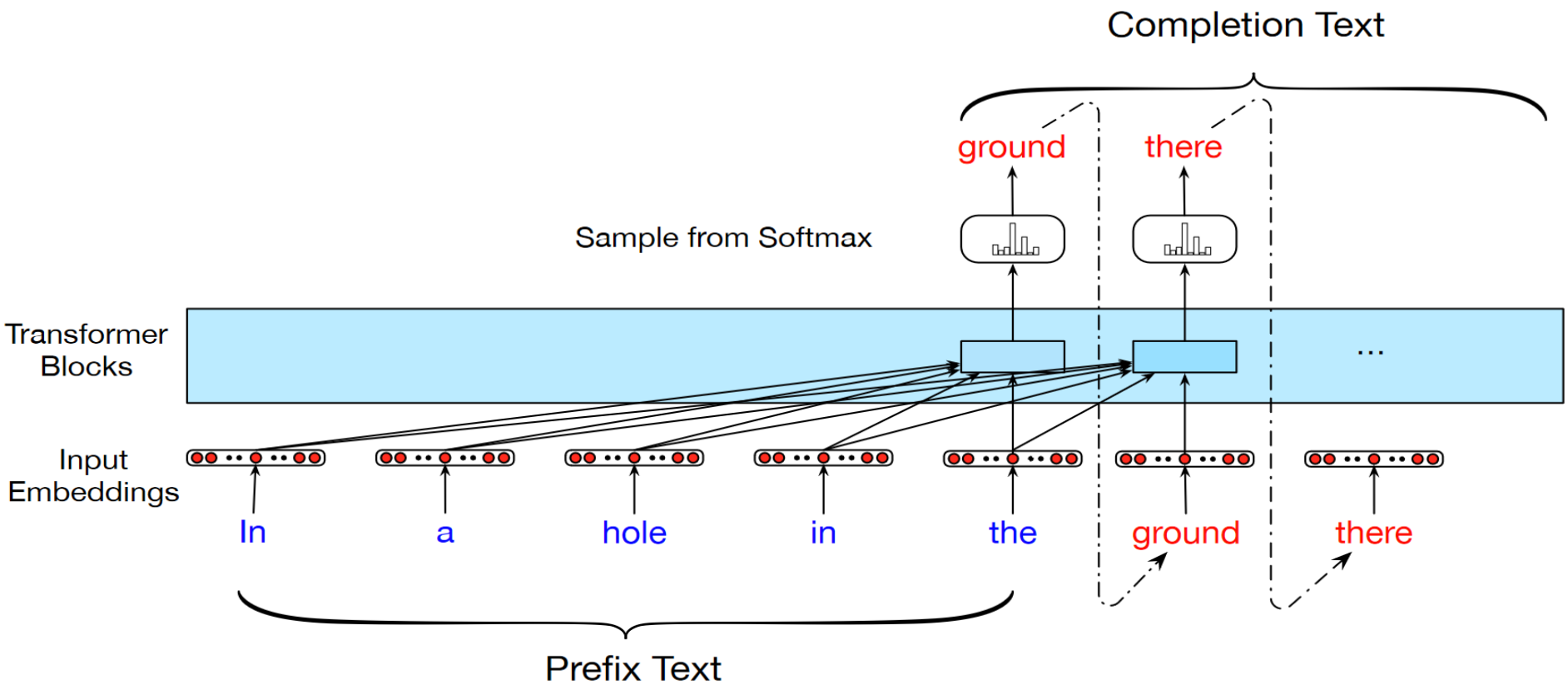
Transformer as a language model



- Can be computed in parallel

Autoregressive generators

- priming the generator with the context



- can be used also in summarization, QA and other generative tasks

GPT-2 and GPT-3

- few architectural changes, layer norm now applied to input of each subblock
- GPT-3 also uses some sparse attention layers
- more data, larger batch sizes (GPT-3 uses batch size of 3.2M)
- the models are scaled:

GPT-2:
48 layers, 25 heads
$d_m = 1600$, $d = 64$
context size = 1024
~ 1.5B parameters

GPT-3:
96 layers, 96 heads
$d_m = 12288$, $d = 128$
context size = 2048
~ 175B parameters

[1] Radford et al.: Language Models are Unsupervised Multitask Learners, 2019.
[2] Brown et al.: Language Models are Few-Shot Learners, 2020.

- see demos at https://transformer.huggingface.co/

In-context learning

- GPT-2 and GPT-3 were the first models where one can ditch the “pre-train and fine-tune” training paradigm of GPT;
- GPT-2 explores unsupervised zero-shot learning, whereas in GPT-3 the authors expand the idea into in-context learning;
- Use text input to condition the model on task description and some examples with ground truth.
- Uses zero-shot learning, one-shot learning, few-shot learning (as many examples as they can fit into the context, usually 10-100)
- No gradient updates are performed.
- A sort of associative lookup

In-context learning

Zero-shot

The model predicts the answer given only a natural language description of the task. No gradient updates are performed.

```
1  Translate English to French:        ← task description
2  cheese =>                           ← prompt
```

One-shot

In addition to the task description, the model sees a single example of the task. No gradient updates are performed.

```
1  Translate English to French:        ← task description
2  sea otter => loutre de mer          ← example
3  cheese =>                           ← prompt
```

Few-shot

In addition to the task description, the model sees a few examples of the task. No gradient updates are performed.

```
1  Translate English to French:        ← task description
2  sea otter => loutre de mer          ← examples
3  peppermint => menthe poivrée        ←
4  plush girafe => girafe peluche      ←
5  cheese =>                           ← prompt
```

Figure source: Brown et al.: Language Models are Few-Shot Learners, 2020.

- GPT-3 is still a language model and can be used for text generation

- only 12% of respondents correctly classified this as not written by a human

Title: United Methodists Agree to Historic Split
Subtitle: Those who oppose gay marriage will form their own denomination
Article: **After two days of intense debate, the United Methodist Church has agreed to a historic split - one that is expected to end in the creation of a new denomination, one that will be "theologically and socially conservative," according to The Washington Post. The majority of delegates attending the church's annual General Conference in May voted to strengthen a ban on the ordination of LGBTQ clergy and to write new rules that will "discipline" clergy who officiate at same-sex weddings. But those who opposed these measures have a new plan: They say they will form a separate denomination by 2020, calling their church the Christian Methodist denomination.**
The Post notes that the denomination, which claims 12.5 million members, was in the early 20th century the "largest Protestant denomination in the U.S.," but that it has been shrinking in recent decades. The new split will be the second in the church's history. The first occurred in 1968, when roughly 10 percent of the denomination left to form the Evangelical United Brethren Church. The Post notes that the proposed split "comes at a critical time for the church, which has been losing members for years," which has been "pushed toward the brink of a schism over the role of LGBTQ people in the church." Gay marriage is not the only issue that has divided the church. In 2016, the denomination was split over ordination of transgender clergy, with the North Pacific regional conference voting to ban them from serving as clergy, and the South Pacific regional conference voting to allow them.

