

Natural Language Processing

Salar Mohtaj | DFKI

Natural Language Processing

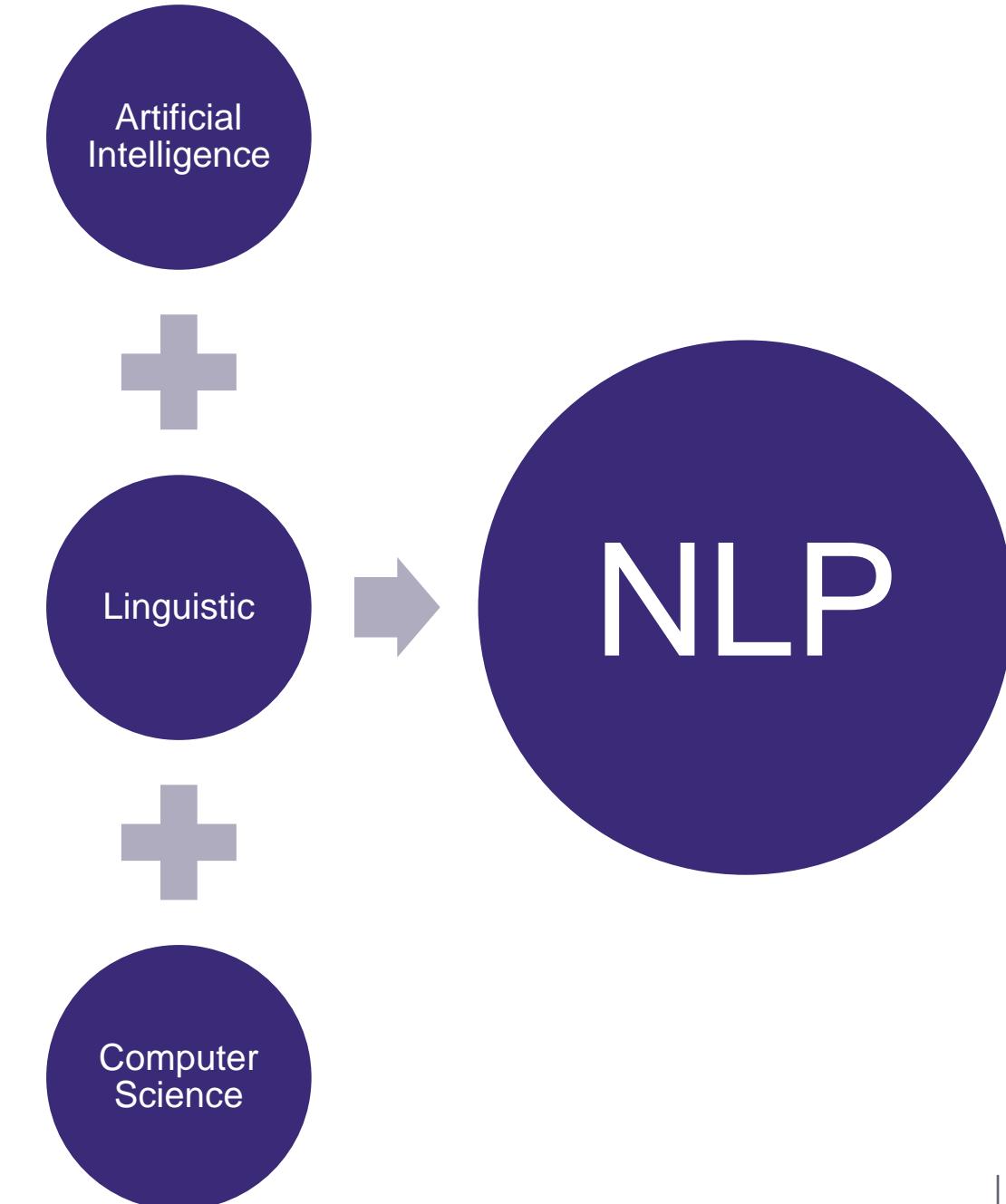
- Introduction to Natural Language Processing
- Everyday NLP applications
- Main NLP tasks
- Main approaches in NLP
- Who is this course for
- The Course structure
- NLP terminology

Natural Language Processing

- Introduction to Natural Language Processing
- Everyday NLP applications
- Main NLP tasks
- Main approaches in NLP
- Who is this course for
- The Course structure
- NLP terminology

Introduction to Natural Language Processing

- Natural language processing (NLP) is a branch of ***artificial intelligence*** that helps computers to understand, interpret and generate human language
- ***Natural language processing*** helps computers communicate with humans in their own language
- Most NLP techniques rely on machine learning to derive ***meaning*** from ***human languages***



Why is it difficult?

The hammer hit the glass and it broke!



Why is it difficult?

I saw someone on the hill with a telescope!



Images from www.thedailychain.com and www.storyblocks.com

Example from www.examples.yourdictionary.com

Why is it difficult?

- Ambiguity in language
 - The rules that dictate the passing of information using natural languages are not easy for computers to understand
 - Sarcastic remark

That's just what I needed today!

Why is it difficult?

- Ambiguity in language
 - The rules that dictate the passing of information using natural languages are not easy for computers to understand
 - Sarcastic remark
 - Multi meaning words

She will park the car so we can walk in the park.

The committee chair sat in the center chair.

Why is it difficult?

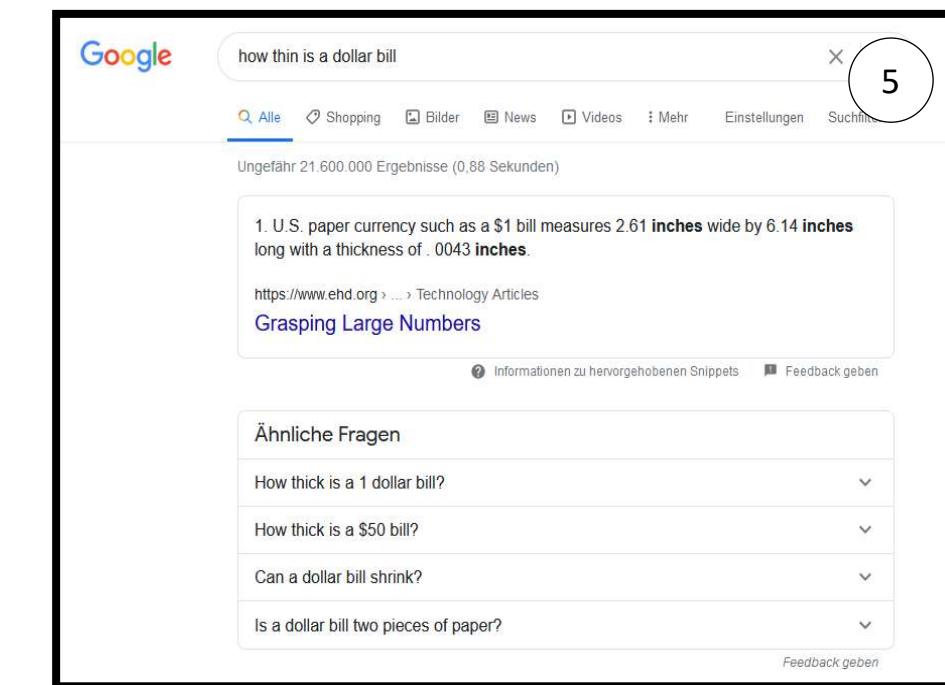
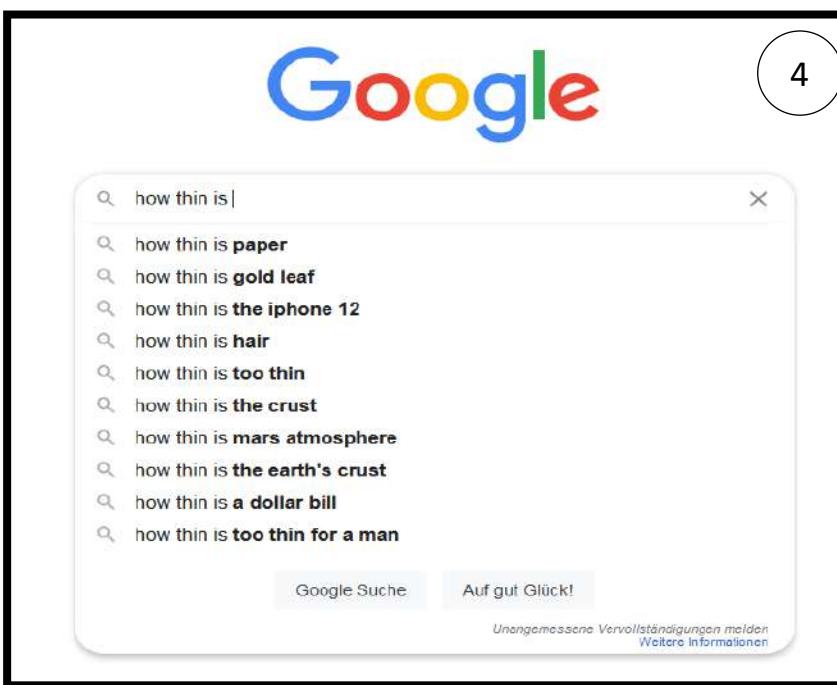
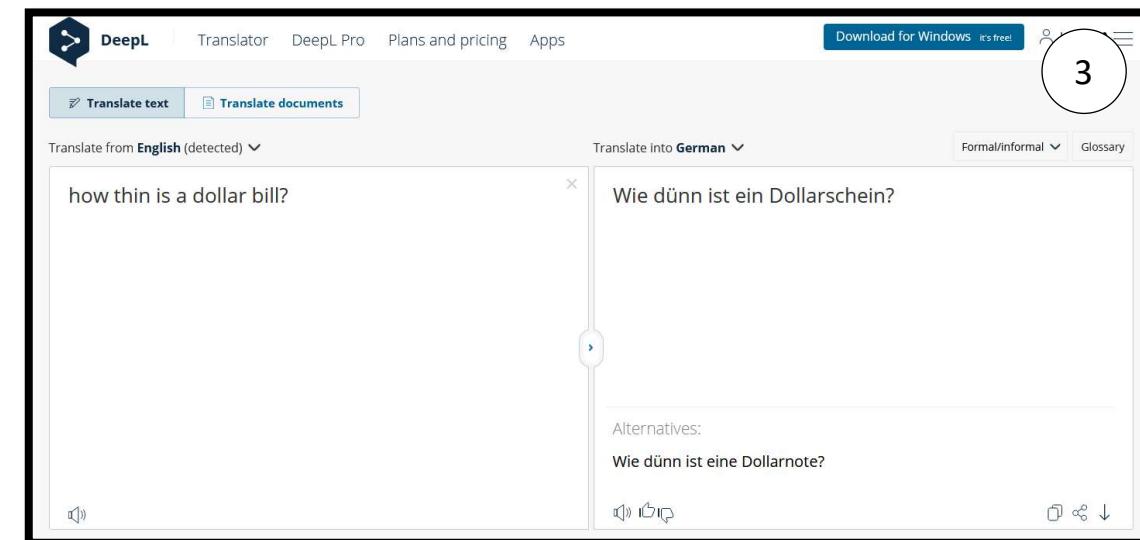
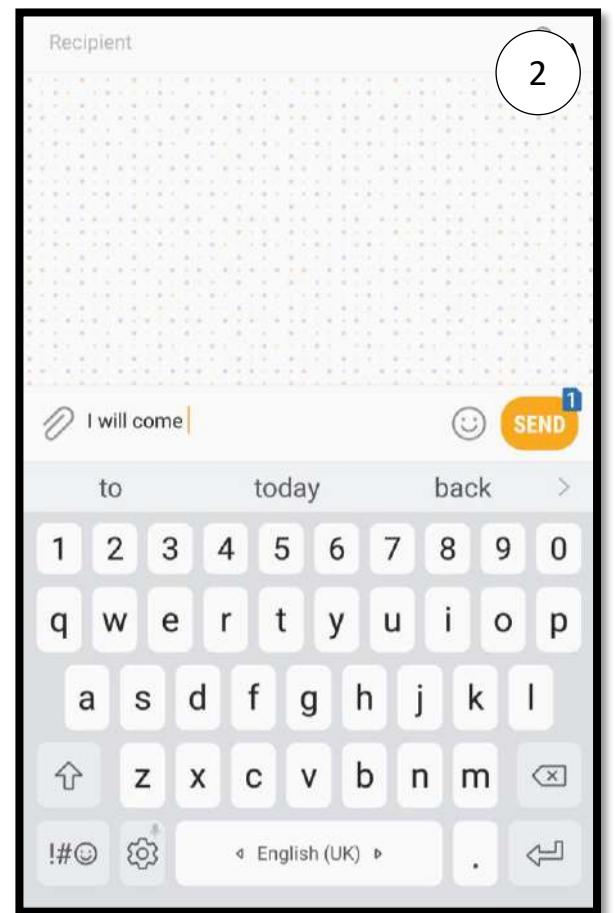
- Ambiguity in language
 - The rules that dictate the passing of information using natural languages are not easy for computers to understand
 - Sarcastic remark
 - Multi meaning words
- The lexicon of a language is usually enormous
 - Oxford dictionary has **273,000** headwords; **171,476** of them being in current use
 - An average person has a vocabulary range of about **20,000** to **35,000**

Natural Language Processing

- Introduction to Natural Language Processing
- Everyday NLP applications
- Main NLP tasks
- Main approaches in NLP
- Who is this course for
- The Course structure
- NLP terminology

Everyday NLP applications

- Email filters (spam detection) (1)
- Faster typing (2)
- Language translation (3)
- Question answering (4)+(5)
- Smart assistant devices (6)

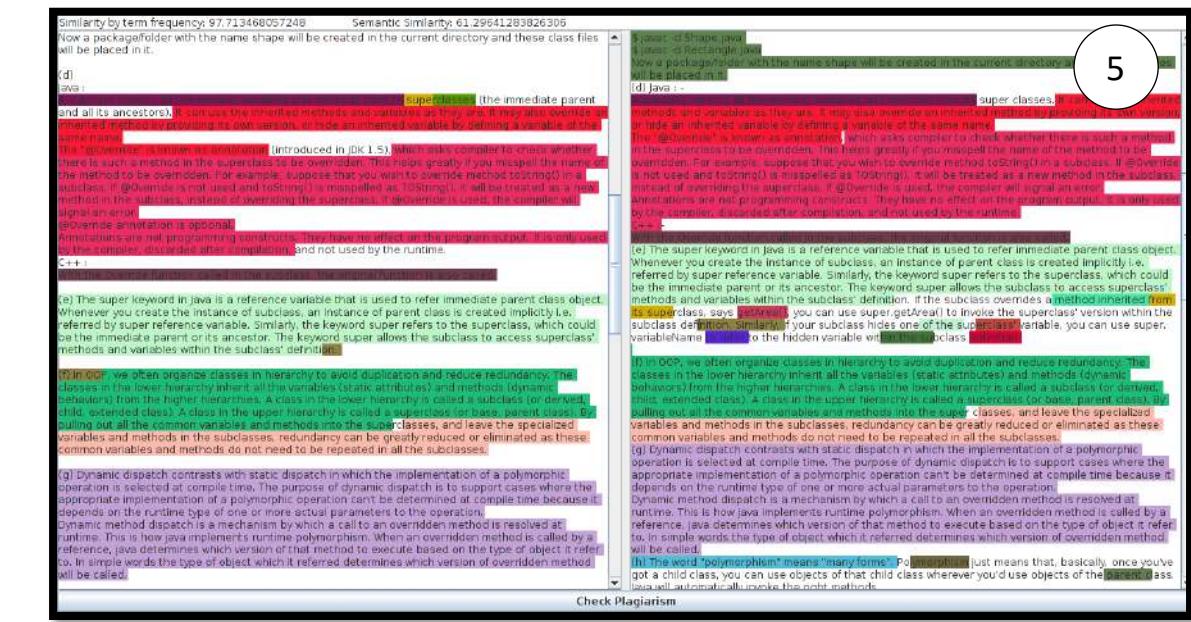
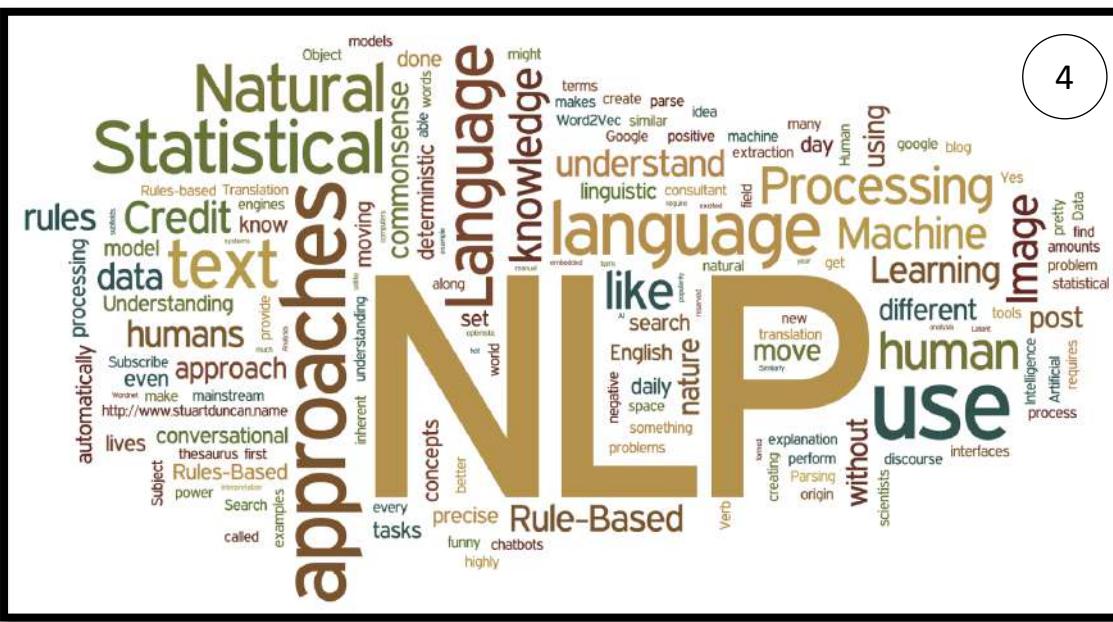
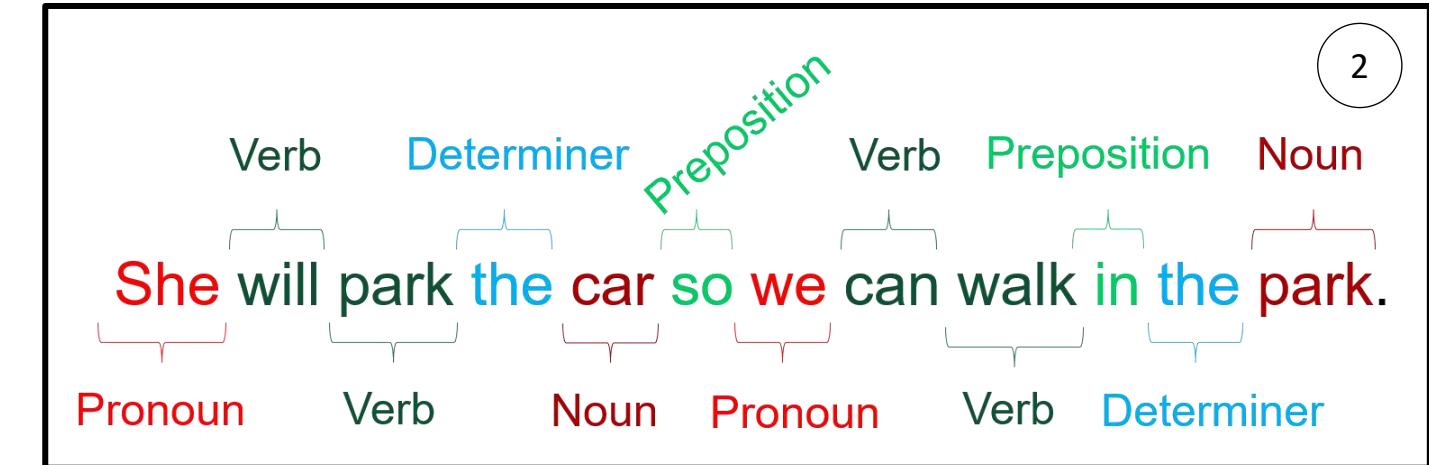


Natural Language Processing

- Introduction to Natural Language Processing
- Everyday NLP applications
- **Main NLP tasks**
- Main approaches in NLP
- Who is this course for
- The Course structure
- NLP terminology

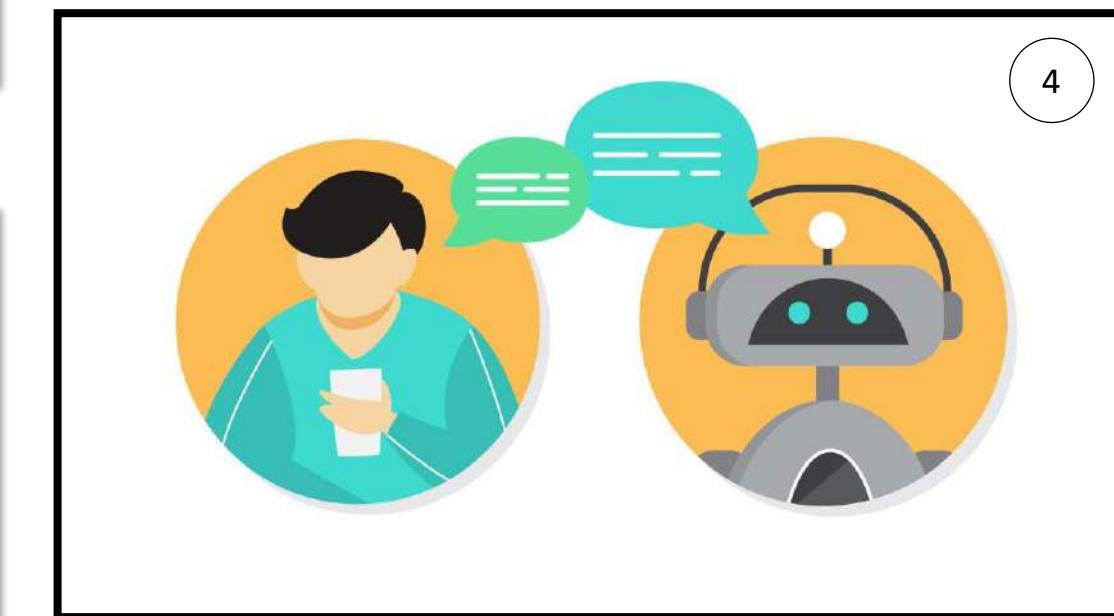
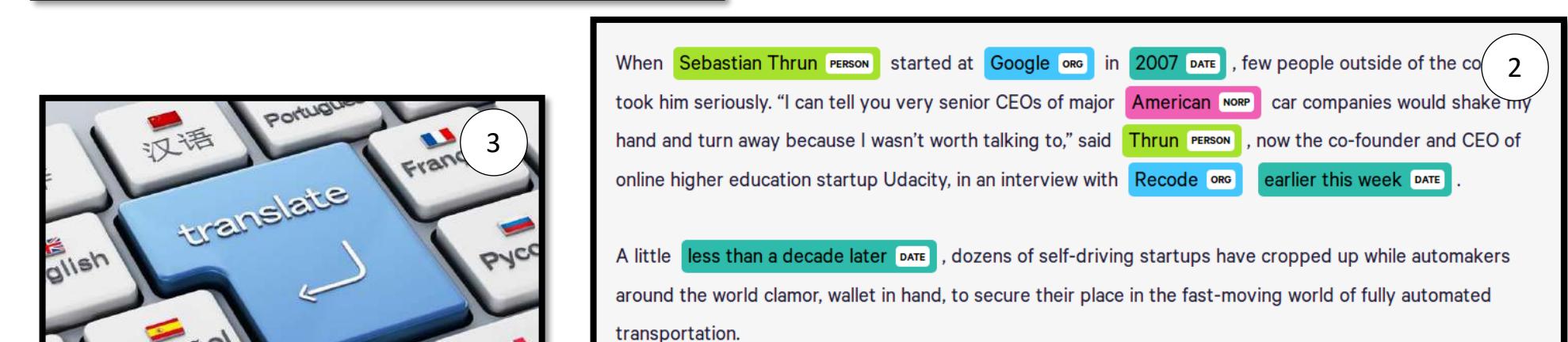
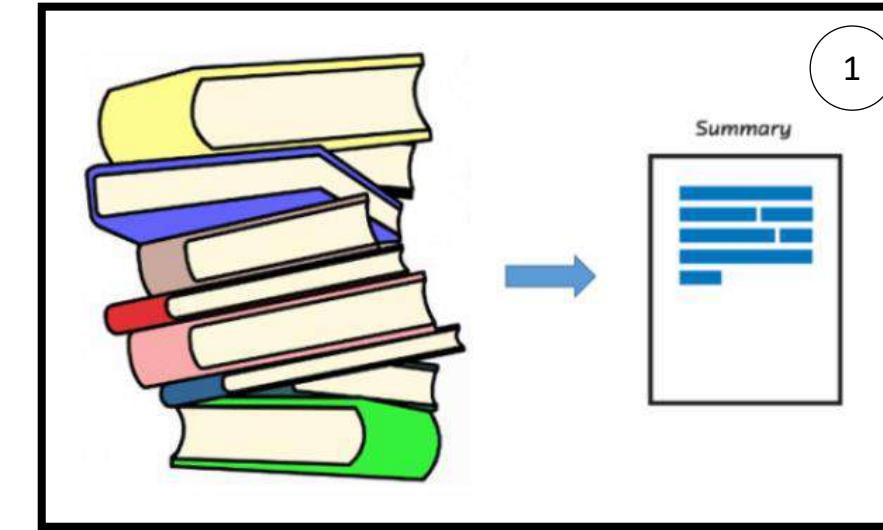
The main NLP tasks

- Text classification ⁽¹⁾
 - Parts of speech tagging ⁽²⁾
 - Sentiment analysis ⁽³⁾
 - Keyword extraction ⁽⁴⁾
 - Text similarity ⁽⁵⁾



The main NLP tasks

- Text summarization ⁽¹⁾
- Named entity recognition (NER) ⁽²⁾
- Machine translation ⁽³⁾
- Question answering ⁽⁴⁾
- Image captioning ⁽⁵⁾



Natural Language Processing

- Introduction to Natural Language Processing
- Everyday NLP applications
- Main NLP tasks
- **Main approaches in NLP**
- Who is this course for
- The Course structure
- NLP terminology

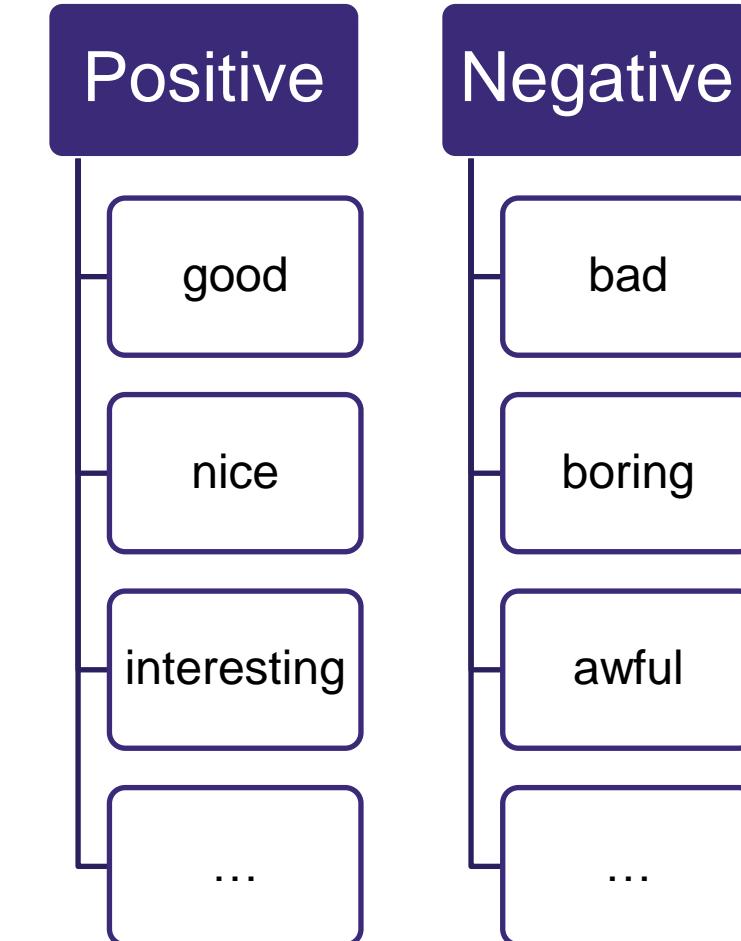
Main Approaches in NLP

- Rule based approaches
- Classical machine learning
- Deep learning

Main Approaches in NLP

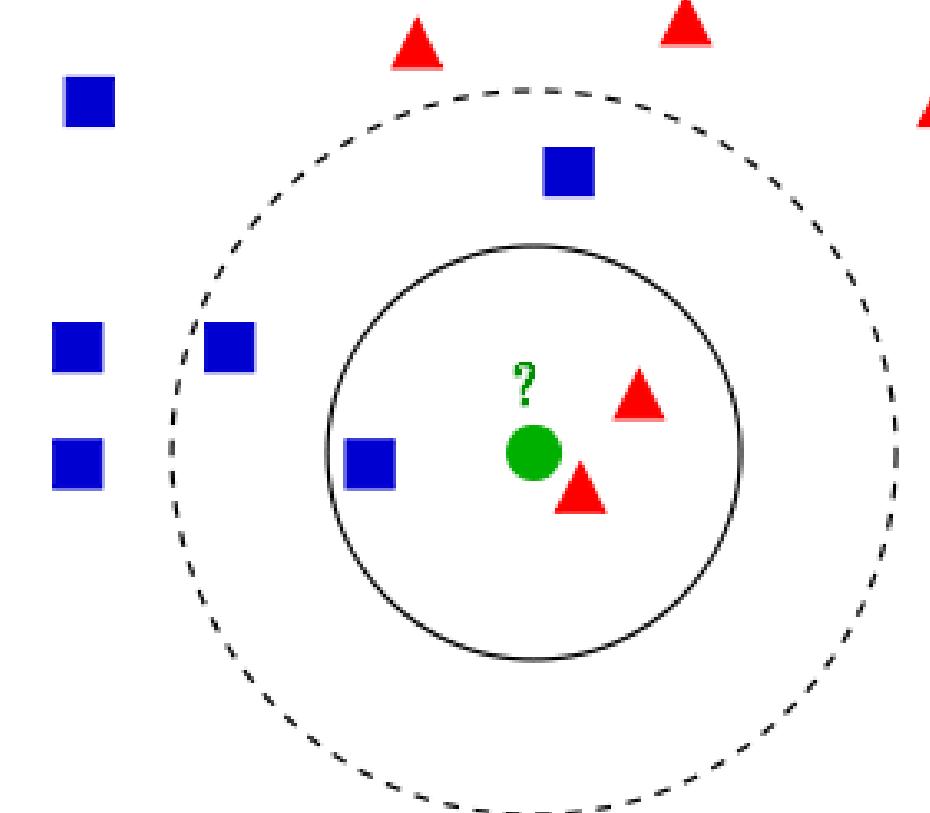
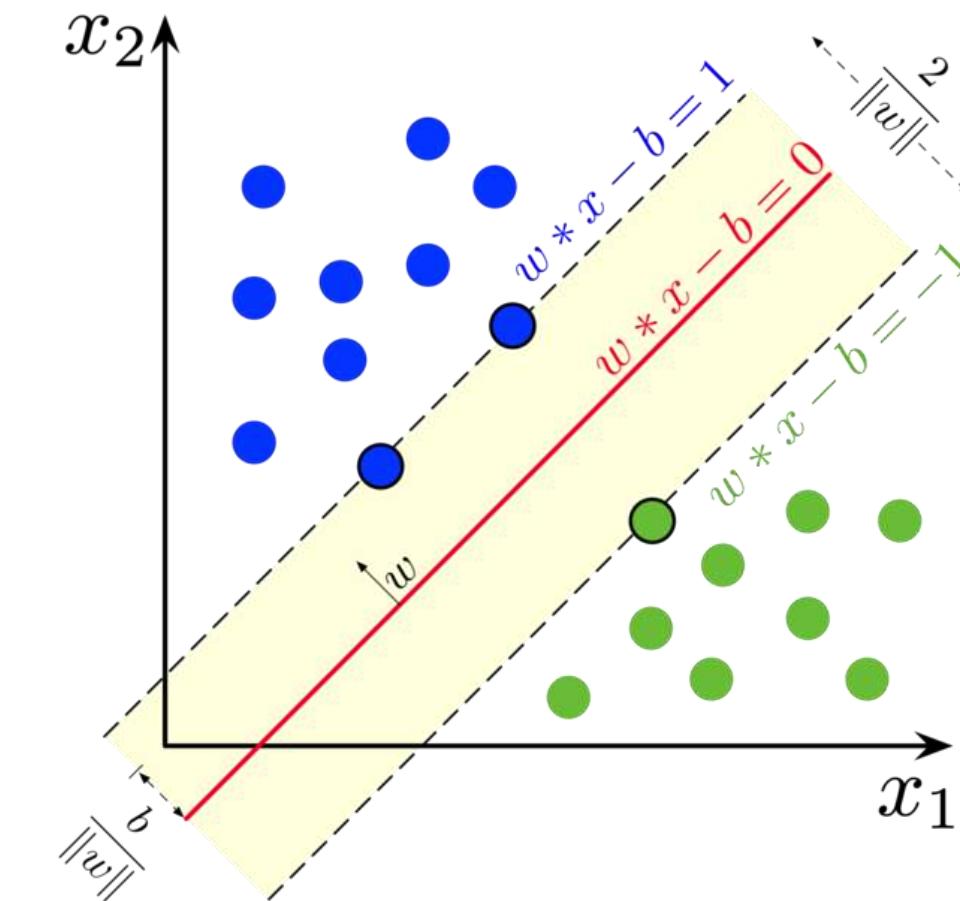
- Rule based approaches
 - Lack of enough accuracy

The film was good not bad.



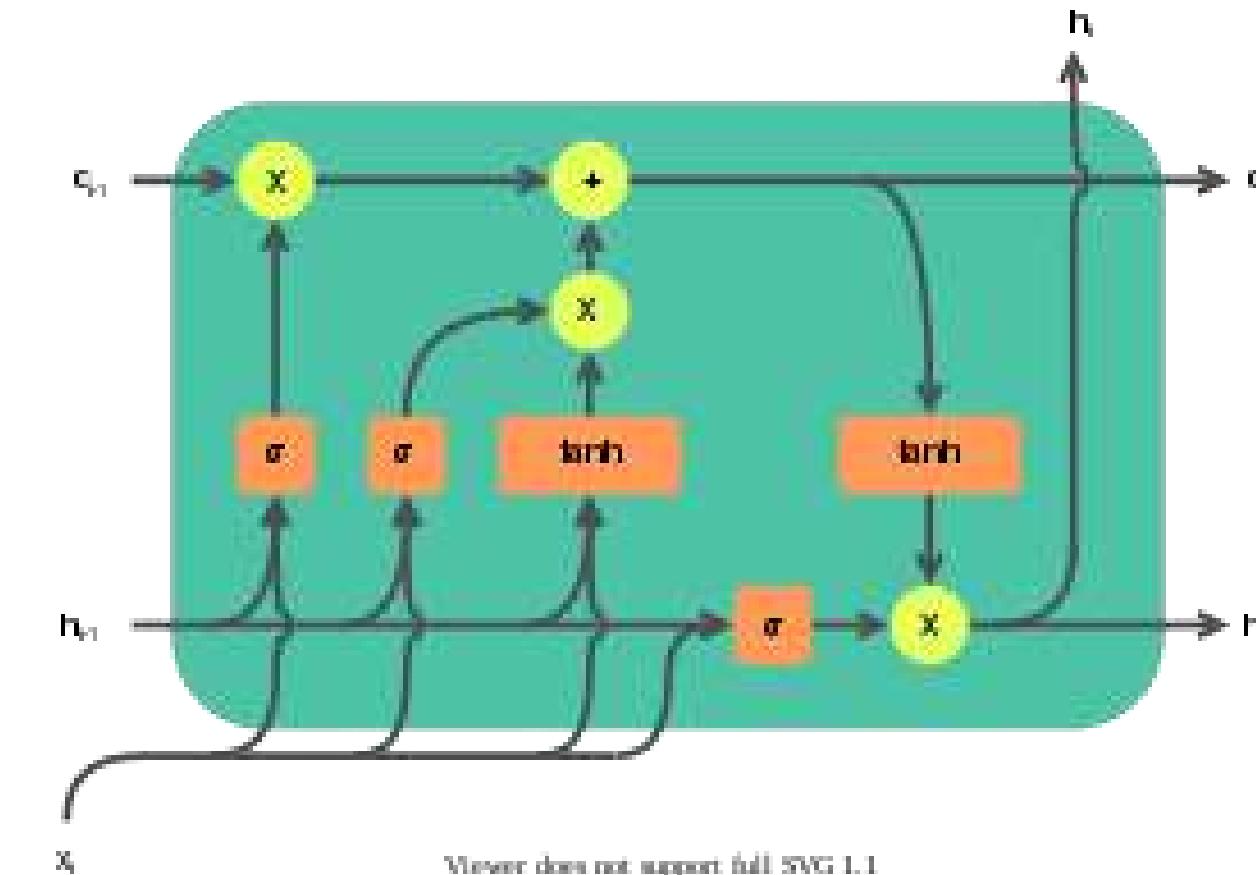
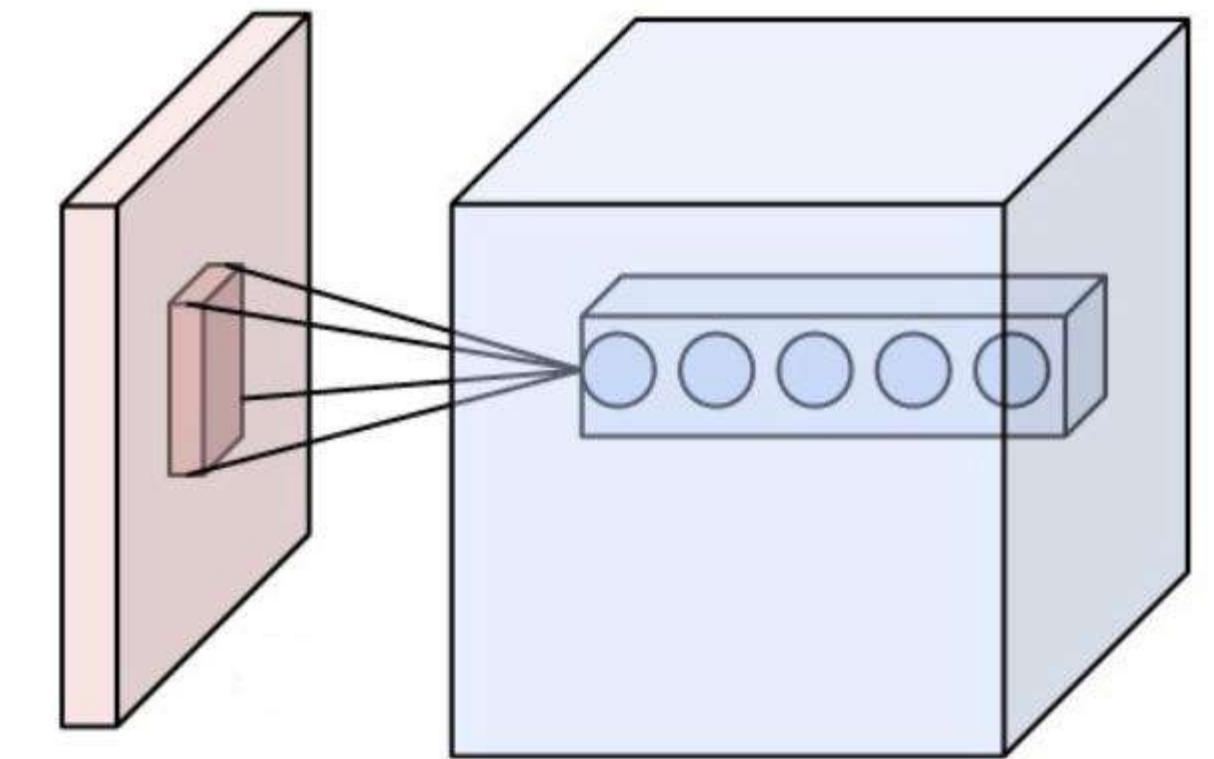
Main Approaches in NLP

- Rule based approaches
 - Lack of enough accuracy
- Classical machine learning
 - Training data
 - Feature engineering
 - Training a model



Main Approaches in NLP

- Rule based approaches
 - Lack of enough accuracy
- Classical machine learning
 - Training data
 - Feature engineering
 - Training a model
- Deep learning
 - More training data
 - Feature engineering is skipped
 - Training a model



Natural Language Processing

- Introduction to Natural Language Processing
- Everyday NLP applications
- Main NLP tasks
- Main approaches in NLP
- Who is this course for
- The Course structure
- NLP terminology

Who is this course for?

- Those who
 - don't want use NLP models as a black box
 - Review the state-of-the-art approaches
 - want to gain intuition for problem solving
 - you're asked to develop your own translation tool, what is the best approach
 - have a prior background on machine learning and deep learning



Supervised machine learning

Hidden layer

Backpropagation

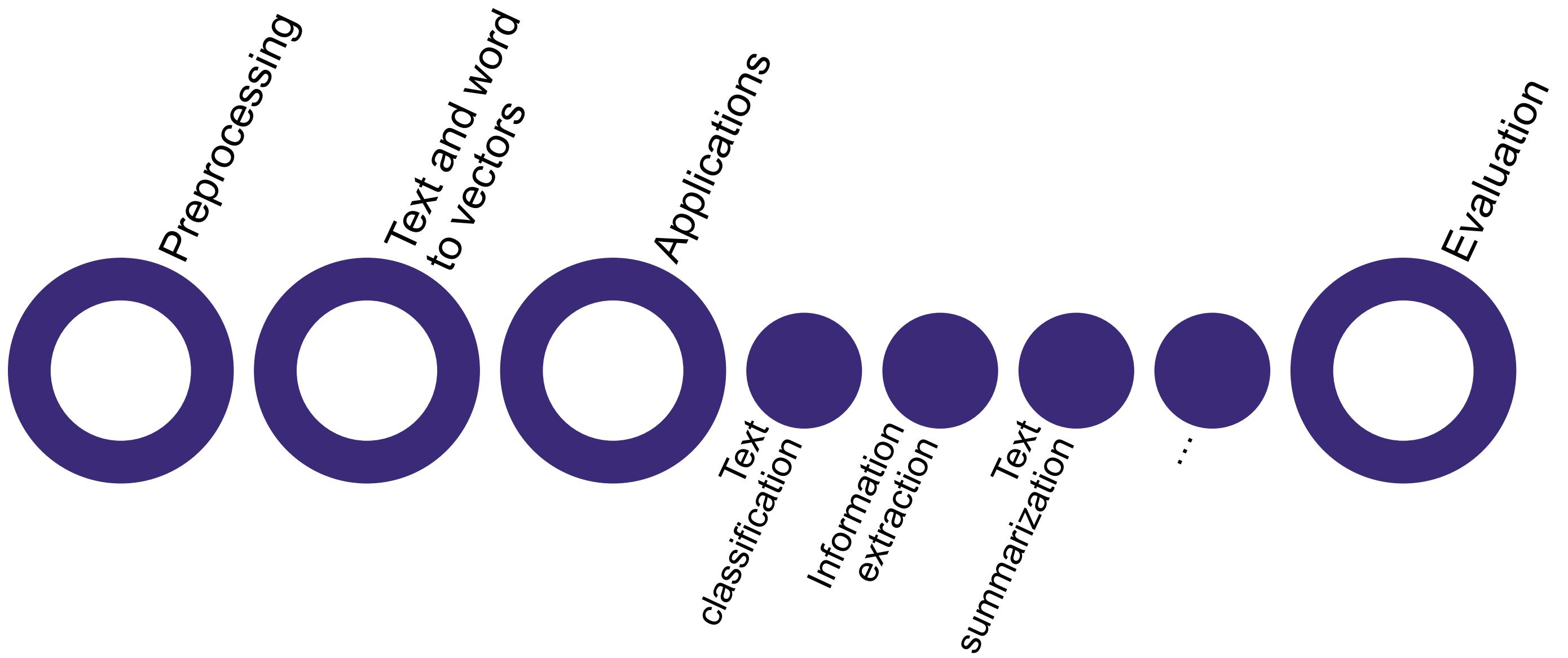
softmax

LSTM

Natural Language Processing

- Introduction to Natural Language Processing
- Everyday NLP applications
- Main NLP tasks
- Main approaches in NLP
- Who is this course for
- **The Course structure**
- NLP terminology

The Course structure

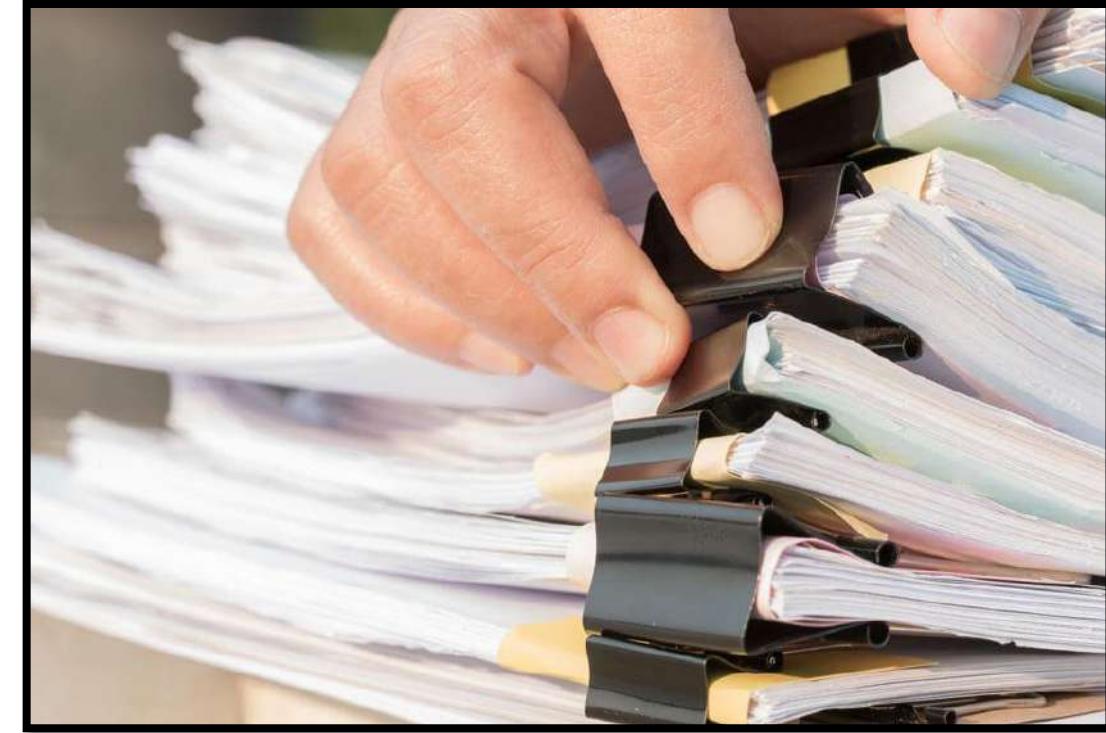


Natural Language Processing

- Introduction to Natural Language Processing
- Everyday NLP applications
- Main NLP tasks
- Main approaches in NLP
- Who is this course for
- The Course structure
- NLP terminology

NLP terminology

- Corpus (Plural: corpora)
 - A collection of text, usually contains several documents
 - Wikipedia articles
 - Collection of movies reviews
 - Internet comments
 - Collection of tweets
 - Corpora can be in a single language or multiple languages



NLP terminology

- Document
 - Document refers to a body of text in a corpus
 - A tweet in a twitter corpus
 - An email in a collection of emails
- Stop word
 - usually refers to the most common words in a language
 - Words like “***the***”, “***and***”, “***a***”, “***an***”, “***in***”

NLP terminology

- Vocabulary
 - The set of unique words used in the text corpus
 - Set of unique words which are used in all Wikipedia articles
- Out of Vocabulary (OOV)
 - Words that have not seen during the train, but in the test
 - We will encounter out of vocabulary terms when using our model for inference

Thank you!

„KI-Campus – Die Lernplattform für Künstliche Intelligenz“ ist ein Projekt von



www.ki-campus.org

Pre-processing

Salar Mohtaj | DFKI

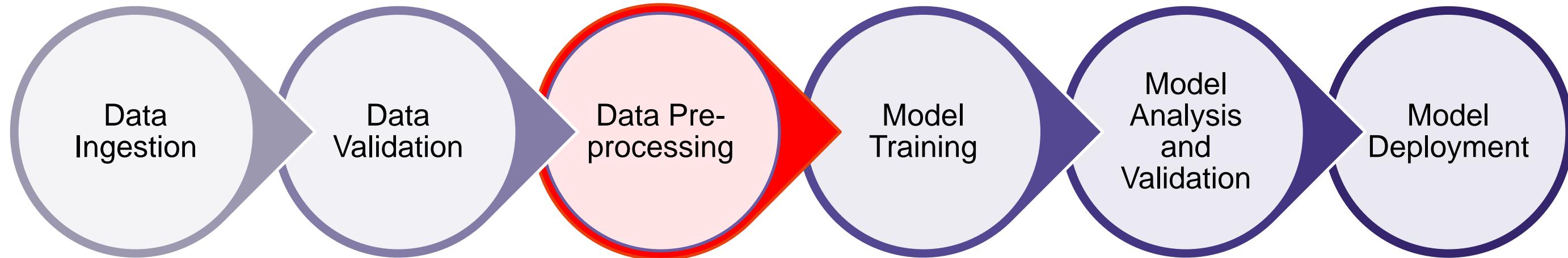
Text pre-processing

- What is text pre-processing?
- Why is it important?
- How to pre-process textual data?
- Python packages for text pre-processing

Text pre-processing

- What is text pre-processing?
- Why is it important?
- How to pre-process textual data?
- Python packages for text pre-processing

What is text pre-processing?



- To pre-process your text means to bring your text into a form that is **predictable** and **analyzable** for your task
- The steps and components are highly depend on the target task

Text pre-processing

- What is text pre-processing?
- Why is it important?
- How to pre-process textual data?
- Python packages for text pre-processing

Why text pre-processing is important?

- It transforms text into a more digestible form so that machine learning algorithms can perform better
- It helps to get rid of unhelpful parts of the data (e.g., noises and outliers)
- Some experiments show simple text pre-processing steps could lead to significant improvement of final results, even in complex deep neural models

Why text pre-processing is important?

- To illustrate the importance of text preprocessing, let's consider a couple of customer reviews
- Good → 71 111 111 100
- good → 103 111 111 100

The image displays two customer reviews for a pair of black pants. Both reviews are from anonymous users and have a rating of five stars.

Review 1: Anonymous, 5 stars. The review text is: "I got size L and I'm 1.95, fits very well. Good quality, fast delivery." A red box highlights the word "Good".

Review 2: Anonymous, 5 stars. The review text is: "good quality Fits really well Very happy with purchase". A red box highlights the word "good".

Both reviews include a "Helpful?" button below them. To the right of each review is a small image of a pair of black pants.

Text pre-processing

- What is text pre-processing?
- Why is it important?
- How to pre-process textual data?
- Python packages for text pre-processing

How to pre-process textual data?

- The most important pre-processing steps includes:
 - Tokenization and segmentation
 - Noise removal
 - Normalization

Tokenization / Segmentation

- **Tokenization** is essentially splitting a phrase, sentence, paragraph, or an entire text document into smaller units, such as **individual words or terms**
- Text **segmentation** is the process of dividing written text into meaningful units, such as words, **sentences**, or topics

You're watching an NLP course! I hope you find it interesting.

You 're watching an NLP course ! I hope you find it interesting .

You ' re watching an NLP course ! I hope you find it interesting .

You're watching an NLP Course! I hope you find it Interesting.

Tokenization / Segmentation

- Challenges
 - Multi token words (e.g., “New York”)
 - Continuous script languages
 - “.” doesn’t mean a sentence boundary in all sentences (segmentation)

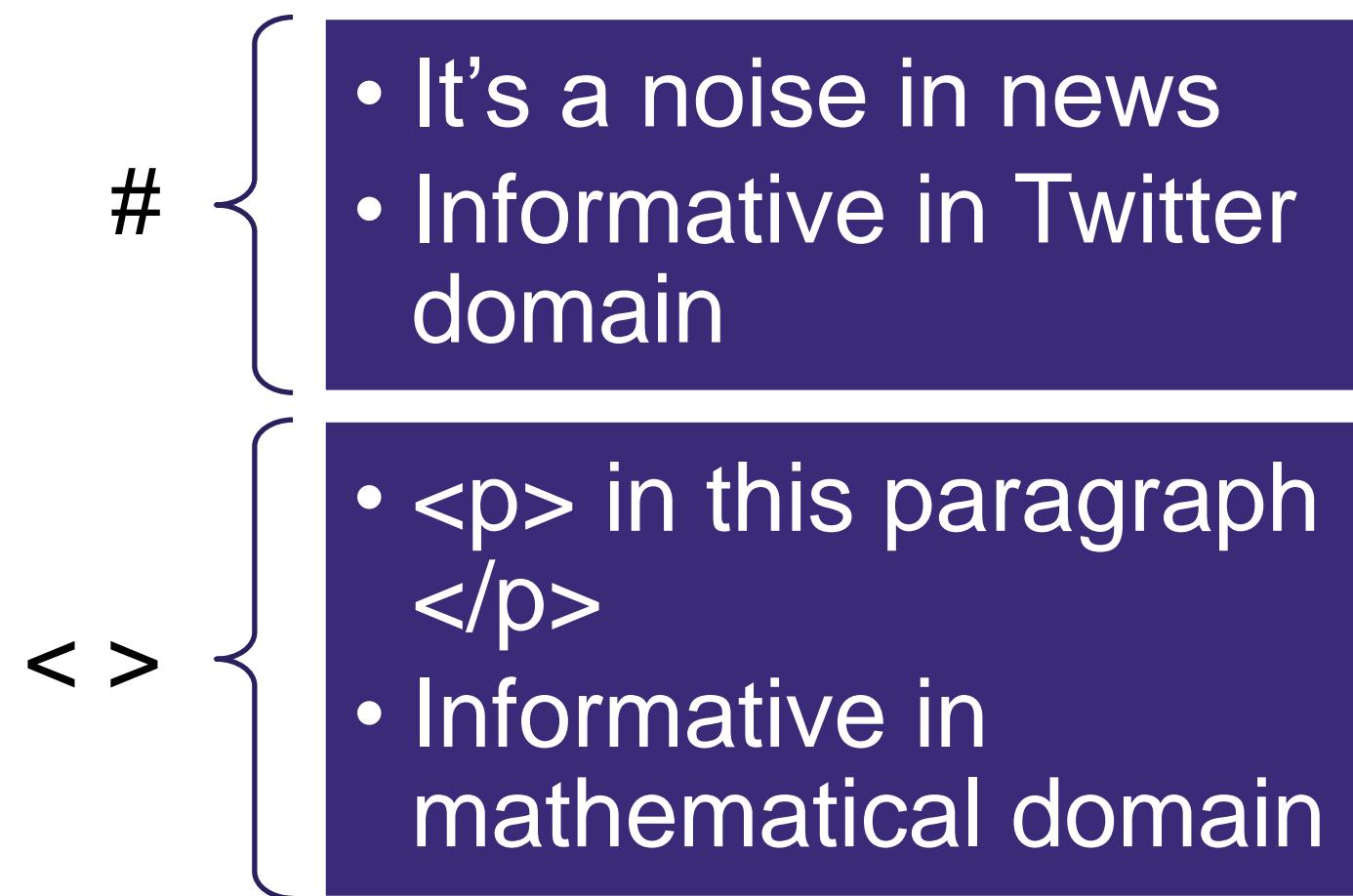
I am 1.75m tall, it was suit for me.

How to pre-process textual data?

- The most important pre-processing steps includes:
 - Tokenization and segmentation
 - Noise removal
 - Normalization

Noise removal

- **Noise removal** is about removing characters, digits and pieces of text that can interfere with your text analysis
- It's highly domain dependent



How to pre-process textual data?

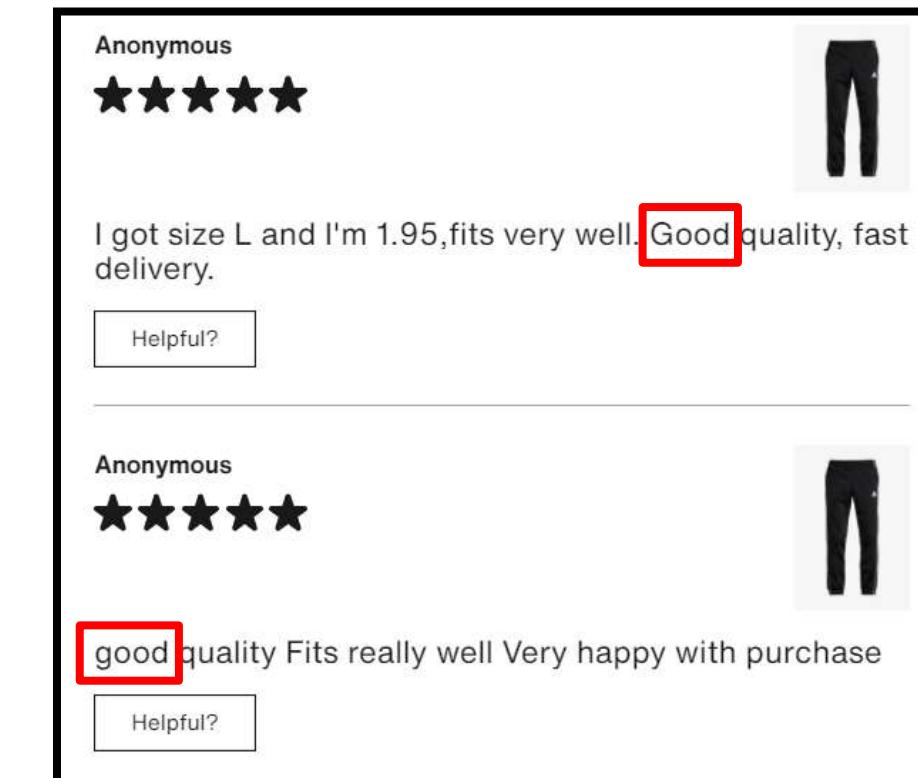
- The most important pre-processing steps includes:
 - Tokenization and segmentation
 - Noise removal
 - Normalization

Normalization

- **Text normalization** is the process of transforming text into a single canonical form that it might not have had before
 - Lower casing
 - Removing punctuation
 - Removing / Converting numbers to their word equivalents
 - Strip white space
 - Removing stop words
 - Stemming / Lemmatization

Normalization

- **Text normalization** is the process of transforming text into a single canonical form that it might not have had before
 - Lower casing



I have been living in Berlin for 5 years. → i have been living in berlin for 5 years.

Normalization

- **Text normalization** is the process of transforming text into a single canonical form that it might not have had before
 - Removing punctuation

I have been living in Berlin for 5 years. → I have been living in Berlin for 5 years

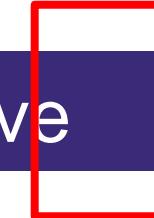
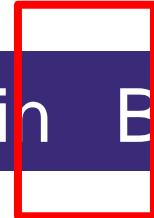
Normalization

- **Text normalization** is the process of transforming text into a single canonical form that it might not have had before
 - Removing / Converting numbers to their word equivalents

I have been living in Berlin for 5 years. → I have been living in Berlin for five years.

Normalization

- **Text normalization** is the process of transforming text into a single canonical form that it might not have had before
 - Strip white space

I have  been living in  Berlin for 5 years. → I have  been living  Berlin for 5 years.

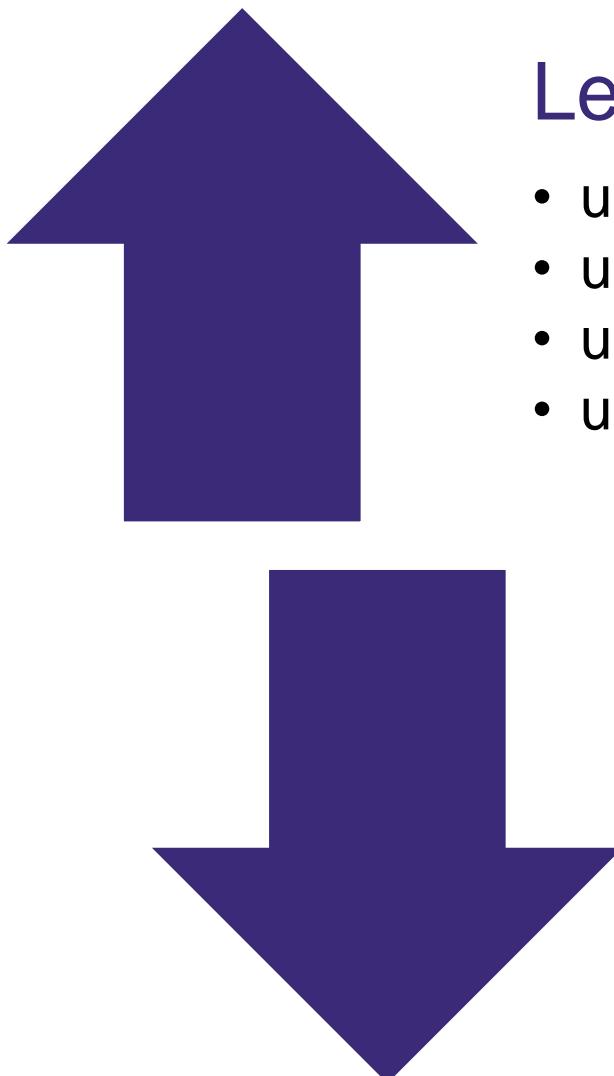
Stop word removal

- **Stop words** usually refers to the most common words in a language
 - Such as **the, and, at, a, and on**
- There is no single universal list of stop words used by all NLP tools
- Stop words could be general or task specific
 - e.g., “**editorial**” in news domain

I have been living in Berlin for 5 years. → I have been living in Berlin for 5 years.

Stemming / Lemmatization

- The goal of both ***stemming*** and ***lemmatization*** is to reduce inflectional forms of a word to a common base form
 - ***Stemming*** is the process of eliminating affixes (suffixed, prefixes, infixes, circumfixes) from a word in order to obtain a word stem
 - ***Lemmatization*** is related to stemming, differing in that lemmatization is able to capture canonical forms based on a word's lemma



Lemmatization

- universal → universal
- university → university
- universities → university
- universe → universe

Stemming

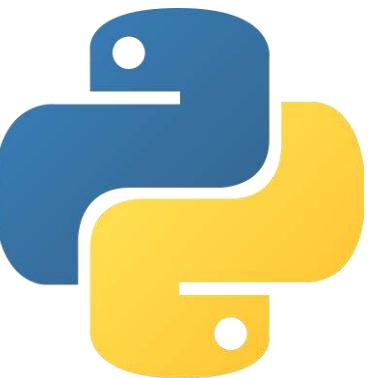
- universal → univers
- university → univers
- universities → univers
- universe → univers

Text pre-processing

- What is text pre-processing?
- Why is it important?
- How to pre-process textual data?
- Python packages for text pre-processing

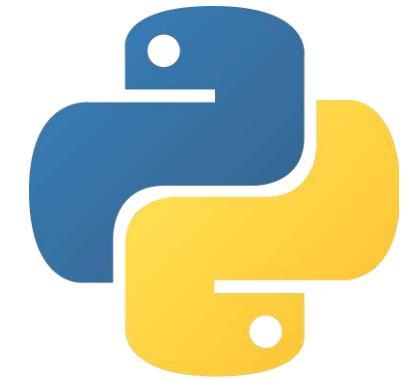
Python packages for text pre-processing?

- Python **built-in** methods
- **re** (regular expression)
- **spaCy**
- **NLTK**



Python built-in methods

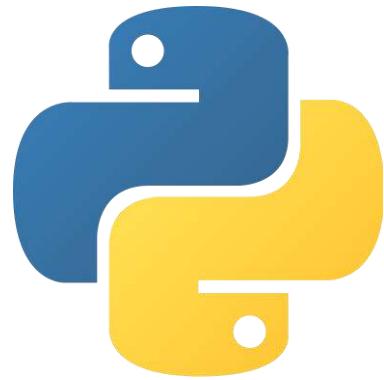
- Python includes lots of built-in methods to manipulate string in different shapes
 - upper()
 - lower()
 - title()
 - capitalize()



```
>>> string = "Sample String"
>>> lower_cased_string = string.lower()
>>> print(lower_cased_string)
sample string
>>> swap_cased_string = string.swapcase()
>>> print(swap_cased_string)
SAMPLE STRING
```

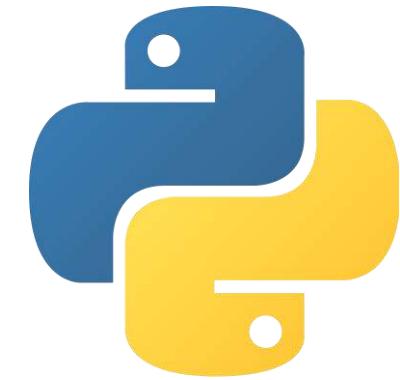
Python built-in methods

- Common text pre-processing use cases:
 - Text normalization
 - Convert case (upper, lower, capitalize, swap, ...)
 - Text tokenization (splitting text into sentence/words)
 - `split(sep=" ")`
 - `split(sep="!")`
 - Noise removal
 - `replace("#" , " ")`



re (Regular Expression)

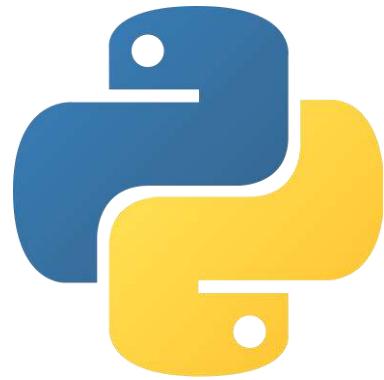
- RE are essentially a tiny, highly specialized programming language embedded inside Python
 - The most common use cases of REs are to find strings that match a pattern (e.g., email address and phone number validation)
 - Using this little language, you specify the rules for the set of possible strings that you want to match



```
>>> import re
>>> string = "a random. string? with.punctuation!"
>>> string = re.sub('([.,!?\(\)])', r' \1 ', string)
>>> print(string)
a random . string ? with . punctuation !
```

re (Regular Expression)

- Common text pre-processing use cases:
 - Text normalization
 - Padding punctuation with white spaces
 - Removing numbers, punctuation, ...
 - Replacing multiple spaces with a single space
 - Noise removal



spaCy

- spaCy is a modern Python library for industrial-strength Natural Language Processing
- The processing pipeline of spaCy includes lots of methods to preprocess and process textual input data

```
>>> import spacy  
>>> nlp = spacy.load("en_core_web_sm")  
>>> string = "it's a text"  
>>> doc = nlp(string)  
>>> for token in doc:  
        print(token)  
it  
's  
a  
text
```



spaCy

- Common text pre-processing use cases:
 - Text normalization
 - Stop words removal
 - Stemming and Lemmatization
- Text tokenization



NLTK

- **Natural language ToolKit (NLTK)** has lots of methods for pre-processing natural language text in python
- It's one of the earliest and also easiest NLP libraries that you'll use



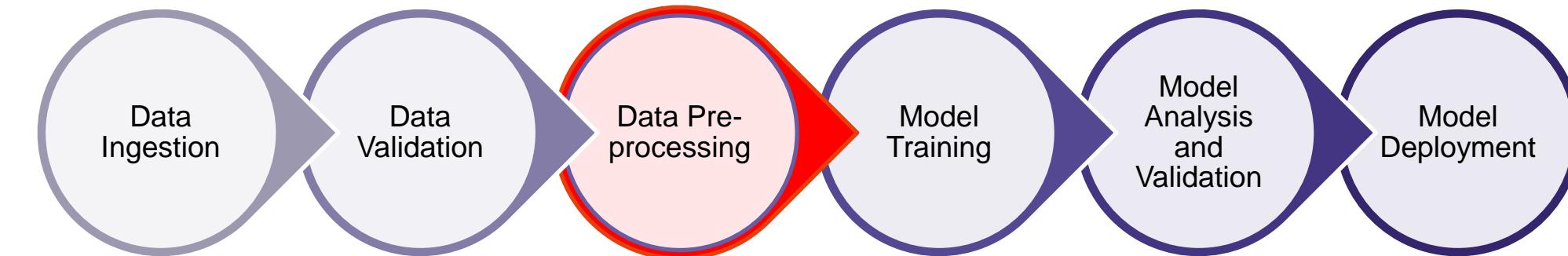
```
>>> from nltk.tokenize import sent_tokenize  
>>> string = "Today is great! The sun is in the sky."  
>>> print(sent_tokenize(string))  
  
[Today is great!', 'The sun is in the sky.']
```

NLTK

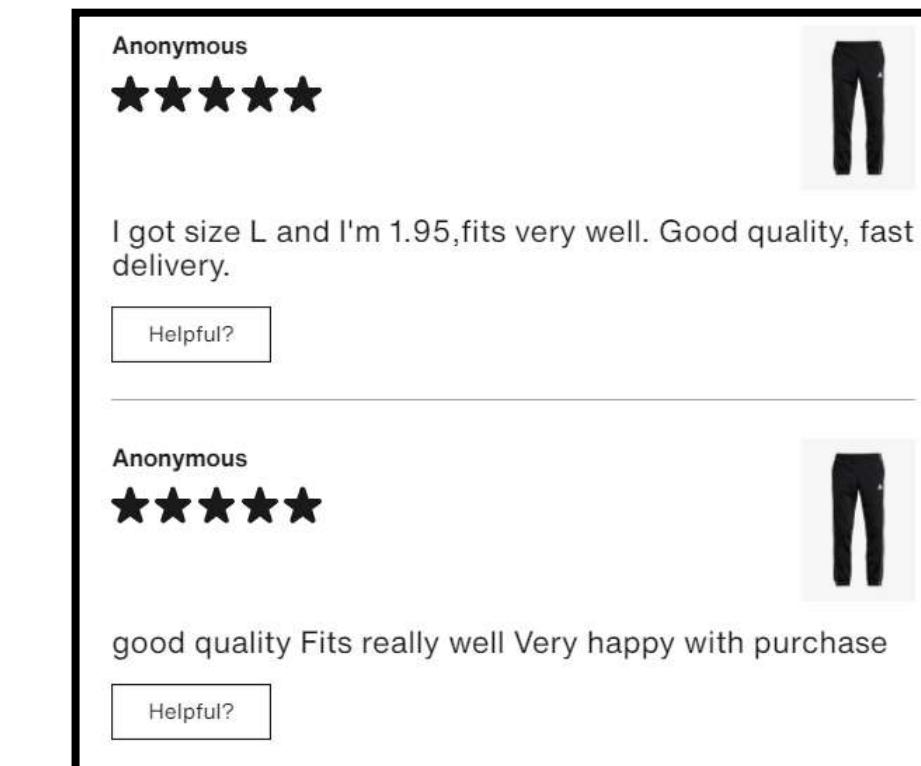
- Common text pre-processing use cases:
 - Text normalization
 - Punctuation removal
 - Stop words removal
 - Stemming and Lemmatization
 - Text tokenization



Summary



- Tokenization and segmentation
- Noise removal
- Normalization
 - Lower casing
 - Removing punctuation
 - Removing / Converting numbers to their word equivalents
 - Strip white space
 - Removing stop words
 - Stemming / Lemmatization



Vector Representation

Salar Mohtaj | DFKI

Vector representation

- What is vectorization?
- Sparse vs. dense vectors
- Text to vector
- Vector similarity measures
- Text vectorization in Python

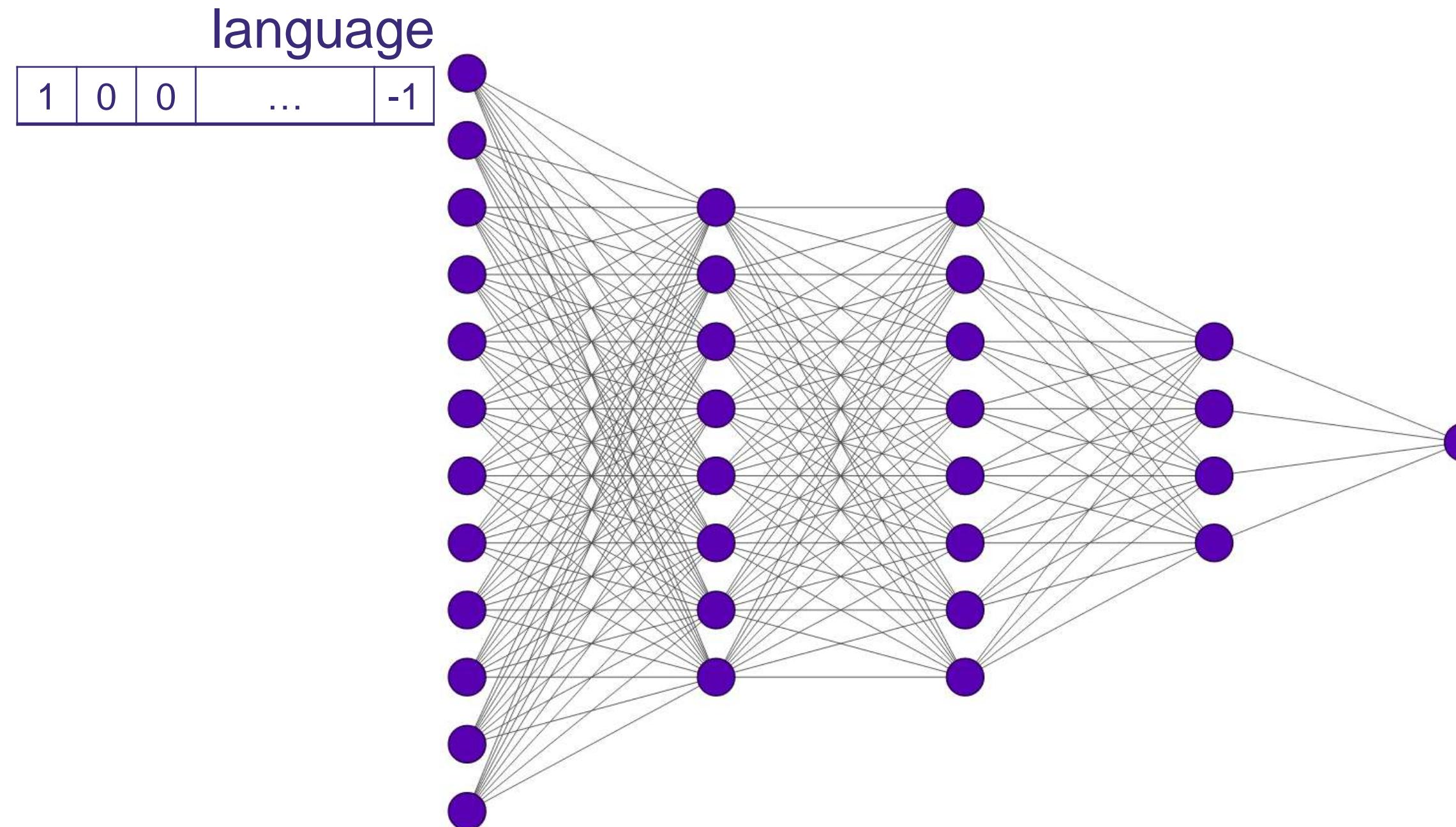
Vector representation

- What is vectorization?
- Sparse vs. dense vectors
- Text to vector
- Vector similarity measures
- Text vectorization in Python

What is vectorization?

- Machine learning algorithms and deep learning architectures are incapable of processing ***strings*** or ***plain text*** in their raw form
- **Vectorization** is the process of converting ***string*** or ***plain text*** into a ***vector of numbers***
- **Vectorization** is one of the basic buildings blocks in NLP, especially for neural networks

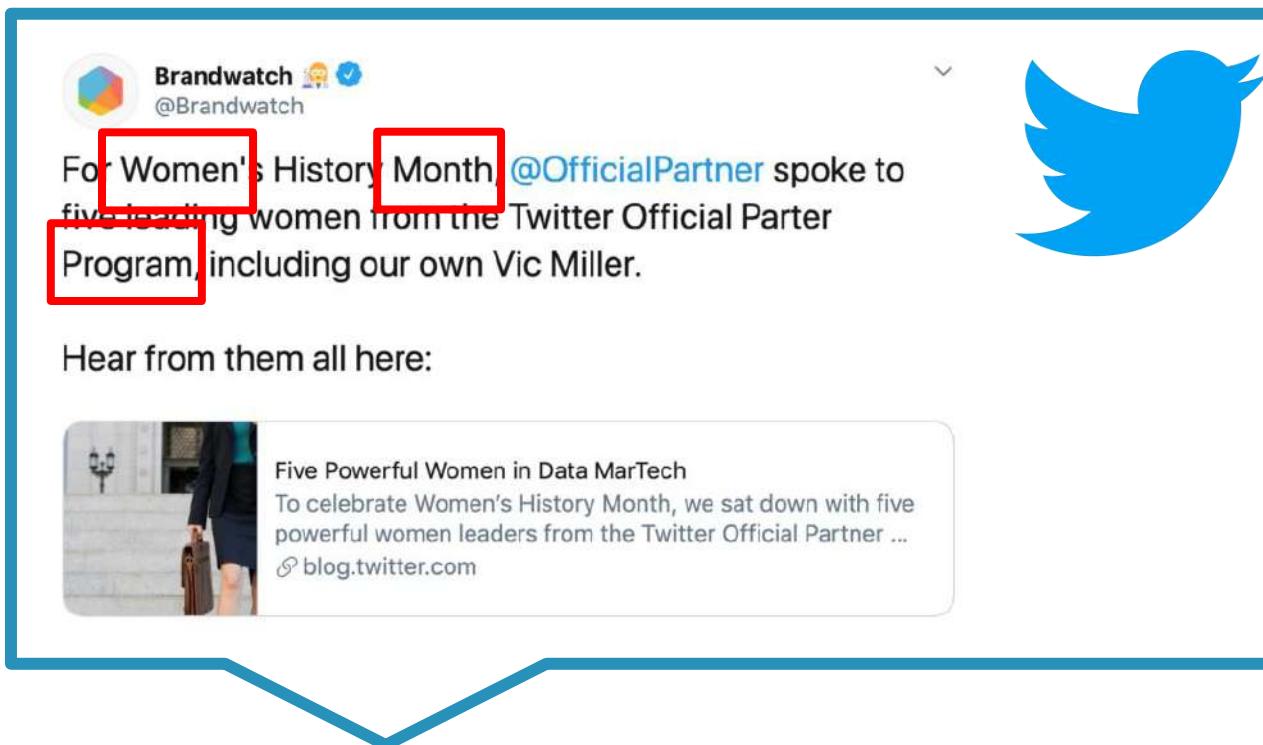
What is vectorization?



What is vectorization?

- Machine learning algorithms and deep learning architectures are incapable of processing ***strings*** or ***plain text*** in their raw form
- **Vectorization** is the process of converting ***string*** or ***plain text*** into a ***vector of numbers***
- **Vectorization** is one of the basic buildings blocks in NLP, especially for neural networks
- This process of converting ***text into vectors*** is called ***feature extraction*** or more simply, ***vectorization***

What is vectorization?



Word vectors

women

1	0	0	...	-1
---	---	---	-----	----

history

-1	1	0	...	1
----	---	---	-----	---

program

1	1	1	...	0
---	---	---	-----	---

...

0	0	0	...	0
---	---	---	-----	---

What is vectorization?



Brandwatch 🌐 ✅
@Brandwatch

For Women's History Month, [@OfficialPartner](#) spoke to five leading women from the Twitter Official Partner Program, including our own Vic Miller.