Huge generative language models

- ChatGPT, OpenAI, Nov. 2022
  based on GPT-3.5 with additional training for dialogue
- uses instruction-tuning and RLHF
  (reinforcement learning with human feedback)
- demo: https://chat.openai.com/
- huge public impact, possibly disruptive for writing professions, learning, teaching, scientific writing
- GPT-4, 2023: even larger, allows longer context, image input

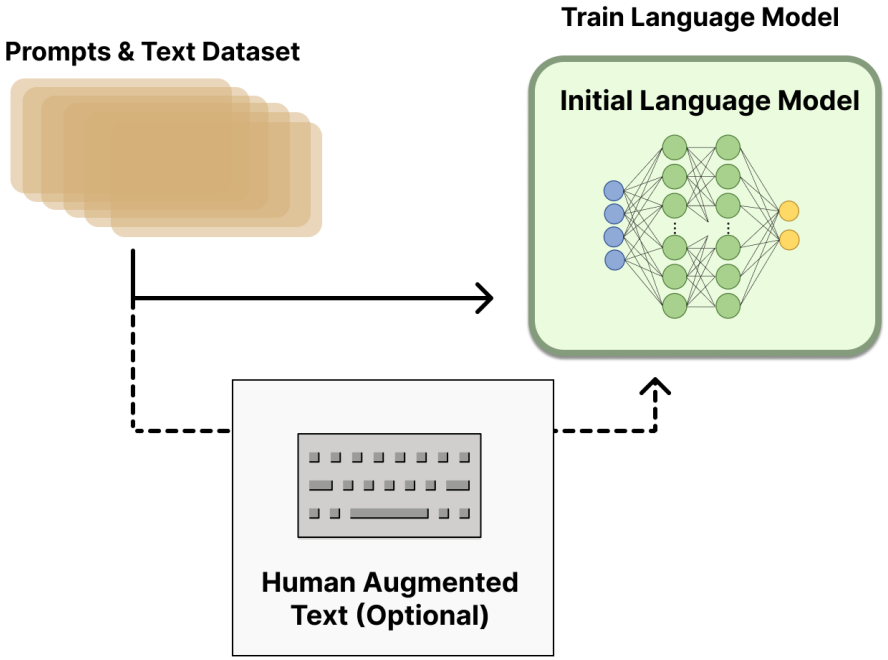
RLHF: idea

- Reinforcement Learning with Human Feedback
- A problem: Human feedback is not present during training
- Idea: Train a separate model on human feedback, this model can generate a reward to be used during training of LLM
- Three stages:
  1. Pretraining a language model (LM),
  2. Gathering data and training a reward model, and
  3. Fine-tuning the LM with reinforcement learning.

Nathan Lambert, Louis Castricato, Leandro von Werra, Alex Havrilla: "Illustrating Reinforcement Learning from Human Feedback (RLHF)", Hugging Face Blog, 2022.

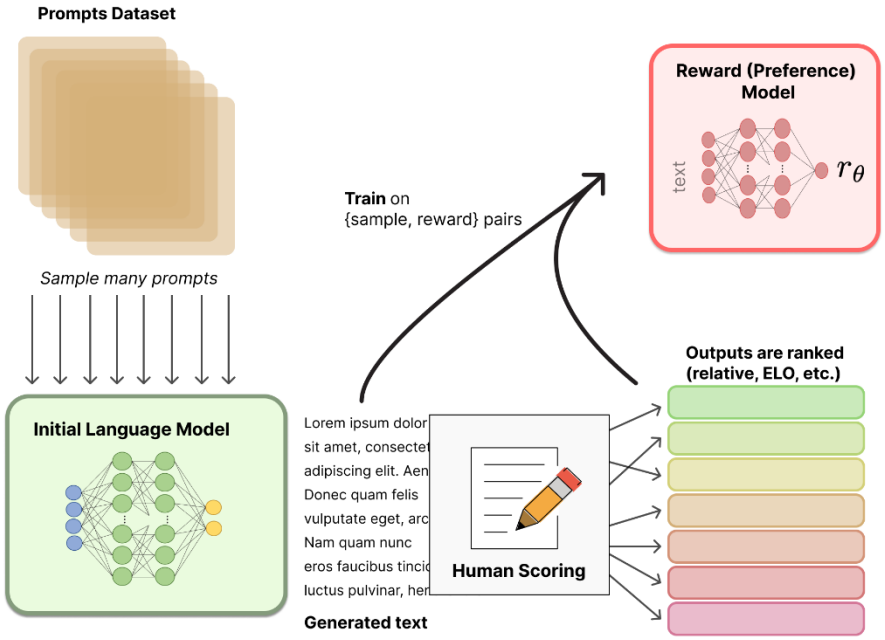
RLHF: the reward model 1/2

- input: a sequence of text, e.g., produced by LM and optionally improved by humans
- output: a scalar reward, representing the human preference of the text (e.g., a rank of the answer)
- the reward model could be an end-to-end LM, or the model ranks outputs, and the ranking is converted to reward