Hear from them all here:

Five Powerful Women in Data MarTech
To celebrate Women's History Month, we sat down with five powerful women leaders from the Twitter Official Partner ...
[blog.twitter.com](#)

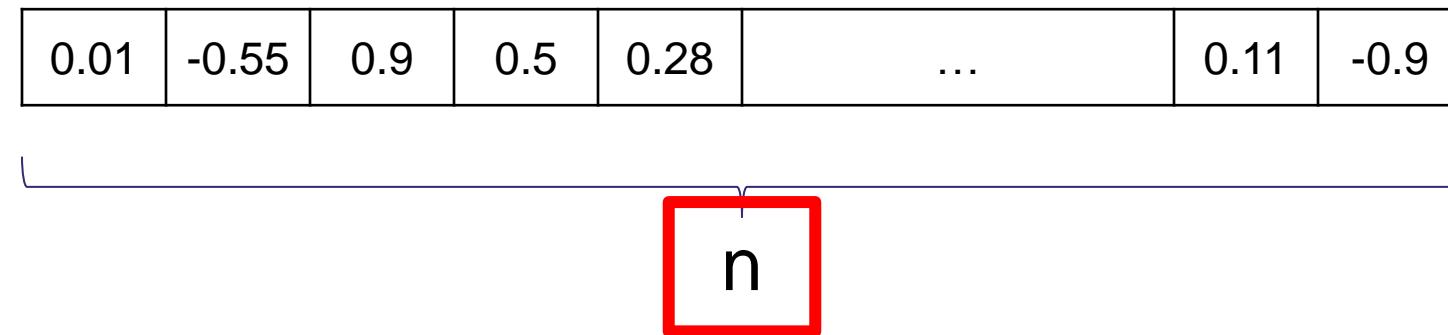
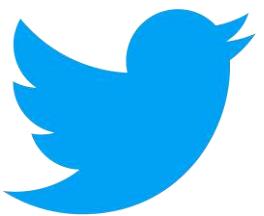
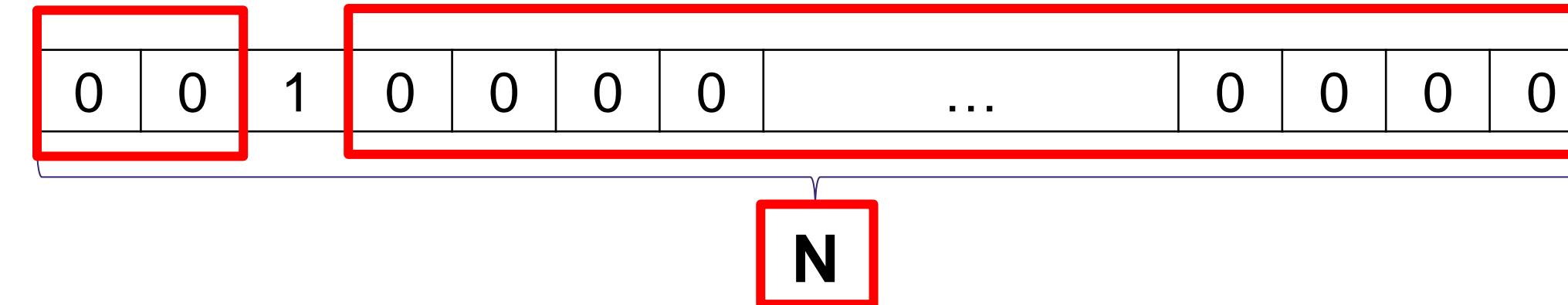
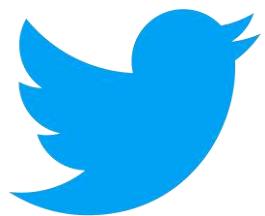
Text vectors

1	-1	0	1	0	0	...	-1
---	----	---	---	---	---	-----	----

Vector representation

- What is vectorization?
- Sparse vs. dense vectors
- Text to vector
- Vector similarity measures
- Text vectorization in Python

Sparse vs. dense vectors

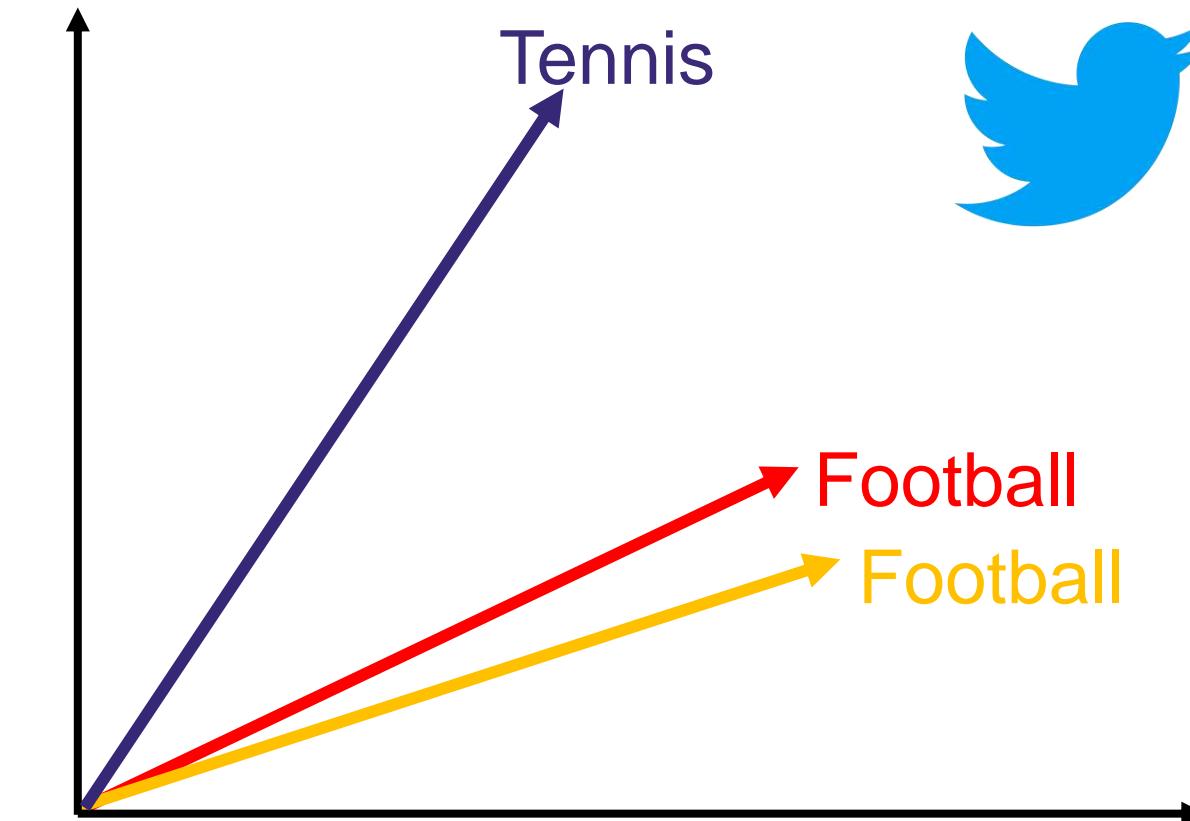
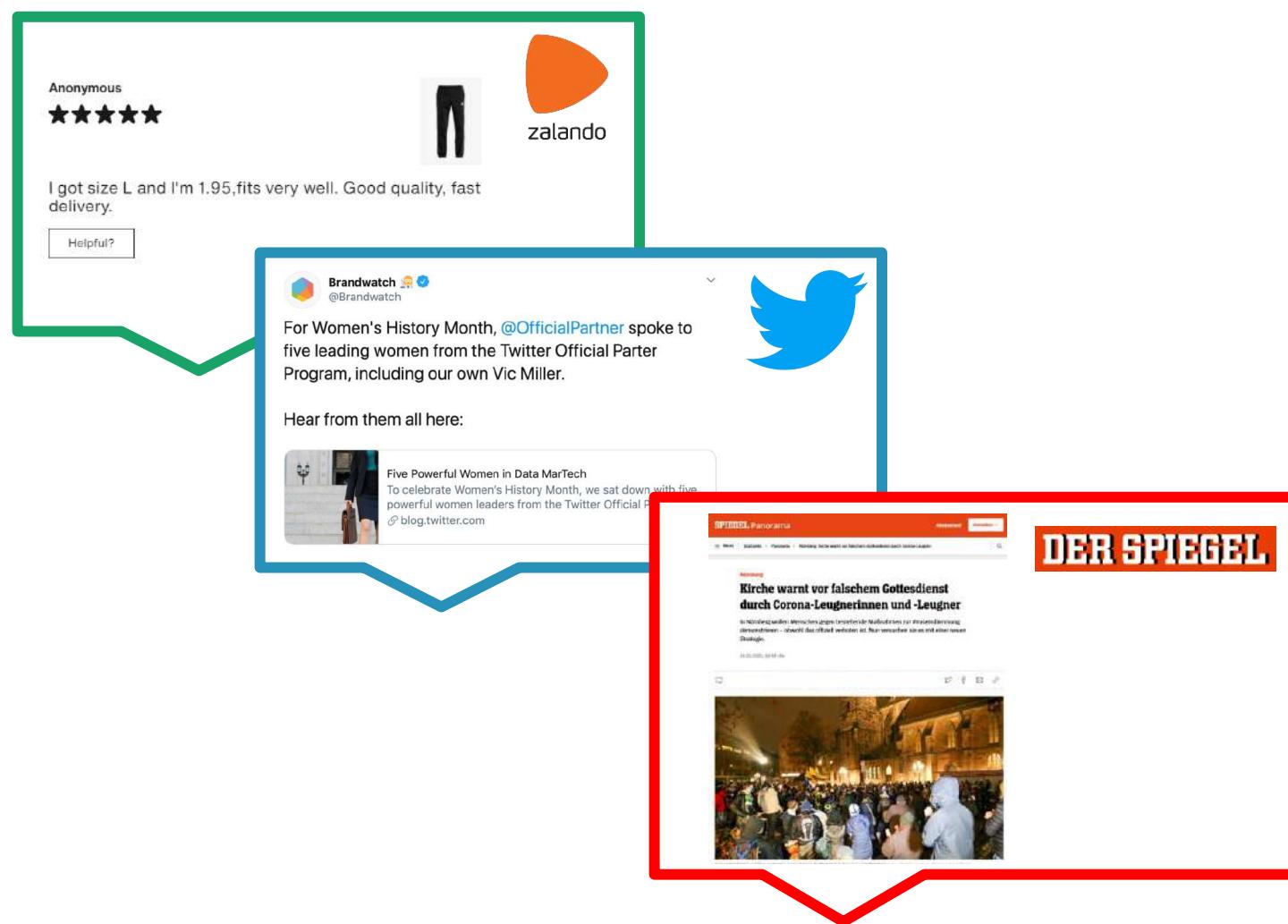


Vector representation

- What is vectorization?
- Sparse vs. dense vectors
- **Text to vector**
- Vector similarity measures
- Text vectorization in Python

Text vectorization

- Converting textual documents into machine readable vector of numbers
- Main objective: **similar text must result in closer vector**



Text vectorization

- D_1 = “Text is a complex human language representation.”
 - D_2 = “Natural human language is complex and also is diverse.”
 - D_3 = “Natural human body clock is complex.”
 - D_4 = “Text representation differs from human to human.”
-
- ***Similar*** text must result in ***closer*** vector

Text vectorization

- Bag of Words (BoW)
- Bag of N-Gram
- TF-IDF

Bag of Words (BoW)

- The main idea is that similar documents contain similar terms
- It counts how many times a word appears in a document
- Why it is called **bag of words?**
 - Because any order of the words in the document is discarded

Bag of Words (BoW)

- D_1 = “Text is a complex human language representation.”
 - D_2 = “Natural human language is complex and also is diverse.”
 - D_3 = “Natural human body clock is complex.”
 - D_4 = “Text representation differs from human to human.”
-
- Preprocessing
 - Lowercasing
 - Punctuation removal
 - Word tokenization
 - Stop word removal

clock
is
human
language
by
natural
a
diverse
to
text
differs
representation
of
complex
body
and
also

Bag of Words (BoW)

- D_1 = “Text is a complex human language representation.”
- D_2 = “Natural human language is complex and also is diverse.”
- D_3 = “Natural human body clock is complex.”
- D_4 = “Text representation differs from human to human.”

clock
is
human
language
natural
diverse
text
differs
representation
complex
body

clock
is
human
language
by
natural
a
diverse
to
text
differs
representation
of
complex
body
and
also

	clock	is	human	language	natural	diverse	text	differs	representation	complex	body
D_1	0	1	1	1	0	0	1	0	1	1	0
D_2	0	2	1	1	1	1	0	0	0	1	0
D_3	1	1	1	0	1	0	0	0	0	1	1
D_4	0	0	2	0	0	0	1	1	1	0	0

Bag of Words (BoW)

- Pros
 - Simple and easy to understand
- Cons
 - Ignores the location information of words



Forrest Gump is better than Shawshank redemption
Shawshank redemption is better than Forrest Gump

Bag of Words (BoW)

- Pros
 - Simple and easy to understand
- Cons
 - Ignores the location information of words
 - The intuition that high frequency words are more important fails in many cases

	clock	is	human	language	natural	diverse	text	differs	representation	complex	body
D ₁	0	1	1	1	0	0	1	0	1	1	0
D ₂	0	2	1	1	1	1	0	0	0	1	0
D ₃	1	1	1	0	1	0	0	0	0	1	1
D ₄	0	0	2	0	0	0	1	1	1	0	0

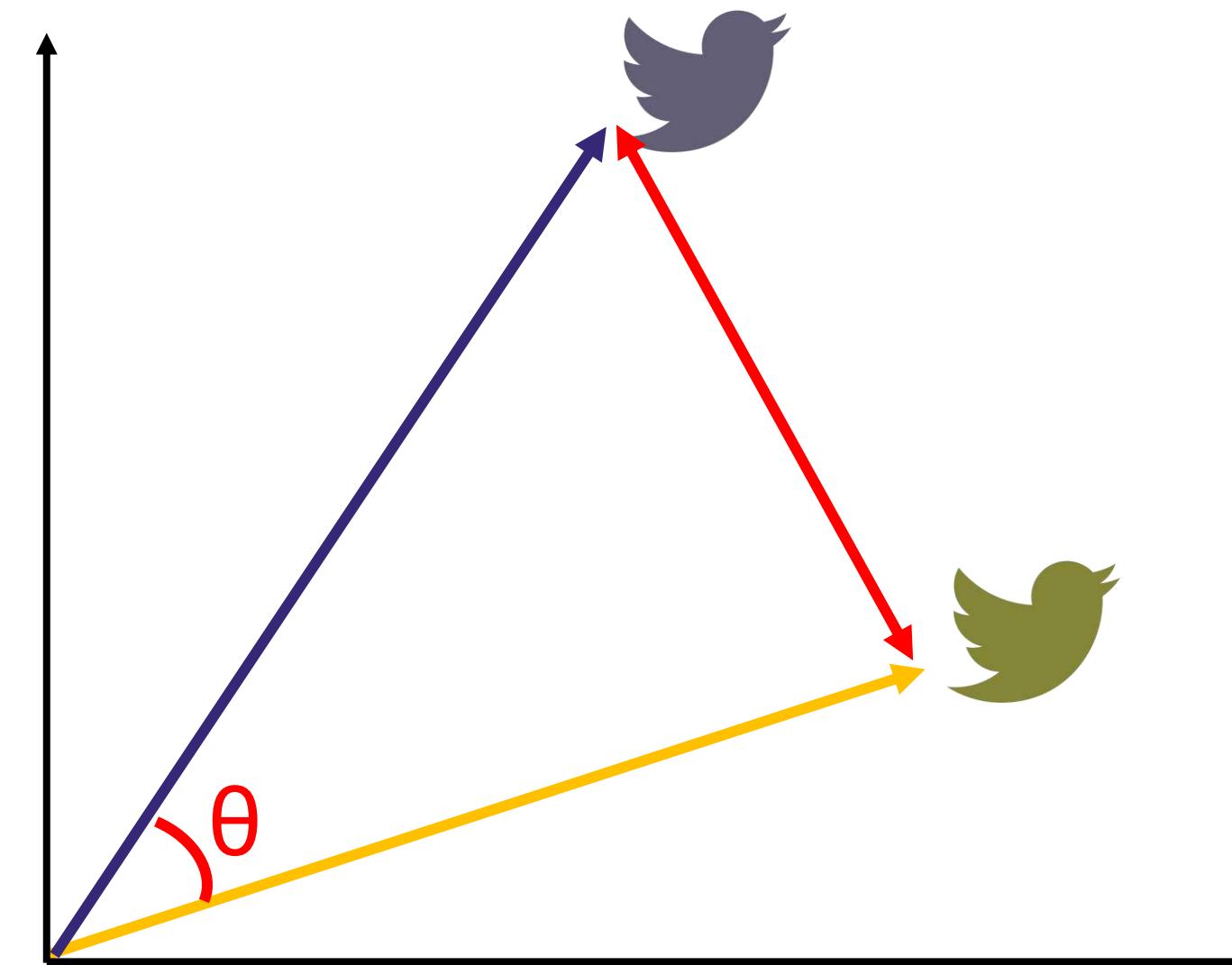
Vector representation

- What is vectorization?
- Sparse vs. dense vectors
- Text to vector
- **Vector similarity measures**
- Text vectorization in Python

Vector similarity measures

- The objective is to measure and quantify the similarity between two or more vectors (documents)
- Main similarity metrics
 - Cosine similarity
 - Euclidean distance

Distance != Similarity



Cosine similarity

- It is measured by the cosine of the angle between two vectors and determines whether two vectors are pointing in roughly the same direction
- It is often used to measure document similarity in text analysis
- It ranges from 0 to 1
 - 1 means two vectors are in exactly the same direction
 - 0 means two vectors are orthogonal

Cosine similarity

$$\cos(\theta) = \frac{A \cdot B}{\|A\| \|B\|} = \frac{\sum_1^n A_i B_i}{\sqrt{\sum_{i=1}^n A_i^2} \sqrt{\sum_{i=1}^n B_i^2}}$$

- D_1 = “natural language processing”
- D_2 = “natural language understanding”

	natural	language	processing	understanding
D_1	1	1	1	0
D_2	1	1	0	1

$$D_1 = [1, 1, 1, 0]$$

$$D_2 = [1, 1, 0, 1]$$

$$\cos(\theta) = \frac{(1 * 1) + (1 * 1) + (1 * 0) + (0 * 1)}{\sqrt{1 + 1 + 1} \sqrt{1 + 1 + 1}} = \frac{2}{3}$$

Euclidean distance

- The Euclidean distance metric allows to identify how far two points or two vectors are apart from each other
- It's the length of a line segment between the two points
- It ranges from 0 to N
 - 0 means two vectors are exactly the same
 - N is the distance and can be any positive number

Euclidean distance

$$d(A, B) = \sqrt{\sum_{i=1}^n (A_i - B_i)^2}$$

- D_1 = “natural language processing”
- D_2 = “natural language understanding”

	natural	language	processing	understanding
D_1	1	1	1	0
D_2	1	1	0	1

$$\begin{aligned}D_1 &= [1, 1, 1, 0] \\D_2 &= [1, 1, 0, 1]\end{aligned}$$

$$d(D_1, D_2) = \sqrt{(1-1)^2 + (1-1)^2 + (1-0)^2 + (0-1)^2} = \sqrt{2}$$

Backing to our example

- D_1 = “Text is a complex human language representation.”
- D_2 = “Natural human language is complex and also is diverse.”
- D_3 = “Natural human body clock is complex.”
- D_4 = “Text representation differs from human to human.”

	clock	is	human	language	natural	diverse	text	differs	representation	complex	body
D_1	0	1	1	1	0	0	1	0	1	1	0
D_2	0	2	1	1	1	1	0	0	0	1	0
D_3	1	1	1	0	1	0	0	0	0	1	1
D_4	0	0	2	0	0	0	1	1	1	0	0

Backing to our example

```
D1 = [0,1,1,1,0,0,1,0,1,1,0]  
D2 = [0,2,1,1,1,1,0,0,0,1,0]  
D3 = [1,1,1,0,1,0,0,0,0,1,1]  
D4 = [0,0,2,0,0,0,1,1,1,0,0]
```

Cosine Similarity				
	D ₁	D ₂	D ₃	D ₄
D ₁	1			
D ₂	0.68	1		
D ₃	0.5	0.68	1	
D ₄	0.61	0.25	0.30	1

Euclidean Distance				
	D ₁	D ₂	D ₃	D ₄
D ₁	0			
D ₂	2.23	0		
D ₃	2.44	2.23	0	
D ₄	2.23	3.46	3.0	0

Cosine vs. Euclidean

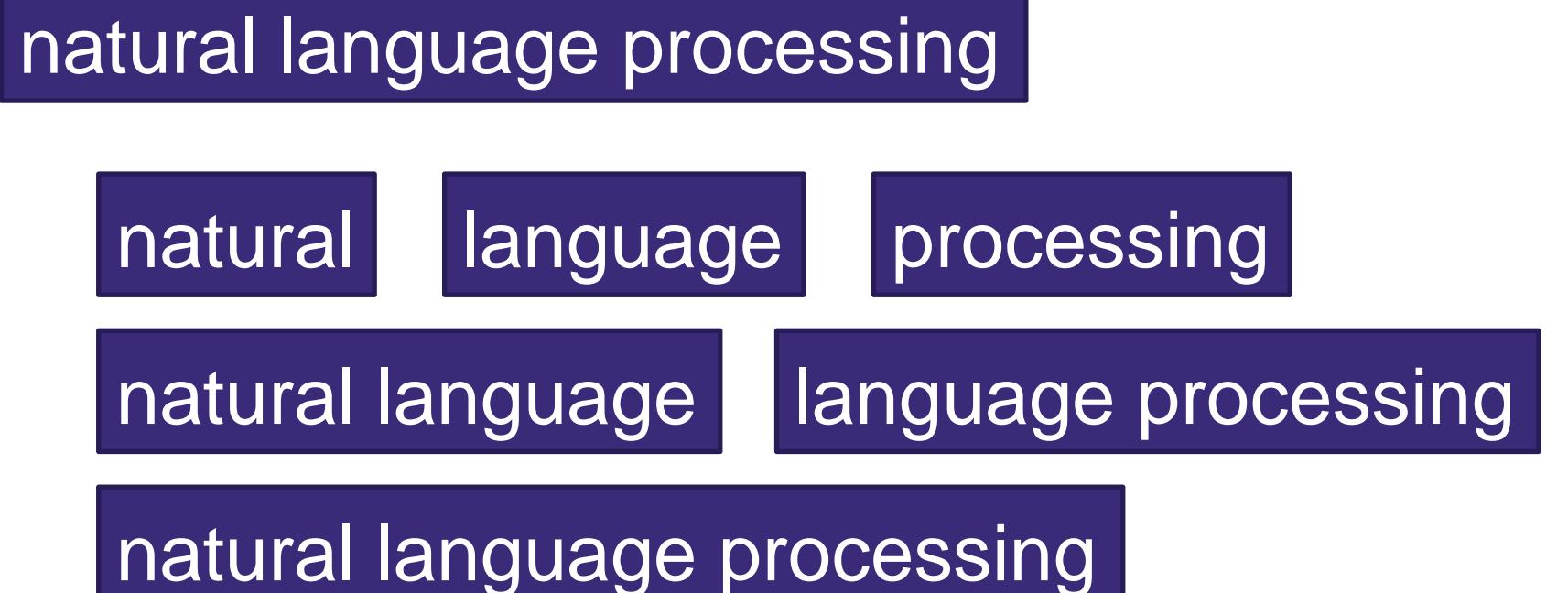
- Both metrics are **symmetric**
- ***Euclidean distance*** strongly relies on length
- ***Cosine similarity*** is generally used when the magnitude of the vectors does not matter
 - Documents of uneven lengths (e.g., Wikipedia articles)

Text vectorization

- Bag of Words (BoW)
- Bag of N-Gram
- TF-IDF

Bag of N-Gram

- An ***n-gram*** is a contiguous sequence of n items (words) from a given sample of text



- 1-gram (unigram)
- 2-gram (bigram)
- 3-gram (trigram)
- ...

Bag of N-Gram

- It allows the bag-of-words to capture a little bit more meaning from the document
- Comparing to simple bag of word model, it helps to keep the order of words

good not bad

bad not good

Bag of N-Gram

- D_1 = “Text is a complex human language representation.”
- D_2 = “Natural human language is complex and also is diverse.”
- Preprocessing
 - Lowercasing
 - Punctuation removal
 - Word tokenization
- Generating 2-grams

text is
is a
a complex
complex human
human language
language representation
natural human
language is
is complex
complex and
and also
also is
is diverse

Bag of N-Gram

- D_1 = “Text is a complex human language representation.”
- D_2 = “Natural human language is complex and also is diverse.”

	text	is	a	complex	human	language	representation	natural	language	is	complex	and	also	is	also	is	diverse
D_1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0
D_2	0	0	0	0	1	0	1	1	1	1	1	1	1	1	1	1	1

text is
is a
a complex
complex human
human language
language representation
natural human
language is
is complex
complex and
and also
also is
is diverse

Backing to our example

- D_1 = “Text is a complex human language representation.”
- D_2 = “Natural human language is complex and also is diverse.”
- D_3 = “Natural human body clock is complex.”
- D_4 = “Text representation differs from human to human.”

Cosine Similarity				
	D_1	D_2	D_3	D_4
D_1	1			
D_2	0.14	1		
D_3	0.00	0.31	1	
D_4	0.00	0.00	0.00	1

Text vectorization

- Bag of Words (BoW)
- Bag of N-Gram
- TF-IDF

TF-IDF

- A problem with scoring word frequency (e.g., bag of word) is that highly frequent words start to dominate in the document (e.g. larger score)
- But may not contain as much ***informational content*** to the model as rarer but perhaps domain specific words

Natural human language is complex and also is diverse.

- TF-IDF rescale the frequency of words by counting how often they appear in all documents
 - Scores for frequent words like “is” that are also frequent across all documents are penalized

TF-IDF

- **Term Frequency** is a scoring of the frequency of the word in the current document
- **Inverse Document Frequency** is a scoring of how rare the word is across documents

$$tf = \frac{\text{Number of repetitions of word in a document}}{\text{Total number of words in document}}$$

$$idf = \log\left(\frac{\text{Total number of documents}}{\text{Number of documents containing the word}}\right)$$

$$tfidf = tf * idf$$

TF-IDF

- D_1 = “Text is a complex human language representation.”
- D_2 = “Natural human language is complex and also is diverse.”
- D_3 = “Natural human body clock is complex.”
- D_4 = “Text representation differs from human to human.”

- Preprocessing
 - Lowercasing
 - Punctuation removal
 - Word tokenization
 - Stop word removal

clock
is
human
language
by
natural
a
diverse
to
text
differs
representation
of
complex
body
and
also

TF-IDF

$$tf = \frac{\text{Number of repetitions of word in a document}}{\text{Total number of words in document}}$$

- D_1 = “Text is a complex human language representation.”
- D_2 = “Natural human language is complex and also is diverse.”
- D_3 = “Natural human body clock is complex.”
- D_4 = “Text representation differs from human to human.”

TF	is	complex	human
D_1	1/7	0	1/7
D_2	2/9	1/9	1/9
D_3	1/6	1/6	1/6
D_4	0	0	2/7

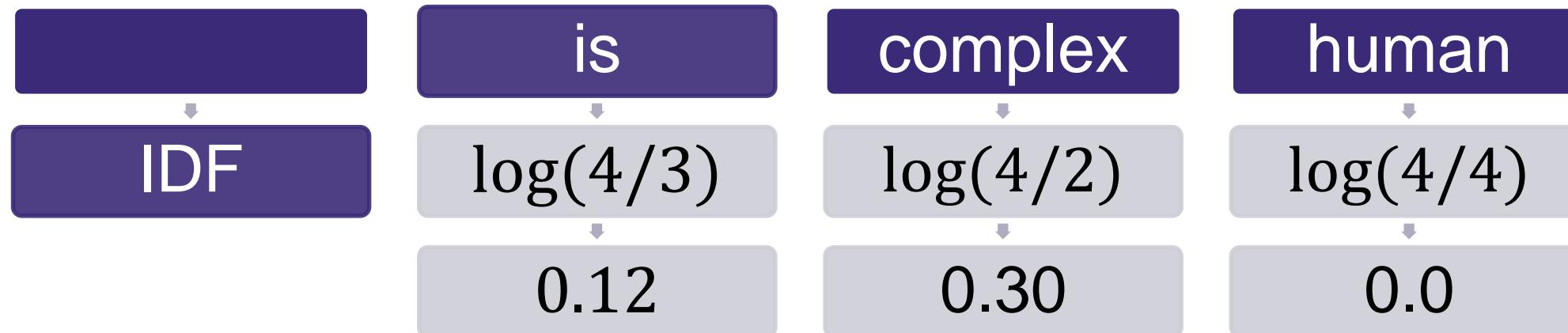
clock
is
human
language
by
natural
a
diverse
to
text
differs
representation
of
complex
body
and
also

clock
is
human
language
natural
diverse
text
differs
representation
complex
body

TF-IDF

$$idf = \log\left(\frac{\text{Total number of documents}}{\text{Number of documents containing the word}}\right)$$

- D_1 = “Text is a complex human language representation.”
- D_2 = “Natural human language is complex and also is diverse.”
- D_3 = “Natural human body clock is complex.”
- D_4 = “Text representation differs from human to human.”



clock
is
human
language
natural
diverse
text
differs
representation
complex
body

TF-IDF

- D_1 = “Text is a complex human language representation.”
- D_2 = “Natural human language is complex and also is diverse.”
- D_3 = “Natural human body clock is complex.”
- D_4 = “Text representation differs from human to human.”

TF-IDF	is	complex	human
D_1	$0.14 * 0.12 = 0.01$	0	0
D_2	0.02	0.03	0
D_3	0.02	0.05	0
D_4	0	0	0

clock
is
human
language
natural
diverse
text
differs
representation
complex
body

TF-IDF

- D_1 = “Text is a complex human language representation.”
- D_2 = “Natural human language is complex and also is diverse.”
- D_3 = “Natural human body clock is complex.”
- D_4 = “Text representation differs from human to human.”

	clock	is	human	language	natural	diverse	text	differs	representation	complex	body
D_1	0	0.01	0	0.45	0	0	0.45	0	0.45	0	0
D_2	0	0.02	0	0.30	0.30	0.39	0	0	0	0.03	0
D_3	0.51	0.02	0	0	0.40	0	0	0	0	0.05	0.51
D_4	0	0	0	0	0	0	0.34	0.43	0.34	0	0

Backing to our example

- D_1 = “Text is a complex human language representation.”
- D_2 = “Natural human language is complex and also is diverse.”
- D_3 = “Natural human body clock is complex.”
- D_4 = “Text representation differs from human to human.”

Cosine Similarity				
	D_1	D_2	D_3	D_4
D_1	1			
D_2	0.48	1		
D_3	0.32	0.42	1	
D_4	0.45	0.09	0.12	1

TF-IDF

- Different **TF** variations

binary = 0,1

raw count = Number of repetitions of word in a document

$$\text{term frequency} = \frac{\text{Number of repetitions of word in a document}}{\text{Total number of words in document}}$$

log normalization = $\log(1 + \text{Number of repetitions of word in a document})$

TF-IDF

- Different *IDF* variations

unary = 1

$$\text{inverse document frequency} = \log \left(\frac{\text{Total number of documents}}{\text{Number of documents containing the word}} \right)$$

$$\text{inverse document frequency smooth} = \log \left(\frac{\text{Total number of documents}}{1 + \text{Number of documents containing the word}} \right) + 1$$

From TF-IDF to BoW

- Different ***TF*** and ***IDF*** variations

raw count = Number of repetitions of word in a document

unary = 1

tfidf = Bag of Word

IDF and Stop-words

- Low *IDF* score can hint to **Stop-word**
 - “**and**”, “**a**”, “**the**”, “**that**” and ...
 - High representation in almost all documents means very low *IDF* score
 - Specific domains (e.g., medical texts)

Vector representation

- What is vectorization?
- Sparse vs. dense vectors
- Text to vector
- Vector similarity measures
- Text vectorization in Python

Text vectorization in Python

- scikit-learn



scikit-learn

- CountVectorizer (Bag of Word)

```
>>> from sklearn.feature_extraction.text import CountVectorizer
>>> corpus = ["sample sentence one",
              "sample sentence two"]
>>> BoF_vectorizer = CountVectorizer()
>>> X = BoF_vectorizer.fit_transform(corpus)
>>> print(BoF_vectorizer.get_feature_names())
['one', 'sample', 'sentence', 'two']
>>> print(X.toarray())
[[1, 1, 1, 0],
 [0, 1, 1, 1]]
```

scikit-learn

- CountVectorizer (Bag of Word)

```
>>> from sklearn.feature_extraction.text import CountVectorizer
>>> corpus = ["sample sentence one",
              "sample sentence two"]
>>> BoF_vectorizer = CountVectorizer()
>>> X = BoF_vectorizer.fit_transform(corpus)
>>> print(BoF_vectorizer.get_feature_names())
['one', 'sample', 'sentence', 'two']
>>> print(X.toarray())
[[1, 1, 1, 0],
 [0, 1, 1, 1]]
```

scikit-learn

- CountVectorizer (Bag of N-Gram)

```
>>> from sklearn.feature_extraction.text import CountVectorizer
>>> corpus = ["sample sentence one",
              "sample sentence two"]
>>> bigram_vectorizer = CountVectorizer(ngram_range=(2, 2))
>>> X = bigram_vectorizer.fit_transform(corpus)
>>> print(bigram_vectorizer.get_feature_names())
['sample sentence', 'sentence one', 'sentence two']
>>> print(X.toarray())
[[1, 1, 0],
 [1, 0, 1]]
```

scikit-learn

- CountVectorizer

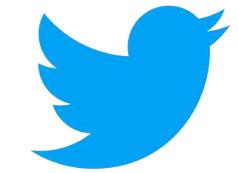
```
>>> from sklearn.feature_extraction.text import CountVectorizer
>>> corpus = ["sample sentence one",
              "sample sentence two"]
>>> vectorizer = CountVectorizer(ngram_range=(1, 2))
>>> X = vectorizer.fit_transform(corpus)
>>> print(vectorizer.get_feature_names())
['one', 'sample', 'sample sentence', 'sentence', 'sentence one',
'sentence two', 'two']
>>> print(X.toarray())
[1, 1, 1, 1, 1, 0, 0],
[0, 1, 1, 1, 0, 1, 1]]
```

scikit-learn

- TfidfVectorizer

```
>>> from sklearn.feature_extraction.text import TfidfVectorizer
>>> corpus = ["sample sentence one",
              "sample sentence two"]
>>> bigram_vectorizer = TfidfVectorizer(ngram_range=(2, 2))
>>> X = bigram_vectorizer.fit_transform(corpus)
>>> print(bigram_vectorizer.get_feature_names())
['sample sentence', 'sentence one', 'sentence two']
>>> print(X.toarray())
[[0.57973867, 0.81480247, 0.],
 [0.57973867, 0. , 0.81480247]]
```

Summary

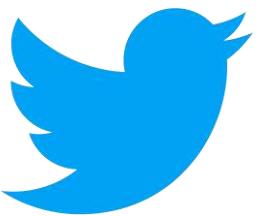


Brandwatch 🇺🇸 ✅
For Women's History Month, @OfficialPartner spoke to five leading women from the Twitter Official Partner Program, including our own Vic Miller.

Hear from them all here:

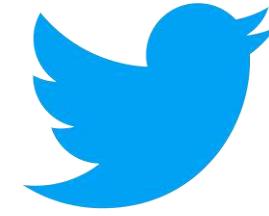


Five Powerful Women in Data MarTech
To celebrate Women's History Month, we sat down with five powerful women leaders from the Twitter Official Partner ...
[blog.twitter.com](#)



0	0	1	0	0	0	0	...	0	0	0	0
---	---	---	---	---	---	---	-----	---	---	---	---

N



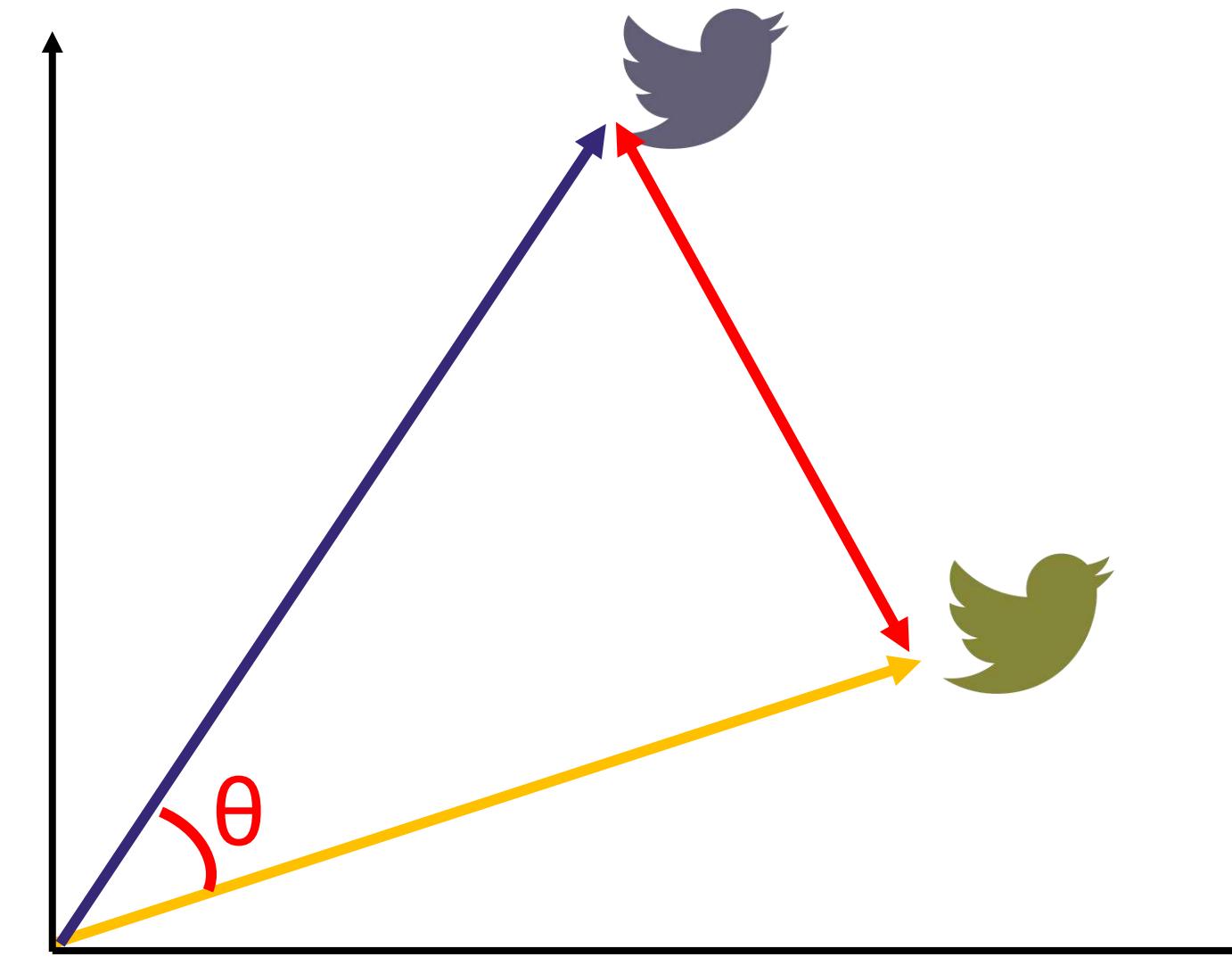
0.01	-0.55	0.9	0.5	0.28	...	0.11	-0.9
------	-------	-----	-----	------	-----	------	------

n

- D_1 = “Text is a complex human language representation.”
- D_2 = “Natural human language is complex and also is diverse.”
- D_3 = “Natural human body clock is complex.”
- D_4 = “Text representation differs from human to human.”

Summary

- Bag of Words (BoW)
- Bag of N-Gram
- TF-IDF



Word Embedding

Salar Mohtaj | DFKI

Word embedding

- What is word embedding?
- One-hot word representation
- Distributional word vectors
 - Frequency based
 - Prediction based
- Word embedding evaluation
- Word embedding in Python

Word embedding

- What is word embedding?
- One-hot word representation
- Distributional word vectors
 - Frequency based
 - Prediction based
- Word embedding evaluation
- Word embedding in Python

What is word embedding?

- **Word vectors** are simply vectors of numbers that represent the **meaning** of a word
- Vector models are also called **embeddings** (i.e., word embedding)
- The objective is to represent words in vectors in a way that those with similar meaning have similar representation



What is word embedding?



Brandwatch 🎉
@Brandwatch

For Women's History Month, [@OfficialPartner](#) spoke to five leading women from the Twitter Official Partner Program, including our own Vic Miller.

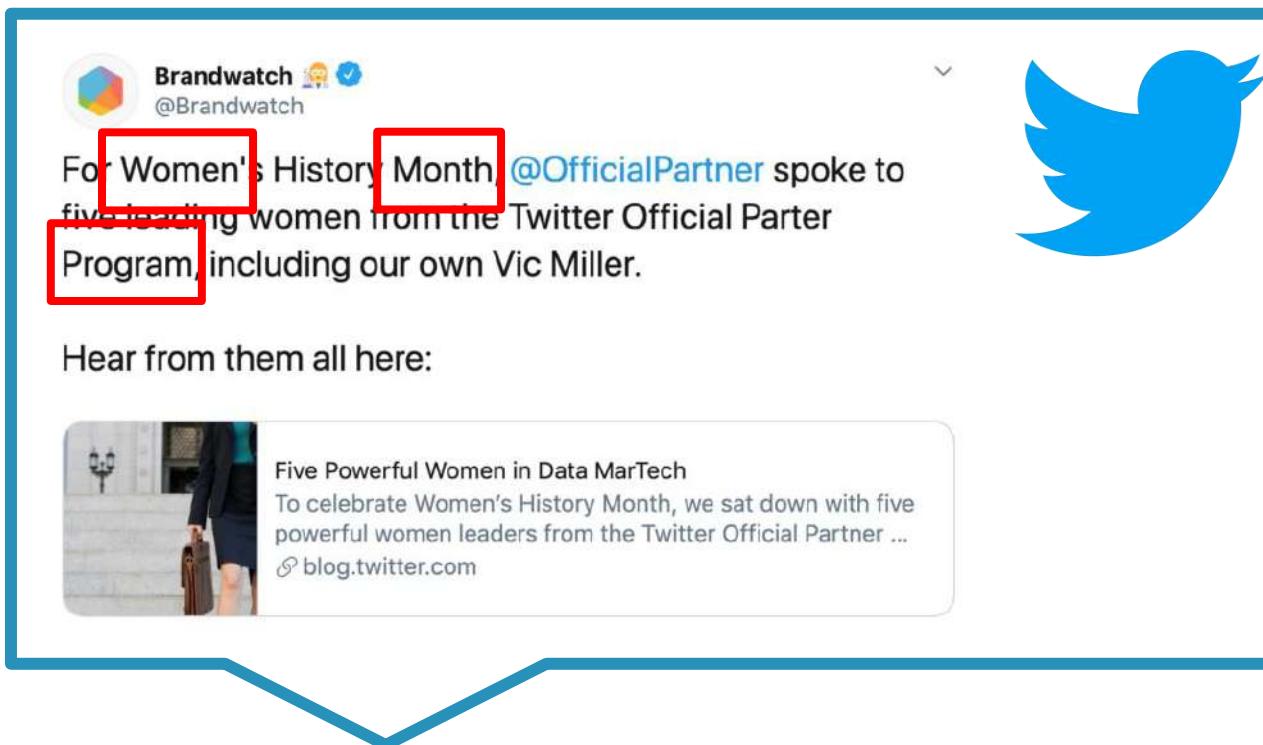
Hear from them all here:

Five Powerful Women in Data MarTech
To celebrate Women's History Month, we sat down with five powerful women leaders from the Twitter Official Partner ...
[blog.twitter.com](#)

Text vectors

1	-1	0	1	0	0	...	-1
---	----	---	---	---	---	-----	----

What is word embedding?



Word vectors

women

1	0	0	...	-1
---	---	---	-----	----

history

-1	1	0	...	1
----	---	---	-----	---

program

1	1	1	...	0
---	---	---	-----	---

...

0	0	0	...	0
---	---	---	-----	---

Word similarity, why does it matter?

The screenshot shows a Google search results page with the query "how thin is a dollar bill" in the search bar. The first result is a snippet from a page about U.S. paper currency, specifically mentioning a \$1 bill's dimensions: 2.61 inches wide by 6.14 inches long with a thickness of .0043 inches. The word "inches" appears twice in this sentence, both highlighted with red boxes. Below the snippet is the URL <https://www.ehd.org> and the title "Grasping Large Numbers". At the bottom of the snippet box, there are links for "Informationen zu hervorgehobenen Snippets" and "Feedback geben".

Google

how thin is a dollar bill

Alle Shopping Bilder News Videos Mehr Einstellungen Suchfilter

Ungefähr 21.600.000 Ergebnisse (0,88 Sekunden)

1. U.S. paper currency such as a \$1 bill measures 2.61 **inches** wide by 6.14 **inches** long with a thickness of .0043 **inches**.

<https://www.ehd.org> › ... › Technology Articles

Grasping Large Numbers

Informationen zu hervorgehobenen Snippets Feedback geben

Ähnliche Fragen

How thick is a 1 dollar bill? ▾

How thick is a \$50 bill? ▾

Can a dollar bill shrink? ▾

Is a dollar bill two pieces of paper? ▾

Feedback geben

Word similarity, why does it matter?

The screenshot shows the DeepL translation interface. At the top, there is a navigation bar with the DeepL logo, 'Translator', 'DeepL Pro', 'Plans and pricing', 'Apps', 'Download for Windows' (with a note 'it's free!'), 'Login', and a menu icon. Below the navigation bar, there are two tabs: 'Translate text' (selected) and 'Translate documents'. The main area has two input fields: 'Translate from English (detected)' containing the sentence 'how thin is a dollar bill?' and 'Translate into German' containing the sentence 'Wie dünn ist ein Dollarschein?'. Both sentences have red boxes around the word 'thin/dünn'. Below the German sentence, there is a section titled 'Alternatives:' with the sentence 'Wie dünn ist eine Dollarnote?'. At the bottom of each input field, there are icons for audio playback, a copy button, a download button, and a refresh/circular arrow button.

Word embedding

- What is word embedding?
- One-hot word representation
- Distributional word vectors
 - Frequency based
 - Prediction based
- Word embedding evaluation
- Word embedding in Python

One-hot word representation

- In **one-hot** representation each word is represented with a large vector of size $|V|$ (V is vocabulary's size for the given corpus)
- There is just one element of **1** for each word in the corpus

$v = [\text{book}, \text{machine}, \text{artificial}, \text{NLP}, \text{code}]$

machine

0	1	0	0	0
---	---	---	---	---

artificial

0	0	1	0	0
---	---	---	---	---

code

0	0	0	0	1
---	---	---	---	---

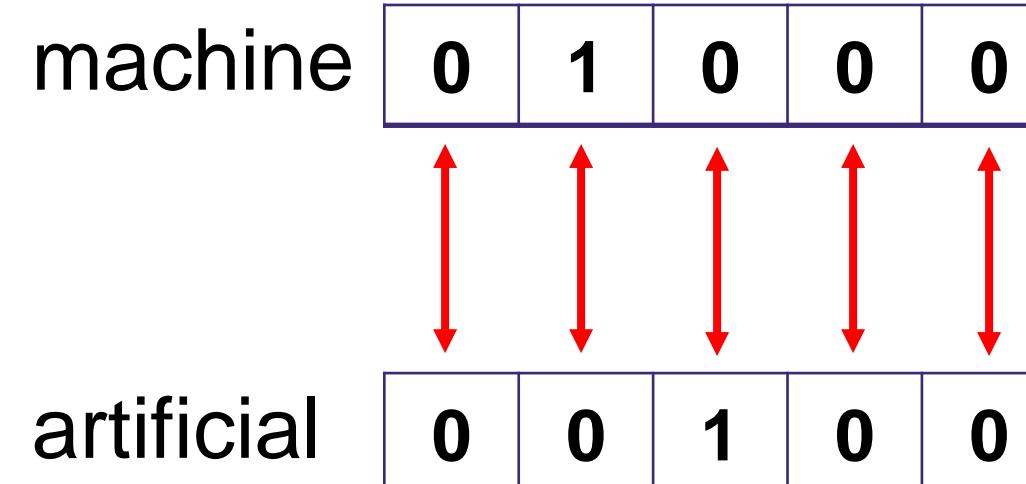
One-hot word representation

- Pros
 - Simple and easy to understand
- Cons
 - The resulting vectors are long ($|V|$) and sparse
 - We represent each word as a completely independent entity
 - The vector representation is in binary form, therefore no frequency information is taken into account
 - This word representation does not give us directly any notion of similarity

One-hot word representation

$$\cos(\theta) = \frac{A \cdot B}{\|A\| \|B\|} = \frac{\sum_1^n A_i B_i}{\sqrt{\sum_{i=1}^n A_i^2} \sqrt{\sum_{i=1}^n B_i^2}}$$

$v = [\text{book}, \text{machine}, \text{artificial}, \text{NLP}, \text{code}]$



Word embedding

- What is word embedding?
- One-hot word representation
- **Distributional word vectors**
 - Frequency based
 - Prediction based
- Word embedding evaluation
- Word embedding in Python

Distributional word vectors

- It aims to quantify and categorize semantic ***similarities*** between words based on their ***distributional properties*** in large data
- Two words are similar if they have similar ***word contexts***
 - Football and basketball have similar context words (run, ball, referee, ...)
- Humans also can guess the meaning of an unknown word from context words

Memes generally replicate through exposure to humans, who have evolved as efficient copiers of information and behavior.

Distributional word vectors

- Frequency based
 - Document-term matrix
 - Term-term matrix
 - Pointwise mutual information (PMI)
- Prediction based
 - Word2Vec

Document-term matrix

- **Similar words** tend to occur together in the **same documents**
- It describes the frequency of terms that occur in a collection of documents
- In a **document-term** matrix, rows correspond to documents in the collection and columns correspond to terms

Document-term matrix

- D_1 = “Text is a complex human language representation.”
- D_2 = “Natural human language is complex and also is diverse.”
- D_3 = “Natural human body clock is complex.”
- D_4 = “Text representation differs from human to human.”

$$\begin{aligned} \text{clock} &= [0,0,1,0] \\ \text{human} &= [1,1,1,2] \end{aligned}$$

	clock	is	human	language	natural	diverse	text	differs	representation	complex	body
D_1	0	1	1	1	0	0	1	0	1	1	0
D_2	0	2	1	1	1	1	0	0	0	1	0
D_3	1	1	1	0	1	0	0	0	0	1	1
D_4	0	0	2	0	0	0	1	1	1	0	0

Document-term matrix

- Pros
 - Simple
 - Fast to implement
- Cons
 - The resulting vectors are long ($|D|$) and sparse
 - It captures relatedness rather than similarity
 - It's not a good idea in very long documents

Document-Term matrix

Not logged in | Talk | Contributions | Create account | Log in

Article | Talk | Read | View source | View history | Search Wikipedia | 

Germany

From Wikipedia, the free encyclopedia

Coordinates: 51°N 9°E

Federal Republic of Germany
Bundesrepublik Deutschland (German)

 
Flag | Coat of arms

Anthem: "Deutschlandlied"
(English: "Song of Germany")
0:00 |  |  | MENU



Location of Germany (dark green)
– in Europe (light green & dark grey)
– in the European Union (light green)

Show globe | Show map of Europe | Show all

Capital and largest city: Berlin
52°31'N 13°23'E

Official language and national language: German

Religion: See Religion in Germany

Federal

Car

Football

Main page | Contents | Current events | Random article | About Wikipedia | Contact us | Donate | Contribute | Help | Learn to edit | Community portal | Recent changes | Upload file | Tools | What links here | Related changes | Special pages | Permanent link | Page information | Cite this page | Wikidata item | Print/export | Download as PDF | Printable version | In other projects | Wikimedia Commons | Wikinews | Wikiquote | Wikivoyage | Languages |  | Boarisch | Deutsch

Distributional word vectors

- Frequency based
 - Document-Term matrix
 - Term-term matrix
 - Pointwise Mutual Information (PMI)
- Prediction based
 - Word2Vec

Term-term matrix

- Term-document does not work well, especially in the case of long documents
- Instead of entire documents, use smaller contexts
 - Paragraph
 - Window of surrounding words (e.g., ± 3 words)
- Context words refers to surrounding words (i.e., Term-context matrix)
- The vector length is $|V|$

Term-term matrix

- D_1 = “Text is a **complex** human **language** representation.”
 - D_2 = “Natural human **language** is **complex** and also is diverse.”

Term-term matrix

- D_1 = “Text is a **complex** human **language** representation.”
- D_2 = “Natural human **language** is **complex** and also is **diverse**.”

	text	is	a	complex	human	language	representation	natural	and	also	diverse	Context
text												
is												
a												
complex		2	1		1	2			1	1		
human												
language		1		2	2		1	1				
representation												
natural												
and												
also												
diverse												

±2

23

Term-term matrix

- D_1 = “Text is a **complex** human **language** representation.”
- D_2 = “Natural human **language** is **complex** and also is diverse.”

	text	is	a	complex	human	language	representation	natural	and	also	diverse
text	0	1	1	0	0	0	0	0	0	0	0
is	1	0	1	2	1	1	0	0	1	0	0
a	1	1	0	1	1	0	0	0	0	0	0
complex	0	2	1	0	1	2	0	0	1	1	0
human	0	1	1	1	0	2	1	1	0	0	0
language	0	1	0	2	2	0	1	1	0	0	0
representation	0	0	0	0	1	1	0	0	0	0	0
natural	0	0	0	0	1	1	0	0	0	0	0
and	0	2	0	1	0	0	0	0	0	1	0
also	0	1	0	1	0	0	0	0	1	0	1
diverse	0	1	0	0	0	0	0	0	0	1	0

Term-term matrix

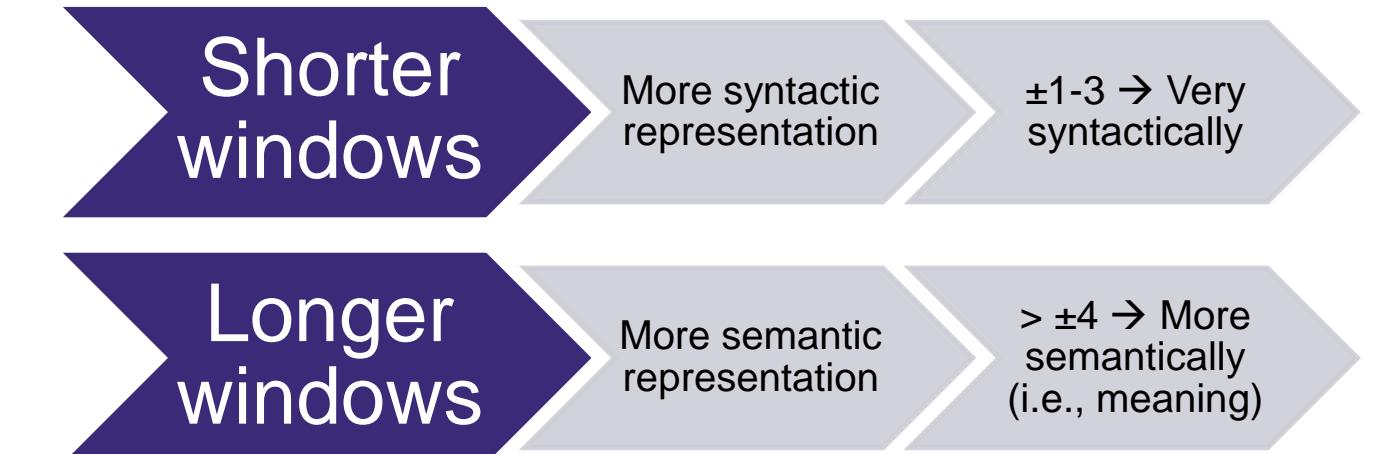
- How to set the window size? (e.g., $\pm n$)
 - $n = 1, 2, 3, \dots$

Natural human language is **complex** and also is diverse.

Natural human language is **complex** and also is diverse.

Natural human language is **complex** and also is diverse.

Natural human language is **complex** and also is diverse.



First/second order co-occurrence

- Syntagmatic association (first order co-occurrence)
 - Words that are typically nearby each other
- Paradigmatic association (second order co-occurrence)
 - Words that have similar neighbors

Why is the water in the glass?
Drinking a glass of milk is part of maintaining a healthy diet

First/second order co-occurrence

- D_1 = “Text is a **complex** human **language** representation.”
- D_2 = “Natural human **language** is **complex** and also is diverse.”

	text	is	a	complex	human	language	representation	natural	and	also	diverse
text	0	1	1	0	0	0	0	0	0	0	0
is	1	0	1	2	1	1	0	0	1	0	0
a	1	1	0	1	1	0	0	0	0	0	0
complex	0	2	1	0	1	2	0	0	1	1	0
human	0	1	1	1	0	2	1	1	0	0	0
language	0	1	0	2	2	0	1	1	0	0	0
representation	0	0	0	0	1	1	0	0	0	0	0
natural	0	0	0	0	1	1	0	0	0	0	0
and	0	2	0	1	0	0	0	0	0	1	0
also	0	1	0	1	0	0	0	0	1	0	1
diverse	0	1	0	0	0	0	0	0	0	1	0

First/second order co-occurrence

- D_1 = “Text is a **complex** human **language** representation.”
- D_2 = “Natural human **language** is **complex** and also is diverse.”

	text	is	a	complex	human	language	representation	natural	and	also	diverse
text	0	1	1	0	0	0	0	0	0	0	0
is	1	0	1	2	1	1	0	0	1	0	0
a	1	1	0	1	1	0	0	0	0	0	0
complex	0	2	1	0	1	2	0	0	1	1	0
human	0	1	1	1	0	2	1	1	0	0	0
language	0	1	0	2	2	0	1	1	0	0	0
representation	0	0	0	0	1	1	0	0	0	0	0
natural	0	0									0
and	0	2									0
also	0	1									1
diverse	0	1	0	0	0	0	0	0	0	1	0

Syntagmatic Association (First order co-occurrence)

- Word that are typically nearby each other

First/second order co-occurrence

- D_1 = “Text is a **complex** human **language** representation.”
- D_2 = “Natural human **language** is **complex** and also is diverse.”

	text	is	a	complex	human	language	representation	natural	and	also	diverse
text	0	1	1	0	0	0	0	0	0	0	0
is	1	0	1	2	1	1	0	0	1	0	0
a	1	1	0	1	1	0	0	0	0	0	0
complex	0	2	1	0	1	2	0	0	1	1	0
human	0	1	1	1	0	2	1	1	0	0	0
language	0	1	0	2	2	0	1	1	0	0	0
representation	0	0	0	0	1	1	0	0	0	0	0
natural	0	0									
and	0	2									
also	0	1									
diverse	0	1	0	0	0	0	0	0	0	1	0

Paradigmatic Association (Second order co-occurrence)

- Word that have similar neighbors

Term-term matrix

- Pros
 - Simple to understand
 - Better capture word meaning than the term-document matrix
- Cons
 - The resulting vectors are long ($|V|$) and sparse
 - Some common words (e.g., “is”) relate some unrelated words to each other

Distributional word vectors

- Frequency based
 - Document-Term matrix
 - Term-Term matrix
 - Pointwise mutual information (PMI)
- Prediction based
 - Word2Vec

Pointwise mutual information (PMI)

- Problem with raw counts (e.g., term-term matrix)
 - Some words (like “is”) are very frequent, but maybe not the most **discriminative**
- We try to measure whether a context word is **informative**

$$PMI(W_1, W_2) = \log_2 \frac{P(W_1, W_2)}{P(W_1)P(W_2)}$$

- Do words W_1 and W_2 co-occur more than if they were independent?

PMI

$$PMI(W_1, W_2) = \log_2 \frac{P(W_1, W_2)}{P(W_1)P(W_2)}$$

Two events W_1, W_2 are independent if their joint probability is equal to the product of their individual probabilities

$$P(W_1, W_2) = P(W_1)P(W_2)$$

$$\frac{P(W_1, W_2)}{P(W_1)P(W_2)} = 1$$
$$\log_2 1 = 0$$

PMI

- D_1 = “Text is a complex human language representation.”
- D_2 = “Natural human language is complex and also is diverse.”

	text	is	a	complex	human	language	representation	natural	and	also	diverse
text	0	1	0	0	0	0	0	0	0	0	0
is	1	0	1	2	1	1	0	0	1	0	0
a	1	1	0	1	1	0	0	0	0	0	0
complex	0	2	1	0	1	2	0	0	1	1	0
human	0	1	1	1	0	2	1	1	0	0	0
language	0	1	0	2	2	0	1	1	0	0	0
representation	0	0	0	0	1	1	0	0	0	0	0
natural	0	0	0	0	1	1	0	0	0	0	0
and	0	2	0	1	0	0	0	0	0	1	0
also	0	1	0	1	0	0	0	0	1	0	1
diverse	0	1	0	0	0	0	0	0	0	1	0

PMI

- D_1 = “Text is a complex human language representation.”
- D_2 = “Natural human language is complex and also is diverse.”

	text	is	a	complex	human	language	representation	natural	and	also	diverse
text	0	1	1	0	0	0	0	0	0	0	0
is	1	0	1	2	1	1	0	0	1	0	0
a	1	1	0	1	1	0	0	0	0	0	0
complex	0	2	1	0	1	2	0	0	1	1	0
human	0	1	1	1	0	2	1	1	0	0	0
language	0	1	0	2	2	0	1	1	0	0	0
representation	0	0	0	0	1	1	0	0	0	0	0
natural	0	0	0	0	1	1	0	0	0	0	0
and	0	2	0	1	0	0	0	0	0	1	0
also	0	1	0	1	0	0	0	0	1	0	1
diverse	0	1	0	0	0	0	0	0	0	1	0

PMI

$$PMI(W_1, W_2) = \log_2 \frac{P(W_1, W_2)}{P(W_1)P(W_2)}$$

- $P(W_1, W_2) = \frac{\# \text{ of times } W_1 \text{ occurs in context of } W_2}{\# \text{ of times all words occur in context of all the other words}}$
- $P(W_1) = \frac{\# \text{ of times } W_1 \text{ occurs in context of all context words}}{\# \text{ of times all words occur in context of all the other words}}$
- $P(W_2) = \frac{\# \text{ of times that all the words occurs in context of } W_2}{\# \text{ of times all words occur in context of all the other words}}$

PMI

	text	is	a	complex	human	language	representation	natural	and	also	diverse
text	0	1	1	0	0	0	0	0	0	0	0
is	1	0	1	2	1	1	0	0	1	0	0
a	1	1	0	1	1	0	0	0	0	0	0
complex	0	2	1	0	1	2	0	0	1	1	0
human	0	1	1	1	0	2	1	1	0	0	0
language	0	1	0	2	2	0	1	1	0	0	0
representation	0	0	0	0	1	1	0	0	0	0	0
natural	0	0	0	0	1	1	0	0	0	0	0
and	0	2	0	1	0	0	0	0	0	1	0
also	0	1	0	1	0	0	0	0	1	0	1
diverse	0	1	0	0	0	0	0	0	0	1	0

PMI

PMI(human, is)

$$p(\text{human}, \text{is}) = ^1/_{49} \mid p = (\text{human}) = ^7/_{49} \mid p(\text{is}) = ^{10}/_{49}$$

	text	is	a	complex	human	language	representation	natural	and	also	diverse
text	0	1	1	0	0	0	0	0	0	0	0
is	1	0	1	2	1	1	0	0	1	0	0
a	1	1	0	1	1	0	0	0	0	0	0
complex	0	2	1	0	1	2	0	0	1	1	0
human	0	1	1	1	0	2	1	1	0	0	0
language	0	1	0	2	2	0	1	1	0	0	0
representation	0	0	0	0	1	1	0	0	0	0	0
natural	0	0	0	0	1	1	0	0	0	0	0
and	0	2	0	1	0	0	0	0	0	1	0
also	0	1	0	1	0	0	0	0	1	0	1
diverse	0	1	0	0	0	0	0	0	0	1	0

PMI

PMI(human, is)

$$p(\text{human}, \text{is}) = ^1/_{49} \mid p = (\text{human}) = ^7/_{49} \mid p(\text{is}) = ^{10}/_{49}$$

	text	is	a	complex	human	language	representation	natural	and	also	diverse
text	0	1	1	0	0	0	0	0	0	0	0
is	1	0	1	2	1	1	0	0	1	0	0
a	1	1	0	1	1	0	0	0	0	0	0
complex	0	2	1	0	1	2	0	0	1	1	0
human	0	1	1	1	0	2	1	1	0	0	0
language	0	1	0	2	2	0	1	1	0	0	0
representation	0	0	0	0	1	1	0	0	0	0	0
natural	0	0	0	0	1	1	0	0	0	0	0
and	0	2	0	1	0	0	0	0	0	1	0
also	0	1	0	1	0	0	0	0	1	0	1
diverse	0	1	0	0	0	0	0	0	0	1	0

PMI

PMI(human, is)

$$p(\text{human}, \text{is}) = ^1/_{49} \mid p = (\text{human}) = ^7/_{49} \mid p(\text{is}) = ^{10}/_{49}$$

	text	is	a	complex	human	language	representation	natural	and	also	diverse
text	0	1	1	0	0	0	0	0	0	0	0
is	1	0	1	2	1	1	0	0	1	0	0
a	1	1	0	1	1	0	0	0	0	0	0
complex	0	2	1	0	1	2	0	0	1	1	0
human	0	1	1	1	0	2	1	1	0	0	0
language	0	1	0	2	2	0	1	1	0	0	0
representation	0	0	0	0	1	1	0	0	0	0	0
natural	0	0	0	0	1	1	0	0	0	0	0
and	0	2	0	1	0	0	0	0	0	1	0
also	0	1	0	1	0	0	0	0	1	0	1
diverse	0	1	0	0	0	0	0	0	0	1	0

PMI

$PMI(human, is)$

$$p(human, is) = 1/49 \mid p = (human) = 7/49 \mid p(is) = 10/49$$

$$PMI(human, is) = \log_2 \frac{1/49}{7/49 * 10/49} = -0.51$$

	text	is	a	complex	human	language	representation	natural	and	also	diverse
text	0	1	1	0	0	0	0	0	0	0	0
is	1	0	1	2	1	1	0	0	1	0	0
a	1	1	0	1	1	0	0	0	0	0	0
complex	0	2	1	0	1	2	0	0	1	1	0
human	0	1	1	1	0	2	1	1	0	0	0
language	0	1	0	2	2	0	1	1	0	0	0
representation	0	0	0	0	1	1	0	0	0	0	0
natural	0	0	0	0	1	1	0	0	0	0	0
and	0	2	0	1	0	0	0	0	0	1	0
also	0	1	0	1	0	0	0	0	1	0	1
diverse	0	1	0	0	0	0	0	0	0	1	0

PMI

PMI(human, natural)

$$p(\text{human}, \text{natural}) = {}^1/_{49} \mid p = (\text{human}) = {}^7/_{49} \mid p(\text{natural}) = {}^2/_{49}$$

	text	is	a	complex	human	language	representation	natural	and	also	diverse
text	0	1	1	0	0	0	0	0	0	0	0
is	1	0	1	2	1	1	0	0	1	0	0
a	1	1	0	1	1	0	0	0	0	0	0
complex	0	2	1	0	1	2	0	0	1	1	0
human	0	1	1	1	0	2	1	1	0	0	0
language	0	1	0	2	2	0	1	1	0	0	0
representation	0	0	0	0	1	1	0	0	0	0	0
natural	0	0	0	0	1	1	0	0	0	0	0
and	0	2	0	1	0	0	0	0	0	1	0
also	0	1	0	1	0	0	0	0	1	0	1
diverse	0	1	0	0	0	0	0	0	0	1	0

PMI

PMI(human, natural)

$$p(\text{human}, \text{natural}) = 1/49 \mid p = (\text{human}) = 7/49 \mid p(\text{natural}) = 2/49$$

	text	is	a	complex	human	language	representation	natural	and	also	diverse
text	0	1	1	0	0	0	0	0	0	0	0
is	1	0	1	2	1	1	0	0	1	0	0
a	1	1	0	1	1	0	0	0	0	0	0
complex	0	2	1	0	1	2	0	0	1	1	0
human	0	1	1	1	0	2	1	1	0	0	0
language	0	1	0	2	2	0	1	1	0	0	0
representation	0	0	0	0	1	1	0	0	0	0	0
natural	0	0	0	0	1	1	0	0	0	0	0
and	0	2	0	1	0	0	0	0	0	1	0
also	0	1	0	1	0	0	0	0	1	0	1
diverse	0	1	0	0	0	0	0	0	0	1	0

PMI

$PMI(human, natural)$

$$p(human, natural) = ^1/_{49} \mid p = (human) = ^7/_{49} \mid p(natural) = ^2/_{49}$$

	text	is	a	complex	human	language	representation	natural	and	also	diverse
text	0	1	1	0	0	0	0	0	0	0	0
is	1	0	1	2	1	1	0	0	1	0	0
a	1	1	0	1	1	0	0	0	0	0	0
complex	0	2	1	0	1	2	0	0	1	1	0
human	0	1	1	1	0	2	1	1	0	0	0
language	0	1	0	2	2	0	1	1	0	0	0
representation	0	0	0	0	1	1	0	0	0	0	0
natural	0	0	0	0	1	1	0	0	0	0	0
and	0	2	0	1	0	0	0	0	0	1	0
also	0	1	0	1	0	0	0	0	1	0	1
diverse	0	1	0	0	0	0	0	0	0	1	0

PMI

$PMI(human, natural)$

$$p(human, natural) = 1/49 \mid p = (human) = 7/49 \mid p(natural) = 2/49$$

$$PMI(human, natural) = \log_2 \frac{1/49}{7/49 * 2/49} = 1.8$$

	text	is	a	complex	human	language	representation	natural	and	also	diverse
text	0	1	1	0	0	0	0	0	0	0	0
is	1	0	1	2	1	1	0	0	1	0	0
a	1	1	0	1	1	0	0	0	0	0	0
complex	0	2	1	0	1	2	0	0	1	1	0
human	0	1	1	1	0	2	1	1	0	0	0
language	0	1	0	2	2	0	1	1	0	0	0
representation	0	0	0	0	1	1	0	0	0	0	0
natural	0	0	0	0	1	1	0	0	0	0	0
and	0	2	0	1	0	0	0	0	0	1	0
also	0	1	0	1	0	0	0	0	1	0	1
diverse	0	1	0	0	0	0	0	0	0	1	0

PMI

	text	is	a	complex	human	language	representation	natural	and	also	diverse
text	0	1	1	0	0	0	0	0	0	0	0
is	1	0			1	1	0	0	1	0	0
a	1	1			1	0	0	0	0	0	0
complex	0	2			1	2	0	0	1	1	0
human	0	1	1	1	0	2	1	1	0	0	0
language	0	1	0	2	2			1	0	0	0
representation	0	0	0	0	1			0	0	0	0
natural	0	0	0	0	1			0	0	0	0
and	0	2	0	1	0			0	0	1	0
also	0	1	0	1	0	0	0	0	1	0	1
diverse	0	1	0	0	0	0	0	0	0	1	0

$$PMI = -0.51$$

$$PMI = +1.8$$

Positive pointwise mutual information (PPMI)

$$PPMI(W_1, W_2) = \max \left(\log_2 \frac{P(W_1, W_2)}{P(W_1)P(W_2)}, 0 \right)$$

- The values should be counted on a huge corpus to be sure if two terms are really unrelated
- It's also difficult to interpret if larger negative value means more un-relatedness

PMI

- PMI is biased toward infrequent events
 - Very rare words have very high PMI values
- Possible solution
 - Use add-one smoothing (Laplace smoothing)

Use add-one smoothing

	text	is	a	complex	human	language	representation	natural	and	also	diverse
text	0	1	1	0	0	0	0	0	0	0	0
is	1	0	1	2	1	1	0	0	1	0	0
a	1	1	0	1	1	0	0	0	0	0	0
complex	0	2	1	0	1	2	0	0	1	1	0
human	0	1	1	1	0	2	1	1	0	0	0
language	0	1	0	2	2	0	1	1	0	0	0
representation	0	0	0	0	1	1	0	0	0	0	0
natural	0	0	0	0	1	1	0	0	0	0	0
and	0	2	0	1	0	0	0	0	0	1	0
also	0	1	0	1	0	0	0	0	1	0	1
diverse	0	1	0	0	0	0	0	0	0	1	0

Use add-one smoothing

	text	is	a	complex	human	language	represen tation	natural	and	also	diverse
text	2	3	3	2	2	2	2	2	2	2	2
is	3	2	3	4	3	3	2	2	3	2	2
a	3	3	2	3	3	2	2	2	2	2	2
complex	2	4	3	2	3	4	2	2	3	3	2
human	2	3	3	3	2	4	3	3	2	2	2
language	2	3	2	4	4	2	3	3	2	2	2
representation	2	2	2	2	3	3	2	2	2	2	2
natural	2	2	2	2	3	3	2	2	2	2	2
and	2	4	2	3	2	2	2	2	2	3	2
also	2	3	2	3	2	2	2	2	3	2	3
diverse	2	3	2	2	2	2	2	2	2	3	2

+2

PMI

- Pros
 - Better capture word meaning then the term-term matrix
 - Penalize scores by the common words
- Cons
 - The resulting vectors are long ($|V|$) and sparse

PMI

- How to resolve the sparsity issue in PMI
- Matrix factorization
 - Singular value decomposition (SVD)

$$\text{Original Matrix} = U \begin{pmatrix} S \\ 0 \end{pmatrix} V^T$$

Word embedding

- What is word embedding?
- One-hot word representation
- **Distributional word vectors**
 - Frequency based
 - Prediction based
- Word embedding evaluation
- Word embedding in Python

Distributional word vectors

- Frequency based
 - Document-Term matrix
 - Term-Term matrix
 - Pointwise Mutual Information (PMI)
- Prediction based
 - Word2Vec

From sparse to dense vectors

- Frequency based embedding
 - Long (~10,000 to 50,000)
 - Sparse (most elements are 0)
- Prediction based embedding (word embedding)
 - Short (~100 to 1,000)
 - Dense (most element are non-zero)

Why dense vectors

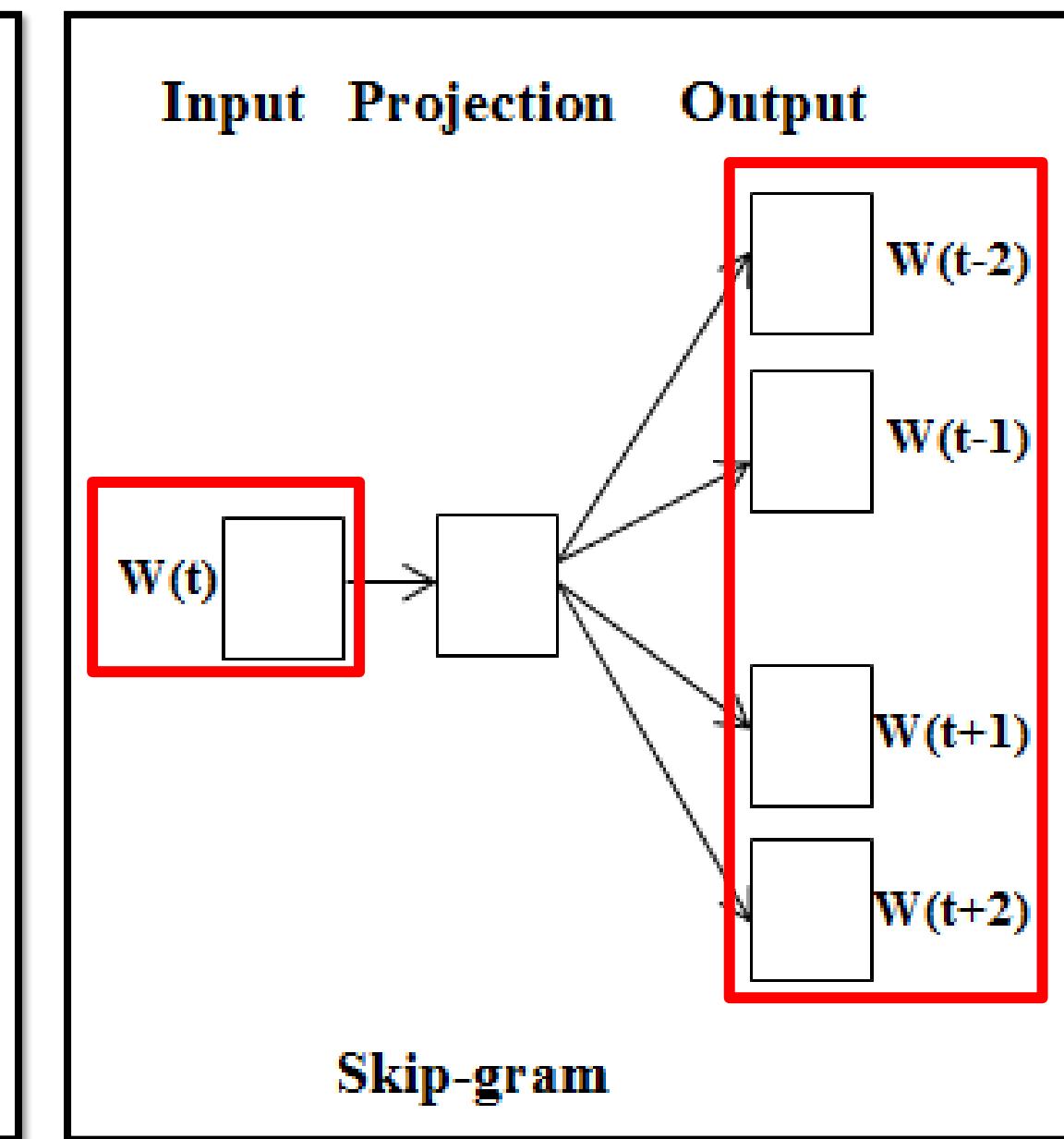
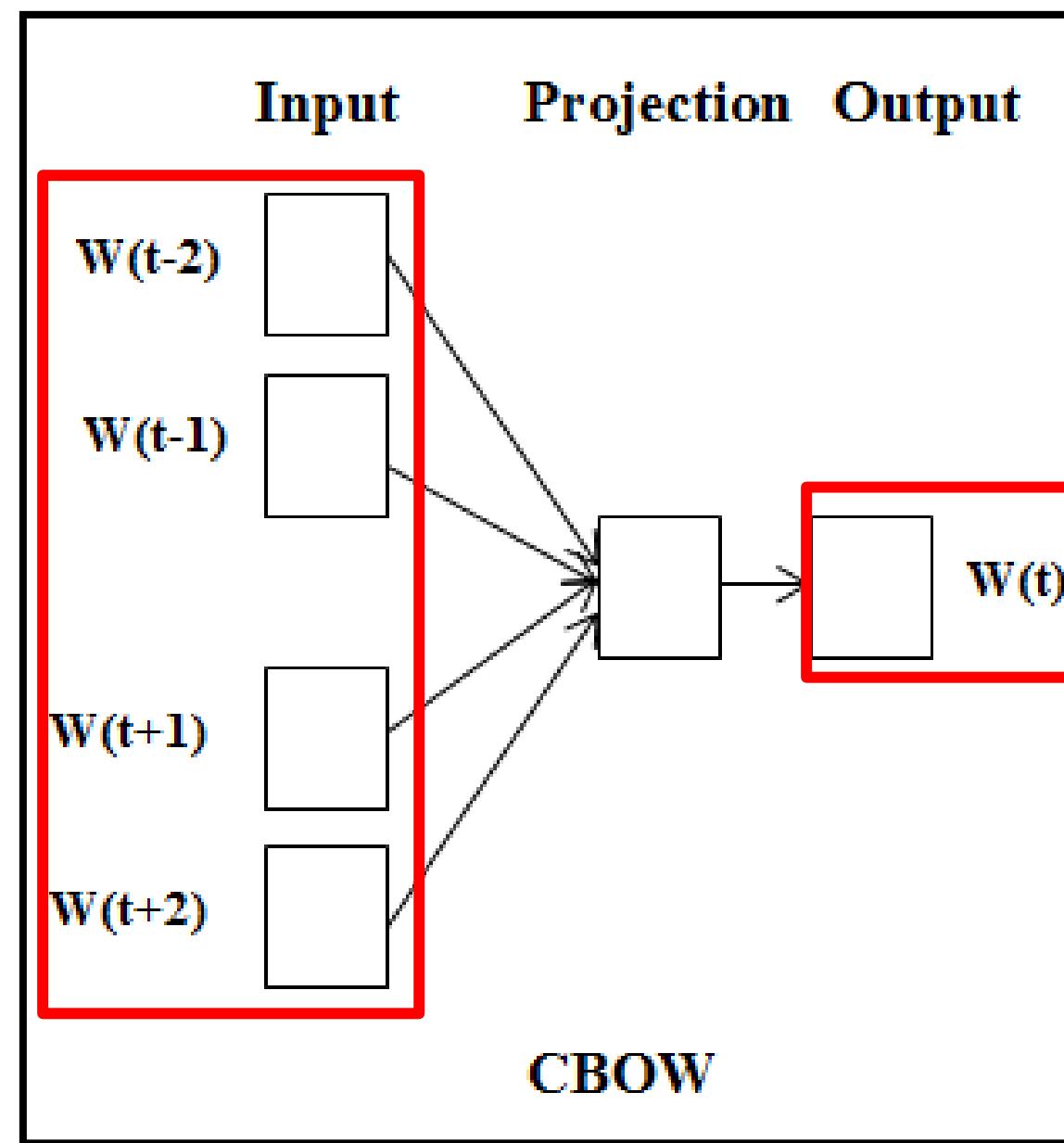
- They usually better capture meaning (e.g., work better in finding synonyms)
- Leads to less weights to trains in machine learning models

Word2Vec

- The **word2vec** model uses a **neural network** architecture (**two-layer neural net**) to learn word associations from a **large corpus of text**
- **Word2vec** was created and published in **2013** by a team of researchers led by **Tomas Mikolov** at **Google** over two papers
- While word2vec **is not a deep neural network**, it turns text into a numerical form that deep neural networks can understand
- Two word2Vec models:
 - continuous bag-of-words (CBOW)
 - skip-gram

Word2Vec

- Given context words
- Predict the probability of a target word



- Given a target word
- Predict the probability of context words

Word2Vec

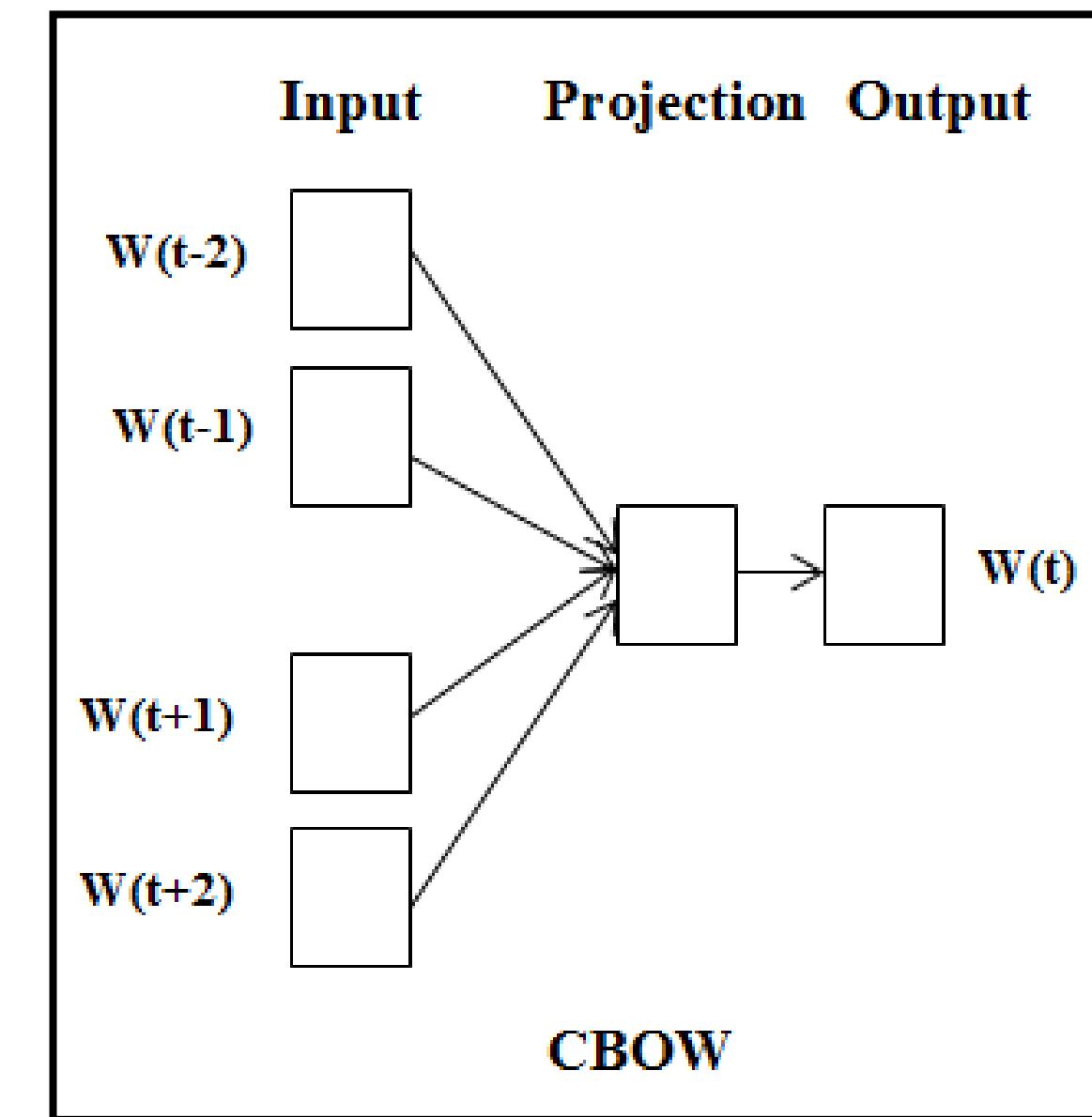
- We won't be interested in the ***inputs*** and ***outputs*** of this network
- Rather the goal is actually just to learn the weights of the hidden layer that are actually the ***word vectors*** that we're trying to learn

CBOW

Natural human language is complex and also is diverse

- Window size: ±2 (hyperparameter)
- Vocabulary size: 8
- Vector size: 5 (hyperparameter)

natural
human
language
is
complex
and
also
diverse



CBOW

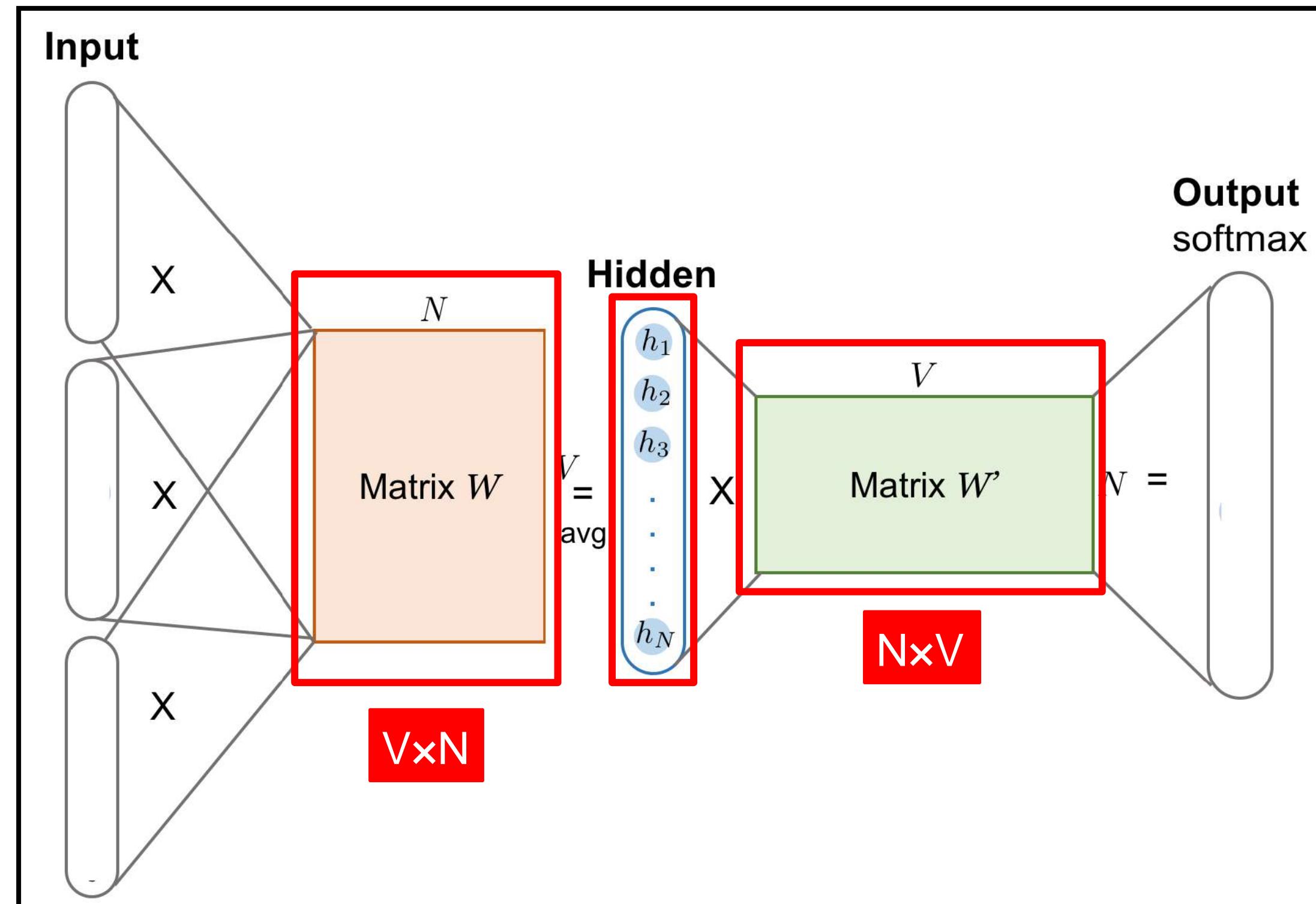


Image from www.lilianweng.github.io

CBOW

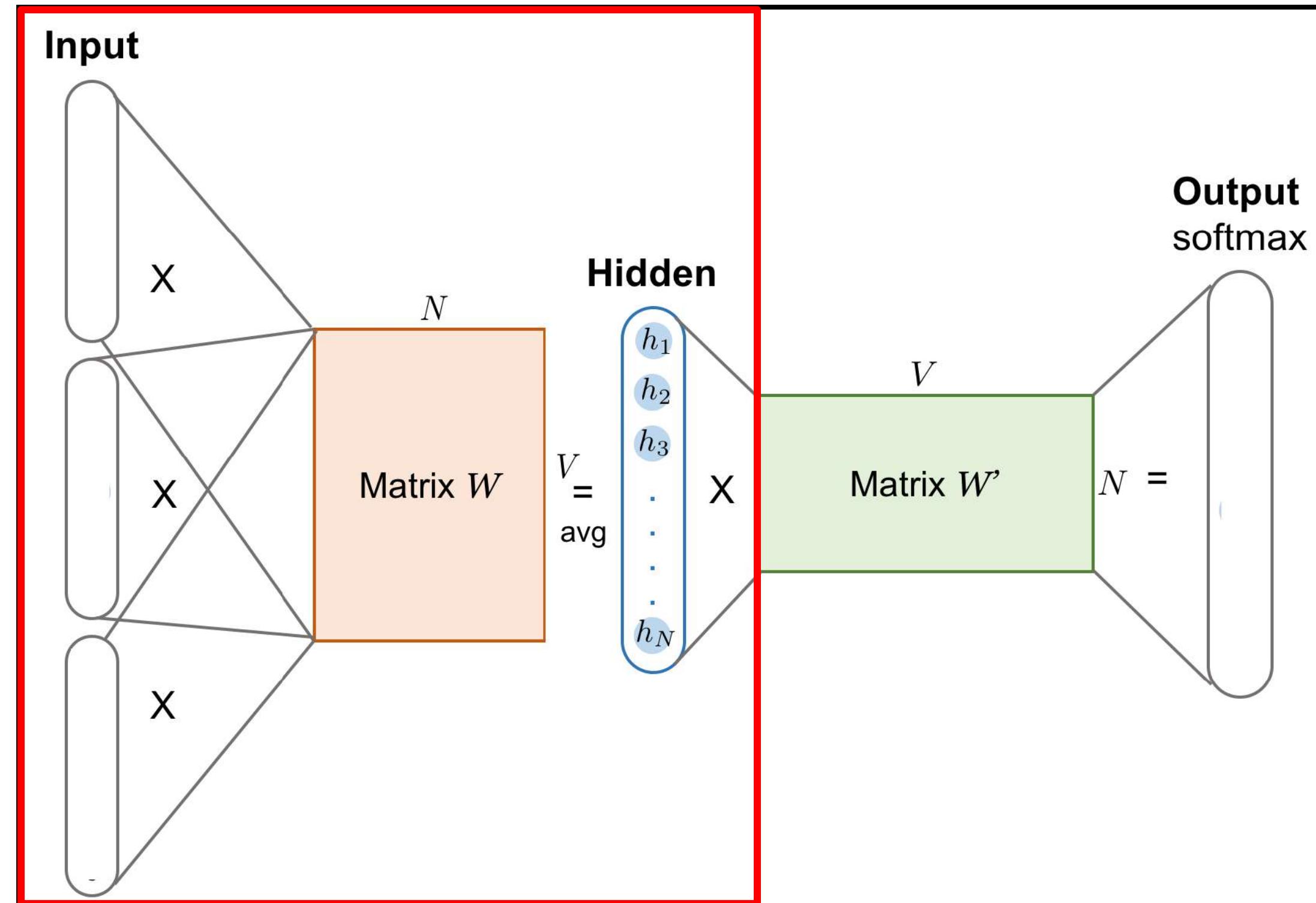
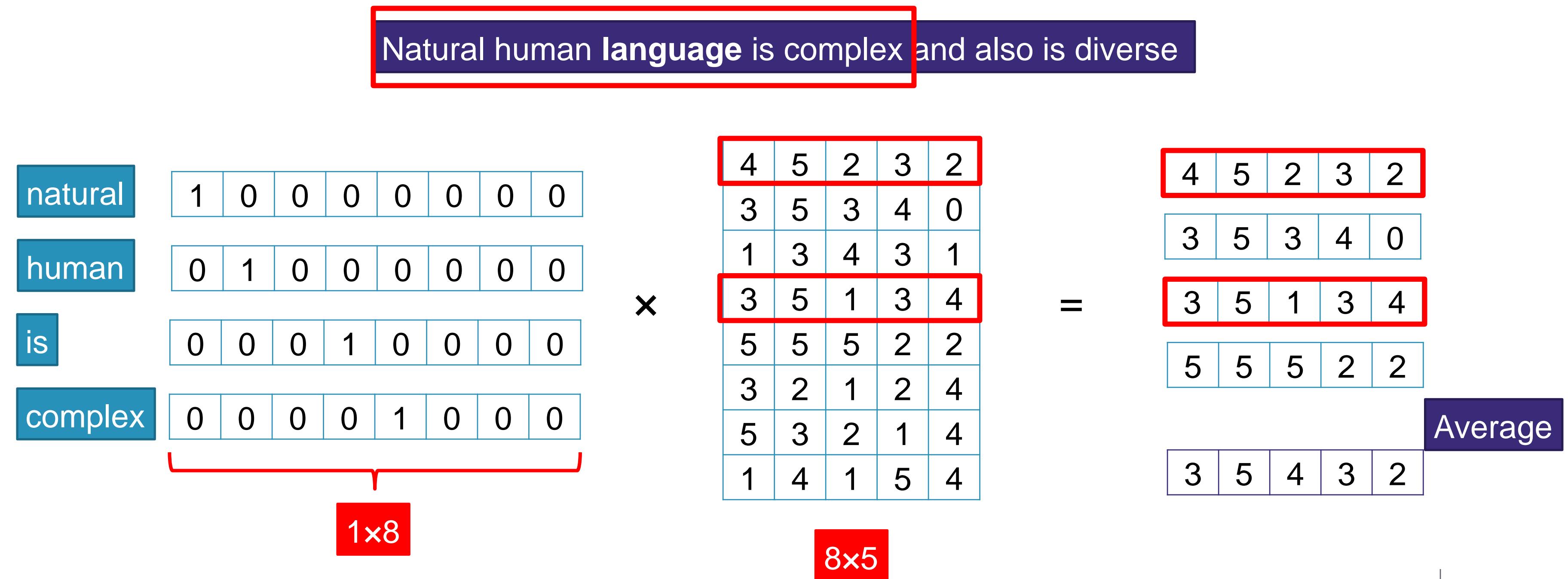


Image from www.lilianweng.github.io

CBOW

Window size: ±2 (hyperparameter)
Vocabulary size: 8
Vector size: 5 (hyperparameter)



CBOW

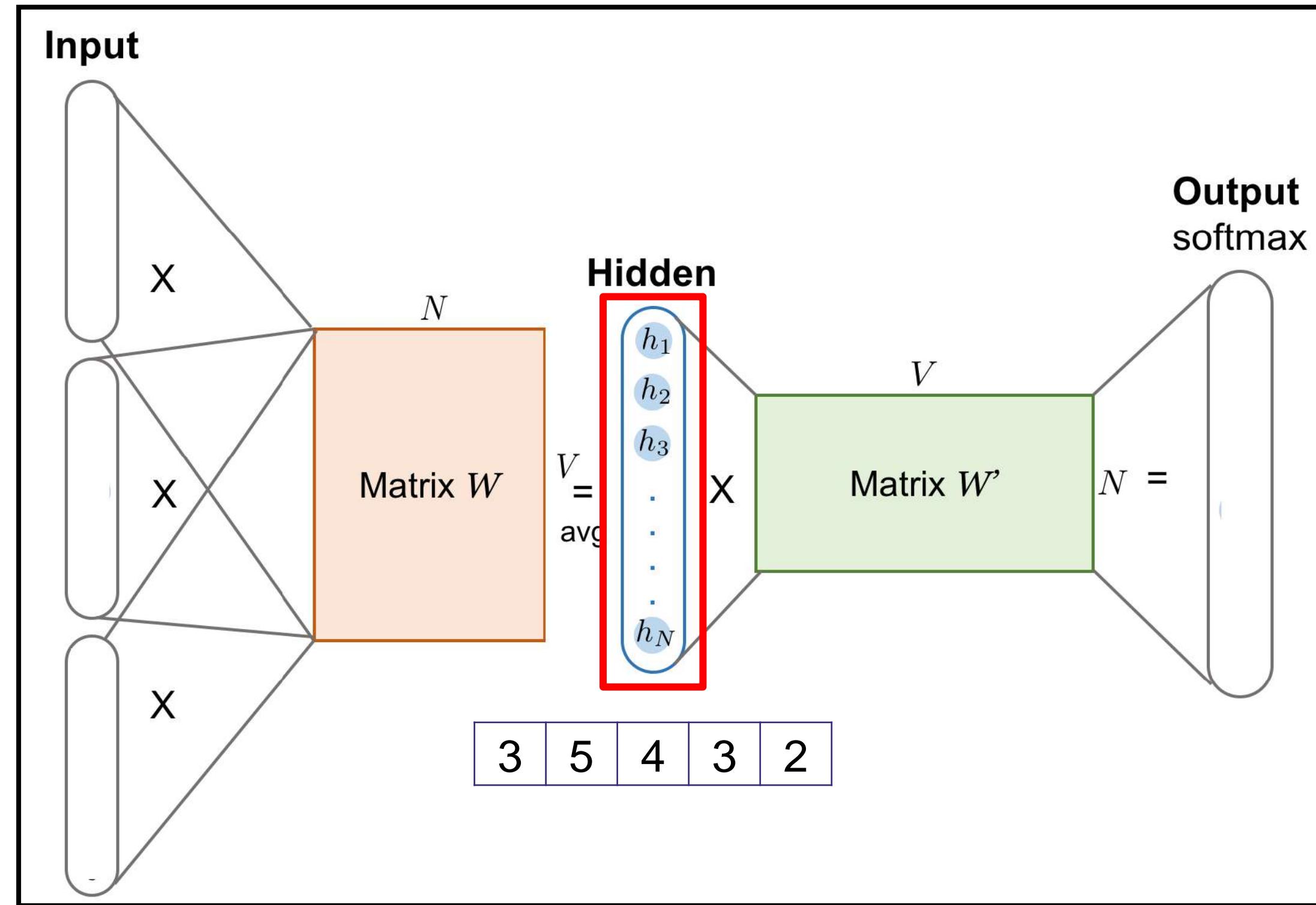


Image from www.lilianweng.github.io

CBOW

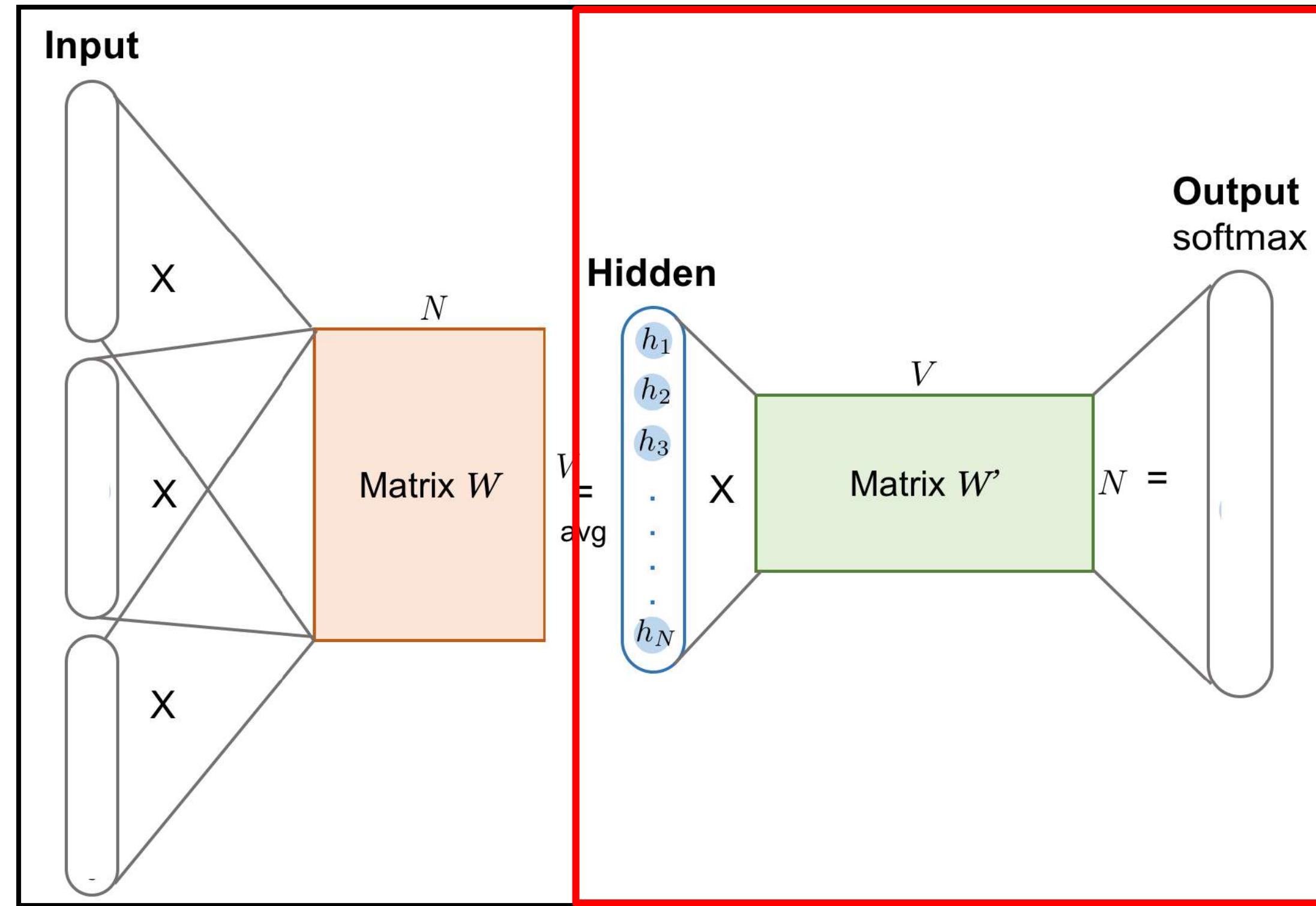


Image from www.lilianweng.github.io

CBOW

Window size: ±2 (hyperparameter)
Vocabulary size: 8
Vector size: 5 (hyperparameter)

$$\begin{matrix} 3 & 5 & 4 & 3 & 2 \end{matrix} \times \begin{matrix} 3 & 2 & 3 & 1 & 5 & 2 & 3 & 0 \\ 3 & 5 & 1 & 0 & 3 & 5 & 5 & 4 \\ 2 & 0 & 0 & 1 & 5 & 2 & 2 & 2 \\ 2 & 4 & 3 & 1 & 0 & 5 & 3 & 4 \\ 3 & 3 & 5 & 2 & 5 & 3 & 5 & 5 \end{matrix} = \begin{matrix} 44 & 49 & 33 & 14 & 60 & 60 & 61 & 50 \end{matrix}$$

5×8 1×8

Output layer

CBOW

Natural human language is complex and also is diverse

44	49	33	14	60	60	61	50
natural	human	language	is	complex	and	also	diverse

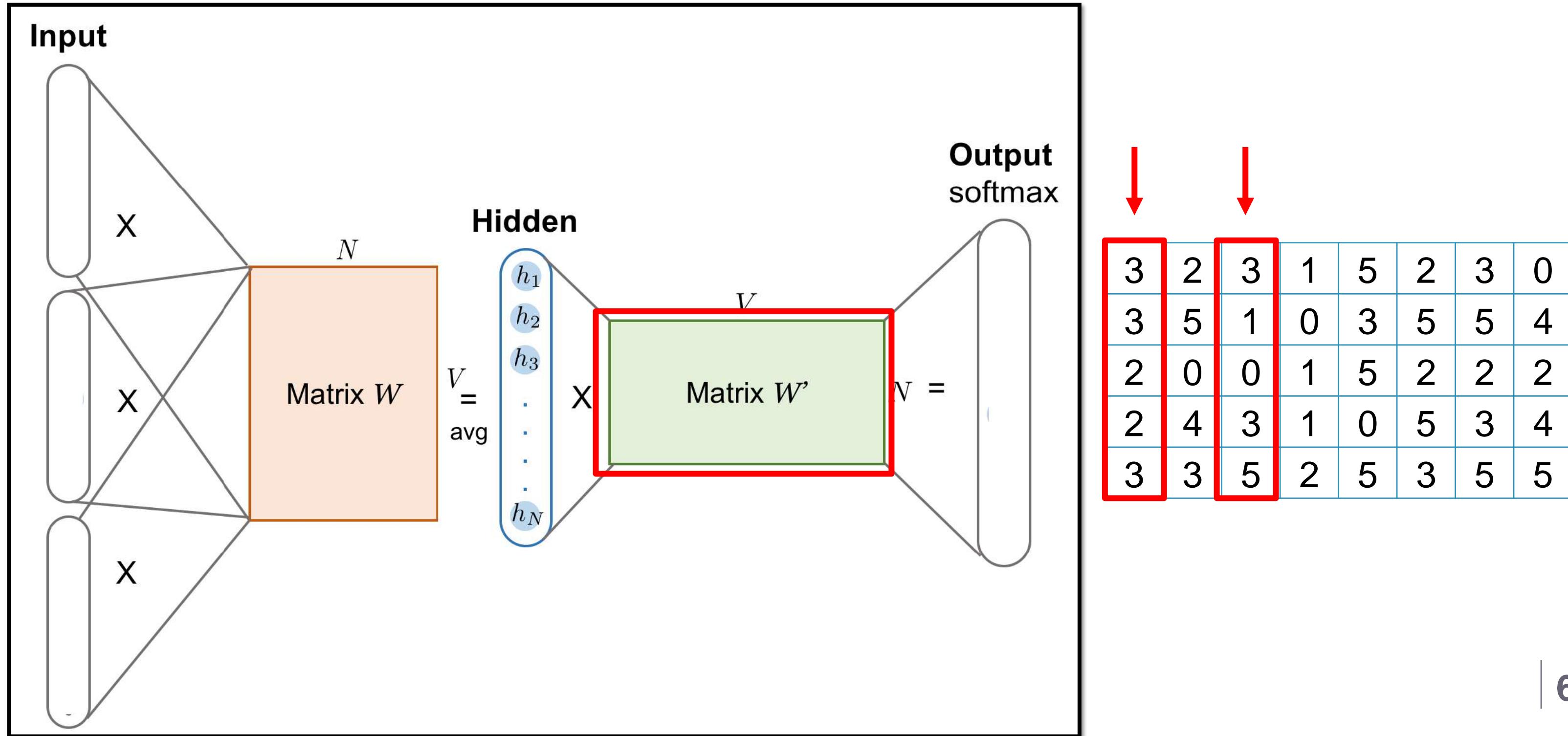


language

0	0	1	0	0	0	0	0
---	---	---	---	---	---	---	---

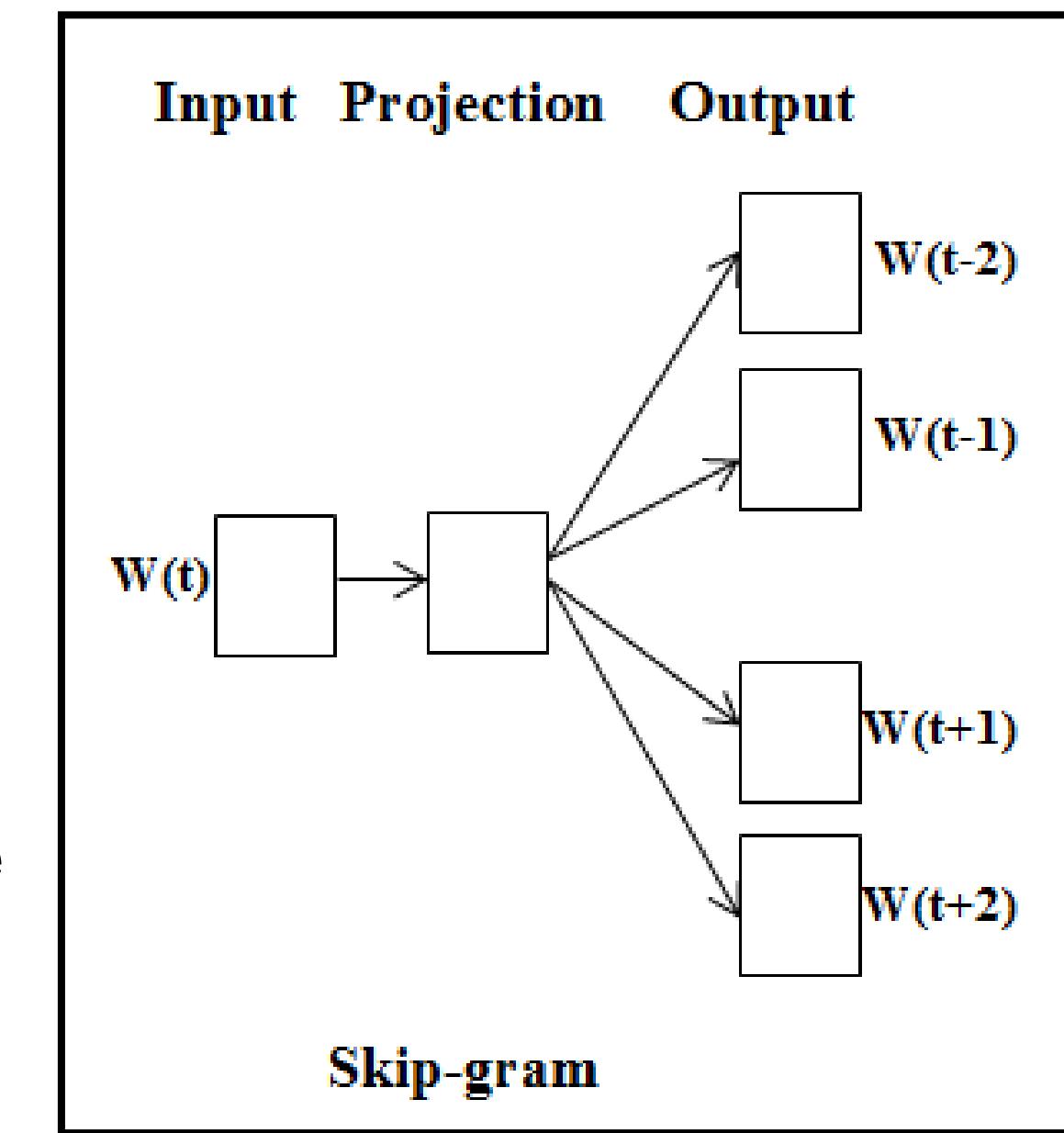
0.00	0.00	0.00	0.00	0.21	0.21	0.58	0.00
natural	language	human	is	complex	and	also	diverse

CBOW



Skip-gram

- The calculations up to hidden layer activations are the same as CBOW
- The difference will be in the target variable
 - Considering a context window of 2 words in each side, there will be **4** one hot encoded target variables and **4** corresponding outputs
 - So we calculate **4** errors and the error vectors obtained are added element-wise to obtain a final error vector which is propagated back to update the weights



Problems with CBOW/Skip-gram

1. For each training sample, **only the weights corresponding to the target word might get a significant update.**
 - The weight corresponding to non-target words would receive a marginal or no change at all
2. For every training sample, **the calculation of the final probabilities using the softmax is quite an expensive operation**
 - Possible solutions
 - Negative sampling
 - Sub sampling

Problems with CBOW/Skip-gram

- Negative Sampling
 - Instead of trying to predict the probability of being a nearby word for all the words in the vocabulary, we try to predict the probability that our training sample words are neighbors or not
 - Referring to our previous example of **(human, language)**, we don't try to predict the probability for human to be a nearby word, we try to predict whether **(human, language)** are nearby words or not
 - Modifying the problem from a **multi-class classification** with N classes into **N binary classification** problem

Problems with CBOW/Skip-gram

- Sub Sampling
 - The distribution of words in a corpus is not uniform. Some words occur more frequently than the other
 - Analyzing the occurrence of words with “**the**” doesn’t tell us much about the meaning of words. “**the**” appears in the context of pretty much every word.
 - We will have many more samples of (“**the**”, ...) than we need to learn a good vector for “**the**”.
 - In sub-sampling, we limit the number of samples for a word by capping their frequency of occurrence. For frequently occurring words, we remove a few of their instances both as a neighboring word and as the input word

CBOW vs. Skip-gram

Skip-gram

- Works well with a small training data
- Represents well for rare words or phrases

CBOW

- Several times faster
- Better accuracy for the frequent words

Word Embedding

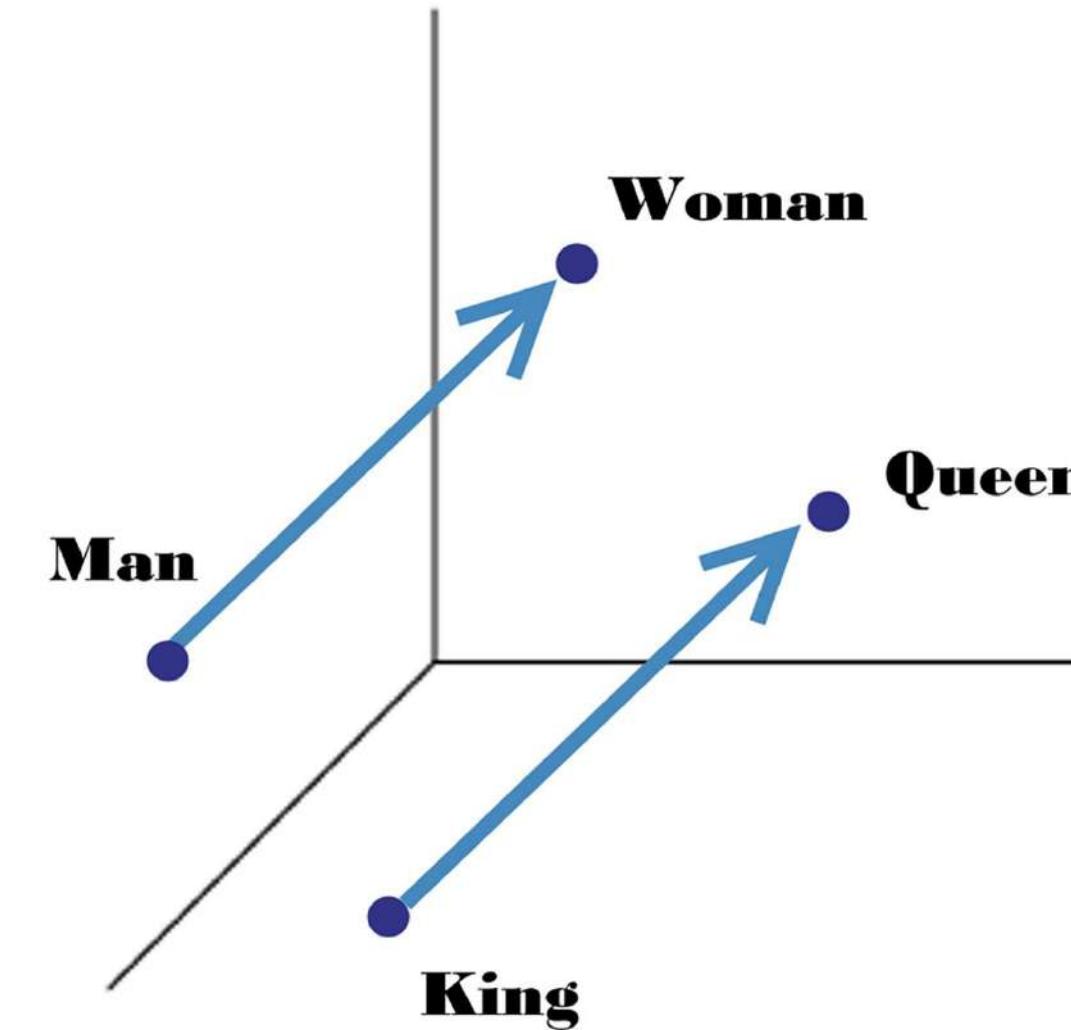
- What is word embedding?
- One-hot word representation
- Distributional word vectors
 - Frequency based
 - Prediction based
- Word embedding evaluation
- Word embedding in Python

Word embedding arithmetic properties

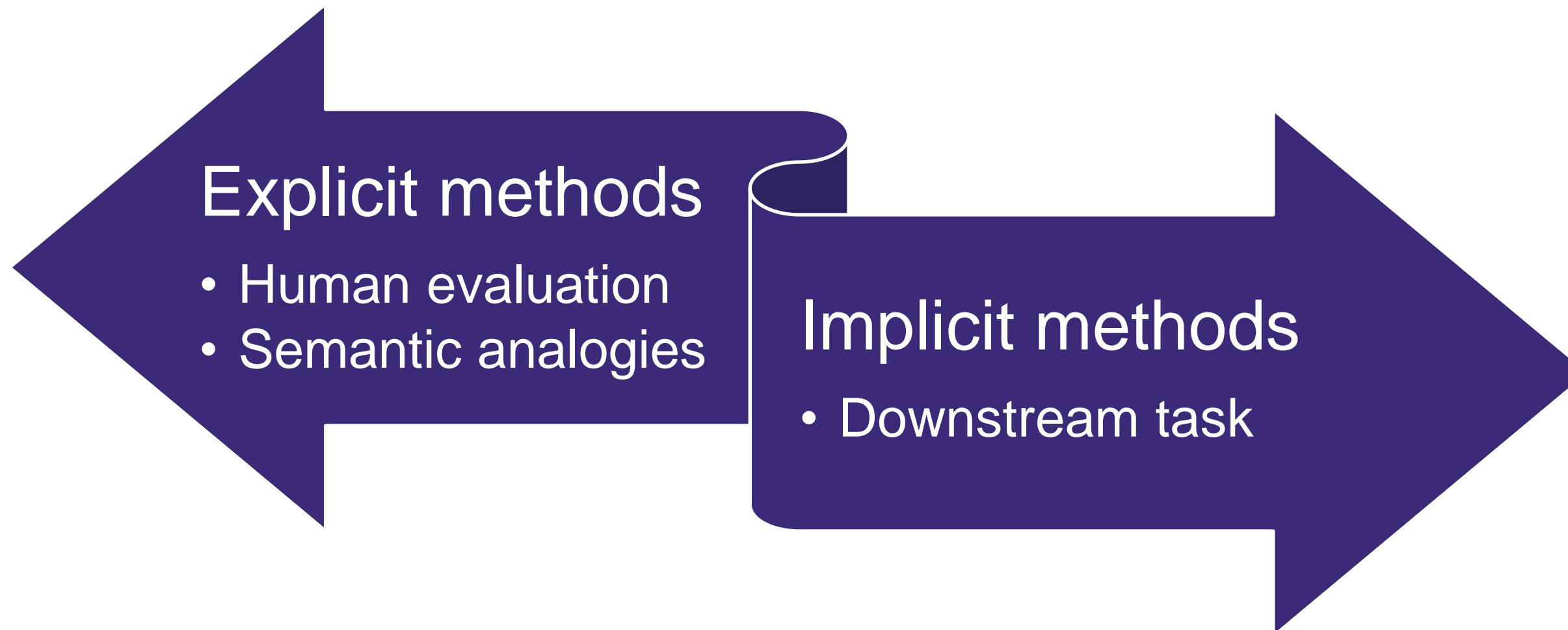
- A surprising property of word vectors is that word analogies can often be solved with vector arithmetic

$$\text{King} - \text{Man} + \text{Woman} = \text{Queen}$$

$$\text{Rome} - \text{Italy} = \text{Berlin} - \text{Germany}$$



Word embedding evaluation



Word embedding evaluation

- Explicit methods
 - Human evaluation
 - Semantic analogies

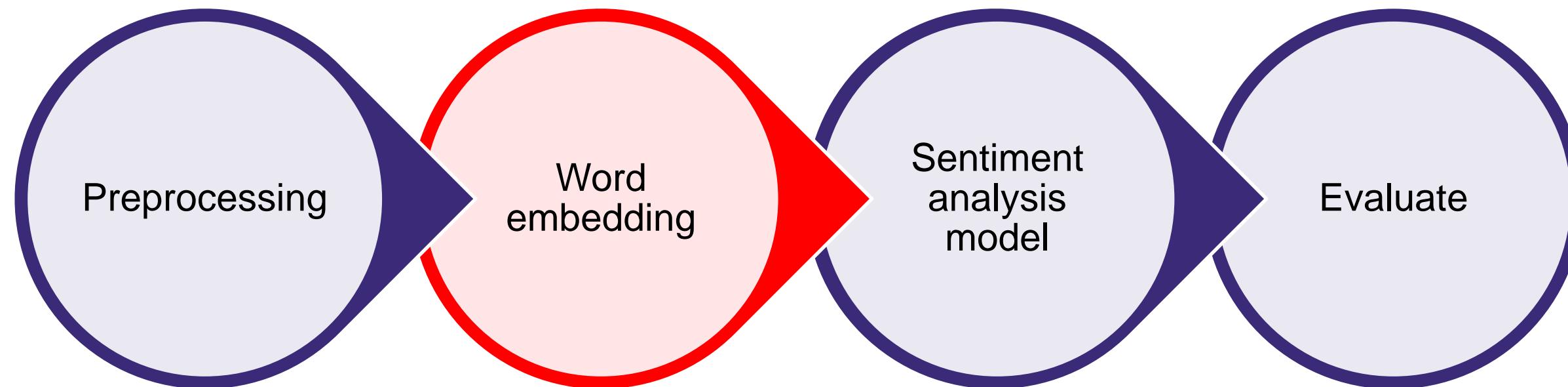
(Man , King) = (Woman , ?)

(Germany , Berlin) = (France , ?)

doctor	nurse	7.00
professor	doctor	6.62
stock	jaguar	0.92
stock	market	8.08
company	stock	7.08

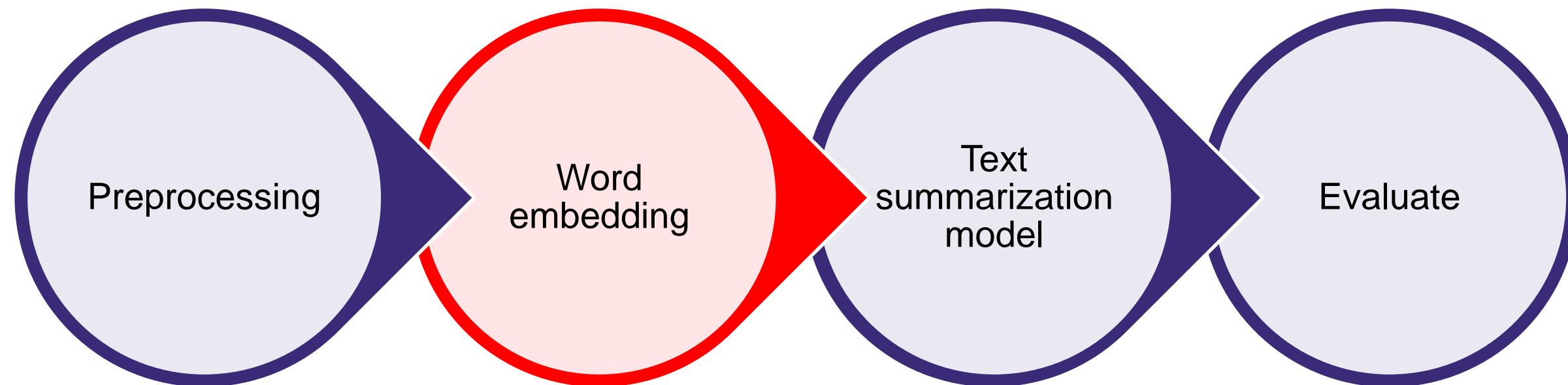
Word embedding evaluation

- Implicit methods
 - Measure performance in a downstream task



Word embedding evaluation

- Implicit methods
 - Measure performance in a downstream task



Word Embedding

- What is word embedding?
- One-hot word representation
- Distributional word vectors
 - Frequency based
 - Prediction based
- Word embedding evaluation
- Word embedding in Python

Word embedding in Python

- Gensim



Gensim

- Train a model

```
>>> from gensim.models import Word2Vec  
>>> model = Word2Vec(sentences=sample_texts, vector_size=100,  
window=5)  
>>> vector = model.wv['computer']  
  
>>> vector.most_similar('computer')  
[('laptop', 0.948005199432373),  
 ('mouse', 0.9403423070907593)]
```

Gensim

- Load a model

```
>>> import gensim.downloader
>>> print(list(gensim.downloader.info()['models'].keys()))
['word2vec-ruscorpora-300',
 'word2vec-google-news-300',
 'glove-wiki-gigaword-50',
 'glove-wiki-gigaword-100',
 ...
 'glove-twitter-100',
 'glove-twitter-200']
>>> word2vec_vectors = gensim.downloader.load('word2vec-google-news-300')
```

Summary

A screenshot of a Google search results page. The search query "how thin is a dollar bill" is entered in the search bar. The first result is a snippet from a website about the thickness of a dollar bill, with the words "inches" highlighted in red. Below the snippet is a link to the source: "https://www.ehd.org > ... > Technology Articles Grasping Large Numbers". The snippet text reads: "1. U.S. paper currency such as a \$1 bill measures 2.61 inches wide by 6.14 inches long with a thickness of .0045 inches." The page also features a sidebar with similar questions like "How thick is a 1 dollar bill?" and "Is a dollar bill two pieces of paper?".



$v = [\text{book}, \text{machine}, \text{artificial}, \text{NLP}, \text{code}]$

machine

0	1	0	0	0
---	---	---	---	---

artificial

0	0	1	0	0
---	---	---	---	---

code

0	0	0	0	1
---	---	---	---	---

Summary

- Distributional word vectors

- Frequency based
- Prediction based

Memes generally replicate through exposure to humans, who have evolved as efficient copiers of information and behavior.

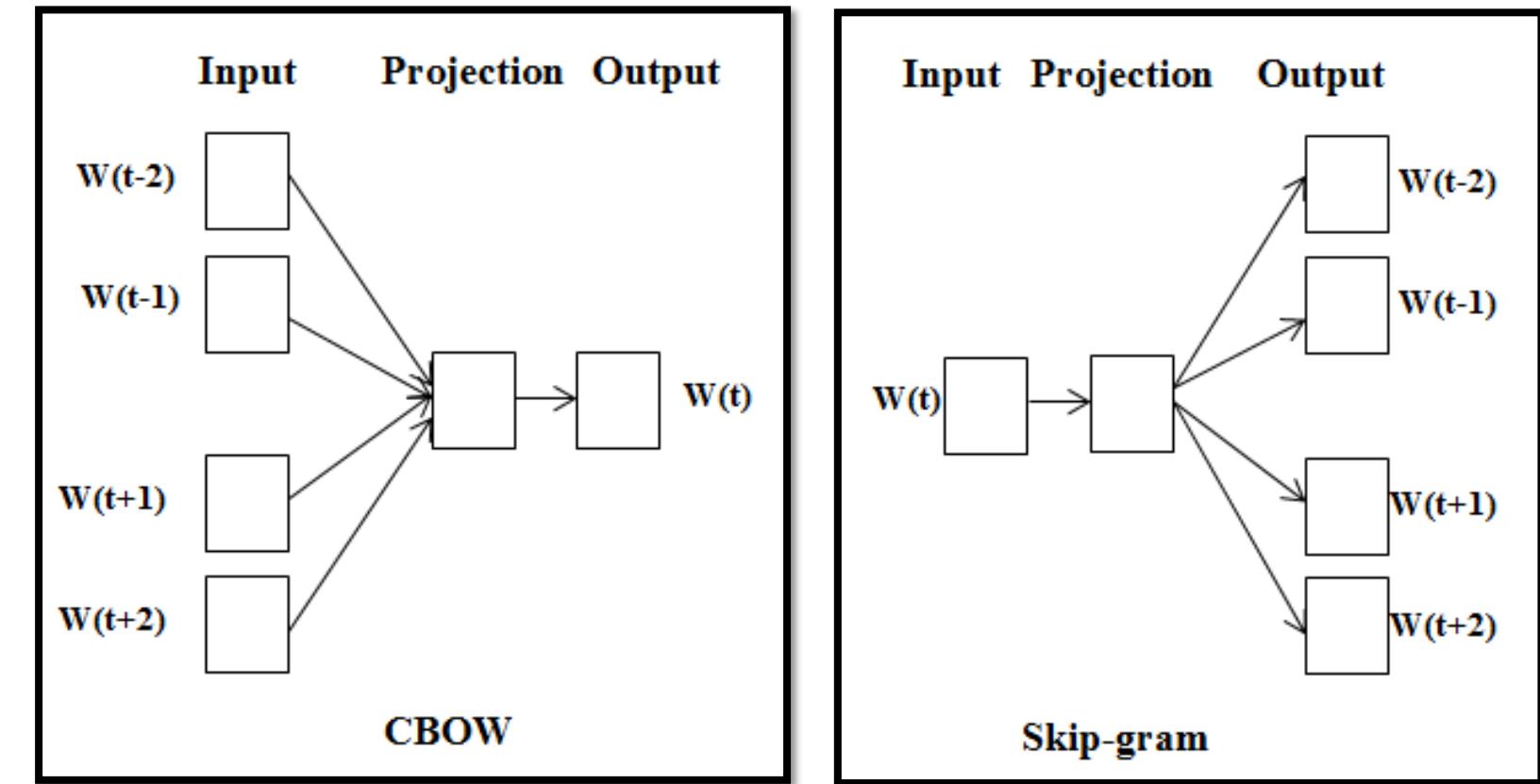
- Frequency based

- Document-Term matrix
- Term-Term matrix
- PMI

Why is the water in the glass?
Drinking a glass of milk is part of maintaining a healthy diet

Summary

- Prediction based (dense word embedding)
 - **Word2vec**



Explicit methods

- Human evaluation
- Semantic analogies

Implicit methods

- Downstream task

„KI-Campus – Die Lernplattform für Künstliche Intelligenz“ ist ein Projekt von



www.ki-campus.org

Text Corpus

Salar Mohtaj | DFKI

Text corpus

- What is a text corpus?
- Sample text corpora
- Corpus annotation
- Underfitting and overfitting
- Data splitting
- Text corpora in Python

Text corpus

- What is a text corpus?
- Sample text corpora
- Corpus annotation
- Underfitting and overfitting
- Data splitting
- Text corpora in Python

What is a text corpus?

- Text corpus is a collection of text, usually contains several documents
 - Wikipedia articles
 - Collection of movies reviews
 - Internet comments
 - Collection of tweets
- A corpus may be quite small, for example, containing only **thousands words** of text, or very large, containing **millions of words**
- NLP corpora are in different standards based on the target task

What is a text corpus?

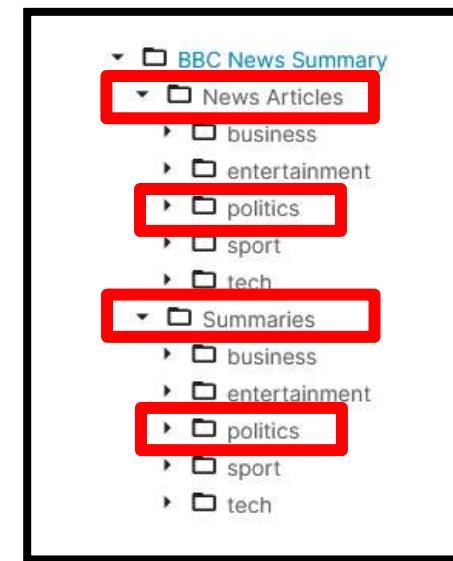
- Text corpora can be compared based on different factors
 - **Size**: larger corpora are better for training deep learning models
 - **Domain**: unless corpus has been collected for specific tasks, it should include different genres such as newspapers, magazines, blogs, academic journals, etc
 - **Metadata**: metadata should indicate the sources, assumptions, limitations and what's included in the corpus
 - **Clean**: a wordlist giving word forms of the same word can be messy to process

Text corpus

- What is a text corpus?
- Sample text corpora
- Corpus annotation
- Underfitting and overfitting
- Data splitting
- Text corpora in Python

Sample text corpora

- Common file formats
 - TXT
 - CSV
 - JSON
 - XML



```

<?xml version="1.0" encoding="UTF-8"?>
<art orl="ja" trl="en">
<inf>jawiki-20080607-pages-articles.xml</inf>
<tit>
<j>雪舟</j>
<e type="trans" ver="1">Sesshu</e>
<cmt></cmt>
<e type="trans" ver="2">Sesshu</e>
<cmt>修正なし</cmt>
<e type="check" ver="1">Sesshu</e>
<cmt>修正なし</cmt>
</tit>
<par id="1">

<j>雪舟 (せっしゅう、1420年 (応永27年) ~ 1506年 (永正3年) ) は号で、15世紀後半室町時代に活躍した水墨画家・禅僧で、画聖とも称えられる。</j>
<e type="trans" ver="1">Known as Sesshu (1420 (Oei Year 27) - 1506 (Eisho year 3)), he was an ink painter and Zen monk active in the Muromachi period in the 2nd half of the 15th century. He is known for his landscape paintings and calligraphy. He is also known as Sesshū Tōyō. </e>
<cmt></cmt>
<e type="trans" ver="2">Known as Sesshu (1420 - 1506), he was an ink painter and Zen monk active in the Muromachi period in the latter half of the 15th century, and was called a 'Zen master'. </e>
<cmt>西洋層のみとする。自然な表現に修正。</cmt>
<e type="check" ver="1">Known as Sesshu (1420 - 1506), he was an ink painter and Zen monk active in the Muromachi period in the latter half of the 15th century, and was called a 'Zen master'. </e>
<cmt>フィードバックに基づき翻訳を修正しました。</cmt>
</sen>
<j>日本の水墨画を一変させた。</j>
<e type="trans" ver="1">He changed Japanese ink painting.</e>
<cmt></cmt>
<e type="trans" ver="2">He changed Japanese ink painting.</e>
<cmt>修正なし</cmt>
<e type="check" ver="1">He revolutionized the Japanese ink painting.</e>
<cmt>フィードバックに基づき翻訳を修正しました。</cmt>
  
```

topic	link	domain	published...	title	lang
SCIENCE	https://www.cnn.com/2020/08/03/us/zombie-cicadas-west-virginia-fungus-scn-trnd/index.html	cnn.com	2020-08-03 21:59:00	'Zombie cicadas' under the influence of a mind controlling fungus have returned to West Virginia	en
SCIENCE	https://www.sciencealert.com/astronomers-may-have-just-found-a-missing-baby-neutron-star	sciencealert.co	2020-08-04 06:01:29	Astronomers May Have Found a Lost Neutron Star That's Been Missing For Decades	en
SCIENCE	https://www.somagnews.com/spaceship-starship-spacecraft-saw-150-meters-high/	somagnews.com	2020-08-05 17:12:00	SpaceX's Starship spacecraft saw 150 meters high	en
SCIENCE	https://www.sciencealert.com/these-orbs-look-like-candy-but-they're-actually-different-flavours-of-a...	sciencealert.co	2020-08-13 23:23:31	These Orbs Look Like Candy, But They're Actually Different Flavours of Phobos	en
TECHNOLOGY	https://www.kotaku.com.au/2020/08/come-see-what-its-like-to-play-microsoft-flight-simulator/	kotaku.com.au	2020-08-14 06:29:00	Come See What It's Like To Play Microsoft Flight Simulator	en

author	date	headlines	read_more	text	ctxext
Chhavi Tyagi	03 Aug 2017, Thursday	Daman & Diu revokes mandatory Rakshabandhan in offices order	http://www.hindustantimes.com/india-news/rakshabandhan-compulsory-in-daman-and-dui-women-employees-t...	The Daman and Diu administration on Wednesday withdrew a circular that asked women staff to tie rakhi...	The Daman and Diu administration on Wednesday withdrew a circular that asked women staff to tie rakhi...
Daisy Mowke	03 Aug 2017, Thursday	Malaika slams user who trolled her for 'divorcing rich man'	http://www.hindustantimes.com/bollywood/malaika-arora-khan-was-trolled-for-divorcing-a-rich-man-h...	Malaika Arora slammed an Instagram user who trolled her for "divorcing a rich man" and "having fun w..."	From her special numbers to TV appearances, Bollywood actor Malaika Arora Khan has managed to carve ...
Arshiya Chopra	03 Aug 2017, Thursday	'Virgin' now corrected to 'Unmarried' in IGIMS' form	http://www.hindustantimes.com/patna/bihar-igims-form-looses-virginity-after-row-opted-for-unmarried-in...	The Indira Gandhi Institute of Medical Sciences (IGIMS) in Patna on Thursday made corrections in its...	The Indira Gandhi Institute of Medical Sciences (IGIMS) in Patna amended its marital declaration for...
Sumedha Sehra	03 Aug 2017, Thursday	Aaj apne pakdiya: LeT man Dujana before being killed	http://indiadaily.intoday.in/story/abu-dujana-last-phone-call-lashkar-e-Taiba-jammu-and-kashmir/1/10...	Lashkar-e-Taiba's Kashmir commander Abu Dujana, who was killed by security forces, said 'Kabhi hum a...	Lashkar-e-Taiba's Kashmir commander Abu Dujana was killed in an encounter in a village in Pulwama di...

Sample text corpora

- Where to find text corpora?
- NLP competitions
 - NLP shared task
- Kaggle
- Active NLP groups websites

The SemEval-2020 website features a navigation bar with links to Home, Tasks, Program, CodaLab, Paper Submissions, and Frequently Asked Questions. The main content area is titled "SemEval-2020 International Workshop on Semantic Evaluation Sponsored by SIGLEX". It lists several tasks under categories such as Lexical semantics, Common Sense Knowledge and Reasoning, Knowledge Extraction, Humour, Emphasis, and Sentiment, and Societal Applications of NLP. Each task has a link to its mailing list and email organizers. On the right side, there is a "Contact Info" section with a list of organizers and their institutions, and a "Specific tasks" section with red links to the task pages. A "Text" and "Announcements" sidebar provides additional information.

The Stanford Natural Language Inference (SNLI) Corpus website has a header with the Stanford logo and links to people, publications, research blog, software, teaching, join, and local. The main content is titled "The Stanford Natural Language Inference (SNLI) Corpus". It includes a "New" section about the MultiGenre NLI (MultiNLIP) Corpus. Below it is a "The Corpus" section with a detailed description of the corpus, mentioning its size and purpose. A "Text" and "Judgments" table shows example sentence pairs with their judgments from five MT workers and a consensus judgment. The table columns are Text, Judgments (contradiction, neutral, contradiction), and Hypothesis.

The PAN website has a dark header with the PAN logo and links to SHARED TASKS, EVENTS, DATA, PUBLICATIONS, and ORGANIZATION. The main content area features a banner stating "PAN is a series of scientific events and shared tasks on digital text forensics and stylometry". Below this are two boxes: "PAN at CLEF 2021" containing links to Overview, Authorship Verification, Profiling Hate Speech Spreaders on Twitter, and Style Change Detection; and "PAN Online" containing links to Authorship Obfuscation, Hyperpartisan News Detection, and Plagiarism Detection - Text Alignment. Each box has a "REGISTER NOW" button at the bottom.

The Leipzig Corpora Collection Download Page has a header with the WORTSCHATZ logo and a search bar. It includes links to Download Corpora, Download SentiWS, Download TinyCC, and Download ASV Toolbox. The main content is titled "Leipzig Corpora Collection Download Page" and provides information about the collection, including its purpose and the format of the data. It also includes a note about citation requirements and a link to a paper by D. Goldhahn, T. Eckart & U. Quasthoff. At the bottom, there is a language selection menu with options for German, English, French, Arabic, Russian, and All Languages.

Text corpus

- What is a text corpus?
- Sample text corpora
- **Corpus annotation**
- Underfitting and overfitting
- Data splitting
- Text corpora in Python

Corpus annotation

- It would happen that you have to compile your own corpus for the task and domain of interests
 - e.g., generating a text summarization corpus for medical texts in German
- Apart from the pure text, a corpus can also be provided with additional linguistic information, called **annotation**
- Corpus annotation is the practice of adding interpretative linguistic information to a piece of text

Corpus annotation

The screenshot shows a user interface for corpus annotation. On the left, there is a sidebar with a search bar labeled "Search document" and a dropdown arrow. Below it, the text "About 6 results" is displayed. A list of six movie reviews is shown, each with a checkmark and a truncated text snippet:

- Fair drama/love story movie that focuses on the lives of blu...
- If you like adult comedy cartoons, like South Park, then thi...
- I came in in the middle of this film so I had no idea about ...
- Story of a man who has unnatural feelings for a pig. Starts ...
- Robert DeNiro plays the most unbelievably intelligent illite...
- From the beginning of the movie, it gives the feeling the di...

The main content area shows the first review in detail. At the top, there is a navigation bar with a progress bar showing "1/6" and a small icon. Below the progress bar is a row of buttons: "Negative" (red), "n" (white), "Positive" (blue), and "p" (white). The text of the review is enclosed in a large red box:

If you like adult comedy cartoons, like South Park, then this is nearly a similar format about the small adventures of three teenage girls at Bromwell High. Keisha, Natella and Latrina have given exploding sweets and behaved like bitches, I think Keisha is a good leader. There are also small stories going on with the teachers of the school. There's the idiotic principal, Mr. Bip, the nervous Maths teacher and many others. The cast is also fantastic, Lenny Henry's Gina Yashere, EastEnders Chrissie Watts, Tracy-Ann Oberman, Smack The Pony's Doon Mackichan, Dead Ringers' Mark Perry and Blunder's Nina Conti. I didn't know this came from Canada, but it is very good. Very good!

At the bottom of the main content area are two navigation buttons: "< Prev" and "Next >".

Corpus annotation

The screenshot shows a corpus annotation interface. On the left, a sidebar displays a search bar and a list of results. The main area shows a document with a specific sentence highlighted in blue and framed by a red border. Below the document, there is a response field with two versions of the same sentence in French, also framed by a red border. Navigation buttons for 'Prev' and 'Next' are at the bottom.

Search document:

About 6 results

✓ If it had not been for his help, I would have failed....

According to this magazine, my favorite actress will marry a...

It's not always possible to eat well when you are traveling ...

It's still early. We should all just chill for a bit....

She got a master's degree three years ago....

We adopted an alternative method....

If it had not been for his help, I would have failed.

What is your response?

S'il ne m'avait pas aidé, j'aurais échoué.

S'il ne m'avait pas aidée, j'aurais échoué.

< Prev

Next >

Corpus annotation

- The annotation process could meet lots of questions and ambiguities
- Annotation guideline
 - Describe the annotation procedure as generic as possible but as precise as necessary
 - So that human annotators can annotate the concept or phenomenon in any text without running into problems or ambiguity issues

PROPOSED SEMANTIC-ROLE BASED SENTIMENT QUESTIONNAIRE

Q1. From reading the text, the speaker's emotional state can best be described as:

- *positive state*: there is an explicit or implicit clue in the text suggesting that the speaker is in a positive state, i.e., happy, admiring, relaxed, forgiving, etc.
- *negative state*: there is an explicit or implicit clue in the text suggesting that the speaker is in a negative state, i.e., sad, angry, anxious, violent, etc.
- *both positive and negative, or mixed, feelings*: there is an explicit or implicit clue in the text suggesting that the speaker is experiencing both positive and negative feelings
- *unknown state*: there is no explicit or implicit indicator of the speaker's emotional state

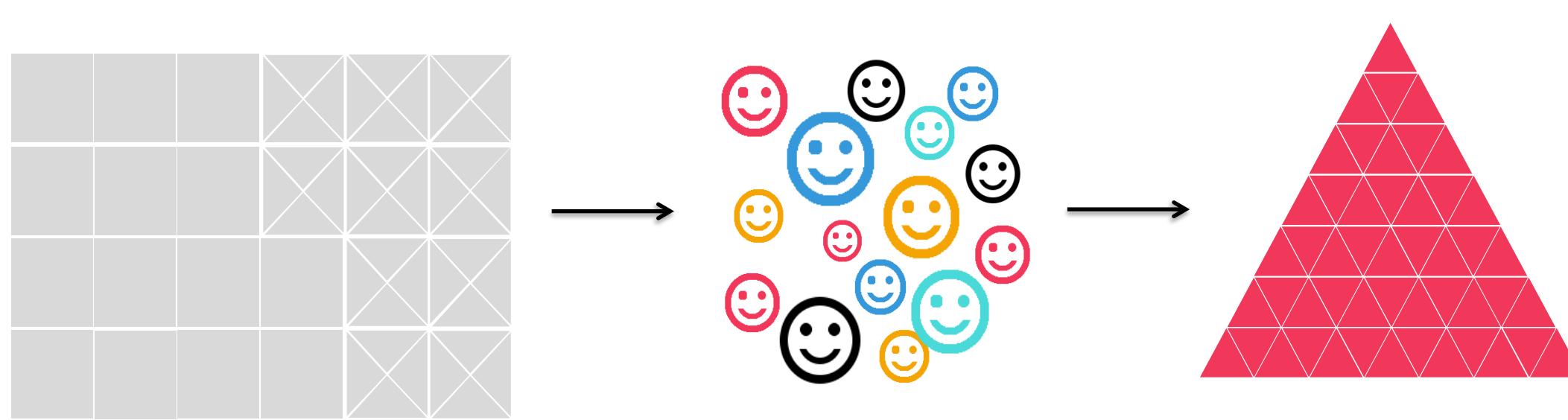
Q2. From reading the text, identify the entity towards which opinion is being expressed or the entity towards which the speaker's attitude can be determined.

Corpus annotation

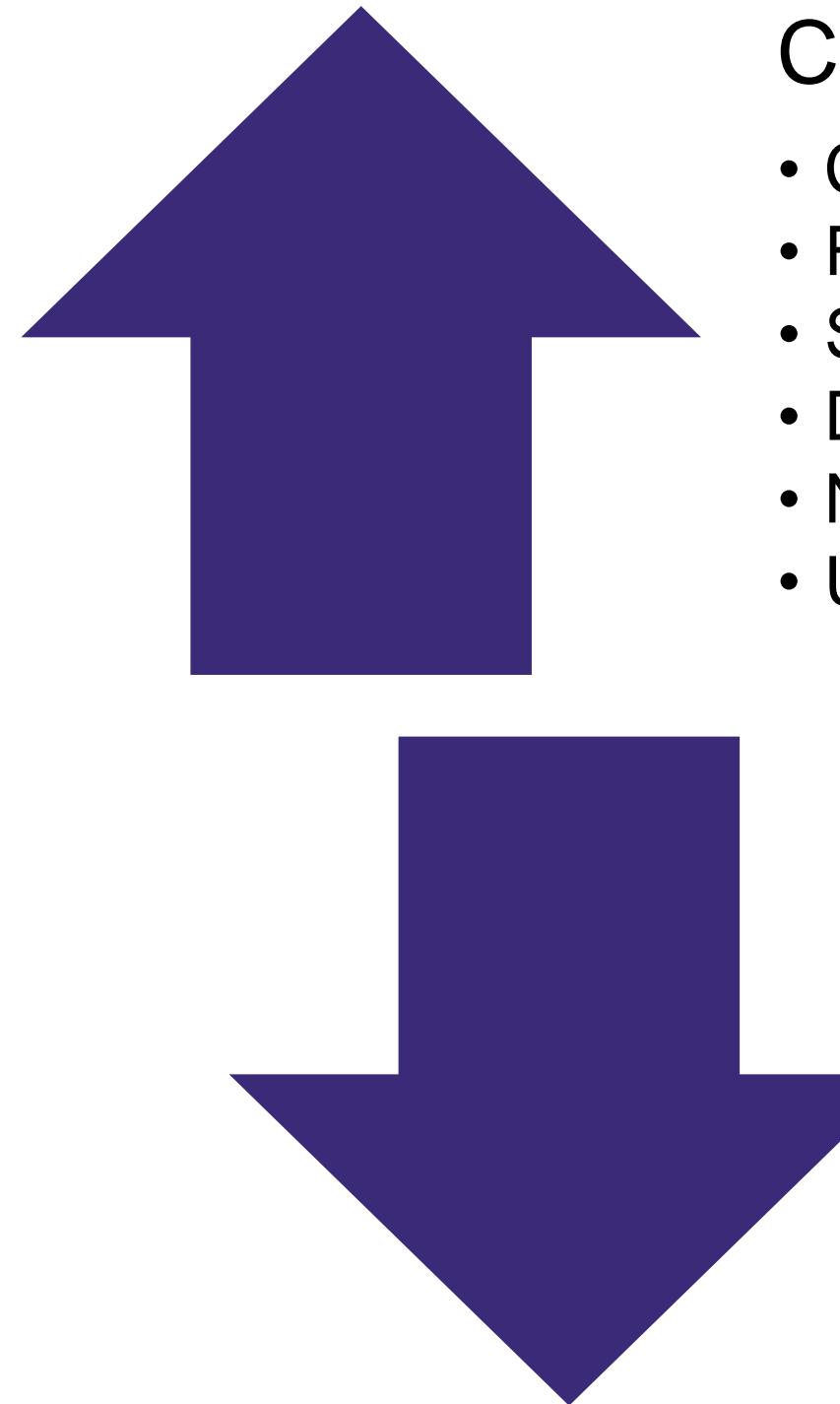
- Laboratory vs. crowdsourcing
 - Laboratory
 - asking experts (e.g., linguists) for data annotation
 - Crowdsourcing
 - Is a participatory method of building a dataset with the help of a large group of people

Corpus annotation

- Crowdsourcing



Corpus annotation



Crowdsourcing

- Cheaper
- Faster
- Scalable
- Diverse group of participants
- No moderator
- Unreliable data

Laboratory

- High reliability
- Controlled unwanted factors
- Environment constant, no noise
- Participants: one-to-one contact to moderator

Text corpus

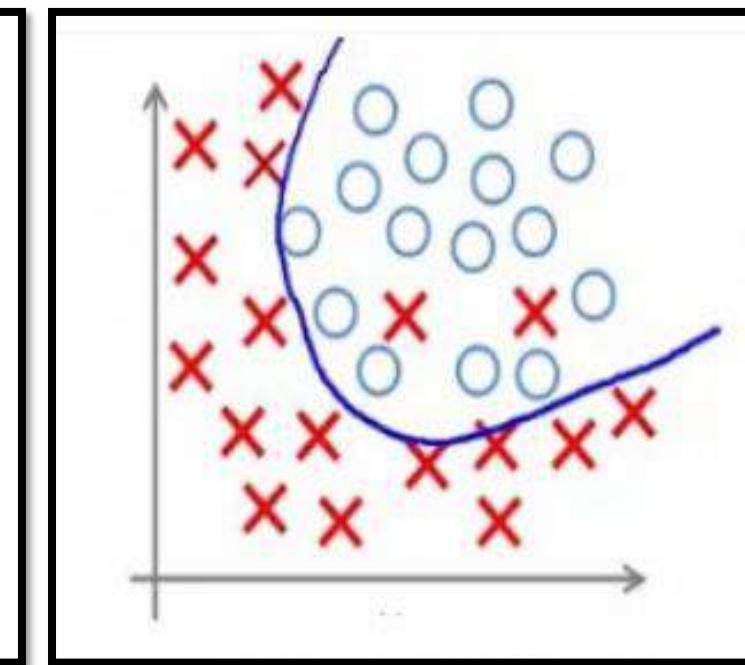
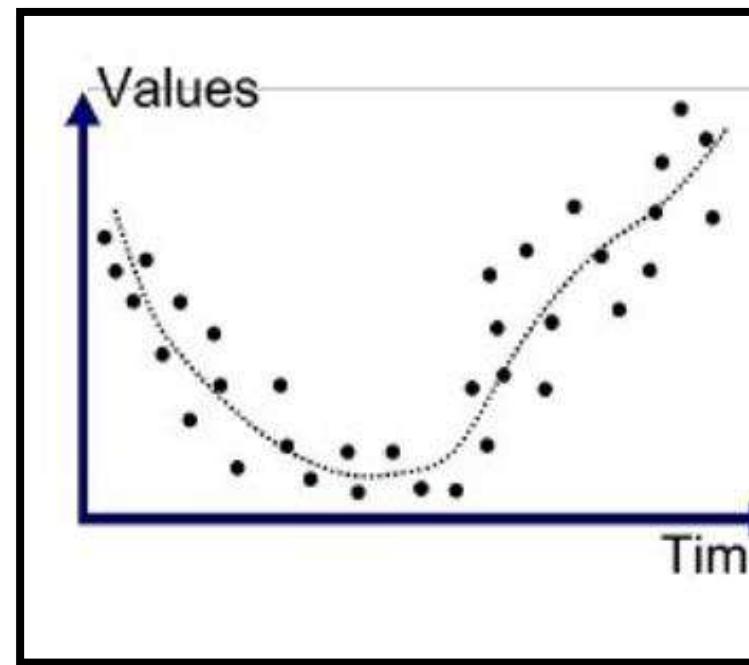
- What is a text corpus?
- Sample text corpora
- Corpus annotation
- Underfitting and overfitting
- Data splitting
- Text corpora in Python

Underfitting and overfitting

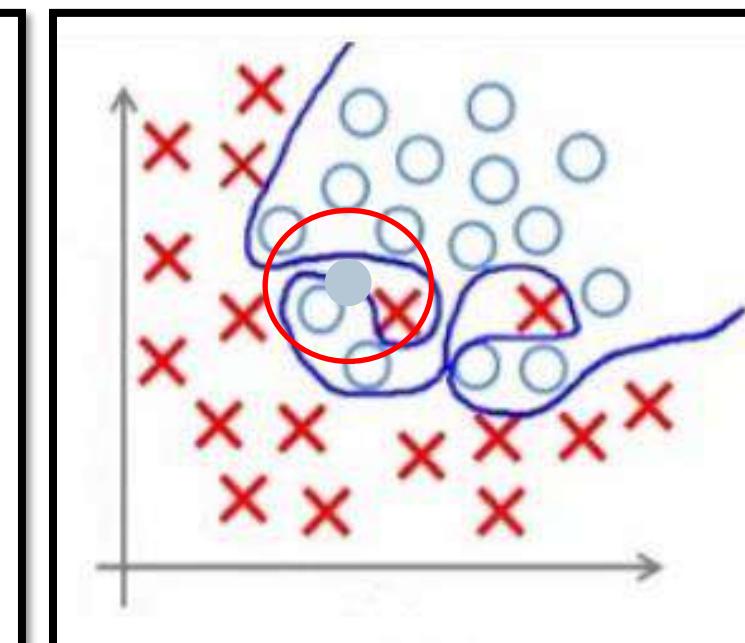
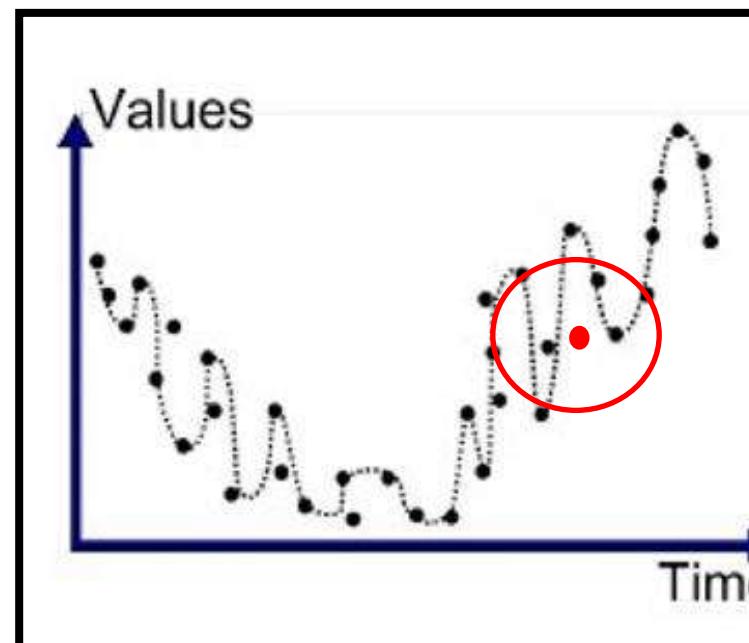
- Overfitting
 - Overfitting refers to a model that models the training data too well
 - It happens when a model learns the detail and noise in the training data to the extent that it negatively impacts the performance of the model on new data
 - This means that the noise or random fluctuations in the training data is picked up and learned as concepts by the model
 - The problem is that these concepts do not apply to new data and negatively impact the models ability to generalize

Underfitting and overfitting

- Overfitting



Appropriate-fitting



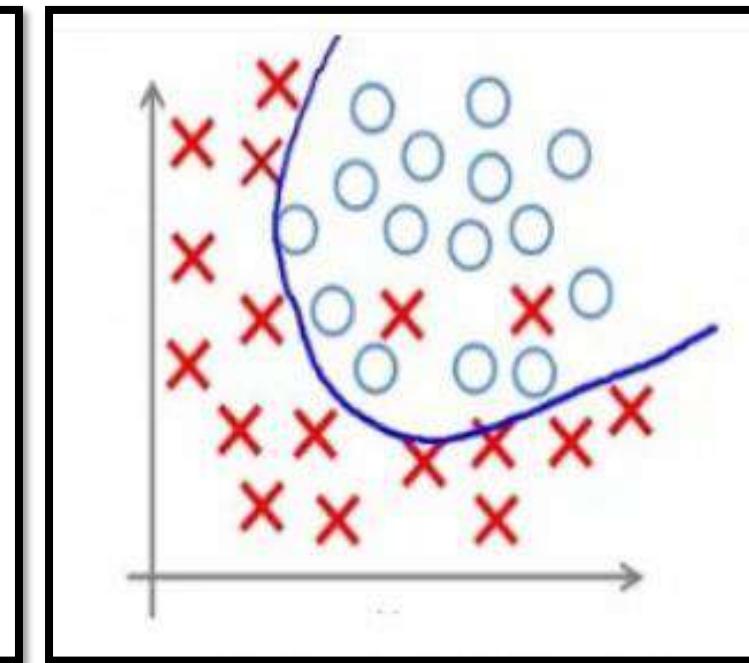
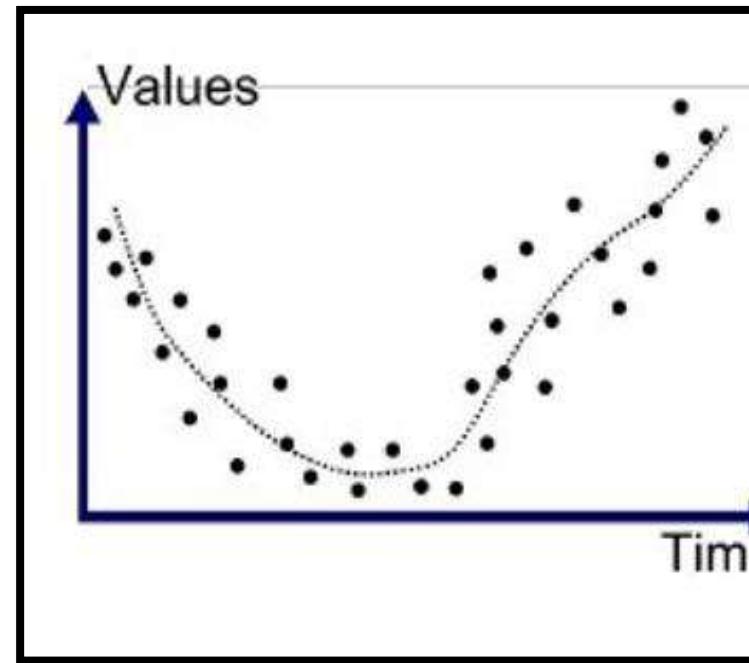
Overfitting

Underfitting and overfitting

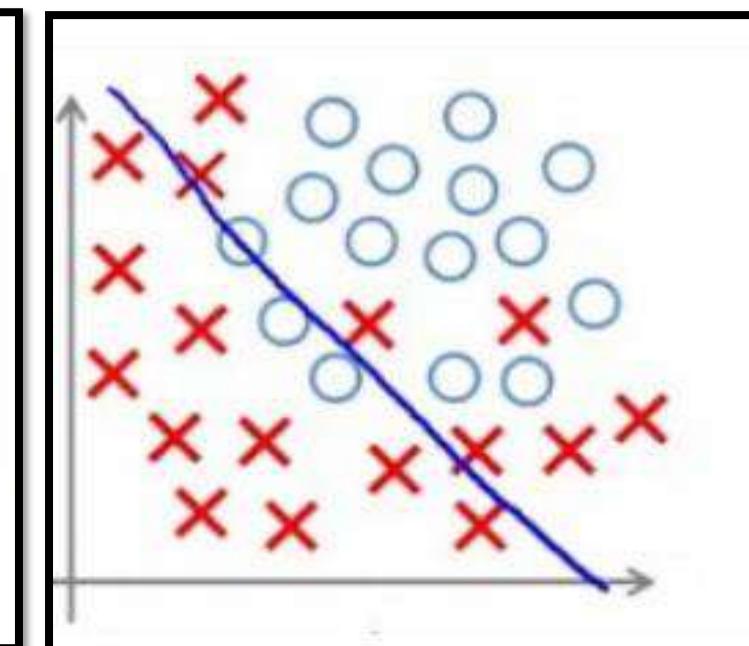
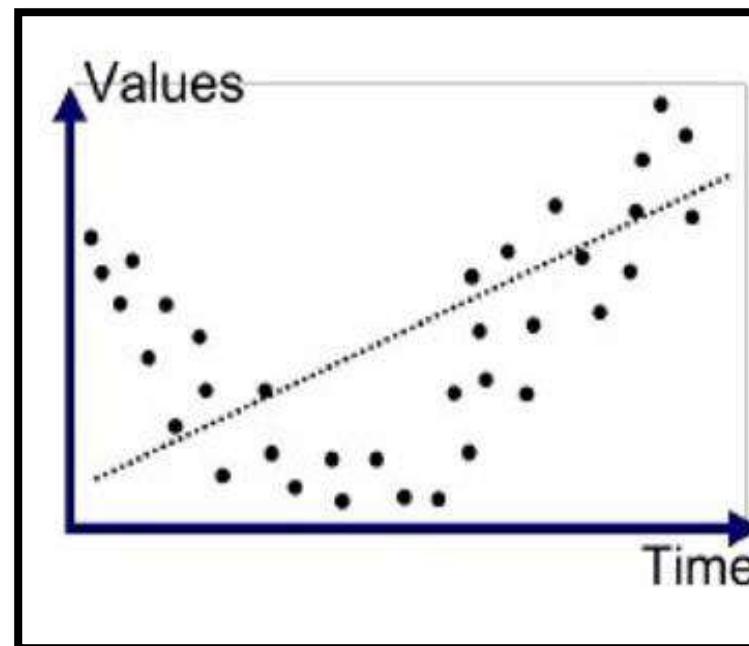
- Underfitting
 - Underfitting refers to a model that can neither model the training data nor generalize to new data
 - An underfit machine learning model is not a suitable model and will be obvious as it will have poor performance on the training data

Underfitting and overfitting

- Underfitting



Appropriate-fitting



Underfitting

Underfitting and overfitting

- Overfitting
 - Increase training data
 - Reduce model complexity
 - Cross-validation
 - Early stopping during the training phase
 - Regularization
- Underfitting
 - Increase model complexity
 - Increase number of features
 - Remove noise from the data

Text corpus

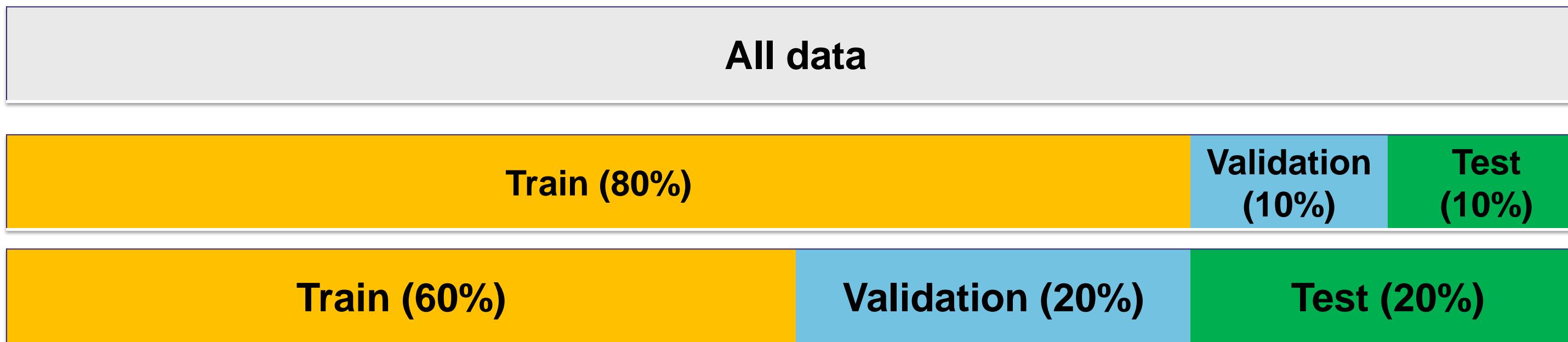
- What is a text corpus?
- Sample text corpora
- Corpus annotation
- Underfitting and overfitting
- **Data splitting**
- Text corpora in Python

Data splitting

- As a common practice in machine learning (NLP), we have to split the dataset (corpus) into different part in order to train a model and test it
- Two main approaches for splitting data
 - Train / Validation / Test
 - Cross validation

Data splitting

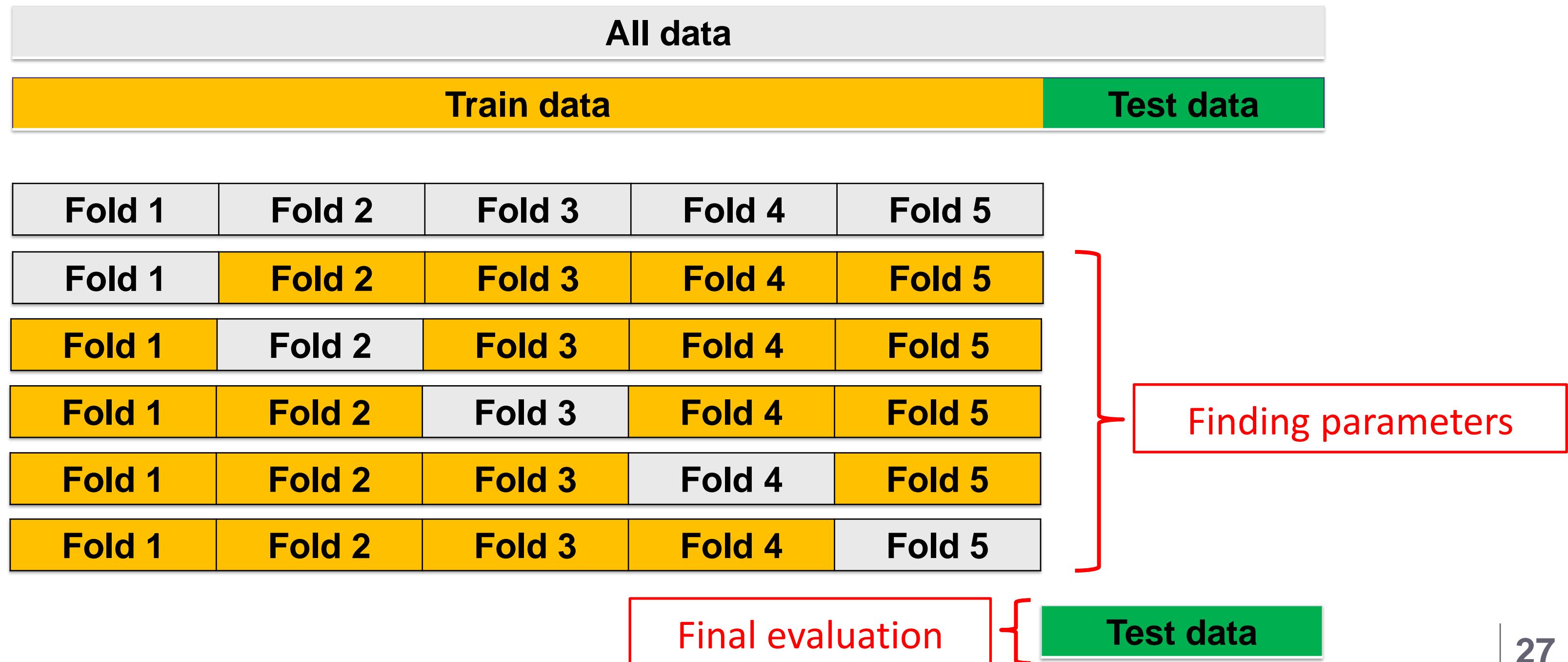
- Train / Validation / Test
 - **Training Dataset:** The sample of data used to fit the model
 - **Validation Dataset:** The sample of data used to provide an unbiased evaluation of a model fit on the training dataset while tuning model hyperparameters
 - **Test Dataset:** The sample of data used to provide an unbiased evaluation of a final model fit on the training dataset



Data splitting

- Cross validation
 - Cross-validation is a resampling procedure used to evaluate machine learning models on a limited data sample
 - The procedure has a single parameter called k that refers to the number of groups that a given data sample is to be split into
 - the procedure is often called k -fold cross-validation

Data splitting



Text corpus

- What is a text corpus?
- Sample text corpora
- Corpus annotation
- Underfitting and overfitting
- Data splitting
- Text corpora in Python

Text corpora in Python

- Reuters Corpus
- Brown Corpus
- Web and Chat Text
- Gutenberg Corpus
- ...
- And Corpora in Other Languages



```
>>> from nltk.corpus import reuters
>>> reuters.categories()
['acq', 'alum', 'barley', 'bop',
'carcass', 'castor-oil', 'cocoa',
'coconut', 'coconut-oil', 'coffee', ...]
>>> from nltk.corpus import brown
>>> brown.words()
['The', 'Fulton', 'County', 'Grand',
'Jury', 'said', ...]
```

Summary

- A corpus is a collection of texts, **written or spoken**



[Home](#) | [Tasks](#) | [Program](#) | [CodaLab](#) | [Paper Submissions](#) | [Frequently Asked Questions](#)

SemEval-2020
International Workshop on Semantic Evaluation

Sponsored by SIGLEX

Tasks

We are pleased to announce the following tasks in SemEval-2020.

Lexical semantics

- [Task 1: Unsupervised Lexical Semantic Change Detection \[mailing list\] \[email organizers\]](#)
- [Task 2: Predicting Multilingual and Cross-Lingual \(Graded\) Lexical Entailment \[mailing list\] \[email organizers\]](#)
- [Task 3: Graded Word Similarity in Context \(GWSC\) \[discussion forum\] \[mailing list\] \[email organizers\]](#)

Common Sense Knowledge and Reasoning, Knowledge Extraction

- [Task 4: Commonsense Validation and Explanation \[mailing list\] \[email organizers\]](#)
- [Task 5: Modeling Causal Reasoning in Language: Detecting Counterfactuals \[mailing list\] \[email organizers\]](#)
- [Task 6: SemEval-Extracting Definitions from Free Text in Textbooks \[mailing list\] \[email organizers\]](#)

Humour, Emphasis, and Sentiment

- [Task 7: Assessing Humor in Edited News Headlines \[mailing list\] \[email organizers\]](#)
- [Task 8: Emotion Analysis \[mailing list\] \[email organizers\]](#)
- [Task 9: Sentiment Analysis for Code-Mixed Social Media Text \[mailing list\] \[email organizers\]](#)
- [Task 10: Emphasis Selection for Written Text in Visual Media \[mailing list\] \[email organizers\]](#)

Societal Applications of NLP

- [Task 11: Detection of Propaganda Techniques in News Articles \[mailing list\] \[email organizers\]](#)
- [Task 12: OffensEval 2: Multilingual Offensive Language Identification in Social Media \[mailing list\] \[email organizers\]](#)

Contact Info

Organizers

- Aurelio Herbelot, University of Trento
- Xianan Zhu, Queen's University
- Nathan Schneider, Georgetown University
- Alexis Palmer, University of North Texas
- Jonathan May, ISI, University of Southern California
- Ekatrina Shatova, University of Amsterdam

Email

- Specific tasks: See red links on the tasks page
- General SemEval organization: semeval-organizers@googlegroups.com

Other Info

Announcements

- 2020/12/11: Task and paper awards announced
- 2020/11/29: Program announced
- 2020/06/23: Camera-ready deadlines extended (July 24, July 31)
- 2020/04/25: Paper submission deadlines extended (May 1, May 22)
- 2020/03/31: Paper submission deadlines extended (May 1, May 8)
- 2020/03/31: COLING 2020 announces new dates in December

PROPOSED SEMANTIC-ROLE BASED SENTIMENT QUESTIONNAIRE

Q1. From reading the text, the speaker's emotional state can best be described as:

- **positive state:** there is an explicit or implicit clue in the text suggesting that the speaker is in a positive state, i.e., happy, admiring, relaxed, forgiving, etc.
- **negative state:** there is an explicit or implicit clue in the text suggesting that the speaker is in a negative state, i.e., sad, angry, anxious, violent, etc.
- **both positive and negative, or mixed, feelings:** there is an explicit or implicit clue in the text suggesting that the speaker is experiencing both positive and negative feelings
- **unknown state:** there is no explicit or implicit indicator of the speaker's emotional state

Q2. From reading the text, identify the entity towards which opinion is being expressed or the entity towards which the speaker's attitude can be determined.

Search document

About 6 results

Fair drama/love story movie that focuses on the lives of blu...

If you like adult comedy cartoons, like South Park, then thi...

I came in in the middle of this film so I had no idea about ...

Story of a man who has unnatural feelings for a pig. Starts ...

Robert DeNiro plays the most unbelievably intelligent ille...

From the beginning of the movie, it gives the feeling the di...

1/6

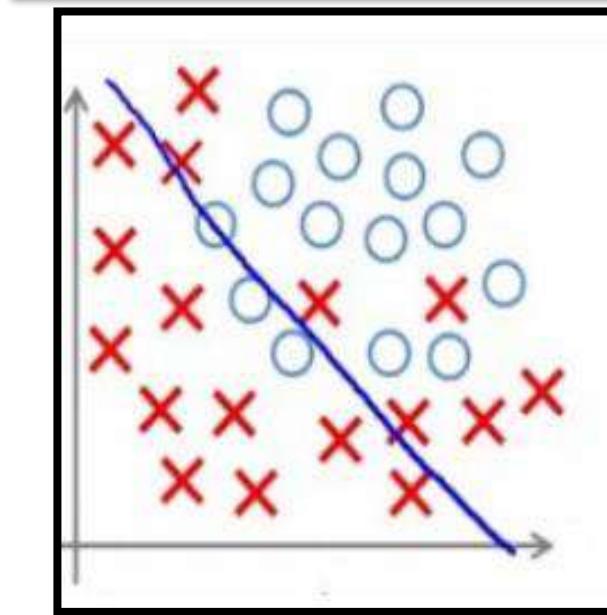
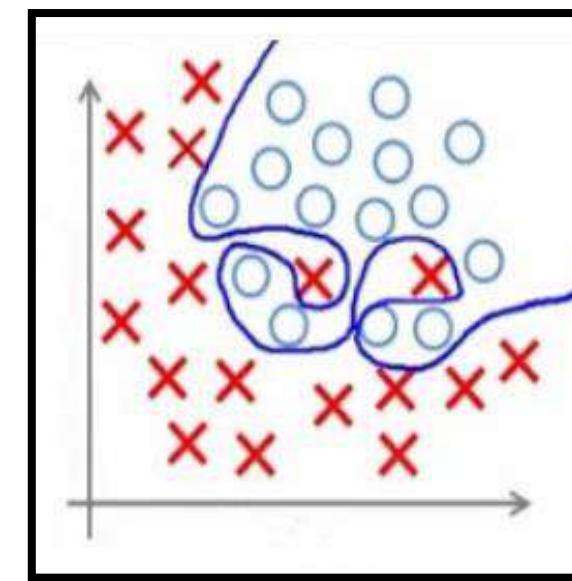
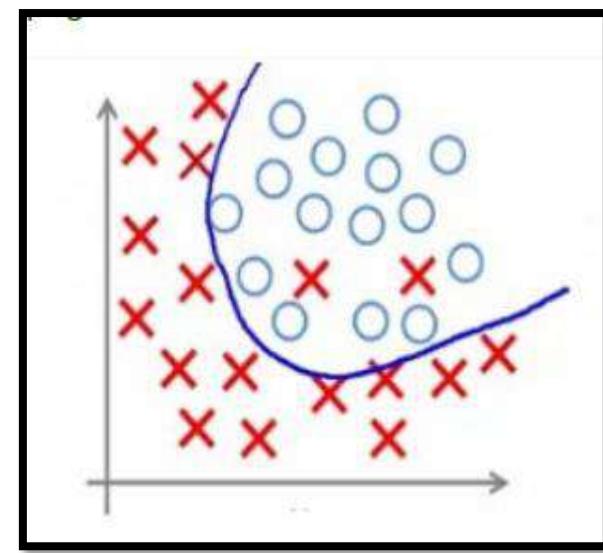
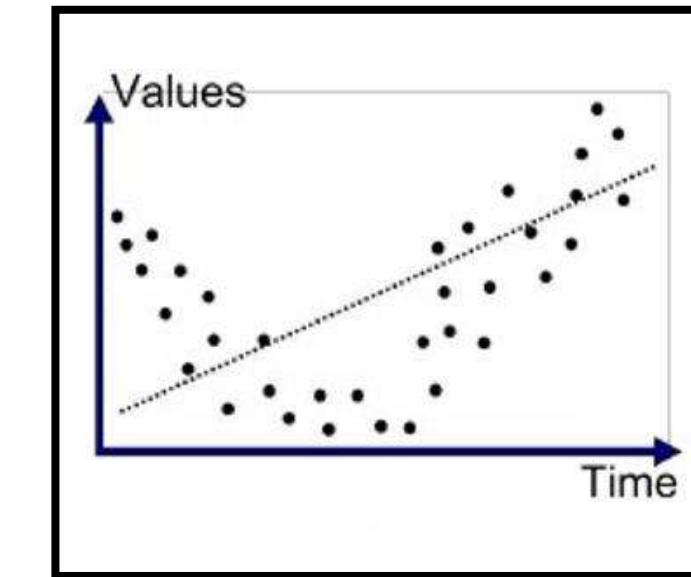
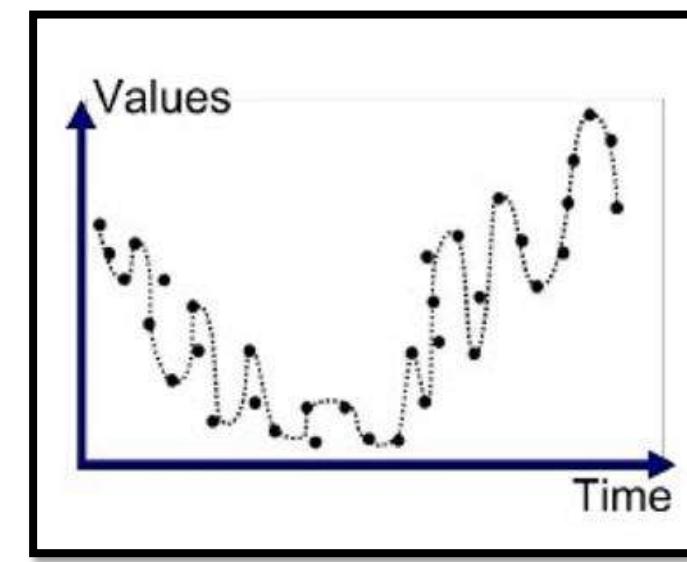
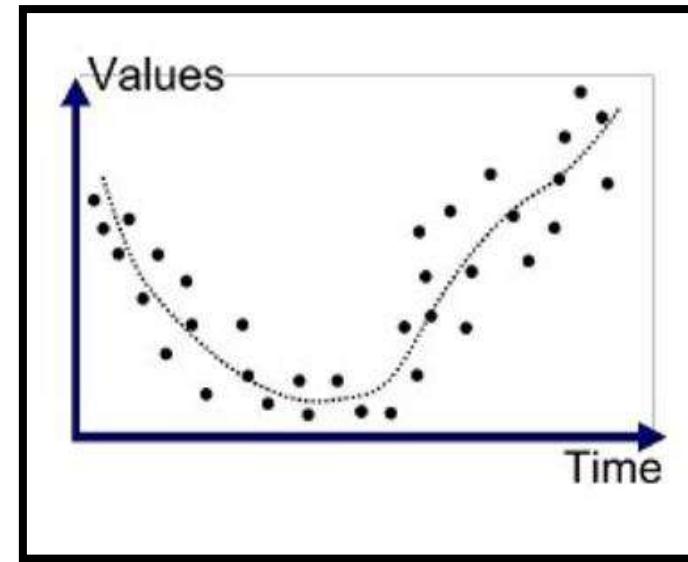
Negative n Positive p

If you like adult comedy cartoons, like South Park, then this is nearly a similar format about the small adventures of three teenage girls at Bromwell High. Keisha, Natella and Latrina have given exploding sweets and behaved like bitches, I think Keisha is a good leader. There are also small stories going on with the teachers of the school. There's the idiotic principal, Mr. Bip, the nervous Maths teacher and many others. The cast is also fantastic, Lenny Henry's Gina Yashere, EastEnders Chrissie Watts, Tracy-Ann Oberman, Smack The Pony's Doon Mackichan, Dead Ringers' Mark Perry and Blunder's Nina Conti. I didn't know this came from Canada, but it is very good. Very good!