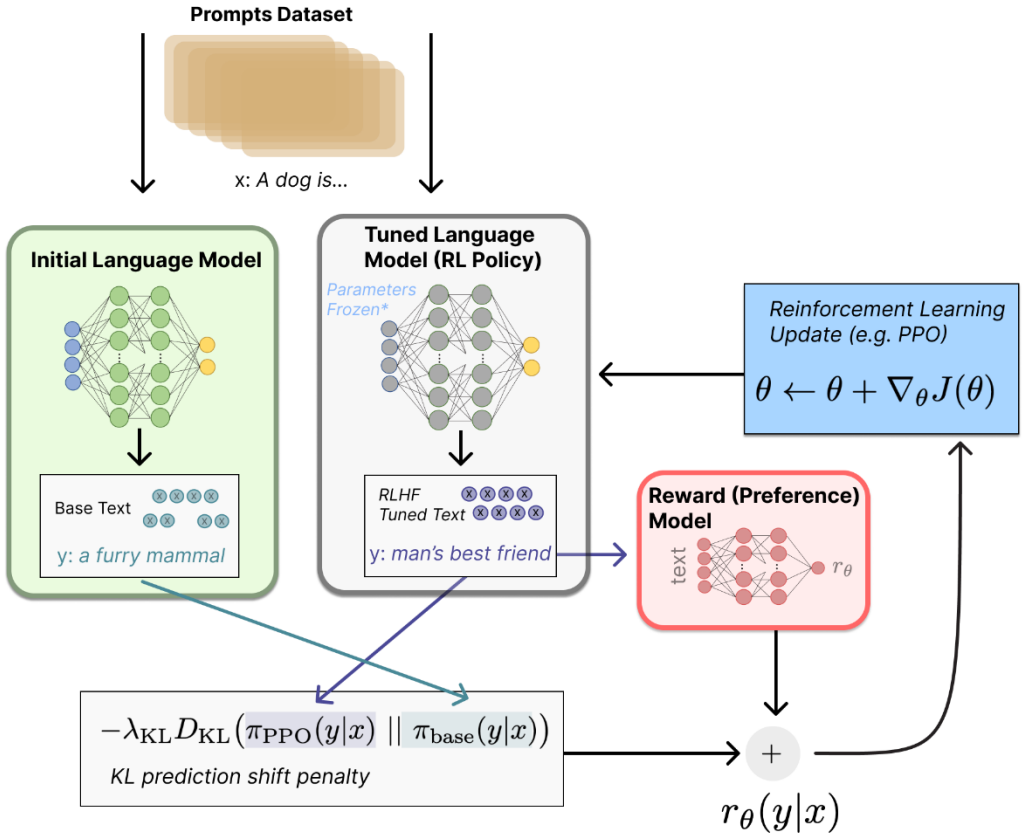
RLHF: the reward model 2/2

- the training dataset are pairs of prompts and (human improved) LM responses, e.g., 50k instances
- humans rank the responses instead of producing the direct reward as this produces better calibrated scores



RLHF: fine-tuning with RL

- RL does not change all parameters, most of parameters are frozen
- the algorithm: Proximal Policy Optimization (PPO)



NLP application examples

Sentiment analysis (SA)

- Definition: a computational study of opinions, sentiments, emotions, and attitudes expressed in texts towards an entity.
- Purpose: detecting public moods, i.e. understanding the opinions of the general public and consumers on social events, political movements, company strategies, marketing campaigns, product preferences etc.
- Part of Affective Computing (emotion, mood, personality traits, interpersonal stance, attitude)
- Can be target-based or general

SA: getting and preprocessing data

- Frequent data sources:
  - Twitter-X, forum comments, product review sites, company's Facebook pages
- Data cleaning
  - quality assessment, annotator (self-) agreement
  - preprocessing: tokenization, emojis, links, hashtags, etc,

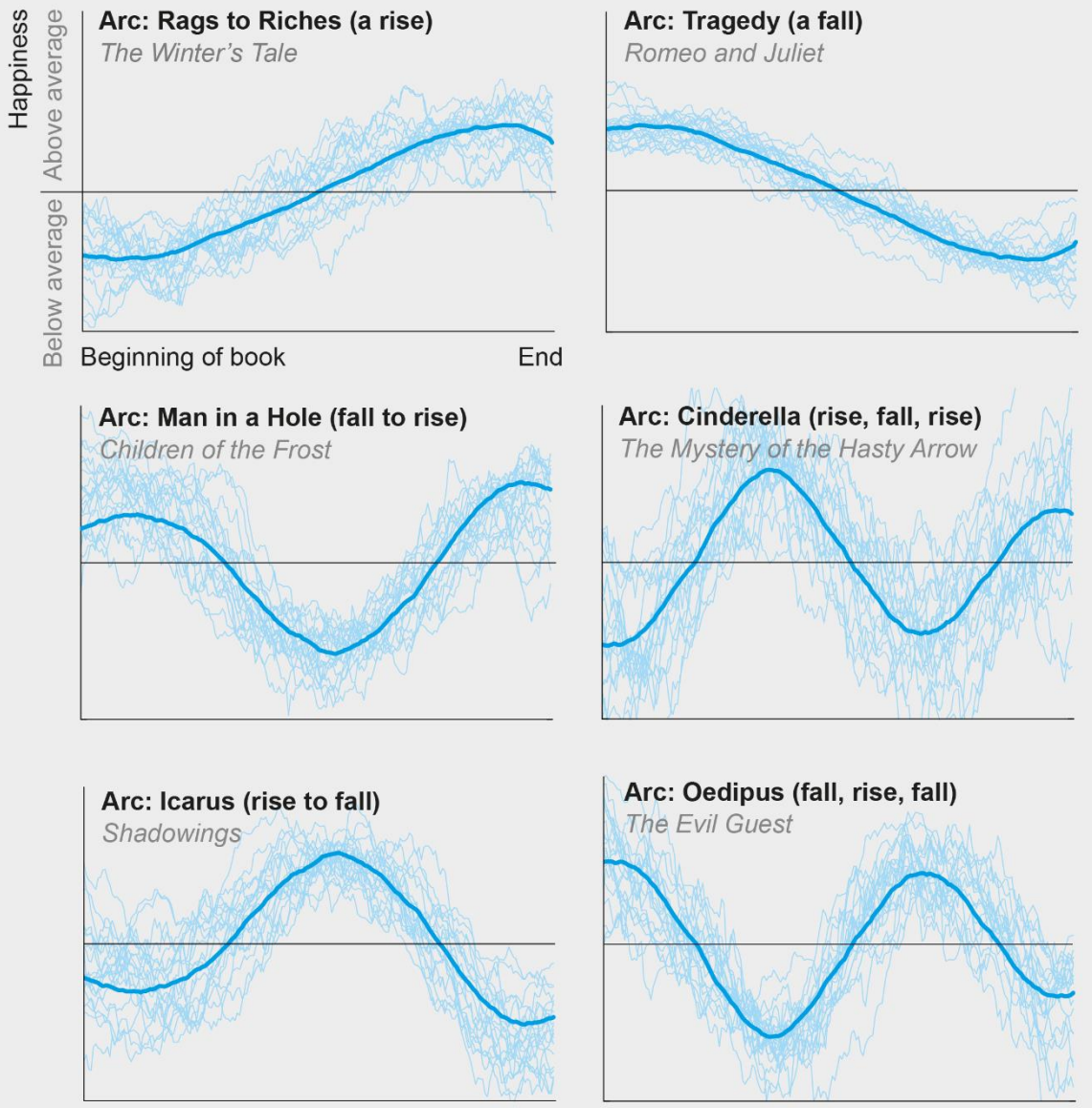
SA tasks

- sentiment classification (binary (polarity), ternary, n-ary
- subjectivity classification (vs. objectivity)
- review usefulness classification
- opinion spam classification
- emotion analysis
- hate speech, offensive speech, socially unacceptable speech
- stance detection

Emotional states in English fiction

Emotional Arcs

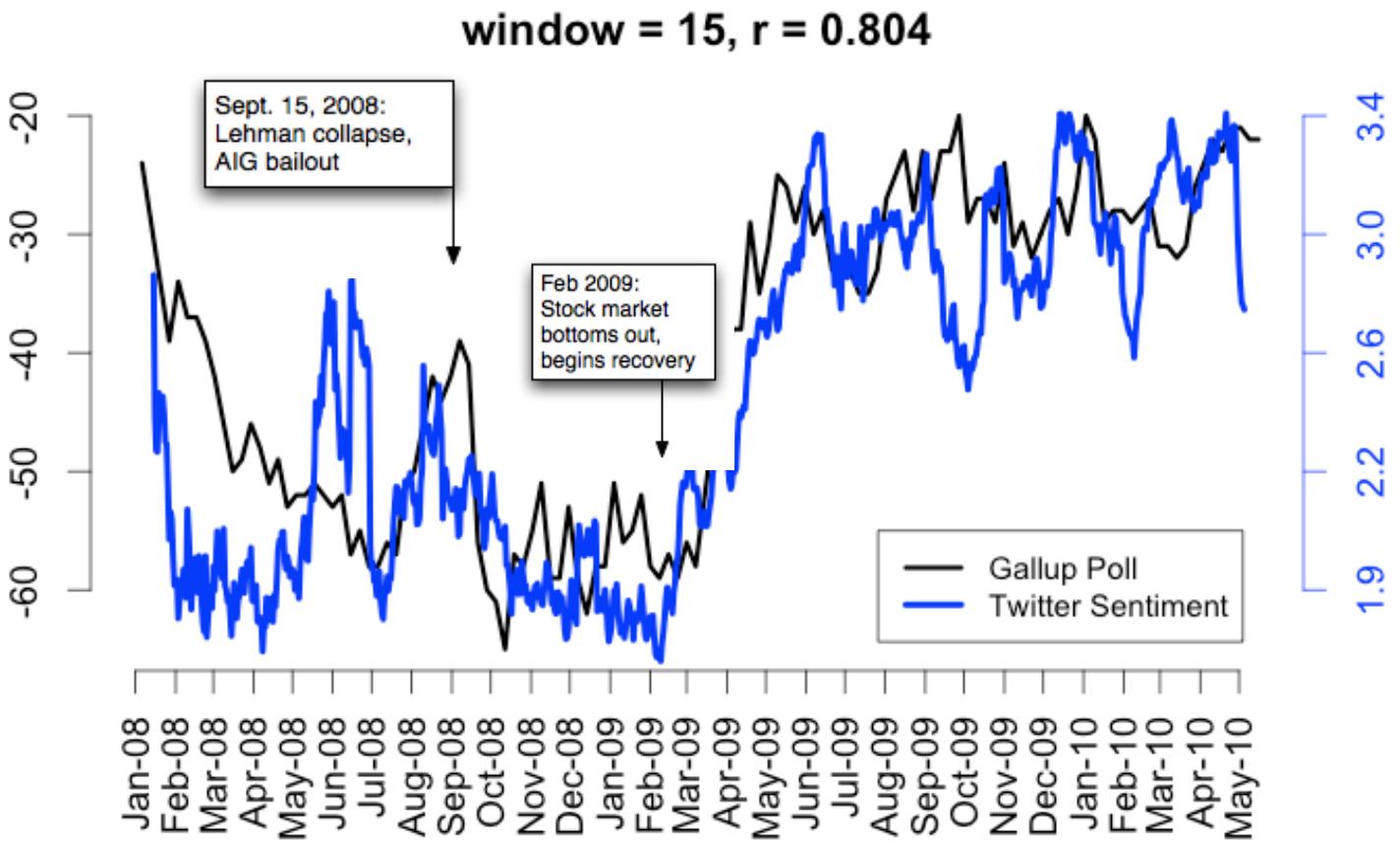
About 85 percent of 1,327 fiction stories in the digitized Project Gutenberg collection follow one of six emotional arcs—a pattern of highs and lows from beginning to end (*dark curves*). The arcs are defined by the happiness or sadness of words in the running text (*jagged plots*). All books were in English and less than 100,000 words; examples are noted.



Public opinion surveys

Twitter sentiment vs. Gallup on consumer sentiment

Brendan O'Connor, Ramnath Balasubramanyan, Bryan R. Routledge, and Noah A. Smith. 2010. From Tweets to Polls: Linking Text Sentiment to Public Opinion Time Series. In ICWSM-2010



Statistical machine translation

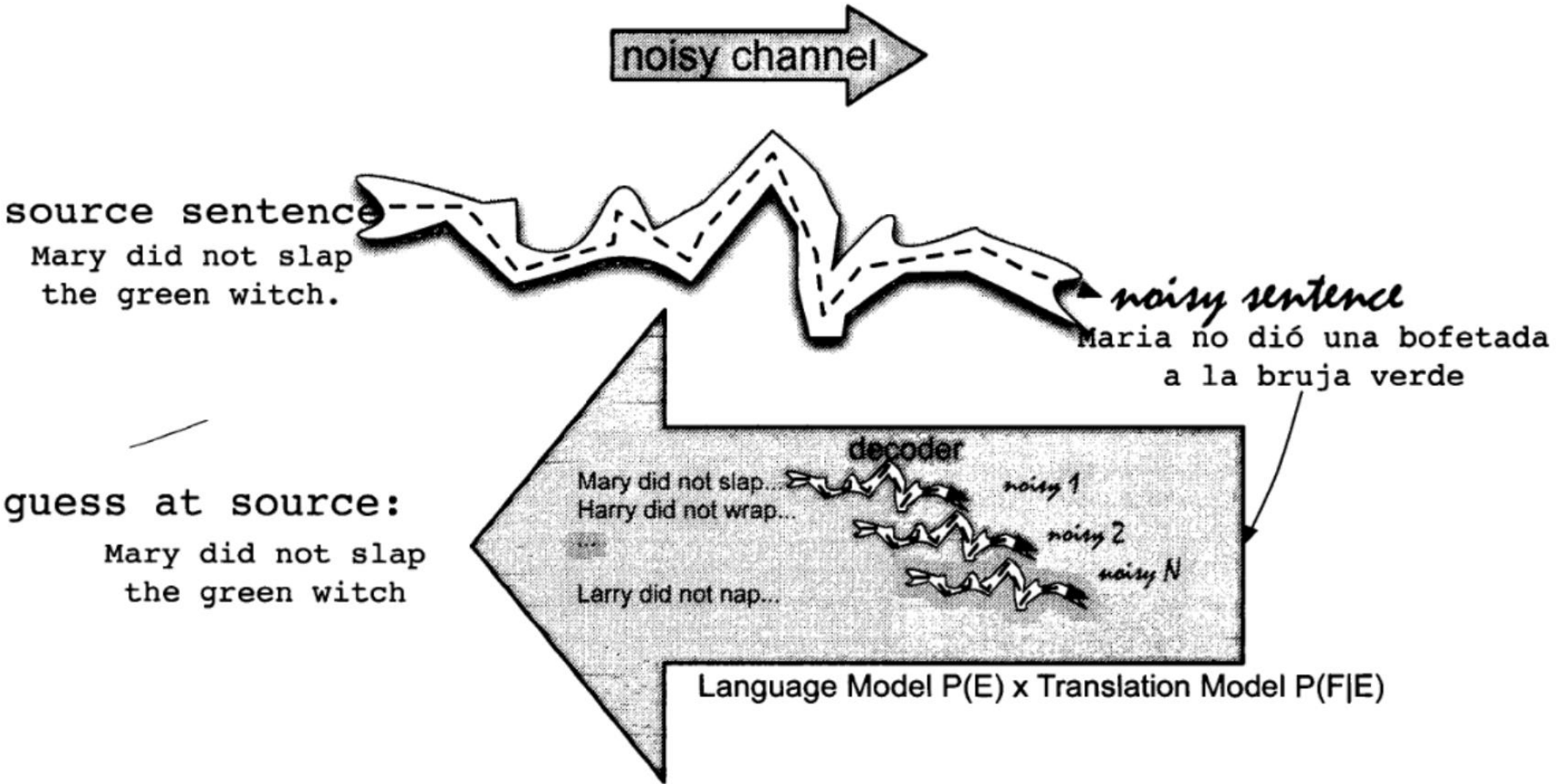
- non-neural approach: no longer used in practice but gives insight of what is needed
- idea from the theory of information
- we translate from foreign language F to English E
- a document is translated based on the probability distribution $p(e \mid f)$, i.e. the probability of the sentence e in target language based on the sentence in source language f
- Bayes rule
  $\arg\max_e p(e \mid f) = \arg\max_e p(f \mid e)\, p(e) \,/\, p(f)$
- $p(f)$ can be ignored as it is a constant for a given fixed sentence
- traditional (non-neural) approaches split the problem into subproblems
  - create a language model $p(e)$
  - a separate translation model $p(f \mid e)$
  - decoder forms the most probable e based on f