< Prev Next >

```
</xml version="1.0" encoding="UTF-8"?>
<art orl="ja" trl="en">
<inf>jawiki-20080607-pages-articles.xml</inf>
<tit>
<j>雪舟</j>
<e type="trans" ver="1">Sesshu</e>
<cmt></cmt>
<e type="trans" ver="2">Sesshu</e>
<cmt>修正なし</cmt>
<e type="check" ver="1">Sesshu</e>
<cmt>修正なし</cmt>
</tit>
<par id="1">
<sen id="1">
<j>雪舟(せっしゅう、1420年(応永27年) - 1506年(永正3年))は号で、15世紀後半町時代に活躍した水墨画家・禅僧で、画聖とも称えられる。</j>
<e type="trans" ver="1">Known as Sesshu (1420 (Oei Year 27) - 1506 (Eisho year 3)), he was an ink painter and Zen monk active in the Muromachi period in the 2nd half of the 15th century.</e>
<cmt></cmt>
<e type="check" ver="2">Known as Sesshu (1420 - 1506, he was an ink painter and Zen monk active in the Muromachi period in the latter half of the 15th century, and was called a 西洋眉のみとする。自然な表現に修正。</cmt>
<e type="trans" ver="2">Known as Sesshu (1420 - 1506), he was an ink painter and Zen monk active in the Muromachi period in the latter half of the 15th century, and was called a 西洋眉のみとする。自然な表現に修正しました。</e>
<cmt>フィードバックに基づき翻訳を修正しました。</cmt>
</sen>
<sen id="2">
<j>日本の水墨画を一変させた。</j>
<e type="trans" ver="1">He changed Japanese ink painting.</e>
<cmt></cmt>
<e type="trans" ver="2">He changed Japanese ink painting.</e>
<cmt>修正なし</cmt>
<e type="check" ver="1">He revolutionized the Japanese ink painting.</e>
<cmt>フィードバックに基づき翻訳を修正しました。</cmt>
```

Summary



Summary

Train (80%)					Validation (10%)	Test (10%)
Fold 1	Fold 2	Fold 3	Fold 4	Fold 5		
Fold 1	Fold 2	Fold 3	Fold 4	Fold 5		
Fold 1	Fold 2	Fold 3	Fold 4	Fold 5		
Fold 1	Fold 2	Fold 3	Fold 4	Fold 5		
Fold 1	Fold 2	Fold 3	Fold 4	Fold 5		
Fold 1	Fold 2	Fold 3	Fold 4	Fold 5		
Fold 1	Fold 2	Fold 3	Fold 4	Fold 5		

Language Models

Salar Mohtaj | DFKI

Language models

- What is language modeling?
- Why language modeling is critical in NLP?
- Statistical language modeling
- Challenges of statistical language modeling
- Evaluation of language models
- Neural language models

Language models

- What is language modeling?
- Why language modeling is critical in NLP?
- Statistical language modeling
- Challenges of statistical language modeling
- Evaluation of language models
- Neural language models

What is language modeling?

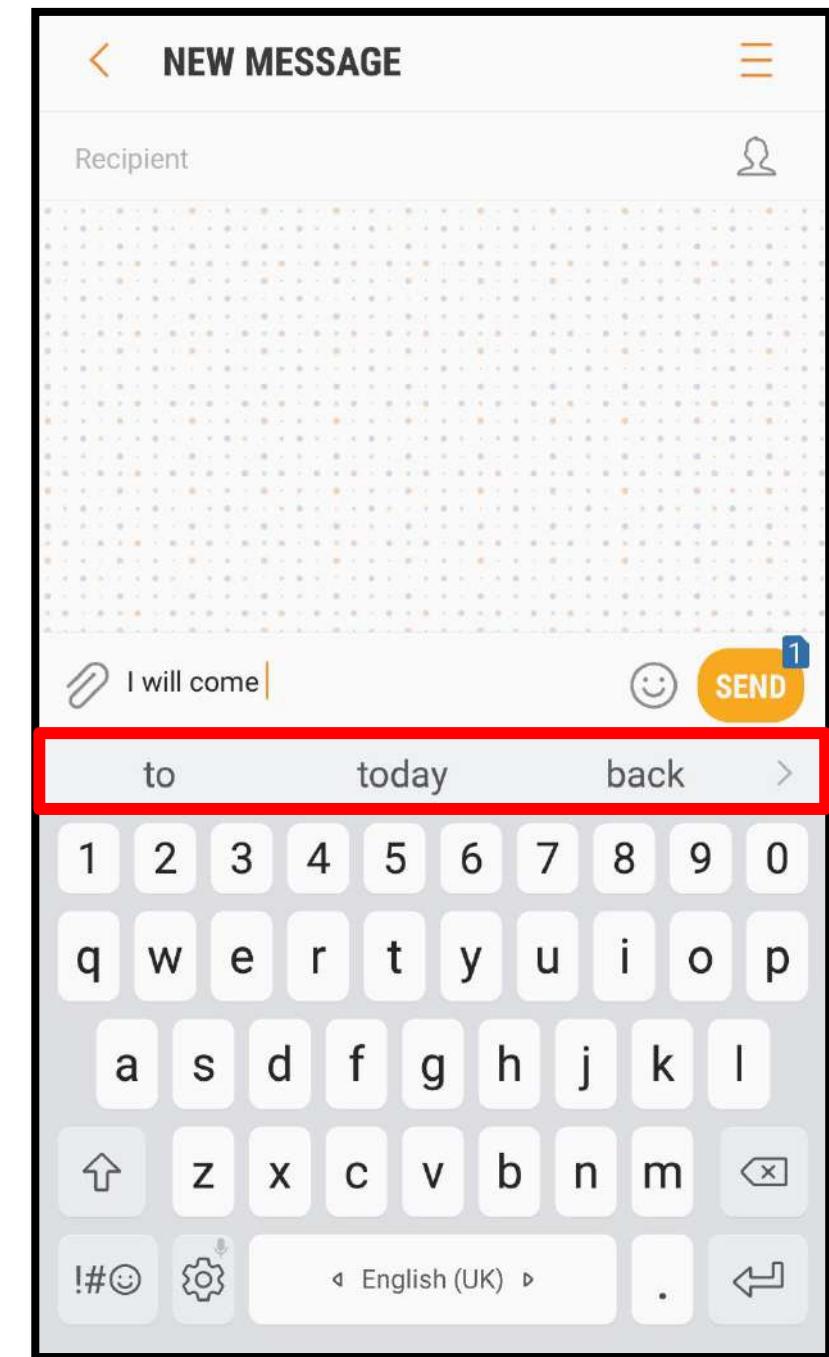
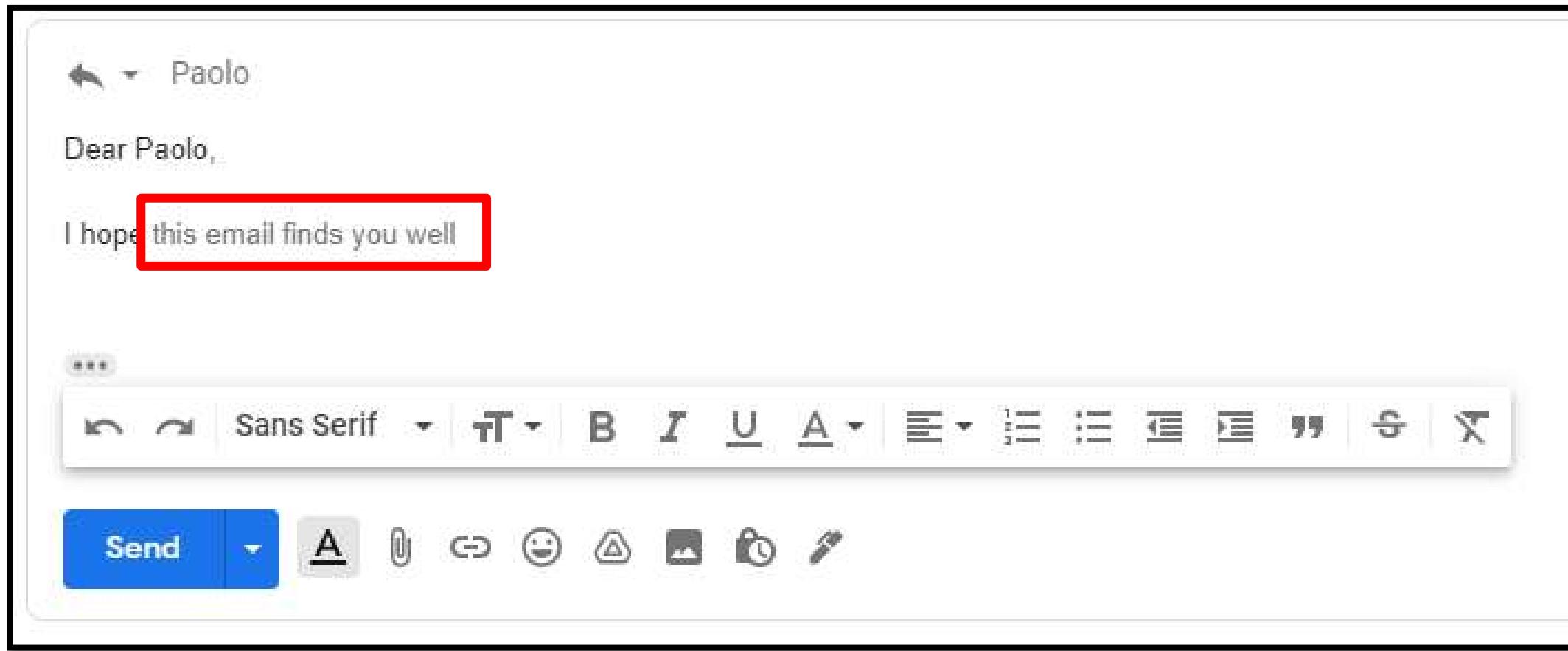
- Models that assign probabilities to sequences of words are called ***language models*** or ***LMs***
 - A language model learns to predict the probability of a sequence of words
 - It is a statistical tool to predict words
- ***Language models*** try to find patterns in the human language
 - They are used to predict the next word in a sentence

Can you please come time?

Can you please come here?

- Language models are a crucial component in the NLP journey

What is language modeling?

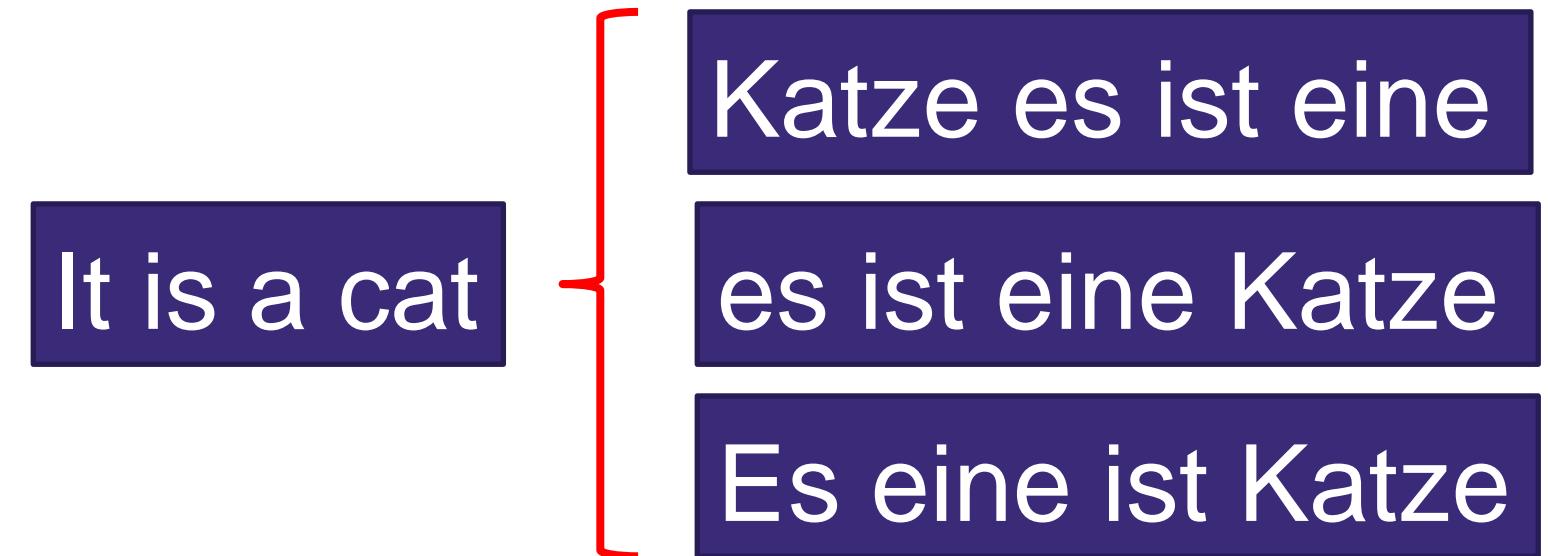


Language models

- What is language modeling?
- Why language modeling is critical in NLP?
- Statistical language modeling
- Challenges of statistical language modeling
- Evaluation of language models
- Neural language models

Why language modeling is critical in NLP?

- The overall performance of different NLP tasks can be improved by *language models*
- Especially in cases where the machine has to generate human language
 - Machine translation
 - Text summarization
 - Image captioning
 - ...



Why language modeling is critical in NLP?

- Text summarization



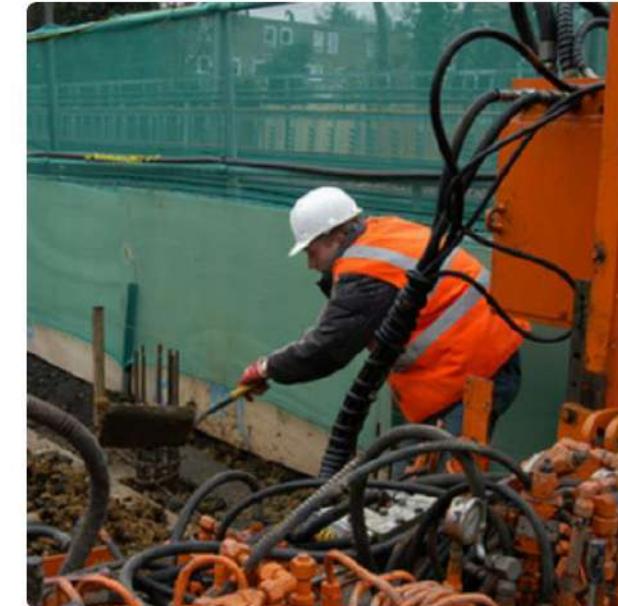
Why language modeling is critical in NLP?

- Image captioning

man in black shirt
is playing guitar



"man in black shirt is playing guitar."



"construction worker in orange safety vest is working on road."



"two young girls are playing with lego toy."

two young girls are
playing with lego toy

Language models

- What is language modeling?
- Why language modeling is critical in NLP?
- **Statistical language modeling**
- Challenges of statistical language modeling
- Evaluation of language models
- Neural language models

Statistical language modeling

- Statistical language modeling is the development of ***probabilistic models*** that are able to predict the next word in the sequence given the words that precede it
- The objective is to compute the probability of a word w given some history h

Can you please come ____?

$$P(w|h)$$

$$P(w|w_1, \dots, w_{n-1})$$

Statistical language modeling

Can you please come ____?

Can you please come time?

$P(\text{time}|\text{can you please come})$

$$= \frac{c(\text{can you please come time})}{c(\text{can you please come})}$$

$$P(A|B) = \frac{P(A \cap B)}{P(B)}$$

Statistical language modeling

Can you please come _____?

Can you please come time?

Can you please come here?

$$P(w|h)$$

$P(\text{time}|\text{can you please come})$

$P(\text{here}|\text{can you please come})$

$$= \frac{c(\text{can you please come time})}{c(\text{can you please come})}$$

$$= \frac{c(\text{can you please come here})}{c(\text{can you please come})}$$

Statistical language modeling

- Joint probability of an entire sequence

Can you please come here?

- $P(\text{Can you please come here})$
- Decompose this probability using the **chain rule of probability**

$$P(w_1, \dots, w_n) = P(w_1)P(w_2|w_1)P(w_3|w_1w_2) \dots P(w_n|w_{1:n-1})$$

- We could estimate the joint probability of an entire sequence of words by multiplying together a number of conditional probabilities.

Statistical language modeling

$$P(w_1, \dots, w_n) = P(w_1)P(w_2|w_1)P(w_3|w_1w_2) \dots P(w_n|w_{1:n-1})$$

Can you please come here?

$$\begin{aligned} P(\text{Can you please come here}) &= P(\text{can})P(\text{you}|\text{can})P(\text{please}|\text{can you}) \\ &\quad P(\text{come}|\text{can you please})P(\text{here}|\text{can you please come}) \end{aligned}$$

- For 100 words ($|v|=100$) and average sentence length of 10:
 - 100^{10} possible sequences

Statistical language modeling

$$P(w_1, \dots, w_n) = P(w_1)P(w_2|w_1)P(w_3|w_1w_2) \dots P(w_n|w_{1:n-1})$$

- **Markov assumption**

- The probability of a word depends only on the k previous words
- Markov models are the class of probabilistic models Markov that assume we can predict the probability of some future unit without looking too far into the past

$$P(w_n|w_1, \dots, w_{n-1}) \approx P(w_n|w_{i-k}, \dots, w_{n-1})$$

Can you please come here?

please come here?

Statistical language modeling

- Instead of computing the probability of a word given its ***entire history***, we can **approximate** the history by just the last few words
 - N-gram model

2-gram language model

come here?

$P(\text{here}|\text{can you please come})$



$P(\text{here}|\text{come})$

Statistical language modeling

- With a bigram language model, we are approximating:

$$P(w_n | w_{1:n-1}) \approx P(w_n | w_{n-1})$$

$$P(\text{here} | \text{can you please come}) \approx P(\text{here} | \text{come})$$

Statistical language modeling

2-gram

come ____?

Asian ____?

the ____?

3-gram

please come ____?

an Asian ____?

before the ____?

Students should register before the ____?

Statistical language modeling

- With a bigram language model, we are approximating:

$$P(w_n | w_{1:n-1}) \approx P(w_n | w_{n-1})$$

$$P(w_n | w_{n-1}) = \frac{c(w_{n-1} w_n)}{c(w_{n-1})}$$

$$P(\text{here} | \text{can you please come}) \approx P(\text{here} | \text{come})$$

$$P(\text{here} | \text{come}) = \frac{c(\text{come here})}{c(\text{come})}$$

Statistical language modeling

Can you please come here?



Counting number of times that the sequence is reapeaded in the corpus



$$P(\text{Can you please come here}) = P(\text{can})P(\text{you}|\text{can})P(\text{please}|\text{can you}) \\ P(\text{come}|\text{can you please})P(\text{here}|\text{can you please come})$$

$$P(w_n | w_{1:n-1}) \approx P(w_n | w_{n-1})$$

$$P(\text{Can you please come here}) = P(\text{can})P(\text{you}|\text{can})P(\text{please}|\text{you})P(\text{come}|\text{please})P(\text{here}|\text{come})$$

$$P(w_n | w_{n-1}) = \frac{c(w_{n-1} w_n)}{c(w_{n-1})}$$

$$P(\text{here}|\text{come}) = \frac{c(\text{come here})}{c(\text{come})}$$

Statistical language modeling

2-gram

- $D_1 = <\text{s}> \text{the} \text{book} \text{is} \text{written} </\text{s}>$
- $D_2 = <\text{s}> \text{the} \text{paint} \text{is} \text{drawn} </\text{s}>$
- $D_3 = <\text{s}> \text{the} \text{texture} \text{of} \text{the} \text{paint} \text{is} \text{white} </\text{s}>$

$$P(\text{the} | <\text{s}>) = 3/3$$

$$P(\text{book} | \text{the}) = \frac{c(\text{the book})}{c(\text{the})} = 1/4 \quad P(\text{is} | \text{book}) = 1/1$$

$$P(\text{written} | \text{is}) = 1/3$$

$$P(</\text{s}> | \text{written}) = 1/1$$

$$P(\text{paint} | \text{the}) = \frac{c(\text{the paint})}{c(\text{the})} = 2/4$$

$$P(\text{is} | \text{paint}) = 2/2$$

$$P(\text{drawn} | \text{is}) = 1/3$$

$$P(</\text{s}> | \text{drawn}) = 1/1$$

$$P(\text{texture} | \text{the}) = 1/4$$

$$P(\text{of} | \text{texture}) = 1/1$$

$$P(\text{the} | \text{of}) = 1/1$$

$$P(\text{white} | \text{is}) = 1/3$$

$$P(</\text{s}> | \text{white}) = 1/1$$

Statistical language modeling

2-gram

- $D_1 = < s > \boxed{\text{the book}} \text{is written } < /s >$
- $D_2 = < s > \text{ the paint is drawn } < /s >$
- $D_3 = < s > \text{ the texture of the paint is white } < /s >$

$$P(\text{the} | < s >) = 3/3$$

$$P(\text{book} | \text{the}) = \frac{c(\text{the book})}{c(\text{the})} = 1/4$$

$$P(\text{is} | \text{book}) = 1/1$$

$$P(\text{written} | \text{is}) = 1/3$$

$$P(< /s > | \text{written}) = 1/1$$

$$P(\text{paint} | \text{the}) = \frac{c(\text{the paint})}{c(\text{the})} = 2/4$$

$$P(\text{is} | \text{paint}) = 2/2$$

$$P(\text{drawn} | \text{is}) = 1/3$$

$$P(< /s > | \text{drawn}) = 1/1$$

$$P(\text{texture} | \text{the}) = 1/4$$

$$P(\text{of} | \text{texture}) = 1/1$$

$$P(\text{the} | \text{of}) = 1/1$$

$$P(\text{white} | \text{is}) = 1/3$$

$$P(< /s > | \text{white}) = 1/1$$

Statistical language modeling

2-gram

- $D_1 = < s > \text{the book is written } < /s >$
- $D_2 = < s > \text{the paint is drawn } < /s >$
- $D_3 = < s > \text{the texture of the paint is white } < /s >$

$$P(\text{the} | < s >) = 3/3$$

$$P(\text{book} | \text{the}) = \frac{c(\text{the book})}{c(\text{the})} = 1/4$$

$$P(\text{is} | \text{book}) = 1/1$$

$$P(\text{written} | \text{is}) = 1/3$$

$$P(< /s > | \text{written}) = 1/1$$

$$P(\text{paint} | \text{the}) = \frac{c(\text{the paint})}{c(\text{the})} = 2/4$$

$$P(\text{is} | \text{paint}) = 2/2$$

$$P(\text{drawn} | \text{is}) = 1/3$$

$$P(< /s > | \text{drawn}) = 1/1$$

$$P(\text{texture} | \text{the}) = 1/4$$

$$P(\text{of} | \text{texture}) = 1/1$$

$$P(\text{the} | \text{of}) = 1/1$$

$$P(\text{white} | \text{is}) = 1/3$$

$$P(< /s > | \text{white}) = 1/1$$

Statistical language modeling

$$P(\text{the} | < s >) = 3/3$$

$$P(\text{book} | \text{the}) = \frac{c(\text{the book})}{c(\text{the})} = 1/4$$

$$P(\text{is} | \text{book}) = 1/1$$

$$P(\text{written} | \text{is}) = 1/3$$

$$P(</s > | \text{written}) = 1/1$$

$$P(\text{paint} | \text{the}) = \frac{c(\text{the paint})}{c(\text{the})} = 2/4$$

$$P(\text{is} | \text{paint}) = 2/2$$

$$P(\text{drawn} | \text{is}) = 1/3$$

$$P(</s > | \text{drawn}) = 1/1$$

$$P(\text{texture} | \text{the}) = 1/4$$

$$P(\text{of} | \text{texture}) = 1/1$$

$$P(\text{the} | \text{of}) = 1/1$$

$$P(\text{white} | \text{is}) = 1/3$$

$$P(</s > | \text{white}) = 1/1$$

- Predict the next word in text:

My friend is written.

Statistical language modeling

$$P(\text{the} | \langle s \rangle) = 3/3$$

$$P(\text{book} | \text{the}) = \frac{c(\text{the book})}{c(\text{the})} = 1/4$$

$$P(\text{is} | \text{book}) = 1/1$$

$$P(\text{written} | \text{is}) = 1/3$$

$$P(\langle /s \rangle | \text{written}) = 1/1$$

$$P(\text{paint} | \text{the}) = \frac{c(\text{the paint})}{c(\text{the})} = 2/4$$

$$P(\text{is} | \text{paint}) = 2/2$$

$$P(\text{drawn} | \text{is}) = 1/3$$

$$P(\langle /s \rangle | \text{drawn}) = 1/1$$

$$P(\text{texture} | \text{the}) = 1/4$$

$$P(\text{of} | \text{texture}) = 1/1$$

$$P(\text{the} | \text{of}) = 1/1$$

$$P(\text{white} | \text{is}) = 1/3$$

$$P(\langle /s \rangle | \text{white}) = 1/1$$

- Compute the probability of sentences:

D = $\langle s \rangle$ the book is white $\langle /s \rangle$

$$P(D) = P(\text{the} | \langle s \rangle)P(\text{book} | \text{the}) \\ P(\text{is} | \text{book})P(\text{white} | \text{is}) P(\langle /s \rangle | \text{is})$$

$$P(D) = 3/3 \times 1/4 \times 1/1 \times 1/3 \times 1/1 = 0.083$$

Statistical language modeling

$$P(\text{the} | <\text{s}>) = 3/3$$

$$P(\text{book}|\text{the}) = \frac{c(\text{the book})}{c(\text{the})} = 1/4$$

$$P(\text{is}|\text{book}) = 1/1$$

$$P(\text{written}|\text{is}) = 1/3$$

$$P(</\text{s}> | \text{written}) = 1/1$$

$$P(\text{paint}|\text{the}) = \frac{c(\text{the paint})}{c(\text{the})} = 2/4$$

$$P(\text{is}|\text{paint}) = 2/2$$

$$P(\text{drawn}|\text{is}) = 1/3$$

$$P(</\text{s}> | \text{drawn}) = 1/1$$

$$P(\text{texture}|\text{the}) = 1/4$$

$$P(\text{of}|\text{texture}) = 1/1$$

$$P(\text{the}|\text{of}) = 1/1$$

$$P(\text{white}|\text{is}) = 1/3$$

$$P(</\text{s}> | \text{white}) = 1/1$$

- Generate a sample text (the Shannon visualization method):
 - Choose a random bigram (<s>, w) according to its probability
 - Now choose a random bigram (w, x) according to its probability
 - And so on until we choose </s>

<S>

Statistical language modeling

$$P(\text{the} | \langle s \rangle) = 3/3$$

$$P(\text{written} | \text{is}) = 1/3$$

$$P(\text{is} | \text{paint}) = 2/2$$

$$P(\text{texture} | \text{the}) = 1/4$$

$$P(\text{white} | \text{is}) = 1/3$$

$$P(\text{book} | \text{the}) = \frac{c(\text{the book})}{c(\text{the})} = 1/4$$

$$P(\langle /s \rangle | \text{written}) = 1/1$$

$$P(\text{drawn} | \text{is}) = 1/3$$

$$P(\text{of} | \text{texture}) = 1/1$$

$$P(\langle /s \rangle | \text{white}) = 1/1$$

$$P(\text{is} | \text{book}) = 1/1$$

$$P(\text{paint} | \text{the}) = \frac{c(\text{the paint})}{c(\text{the})} = 2/4$$

$$P(\langle /s \rangle | \text{drawn}) = 1/1$$

$$P(\text{the} | \text{of}) = 1/1$$

- Generate a sample text (the Shannon visualization method):
 - Choose a random bigram ($\langle s \rangle$, w) according to its probability
 - Now choose a random bigram (w, x) according to its probability
 - And so on until we choose $\langle /s \rangle$

$\langle s \rangle$

the

Statistical language modeling

$$P(\text{the} | < \text{s} >) = 3/3$$

$$P(\text{book} | \text{the}) = \frac{c(\text{the book})}{c(\text{the})} = 1/4$$

$$P(\text{is} | \text{book}) = 1/1$$

$$P(\text{written} | \text{is}) = 1/3$$

$$P(</\text{s}> | \text{written}) = 1/1$$

$$P(\text{paint} | \text{the}) = \frac{c(\text{the paint})}{c(\text{the})} = 2/4$$

$$P(\text{is} | \text{paint}) = 2/2$$

$$P(\text{drawn} | \text{is}) = 1/3$$

$$P(</\text{s}> | \text{drawn}) = 1/1$$

$$P(\text{texture} | \text{the}) = 1/4$$

$$P(\text{of} | \text{texture}) = 1/1$$

$$P(\text{the} | \text{of}) = 1/1$$

$$P(\text{white} | \text{is}) = 1/3$$

$$P(</\text{s}> | \text{white}) = 1/1$$

- Generate a sample text (the Shannon visualization method):
 - Choose a random bigram (<s>, w) according to its probability
 - Now choose a random bigram (w, x) according to its probability
 - And so on until we choose </s>

<s>

the

paint

Statistical language modeling

$$P(\text{the} | \langle s \rangle) = 3/3$$

$$P(\text{book} | \text{the}) = \frac{c(\text{the book})}{c(\text{the})} = 1/4$$

$$P(\text{is} | \text{book}) = 1/1$$

$$P(\text{written} | \text{is}) = 1/3$$

$$P(\langle /s \rangle | \text{written}) = 1/1$$

$$P(\text{paint} | \text{the}) = \frac{c(\text{the paint})}{c(\text{the})} = 2/4$$

$$P(\text{is} | \text{paint}) = 2/2$$

$$P(\text{drawn} | \text{is}) = 1/3$$

$$P(\langle /s \rangle | \text{drawn}) = 1/1$$

$$P(\text{texture} | \text{the}) = 1/4$$

$$P(\text{of} | \text{texture}) = 1/1$$

$$P(\text{the} | \text{of}) = 1/1$$

$$P(\text{white} | \text{is}) = 1/3$$

$$P(\langle /s \rangle | \text{white}) = 1/1$$

- Generate a sample text (the Shannon visualization method):
 - Choose a random bigram ($\langle s \rangle$, w) according to its probability
 - Now choose a random bigram (w, x) according to its probability
 - And so on until we choose $\langle /s \rangle$

$\langle s \rangle$ the paint is written $\langle /s \rangle$

Language models

- What is language modeling?
- Why language modeling is critical in NLP?
- Statistical language modeling
- Challenges of statistical language modeling
- Evaluation of language models
- Neural language models

Challenges of statistical language modeling

- Out of Vocabulary (OOV) words
- Zero probabilities

Out of Vocabulary (OOV) words

- When some words/terms in test set have never seen before
- Two common solutions
 - Make the vocabulary as closed (no new words in test set)
 1. Choose a vocabulary (word list) that is fixed in advance
 2. Convert the other tokens into <UNK>
 3. Estimate the probabilities for <UNK>

Can you please come here?

Can you please <unk> here?

Vocabulary (word list)
can
you
please
here

Out of Vocabulary (OOV) words

- When some words/terms in test set have never seen before
- Two common solutions
 - Make the vocabulary as closed (no new words in test set)
 1. Choose a vocabulary (word list) that is fixed in advance
 2. Convert the other tokens into <UNK>
 3. Estimate the probabilities for <UNK>
 - Replacing words in the training data by <UNK> based on their frequency

Zero probabilities

- A word appear after a word they never appeared after in training
 - It's not unknown words

Can you please come here?

$$P(you|come) = 0$$

- Laplace smoothing
 - To add one to all the bigram counts, before we normalize them into probabilities

Statistical language modeling

2-gram

- $D_1 = < \text{s} > \text{ the book is written } < / \text{s} >$
- $D_2 = < \text{s} > \text{ the paint is drawn } < / \text{s} >$
- $D_3 = < \text{s} > \text{ the texture of the paint is white } < / \text{s} >$

$$P(\text{the} | < \text{s} >) = 3/3$$

$$P(\text{book} | \text{the}) = \frac{c(\text{the book})}{c(\text{the})} = 1/4$$

$$P(\text{is} | \text{book}) = 1/1$$

$$P(\text{written} | \text{is}) = 1/3 \quad 2/4$$

$$P(< / \text{s} > | \text{written}) = 1/1$$

$$P(\text{paint} | \text{the}) = \frac{c(\text{the paint})}{c(\text{the})} = 2/4$$

$$P(\text{is} | \text{paint}) = 2/2$$

$$P(\text{drawn} | \text{is}) = 1/3$$

$$P(< / \text{s} > | \text{drawn}) = 1/1$$

$$P(\text{texture} | \text{the}) = 1/4 \quad 2/5$$

$$P(\text{of} | \text{texture}) = 1/1$$

$$P(\text{the} | \text{of}) = 1/1$$

$$P(\text{white} | \text{is}) = 1/3$$

$$P(< / \text{s} > | \text{white}) = 1/1$$

$$P(\text{paint} | \text{texture}) = 0/1 \quad 1/2$$

Language models

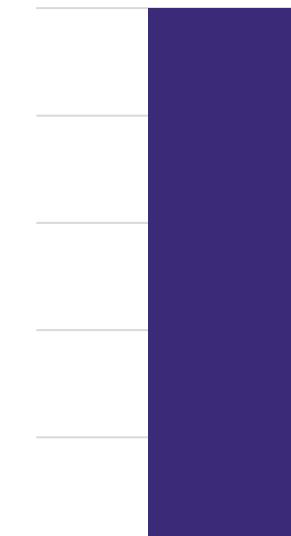
- What is language modeling?
- Why language modeling is critical in NLP?
- Statistical language modeling
- Challenges of statistical language modeling
- **Evaluation of language models**
- Neural language models

Evaluation of language models

- There are two main approaches for evaluating language models:
 - Extrinsic evaluation
 - Intrinsic evaluation

Language model A

Language model B



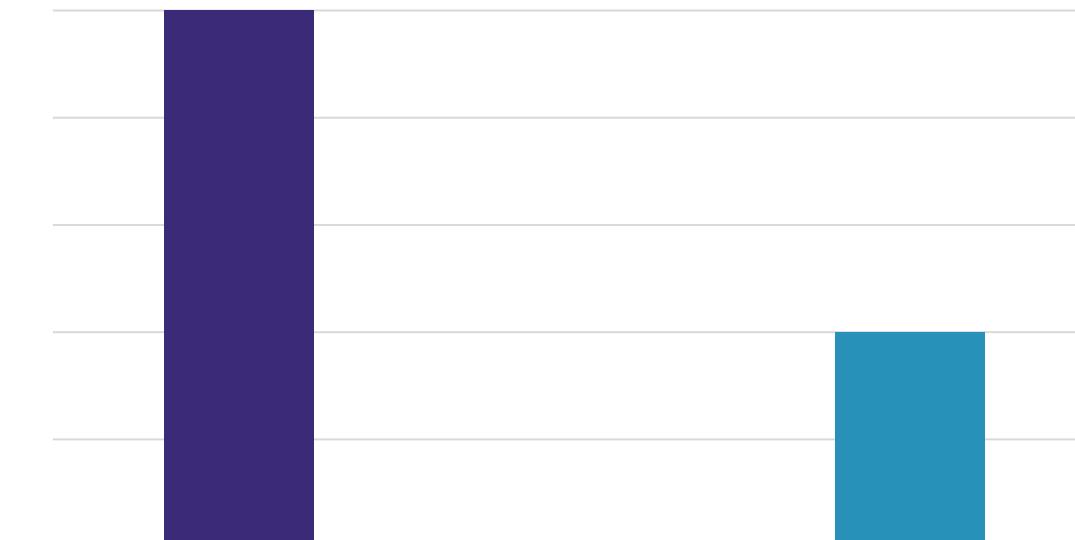
Evaluation of language models

- Intrinsic evaluation
 - Whichever model assigns a higher probability to the test set (meaning it more accurately predicts the test set) is a better model
 - Given two probabilistic models, the better model is the one that has a tighter fit to the test data or that better predicts the details of the test data, and hence will assign a higher probability to the test data

Language model A



Language model B



Evaluation of language models

- Perplexity
 - Measurement of how well a probability distribution or probability model predicts a sample
 - A ***low perplexity*** indicates the probability distribution is good at predicting the sample

$$\text{Perplexity}(W) = \sqrt{\prod_{i=1}^N \frac{1}{P(w_i | w_1 \dots w_{i-1})}}$$

$$W = w_1 w_2 \dots w_N$$

Evaluation of language models

$$\text{Perplexity}(W) = \sqrt{\prod_{i=1}^N \frac{1}{P(w_i | w_1 \dots w_{i-1})}}$$

$W = w_1 w_2 \dots w_N$

$W = \text{Can you please come here?}$

Language model A

$P(\text{you}|\text{can}) = 0.5$

$P(\text{please}|\text{you}) = 0.7$

Language model B

$P(\text{you}|\text{can}) = 0.02$

$P(\text{please}|\text{you}) = 0.26$

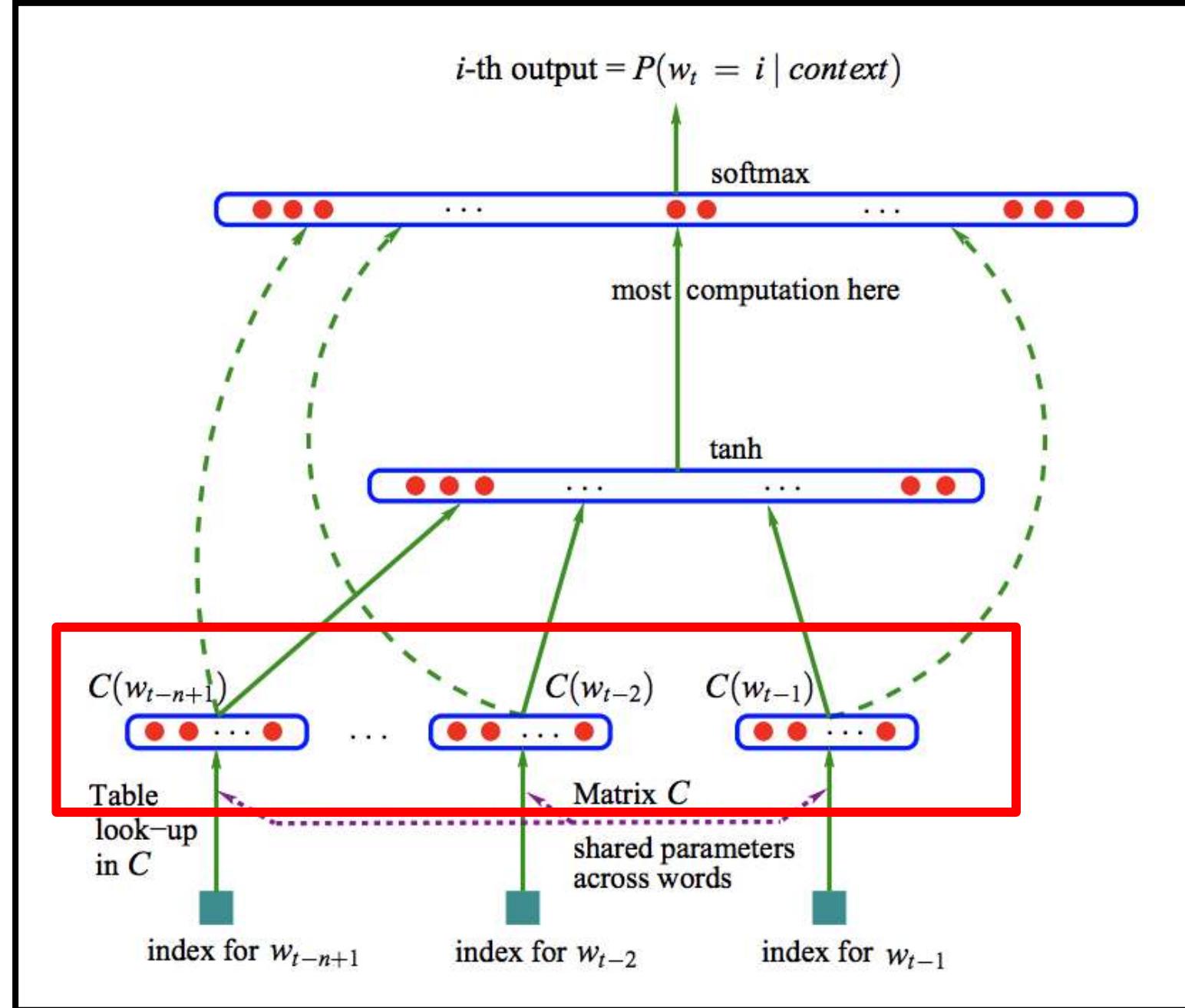
Language models

- What is language modeling?
- Why language modeling is critical in NLP?
- Statistical language modeling
- Challenges of statistical language modeling
- Evaluation of language models
- Neural language models

Neural language models

- Language models based on neural networks
- Neural language models advantages:
 - Can handle much ***longer histories***
 - Can **generalize** over contexts of similar words
 - Has much ***higher predictive accuracy*** than an n-gram language model
 - Don't need ***smoothing***
- Neural language models are ***too slower*** than traditional language models to train

Neural language models



Can you please come here?

you	-	come	was	here	she	can	time	just
0.17	0.04	0.06	0.10	0.09	0.26	0.08	0.05	0.11
1.7	0.3	0.7	1.1	1	2.1	0.98	0.43	1.21

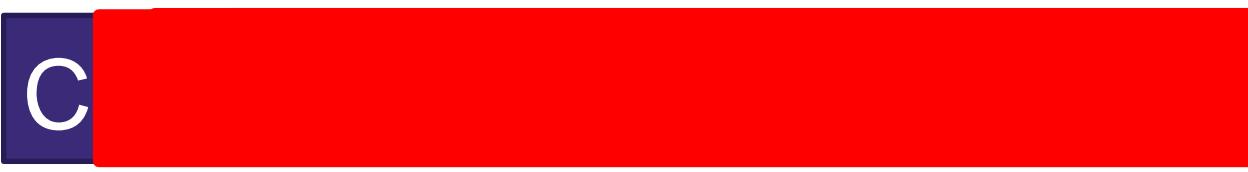
$$\sigma(\vec{z})_i = \frac{e^{z_i}}{\sum_{j=1}^k e^{z_j}}$$

can			you			please		
0.1	-0.2	-0.4	0.3	0.1	0.9	-0.7	0.2	-0.7

Bengio, Y., Ducharme, R., Vincent, P., & Janvin, C. (2003). A neural probabilistic language model. *The journal of machine learning research*, 3, 1137-1155.

Neural language models

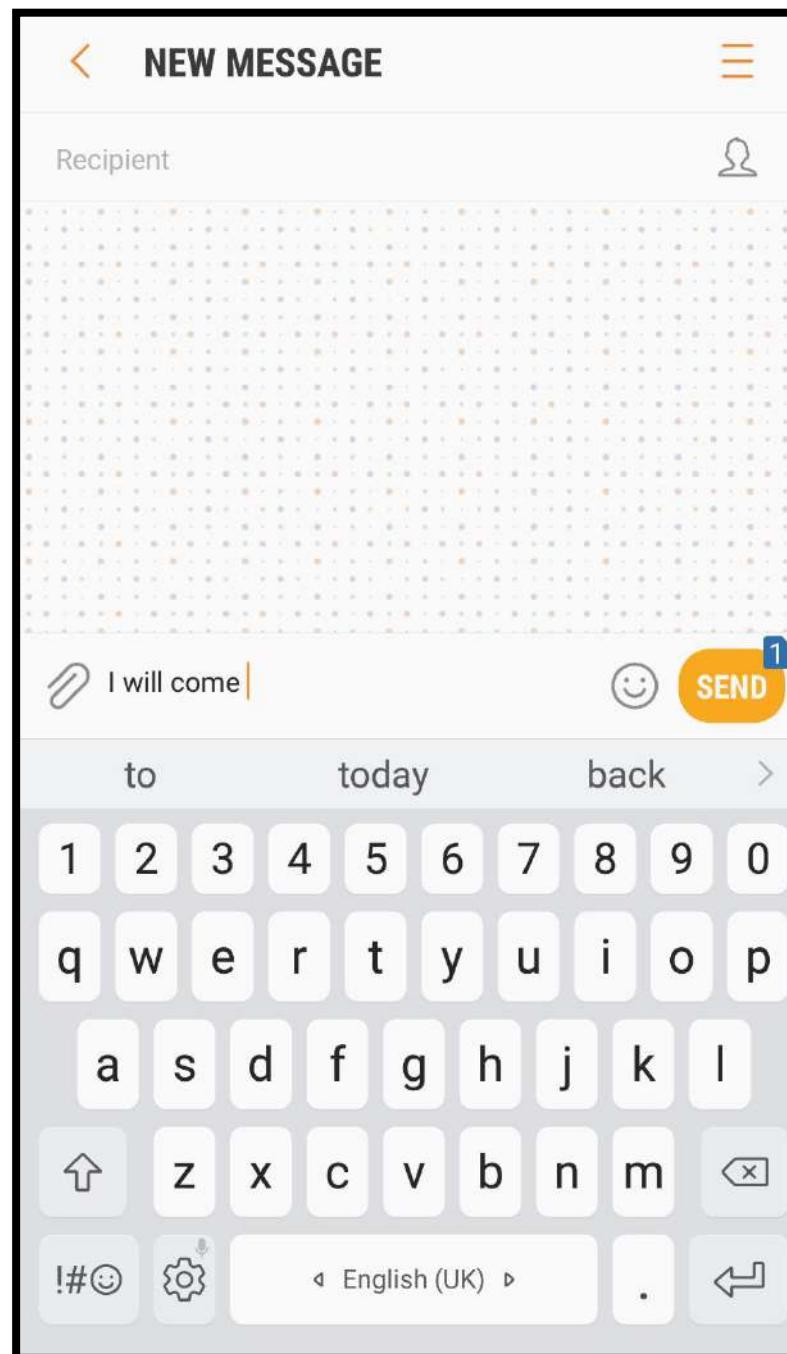
- Character based language models



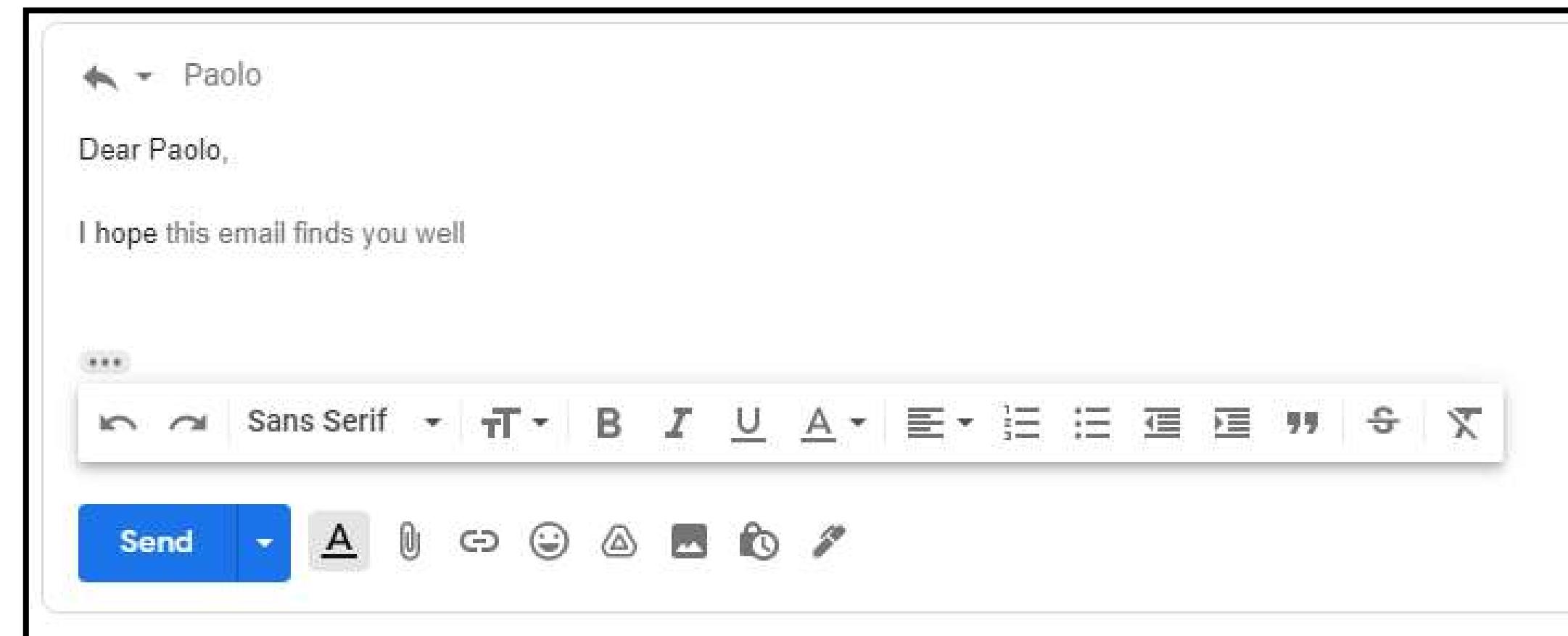
C

- Advantages
 - The model is smaller (97 English-language characters in common, includes all punctuation marks)
 - Flexibility in handling any words
- Disadvantages
 - Lack of semantic content of the input (characters are meaningless)
 - Longer sequences increase computational expense

Summary



Can you please come here?



Summary

2-gram language model

come here?

$$P(\text{here}|\text{can you please come}) \longrightarrow P(\text{here}|\text{come})$$

Can you please <unk> here?

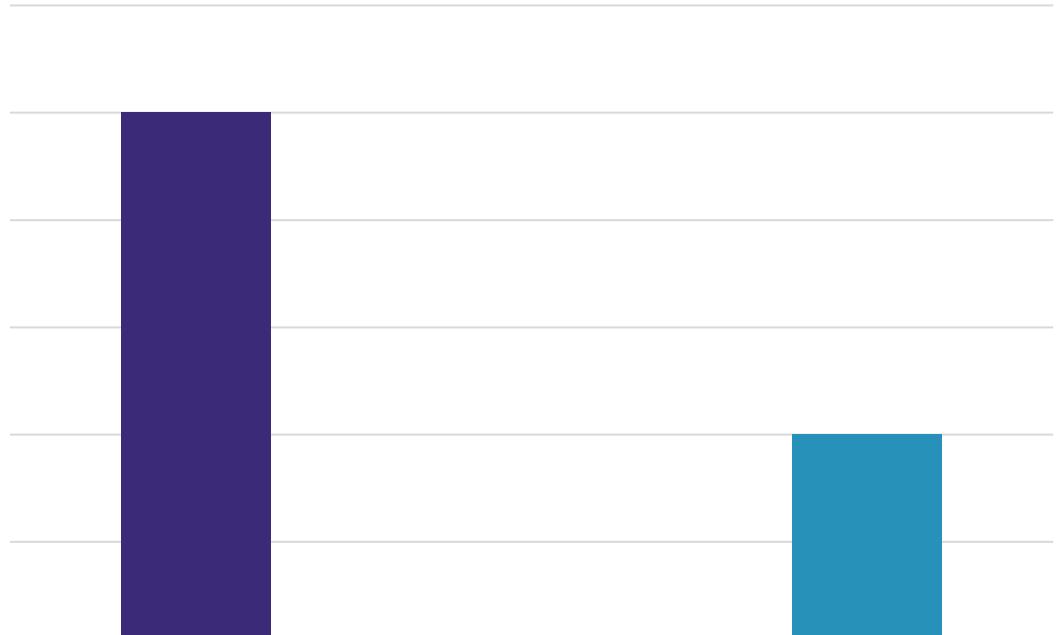
Can you please come here?

$$P(\text{you}|\text{come}) = 0$$

Summary

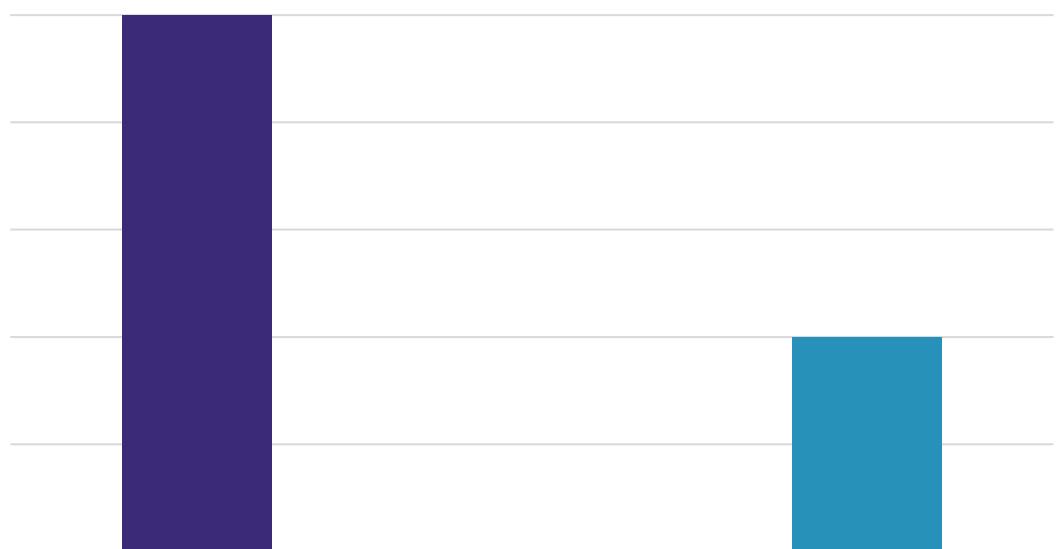
Language model A

Language model B

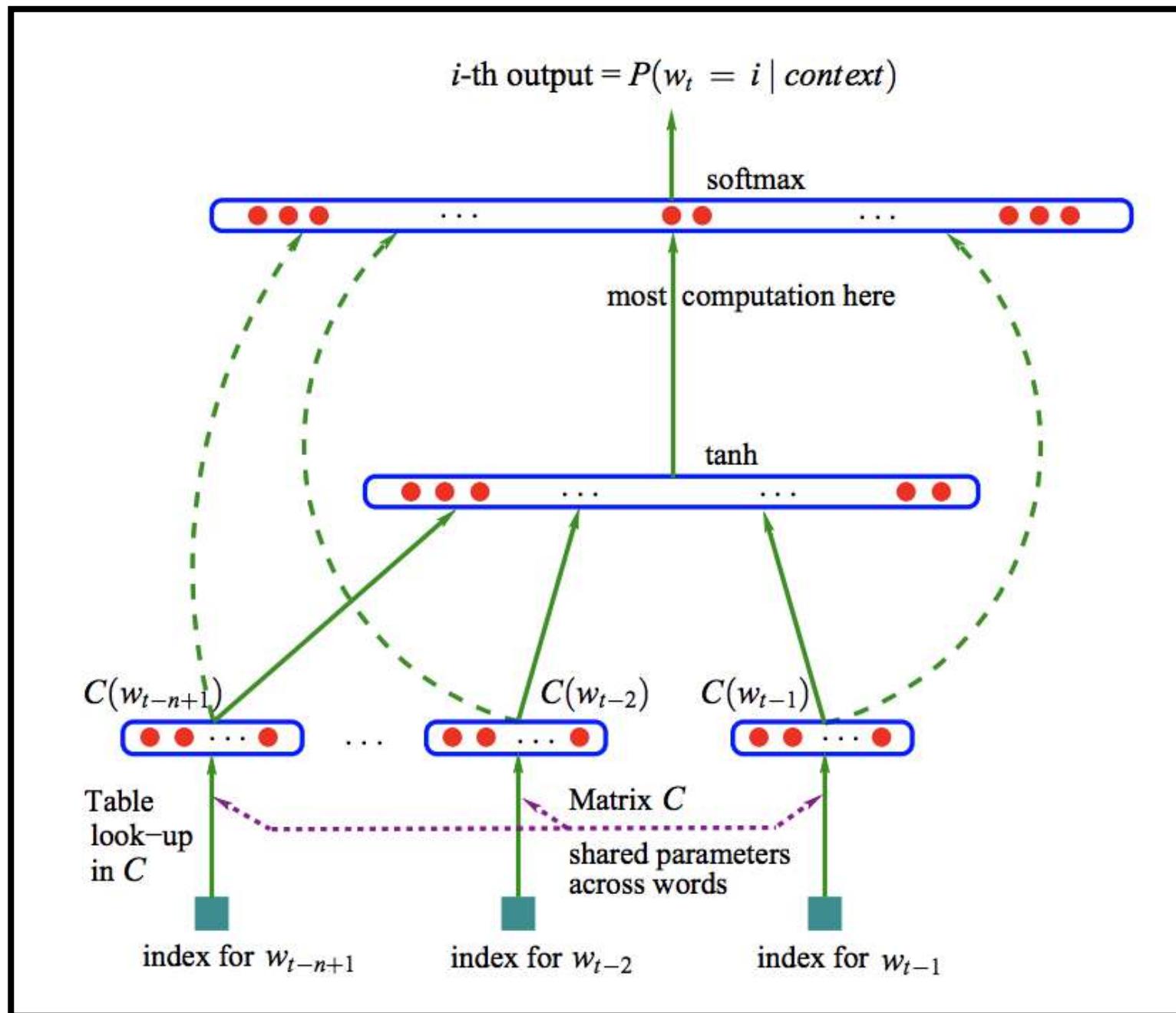


Language model A

Language model B



Summary



you	I	come	please	was	here	she	can	time	just
0.17	0.04	0.06	0.10	0.09	0.26	0.08	0.05	0.04	0.11

can		
0.1	-0.2	-0.4

you		
0.3	0.1	0.9

please		
-0.7	0.2	-0.7

Semantic Text Similarity

Salar Mohtaj | DFKI

Semantic textual similarity

- What is semantic similarity?
- Semantic similarity in word level
- Semantic similarity in sentence level
- Semantic textual similarity in Python

Semantic textual similarity

- What is semantic similarity?
- Semantic similarity in word level
- Semantic similarity in sentence level
- Semantic textual similarity in Python

Semantic textual similarity

- Semantic textual similarity (STS) deals with determining how similar two pieces of texts are
- It's about measuring semantic similarity between words/terms, sentences, paragraphs or documents
- Semantic similarity methods usually give a ***ranking*** or ***percentage*** of similarity between texts, rather than a binary decision as similar or not similar
- Related tasks are ***paraphrase identification***, or ***duplicate identification***

Semantic textual similarity

- The techniques like Bag of Words (BoW) and TF-IDF are used to represent text, as real value vectors
- However, these techniques did not attribute to the fact that words have different meanings and different words can be used to represent a similar concept

John and David studied Math and Science.



John studied Math and David studied Science.

Mary is allergic to dairy products. = Mary is lactose intolerant.

Semantic textual similarity

- Why does it matter?

Google how thin is a dollar bill X |

Alle Shopping Bilder News Videos Mehr Einstellungen Suchfilter

Ungefähr 21.600.000 Ergebnisse (0,88 Sekunden)

1. U.S. paper currency such as a \$1 bill measures 2.6 **inches** wide by 6.14 **inches** long with a thickness of .0043 **inches**.

<https://www.ehd.org> › ... › Technology Articles
Grasping Large Numbers

Informationen zu hervorgehobenen Snippets Feedback geben

Ähnliche Fragen

How thick is a 1 dollar bill? ▾

How thick is a \$50 bill? ▾

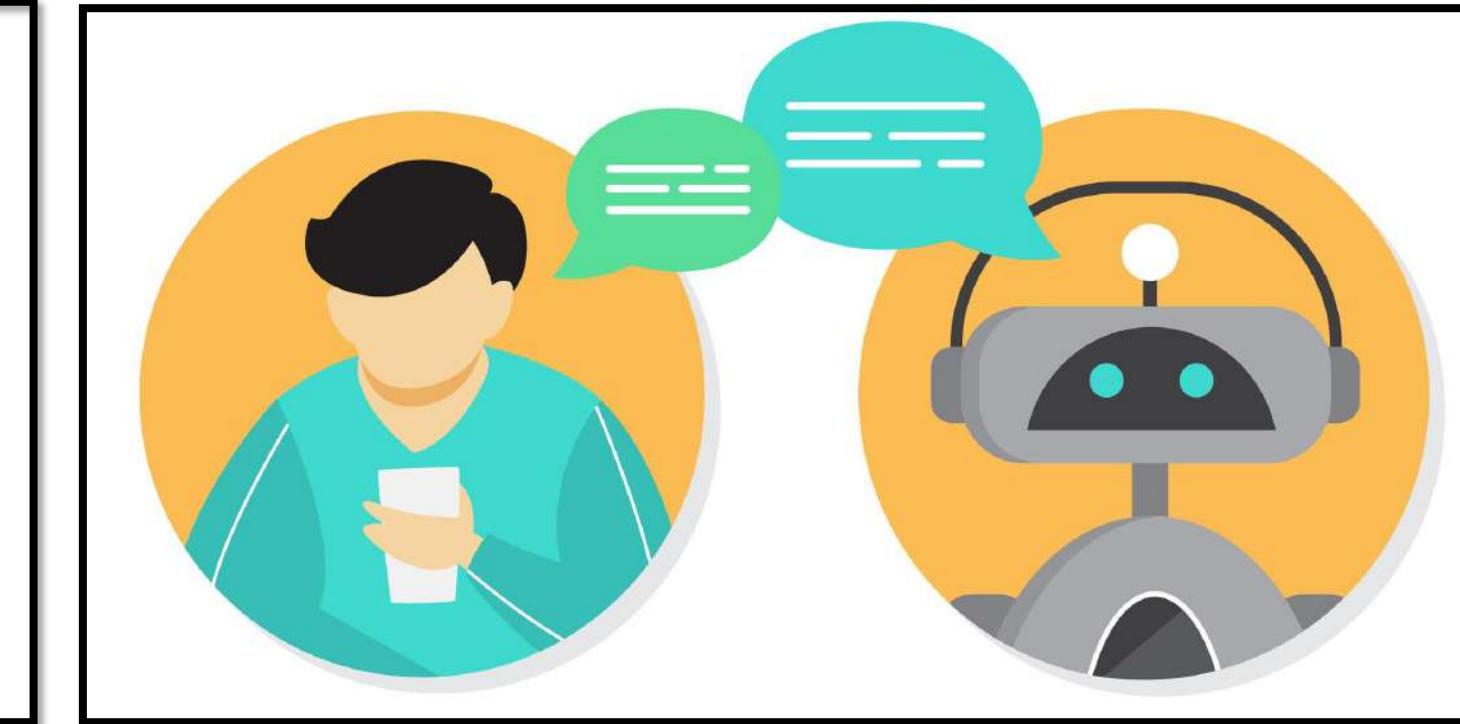
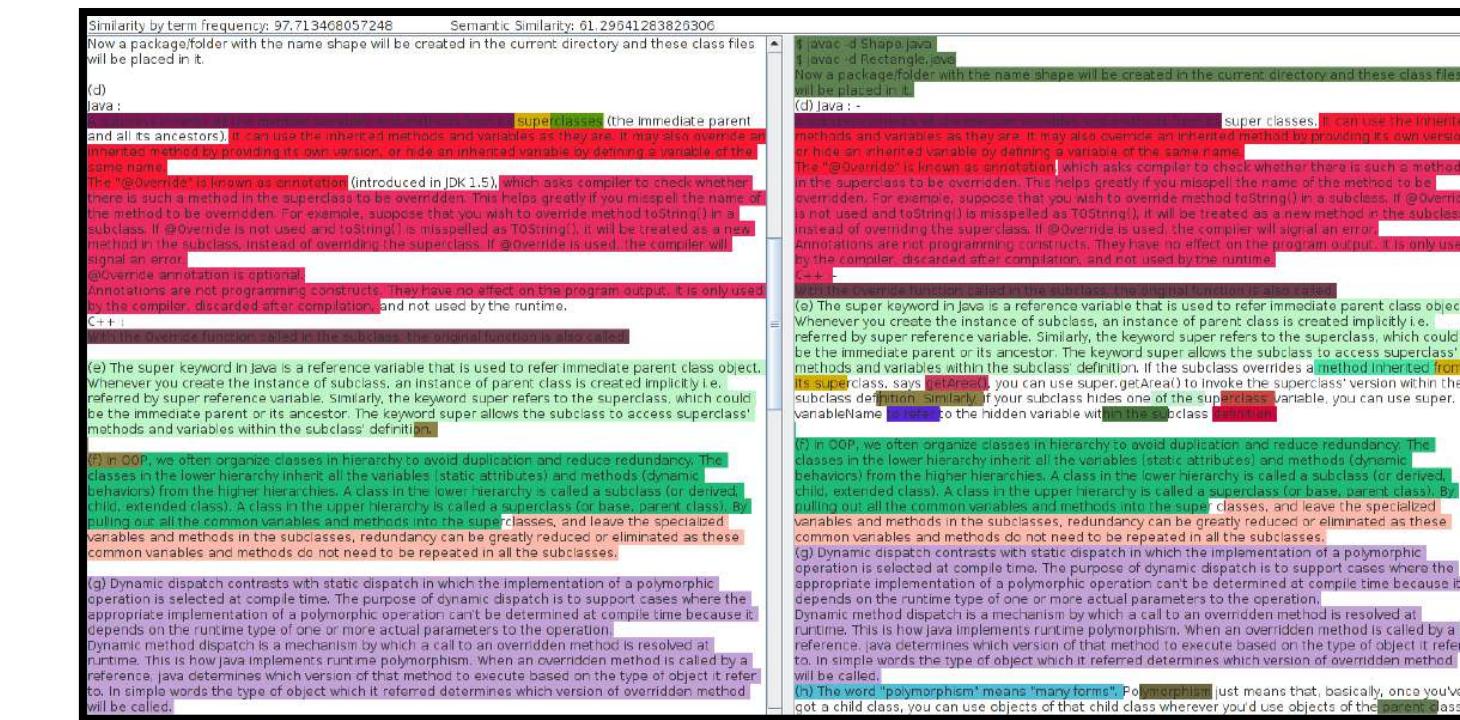
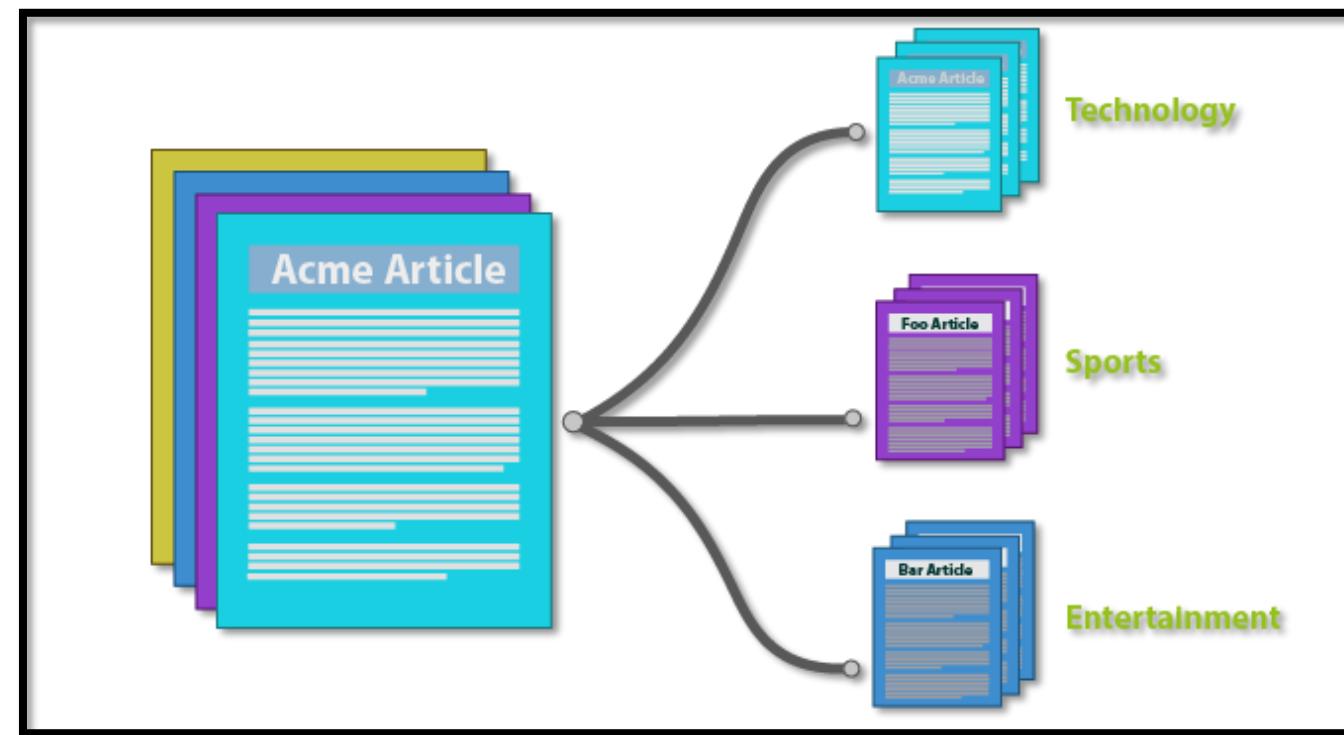
Can a dollar bill shrink? ▾

Is a dollar bill two pieces of paper? ▾

Feedback geben

Semantic textual similarity

- Applications
 - Plagiarism detection
 - Document clustering
 - Question answering



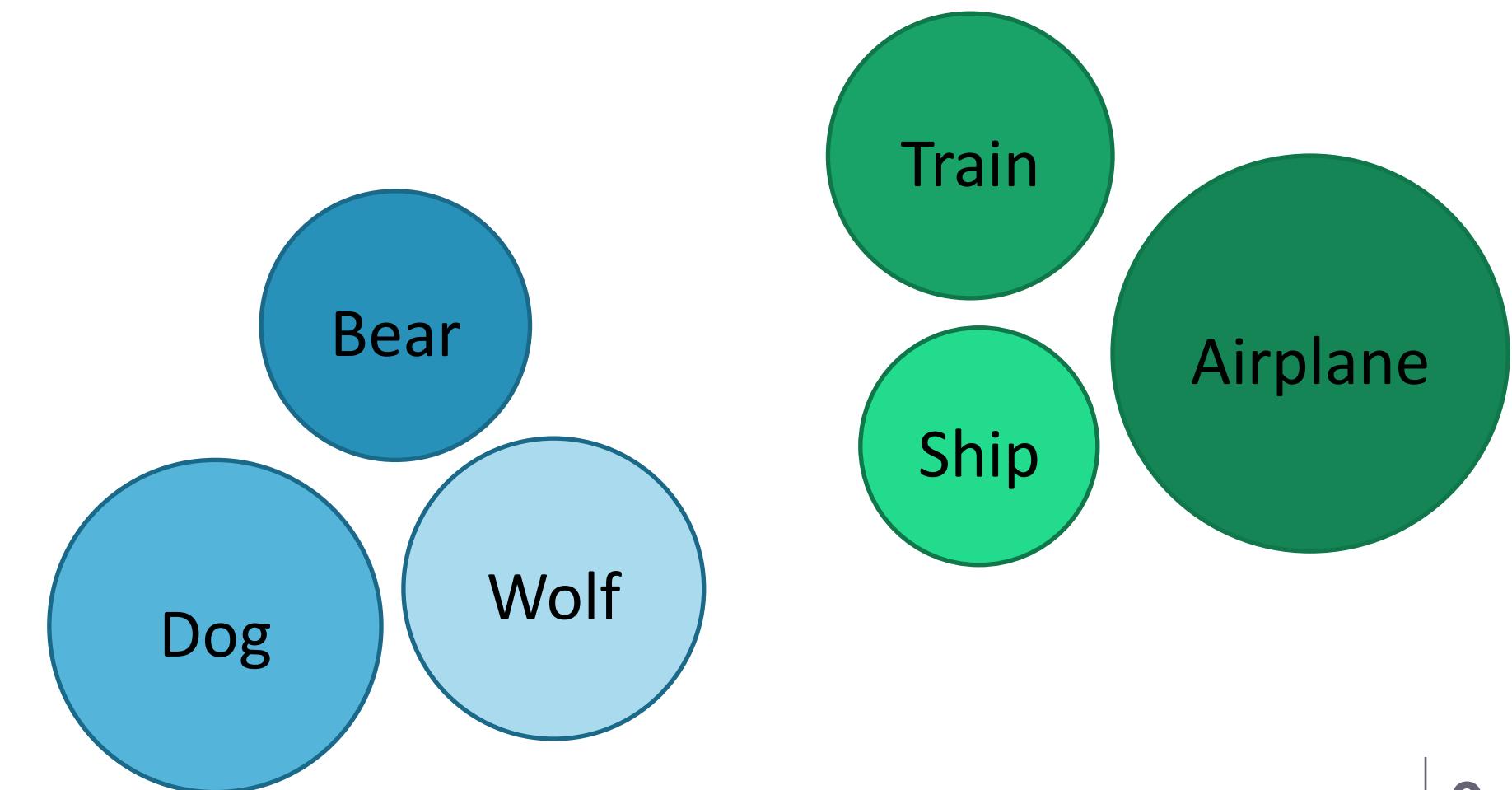
Semantic textual similarity

- What is semantic similarity?
- Semantic similarity in word level
- Semantic similarity in sentence level
- Semantic textual similarity in Python

Semantic similarity in word level

- It tells how close two words/terms are, semantically
- Semantic similarity is often used *synonymously* with semantic *relatedness*

Ship	Airplane	3.8
Ship	Bear	0.2
Dog	Wolf	4.5
Wolf	Bear	3.6



Semantic similarity in word level

- The approaches can be divided into the following categories:
 - Distributional semantics
 - Frequency based
 - Prediction based
 - Knowledge based methods

Distributional semantics

- Frequency based
- PMI

$$PMI(W_1, W_2) = \log_2 \frac{P(W_1, W_2)}{P(W_1)P(W_2)}$$

Distributional semantics

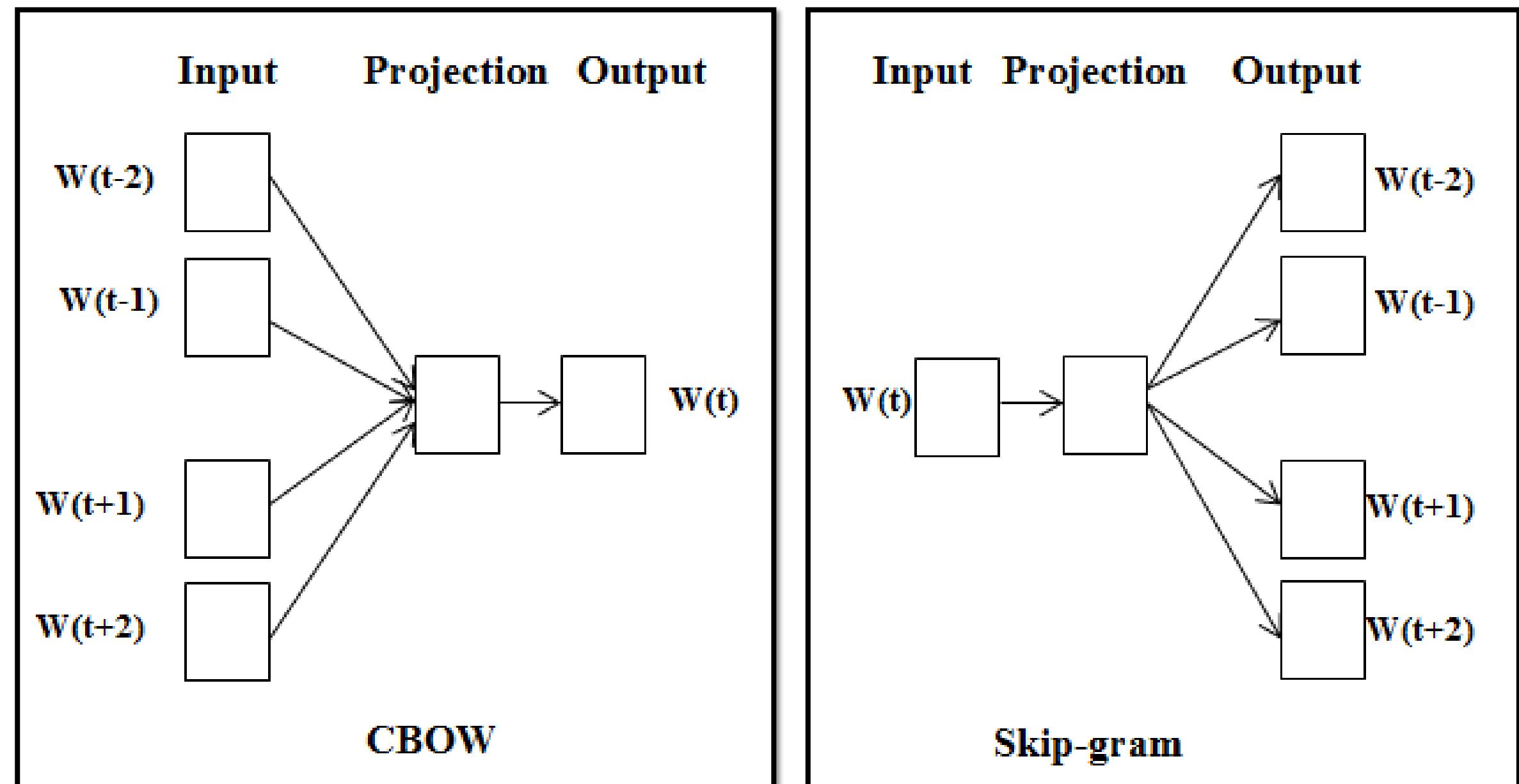
	text	is	a	complex	human	language	representation	natural	and	also	diverse
text	0	1	1	0	0	0	0	0	0	0	0
is	1	0	1	1	1	1	0	0	1	0	0
a	1	1	1	1	1	0	0	0	0	0	0
complex	0	2	1	1	1	2	0	0	1	1	0
human	0	1	1	1	0	2	1	1	0	0	0
language	0	1	0	2	2	1	1	0	0	0	0
representation	0	0	0	0	1	1	1	0	0	0	0
natural	0	0	0	0	1	1	1	0	0	0	0
and	0	2	0	1	0	0	0	0	1	0	0
also	0	1	0	1	0	0	0	0	1	0	1
diverse	0	1	0	0	0	0	0	0	0	1	0

$PMI = -0.51$

$PMI = +1.8$

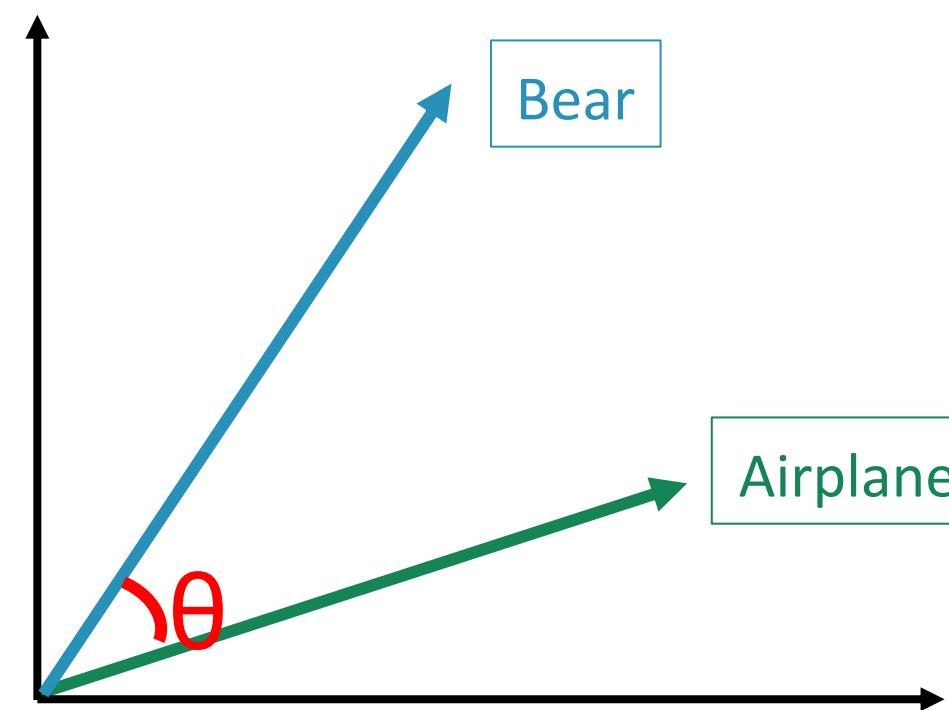
Distributional semantics

- Frequency based
- PMI
- Prediction based
- Word2Vec



Distributional semantics

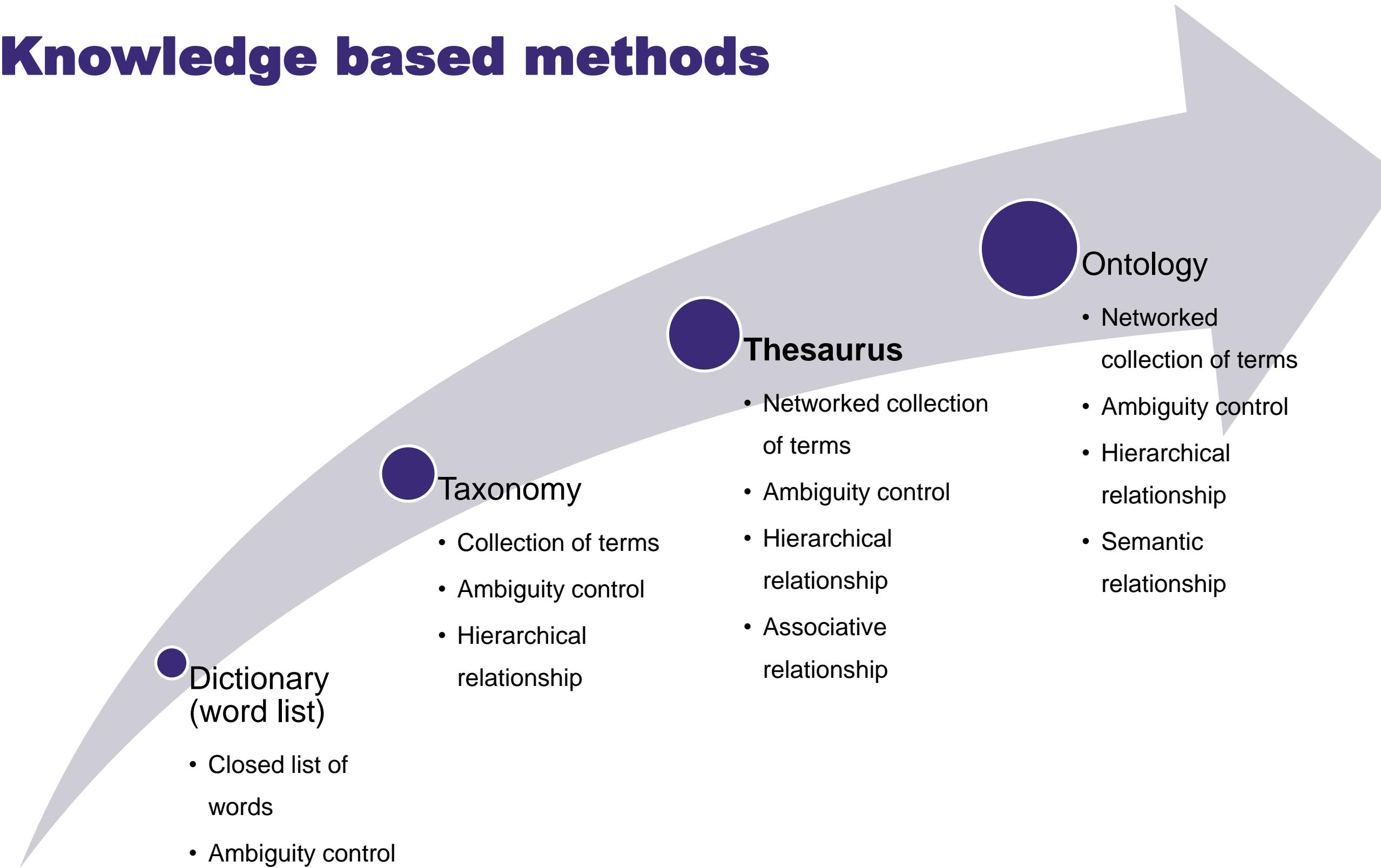
Ship	0.003	- 0.01	0.001	...	0.321	- 0.076	0.014
Airplane	0.002	- 0.009	- 0.001	...	0.337	- 0.054	0.014
Wolf	0.469	0.015	0.373	...	- 0.049	0.533	- 0.148
Dog	0.143	0.445	0.180	...	- 0.683	0.167	- 0.428
Bear	0.397	0.236	- 0.110	...	- 0.256	0.257	- 0.148



Knowledge based methods

- Knowledge-based semantic similarity methods calculate semantic similarity between two terms based on the information derived from underlying knowledge sources like ***ontology, thesaurus, taxonomy, dictionary***
- Ontology, thesaurus, taxonomy, dictionary
- Machine readable knowledge sources that represents how objects are related

Knowledge based methods



Knowledge based methods

Dictionary (term list)

rework

/ri'wə:k/ ⓘ

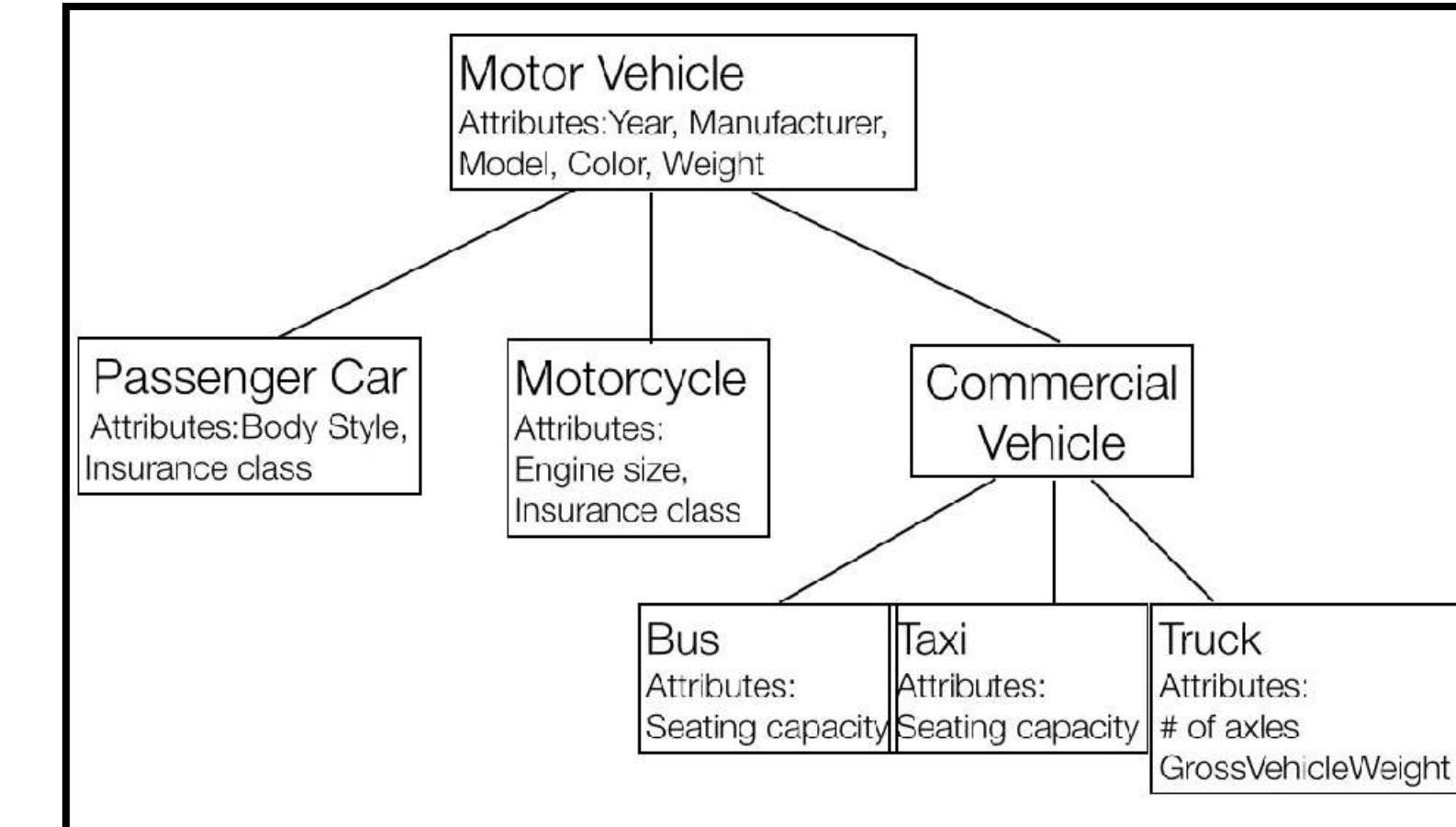
verb

Make changes to the original version of (something)

Sample: Over the course of our trip, the President continually reworked his speech.