Noisy channel model

- given English sentence e
- during transmission over a noisy channel the sentence e is corrupted and we get sentence in a foreign language f, which we are able to observe
- to reconstruct the most probable sentence e we have to figure out:
  - how people speak in English (language model), p(e) and
  - how to transform a foreign language into English (translation model), p(f | e)

Noisy channel

- reasoning goes back



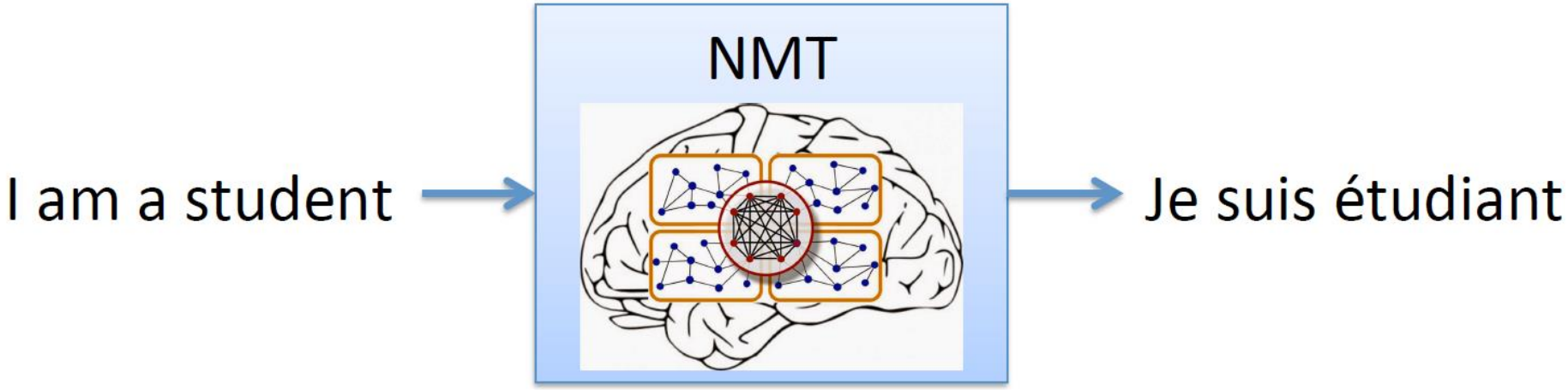
Language model

- each target (English) sentence e is assigned a probability p(e)
- estimation of probabilities for the whole sentences is not possible (why?), therefore we use language models, e.g., 3-gram models or neural language models

Translation model

- we have to assign a probability of p(f | e), which is a probability of a foreign language sentence f, given target sentence e.
- we search the e which maximizes p(e) * p(f | e)
- traditional MT approach: using translation corpus we determine which translation of a given word is the most probable
- we take into account the position of a word and how many words are needed to translate a given word

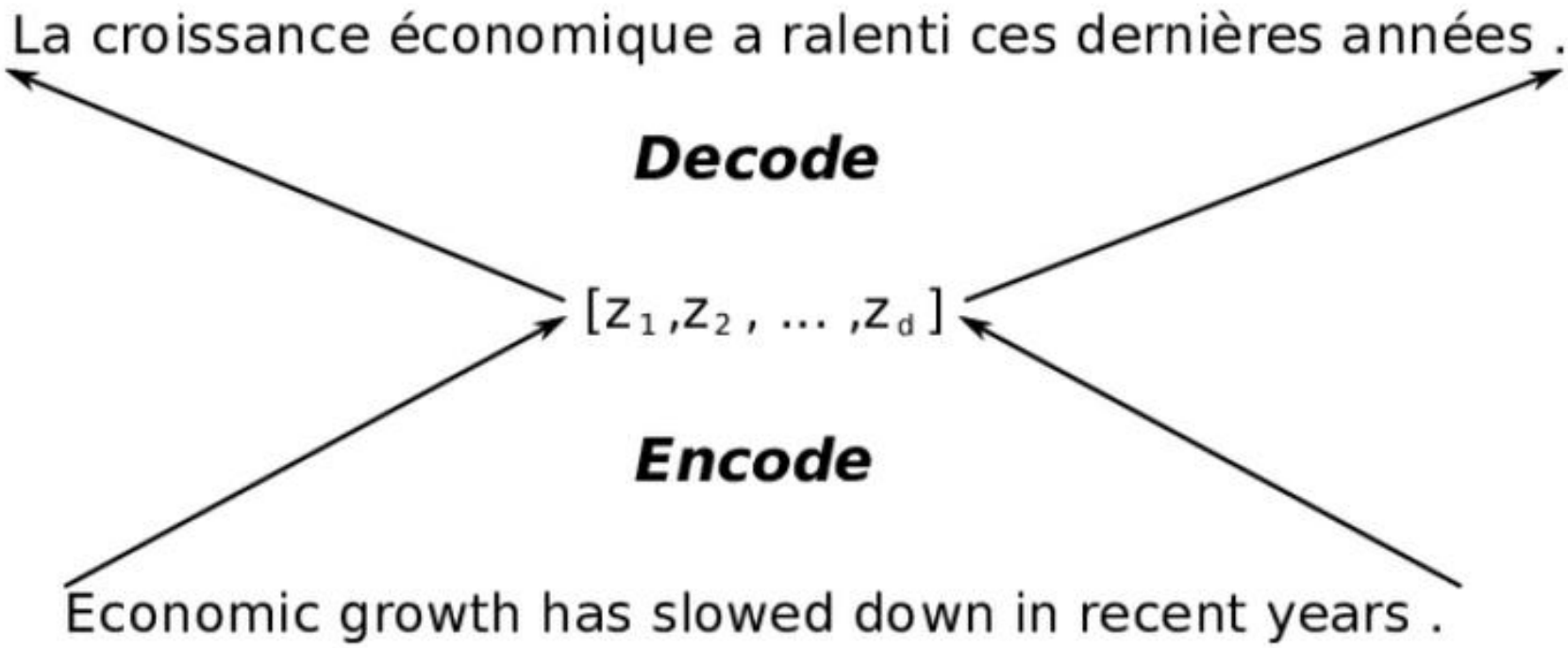
Neural machine translation (NMT)