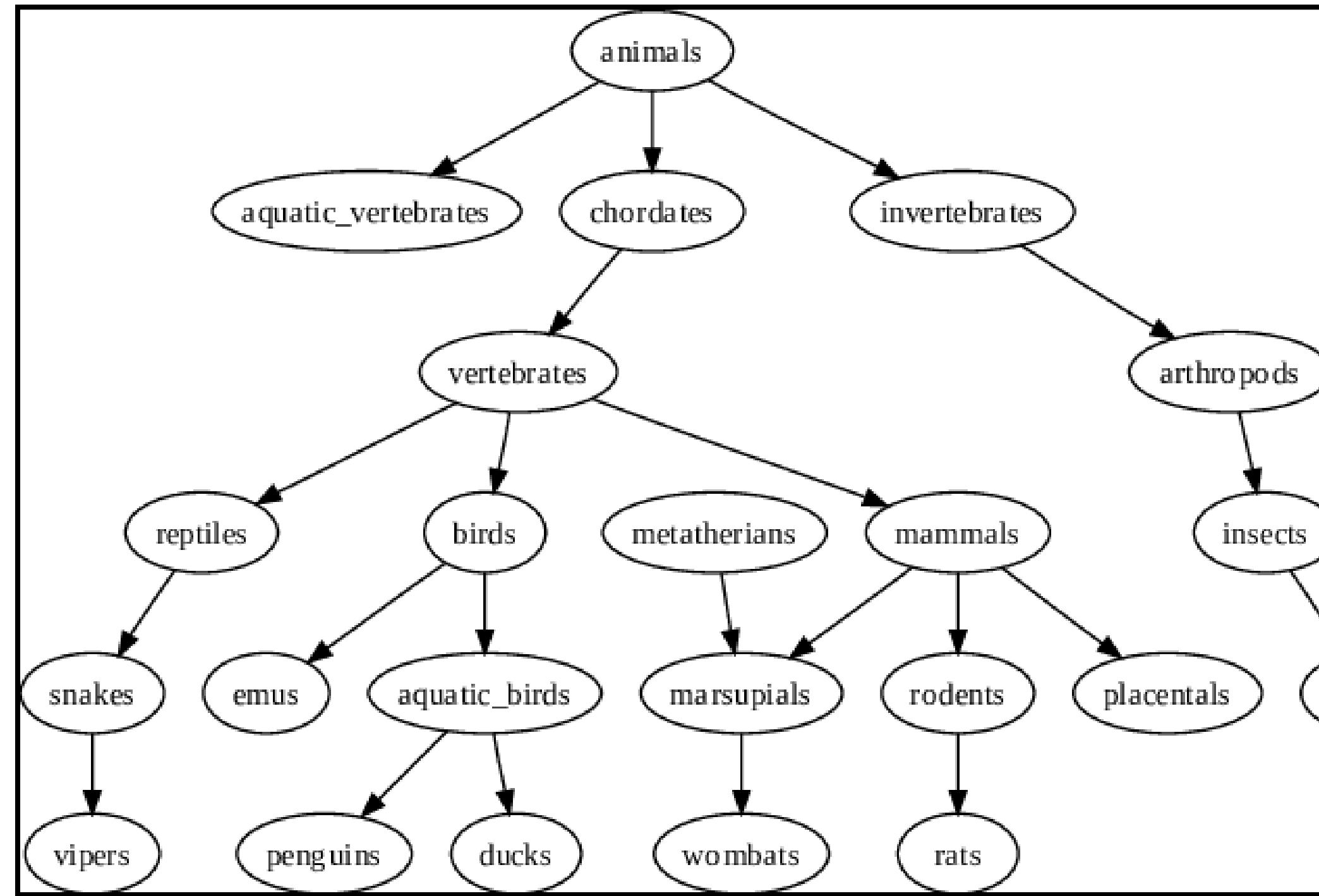
Sample: He reworked the orchestral score for two pianos.

Taxonomy



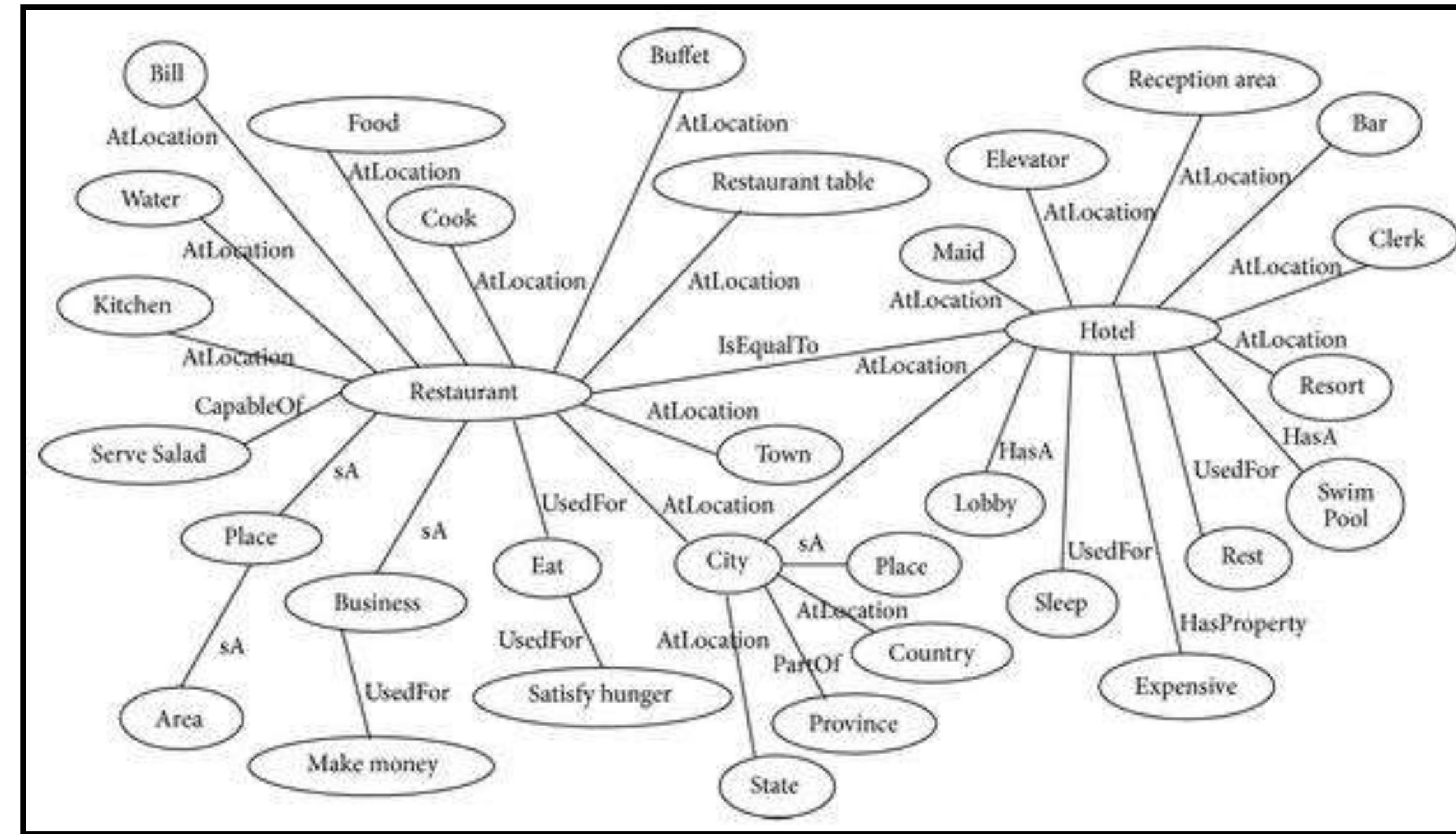
Knowledge based methods

Thesaurus



Knowledge based methods

Ontology



Knowledge based methods

- Thesaurus
- Wordnet
 - WordNet® is a large lexical database of English
 - Nouns, verbs, adjectives and adverbs are grouped into sets of cognitive **synonyms** (**synsets**), each expressing a distinct concept
 - Synsets are interlinked by means of conceptual-semantic and lexical relations

Knowledge based methods

Java

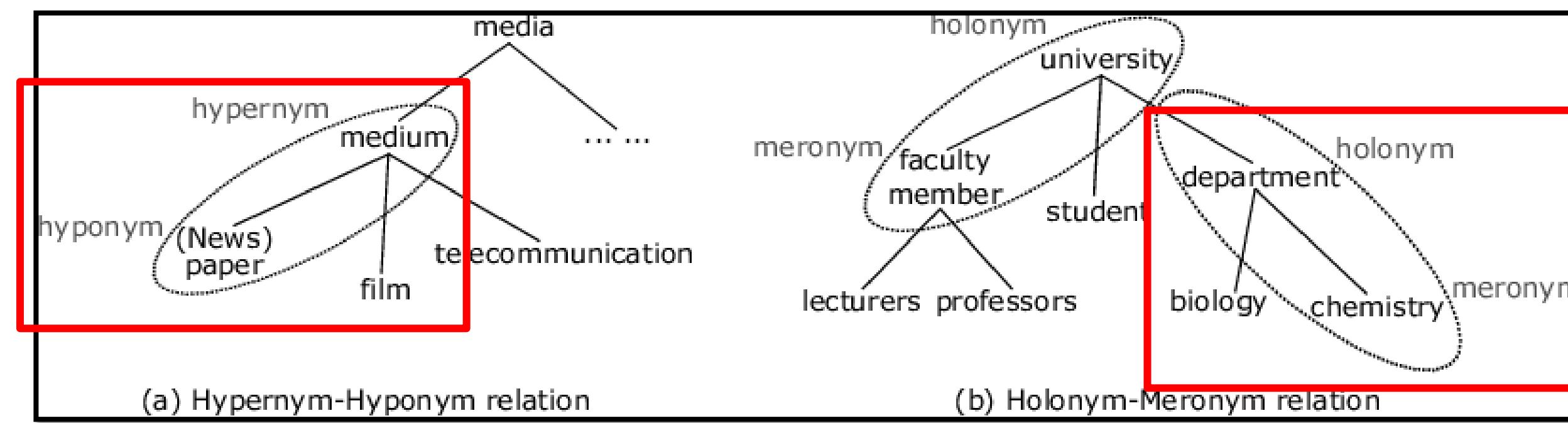
Noun

- S: (n) **Java** (an island in Indonesia to the south of Borneo; one of the world's most densely populated regions)
 - S: (n) **coffee, java** (a beverage consisting of an infusion of ground coffee beans) "he ordered a cup of coffee"
 - S: (n) **Java** (a platform-independent object-oriented programming language)
-
- S: (n) **Java** (a platform-independent object-oriented programming language)
 - *direct hyponym / inherited hyponym / sister term*
 - S: (n) [object-oriented programming language](#), [object-oriented programming language](#) ((computer science) a programming language that enables the programmer to associate a set of procedures with each type of data structure) "C++ is an object-oriented programming language that is an extension of C"

- S: (n) **coffee, java** (a beverage consisting of an infusion of ground coffee beans) "he ordered a cup of coffee"
 - *direct hyponym / full hyponym*
 - S: (n) [coffee substitute](#) (a drink resembling coffee that is sometimes substituted for it)
 - S: (n) [Irish coffee](#) (sweetened coffee with Irish whiskey and whipped cream)
 - S: (n) [cafe au lait](#) (equal parts of coffee and hot milk)
 - S: (n) [cafe noir, demitasse](#) (small cup of strong black coffee without milk or cream)
 - S: (n) [decaffeinated coffee, decaf](#) (coffee with the caffeine removed)
 - S: (n) [drip coffee](#) (coffee made by passing boiling water through a perforated container packed with finely ground coffee)
 - S: (n) [espresso](#) (strong black coffee brewed by forcing hot water under pressure through finely ground coffee beans)
 - S: (n) [cappuccino, cappuccino coffee, coffee cappuccino](#) (equal parts of espresso and hot milk topped with cinnamon and nutmeg and usually whipped cream)
 - S: (n) [iced coffee, ice coffee](#) (a strong sweetened coffee served over ice with cream)
 - S: (n) [instant coffee](#) (dehydrated coffee that can be made into a drink by adding hot water) "the advantages of instant coffee are speed of preparation and long shelf life"
 - S: (n) [mocha, mocha coffee](#) (a superior dark coffee made from beans from Arabia)
 - S: (n) [Turkish coffee](#) (a drink made from pulverized coffee beans; usually sweetened)
 - S: (n) [cafe royale, coffee royal](#) (black coffee with Cognac and lemon peel and sugar)

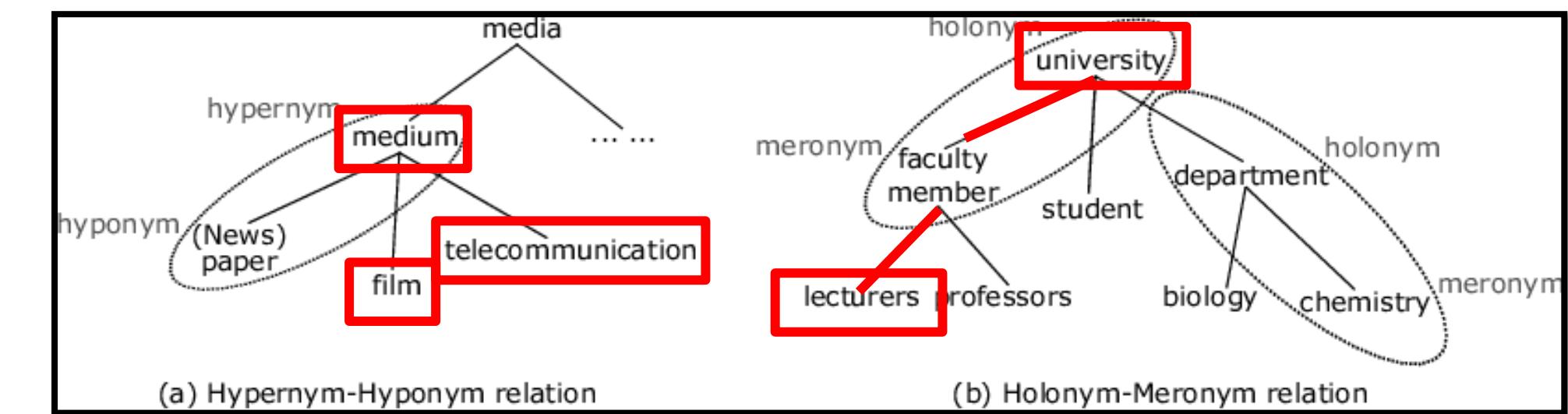
Knowledge based methods

- Semantic relations in Wordnet
 - **Hypernyms:** Y is a hypernym of X if every X is a (kind of) Y (vehicle is a hypernym of bicycle)
 - **Hyponyms:** Y is a hyponym of X if every Y is a (kind of) X (bicycle is a hyponym of vehicle)
 - **Meronym:** Y is a meronym of X if Y is a part of X (window is a meronym of building)
 - **Holonym:** Y is a holonym of X if X is a part of Y (building is a holonym of window)



Knowledge based methods

- How to capture semantic similarity in wordnet?
 - *path* measure
 - *wup* measure



$$sim_{path}(W_1, W_2) = \frac{1}{1 + \min_length(W_1, W_2)}$$

$$sim_{wup}(W_1, W_2) = \frac{2 \times depth(\text{Least Common Subsumer})}{depth(W_1) + depth(W_2)}$$

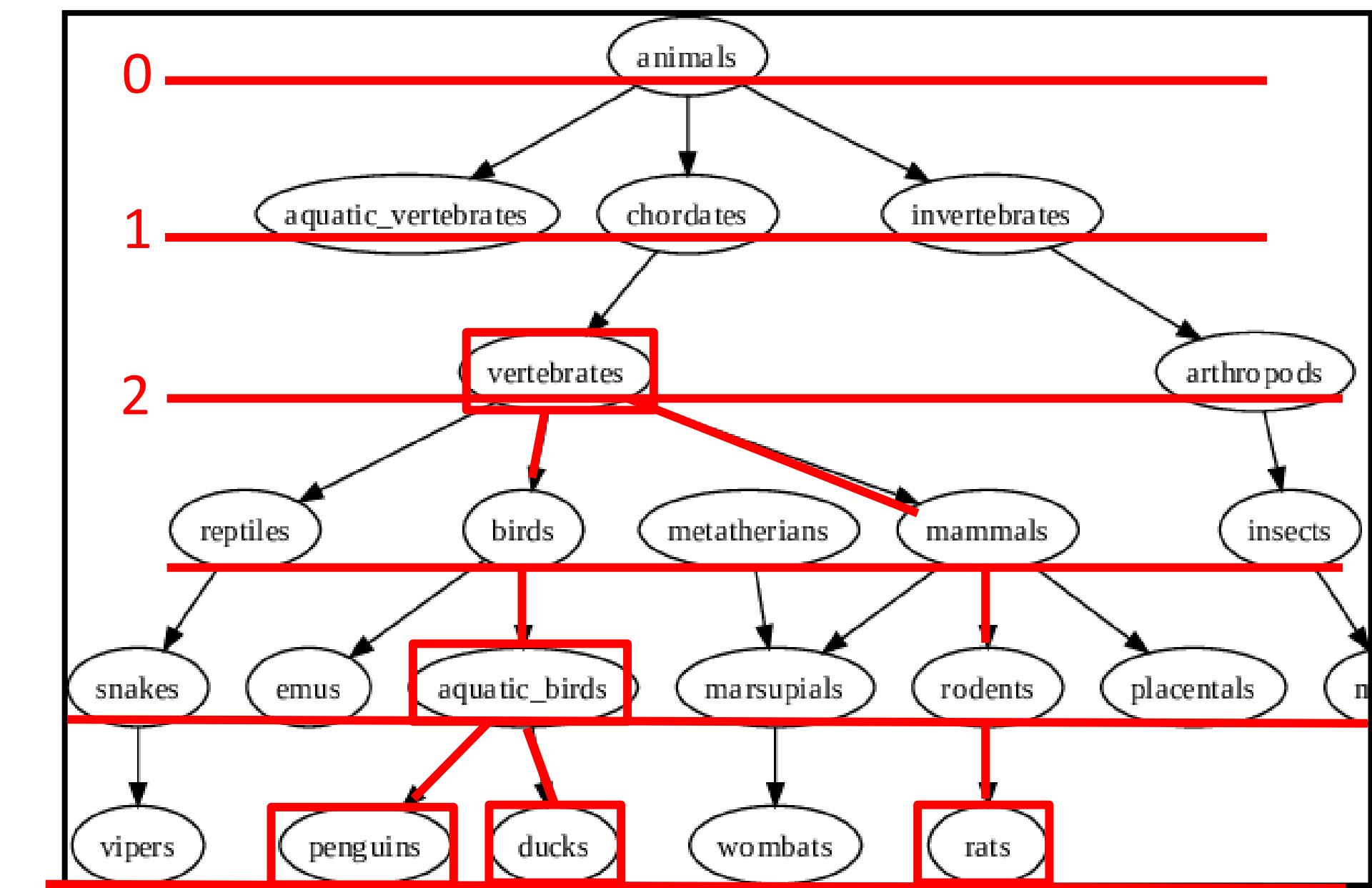
Knowledge based methods

$$sim_{path}(duck, rat) = \frac{1}{7}$$

$$sim_{path}(penguin, duck) = \frac{1}{3}$$

$$sim_{wup}(duck, rat) = \frac{2 \times 2}{5 + 5} = 0.4$$

$$sim_{wup}(penguin, duck) = \frac{2 \times 4}{5 + 5} = 0.8$$

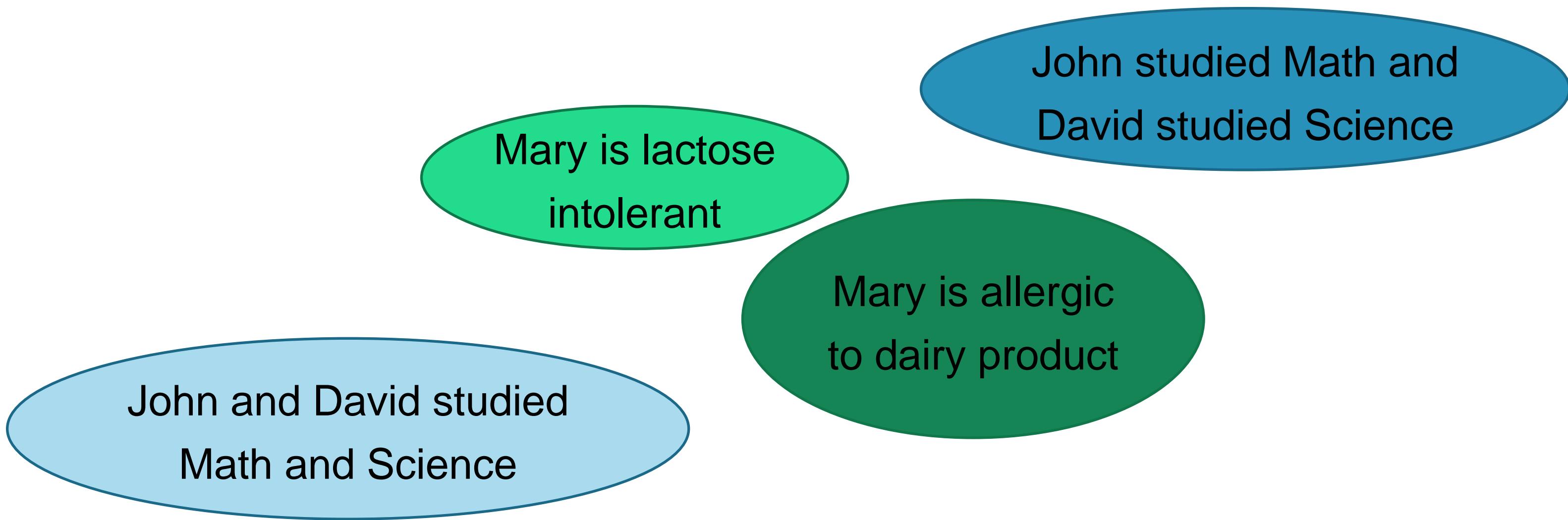


Semantic textual similarity

- What is semantic similarity?
- Semantic similarity in word level
- Semantic similarity in sentence level
- Semantic textual similarity in Python

Semantic similarity in sentence level

- It tells how close two texts (sentences) are, semantically



Semantic similarity in sentence level

- The approaches can be divided into the following categories:
 - Distributional semantics
 - Knowledge based methods

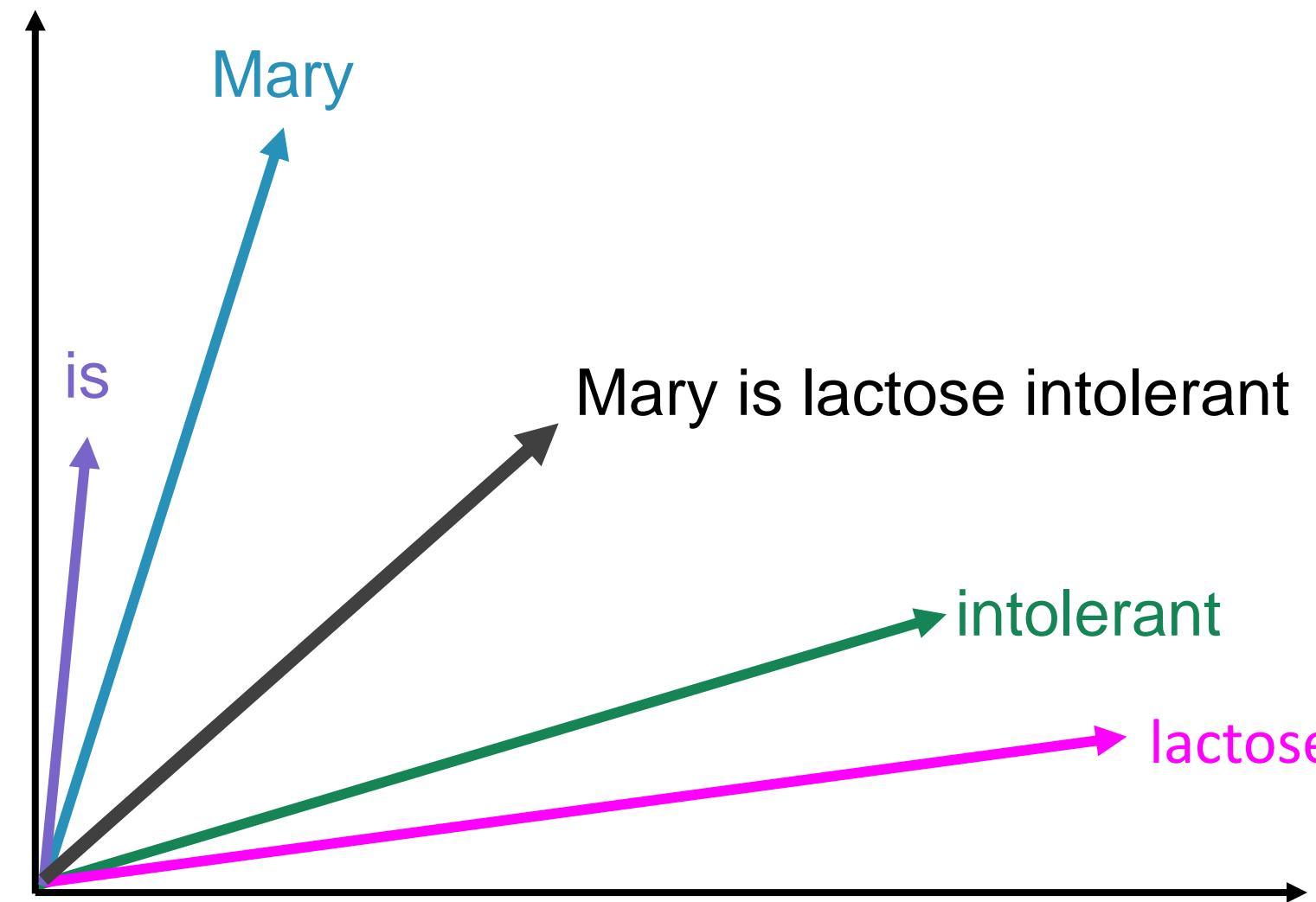
Distributional semantics

- Computing sentence vectors
 - Average of word vectors
 - Average of word vectors with TF-IDF
 - Doc2Vec

Distributional semantics

- Average of word vectors

Mary is lactose intolerant



Distributional semantics

- Average of word vectors

Mary is lactose intolerant

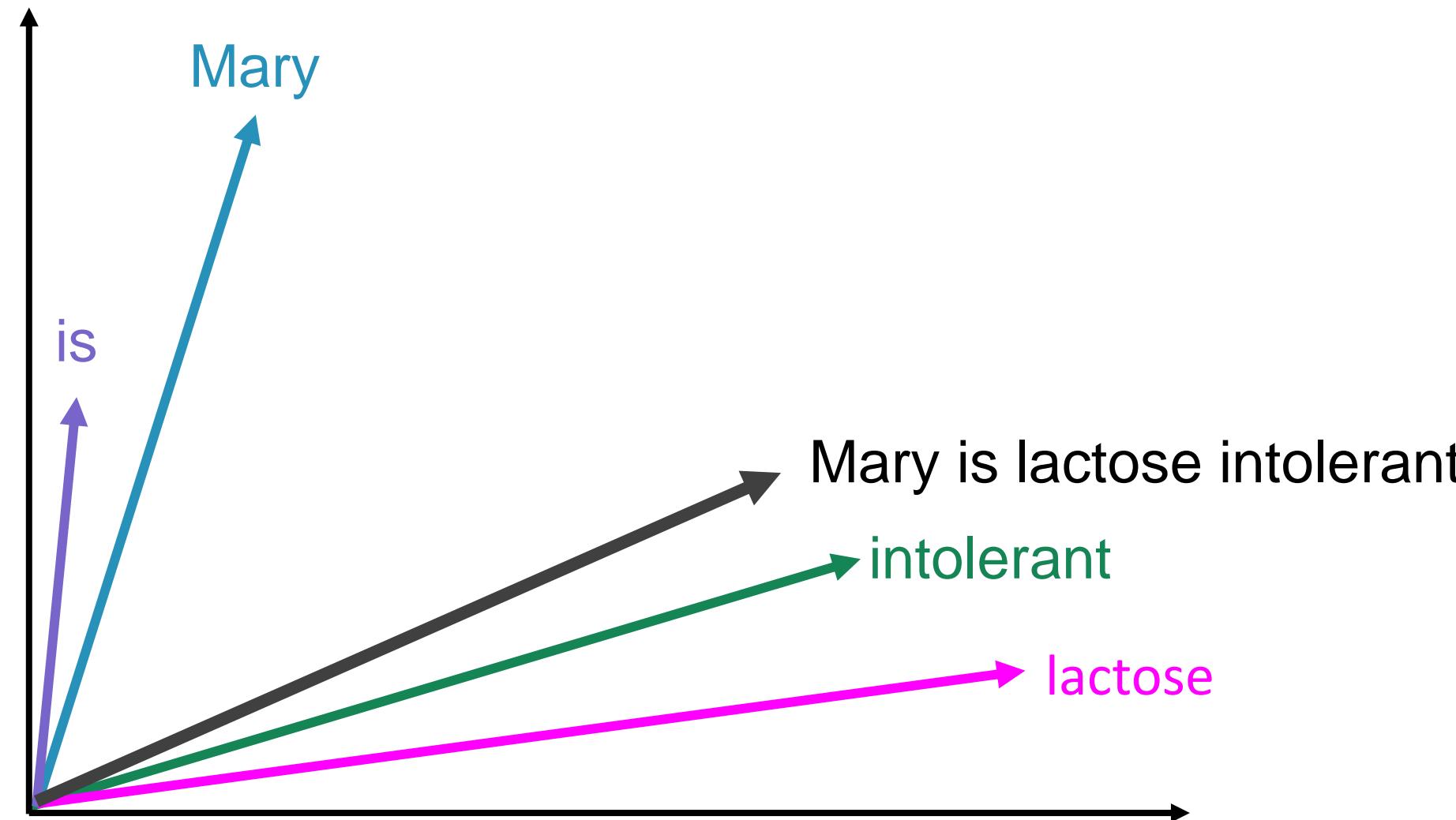
Mary	1	0	1	3	5
is	1	4	2	2	2
lactose	0	3	7	1	1
intolerant	2	1	1	1	2
Sentence	1	2	2.75	1.7	2.5

Distributional semantics

- Average of word vectors with TF-IDF

Mary	TFIDF
is	0.5
lactose	0.1
intolerant	5
	7

Mary is lactose intolerant



Distributional semantics

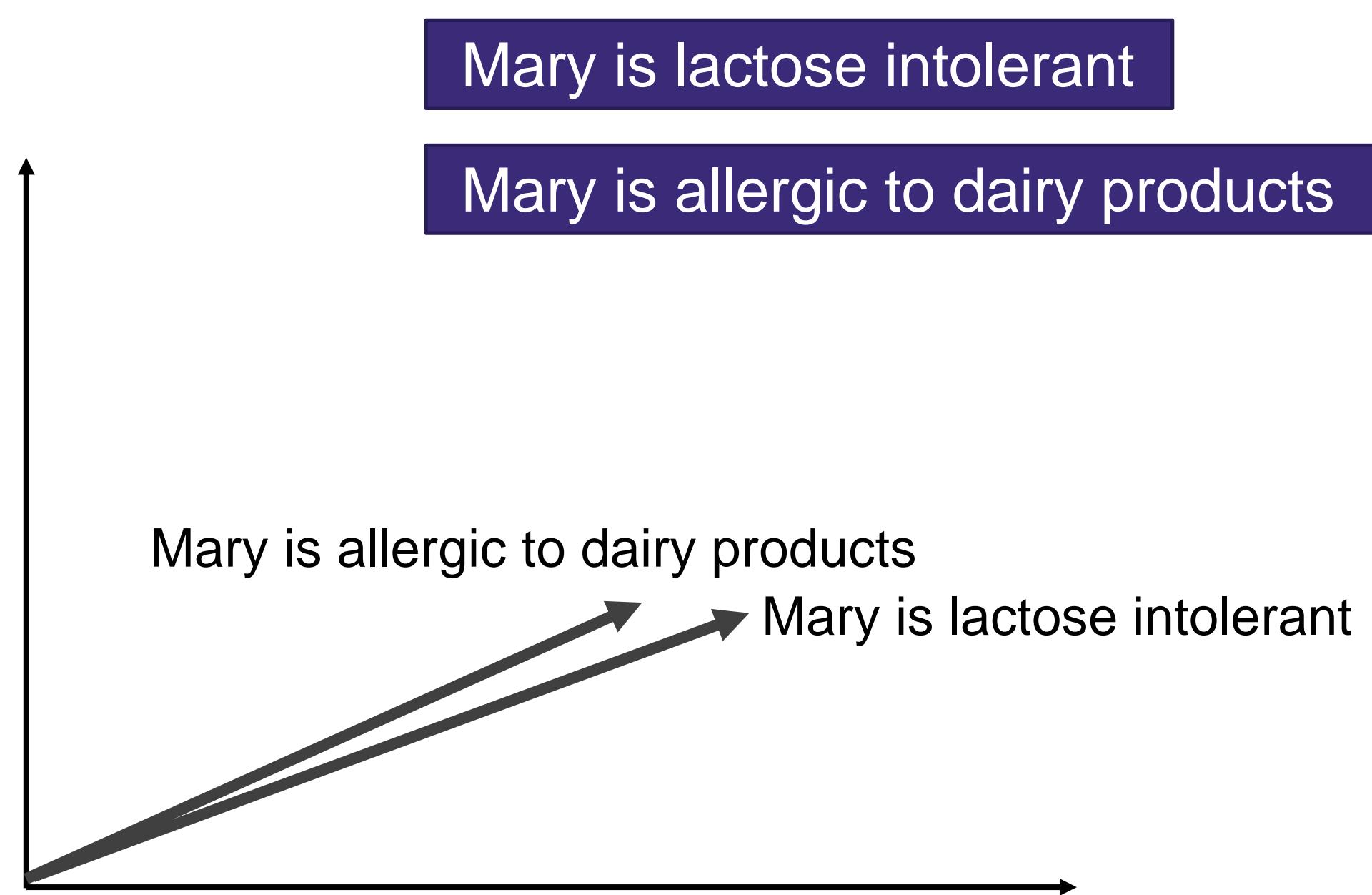
- Average of word vectors with TF-IDF

Mary is lactose intolerant

	TF-IDF					
Mary	0.5	1	0	1	3	5
is	0.1	1	4	2	2	2
lactose	5	0	3	7	1	1
intolerant	7	2	1	1	1	2
Sentence	-	1.15	1.93	3.30	1.15	1.74

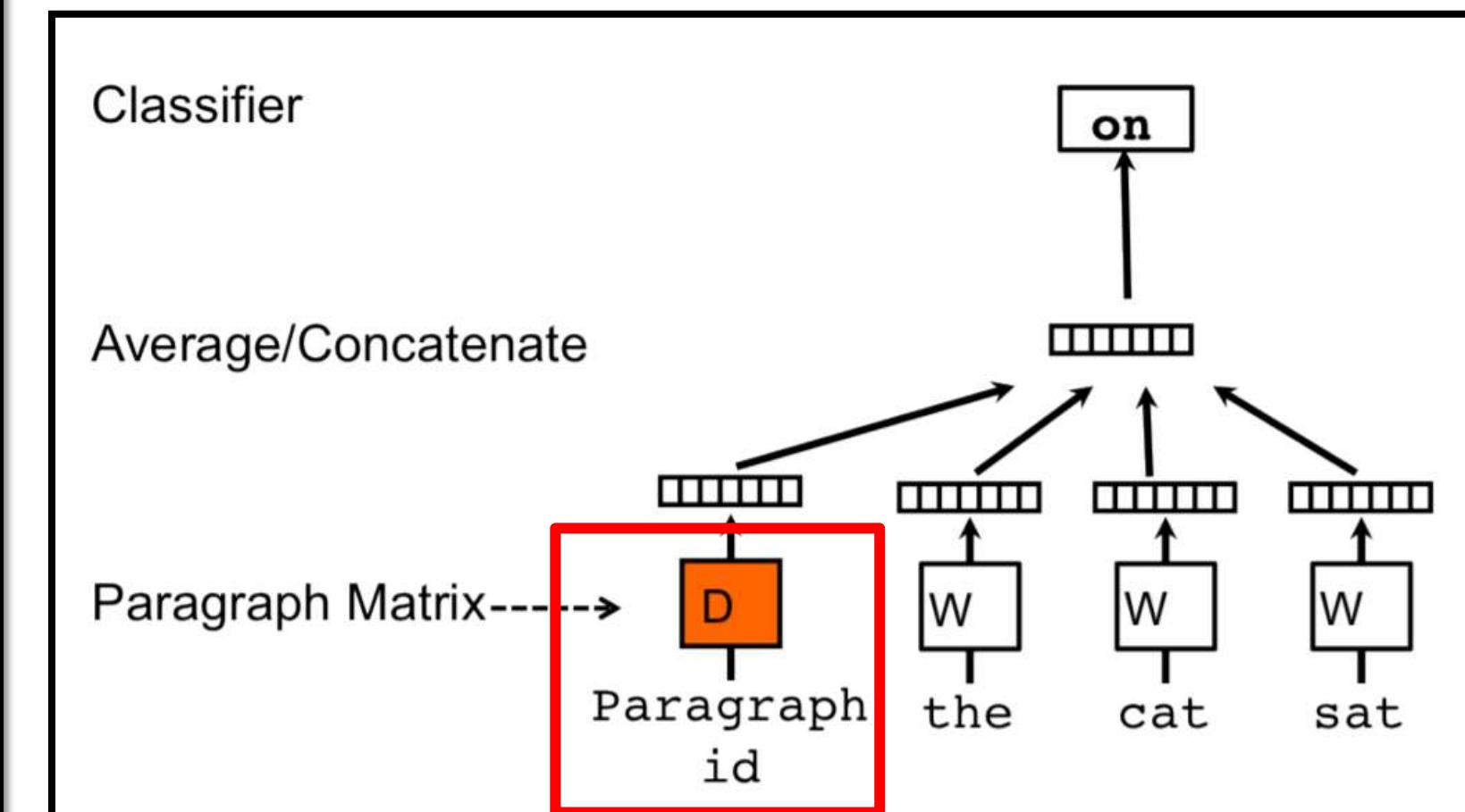
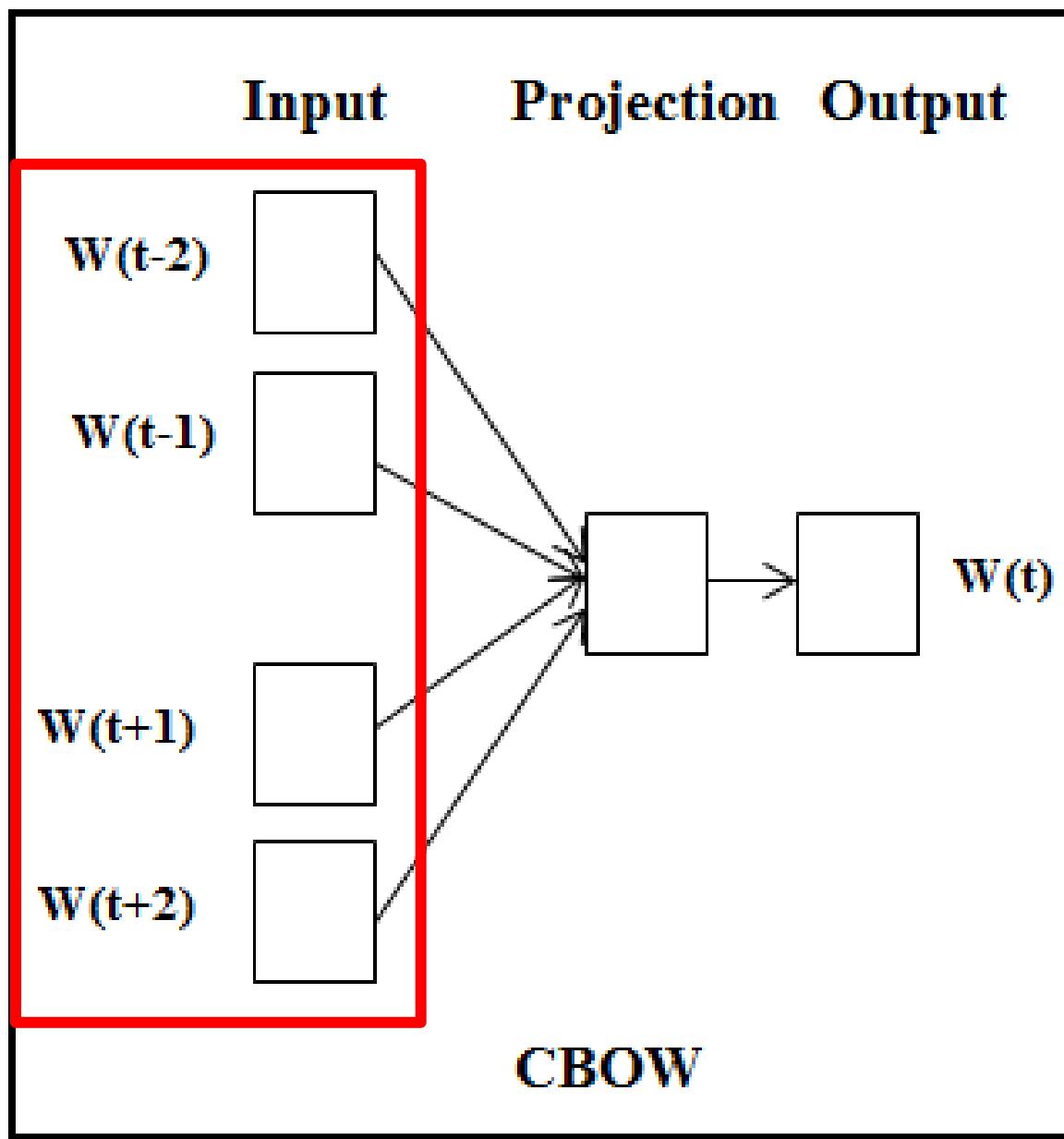
Distributional semantics

- Cosine similarity



Distributional semantics

- Doc2Vec

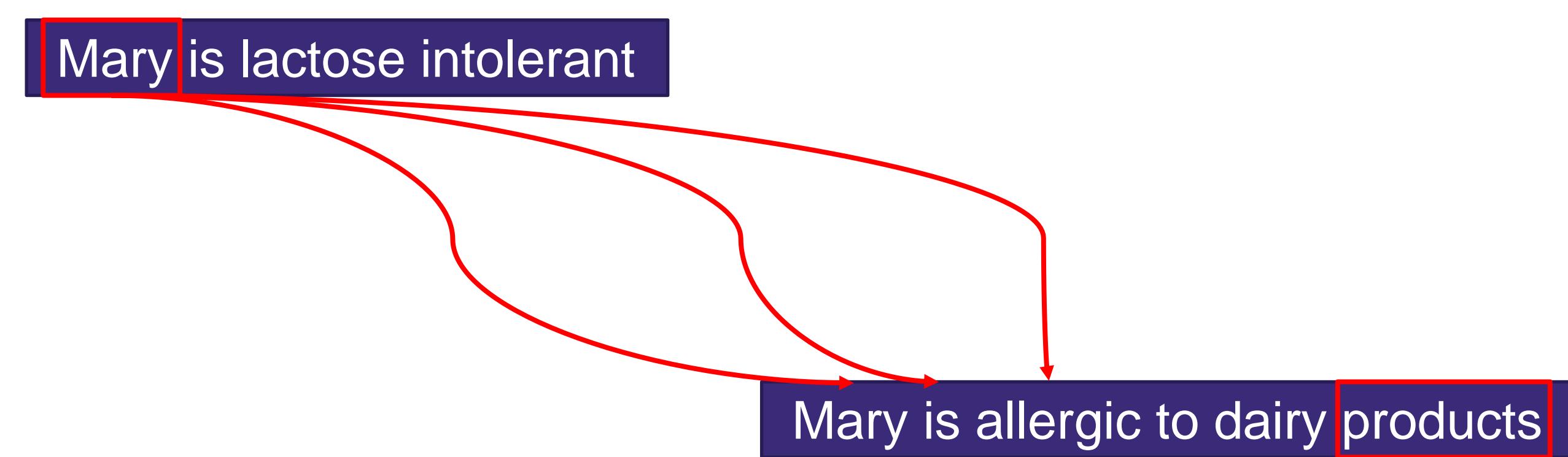


Distributional semantics

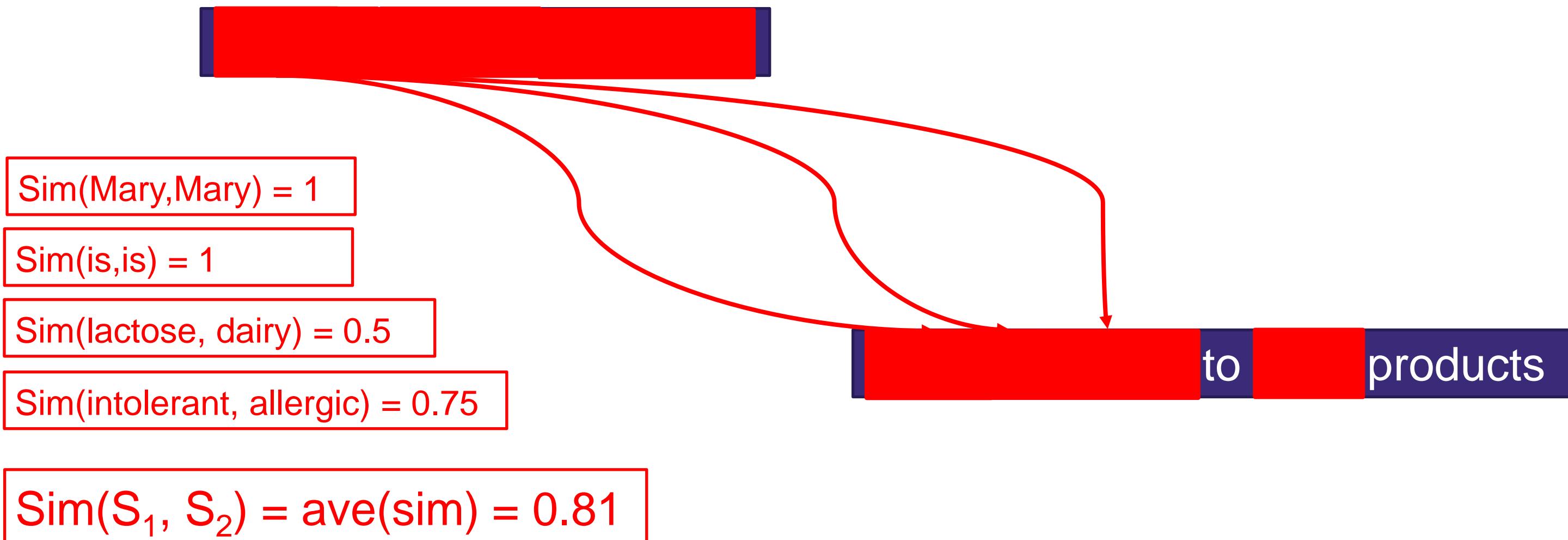
- Doc2Vec

Mary is lactose intolerant	0.112	0.091	0.357	...	- 0.483	0.249	0.747
Mary is allergic to dairy products	0.818	0.343	0.108	...	- 0.777	- 0.310	0.314

Knowledge based methods



Knowledge based methods



Semantic textual similarity

- What is semantic similarity?
- Semantic similarity in word level
- Semantic similarity in sentence level
- Semantic textual similarity in Python

NLTK

- Wordnet synset

```
>>> from nltk.corpus import wordnet  
>>> wordnet.synsets('dog')  
[Synset('dog.n.01'), Synset('frump.n.01'), Synset('dog.n.03'),  
Synset('cad.n.01'), Synset('frank.n.02'), Synset('pawl.n.01'),  
Synset('andiron.n.01'), Synset('chase.v.01')]  
>>> print(wordnet.synset('dog.n.01').definition())  
a member of the genus Canis (probably descended from the common  
wolf) ...
```

Noun

- [S: \(n\) Java](#) (an island in Indonesia to the south of Borneo; one of the world's most densely populated regions)
- [S: \(n\) coffee, java](#) (a beverage consisting of an infusion of ground coffee beans) "he ordered a cup of coffee"
- [S: \(n\) Java](#) (a platform-independent object-oriented programming language)

NLTK

- Wordnet similarity

```
>>> from nltk.corpus import wordnet  
>>> dog = wordnet.synset('dog.n.01')  
>>> cat = wordnet.synset('cat.n.01')  
>>> print(dog.path_similarity(cat))  
0.2  
>>> dog.wup_similarity(cat)  
0.85
```

Gensim

- Doc2Vec

```
>>> from gensim.test.utils import common_texts
>>> from gensim.models.doc2vec import Doc2Vec, TaggedDocument
>>> documents = [TaggedDocument(doc, [i]) for i, doc in
    enumerate(common_texts)]
>>> print(documents[0])
TaggedDocument(['human', 'interface', 'computer'], [0])
>>> model = Doc2Vec(documents, vector_size=5)
>>> vector = model.infer_vector(["semantic", "text", "similarity"])
>>> print(vector)
[-0.07941077, -0.0955774, -0.06963827, -0.02995487, 0.09318832]
```

Summary

- Semantic textual similarity (STS) deals with determining how similar two pieces of texts are

John and David studied Math and Science.



John studied Math and David studied Science.

Mary is allergic to dairy products.



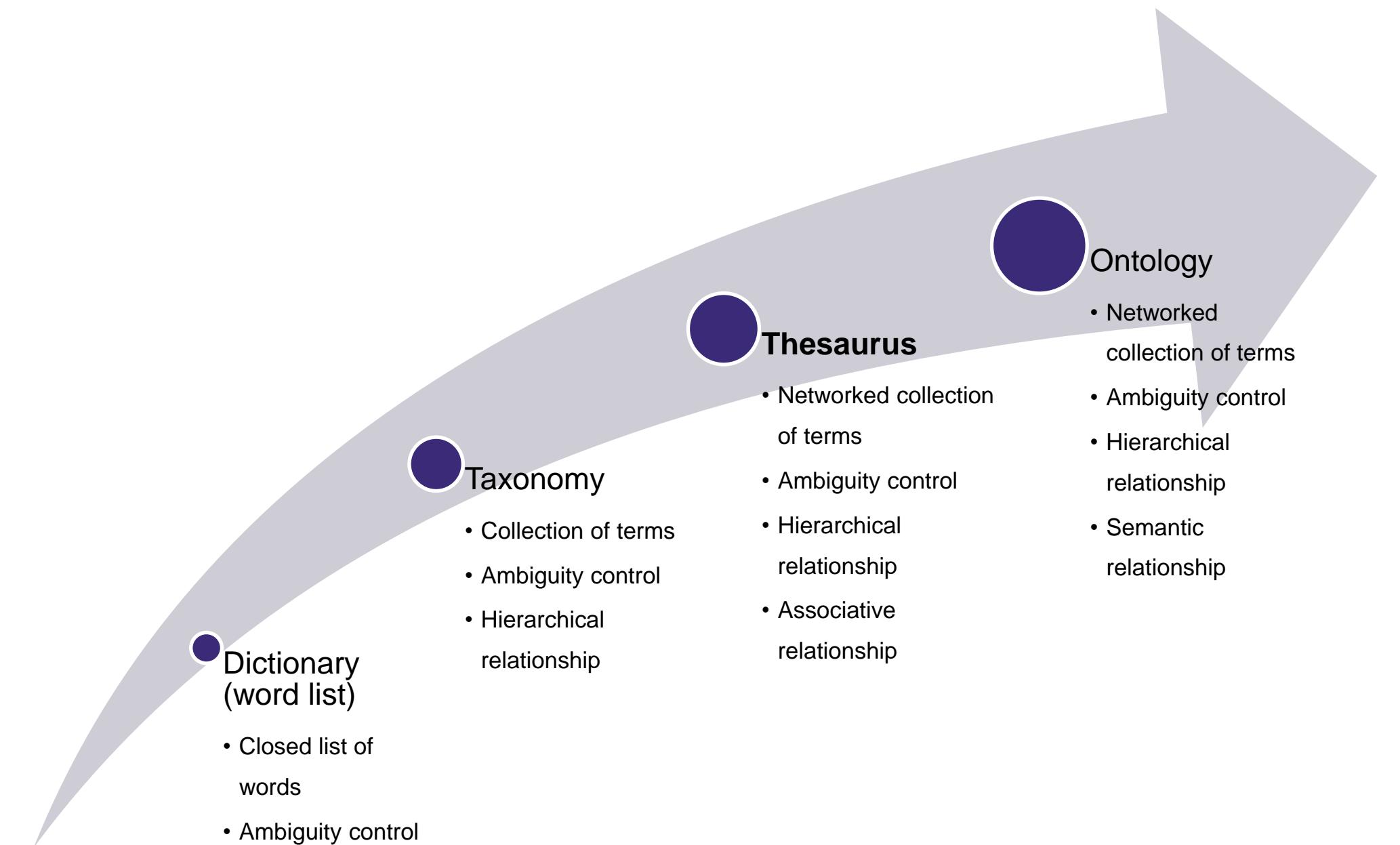
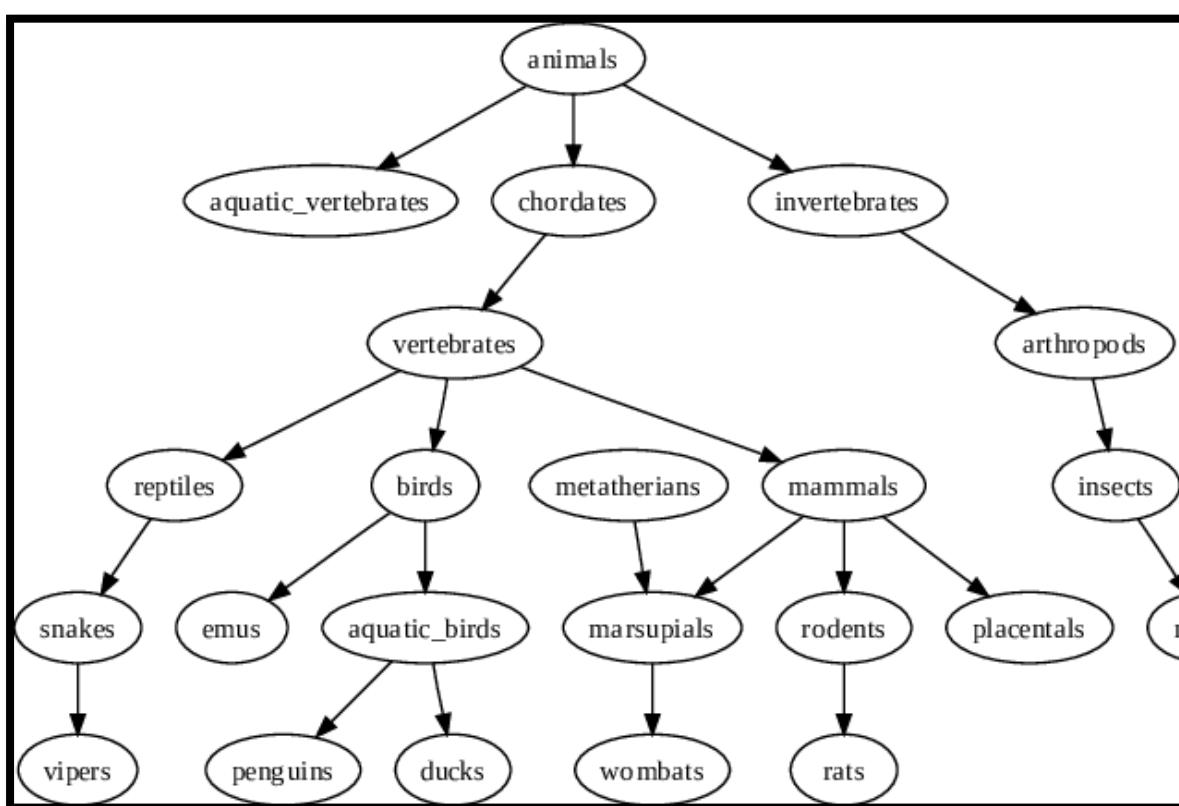
Mary is lactose intolerant.

The screenshot shows a Java code editor with several annotations and code snippets. At the top, it says "Similarity by term frequency: 97.713468057248 Semantic Similarity: 01.29541283829306". The code includes:

```
$ javac d.java  
$ javac d.Rectangle.java  
Now a package/folder with the name shape will be created in the current directory and these class files will be placed in it.  
(d) Java :  
superclasses (the immediate parent and all its ancestors). It can use the inherited methods and variables as they are. It may also override an inherited method by providing its own version, or hide an inherited variable by defining a variable of the same name. The "@Override" is known as annotation (introduced in JDK 1.5), which asks compiler to check whether there is such a method in the superclass to be overridden. This helps greatly if you misspell the name of the method to be overridden. For example, suppose that you wish to override method toString() in a subclass. If @Override is not used and toString() is misspelled as T0String(), it will be treated as a new method in the subclass, instead of overriding the superclass. If @Override is used, the compiler will signal an error. @Override annotation is optional. Annotations are not programming constructs. They have no effect on the program output, it is only used by the compiler, discarded after compilation, and not used by the runtime.  
Annotations are not programming constructs. They have no effect on the program output, it is only used by the compiler, discarded after compilation, and not used by the runtime.  
C++ :  
(e) The super keyword in Java is a reference variable that is used to refer immediate parent class object. Whenever you create the instance of subclass, an instance of parent class is created implicitly i.e. referred by super reference variable. Similarly, the keyword super refers to the superclass, which could be the immediate parent or its ancestor. The keyword super allows the subclass to access superclass' methods and variables within the subclass' definition. If the subclass overrides a method inherited from its superclass, says toString(), you can use super.toString() to invoke the superclass' version within the subclass definition. Similarly, if your subclass hides one of the superclasses' variable, you can use super.variableName to refer to the hidden variable within the subclass.  
In OOP, we often organize classes in hierarchy to avoid duplication and reduce redundancy. The classes in the lower hierarchy inherit all the variables (static attributes) and methods (dynamic behaviors) from the higher hierarchies. A class in the lower hierarchy is called a subclass (or derived, child, extended class). A class in the upper hierarchy is called a superclass (or base, parent class). By pulling out all the common variables and methods into the superclasses, and leave the specialized variables and methods in the subclasses, redundancy can be greatly reduced or eliminated as these common variables and methods do not need to be repeated in all the subclasses.  
(g) Dynamic dispatch contrasts with static dispatch in which the implementation of a polymorphic operation is selected at compile time. The purpose of dynamic dispatch is to support cases where the appropriate implementation of a polymorphic operation can't be determined at compile time because it depends on the runtime type of one or more actual parameters to the operation. Dynamic method dispatch is a mechanism by which a call to an overridden method is resolved at runtime. This is how Java implements runtime polymorphism. When an overridden method is called by a reference, Java determines which version of that method to execute based on the type of object it refers to. In simple words the type of object which it referred determines which version of overridden method will be called.  
(h) The word "polymorphism" means "many forms". Polymorphism just means that, basically, once you've got a child class, you can use objects of that child class wherever you'd use objects of the parent class.
```

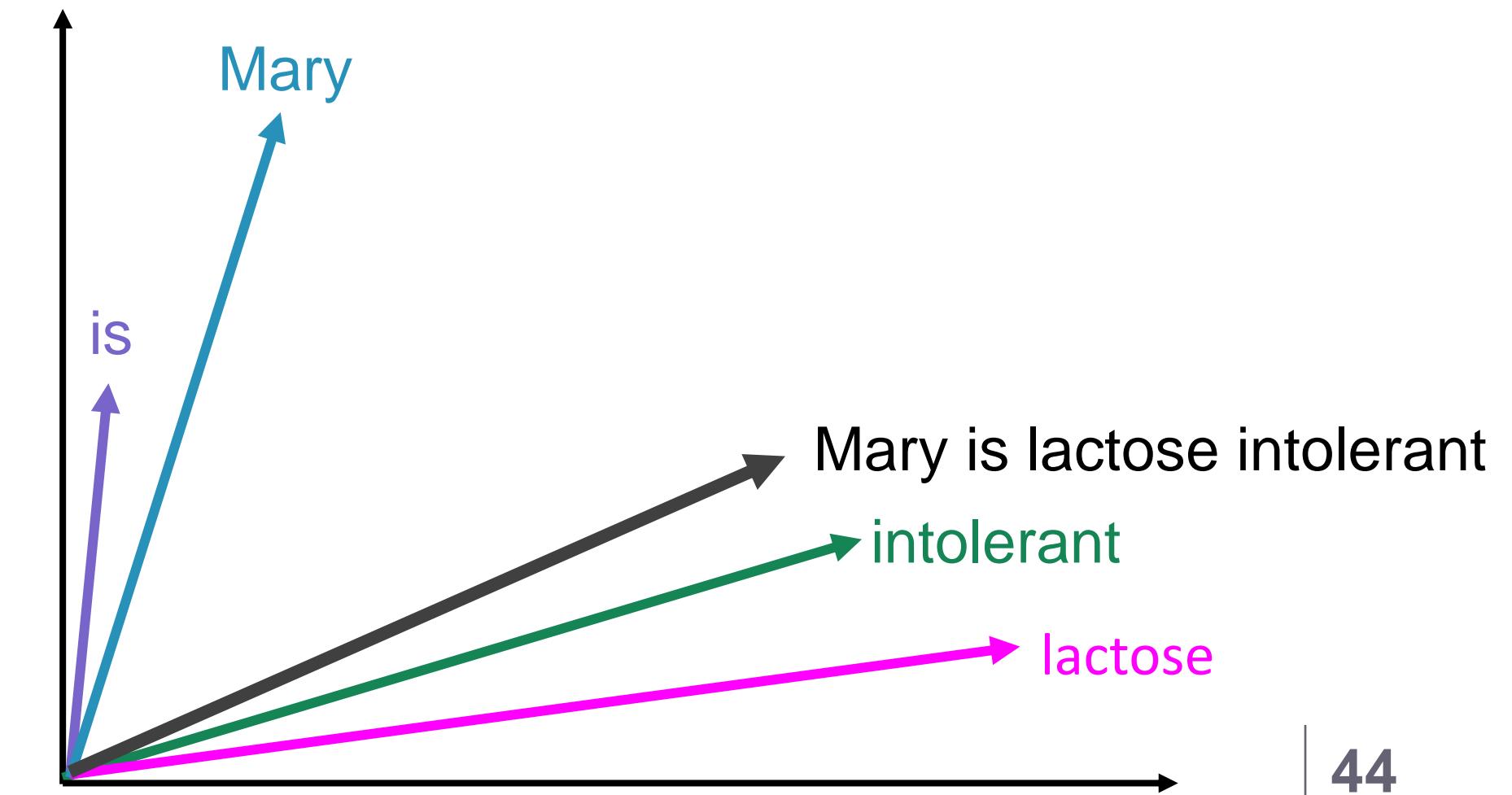
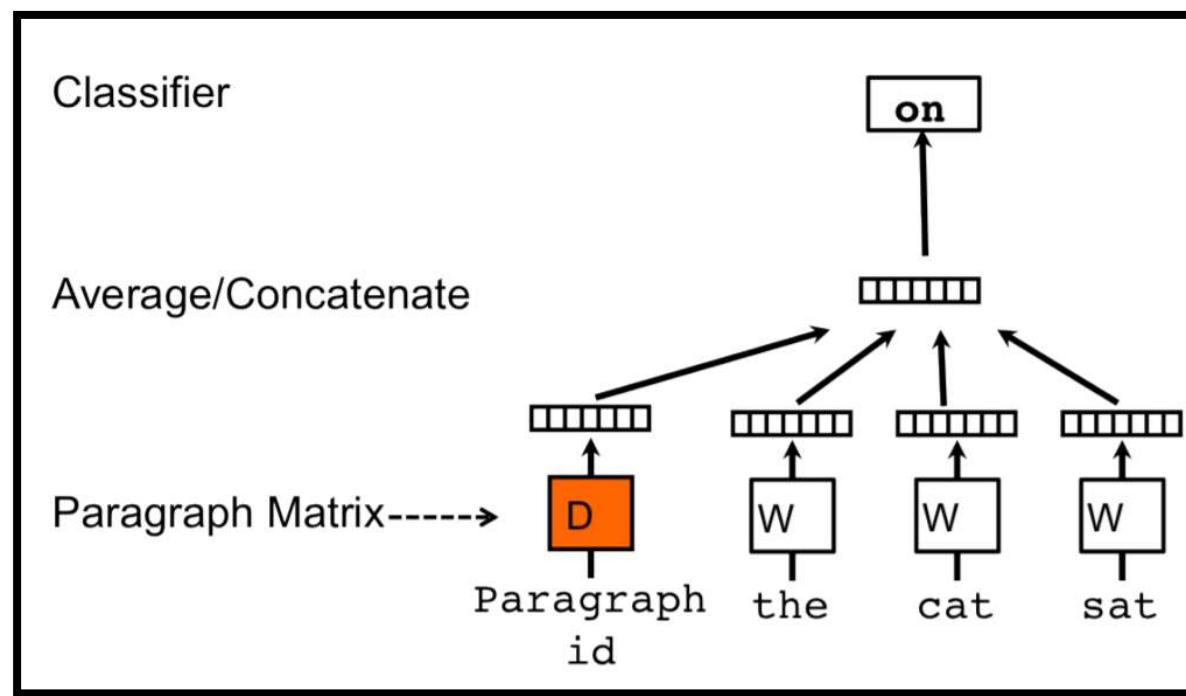
Summary

- Semantic similarity in word level
- Distributional semantics
 - Frequency based
 - Prediction based
- Knowledge based methods



Summary

- Semantic similarity in sentence level
 - Average of word vectors
 - Average of word vectors with TF-IDF
 - Doc2Vec
 - Knowledge based methods



Spam Filtering

Salar Mohtaj | DFKI

Spam filtering

- Spam filtering task
- Naïve Bayes spam filtering
- Feed forward neural network for spam filtering
- Evaluation of spam filters

Spam detection

- Spam filtering task
- Naïve Bayes spam filtering
- Feed forward neural network for spam filtering
- Evaluation of spam filters

Spam filtering task

- A spam filter is a program that is used to detect unsolicited and ***unwanted email*** and prevent those messages from getting to a user's inbox
- The simplest and earliest can be set to watch for ***particular words*** in the subject line of messages and to exclude these from the user's inbox



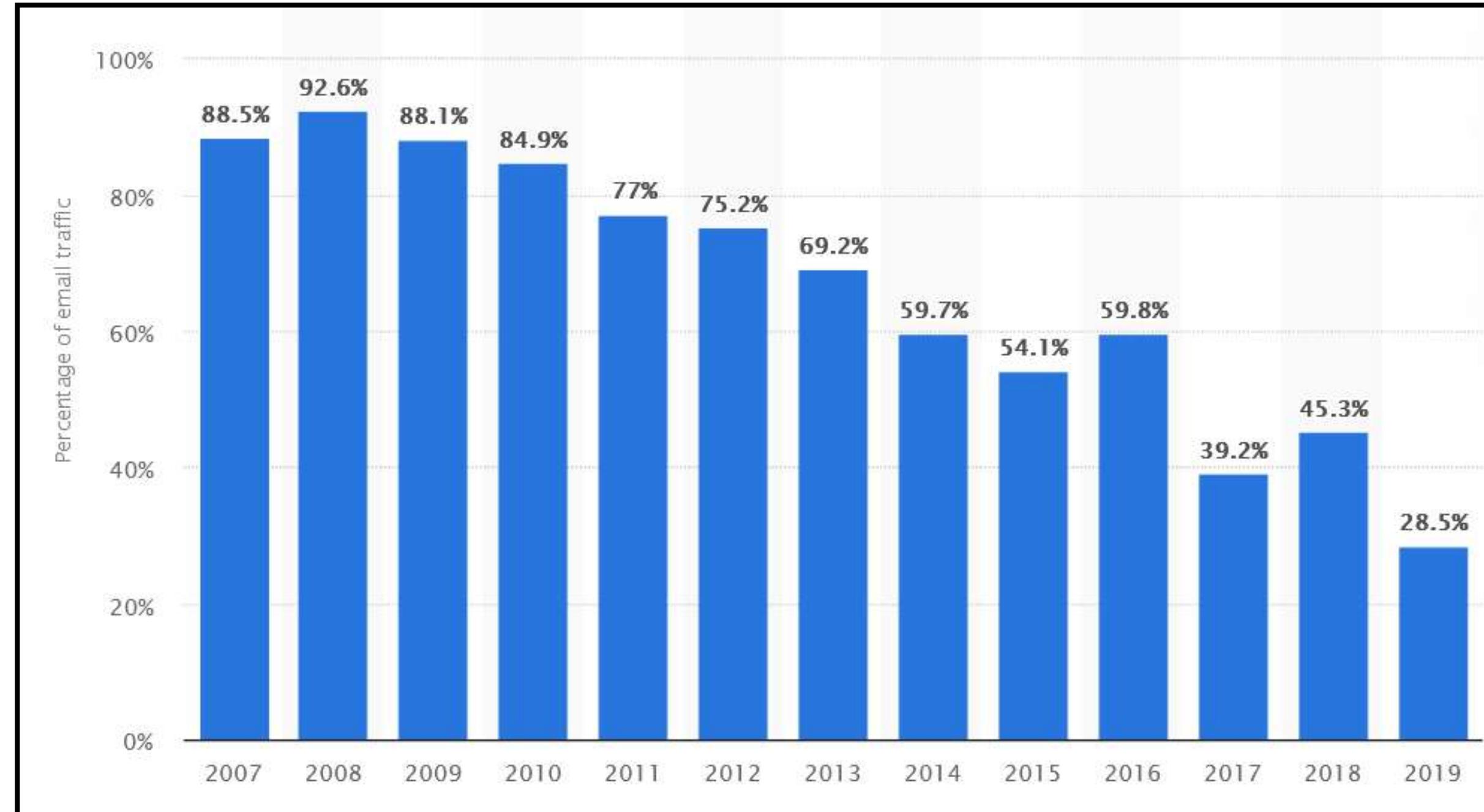
Spam filtering task

- **Content filters:** parse the content of messages, scanning for words that are commonly used in spam emails.
- **Header filters:** examine the email header source to look for suspicious information (such as spammer email addresses).
- **Blocklist filters:** stop emails that come from a blocklist of suspicious IP addresses. Some filters go further and check the IP reputation of the IP address.
- **Rules-based filters:** apply customized rules designed by the organization to exclude emails from specific senders, or emails containing specific words in their subject line or body.

Classification task

▲ type	▲ text
ham	Hope you are having a good week. Just checking in
ham	K...give back my thanks.
ham	Am also doing in cbe only. But have to pay.
spam	complimentary 4 STAR Ibiza Holiday or £10,000 cash needs your URGENT collection. 09066364349 NOW fro...

Spam filtering task



Graph from <http://statista.com>

Spam detection

- Spam filtering task
- Naïve Bayes spam filtering
- Feed forward neural network for spam filtering
- Evaluation of spam filters

Naïve Bayes spam filtering

- The Naive Bayes classifier is a simple classifier that classifies based on probabilities of events
 - It is applied commonly to text classification
- Though it is a simple algorithm, it performs well in many text classification problems
- It is a classification technique based on **Bayes' theorem** with an assumption of independence among predictors
- As with any machine learning model, we need to have an existing set of examples (training set) for each category (spam/non-spam)

Naïve Bayes spam filtering

congratulations you have won a playstation 5

$P(\text{ham}|\text{congratulations you have won a playstation 5})$

$P(\text{spam}|\text{congratulations you have won a playstation 5})$

$P(C_k|X)$

$C_1 = \text{ham}$

$C_2 = \text{spam}$

$X = \text{congratulations you have won a playstation 5}$

Naïve Bayes spam filtering

$$P(C_k|X)$$

- The problem with the above formulation is that if the number of features n is large then basing such a model on probability tables is infeasible

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)}$$

Bayes' theorem

$$P(C_k|X) = \frac{P(C_k)P(X|C_k)}{P(X)}$$

Naïve Bayes spam filtering

$$P(C_k|X) = \frac{P(C_k)P(X|C_k)}{P(X)}$$

naïve" conditional independence assumptions

$$P(C_k)P(X|C_k) = P(C_k)P(x_1|C_k)P(x_2|C_k) \dots P(x_n|C_k)$$

$$\hat{y} = \underset{k \in \{1, \dots, K\}}{\operatorname{argmax}} P(C_k) \prod_{i=1}^n P(x_i|C_k)$$

Naïve Bayes spam filtering

congratulation you have won a gift card	spam
your package is out for delivery	ham
the event is postponed to the next week	ham
your PlayStation account is temporary locked	spam
congratulation you are hired	spam
congratulation you have won a PlayStation 5	?

congratulation you have won gift card	spam
your PlayStation account is temporary locked	spam
congratulation you are hired	spam
your package is out delivery	ham
event is postponed next week	ham

Naïve Bayes spam filtering

congratulation you have won a PlayStation 5

?

congratulation you have won a gift card	spam
your package is out for delivery	ham
the event is postponed to the next week	ham
your PlayStation account is temporary locked	spam
congratulation you are hired	spam
congratulation you have won a PlayStation 5	?

$$P(C_k)P(X|C_k) = P(C_k)P(\text{congratulation}|C_k)P(\text{you}|C_k)P(\text{have}|C_k)P(\text{won}|C_k)P(\text{PlayStation}|C_k)$$

$$P(C_k)$$

$$P(\text{ham}) = 2/5$$

$$P(\text{spam}) = 3/5$$

Naïve Bayes spam filtering

congratulation you have won a gift card	spam
your package is out for delivery	ham
the event is postponed to the next week	ham
your PlayStation account is temporary locked	spam
congratulation you are hired	spam
congratulation you have won a PlayStation 5	?

	you	account	PlayStation	is	hired	...	week	locked	congratulation
S_1	1	0	0	0	0		0	0	1
S_2	0	0	0	1	0		0	0	0
S_3	0	0	0	0	0		1	0	0
S_4	0	1	1	1	0		0	1	0
S_5	1	0	0	0	1		0	0	1

Naïve Bayes spam filtering

	you	account	PlayStation	is	hired	...	week	locked	congratulation
S ₁	1	0	0	0	0		0	0	1
S ₄	0	1	1	1	0		0	1	0
S ₅	1	0	0	0	1		0	0	1
S ₂	0	0	0	1	0		0	0	0
S ₃	0	0	0	0	0		1	0	0

$$P(C_k)P(X|C_k) = P(C_k)P(\text{congratulation}|C_k)P(\text{you}|C_k)P(\text{have}|C_k)P(\text{won}|C_k)P(\text{PlayStation}|C_k)$$

$$P(\text{congratulation}|\text{spam}) = 2/16$$

Naïve Bayes spam filtering

	you	account	PlayStation	is	hired	...	week	locked	congratulation
S_1	1	0	0	0	0		0	0	1
S_4	0	1	1	1	0		0	1	0
S_5	1	0	0	0	1		0	0	1
S_2	0	0	0	1	0		0	0	0
S_3	0	0	0	0	0		1	0	0

$$P(C_k)P(X|C_k) = P(C_k)P(\text{congratulation}|C_k)P(\text{you}|C_k)P(\text{have}|C_k)P(\text{won}|C_k)P(\text{PlayStation}|C_k)$$

$$P(\text{congratulation}|\text{spam}) = 2/16$$

$$P(\text{congratulation}|\text{ham}) = 0/10$$

Naïve Bayes spam filtering

	you	account	PlayStation	is	hired	...	week	locked	congratulation
S_1	1	0	0	0	0		0	0	1
S_4	0	1	1	1	0		0	1	0
S_5	1	0	0	0	1		0	0	1
S_2	0	0	0	1	0		0	0	0
S_3	0	0	0	0	0		1	0	0

$$P(C_k)P(X|C_k) = P(C_k)P(\text{congratulation}|C_k)P(\text{you}|C_k)P(\text{have}|C_k)P(\text{won}|C_k)P(\text{PlayStation}|C_k)$$

$$P(\text{congratulation}|\text{spam}) = 2/16$$

$$P(\text{congratulation}|\text{ham}) = 0/10$$

$$P(\text{is}|\text{spam}) = 1/16$$

$$P(\text{is}|\text{ham}) = 1/10$$

Naïve Bayes spam filtering

$$P(C_k)P(X|C_k) = P(C_k)P(\text{congratulation}|C_k)P(\text{you}|C_k)P(\text{have}|C_k)P(\text{won}|C_k)P(\text{PlayStation}|C_k)$$

$$P(\text{ham}|X) = 2/5 \times 1/10 \times \dots$$

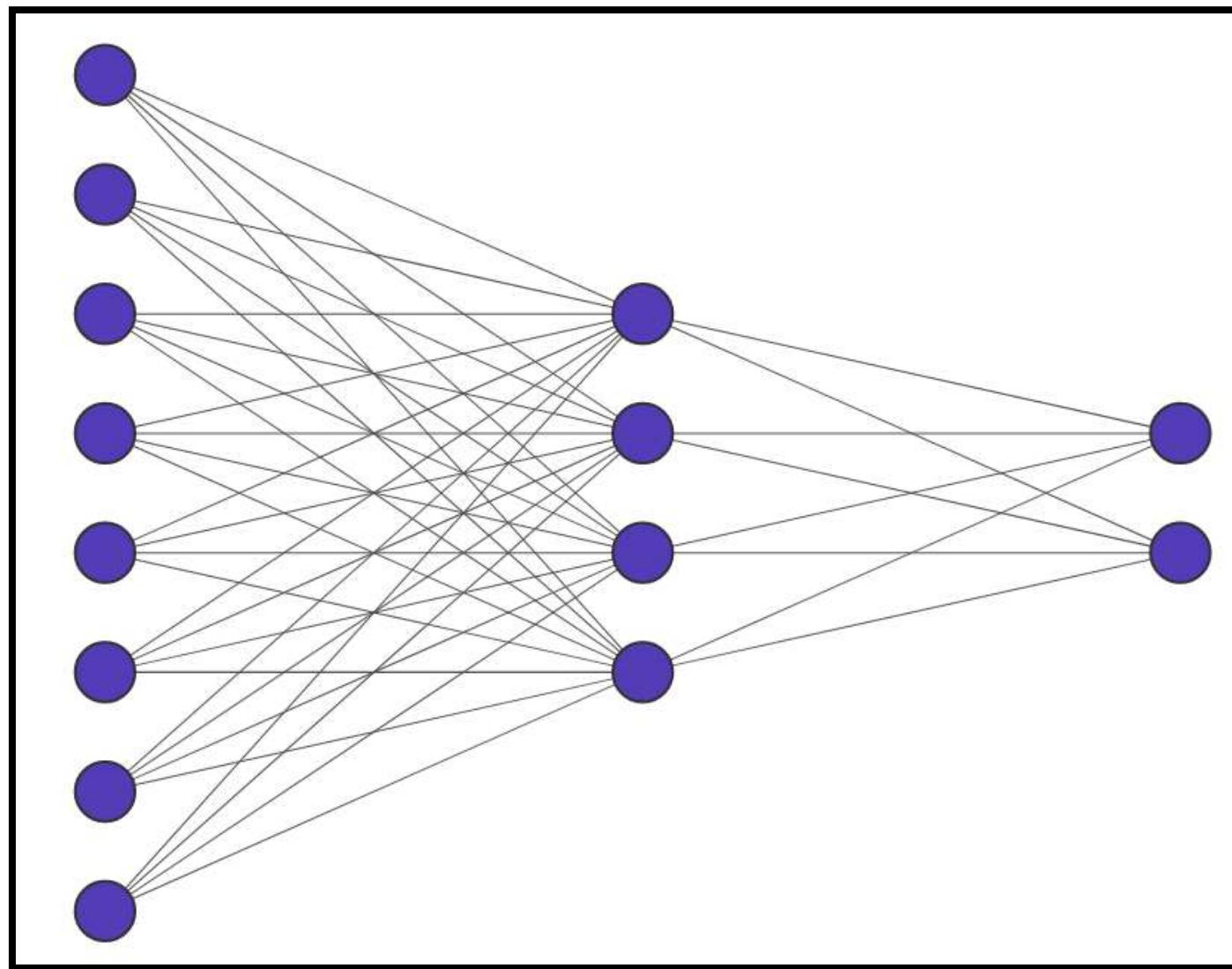
$$P(\text{spam}|X) = 3/5 \times 2/16 \times \dots$$

$$\hat{y} = \underset{k \in \{1, \dots, K\}}{\operatorname{argmax}} P(C_k) \prod_{i=1}^n P(x_i|C_k)$$

Spam detection

- Spam filtering task
- Naïve Bayes spam filtering
- Feed forward neural network for spam filtering
- Evaluation of spam filters

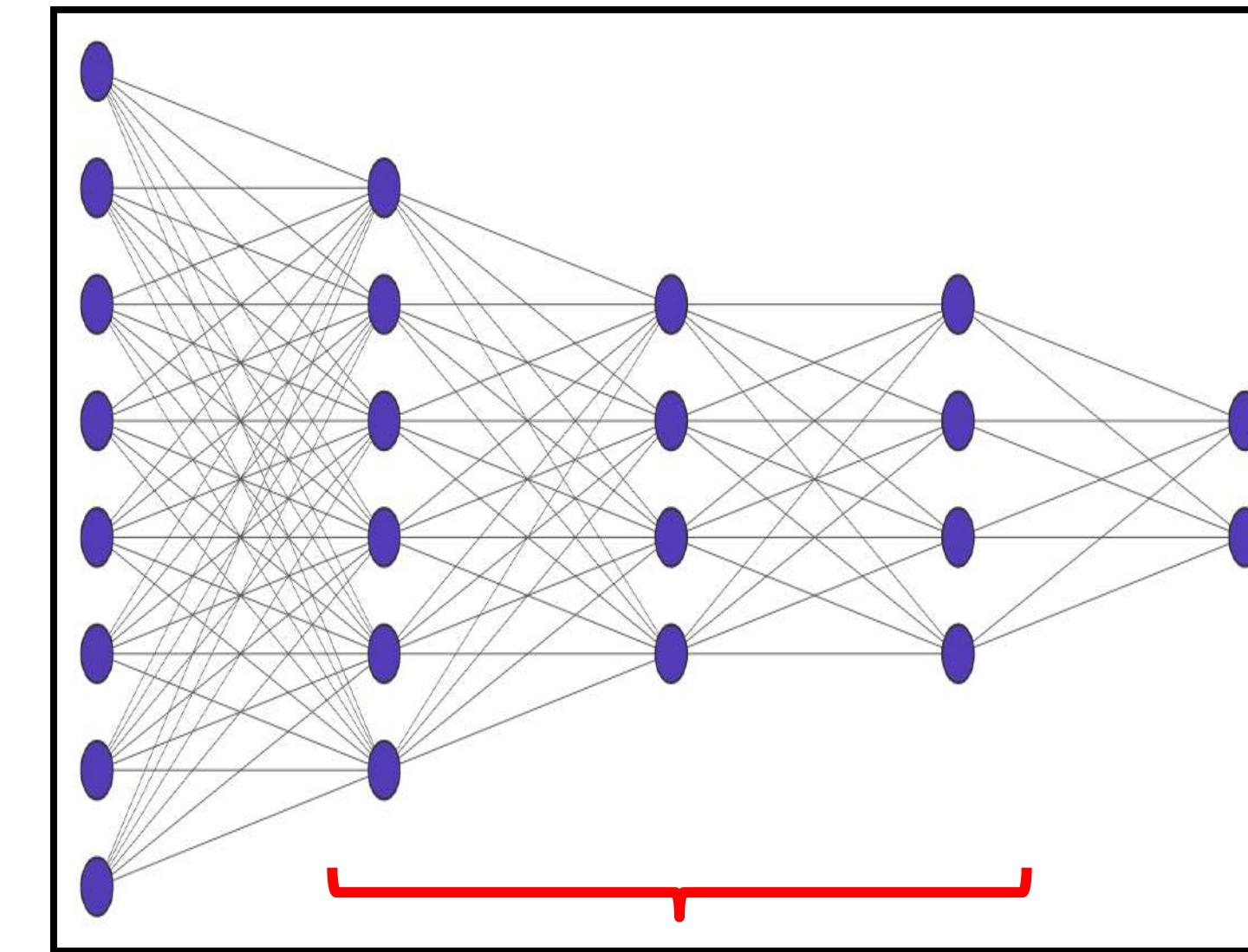
Feed forward neural network for spam filtering