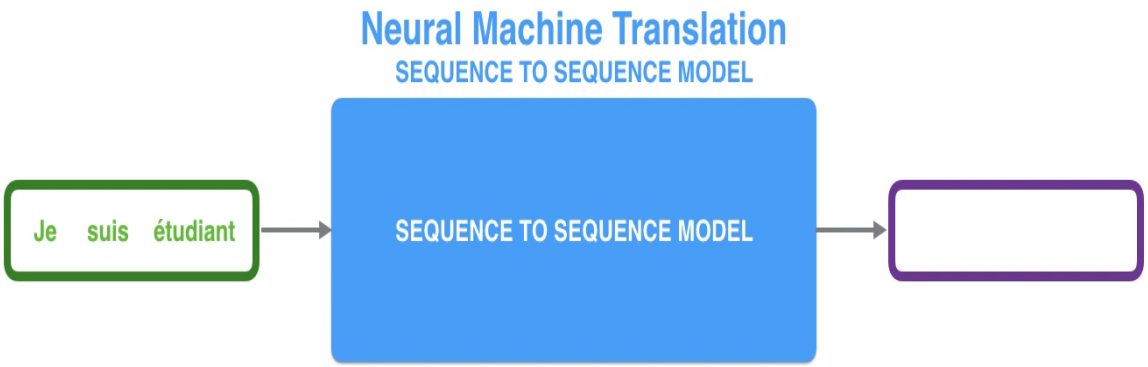
(Sutskever et al., 2014; Cho et al., 2014)