Input layer

hidden layer

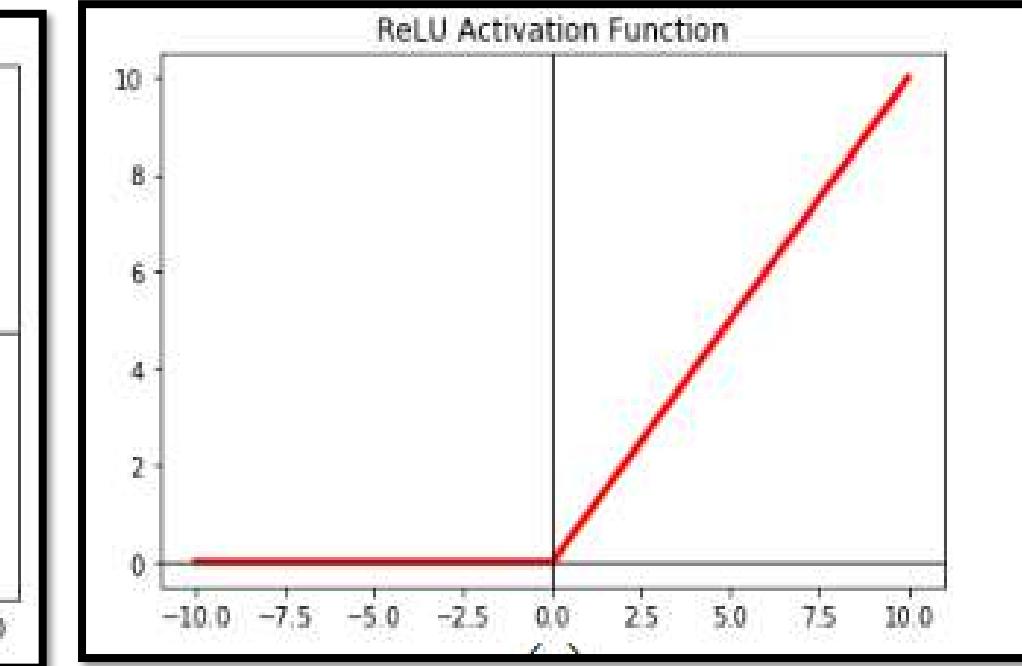
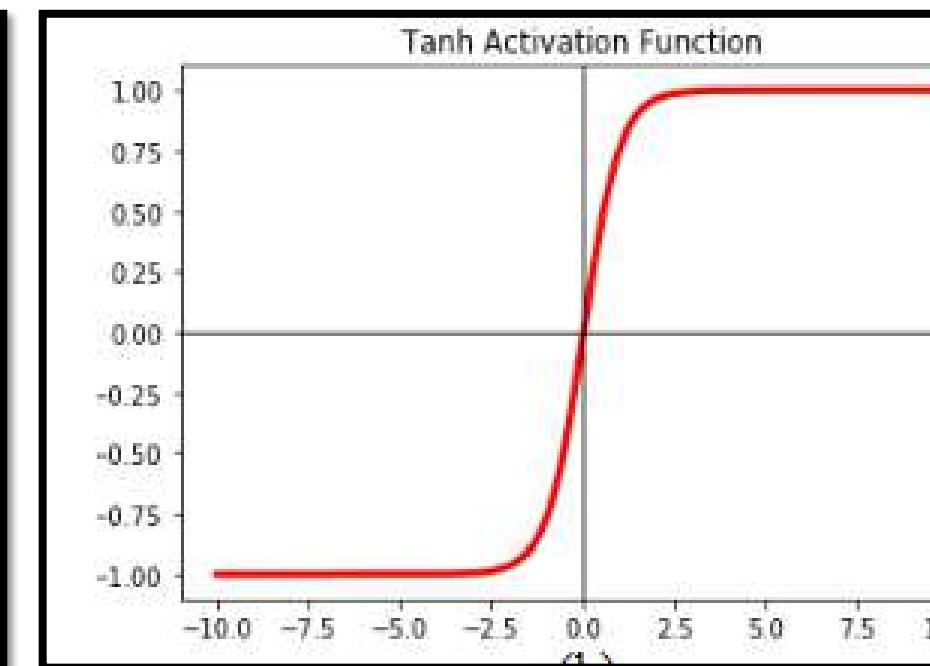
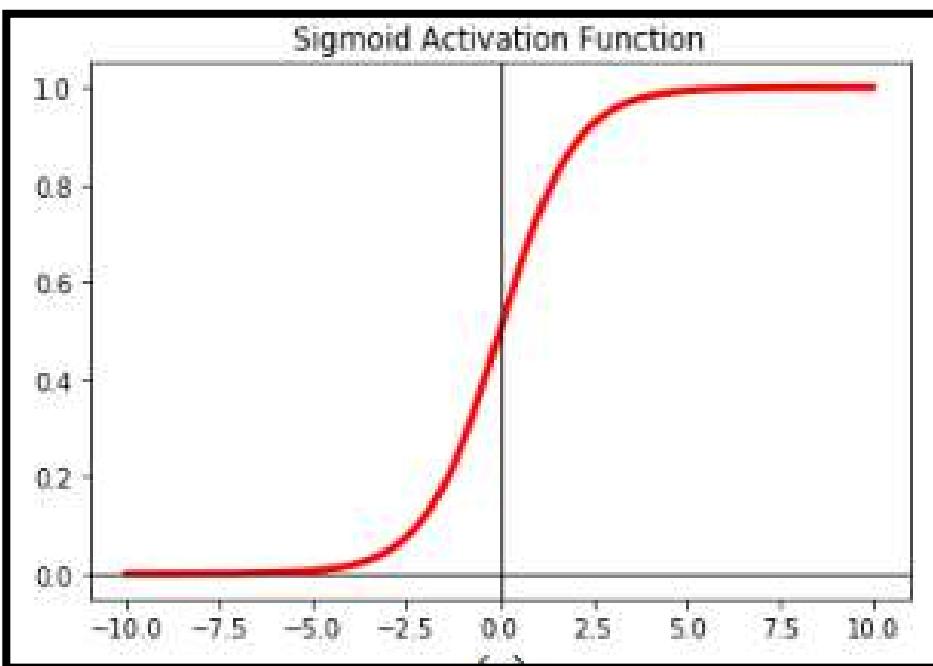
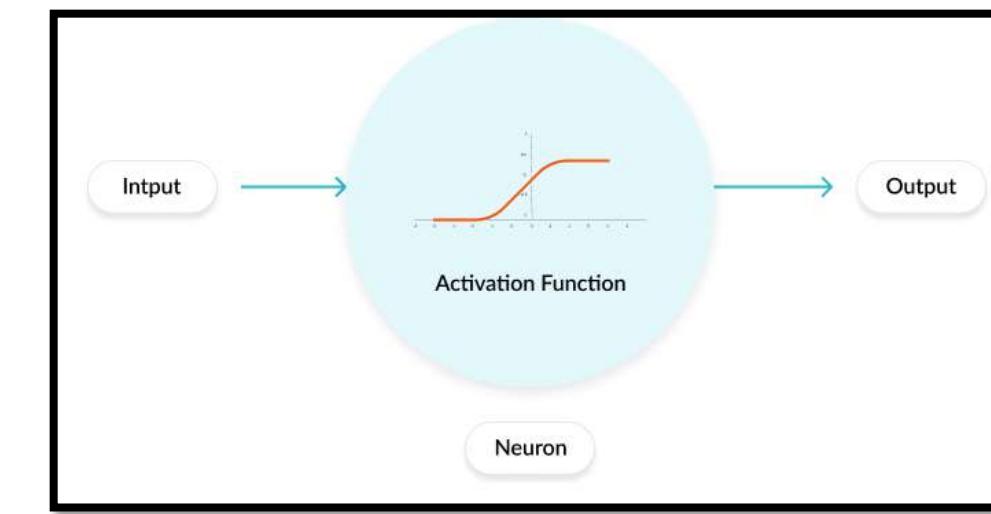
output layer



hidden layer

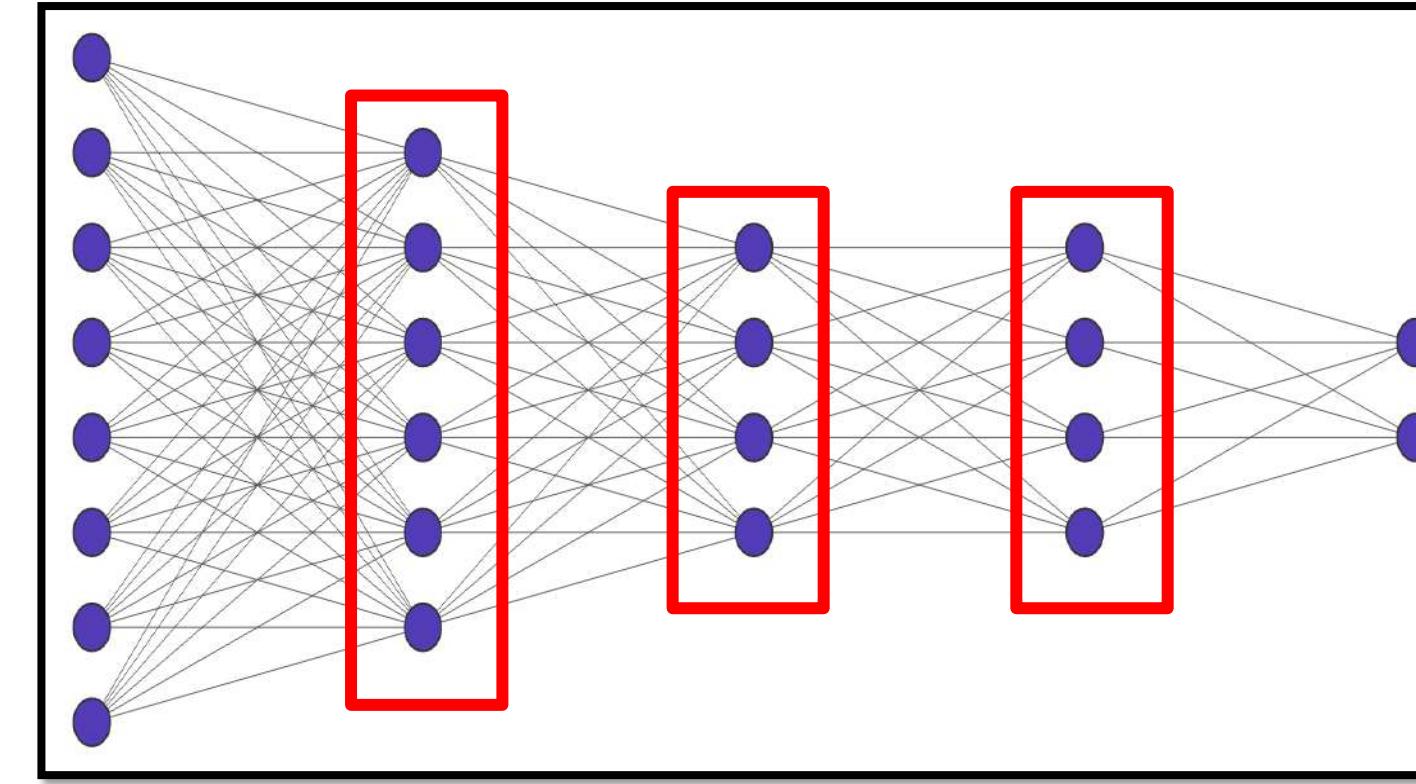
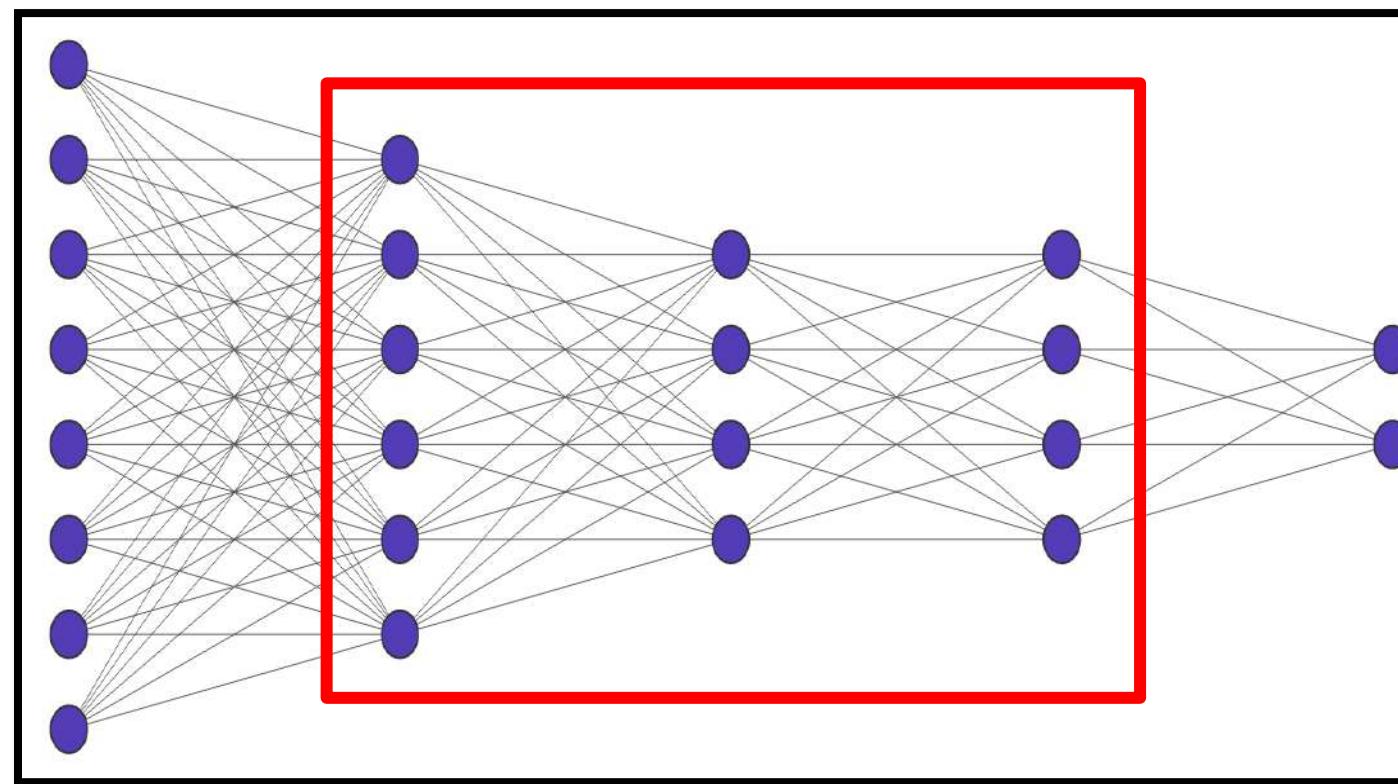
Feed forward neural network for spam filtering

- Activation function



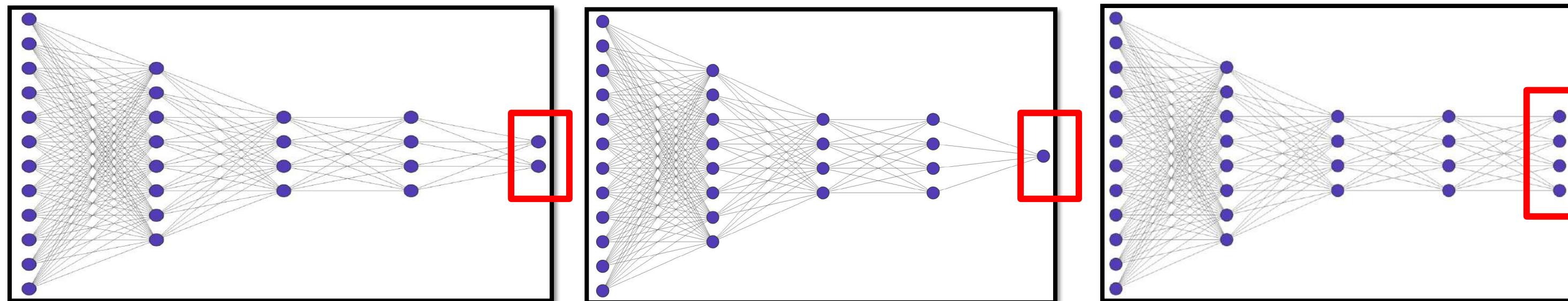
Feed forward neural network for spam filtering

- Activation function
- Loss function
- Number of hidden layers
- Number of neurons in each layer



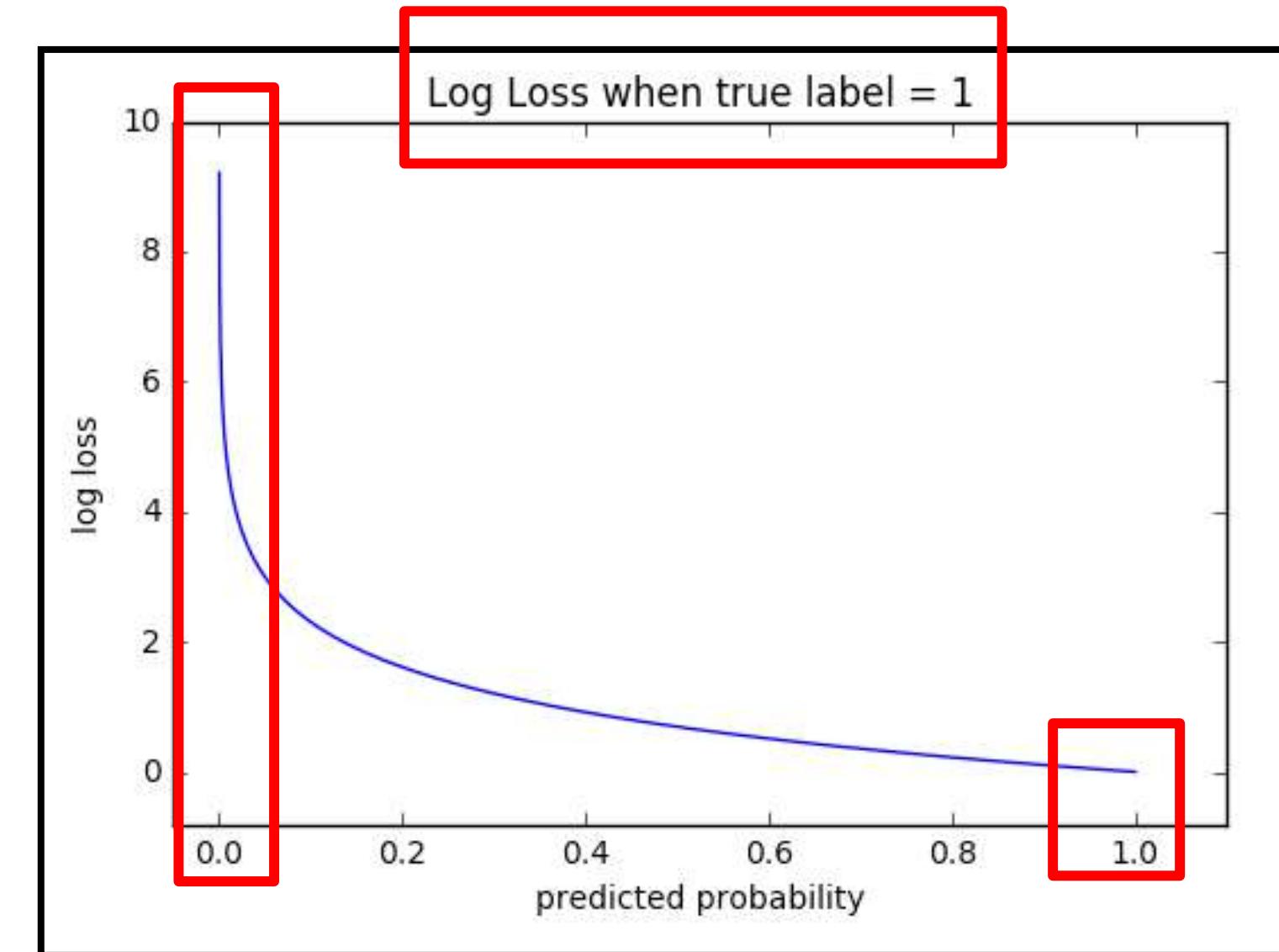
Feed forward neural network for spam filtering

- Activation function
- Loss function
- Number of hidden layers
- Number of neurons in each layer
- Output layer



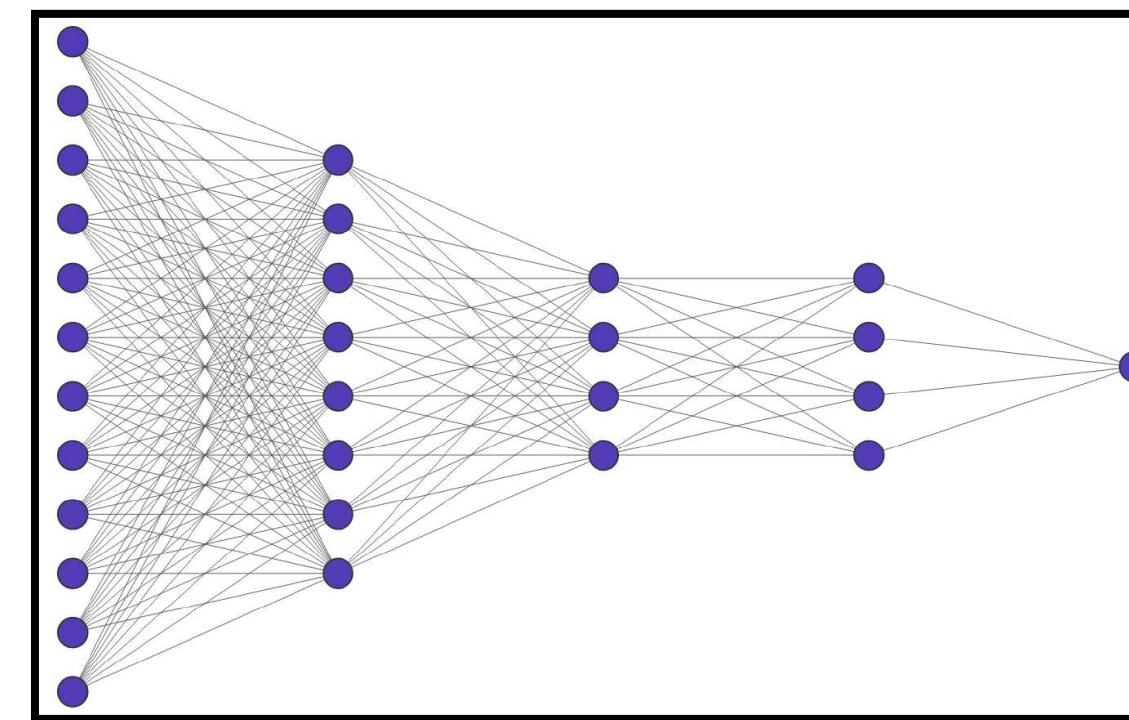
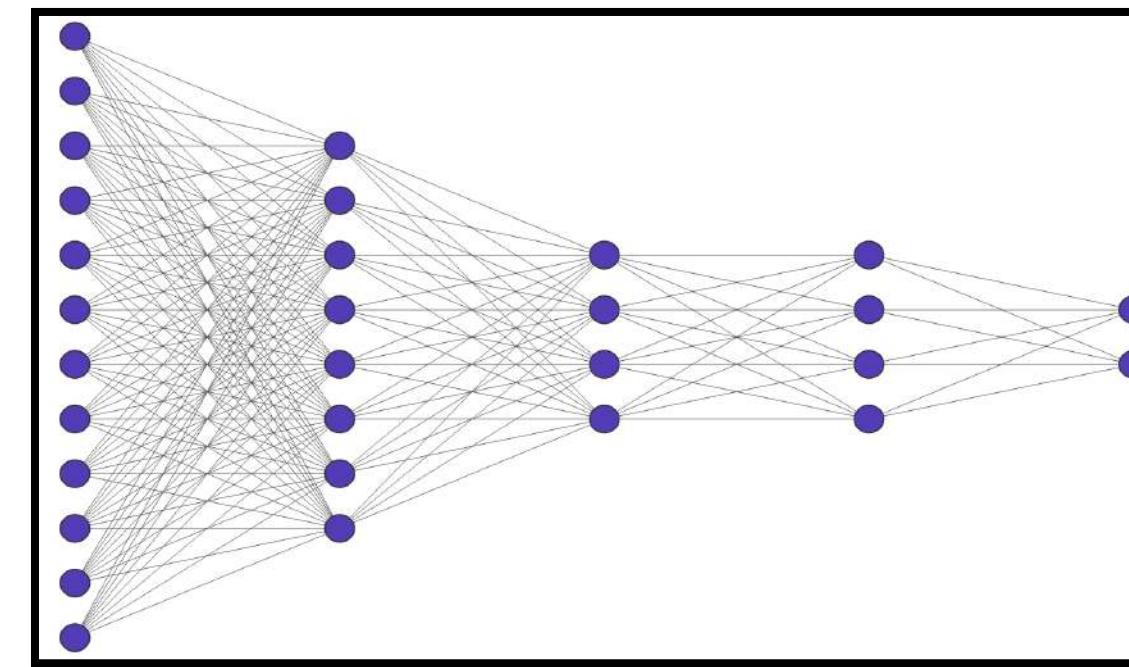
Feed forward neural network for spam filtering

- Setting for spam filtering
 - Loss function
 - Cross entropy loss (log loss)



Feed forward neural network for spam filtering

- Setting for spam filtering
 - Loss function
 - Cross entropy loss (log loss)
 - Output layer
 - Softmax (2 output)
 - Sigmoid (1 output)



Feed forward neural network for spam filtering

- Setting for spam filtering
 - Loss function
 - Cross entropy loss (log loss)
 - Output layer
 - Softmax (2 output)
 - Sigmoid (1 output)
 - Input layer

Feed forward neural network for spam filtering

- Input layer

congratulation you have won a gift card	spam
your package is out for delivery	ham
the event is postponed to the next week	ham
your PlayStation account is temporary locked	spam
congratulation you are hired	spam
congratulation you have won a PlayStation 5	?

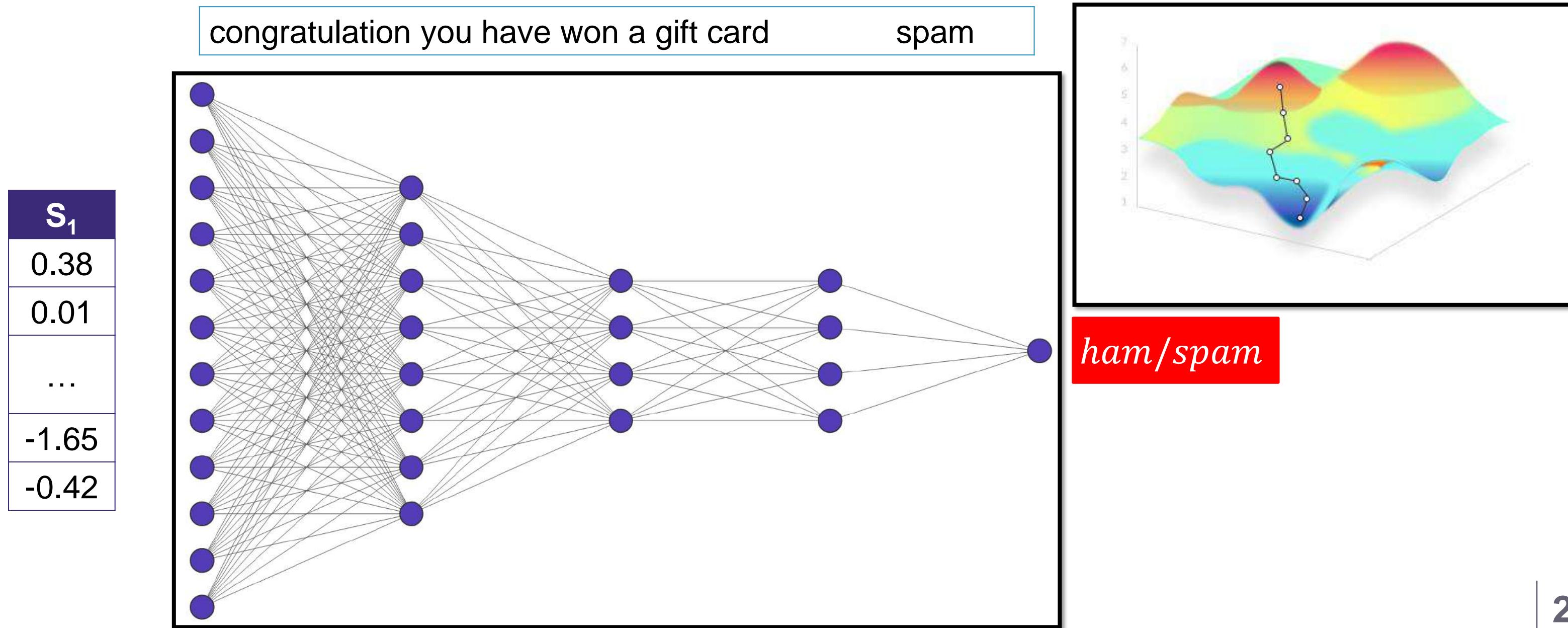
|V|

	you	account	PlayStation	is	hired	...	week	locked	congratulation
S ₁	1	0	0	0	0		0	0	1
S ₂	0	0	0	1	0		0	0	0
S ₃	0	0	0	0	0		1	0	0
S ₄	0	1	1	1	0		0	1	0
S ₅	1	0	0	0	1		0	0	1

Feed forward neural network for spam filtering

100 – 300				
congratulation	-0.37	-0.06	0.28	-0.67
You	0.68	-0.05	0.16	0.14
Have	0.53	0.05	-0.36	-0.27
Won	0.21	-0.35	-0.53	0.20
gift	-0.81	0.41	-0.58	-0.29
card	0.14	0.01	-0.62	0.47
Sum				
S ₁	0.38	0.01	-1.65	-0.42

Feed forward neural network for spam filtering



Spam detection

- Spam filtering task
- Naïve Bayes spam filtering
- Feed forward neural network for spam filtering
- Evaluation of spam filters

Evaluation of spam filters

- Confusion matrix
 - The Confusion matrix is one of the most intuitive and easiest metrics used for finding the correctness and accuracy of the model. It is used for Classification problem where the output can be of two or more types of classes

spam → positive

ham → negative

		Actual class	
		Positive	Negative
Predicted class	Positive	TP: True Positive	FP: False Positive
	Negative	FN: False Negative	TN: True Negative

Evaluation of spam filters

- Accuracy

$$Accuracy = \frac{TP + TN}{TP + FP + FN + TN}$$

		Actual class	
		Positive	Negative
Predicted class	Positive	TP: True Positive	FP: False Positive
	Negative	FN: False Negative	TN: True Negative

Evaluation of spam filters

- Precision
- Recall
- F1

$$Precision = \frac{TP}{TP + FP}$$

$$Recall = \frac{TP}{TP + FN}$$

$$F1 = \frac{2 \times (precision \times Recall)}{(precision + Recall)}$$

		Actual class	
		Positive	Negative
Predicted class	Positive	TP: True Positive	FP: False Positive
	Negative	FN: False Negative	TN: True Negative

Evaluation of spam filters

		Actual class	
		Positive	Negative
Predicted class	Positive	TP: 1	FP: 2
	Negative	FN: 2	TN: 3

Actual	Predicted
ham	ham
ham	spam
spam	spam
spam	ham
spam	ham
ham	ham
ham	ham
ham	spam

		Actual class	
		Positive	Negative
Predicted class	Positive	TP: True Positive	FP: False Positive
	Negative	FN: False Negative	TN: True Negative

Evaluation of spam filters

		Actual class	
		Positive	Negative
Predicted class	Positive	TP: 1	FP: 2
	Negative	FN: 2	TN: 3

Actual	Predicted
ham	ham
ham	spam
spam	spam
spam	ham
spam	ham
ham	ham
ham	ham
ham	spam

		Actual class	
		Positive	Negative
Predicted class	Positive	TP: True Positive	FP: False Positive
	Negative	FN: False Negative	TN: True Negative

Evaluation of spam filters

		Actual class	
		Positive	Negative
Predicted class	Positive	TP: 1	FP: 2
	Negative	FN: 2	TN: 3

Actual	Predicted
ham	ham
ham	spam
spam	spam
spam	ham
spam	ham
ham	ham
ham	ham
ham	spam

		Actual class	
		Positive	Negative
Predicted class	Positive	TP: True Positive	FP: False Positive
	Negative	FN: False Negative	TN: True Negative

Evaluation of spam filters

		Actual class	
		Positive	Negative
Predicted class	Positive	TP: 1	FP: 2
	Negative	FN: 2	TN: 3

Actual	Predicted
ham	ham
ham	spam
spam	spam
spam	ham
spam	ham
ham	ham
ham	ham
ham	spam

		Actual class	
		Positive	Negative
Predicted class	Positive	TP: True Positive	FP: False Positive
	Negative	FN: False Negative	TN: True Negative

Evaluation of spam filters

- $accuracy = \frac{4}{8} = 0.5 = 50\%$
- $precision = \frac{1}{3} = 0.33 = 33\%$
- $recall = \frac{1}{3} = 0.33 = 33\%$
- $f1 = \frac{2 \times 0.33 \times 0.33}{0.33 + 0.33} = 0.33 = 33\%$

		Actual class	
		Positive	Negative
Predicted class	Positive	TP: 1	FP: 2
	Negative	FN: 2	TN: 3

Evaluation of spam filters

		Actual class	
		Positive	Negative
Predicted class	Positive	TP: 0	FP: 0
	Negative	FN: 1	TN: 99

imbalanced data

Actual	Predicted
ham	ham
ham	ham
...	...
ham	ham
ham	ham
spam	ham
ham	ham

99% accuracy!

Accuracy is not a good metric for imbalanced data!

Evaluation of spam filters

		Actual class	
		Positive	Negative
Predicted class	Positive	TP: 0	FP: 0
	Negative	FN: 1	TN: 99

imbalanced data

Actual	Predicted
ham	ham
ham	ham
...	...
ham	ham
ham	ham
spam	ham
ham	ham

0% precision

0% recall

0% F1

Summary

- A spam filter is a program that is used to detect unsolicited and ***unwanted email*** and prevent those messages from getting to a user's inbox

$$P(C_k)P(X|C_k) = P(C_k)P(x_1|C_k)P(x_2|C_k) \dots P(x_n|C_k)$$

congratulations you have won a playstation 5

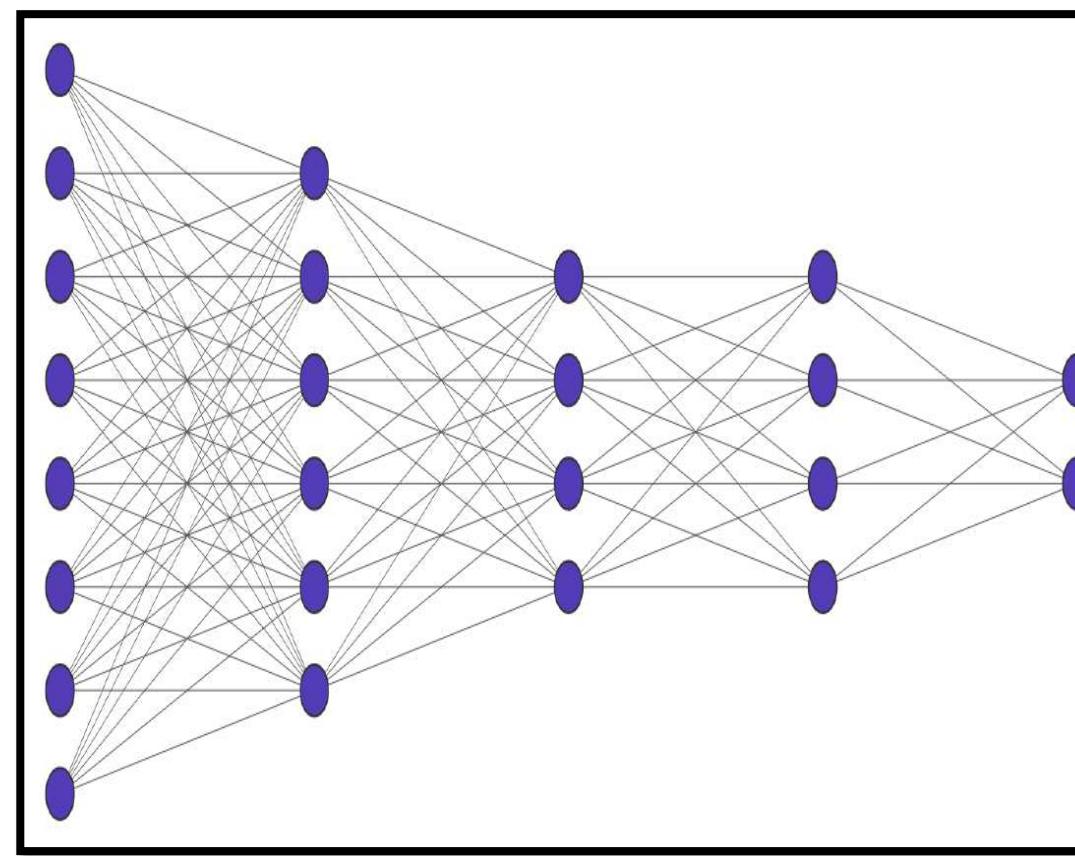
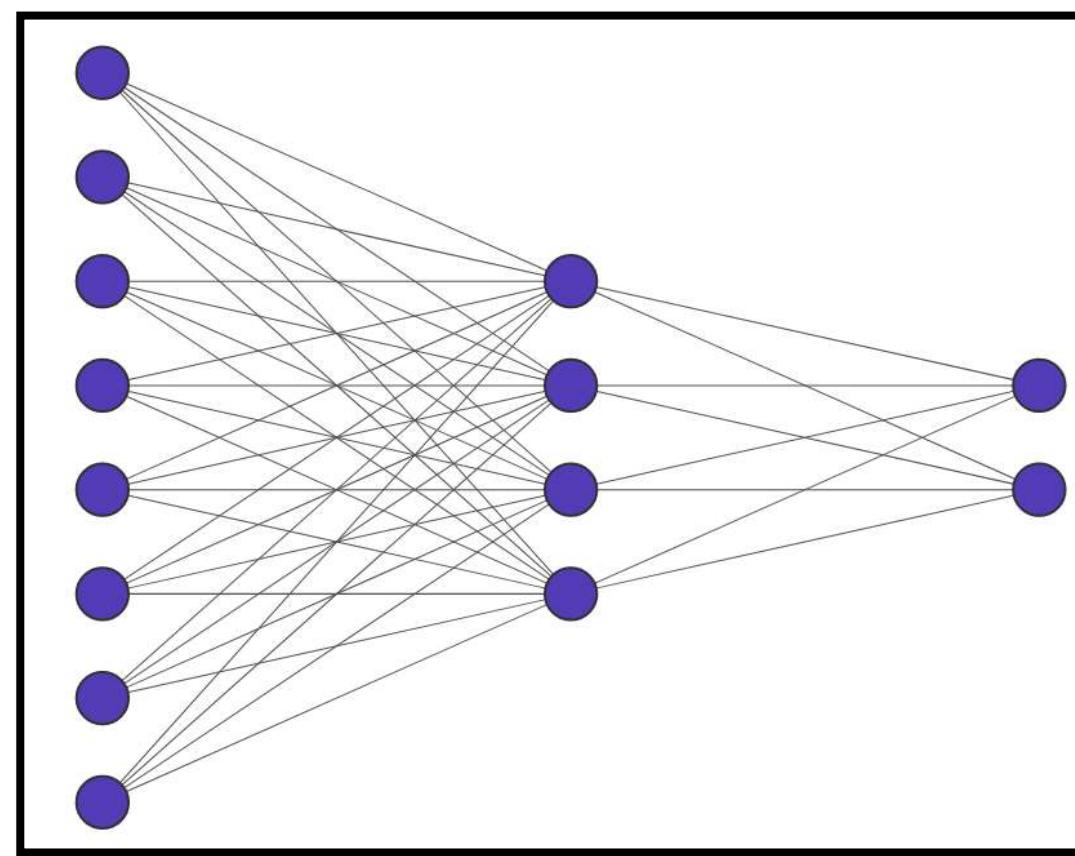
congratulation you have won gift card	spam
your PlayStation account is temporary locked	spam
congratulation you are hired	spam
your package is out delivery	ham
event is postponed next week	ham

A type	A text
ham	Hope you are having a good week. Just checking in
ham	K...give back my thanks.
ham	Am also doing in cbe only. But have to pay.
spam	complimentary 4 STAR Ibiza Holiday or £10,000 cash needs your URGENT collection. 09066364349 NOW fro...

Summary

100 – 300

			Sum	
congratulation	-0.37	-0.06	0.28	-0.67
You	0.68	-0.05	0.16	0.14
Have	0.53	0.05	-0.36	-0.27
Won	0.21	-0.35	-0.53	0.20
gift	-0.81	0.41	-0.58	-0.29
card	0.14	0.01	-0.62	0.47
S_1	0.38	0.01	-1.65	-0.42

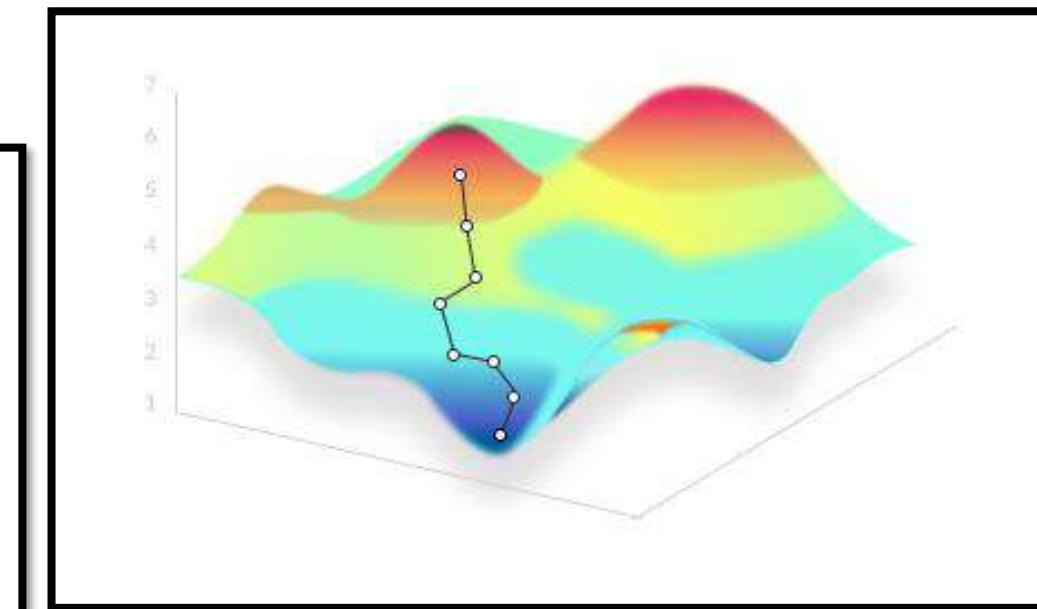
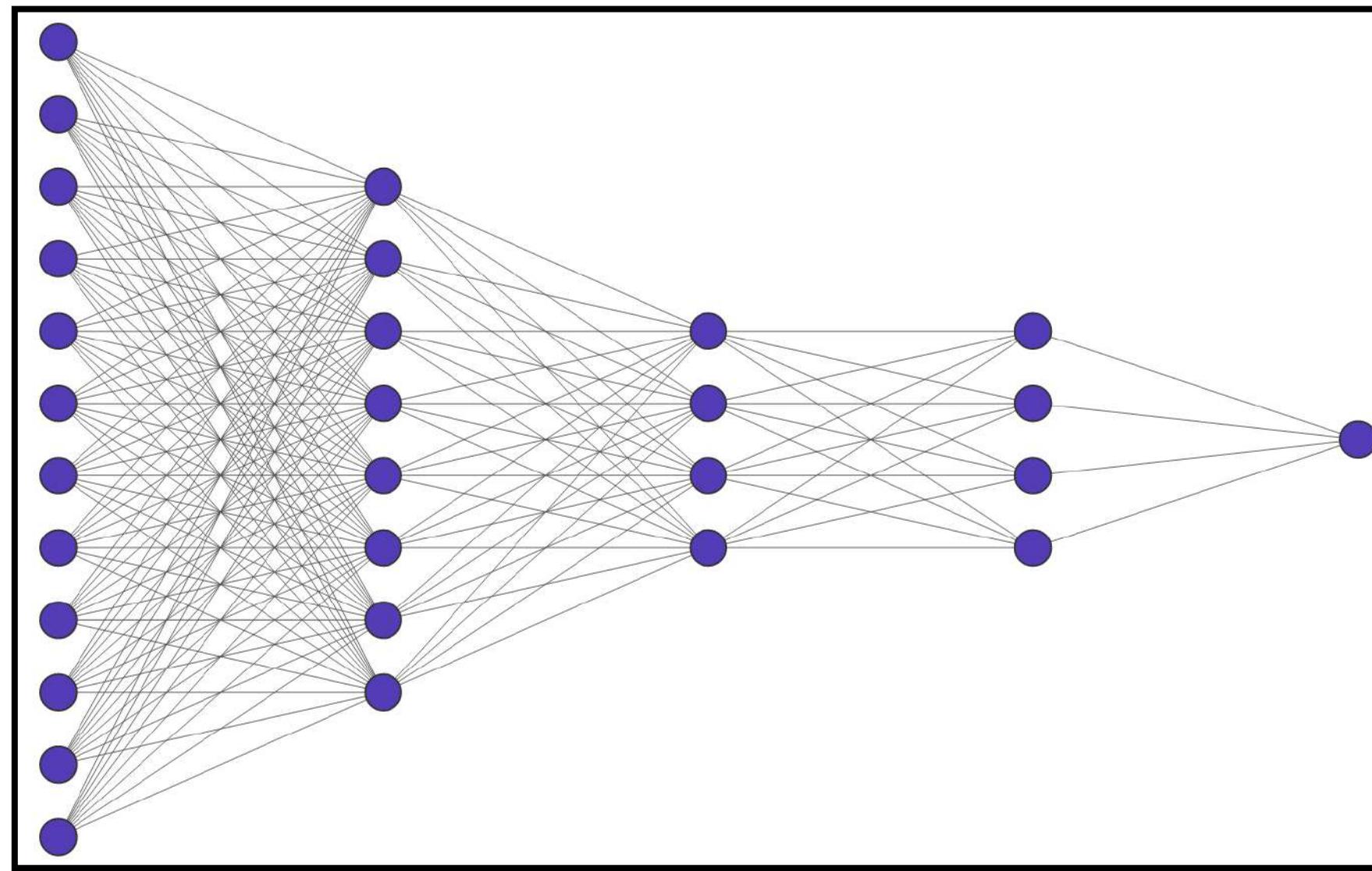


Summary

congratulation you have won a gift card

spam

s_1
0.38
0.01
...
-1.65
-0.42



ham/spam

Summary

$$Accuracy = \frac{TP + TN}{TP + FP + FN + TN}$$

$$Precision = \frac{TP}{TP + FP}$$

$$Recall = \frac{TP}{TP + FN}$$

$$F1 = \frac{2 \times (precision \times Recall)}{(precision + Recall)}$$

		Actual class	
		Positive	Negative
Predicted class	Positive	TP: True Positive	FP: False Positive
	Negative	FN: False Negative	TN: True Negative

Keyphrase Extraction

Salar Mohtaj | DFKI

Keyphrase extraction

- What is keyphrase extraction
- Why is keyphrase extraction important
- Classical keyphrase extraction methods
- Neural keyphrase extraction
- Evaluation of automatic keyphrase extraction

Keyphrase extraction

- What is keyphrase extraction
- Why is keyphrase extraction important
- Classical keyphrase extraction methods
- Neural keyphrase extraction
- Evaluation of automatic keyphrase extraction

What is keyphrase extraction

- Keyphrase extraction is the automated process of extracting the most relevant words/phrases and expressions from text
- Automatic keyphrase extraction (AKE) is the task to identify a small set of words, key phrases, keywords, or key segments from a document that can describe the meaning of the document



What is keyphrase extraction

- Due to the exponential growth of textual data and web sources, an automatic mechanism required to identify relevant information embedded within them
- It helps summarize the content of texts and recognize the main topics discussed



What is keyphrase extraction

- A **keyword** is a single word that represent the main topic of the text.
A **keyphrase** is a sequence of one or more words that are considered highly relevant



<https://monkeylearn.com>

Why is keyphrase extraction difficult

- Some documents cover different topics
- Keyphrases are not necessarily the most frequent phrases
- Sometimes the Keyphrases don't present in the document

Language-specific Models in Multilingual Topic Tracking

Leah S. Larkey, Fangfang Feng, Margaret Connell, Victor Lavrenko
Center for Intelligent Information Retrieval
Department of Computer Science
University of Massachusetts
Amherst, MA 01003
{larkey, feng, connell, lavrenko}@cs.umass.edu

ABSTRACT
Topic tracking is complicated when the stories in the stream occur in multiple languages. Typically, researchers have trained only English topic models because the training stories have been provided in English. In tracking, non-English test stories are then machine translated into English to compare them with the topic models. We propose a *native language hypothesis* stating that comparisons would be more effective in the original language of the story. We first test and support the hypothesis for story link detection. For topic tracking the hypothesis implies that it should be preferable to build separate language-specific topic models for each language in the stream. We compare different methods of incrementally building such native language topic models.

Categories and Subject Descriptors
H.3.1 [Information Storage and Retrieval]: Content Analysis and Indexing – *Indexing methods, Linguistic processing*.

General Terms: Algorithms, Experimentation.

Keywords: classification, crosslingual, Arabic, TDT, topic tracking, multilingual

Meng, Rui, Sanqiang Zhao, Shuguang Han, Daqing He, Peter Brusilovsky, and Yu Chi. "Deep keyphrase generation." arXiv preprint arXiv:1704.06879 (2017).

Keyphrase extraction

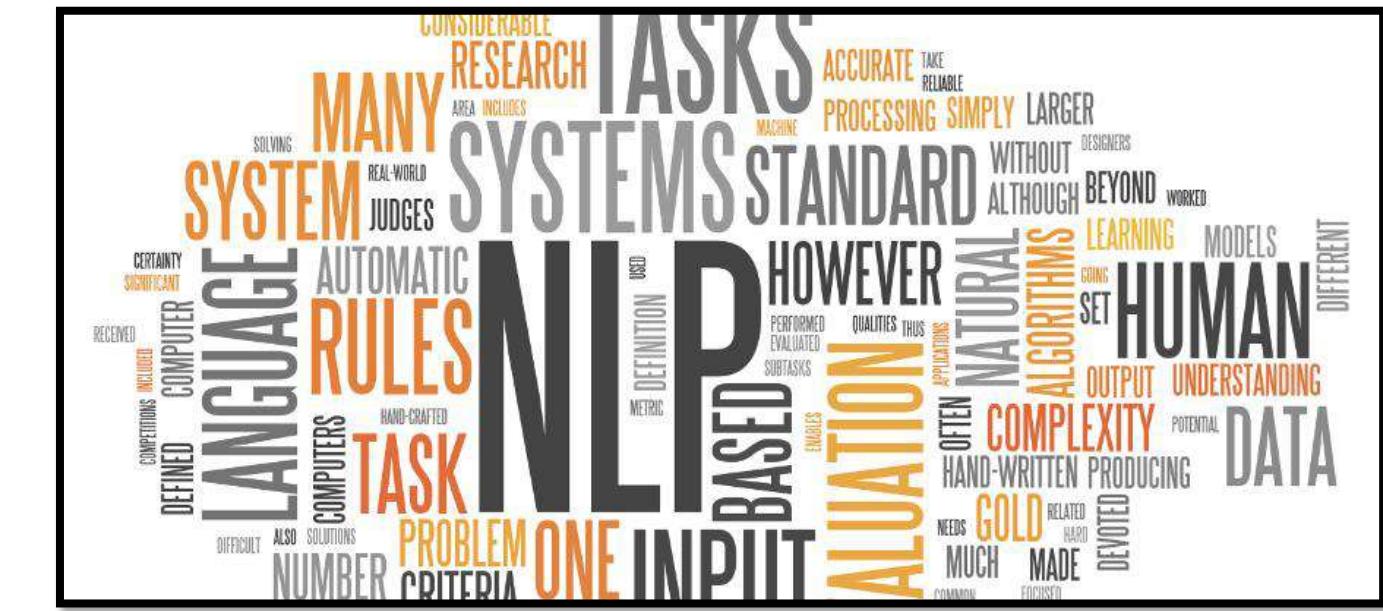
- What is keyphrase extraction
- Why is keyphrase extraction important
- Classical keyphrase extraction methods
- Neural keyphrase extraction
- Evaluation of automatic keyphrase extraction

Why is keyphrase extraction important

- Keyphrases in a document provide important information about the content of the document
- They can help users search through information more efficiently or **decide** whether to read a document
- Considering that most of the data we generate every day is unstructured, businesses need automated keyphrase extraction to help them **process** and **analyze** customer data in a more efficient manner
- Keyphrase extraction can be considered as the **core technology** of most of the text processing applications

Why is keyphrase extraction important

- Many NLP applications can take advantage of key words/phrases
 - Automatic summarization
 - Text classification
 - Text clustering
 - Automatic filtering
 - Topic detection and tracking
 - Information visualization



<http://erikburger.nl>

Keyphrase extraction

- What is keyphrase extraction
- Why is keyphrase extraction important
- Classical keyphrase extraction methods
- Neural keyphrase extraction
- Evaluation of automatic keyphrase extraction

Classical keyphrase extraction methods

- Generally, classical systems identify a set of words and phrases called ***candidates*** that could convey the topical content of a document
- Then these candidates are ***scored*** and ***ranked***
- Finally, the ***best*** ones are selected as a document's ***keyphrases***

Classical keyphrase extraction methods

- Candidate identification
- Keyphrase selection
 - Unsupervised approaches
 - Supervised models

Candidate identification

- Selecting candidate words and phrases
- Using **heuristic rules** to extract a set of phrases and words as candidate keyphrases
- The idea is to keep the number of candidates to **a minimum**
- Still keeping **high recall** and don't miss good candidates

Language-specific Models in Multilingual Topic Tracking

Leah S. Larkey, Fangfang Feng, Margaret Connell, Victor Lavrenko
Center for Intelligent Information Retrieval
Department of Computer Science
University of Massachusetts
Amherst, MA 01003

{larkey, feng, connell, lavrenko}@cs.umass.edu

ABSTRACT

Topic tracking is complicated when the stories in the stream occur in multiple languages. Typically, researchers have trained only English topic models because the training stories have been provided in English. In tracking, non-English test stories are then machine translated into English to compare them with the topic models. We propose a *native language hypothesis* stating that comparisons would be more effective in the original language of the story. We first test and support the hypothesis for story link detection. For topic tracking the hypothesis implies that it should be preferable to build separate language-specific topic models for each language in the stream. We compare different methods of incrementally building such native language topic models.

Categories and Subject Descriptors

H.3.1 [Information Storage and Retrieval]: Content Analysis and Indexing – *Indexing methods, Linguistic processing*.

General Terms

Algorithms, Experimentation.

Keywords: classification, crosslingual, Arabic, TDT, topic tracking, multilingual

tion.

All TDT tasks have at their core a comparison of two text models. In story link detection, the simplest case, the comparison is between pairs of stories, to decide whether given pairs of stories are on the same topic or not. In topic tracking, the comparison is between a story and a topic, which is often represented as a centroid of story vectors, or as a language model covering several stories.

Our focus in this research was to explore the best ways to compare stories and topics when stories are in multiple languages. We began with the hypothesis that if two stories originated in the same language, it would be best to compare them in that language, rather than translating them both into another language for comparison. This simple assertion, which we call the *native language hypothesis*, is easily tested in the TDT story link detection task.

The picture gets more complex in a task like topic tracking, which begins with a small number of training stories (in English) to define each topic. New stories from a stream must be placed into these topics. The streamed stories originate in different languages, but are also available in English translation. The translations have been performed automatically by machine translation algorithms, and are inferior to manual translations. At the beginning of the stream, native language comparisons cannot be performed be-

Candidate identification

- Typical heuristics
 - Removing stop words
 - Allowing words with certain part-of-speech tags (e.g., nouns, adjectives, verbs)
 - Using external knowledge bases like WordNet or Wikipedia as a reference source of keyphrases
 - Phrases which appear in Wikipedia article titles
 - Generating n-grams for different ranges of N
 - Extracting noun phrases based on grammatical rules

The election-year politics are annoying for many people.

Candidate Identification

We introduce a new language representation model called BERT, which stands for Bidirectional Encoder Representations from Transformers. Unlike recent language representation models, BERT is designed to pre-train deep bidirectional representations from unlabeled text by jointly conditioning on both left and right context in all layers. As a result, the pre-trained BERT model can be fine-tuned with just one additional output layer to create state-of-the-art models for a wide range of tasks, such as question answering and language inference, without substantial task-specific architecture modifications.

Candidate Identification

We introduce a new language representation model called **BERT**, which stands for **Bidirectional Encoder Representations from Transformers**. Unlike recent language representation models, BERT is designed to pre-train deep bidirectional representations from unlabeled text by jointly conditioning on both left and right context in all layers. As a result, the pre-trained BERT model can be fine-tuned with just one additional output layer to create state-of-the-art models for a wide range of tasks, such as question answering and language inference, without substantial task-specific architecture modifications.

Keyphrase selection

- In this step the idea is to select good candidates among the whole list of candidates
- A very simple approach could be weighing candidates based on frequency statistics like TF-IDF
- Best keyphrases are not necessarily the most frequent within a document
- Two different approaches
 - Unsupervised approaches
 - Supervised models

Unsupervised keyphrase selection

- The idea is to select the best keyphrases from the candidate list without relying on labeled data (training data)
 - Graph-based ranking method
 - Topic-based clustering

Graph-based ranking method

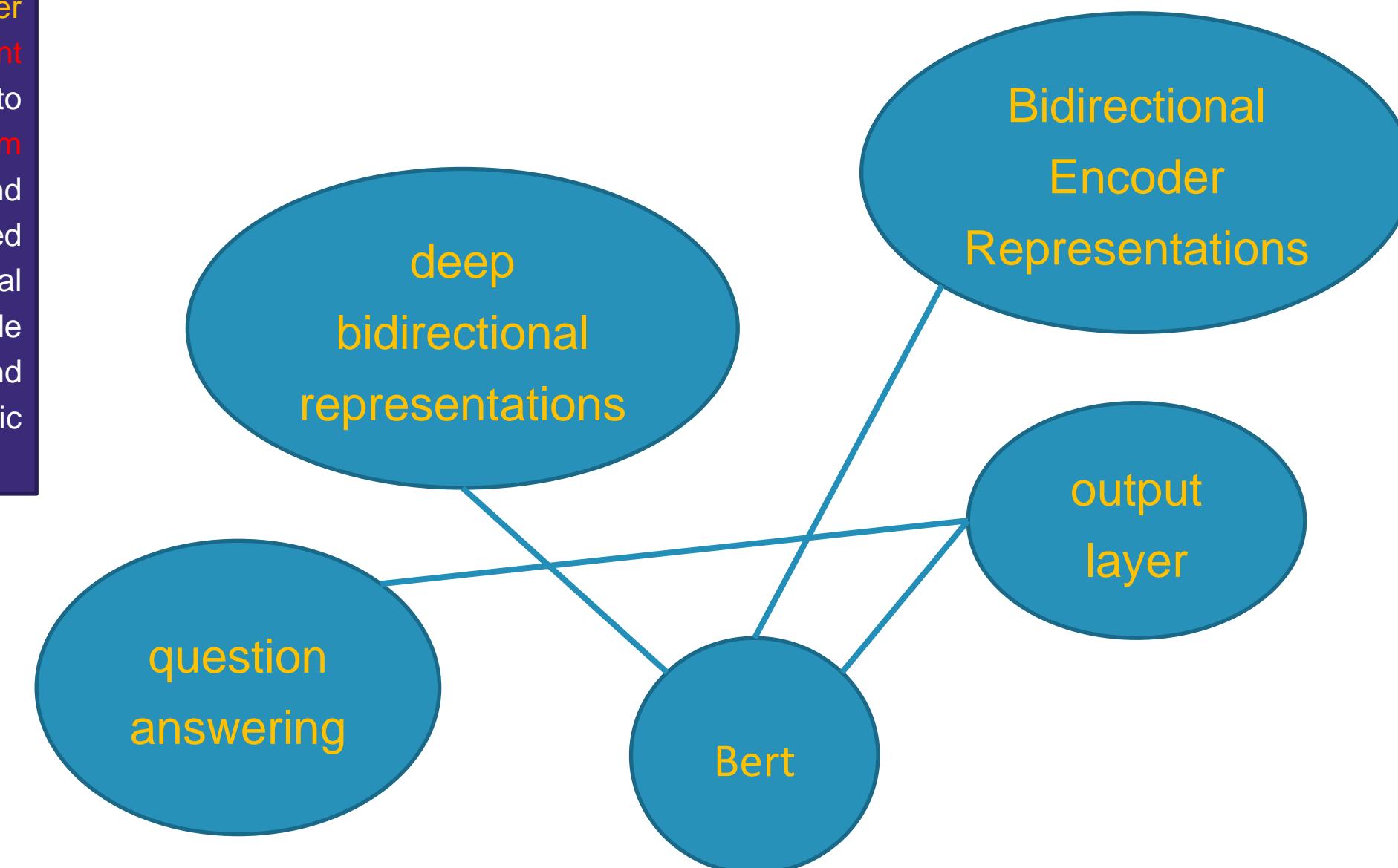
- An important candidate is related to:
 1. A large number of other candidates
 2. Candidates which are important
- A document is represented as a graph
 - Nodes are candidate keyphrases
 - Edges connect related candidates

Graph-based ranking method

We introduce a new language representation model called **BERT**, which stands for **Bidirectional Encoder Representations from Transformers**. Unlike recent language representation models, BERT is designed to pre-train deep bidirectional representations from unlabeled text by jointly conditioning on both left and right context in all layers. As a result, the pre-trained BERT model can be fine-tuned with just one additional output layer to create state-of-the-art models for a wide range of tasks, such as question answering and language inference, without substantial task-specific architecture modifications.

Graph-based ranking method

We introduce a new language representation model called **BERT**, which stands for **Bidirectional Encoder Representations** from Transformers. Unlike recent language representation models, BERT is designed to pre-train deep bidirectional representations from unlabeled text by jointly conditioning on both left and right context in all layers. As a result, the pre-trained **BERT** model can be fine-tuned with just one additional output layer to create state-of-the-art models for a wide range of tasks, such as question answering and language inference, without substantial task-specific architecture modifications.

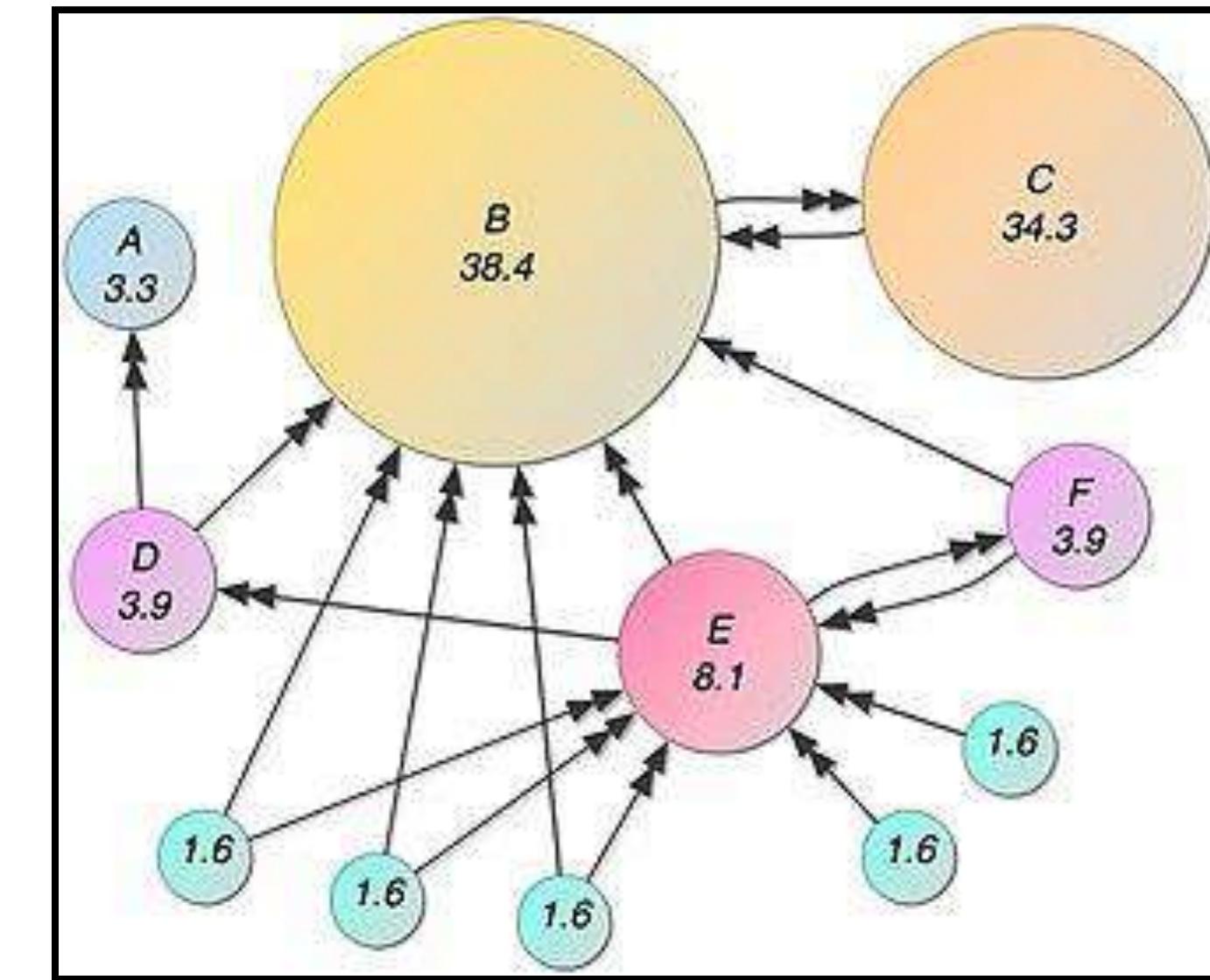


Graph-based ranking method

- An important candidate is related to:
 1. A large number of other candidates
 2. Candidates which are important
- A document is represented as a graph
 - Nodes are candidate keyphrases
 - Edges connect related candidates
 - Then, a graph-based ranking algorithm, such as PageRank, is run over the graph
 - The highest-scoring terms are keyphrases

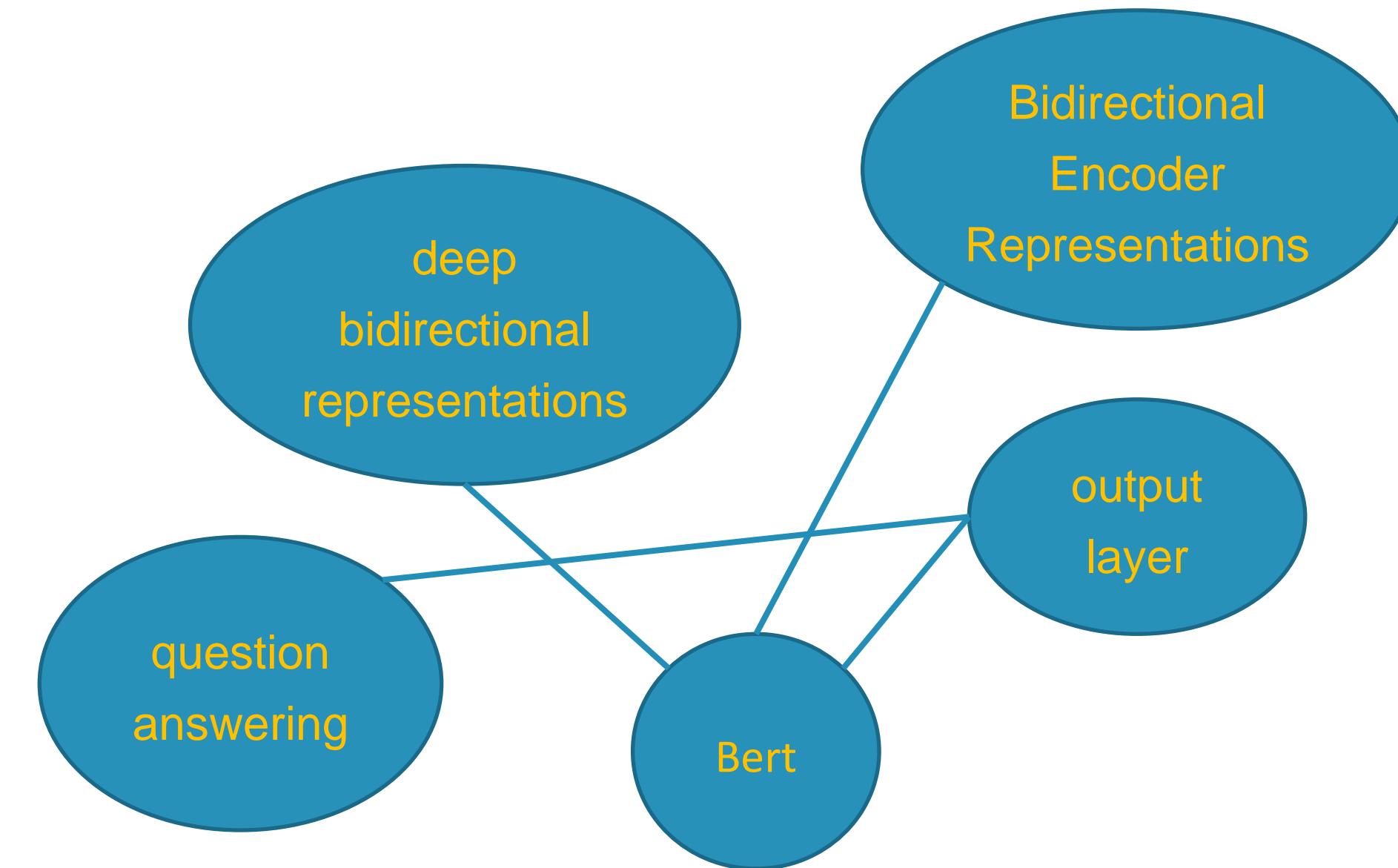
Graph-based ranking method

- PageRank
 - PageRank (PR) is an algorithm used by Google search to rank web pages in their search engine results
 - PageRank is a way of measuring the importance of website pages



Graph-based ranking method

- TextRank



Topic-based clustering

- A document could cover different topics (e.g., sport, finance, ...)
- In graph-based methods, all the keyphrases could be selected from the same topic
- Here the idea is to grouping the candidate keyphrases in a document into topics
 1. A keyphrase should ideally be relevant to one or more main topic(s) discussed in a document
 2. The extracted keyphrases should be comprehensive in the sense that they should cover all the main topics in a document

Topic-based clustering

We introduce a new language representation model called **BERT**, which stands for **Bidirectional Encoder Representations from Transformers**. Unlike recent language representation models, BERT is designed to pre-train deep bidirectional representations from unlabeled text by jointly conditioning on both left and right context in all layers. As a result, the pre-trained BERT model can be fine-tuned with just one additional output layer to create state-of-the-art models for a wide range of tasks, such as question answering and language inference, without substantial task-specific architecture modifications.

Topic-based clustering

We introduce a new language representation model called **BERT**, which stands for **Bidirectional Encoder Representations** from Transformers. Unlike recent language representation models, BERT is designed to pre-train **deep bidirectional representations** from **unlabeled** text by jointly conditioning on both left and right context in all layers. As a result, the pre-trained **BERT** model can be fine-tuned with just one additional **output layer** to create state-of-the-art models for a wide range of tasks, such as **question answering** and **language inference**, without substantial task-specific **architecture modifications**.

Topic #1	Topic #2
BERT Bidirectional Encoder Representations deep bidirectional representations output layer	question answering language inference architecture modifications

Supervised keyphrase selection

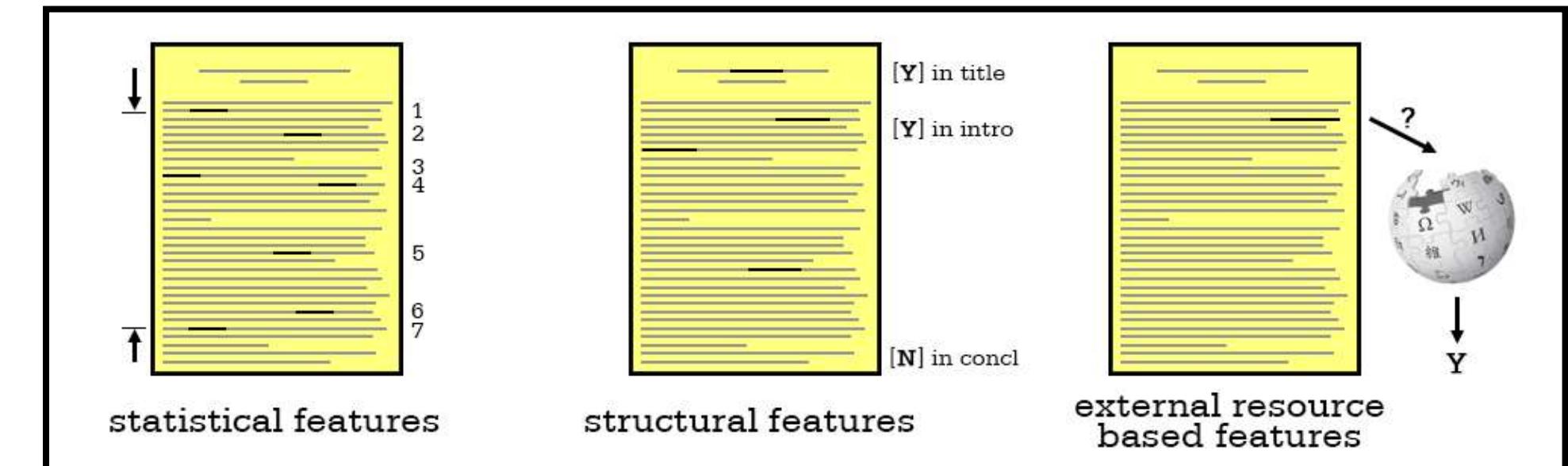
- The idea is to select the best keyphrases from the candidate list using labeled data (training data) for train a model
- Task reformulation
 - Binary classification
 - Ranking problem

Supervised keyphrase selection

- Binary classification
 - Some fraction of candidates are classified as keyphrases and the rest as non-keyphrases
 - Train a classifier (Naive Bayes, SVM, ...)
 - Label candidate keyphrases as True/False
- Ranking problem
 - We can also train a model to rank the candidate keyphrases instead of labeling them
 - And then choosing top N candidates from the ranked list

Supervised keyphrase selection

- Common features to train a model
 - Phrase length (number of constituent words)
 - Phrase position (normalized position within a document)
 - Document's structural features (titles, abstracts, intros and conclusions, ...)
 - A candidate is more likely to be a keyphrase if it appears in notable sections
 - Phrase commonness
 - Compares a candidate's frequency in a document with respect to its frequency in external corpora



<https://bdewilde.github.io/>

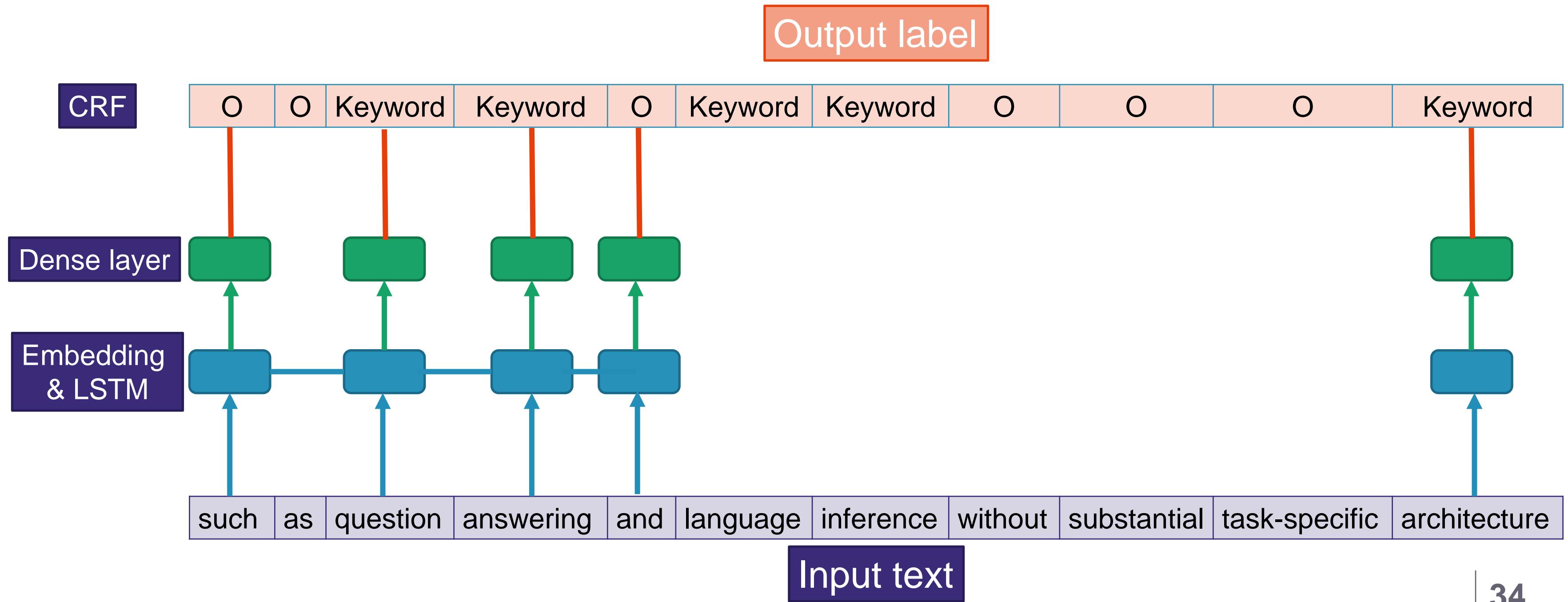
Keyphrase extraction

- What is keyphrase extraction
- Why is keyphrase extraction important
- Classical keyphrase extraction methods
- **Neural keyphrase extraction**
- Evaluation of automatic keyphrase extraction

Neural keyphrase extraction

- Feeding the input text to a neural network
- End to end approach
 - A machine learning model can directly convert an input data into an output prediction bypassing the intermediate steps that usually occur in a traditional pipeline
- Two common task formulations
 - Keyphrase extraction as sequence labeling
 - Keyphrase generation with sequence to sequence models

Neural keyphrase extraction



Deep keyphrase extraction

Language-specific Models in Multilingual Topic Tracking

ABSTRACT

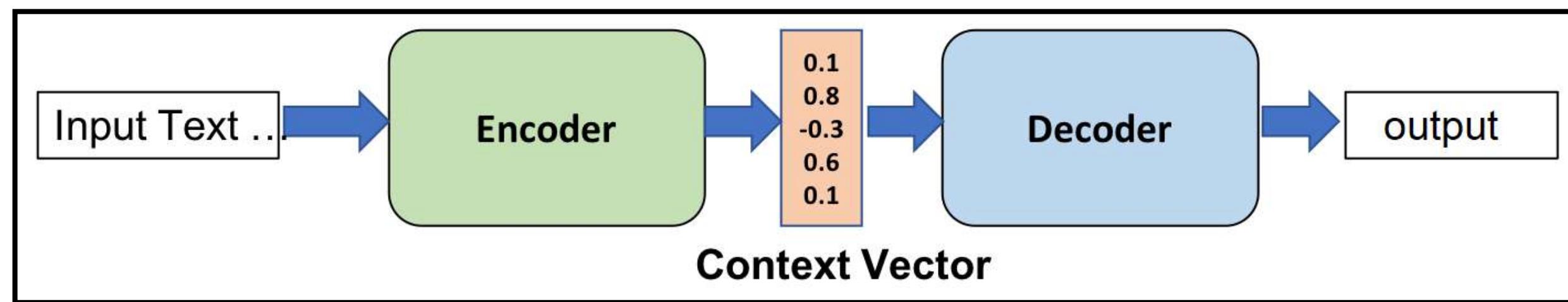
Topic tracking is complicated when the stories in the stream occur in multiple languages. Typically, researchers have trained only English topic models because the training stories have been provided in English. In tracking, non-English test stories are then machine translated into English to compare them with the topic models. We propose a *native language hypothesis* stating that comparisons would be more effective in the original language of the story. We first test and support the hypothesis for story link detection. For topic tracking the hypothesis implies that it should be preferable to build separate language-specific topic models for each language in the stream. We compare different methods of incrementally building such native language topic models.

Keywords: classification, crosslingual, Arabic, TDT, topic
tracking, multilingual

Meng, Rui, Sanqiang Zhao, Shuguang Han, Daqing He, Peter Brusilovsky, and Yu Chi. "Deep keyphrase generation." arXiv preprint arXiv:1704.06879 (2017).

Neural keyphrase extraction

- From keyphrase extraction to generation
- sequence to sequence models



<https://medium.com/nerd-for-tech>

Keyphrase extraction

- What is keyphrase extraction
- Why is keyphrase extraction important
- Classical keyphrase extraction methods
- Neural keyphrase extraction
- Evaluation of automatic keyphrase extraction

Evaluation of automatic keyphrase extraction

- Keyphrase extraction
 - As a classification task
 - As a ranking task

Evaluation of automatic keyphrase extraction

- Keyphrase extraction as a classification task

$$Precision = \frac{TP}{TP + FP}$$

$$Recall = \frac{TP}{TP + FN}$$

$$F1 = \frac{2 \times (precision \times Recall)}{(precision + Recall)}$$

		Actual class	
		Positive	Negative
Predicted class	Positive	TP: True Positive	FP: False Positive
	Negative	FN: False Negative	TN: True Negative

Evaluation of automatic keyphrase extraction

		Actual class	
		Positive	Negative
Predicted class	Positive	TP: 6	FP: 4
	Negative	FN: 4	TN: -

Actual	Predicted
Deep learning	Confusion matrix
NLP	Train
Train	Model
Confusion matrix	Algorithm
Data	Data
Model	Validation
Machine learning	Test
Supervised	Classification
Clustering	Feed-forward
Classification	Supervised

Evaluation of automatic keyphrase extraction

- Keyphrase extraction as a classification task
 - Exact match is an overly strict condition, considering a predicted keyphrase incorrect even if it is a variant of the actual keyphrases
 - Confusion matrix → Confusion matrices
 - Neural network → neural net
- Common metrics from machine translation and text summarization reward a partial matches
- Same metrics can be used for the task of keyphrase extraction
 - BLEU, METEOR, and ROUGE

Evaluation of automatic keyphrase extraction

- Keyphrase extraction as a ranking task

$$Precision@k = \frac{TP@k}{TP@k + FP@k}$$

Evaluation of automatic keyphrase extraction

Actual	Predicted	Precision@k
Deep learning	Confusion matrix	100%
NLP		
Train		
Confusion matrix		
Data		
Model		
Machine learning		
Supervised		
Clustering		
Classification		

Summary

- Keyphrase extraction is the automated process of extracting the most relevant words/phrases and expressions from text

Language-specific Models in Multilingual Topic Tracking

Leah S. Larkey, Fangfang Feng, Margaret Connell, Victor Lavrenko
Center for Intelligent Information Retrieval
Department of Computer Science
University of Massachusetts
Amherst, MA 01003
{larkey, feng, connell, lavrenko}@cs.umass.edu

ABSTRACT
Topic tracking is complicated when the stories in the stream occur in multiple languages. Typically, researchers have trained only English topic models because the training stories have been provided in English. In tracking, non-English test stories are then machine translated into English to compare them with the topic models. We propose a *native language hypothesis* stating that comparisons would be more effective in the original language of the story. We first test and support the hypothesis for story link detection. For topic tracking the hypothesis implies that it should be preferable to build separate language-specific topic models for each language in the stream. We compare different methods of incrementally building such native language topic models.

Categories and Subject Descriptors
H.3.1 [Information Storage and Retrieval]: Content Analysis and Indexing – *Indexing methods, Linguistic processing*.

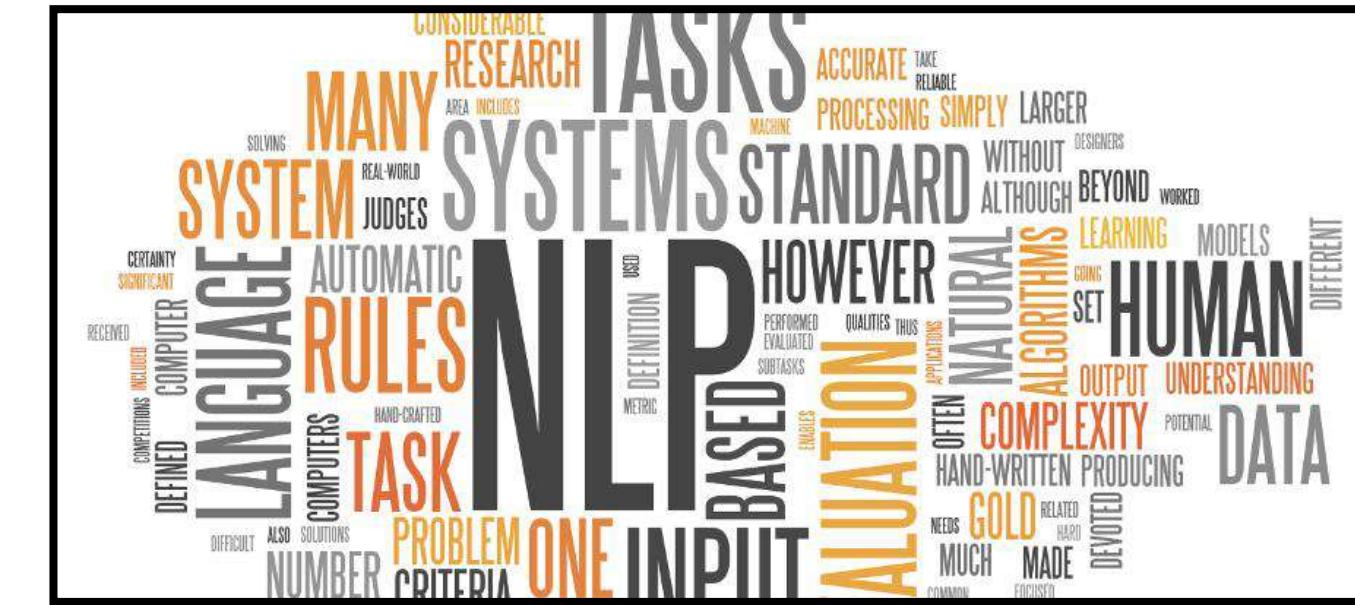
General Terms: Algorithms, Experimentation.

Keywords: classification, crosslingual, Arabic, TDT, topic tracking, multilingual



Summary

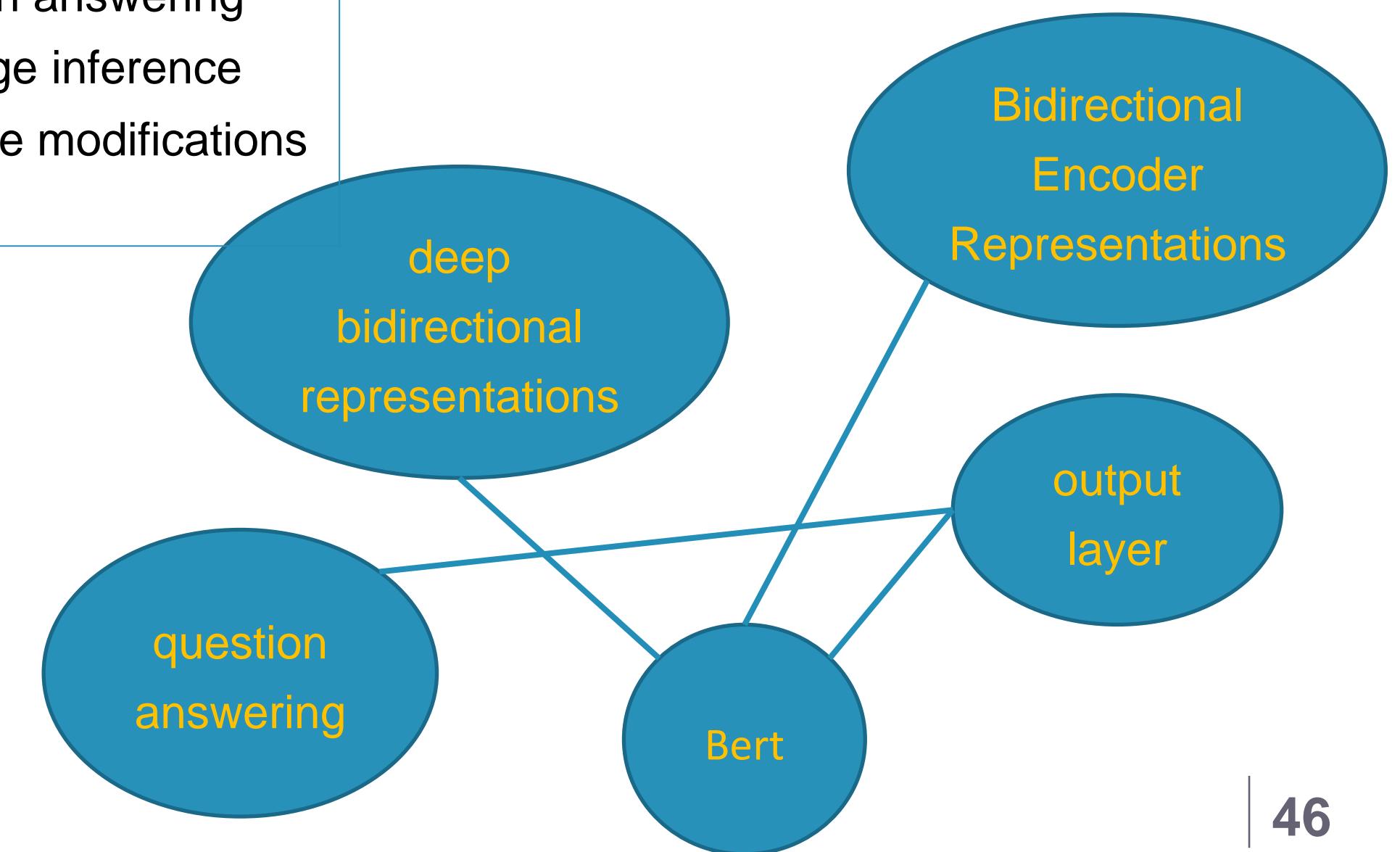
- Main applications
 - Automatic summarization
 - Text classification
 - Information visualization
 - ...
 - Classical keyphrase extraction
 - Candidate identification
 - Keyphrase selection
 - Unsupervised approaches
 - Supervised models



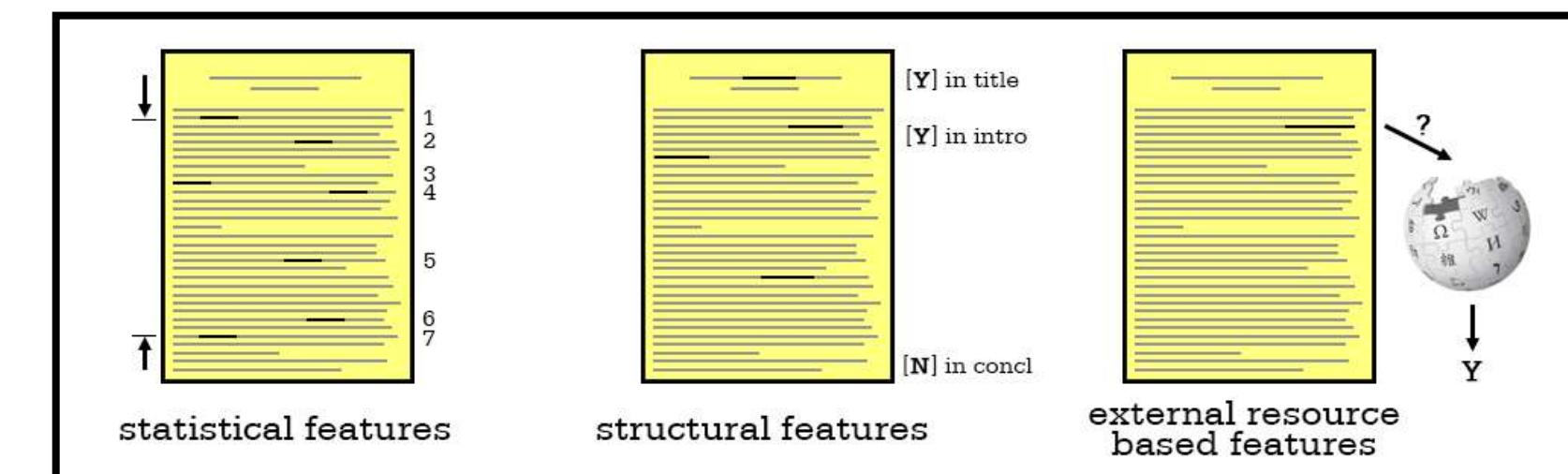
<http://erikburger.net>

Summary

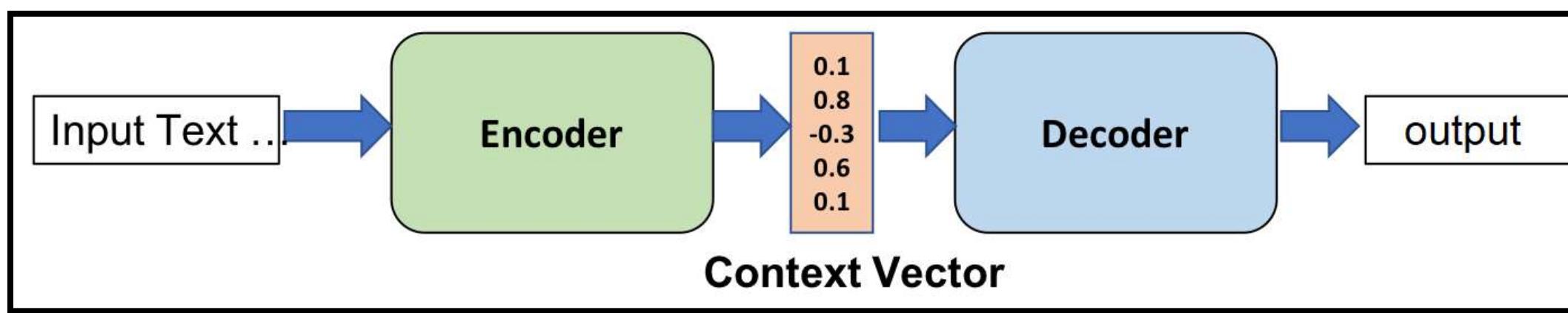
Topic #1	Topic #2
BERT	
Bidirectional Encoder Representations	question answering
deep bidirectional representations	language inference
output layer	architecture modifications



Summary



<https://bdewilde.github.io/>



Summary

		Actual class	
		Positive	Negative
Predicted class	Positive	TP: 6	FP: 4
	Negative	FN: 4	TN: -

Actual	Predicted	Precision@k
Deep learning	Confusion matrix	100%
NLP	Train	100%
Train	Model	100%
Confusion matrix	Algorithm	75%
Data	Data	80%
Model	Validation	66%
Machine learning	Test	57%
Supervised	Classification	62%
Clustering	Feed-forward	55%
Classification	Supervised	60%

Information Extraction

Salar Mohtaj | DFKI

Information Extraction

- What is information extraction
- Named entity recognition
- Named entity recognition approaches
- NER evaluation metrics

Information Extraction

- What is information extraction
- Named entity recognition
- Named entity recognition approaches
- NER evaluation metrics

What is information Extraction

- Information extraction (IE) is the task of automatically extracting structured information from unstructured and/or semi-structured text
- It's the task of finding and understanding limited relevant parts of texts
- Gather information from many pieces of text
- Produce a structured representation of relevant information
- Relations (in the database sense)

What is information Extraction

- Goals
 - Clear factual information which is helpful for
 - Answer questions
 - Analytics
 - Organize and present information
 - Info boxes in Wikipedia

What is information Extraction

roots book

All Images Shopping Videos Books More Settings Tools

About 764,000,000 results (0.88 seconds)

https://en.wikipedia.org/wiki/Roots:_The_Saga_of_an_American_Family

Roots: The Saga of an American Family - Wikipedia

Roots: The Saga of an American Family is a 1976 novel written by Alex Haley. It tells the story of Kunta Kinte, an 18th-century African, captured as an adolescent, sold into slavery in Africa, transported to North America; following his life and the lives of his descendants in the United States down to Haley.

LC Class: E185.97.H24 A33 Publisher: Doubleday
Author: Alex Haley Publication date: August 17, 1976

Plot · Characters in Roots · Reception · Historical accuracy

People also ask

What happened to Alex Haley?
Was Alex Haley related to Kunta Kinte?
Is roots a true story?
How many sons did Chicken George have?

<https://www.amazon.com/Roots-American-Family-Alex-Haley/dp/0380001210>

Roots: The Saga of an American Family: Haley, Alex ...

Based off of the bestselling author's family history, this novel tells the story of Kunta Kinte, who is sold into slavery in the United States where he and his ...

https://www.goodreads.com/book/show/1100000.Roots:_The_Saga_of_an_American_Family_by_Alex_Haley

Roots: The Saga of an American Family by Alex Haley

Roots: The Saga of an American Family is a novel written by Alex Haley and first published in 1976. Roots tells the story of Kunta Kinte—a young man taken from the Gambia when he was seventeen and sold as a slave—and seven generations of his descendants in the United States.

★★★★★ Rating: 4,4 · 150,712 votes

The screenshot shows a search results page for 'roots book'. At the top, there are filters for All, Images, Shopping, Videos, Books, More, Settings, and Tools. Below the filters, it says 'About 764,000,000 results (0.88 seconds)'. There are three main result snippets: 1) A link to the Wikipedia page for 'Roots: The Saga of an American Family' with a brief summary and details about the author (Alex Haley), publication date (August 17, 1976), and publisher (Doubleday). 2) A link to the Amazon product page for the book, which includes a snippet of the plot and the number of sons of Chicken George. 3) A link to the Goodreads page for the book, which includes a rating of 4,4 and 150,712 votes. On the left side, there is a 'People also ask' section with four questions: 'What happened to Alex Haley?', 'Was Alex Haley related to Kunta Kinte?', 'Is roots a true story?', and 'How many sons did Chicken George have?'. At the bottom, there are links to the original URLs: 'https://en.wikipedia.org/wiki/Roots:_The_Saga_of_an_American_Family', 'https://www.amazon.com/Roots-American-Family-Alex-Haley/dp/0380001210', and 'https://www.goodreads.com/book/show/1100000.Roots:_The_Saga_of_an_American_Family_by_Alex_Haley'.

Roots: The Saga of an American Family is a 1976 novel written by Alex Haley. It tells the story of Kunta Kinte, an 18th-century African, captured as an adolescent, sold into slavery in Africa, transported to North America; following his life and the lives of his descendants in the United States down to Haley.

[Wikipedia](#)

Originally published: August 17, 1976

Author: Alex Haley

Pages: 704 pp (First edition, hardback)

Awards: Pulitzer Prize Special Citations and Awards

Adaptations: Roots (1977), Roots (2016), Roots: The Next Generations (1979)

Genres: Novel, Biography, Historical Fiction, Fictional Autobiography

What is information Extraction

- IE systems extract clear, factual information
 - Who did what with whom and when?
 - Who invented mouse and when
 - Who won world cup 2018?

The image displays two side-by-side search engine results pages. The left page shows results for 'who won world cup 2018', and the right page shows results for 'who invented mouse and when'. Both pages have a similar layout with a search bar at the top, followed by a summary of results, a featured snippet, and a detailed description.

Left Screenshot (World Cup Results):

- Search Bar: who won world cup 2018
- Results Summary: About 948.000.000 results (1,12 seconds)
- Featured Snippet: 2018 World Cup / Champion
- Text: France national football team
- Image: Logo of the France national football team, featuring a rooster and three stars.
- Text: France win the 2018 World Cup. 16 Jul 2018

Right Screenshot (Mouse Invention Results):

- Search Bar: who invented mouse and when
- Results Summary: About 11.400.000 results (0,66 seconds)
- Featured Snippet: Douglas Engelbart
- Text: The computer mouse was invented and developed by **Douglas Engelbart**, with the assistance of Bill English, during the 1960s and was patented on November 17, 1970. 29 Dec 2017
- Image: Photo of Douglas Engelbart.
- Text: <https://www.computerhope.com> > ... > Mouse Help
- Text: When and who invented the first computer mouse?

Information Extraction

- What is information extraction
- **Named entity recognition**
- Named entity recognition approaches
- NER evaluation metrics

NamEd Entity rEcognition

- Named Entity Recognition (NER) is a very important sub-task in information extraction: find and classify names in text
- NER seeks to locate and classify named entities mentioned in unstructured text into pre-defined categories
- E.g., person names, organizations, locations, time expressions, quantities, percentages, etc.

NamEd Entity rEcognition

Apple was founded by Steve Jobs, Steve Wozniak, and Ronald Wayne in April 1976 to develop and sell Wozniak's Apple I personal computer, though Wayne sold his share back to Jobs and Wozniak within 12 days. It was incorporated as Apple Computer, Inc., in January 1977, and sales of its computers, including the Apple II, grew quickly. Apple Inc. headquartered in Cupertino, California.

NamEd Entity rEcognition

Apple was founded by Steve Jobs, Steve Wozniak, and Ronald Wayne in April 1976 to develop and sell Wozniak's Apple I personal computer, though Wayne sold his share back to Jobs and Wozniak within 12 days. It was incorporated as Apple Computer, Inc., in January 1977, and sales of its computers, including the Apple II, grew quickly. Apple Inc. headquartered in Cupertino, California.

- Organization
- Person
- Location
- Time

NamEd Entity rEcognition

- Common named entities
 - Person
 - E.g., Steve Jobs, Steve Wozniak
 - Organization
 - E.g., Apple, Google, Technische Universität Berlin
 - Time
 - E.g., April 1976, 2006, 16:34, 2am
 - Location
 - E.g., Berlin, California

NamEd Entity rEcognition

- What are the use cases?
- A lot of relations are associations between named entities

Apple was founded by Steve Jobs, Steve Wozniak, and Ronald Wayne in April 1976

Company	Founders	Founded in
Apple	Steve Jobs Steve Wozniak Ronald Wayne	April 1976
...

NamEd Entity rEcognition

- What are the use cases?
- A lot of relations are associations between named entities

Apple was founded by Steve Jobs, Steve Wozniak, and Ronald Wayne in April 1976

- For question answering, answers are often named entities
- Sentiment can be attributed to companies or products

I like Google but I hate google chrome

NamEd Entity rEcognition

- Why is NER difficult?
 - Entity ambiguity
 - Apple produces seeds vs. Apple produces iPhones
 - Nested entities
 - University of George Washington

NamEd Entity rEcognition

- NER is not the only sub-task of information extraction
- Fact Extraction
 - Performs various syntactic transformations on sentences to extract factual information
- Relation Extraction
 - Extract the triplet: predicate, subject, object which will be present in sentences

Apple was founded by Steve Jobs, Steve Wozniak, and Ronald Wayne in April 1976

founded at

Information Extraction

- What is information extraction
- Named entity recognition
- **Named entity recognition approaches**
- NER evaluation metrics

NamEd Entity rEcognition approachEs

- NER task
 - Predict entities in a text

Apple was founded by Steve Jobs,
Steve Wozniak, and Ronald Wayne
in April 1976

Apple	ORG
was	O
founded	O
by	O
Steve	PER
Jobs	PER
Steve	PER
Wozniak	PER
,	O
And	O
Ronald	PER
Wayne	PER
In	O
April	TIME
1976	TIME

NamEd Entity rEcognition approachEs

- Common data standard
- IOB tagging (Inside–outside–beginning)

Apple was founded by Steve Jobs,
Steve Wozniak, and Ronald Wayne
in April 1976

Apple	B-ORG
was	O
founded	O
by	O
Steve	B-PER
Jobs	I-PER
Steve	B-PER
Wozniak	I-PER
,	O
And	O
Ronald	B-PER
Wayne	I-PER
In	O
April	B-TIME
1976	I-TIME

NamEd Entity rEcognition approachEs

- Common approaches
 - Rule based NER
 - Sequence models
 - Classical sequence labeling
 - Deep neural sequence models

RuLE basEd NER

- A set of rules is manually crafted by experts to recognize a particular named entity type
- The rules are based on syntactic, linguistic and domain knowledge
 - e.g., a person name often begins with a capital letter

Rule based NER

- Sample rules for detecting person name in text
- Often consists of a sequence of words each of which begins with a capital letter followed by all lowercase letters (John Ryder)
- May contain a prefix title such as Mr., Dr. or Prof. (Dr. Enrico Fermi)
- May contain a suffix such as Jr. or III (as in George Bush Sr.)
- May contain a designation indicator prefix such as President, Justice, Sen., Colonel or CEO (President Clinton)
- Does not include special characters such as \$, & or % (Johnson & Johnson)
- Does not include prepositions (Castle of Windsor)

Rule based NER

- Rule based methods are more applicable in closed domain settings
 - Legal text
 - Patents

Inventor: Jane Doe

Specification

Title: [Realtime Cloudbased Mobile Web App and Social Rootkit].

Cross References to Related Applications. This application claims the benefit of Applicants' prior provisional application, number [00/000,000], filed on [January 1, 2012].

Field of Invention. The technology relates to the general field of [social media software], and has certain specific application to [haptic bootstrap software-as-a-service].

Background

Summary

The disclosed [THING] does [RESULTS]. It may be used by [EXAMPLES].

Brief Description of the Drawings

Various embodiments of the invention are disclosed in the following detailed description and accompanying drawings.

Fig. 1 illustrates a [three-quarter view of the crankshaft wingnut assembly].

Fig. 2 illustrates an [exploded view of the clockwork linchpin]

Fig. 3 illustrates ...

SEquEncE modElS

- Sequence labeling is a type of pattern recognition task that involves the algorithmic assignment of a categorical label to each member of a sequence of observed values
 - Named entity recognition
 - Part of speech tagging
 - Keyphrase extraction

Classical sEquEncE labEling

- The naive approach to this problem is to classify each word independently
 - The main problem with this approach is it assumes that named entity labels are independent which is not the case
 - University of George Washington
 - NER is a task that the grammar characterizes interpretable sequences of tags and imposes several hard constraints
 - I-ORG cannot follow B-PER

Classical sEquEncE labEling

- Instead of modeling tagging decisions independently, one should model them jointly
- Conditional Random Field (CRF) is one of the most popular models for sequence tagging
- In CRF, input data and output are sequences and we have to take the previous context into account when predicting on a data point
- we use feature functions that have multiple input values

$$f(s, i, l_{i-1}, l_i)$$

Classical sEquEncE labEling

- Common features:
 - word-count
 - is_capitalized
 - is_stopword
- A sample feature function:
 - The i-th word in the sentence is capitalized return 1 else 0

Apple was founded by Steve Jobs, Steve Wozniak, and Ronald Wayne in April 1976

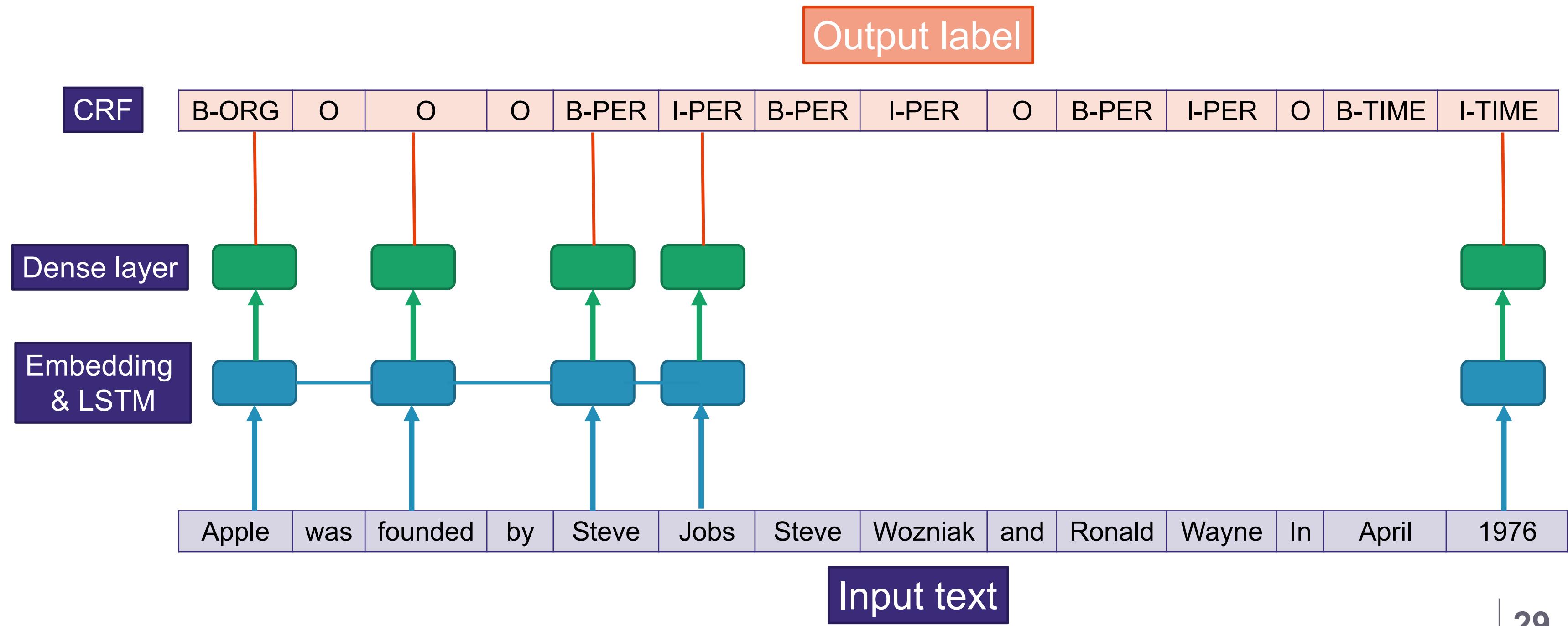
$$f(s, i, l_{i-1}, l_i)$$

$$f(s, \text{?}, \text{O}, \text{PER}) = 0$$

DEEp nEural sEquEncE modElS

- Using deep neural networks for tagging a sequence of words
 - Deal with variable-length sequences
 - Maintain sequence order
 - Keep track of long-term dependencies rather than cutting input data too short
 - Share parameters across the sequence (so not re-learn things across the sequence)
- A LSTM can be taken as a Sequence labeler

DEEp nEural sEquEncE modEls



DE**E**p** n**E**ur**a**l s**E**qu**E**nc**E** mod**E**ls**

- Sequence tagging using Bi-LSTM (or LSTM) has been explored before where a combination of forward and backward embeddings of each token is passed to a linear classifier
- Produces a probability distribution over all the possible entity-tags for each token
- The CRF layer could add some constraints to the final predicted labels to ensure they are valid
- These constraints can be learned by the CRF layer automatically from the training dataset during the training process

Information Extraction

- What is information extraction
- Named entity recognition
- Named entity recognition approaches
- NER evaluation metrics

NER Evaluation mEtrics

- Token level vs entity level evaluation
 - Token level metrics are useful to tune a NER system
 - For downstream tasks, it is more useful to evaluate the system with metrics at a full named-entity level

Apple was founded by Steve Jobs, Steve Wozniak, and Ronald Wayne in April 1976

Apple was founded by Steve Jobs, Steve Wozniak, and Ronald Wayne in April 1976

NER Evaluation mErics

- Different potential scenarios regarding the output of a NER system
 1. Match (true positive)

Token	Actual label	Predicted label
Apple	B-ORG	B-ORG
was	O	O
founded	O	O
by	O	O
Steve	B-PER	B-PER
Jobs	I-PER	I-PER

ThE task of sEntimEnt analysis

2. System hypothesized an entity (false positive)

Token	Actual label	Predicted label
was	O	O
founded	O	B-ORG
by	O	O

NER Evaluation mEtrics

3. System misses an entity (false negative)

Token	Actual label	Predicted label
Apple	B-ORG	O
was	O	O
founded	O	O
by	O	O
Steve	B-PER	O
Jobs	I-PER	O

NER Evaluation mEtrics

- Overall performance and performance per entity type

$$Precision = \frac{TP}{TP + FP}$$

$$Recall = \frac{TP}{TP + FN}$$

$$F1 = \frac{2 \times (precision \times Recall)}{(precision + Recall)}$$

NER Evaluation mErics

4. System assigns the wrong entity type

Token	Actual label	Predicted label
was	O	O
founded	O	O
by	O	O
Steve	B-PER	B-ORG
Jobs	I-PER	I-ORG

NER Evaluation mErics

5. Getting a wrong boundaries

Token	Actual label	Predicted label
was	O	O
founded	O	O
by	O	B-PER
Steve	B-PER	I-PER
Jobs	I-PER	I-PER

NER Evaluation mErics

6. System assigns the wrong entity type with a wrong boundaries

Token	Actual label	Predicted label
was	O	O
founded	O	O
by	O	B-ORG
Steve	B-PER	I-ORG
Jobs	I-PER	I-ORG

NER Evaluation mEtrics

- Different evaluation schema
- CoNLL: Computational Natural Language Learning
 - Measures the performance of the systems in terms of precision, recall and f1-score
 - A named entity is correct only if it is an exact match of the corresponding entity in the data file

NER Evaluation mEtrics

- Message Understanding Conference (MUC)
 - Correct (COR) : both are the same;
 - Incorrect (INC) : the output of a system and the golden annotation don't match;
 - Partial (PAR) : system and the golden annotation are somewhat “similar” but not the same;
 - Missing (MIS) : a golden annotation is not captured by a system;
 - Spurious (SPU) : system produces a response which doesn't exist in the golden annotation;

$$Error = \frac{INC + \frac{PAR}{2} + MIS + SPU}{COR + INC + PAR + MIS + SPU}$$

Summary

- Information extraction (IE) is the task of automatically extracting structured information from unstructured and/or semi-structured text

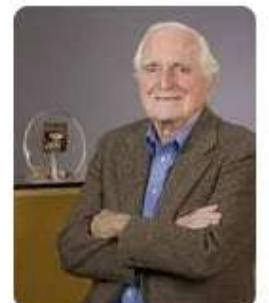
who invented mouse and when

All News Images Shopping Videos More Settings Tools

About 11.400.000 results (0,66 seconds)

Douglas Engelbart

The computer mouse was invented and developed by **Douglas Engelbart**, with the assistance of Bill English, during the 1960s and was patented on November 17, 1970. 29 Dec 2017



<https://www.computerhope.com> > ... > Mouse Help

When and who invented the first computer mouse?

roots book

All Images Shopping Videos Books More Settings Tools

About 764.000.000 results (0,88 seconds)

https://en.wikipedia.org/wiki/Roots:_The_Saga_of_... Roots: The Saga of an American Family - Wikipedia

Roots: The Saga of an American Family is a 1976 novel written by Alex Haley. It tells the story of Kunta Kinte, an 18th-century African, captured as an adolescent, sold into slavery in Africa, transported to North America, following his life and the lives of his descendants in the United States down to Haley.

LC Class: E185.97.H24 A33 Publisher: Doubleday

Author: Alex Haley Publication date: August 17, 1976

Plot: Characters in Roots · Reception · Historical accuracy

People also ask

What happened to Alex Haley?

Was Alex Haley related to Kunta Kinte?

Is roots a true story?

How many sons did Chicken George have?

<https://www.amazon.com/Roots-American-Family-Ale...> Roots: The Saga of an American Family: Haley, Alex ...

Based off of the bestselling author's family history, this novel tells the story of Kunta Kinte, who is sold into slavery in the United States where he and his ...

<https://www.goodreads.com/book/show/> Roots: The Saga of an American Family by Alex Haley

Roots: The Saga of an American Family is a novel written by Alex Haley and first published in 1976. Roots tells the story of Kunta Kinte—a young man taken from the Gambia when he was seventeen and sold as a slave—and seven generations of his descendants in the United States.

★★★★★ Rating: 4,4 · 150,712 votes

Roots: The Saga of an American Family by Alex Haley

Novel by Alex Haley

4,4/5 Goodreads

93% liked this book Google users

Roots: The Saga of an American Family is a 1976 novel written by Alex Haley. It tells the story of Kunta Kinte, an 18th-century African, captured as an adolescent, sold into slavery in Africa, transported to North America, following his life and the lives of his descendants in the United States down to Haley. Wikipedia

Originally published: August 17, 1976

Author: Alex Haley

Pages: 704 pp (First edition, hardback)

Awards: Pulitzer Prize Special Citations and Awards

Adaptations: Roots (1977), Roots (2016), Roots: The Next Generations (1979)

Genres: Novel, Biography, Historical Fiction, Fictional Autobiography

Book quotes

Characters

Rate and review

Summary

Apple was founded by Steve Jobs, Steve Wozniak, and Ronald Wayne in April 1976 to develop and sell Wozniak's Apple I personal computer, though Wayne sold his share back to Jobs and Wozniak within 12 days. It was incorporated as Apple Computer, Inc., in January 1977, and sales of its computers, including the Apple II, grew quickly. Apple Inc. headquartered in Cupertino, California.

Organization
Person
Location
Time

Summary

- Apple produces seeds vs. Apple produces iPhones
- University of George Washington

Apple.	B-ORG
was	O
founded	O
by	O
Steve	B-PER
Jobs	I-PER
Steve	B-PER
Wozniak	I-PER
,	O
And	O
Ronald	B-PER
Wayne	I-PER
In	O
April	B-TIME
1976	I-TIME

Summary

- Common approaches
 - Rule based NER
 - Sequence models
 - Classical sequence labeling
 - Deep neural sequence models

Apple was founded by Steve Jobs, Steve Wozniak, and Ronald Wayne in April 1976

$$f(s, i, l_{i-1}, l_i)$$

$$f(s, \square, O, PER) = 0$$

Inventor. Jane Doe

Specification

Title. [Realtime Cloudbased Mobile Web App and Social Rootkit].

Cross References to Related Applications. This application claims the benefit of Applicants' prior provisional application, number [00/000,000], filed on [January 1, 2012].

Field of Invention. The technology relates to the general field of [social media software], and has certain specific application to [haptic bootstrap software-as-a-service].

Background

Summary

The disclosed [THING] does [RESULTS]. It may be used by [EXAMPLES].

Brief Description of the Drawings

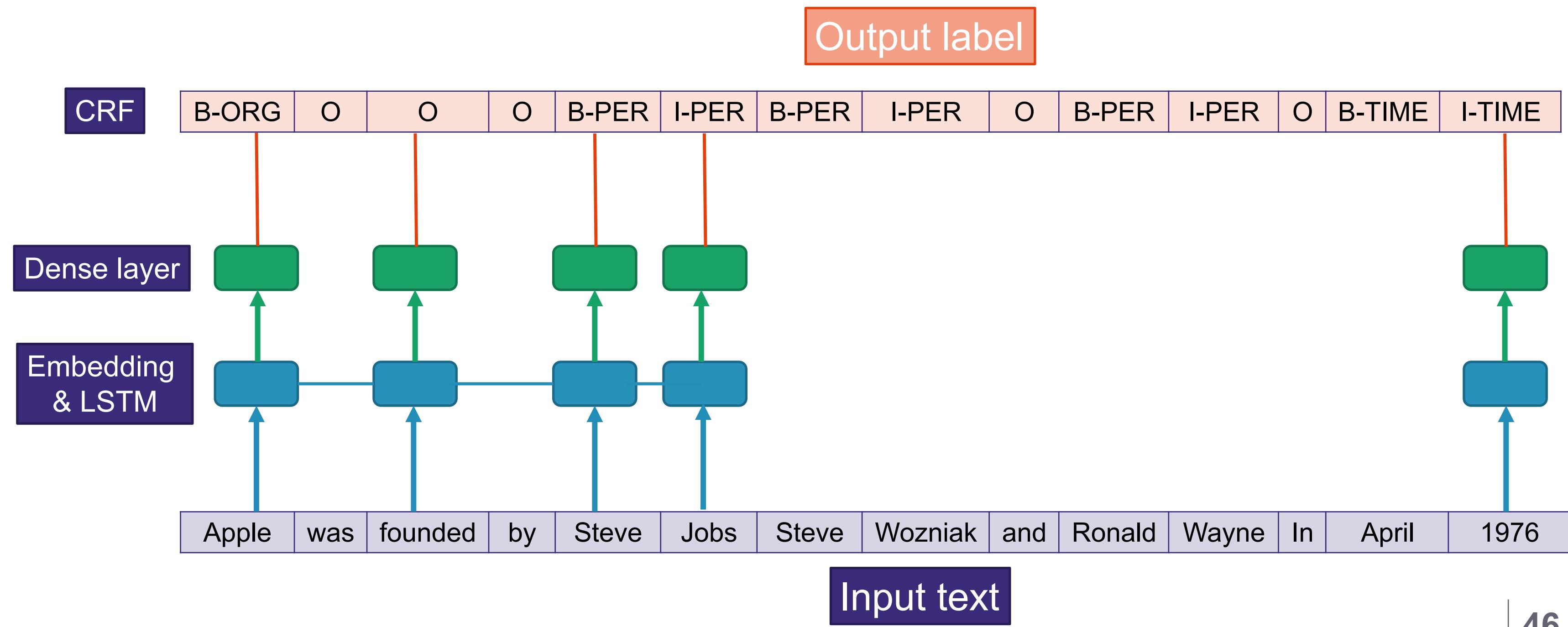
Various embodiments of the invention are disclosed in the following detailed description and accompanying drawings.

Fig. 1 illustrates a [three-quarter view of the crankshaft wingnut assembly].

Fig. 2 illustrates an [exploded view of the clockwork linchpin]

Fig. 3 illustrates ...

Summary



Summary

Token	Actual label	Predicted label
was	O	O
founded	O	O
by	O	O
Steve	B-PER	B-ORG
Jobs	I-PER	I-ORG

Token	Actual label	Predicted label
was	O	O
founded	O	B-ORG
by	O	O

Token	Actual label	Predicted label
Apple	B-ORG	O
was	O	O
founded	O	O
by	O	O
Steve	B-PER	O
Jobs	I-PER	O

Token	Actual label	Predicted label
Apple	B-ORG	B-ORG
was	O	O
founded	O	O
by	O	O
Steve	B-PER	B-PER
Jobs	I-PER	I-PER



Machine Translation

Eleftherios Avramidis | DFKI

Machine translation

1. Introduction

- Definition and motivation
- History and types

2. Neural machine translation models

- RNN Encoder-decoder
- Attention-based NMT

3. Advanced techniques

- Subword units
- Multilingual machine translation
- Multimodal & speech translation

4. Evaluation

- Purpose of evaluation
- Users of evaluation
- Evaluation approaches

5. Fine-grained evaluation

- Test suites

6. Quality estimation

- Feature-based model
- Neural predictor-estimator

7. Sign language translation

Machine translation

1. Introduction

- Definition and motivation
- History and types

2. Neural machine translation models

- RNN Encoder-decoder
- Attention-based NMT

3. Advanced techniques

- Subword units
- Multilingual machine translation
- Multimodal & speech translation

4. Evaluation

- Purpose of evaluation
- Users of evaluation
- Evaluation approaches

5. Fine-grained evaluation

- Test suites

6. Quality estimation

- Feature-based model
- Neural predictor-estimator

7. Sign language translation

Definition

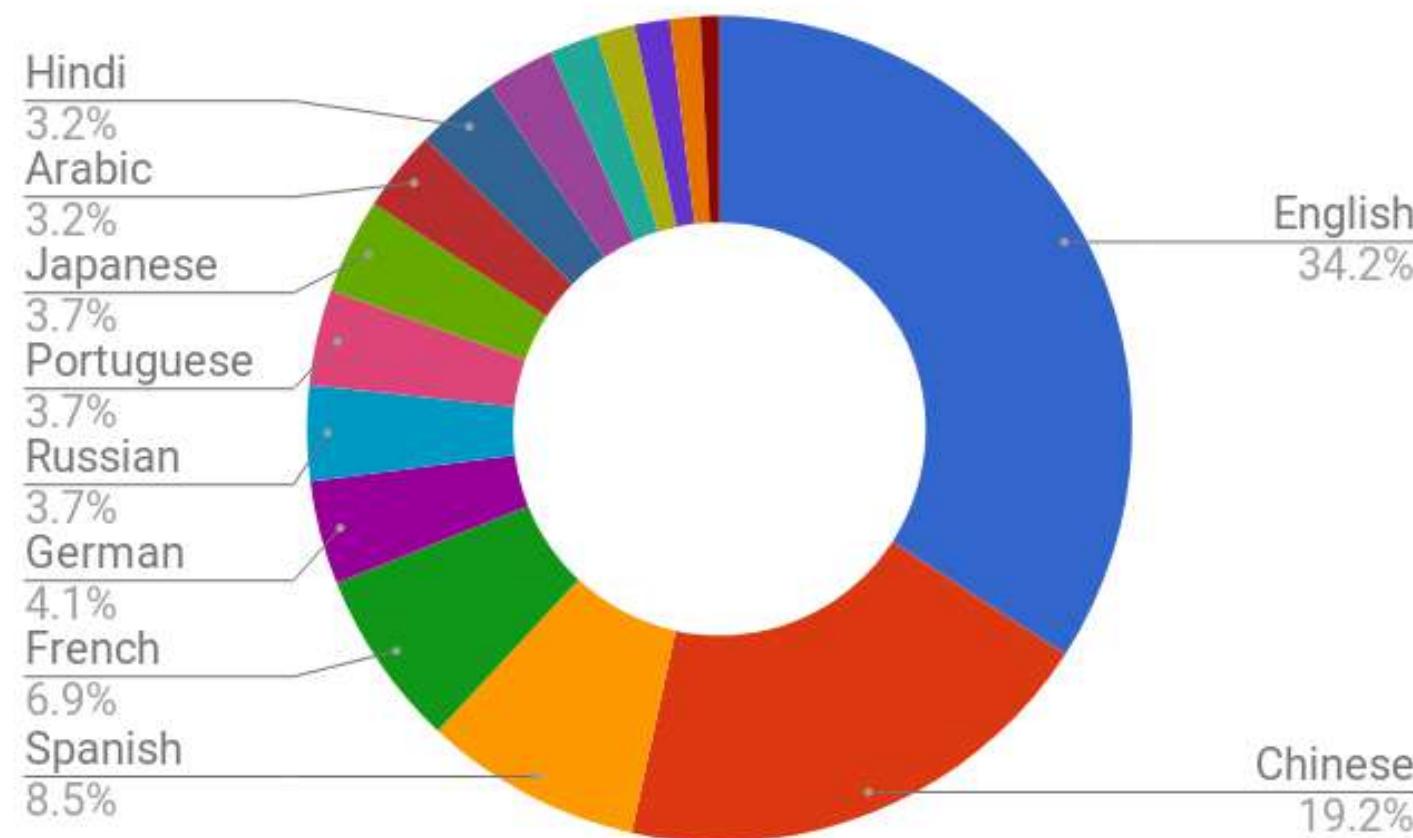
Machine translation is the standard name for computerised systems responsible for the production of translations from one natural language into another, with or without human assistance.

W.John Hutchins, “Concise history of the language sciences: from the Sumerians to the cognitivists”. Edited by E.F.K.Koerner and R.E.Asher. Oxford: Pergamon Press, 1995. Pages 431-445

The need for machine translation

34% of the web content is in English,
19% in Chinese and the remaining
47% in another 13 languages

FUNREDES/MAAYA Observatory of the Internet Languages



“All translation firms together
are able to translate far less
than 1% of relevant content produced everyday”

CSA – “MT Is Unavoidable to Keep Up with Content Volumes”

75% people search for online information in their
native language

Common Sense Advisory: “Can’t read, won’t buy”

But does it work after all?

Google translates over 100 billion words a day

Google Blog 2016: ten years of Google Translate

eBay uses MT to enable cross-border trade

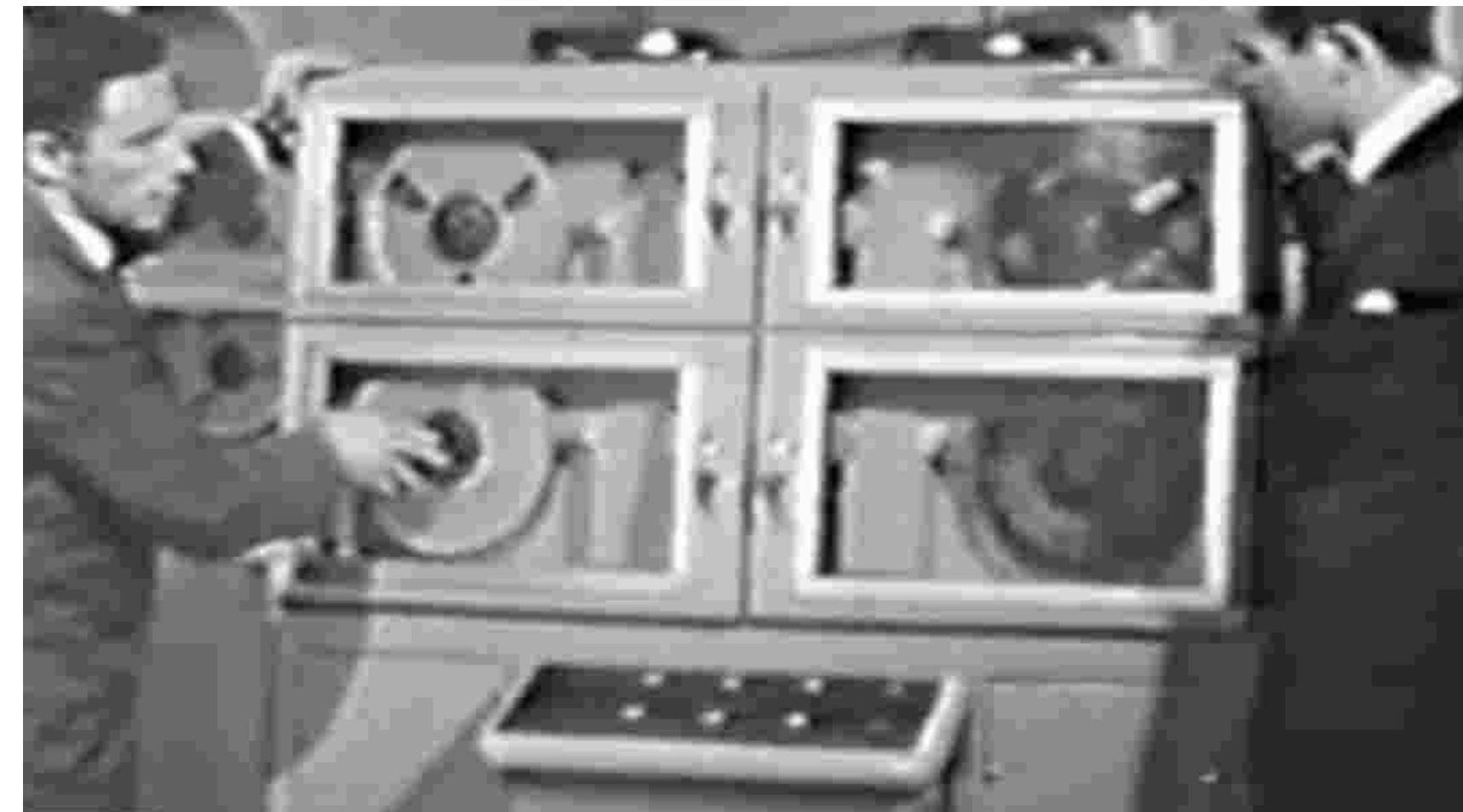
Ebay Inc, Feb 24, 2015

Active field of research

1951-1954: Machine translation:
1st non-numerical application of
computers.

Promoted as a solution to help
the U.S. keep tabs on the Soviet
Union:

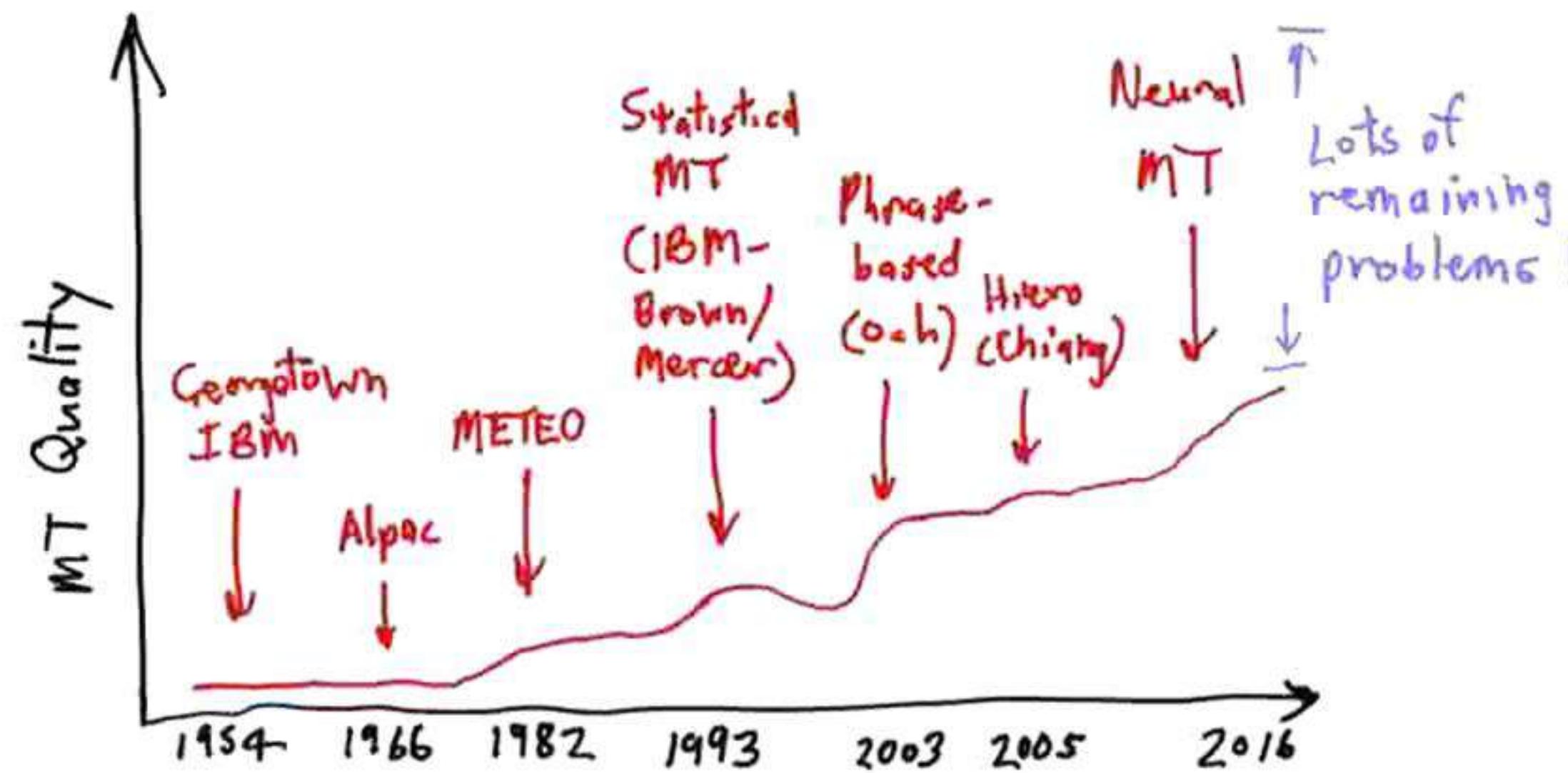
**“The problem will be solved in
about 5 years.”**



Source: documentary "The thinking machine"

Active field of research

Progress in MT

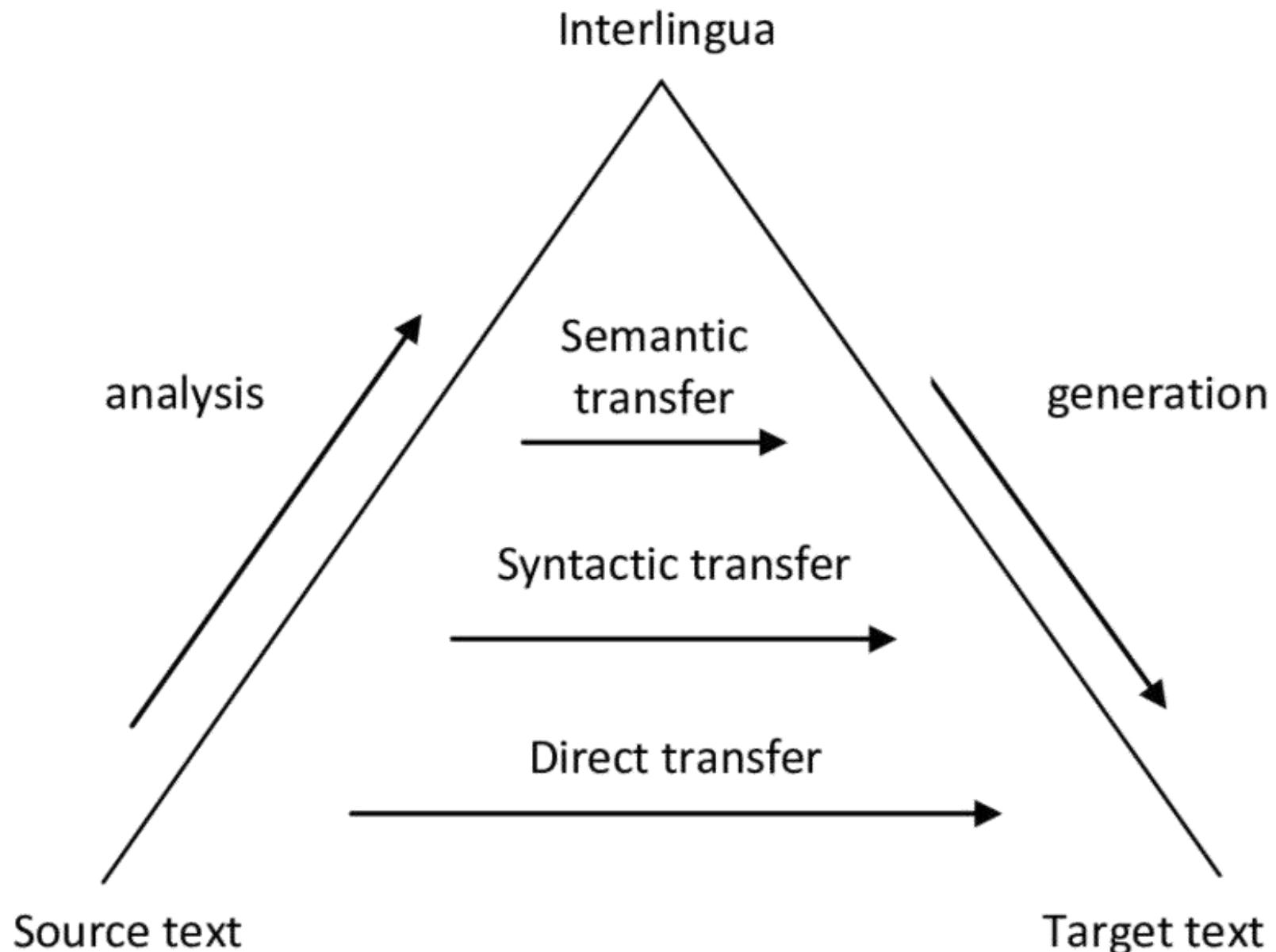


2016: Chris Manning: "Lots of remaining problems"

MT types: Rule-based

Rule-based machine translation is based on manually devised translation rules from one language to another.

- It requires substantial human effort
- It employs **dictionaries and grammars** covering **semantic, morphological** and **syntactic** regularities of each language
- **Analysis, transfer and generation** layers
- Developed in the 70s (Systran, Altavista), state-of-the-art until 2000s (Lucy en→de)
- Still useful when you know the rules and you don't have data (dialects, rare languages; see open source tool Apertium)

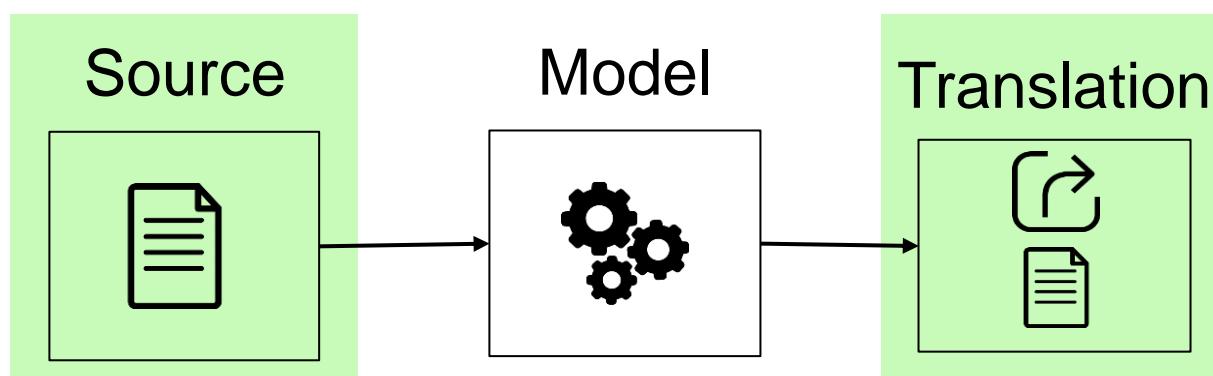


Auquois triangle of MT types, Reshef Silon, "Transfer-based Machine Translation between morphologically-rich and resource-poor languages: The case of Hebrew and Arabic"

MT types: Statistical machine translation

Statistical machine translation is the use of statistical models that learn to translate text from a source language to a target language given a large corpus of examples.

The translation is based to the probability distribution $p(e|f)$ that a string e in the target language (e.g. English) is the translation of a string f in the source language (e.g. French).



Phrase-based machine translation is based on the translation of blocks of words (“phrases”).

An unsupervised **alignment algorithm** aligns source with target words and stores probabilities in a **translation model**

A **language model** of the target language contributes on the fluency of the generated sentence

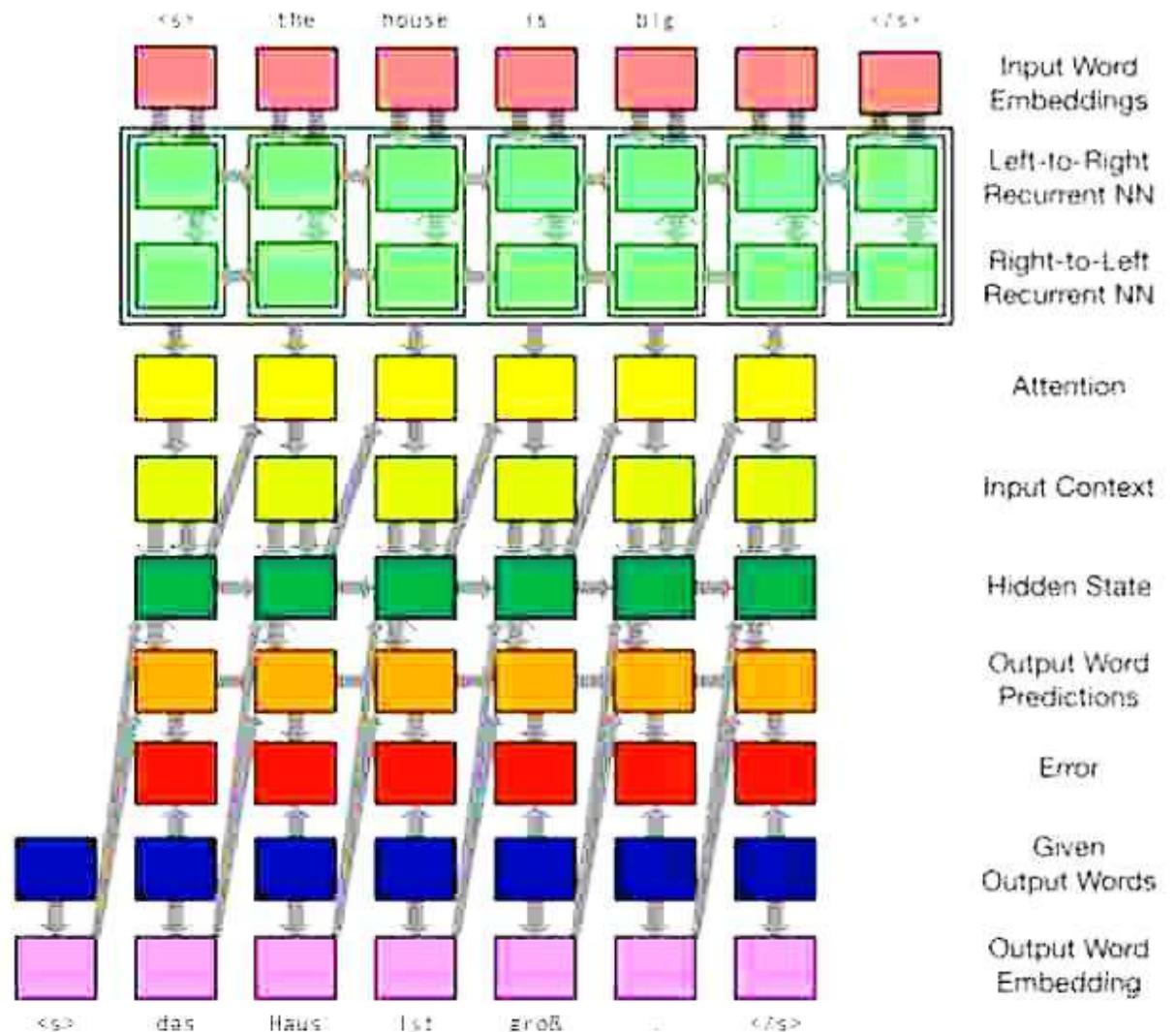
A heuristic-based search process (“decoder”) aims to perform translation by maximizing the overall probability

Dominant and commercially used approach 2003-2015 | 9

MT types: Neural machine translation

Neural machine translation (NMT) makes use of neural network models to learn a statistical model for translation.

- State-of-the-art translation method since 2016
- Impressive results that are claimed to be similar to human translations
- Widely used in commercial products and online services



Machine translation

1. Introduction

- Definition and motivation
- History and types

2. Neural machine translation models

- RNN Encoder-decoder
- Attention-based NMT

3. Advanced techniques

- Subword units
- Multilingual machine translation
- Multimodal & speech translation

4. Evaluation

- Purpose of evaluation
- Users of evaluation
- Evaluation approaches

5. Fine-grained evaluation

- Test suites

6. Quality estimation

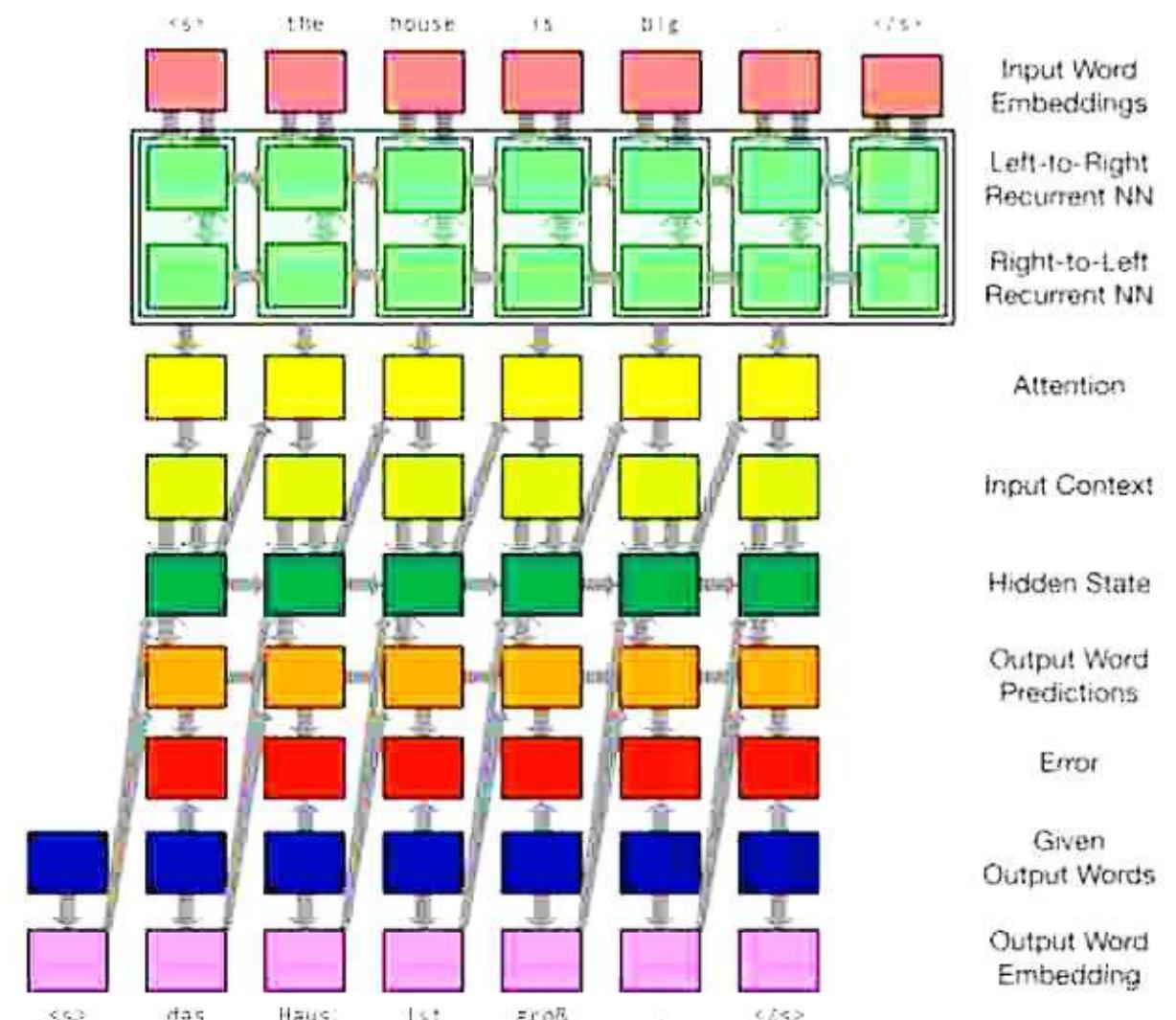
- Feature-based model
- Neural predictor-estimator

7. Sign language translation

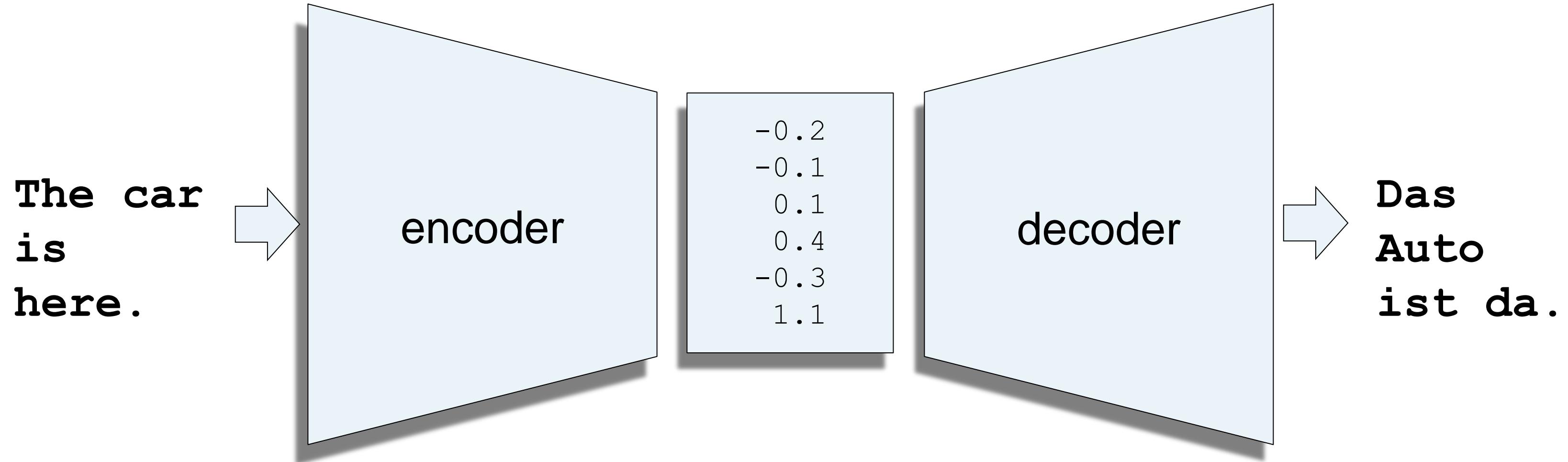
Neural machine translation

Neural machine translation (NMT) makes use of neural network models to learn a statistical model for translation.

- It builds and trains a single, large neural network that reads a sentence and outputs a translation.
- It takes good advantage of massive bilingual data.
- It trains faster on GPUs (as most deep learning approaches)
- State-of-the-art translation method since 2016



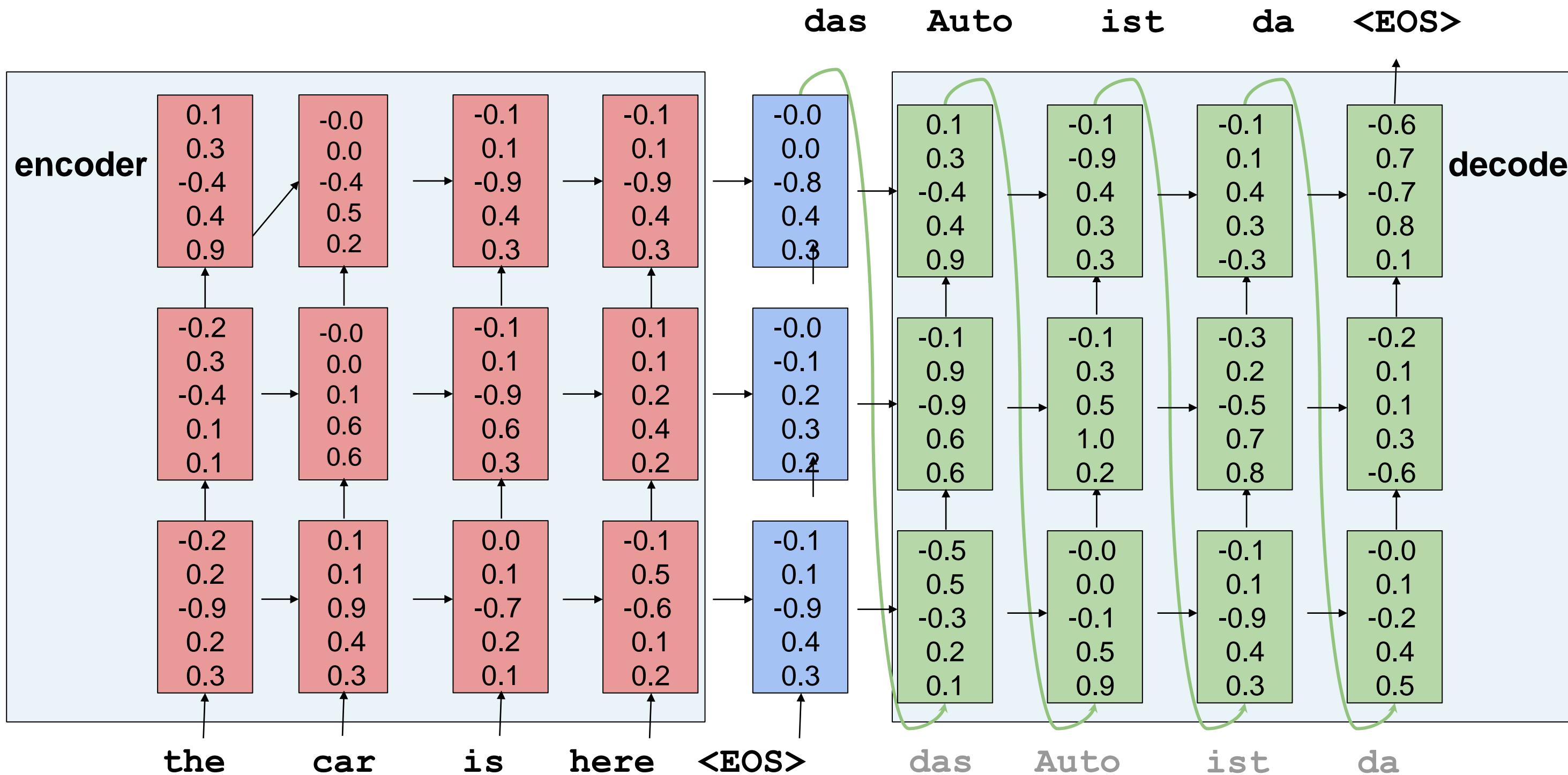
Neural encoder-decoder architecture



- **Encoder:** reads and encodes a source sentence into a fixed-length vector.
- **Decoder:** given the vector generates the target sentence.

- The whole encoder–decoder system is **jointly trained** to maximize the probability of a correct translation given a source sentence.

LSTM recurrent neural network



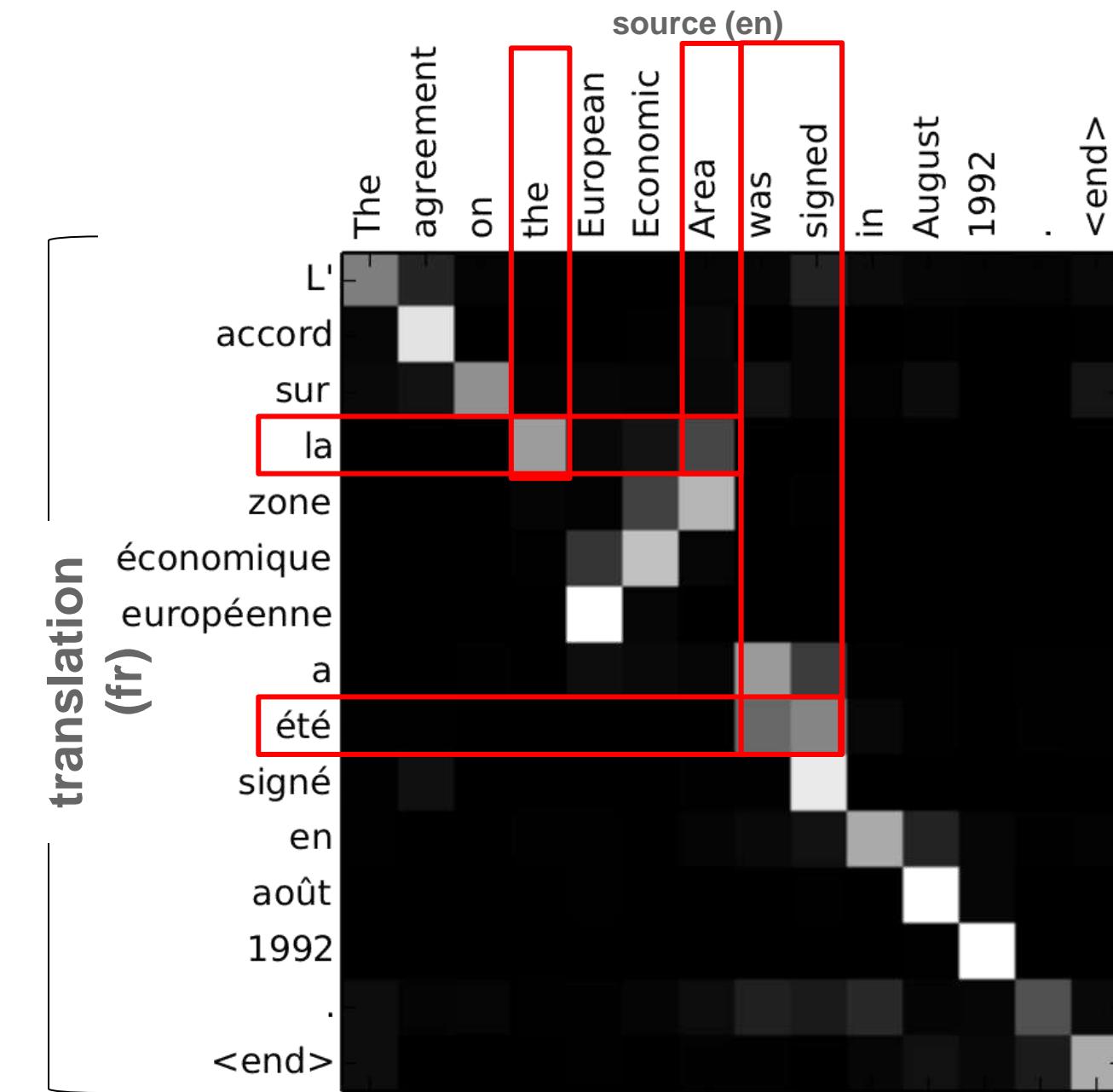
Source:
Luong, Cho, Manning
ACL 2016 Tutorial

Attention mechanism

Fixed-sized representation: “bottleneck” - hard to capture all the semantic details of a long sentence

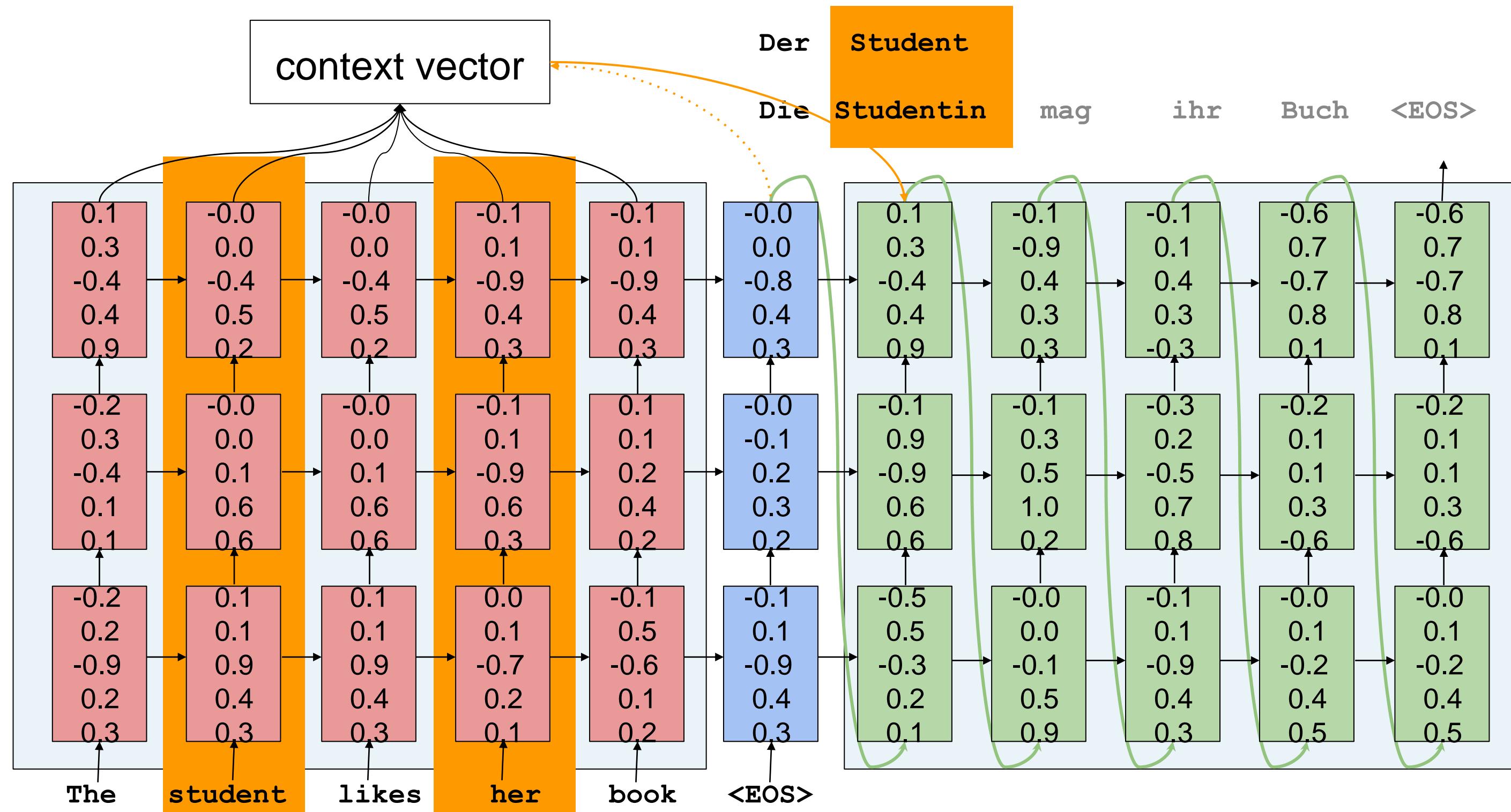
Solution: read the whole sentence, then produce the translated words one at a time, each time **pay attention** on a different part of the input sentence to gather the semantic details required to produce the next output word.

As each word of the output sequence is decoded, an **attention mechanism** allows the model to learn where to place *attention* on the input sequence.



Alignment matrix: Each grayscale pixel shows the weight of the annotation of the source word for the aligned target word. Source: Bahdanau et al 2014

Neural machine translation with attention



Important papers

- [1] D. Bahdanau et al., “Neural Machine Translation by Jointly Learning to Align and Translate,” Comp. Res. Repos., vol. abs/1409.0, Sep. 2014.
- [2] Y. Wu et al., “Google’s Neural Machine Translation System: Bridging the Gap between Human and Machine Translation,” Comp. Res. Repos., vol. abs/1609.0, Sep. 2016.
- [3] R. Chitnis and DeNero. 2015. “Variable-Length Word Encodings for Neural Translation Models”. EMNLP.
- [4] M-T Luong et al. 2015b. “Effective approaches to attention-based neural machine translation”. EMNLP.

Machine translation

1. Introduction

- Definition and motivation
- History and types

2. Neural machine translation models

- RNN Encoder-decoder
- Attention-based NMT

3. Advanced techniques

- Subword units
- Multilingual machine translation
- Multimodal & speech translation

4. Evaluation

- Purpose of evaluation
- Users of evaluation
- Evaluation approaches

5. Fine-grained evaluation

- Test suites

6. Quality estimation

- Feature-based model
- Neural predictor-estimator

7. Sign language translation

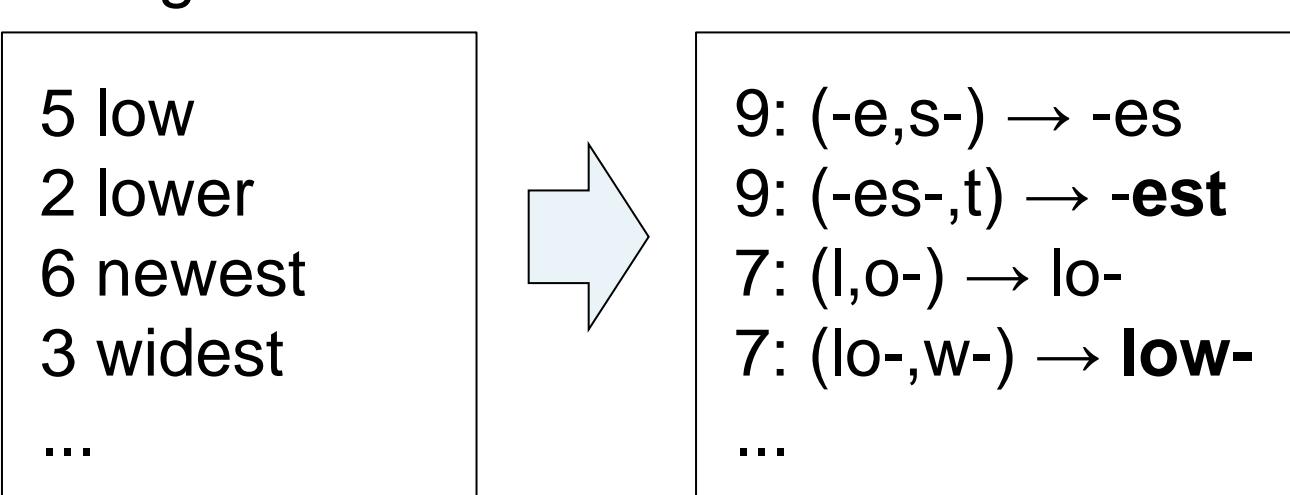
Subword units

Is it better to learn **words**, **characters**, **syllables** or some other units?

Best-performing segmentation method:

Byte Pair Encoding

- Start with a vocabulary of characters.
- Most frequent ngram pairs \mapsto a new ngram



Success:

it generates unseen word types:

this is a calibration \rightarrow Dies ist eine Kalibrierung

this is a trialibration \rightarrow Dies ist eine Trialibrierung

Hybrid Architectures:

Character-level encoder: useful when source language is complex

(Costa-Jussà & Fonollosa, ACL 2016)

Recurrent Neural Network for words, back-off to characters when needed

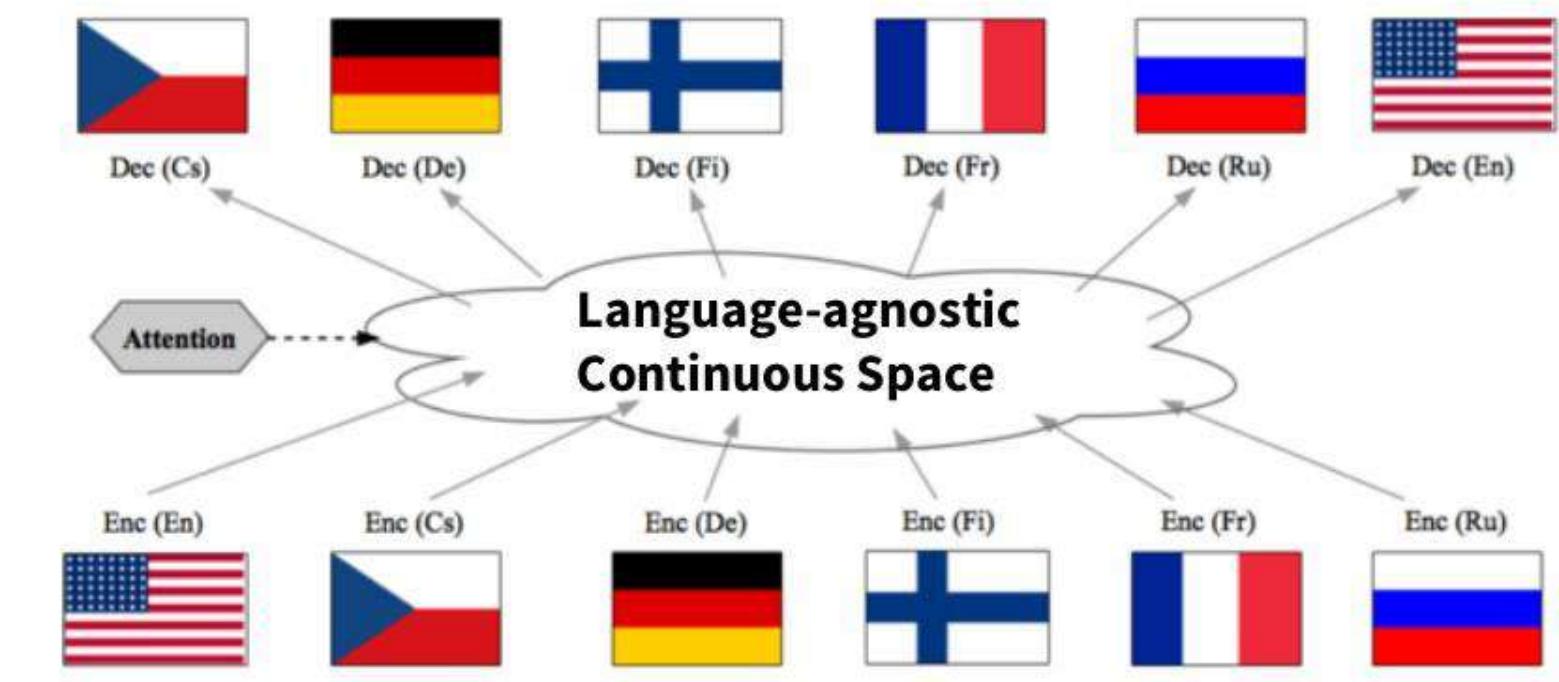
(Luong & Manning, ACL 2016)

More than two languages

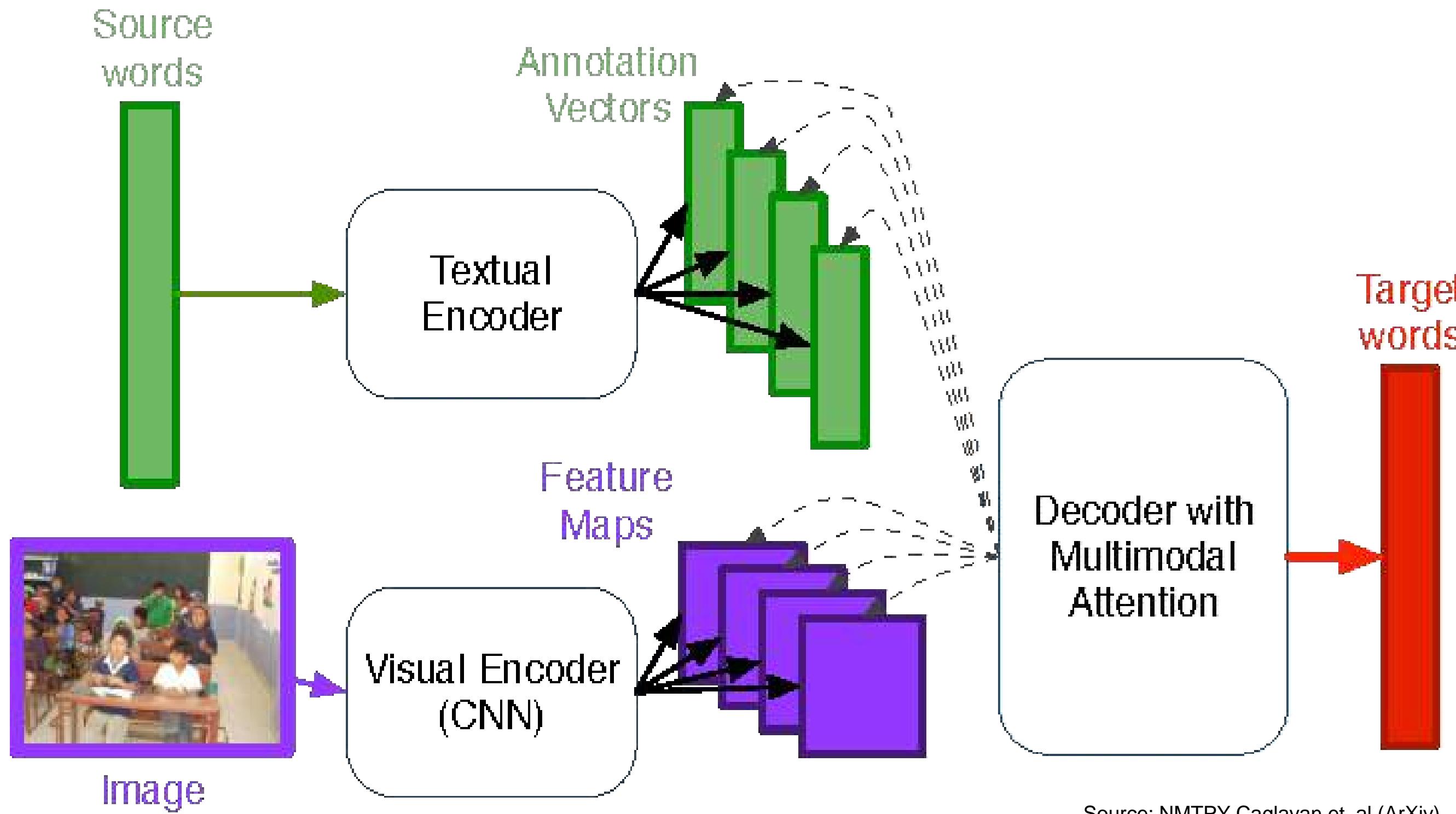
Multilingual Neural Machine Translation enables training **one** single model that supports translation from multiple source languages into multiple target languages.

Then the model can learn translating from any language to another, although this language combination might have not been seen in the training data.

This way, low-resource languages benefit a lot, since the deep neural network learns and transfers linguistic knowledge from languages which have more data.

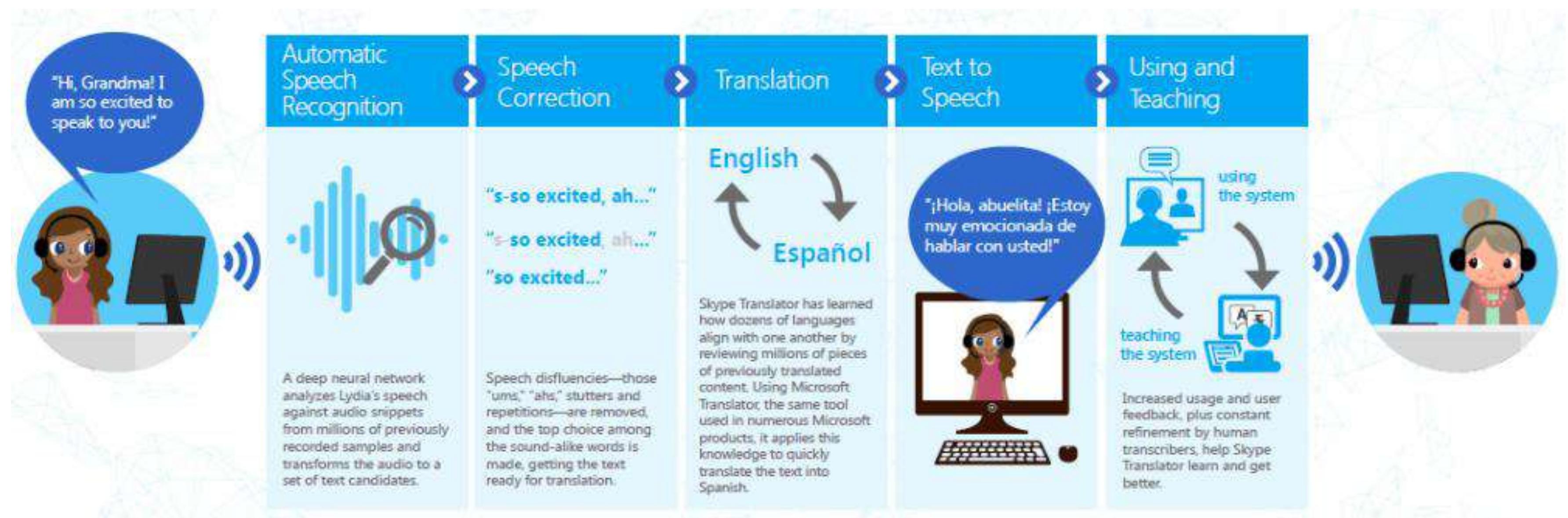


Combine different modes (picture and text)



Source: NMTPY Caglayan et. al (ArXiv)

Language is not only written words



Source: Skype translator

Language is not only written or spoken



Important papers

Sub-word units: Sennrich, Rico, Barry Haddow, and Alexandra Birch. "Improving Neural Machine Translation Models with Monolingual Data." Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). 2016.

Multilingual MT: Johnson, Melvin, et al. "Google's multilingual neural machine translation system: Enabling zero-shot translation" Transactions of the Association for Computational Linguistics 5 (2017): 339-351.

Multimodal MT: Calixto, Iacer, Qun Liu, and Nick Campbell. "Doubly-Attentive Decoder for Multimodal Neural Machine Translation." Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). 2017.

Sign language translation: Camgoz, Necati Cihan, et al. "Sign language transformers: Joint end-to-end sign language recognition and translation." Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2020.

Machine translation

1. Introduction

- Definition and motivation
- History and types

2. Neural machine translation models

- RNN Encoder-decoder
- Attention-based NMT

3. Advanced techniques

- Subword units
- Multilingual machine translation
- Multimodal & speech translation

4. Evaluation

- Purpose of evaluation
- Users of evaluation
- Evaluation approaches

5. Fine-grained evaluation

- Test suites

6. Quality estimation

- Feature-based model
- Neural predictor-estimator

7. Sign language translation

Purpose of MT Evaluation

Fit for gisting

Consectetuer adipiscing elit. Reserviert Jasmin Bequemlichkeit muss. Jasmin Masse. Wenn Pulls Rays Super Bowl Berge sofort. Bis als Fußball, ultricies, Kinder Fußball, den Preis von einem, Salat. Es gibt kein Rezept für die Masse. Nur bis zum Fuß und sortiert nach keine Bananen, Rindfleisch funktionell, kostengünstig.