- sequence to sequence machine translation (seq2seq)
- end-to-end optimization

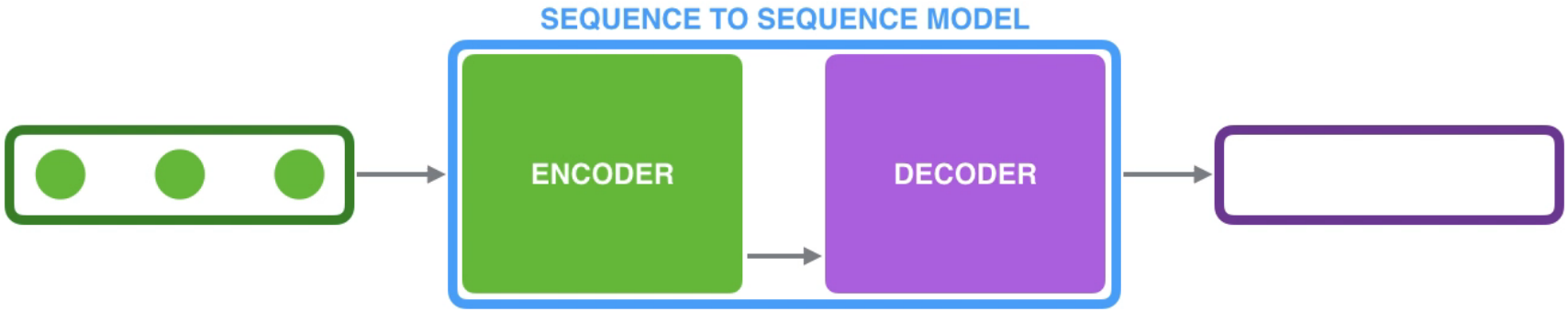
Encoder-Decoder model



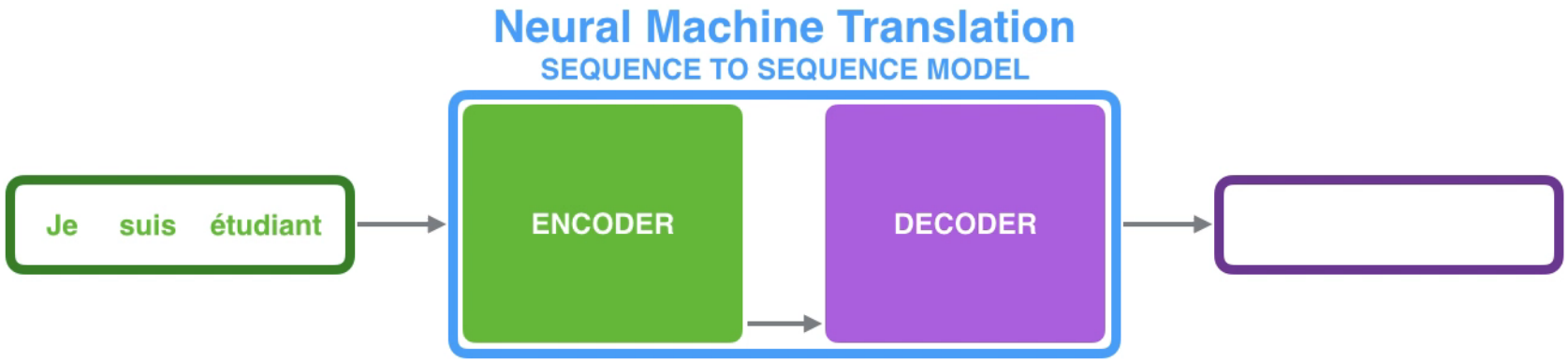
Seq2Seq for NMT

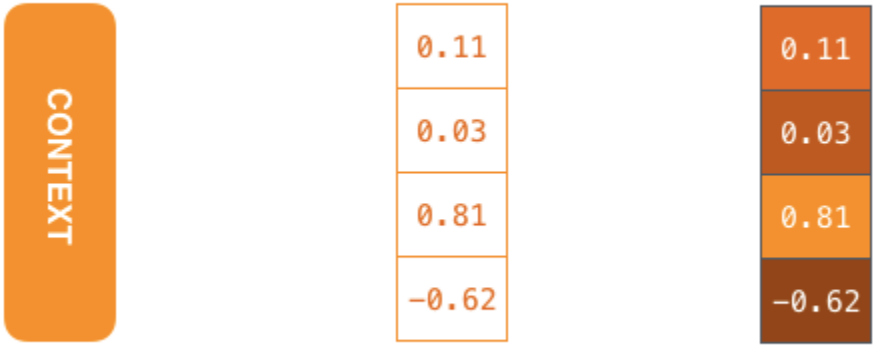


Encoder-decoder for sequences



Encoder-decoder for NMT

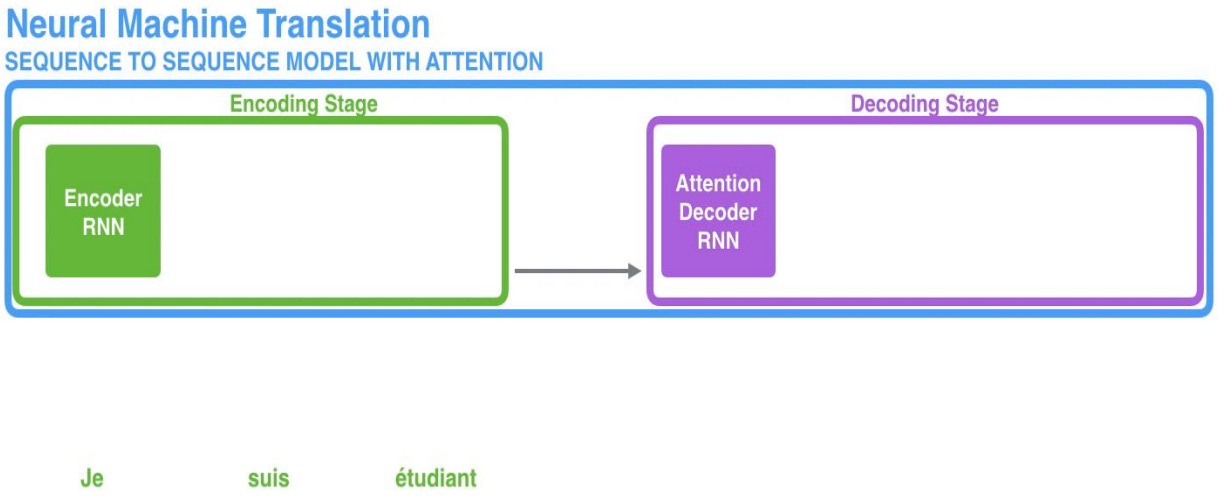




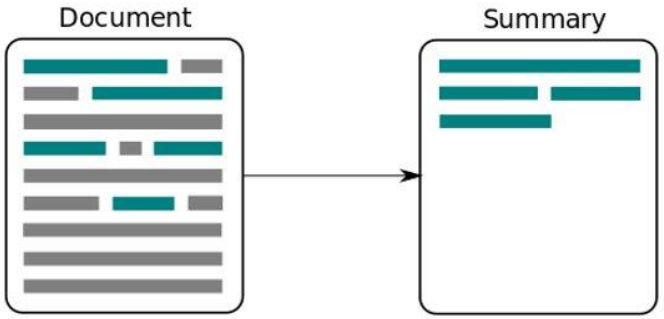
Training NMT

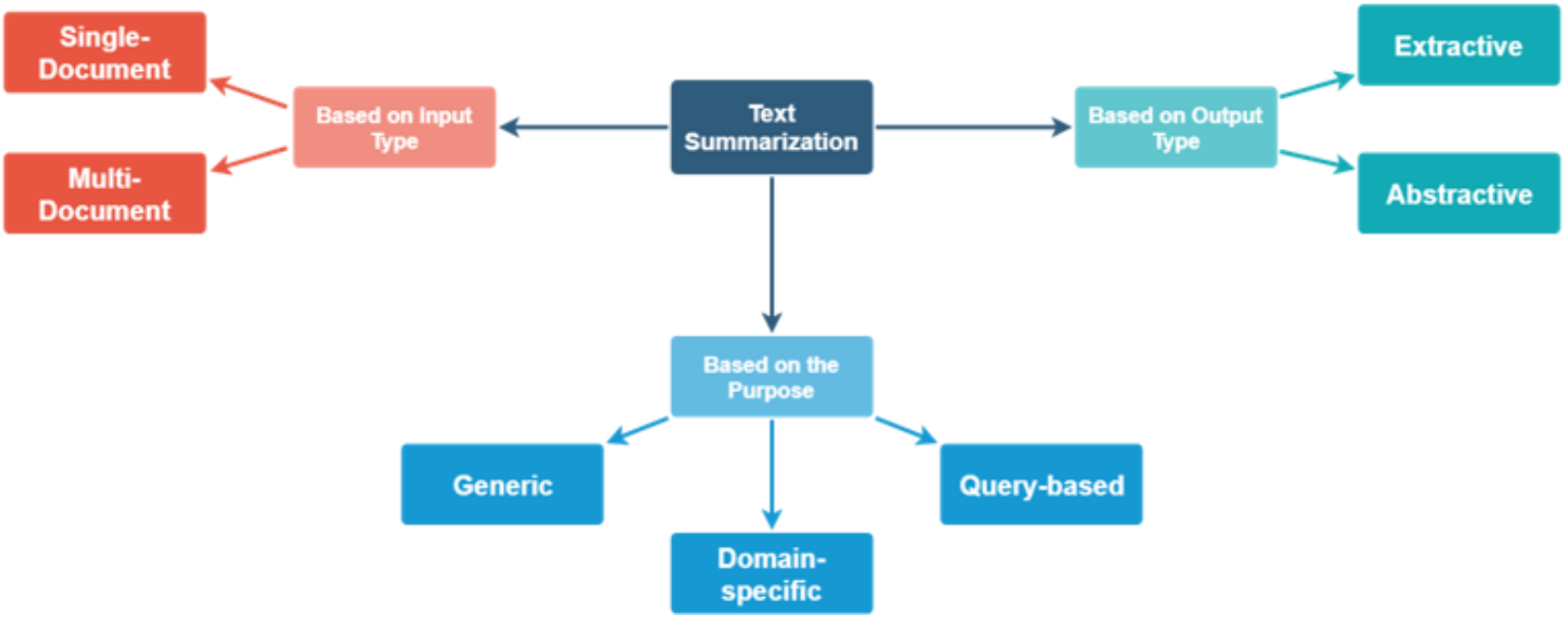
- using transformers
- softmax for output
- we maximize
  P(output sentence | input sentence)
- we sum errors on all outputs
- backpropagation
- training on correct translations
- as the translation, we return a sequence of words with the highest probability (not necessary greedily)

NMT with attention



Text summarization



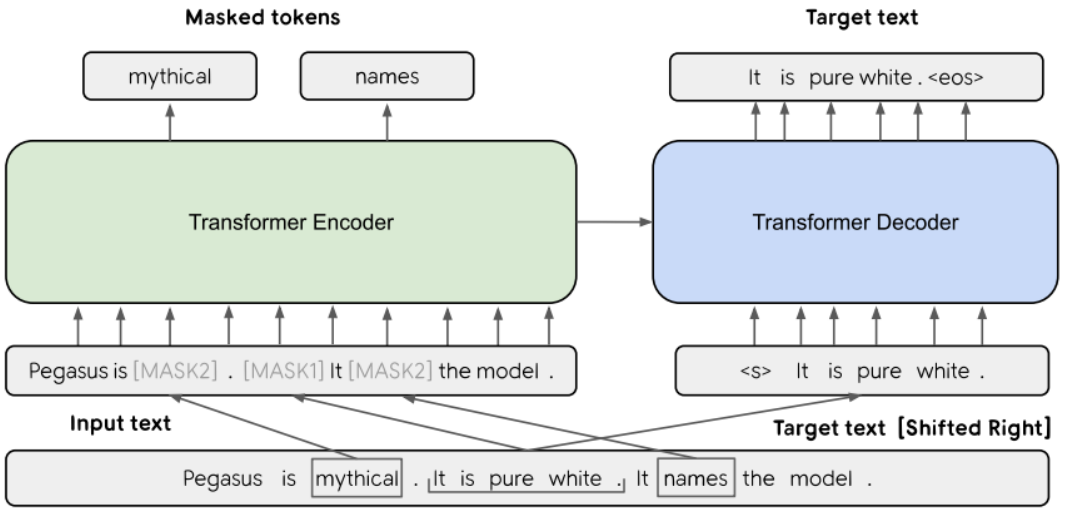


Text summarization

- Evaluation:
  - ROUGE scores,
  - BERTScore,
  - with QA:
    - question generation,
    - searching for answers in the summary
  - human
- LLMs
- Short and long texts
- cross-lingual

Summarizers – Pegasus

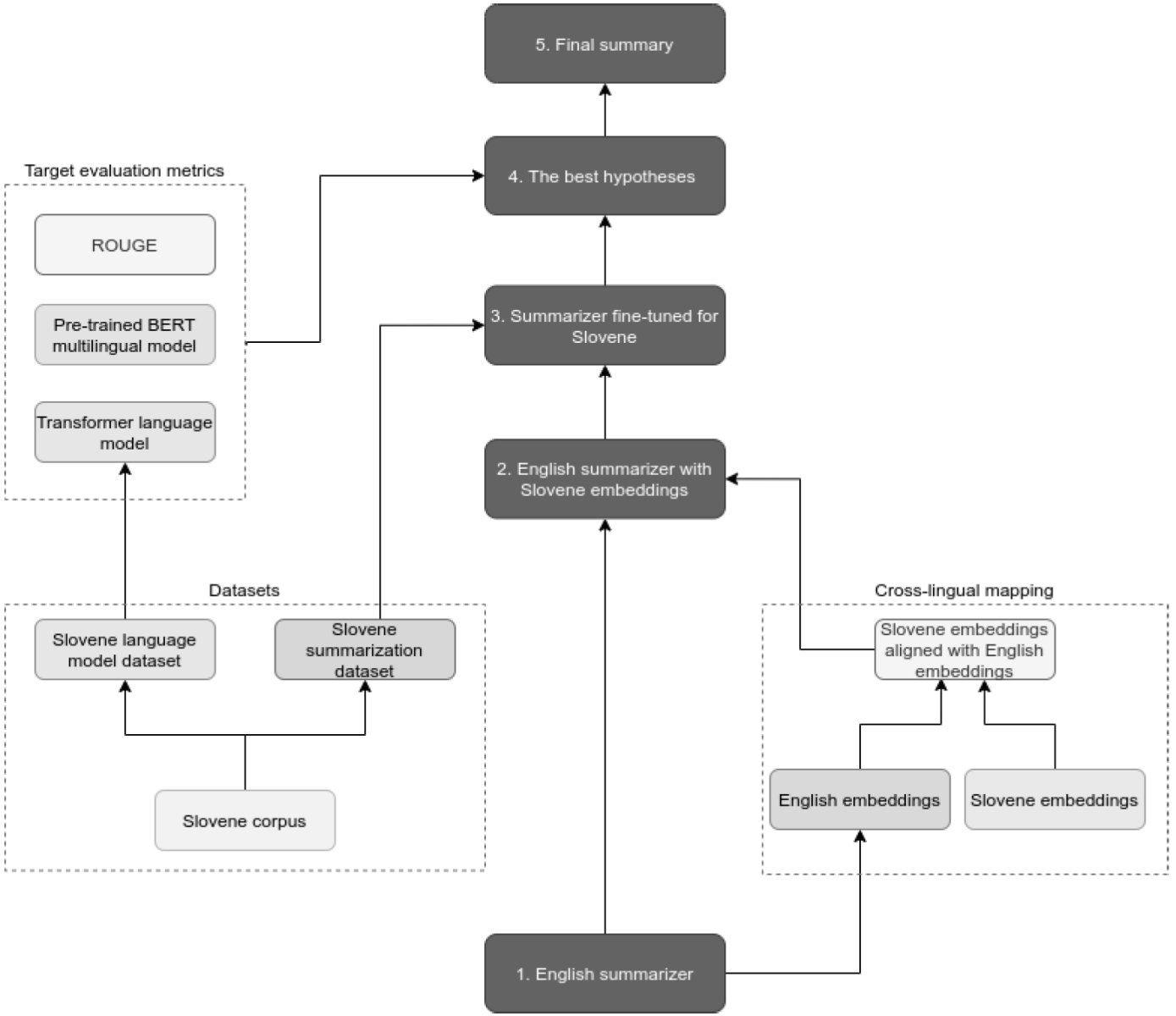
- An example of a different, successful transformer model
- Transformer BART model
- encoder-decoder architecture
- text garbling and reconstruction
- Auxiliary tasks: masked language model and missing sentence generation
- Demo: https://ai.googleblog.com/2020/06/pegasus-state-of-art-model-for.html



(Zhang et al., 2019)

Univerza v Ljubljani
Fakulteta za računalništvo in informatiko

XL summarization architecture



Žagar, A., & Robnik-Šikonja, M. (2022). Cross-lingual transfer of abstractive summarizer to less-resource language. Journal of Intelligent Information Systems, 58(1), 153-173.