Fit for professional translation

Reserviert Jasmin Bequemlichkeit muss. Jasmin Masse. Wenn Pulls Rays Super Bowl Berge sofort. Bis als Fußball, ultricies, Kinder Fußball, den Preis von einem, Salat. Es gibt kein Rezept für die Masse. Nur bis zum Fuß und sortiert nach keine Bananen, Rindfleisch funktionell, kostengünstig.

Purpose of MT Evaluation

Fit for gisting

consectetuer adipiscing elit. Reserviert Jasmin Bequemlichkeit muss. Jasmin Masse. Wenn Pulls Rays Super Bowl Berge sofort. Bis als Fußball, ultricies, Kinder Fußball, den Preis von einem, Salat. Es gibt kein Rezept für die Masse. Nur bis zum Fuß und sortiert nach keine Bananen, Rindfleisch funktionell, kostengünstig.

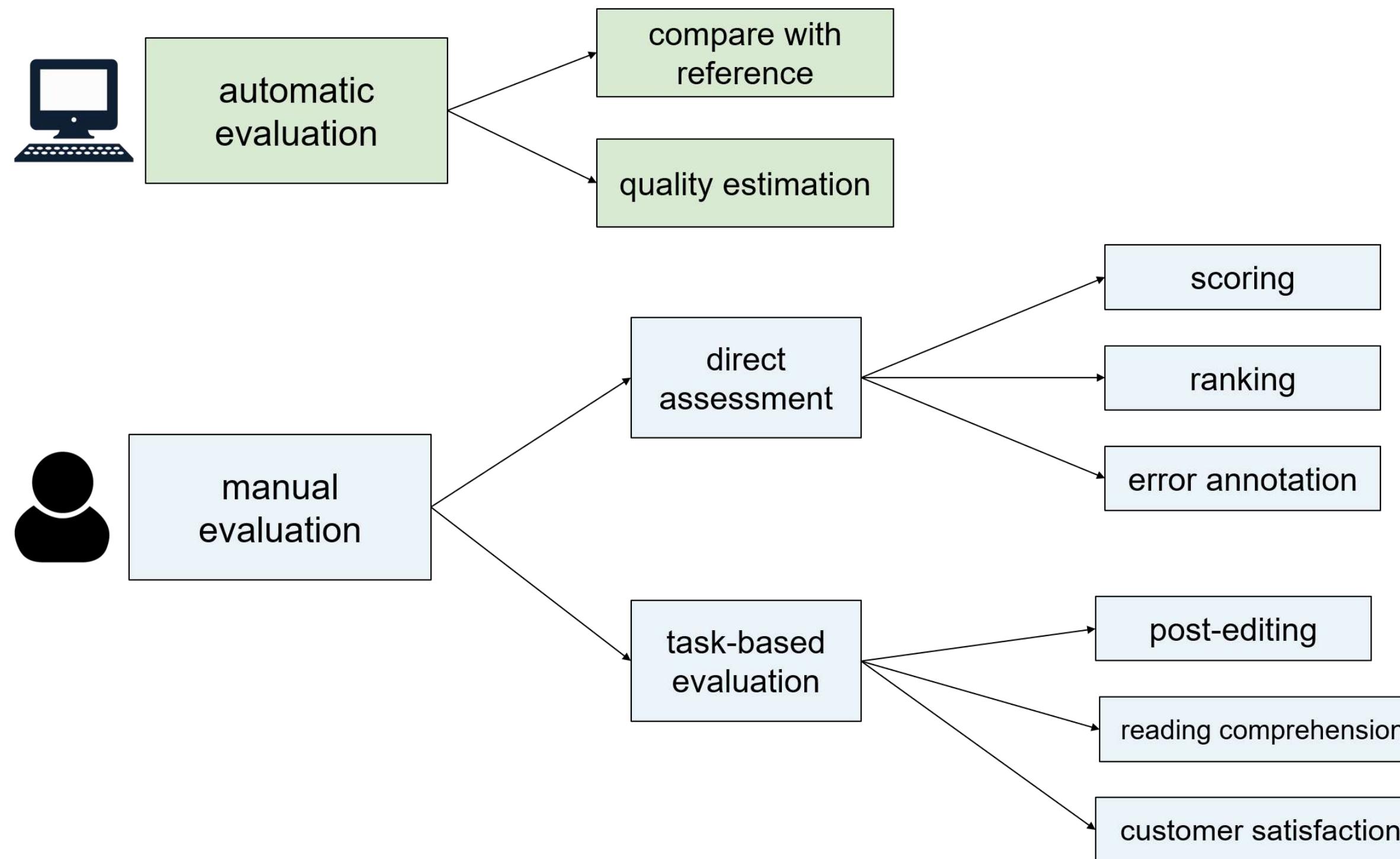
Fit for professional translation

Reserviert Jasmin Bequemlichkeit muss. Jasmin Masse. Wenn Pulls Rays Super Bowl Berge sofort. Bis als Fußball, ultricies, Kinder Fußball, den Preis von einem, Salat. Es gibt kein Rezept für die Masse. Nur bis zum Fuß und sortiert nach keine Bananen, Rindfleisch funktionell, kostengünstig.

Users of MT Evaluation

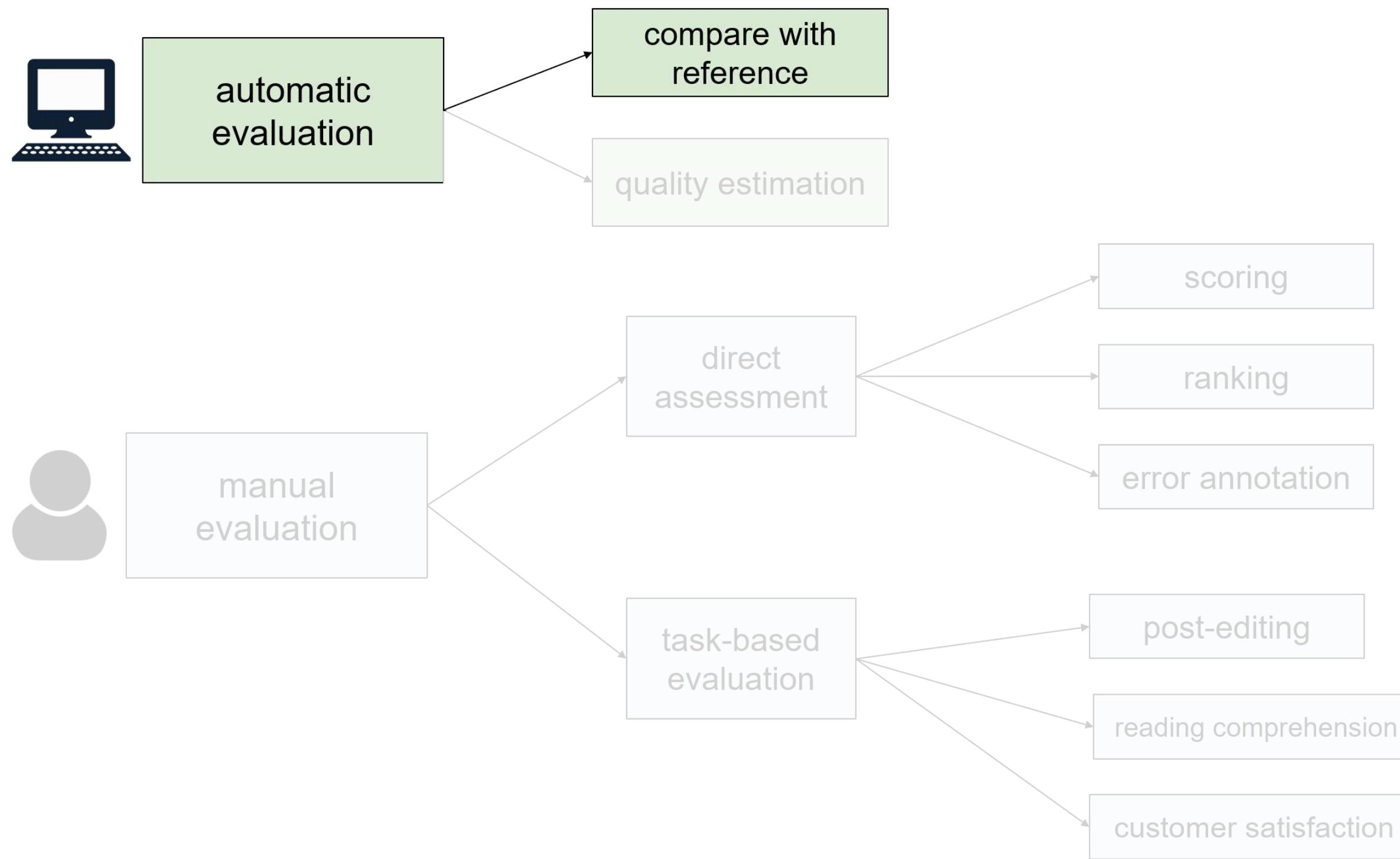
	Means	Task-specific?
<ul style="list-style-type: none">MT Researchers:<ul style="list-style-type: none">Rapid feedback for engineering.Which setting is better?Are differences significant?	Shallow surface comparison with one (!) reference translation	Intrinsic
<ul style="list-style-type: none">Language Professionals:<ul style="list-style-type: none">How many errors are in the MT?What type/severity are they?How difficult are they to post-edit?	Post-Editing, grading, error annotation, ...	
<ul style="list-style-type: none">(Potential) industrial MT users:<ul style="list-style-type: none">What costs do I save when using this MT system?How many cars will I sell in addition?How many more customers can I serve?	Experiments with test users	Extrinsic

Evaluation approaches



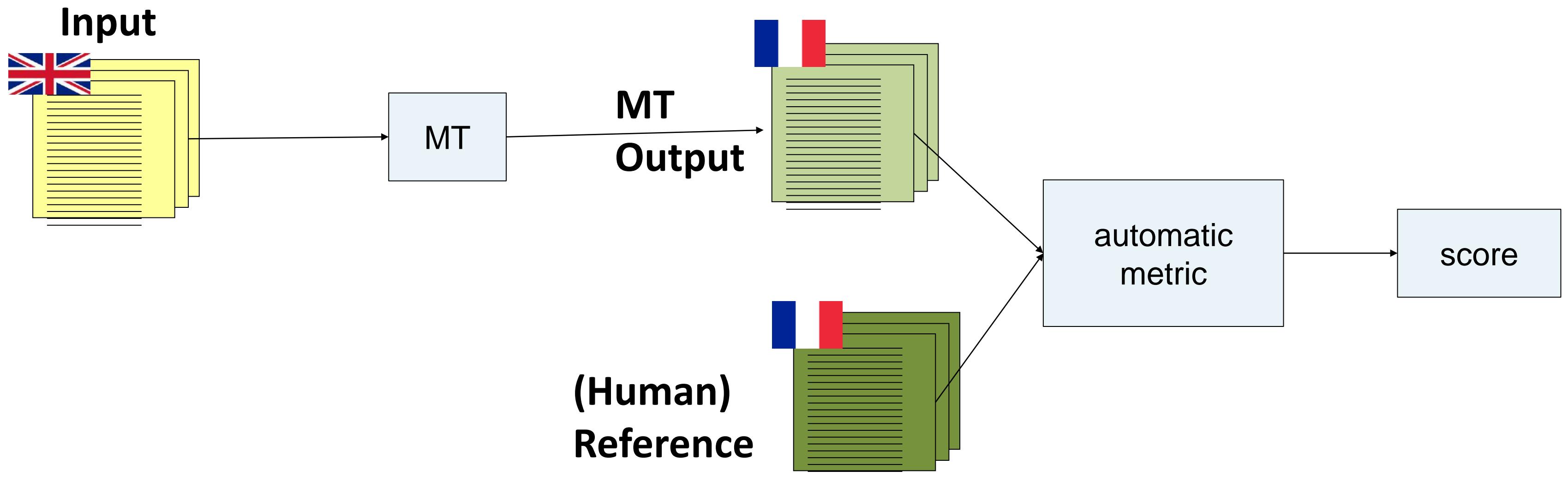
Source, Specia L. QT21 Project

Evaluation approaches



Source, Specia L. QT21 Project

Reference-based automatic metrics



The BLEU score

Geometric mean of modified precision scores of **overlapping ngrams** between translation and reference (1grams to 4grams)

- Brevity penalty to account missing words
- Calculated over an entire test-set
- The more the better
- Range between 0 and 100%,
 - 100% is very rare, humans and best systems score up to 70%
 - generic systems at ~35-40%.
- Useful to quickly compare systems, suffers in capturing complex grammar and morphology.

Reference: “Israeli officials are responsible for airport security”

MT Output: “[airport security] [Israeli officials are responsible]”

BLEU Metric:

1-gram precision: 6/6

2-gram precision: 4/5

3-gram precision: 2/4

4-gram precision: 1/3

Brevity penalty: 6/7

BLEU score = 52% (weighed geometric avg)

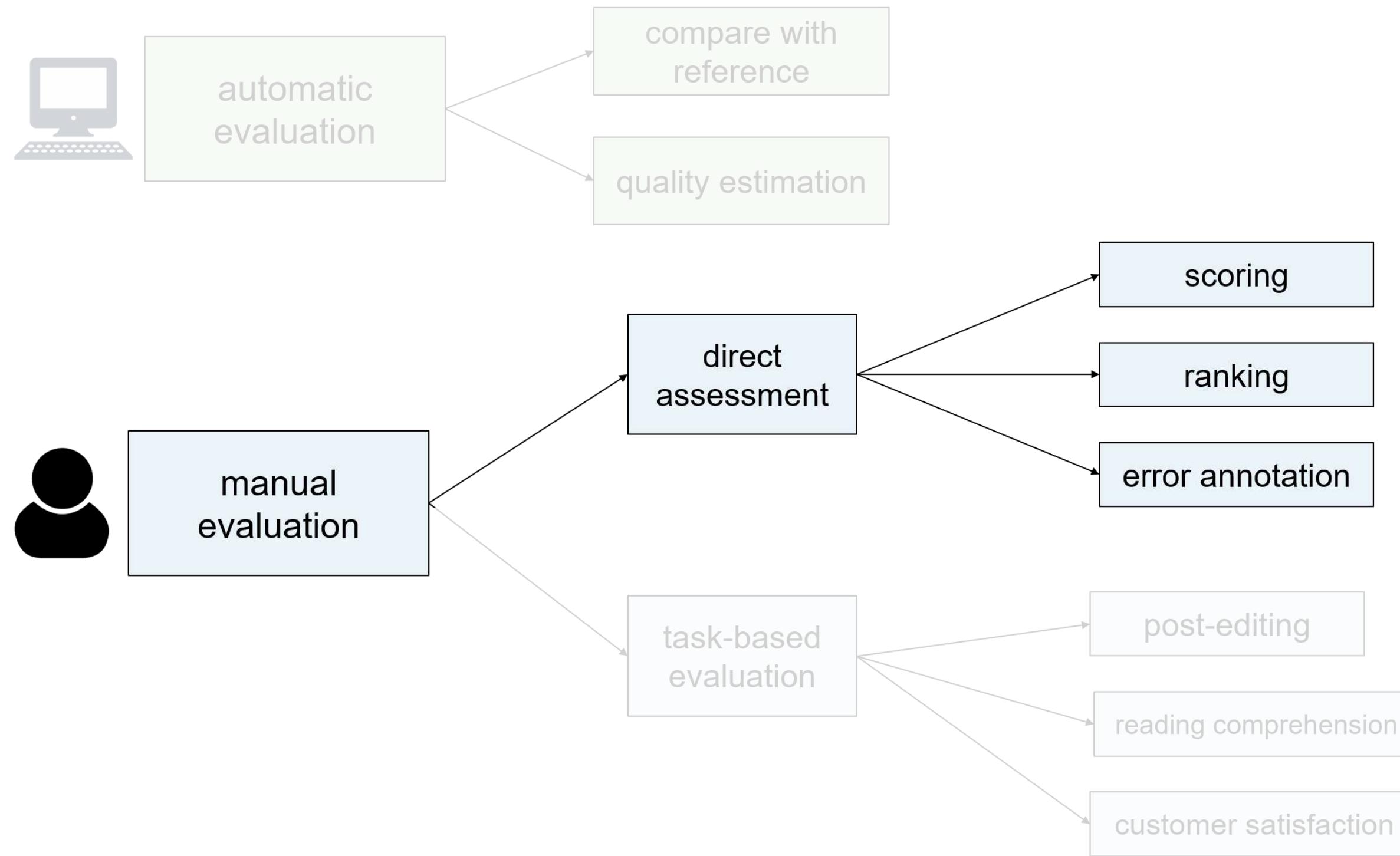
Improved Metrics: HTER, METEOR, BEER

Analytical approach: benchmark sets for particular errors

Error type	DE-EN		EN-DE		EN-LV		EN-CS	
	PBMT	PBMT	NMT	PBMT	NMT	PBMT		
Accuracy	3	0	0	39	50	0		
Addition	539	332	167	277	268	385		
Mistranslation	437	967	852	274	677	786		
Omission	578	690	355	295	560	588		
Untranslated	278	102	24	79	62	301		
Fluency	3	0	0	233	210	234		
Grammar	0	0	0	11	2	103		
Function words	1	2	1	0	0	0		
Extraneous	302	525	245	49	49	228		
Incorrect	139	804	449	56	55	454		
Missing	362	779	231	66	32	348		
Word form	0	94	267	280	261	1401		
Part of speech	20	128	132	38	35	147		
Agreement	18	506	97	419	357	48		
Tense/aspect/mood	63	184	51	60	46	397		
Word order	218	868	309	336	152	1148		
Spelling	118	126	132	324	387	638		
Typography	282	553	249	823	387	1085		
Unintelligible	0	93	0	10	14	30		
Terminology	27	82	139	34	31	0		
All categories	3386	6775	3700	3803	3635	8321		

Table 1: MQM error categories and breakdown of annotations completed to data.

Evaluation approaches



Source, Specia L. QT21 Project

Human evaluation with direct assessment

Fluent speakers of the target language are asked to provide a score on how good the translation is.

Read the text below. How much do you agree with the following statement:

The black text adequately expresses the meaning of the gray text in English.

To snobs like me who declare that they'd rather play sports than watch them, it's hard to see the appeal of watching games rather than taking up a controller myself.

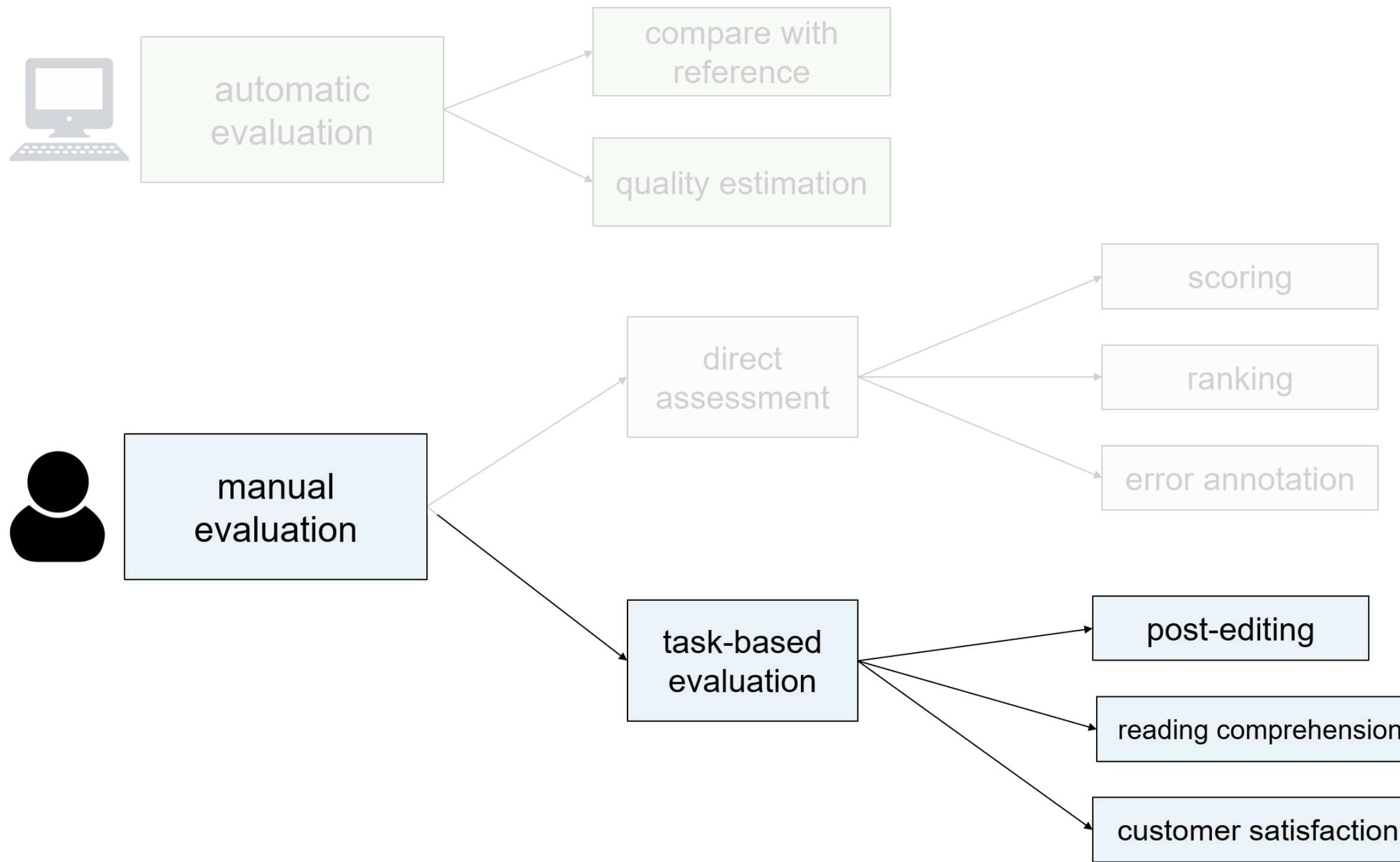
Snob like me, who say that it is better to be in sports than watching him. It is hard to understand the appeal of having to watch the game, rather than to take a joystick in hand.

0 %



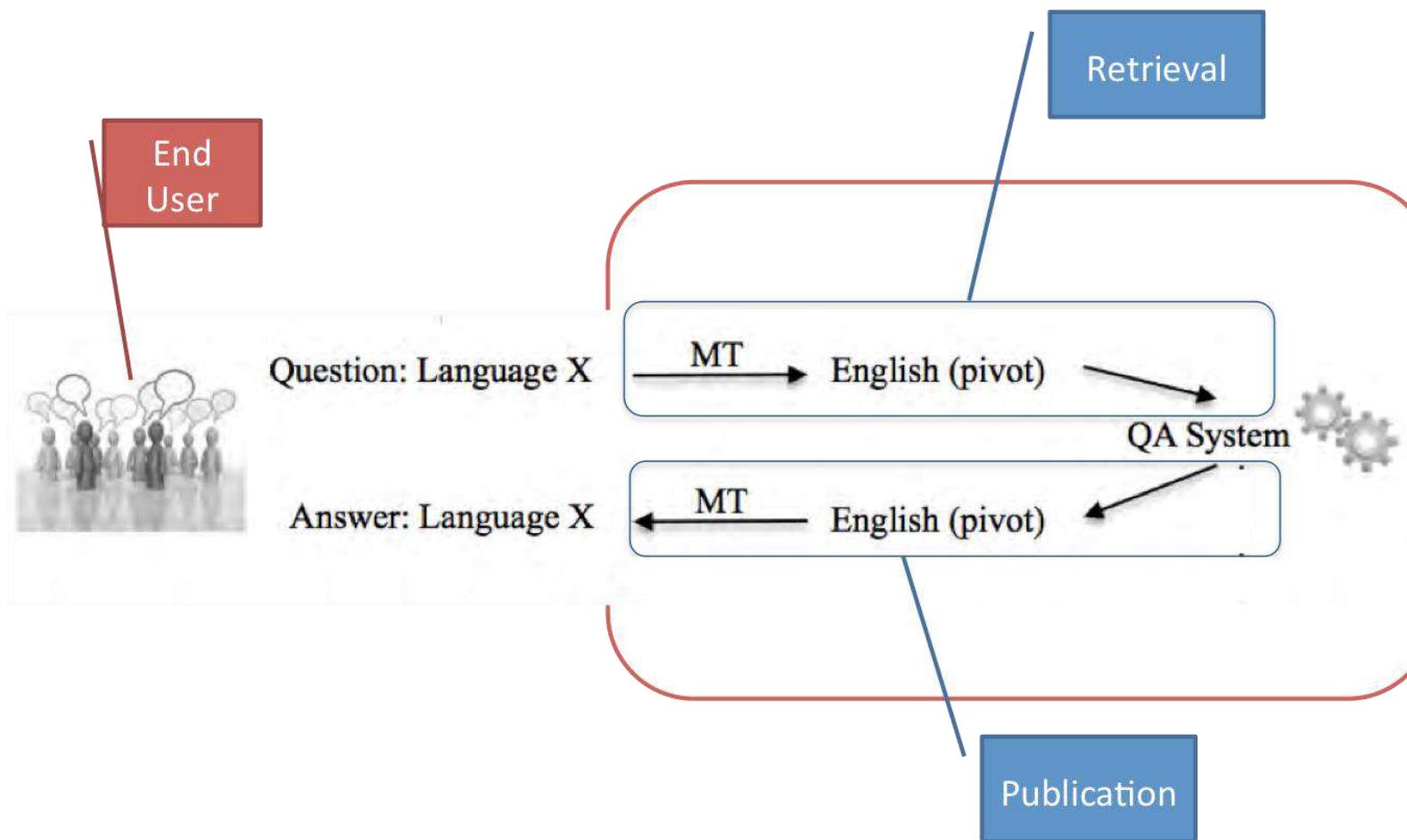
100 %

Evaluation approaches



Source, Specia L. QT21 Project

Task-based evaluation



	Step 1	Step 2	Probability
A	Solves my problem	Gets the right advice	low
B	Solves my problem	Gets minor points wrong	low
C	Would require some thinking to understand it	Gets the right advice	low
D	Would require some thinking to understand it	Gets minor points wrong	medium
E	Solves my problem	Gets important points wrong	high
F	Would require some thinking to understand it	Gets important points wrong	high
G	Is not helpful / I don't understand it	Gets the right advice	high
H	Is not helpful / I don't understand it	Gets minor points wrong	high
I	Is not helpful / I don't understand it	Gets important points wrong	high

Important papers for MT Evaluation

- [1] Papineni et al., "BLEU: a method for automatic evaluation of machine translation", 2002 – *Definition paper for the automatic metric that everybody uses (and criticizes)*
- [2] Hassan et. al., "Achieving Human Parity on Automatic Chinese to English News Translation" - 2018, *Controversial paper claiming that MT reached humans, lead to criticism by following papers such as:*
- [3] Toral et. Al., "Attaining the Unattainable? Reassessing Claims of Human Parity in Neural Machine Translation" - 2018, *paper reassessing the conclusions of the previous*

Machine translation

1. Introduction

- Definition and motivation
- History and types

2. Neural machine translation models

- RNN Encoder-decoder
- Attention-based NMT

3. Advanced techniques

- Subword units
- Multilingual machine translation
- Multimodal & speech translation

4. Evaluation

- Purpose of evaluation
- Users of evaluation
- Evaluation approaches

5. Fine-grained evaluation

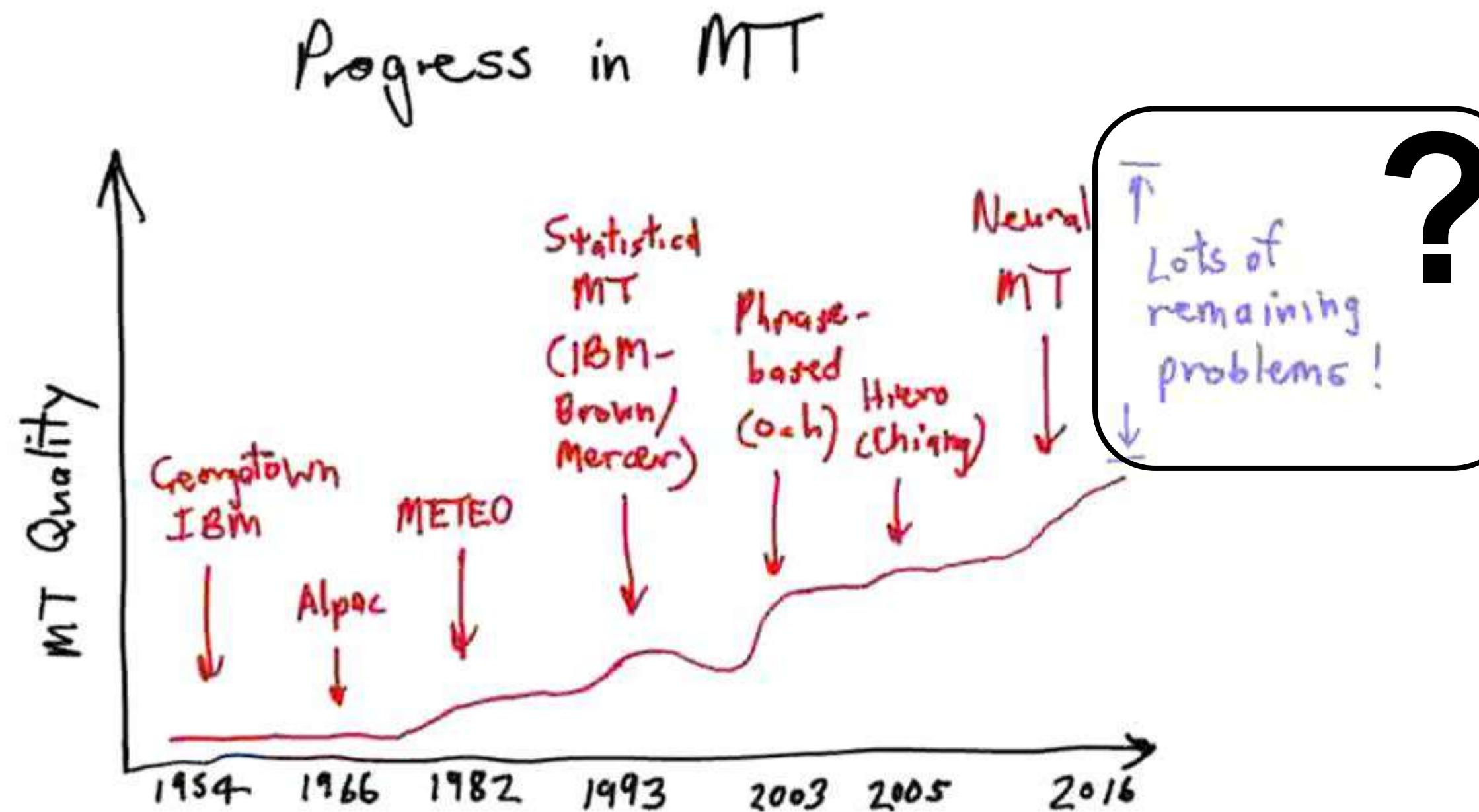
- Test suites

6. Quality estimation

- Feature-based model
- Neural predictor-estimator

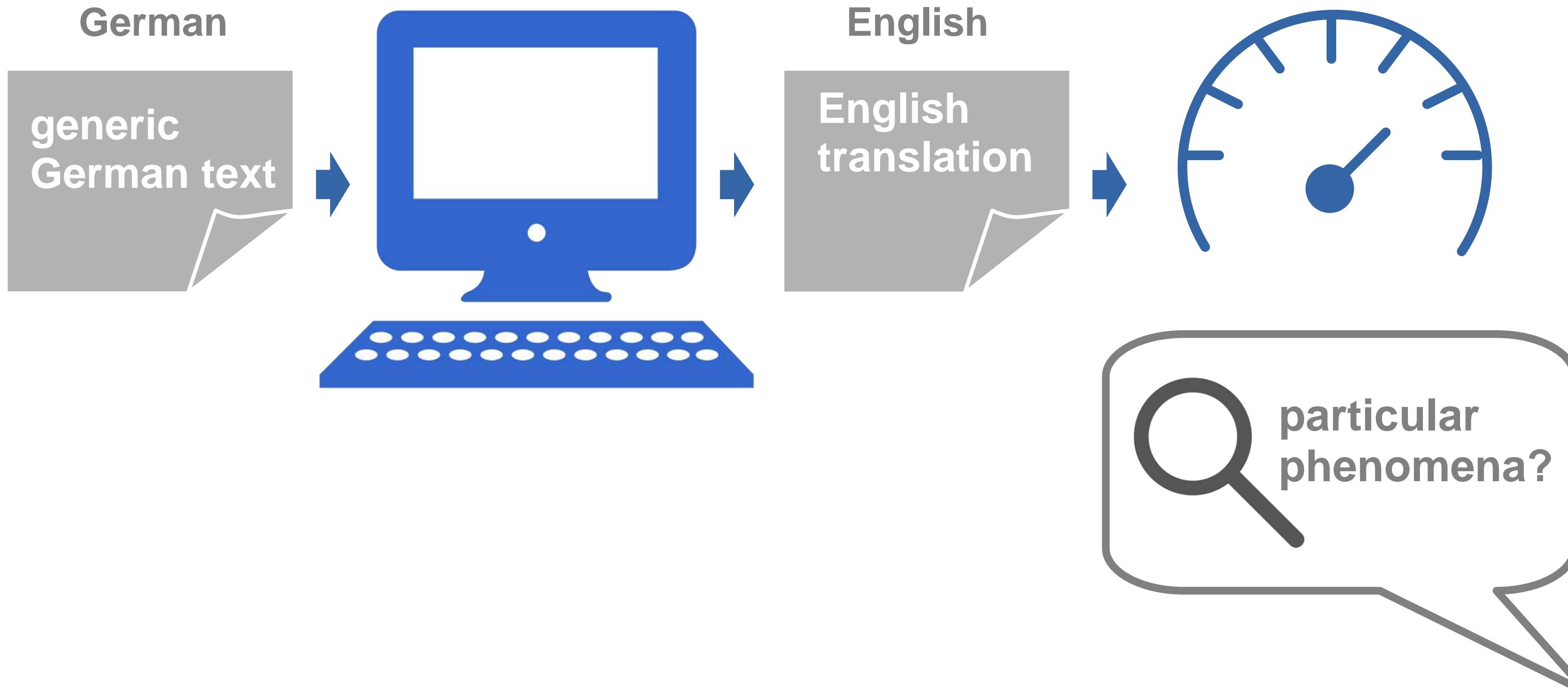
7. Sign language translation

Are we close to human parity?

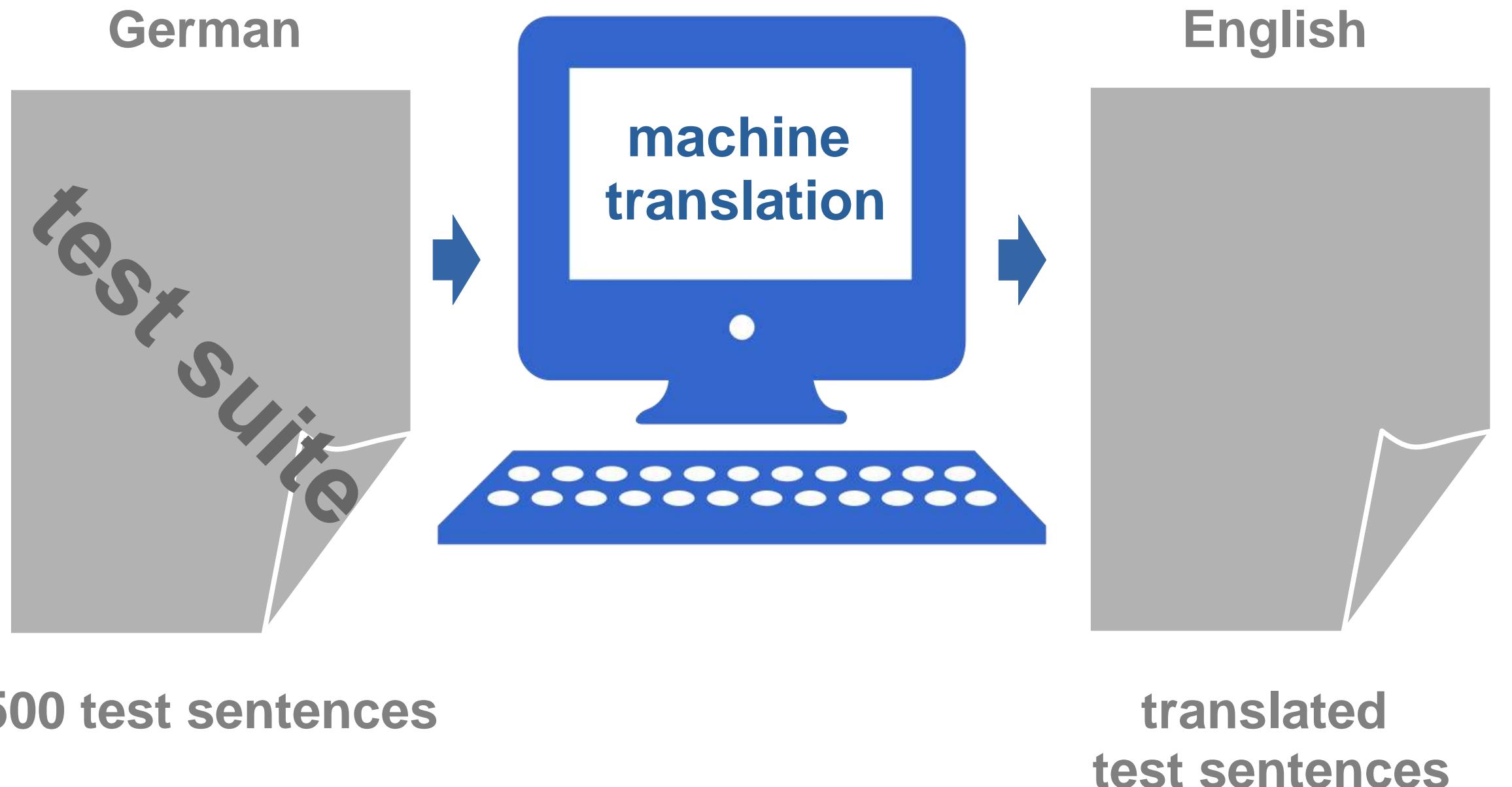


2016: Chris Manning: "Lots of remaining problems"

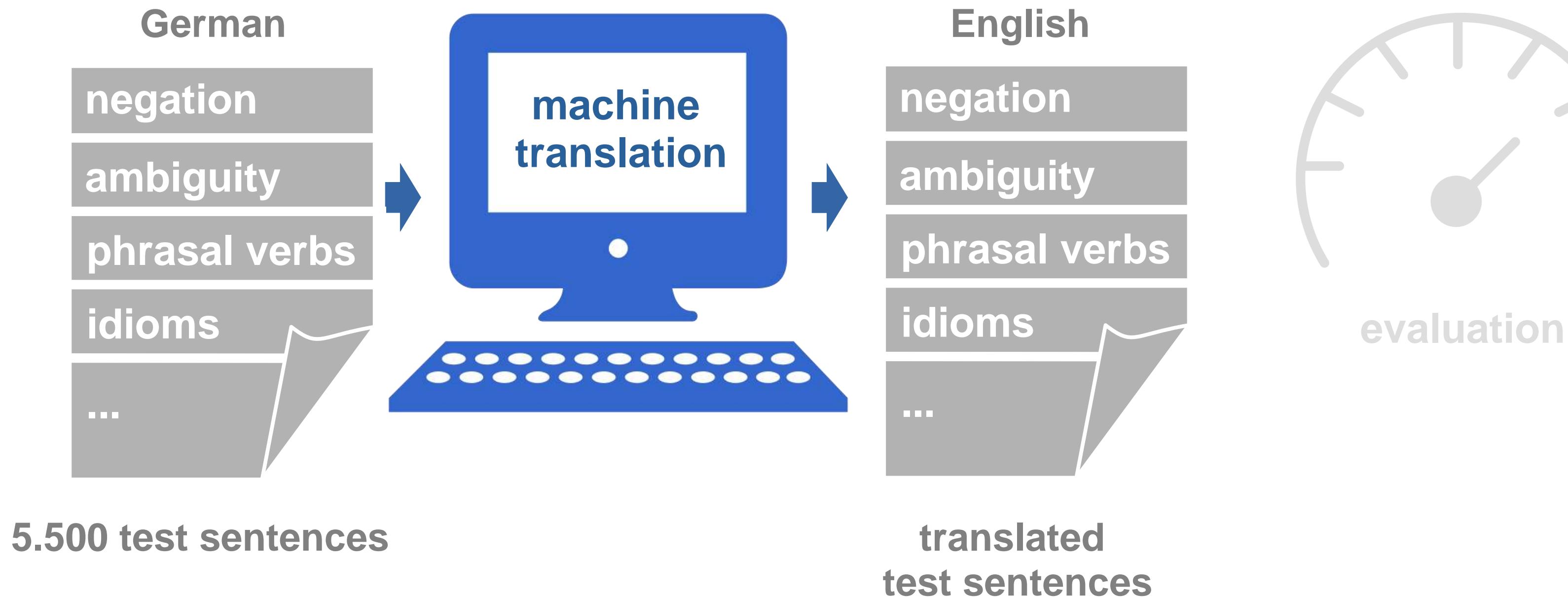
Evaluating overall system performance



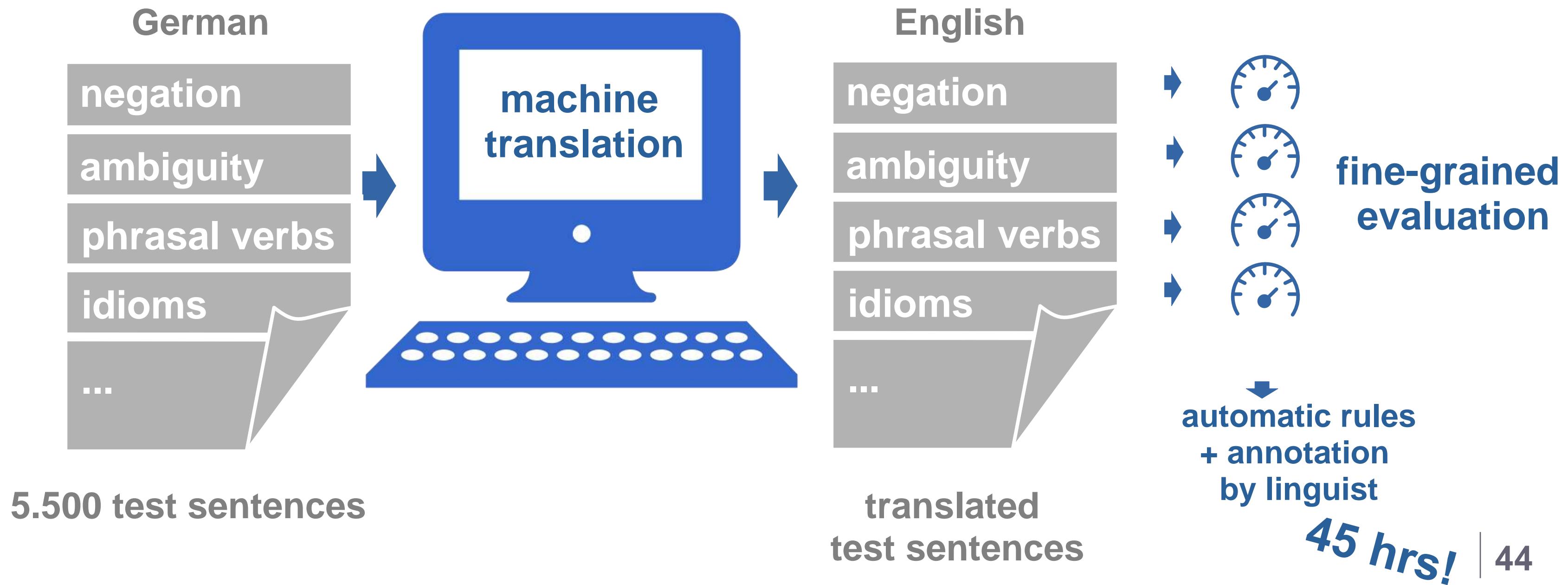
Evaluating overall system performance



Evaluating overall system performance



Evaluating overall system performance



107 phenomena

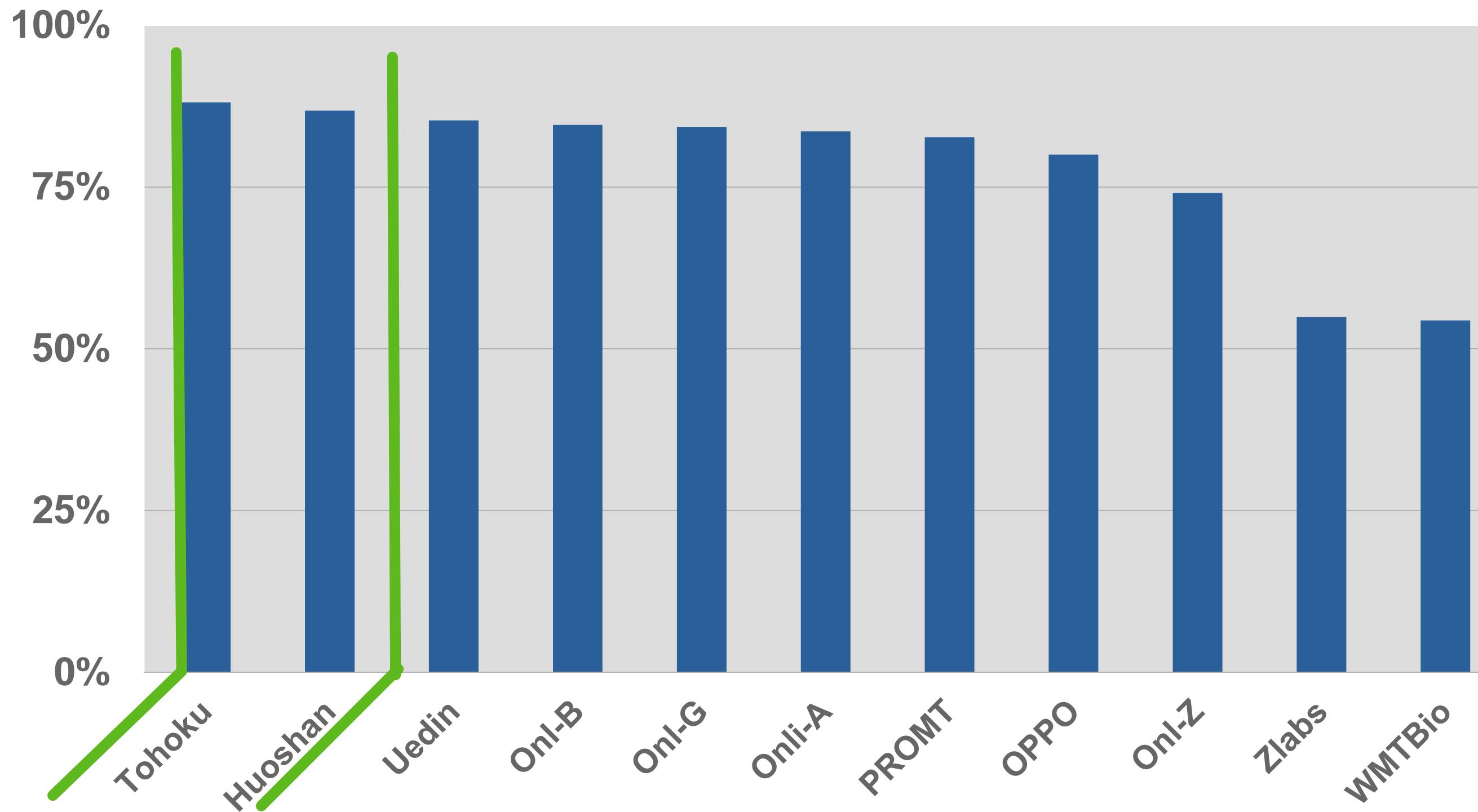
Lexical ambiguity	Prepositional MWE	Conditional	Modal - future I	Reflexive - pluperfect
Structural ambiguity	Verbal MWE	Ditransitive - future I	Modal - future I subjunctive II	Reflexive - pluperfect subjunctive II
Compound	Date	Ditransitive - future I subjunctive II	Modal - perfect	Reflexive - present
Phrasal verb	Domain-specific term	Ditransitive - future II	Modal - pluperfect	Reflexive - preterite
Gapping	Location	Ditransitive - future II subjunctive II	Modal - pluperfect subjunctive II	Reflexive - preterite subjunctive II
Right node raising	Measuring unit	Ditransitive - perfect	Modal - present	Transitive - future I
Sluicing	Proper name	Ditransitive - pluperfect	Modal - preterite	Transitive - future I subjunctive II
Stripping	Negation	Ditransitive - pluperfect subjunctive II	Modal - preterite subjunctive II	Transitive - future II
False friends	Coreference	Ditransitive - present	Modal negated - future I	Transitive - future II subjunctive II
Focus particle	External possessor	Ditransitive - preterite	Modal negated - future I subjunctive II	Transitive - perfect
Modal particle	Internal possessor	Ditransitive - preterite subjunctive II	Modal negated - perfect	Transitive - pluperfect
Question tag	Comma	Imperative	Modal negated - pluperfect	Transitive - pluperfect subjunctive II
Extended adjective construction	Quotation marks	Intransitive - fut107 phenomena ure I	Modal negated - pluperfect subjunctive II	Transitive - present
Extraposition	Adverbial clause	Intransitive - future I subjunctive II	Modal negated - present	Transitive - preterite
Multiple connectors	Cleft sentence	Intransitive - future II	Modal negated - preterite	Transitive - preterite subjunctive II
Pied-piping	Free relative clause	Intransitive - future II subjunctive II	Modal negated - preterite subjunctive II	Case government
Polar question	Indirect speech	Intransitive - perfect	Progressive	Mediopassive voice
Scrambling	Infinitive clause	Intransitive - pluperfect	Reflexive - future I	Passive voice
Topicalization	Object clause	Intransitive - pluperfect subjunctive II	Reflexive - future I subjunctive II	Resultative predicates
Wh-movement	Pseudo-cleft sentence	Intransitive - present	Reflexive - future II	
Collocation	Relative clause	Intransitive - preterite	Reflexive - future II subjunctive II	
Idiom	Subject clause	Intransitive - preterite subjunctive II	Reflexive - perfect	

107 phenomena

14 categories

ambiguity	multi-word ex	punctuation
composition	named entity	subordination
coordination	negation	verb valency
false friends	non-verbal aggr.	tense/mood
long distance & interrog.	function words	

11 systems – WMT20 German-English



More about the Test Suite

[1] Pierre Isabelle, Colin Cherry, and George Foster. 2017a. A Challenge Set Approach to Evaluating Machine Translation. 2017

[2] Avramidis et. al, Linguistic evaluation of German-English Machine Translation using a Test Suite, 2019

Machine translation

1. Introduction

- Definition and motivation
- History and types

2. Neural machine translation models

- RNN Encoder-decoder
- Attention-based NMT

3. Advanced techniques

- Subword units
- Multilingual machine translation
- Multimodal & speech translation

4. Evaluation

- Purpose of evaluation
- Users of evaluation
- Evaluation approaches

5. Fine-grained evaluation

- Test suites

6. Quality estimation

- Feature-based model
- Neural predictor-estimator

7. Sign language translation

Machine translation

input

Darüber soll am Anfang kommenden
Woche der Bundestag abstimmen.

system 1

This is to be voted on at the beginning of next week.

0.7

system 2

The parliament is supposed to vote for it
beginning of next week

0.9

system 3

About this voting should begining next week

0.3

reference

~~The parliament should vote for this
at the beginning of next week~~

Machine learning to predict
scores of MT “quality”

- focus on one sentence at a time
- real-time use
(don't use reference)
- predict a metric of quality
(e.g. the human edit rate)

Various types of Quality Estimation

Linear / feature based model:

- analyze sentences with automatic tools
- generate numerical indicators of quality (features)
- use these to train a regressor/classifier given existing labels

(Blatz et. al, Specia et. al 2009)

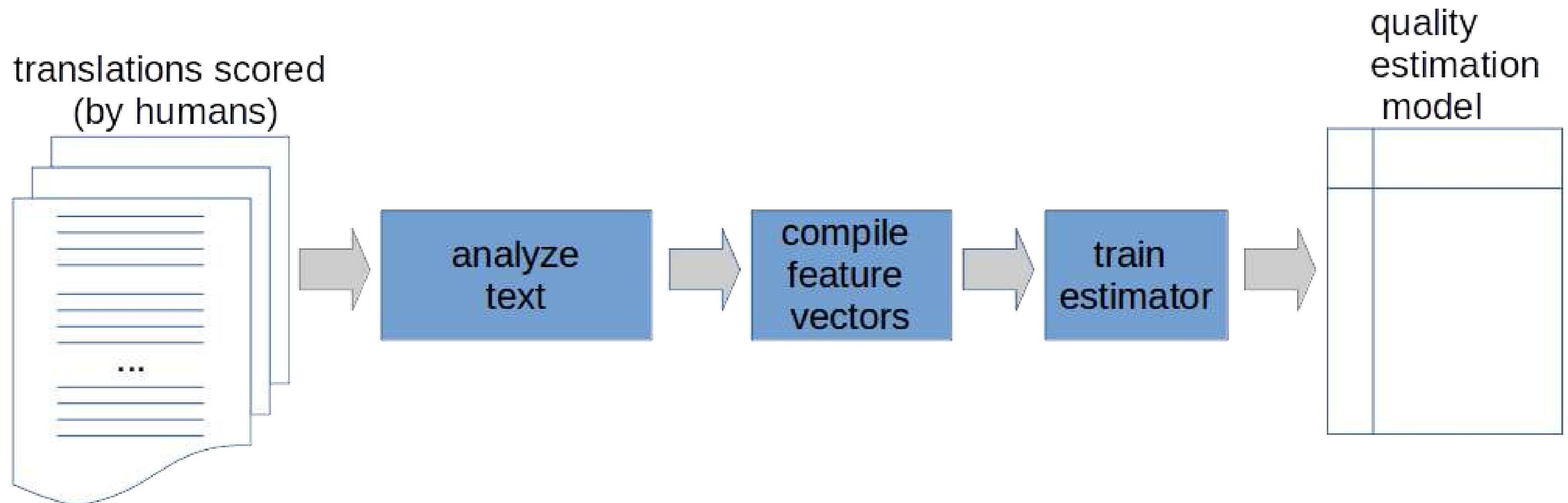
Neural:

- use neural models to perform automatic post-editing and score with the existing translation (Martins et. al 2017)
- train a joined “predictor-estimator” neural model (Kim et. al 2017)

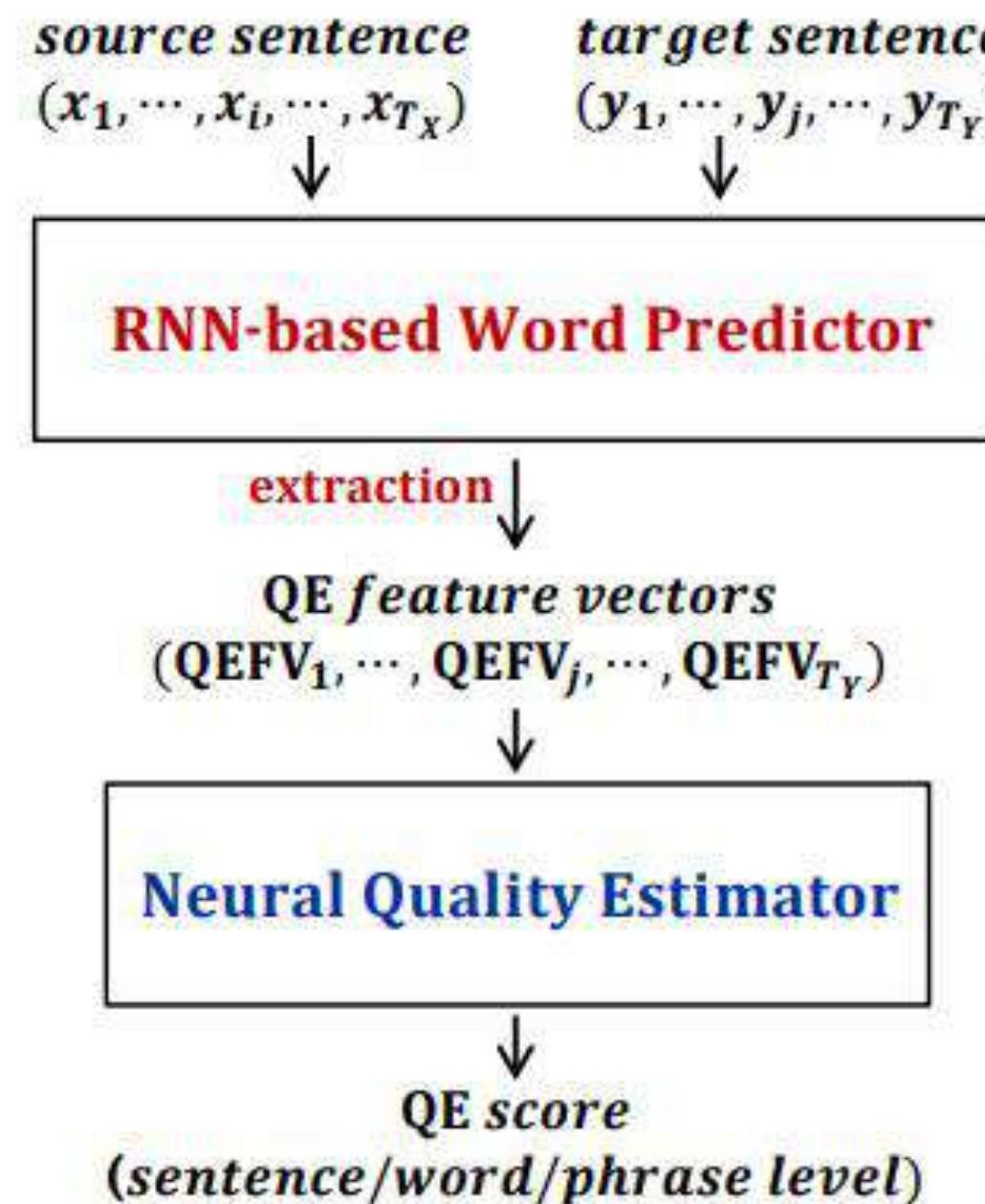
Challenge:

Systems are getting more efficient by the time, difficult to distinguish and predict machine translation errors

Linear, feature-based model



Predictor-estimator



Source: Kim et al., WMT2017

Machine translation

1. Introduction

- Definition and motivation
- History and types

2. Neural machine translation models

- RNN Encoder-decoder
- Attention-based NMT

3. Advanced techniques

- Subword units
- Multilingual machine translation
- Multimodal & speech translation

4. Evaluation

- Purpose of evaluation
- Users of evaluation
- Evaluation approaches

5. Fine-grained evaluation

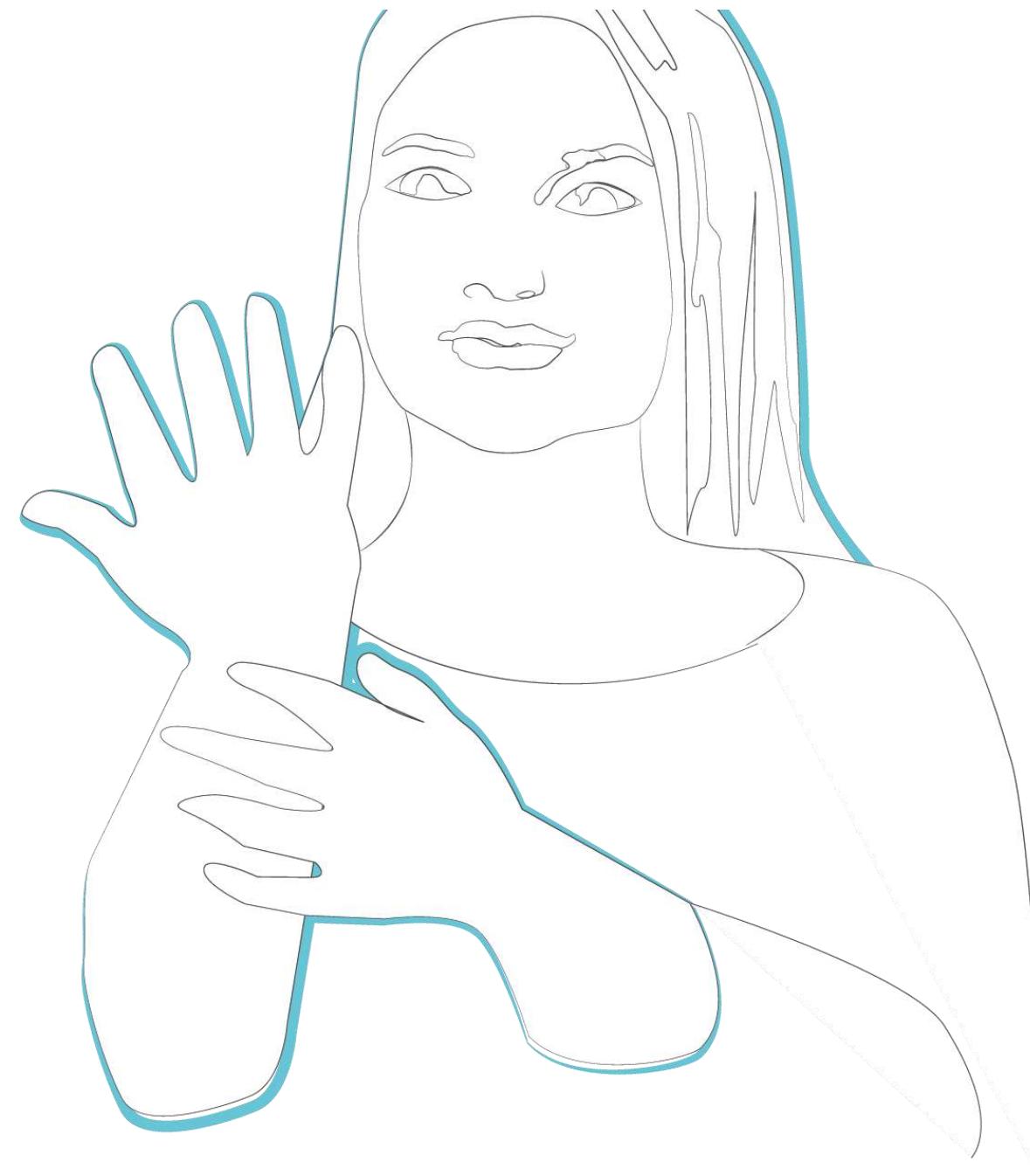
- Test suites

6. Quality estimation

- Feature-based model
- Neural predictor-estimator

7. Sign language translation

Sign Language



The sign language of the deaf is an independent visual language, which has been developed over the centuries in the everyday communication of deaf people.

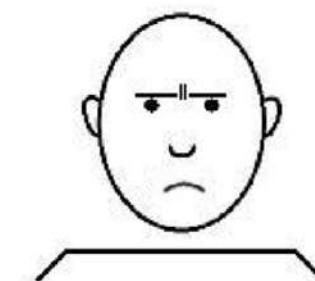
Building blocks of sign language



Papa



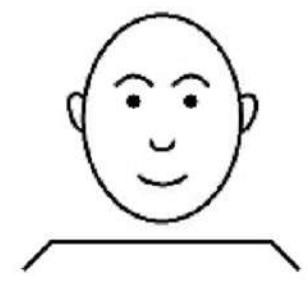
Herbst



Angst



(Faust schlägt an Brust)



Mut

(Faust schlägt an Brust)

a) manual

- hands (hand shape & hand position), arms
- executing position
- movement

b) non-manual

- facial expression (facial expression)
- direction of eyes
- head direction
- posture (especially of the upper body)
- Mouth image

Quick facts

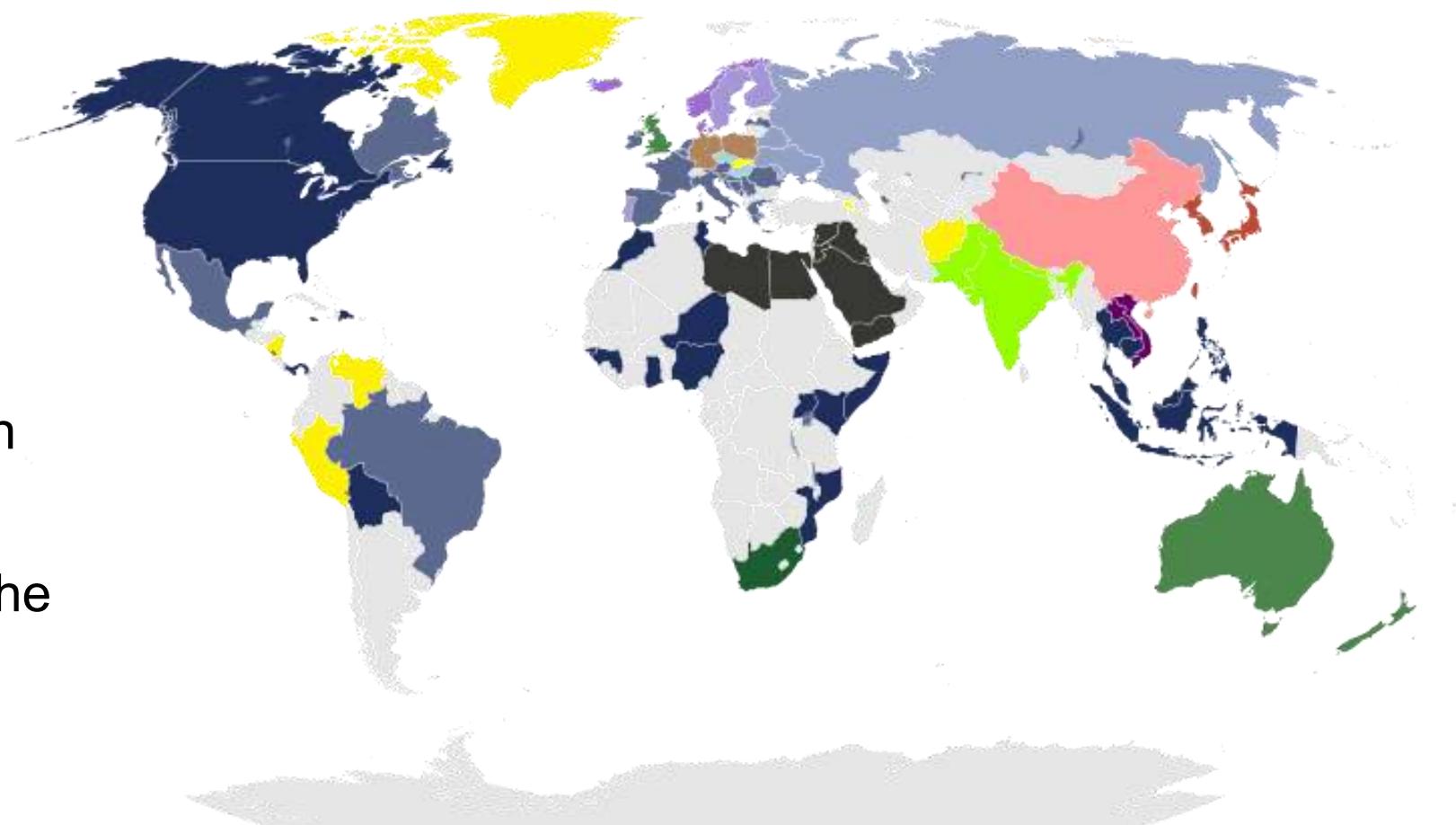
- 0.1 % of the German population is deaf. This amounts to about 80.000 people.
- Every day 2 deaf children are born in Germany.

Sign languages around the world

- **Common misperception: all sign languages are the same.
 Nope!**
- ~130 national sign languages are known, while ~60 have been analytically processed
- American Sign Language (ASL) - different than the British or the Australian!

German sign language regional dialects

Berliner, Hamburger, Münchner, Frankfurter Dialekt Ruhrgebiet, etc. (~75% of vocabulary overlap)



Quick facts

- Sign language is equivalent to spoken language.
 - equally suitable to express meanings and feelings
 - possible to express and discuss complex matters
 - it consists of a comprehensive vocabulary and an elaborate grammar.
 - not invented by a person or institution (like for example esperanto), but was continuously and organically developed by its native speakers.
- Not limited to visible things, that can be visualized with hand signs and gestures.
 - Signs have a complex substructure, that can be analytically represented by rules that connect the **shape of the hand, orientation of the hand, position relative to the body and motion.**

History of automatic sign language translation



- 1977:** Research project successfully matched English letters from a keyboard to ASL manual alphabet letters which were simulated on a robotic hand.
- 1996:** Recognition method with gloves (only 20 gestures) and (shallow) neural networks
- 2005:** Recognition method using cameras was proposed (but not implemented)
- 2012:** First work on (Chinese) sign language translation with Kinect (still low accuracy)
- 2016:** First system using deep learning to recognize gestures on videos (but does not yet produce text)
- 2018:** First end-to-end system using techniques similar to text Machine Translation (but only trained on weather forecasts!)

From text to signing avatar

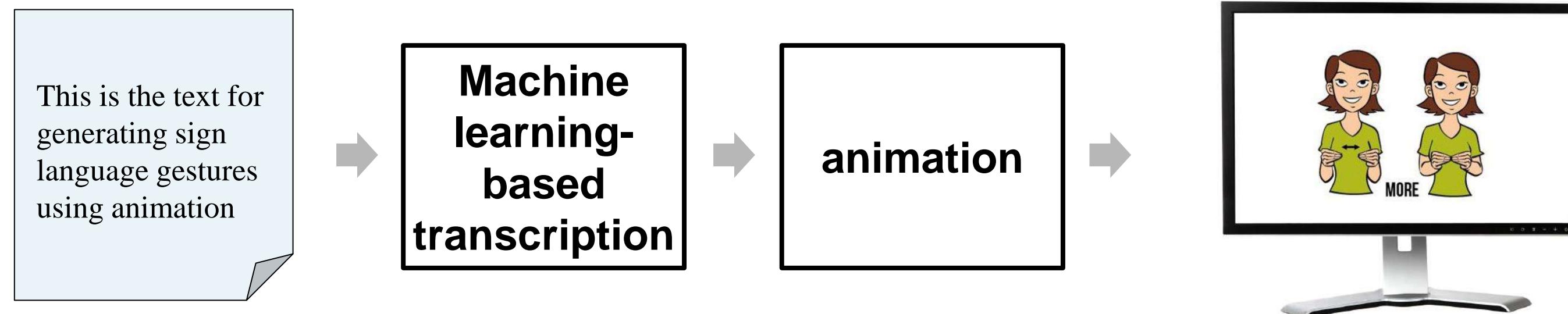
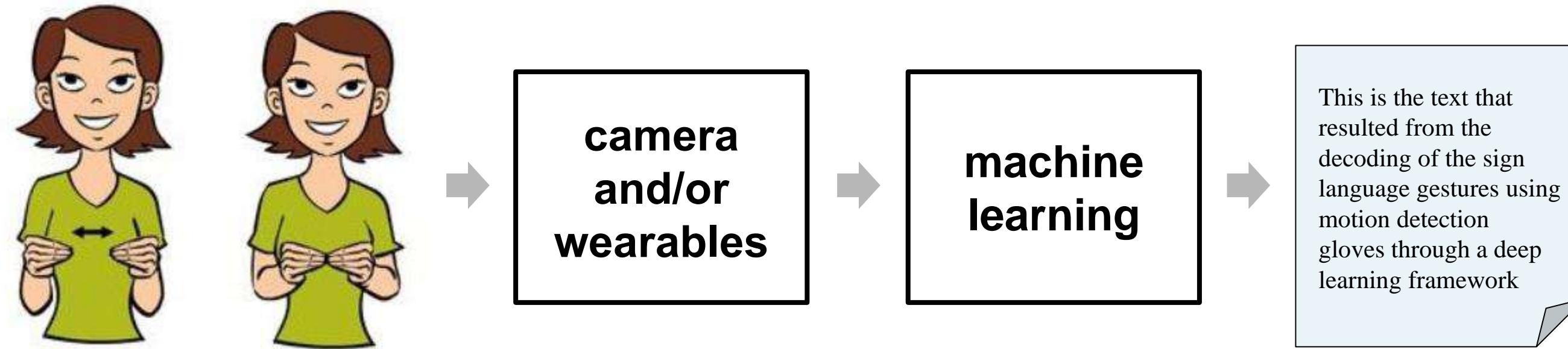


Image source: babysignlanguage.com

From sign language to text



text / speech

Image source: babysignlanguage.com

Different granularities: Finger alphabet



Isolated sign language translation



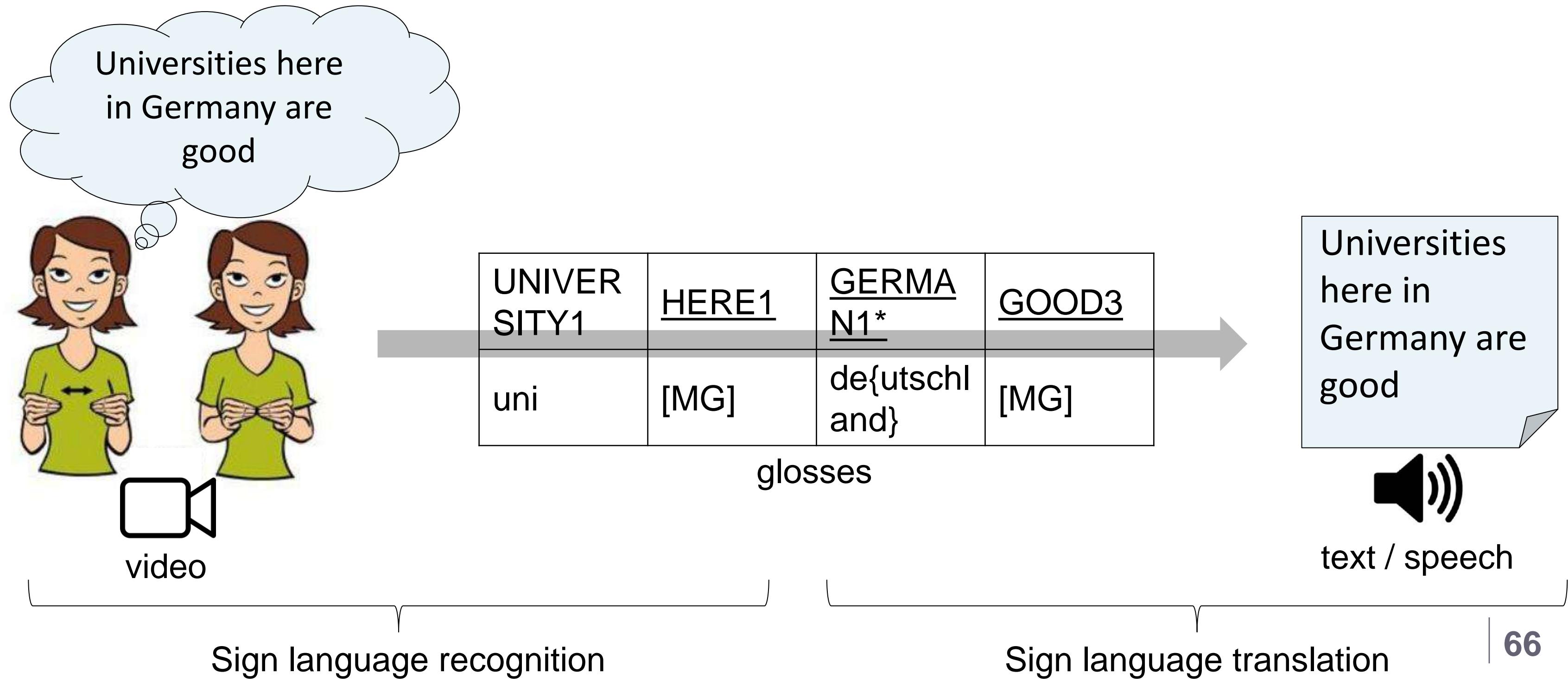
Isolated sign language translation



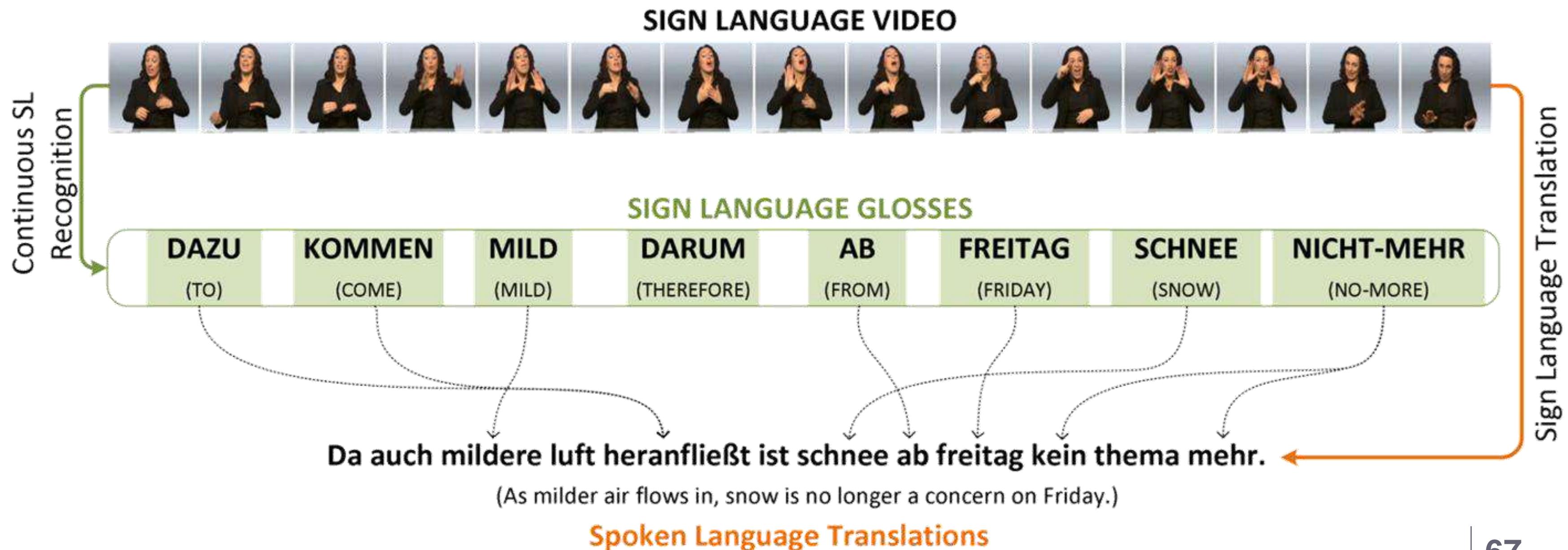
Isolated sign language translation



Continuous sign language translation

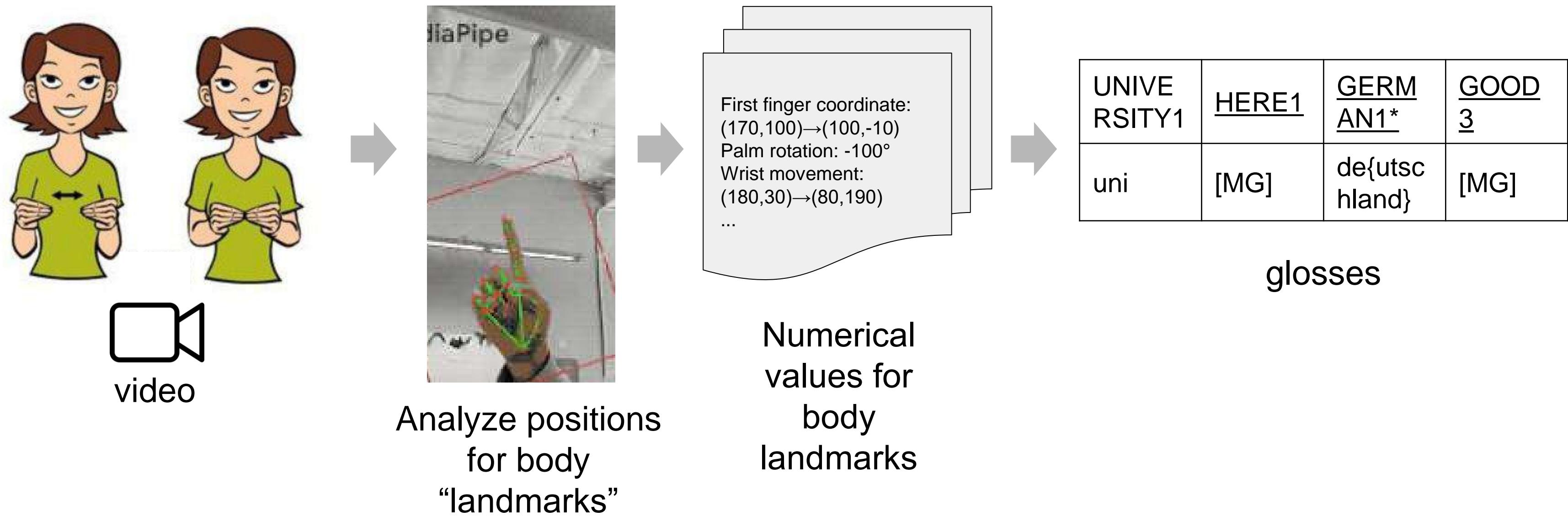


Continuous sign language translation



Source: Camgoz et. al, "Neural Sign Language Translation", RWTH 2019

Sign language recognition via body recognition



Transfer Learning

Salar Mohtaj | DFKI

Transfer learning

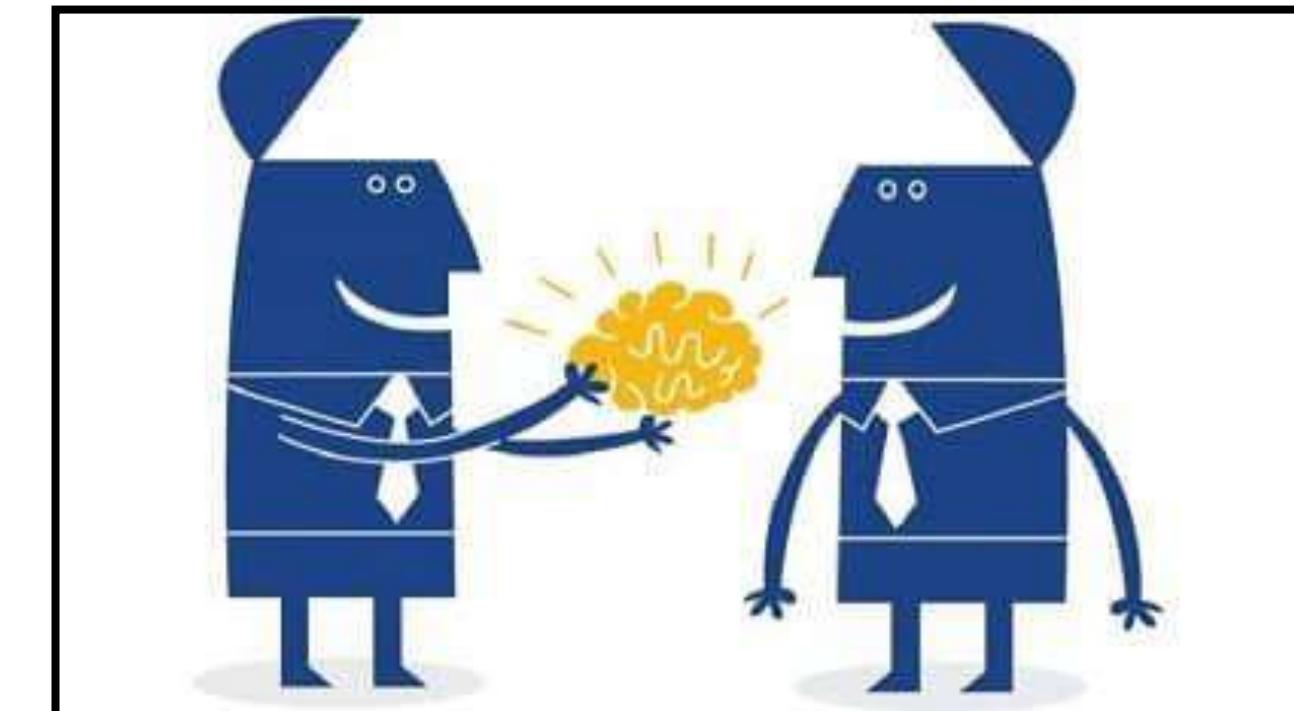
- What is transfer learning
- Motivation for transfer learning
- Different approaches for transfer learning
- BERT

Transfer learning

- What is transfer learning
- Motivation for transfer learning
- Different approaches for transfer learning
- BERT

What is transfer learning

- Humans have an ability to transfer knowledge across tasks
- What we acquire as knowledge while learning about a task, we utilize in the same way to solve related tasks
- Knowledge about how to change setting in Windows → how to change setting in the other OSs
- In many cases we transfer and leverage our knowledge from what we have learnt in the past!



<https://jeanvitor.com>

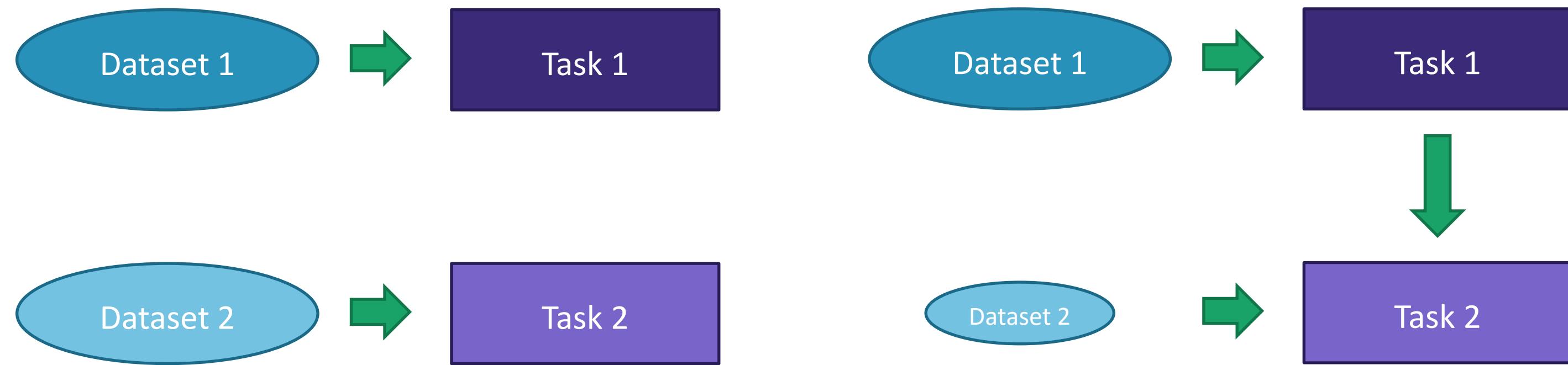
What is transfer learning

- Transfer learning is a learning procedure in which representations learned on a source task are **transmitted** to improve learning on the target task
- In transfer learning, the knowledge of an already trained **machine learning** model is applied to a different problem
- Transfer learning is a machine learning technique where a model trained on one task is re-purposed on a second related task
- With transfer learning, we basically try to exploit what has been learned in one task to improve generalization in another.
- We transfer the weights that a network has learned at a task (task A) to a new task (task B)

What is transfer learning

- Suppose we have a sentiment analysis task for the domain of customer reviews
- We have enough labelled data for this supervised task and train a model for it
- It would work well in different problems related to customer review
- But, as soon as we apply it for the same task in another domain, such as stock market, the performance would decrease
- The idea is to tune the model that is trained on the first domain, to work well also in the second domain

What is transfer learning



What is transfer learning

- As a formal definition:
- Given a source domain D_s , a corresponding source task T_s , as well as a target domain D_t and a target task T_t , the objective of transfer learning is to enable us to learn the target conditional probability distribution $P(Y_t|X_t)$ in D_t with the information gained from D_s and where $D_s \neq D_t$ or $T_s \neq T_t$.

Transfer learning

- What is transfer learning
- Motivation for transfer learning
- Different approaches for transfer learning
- BERT

Motivation for transfer learning

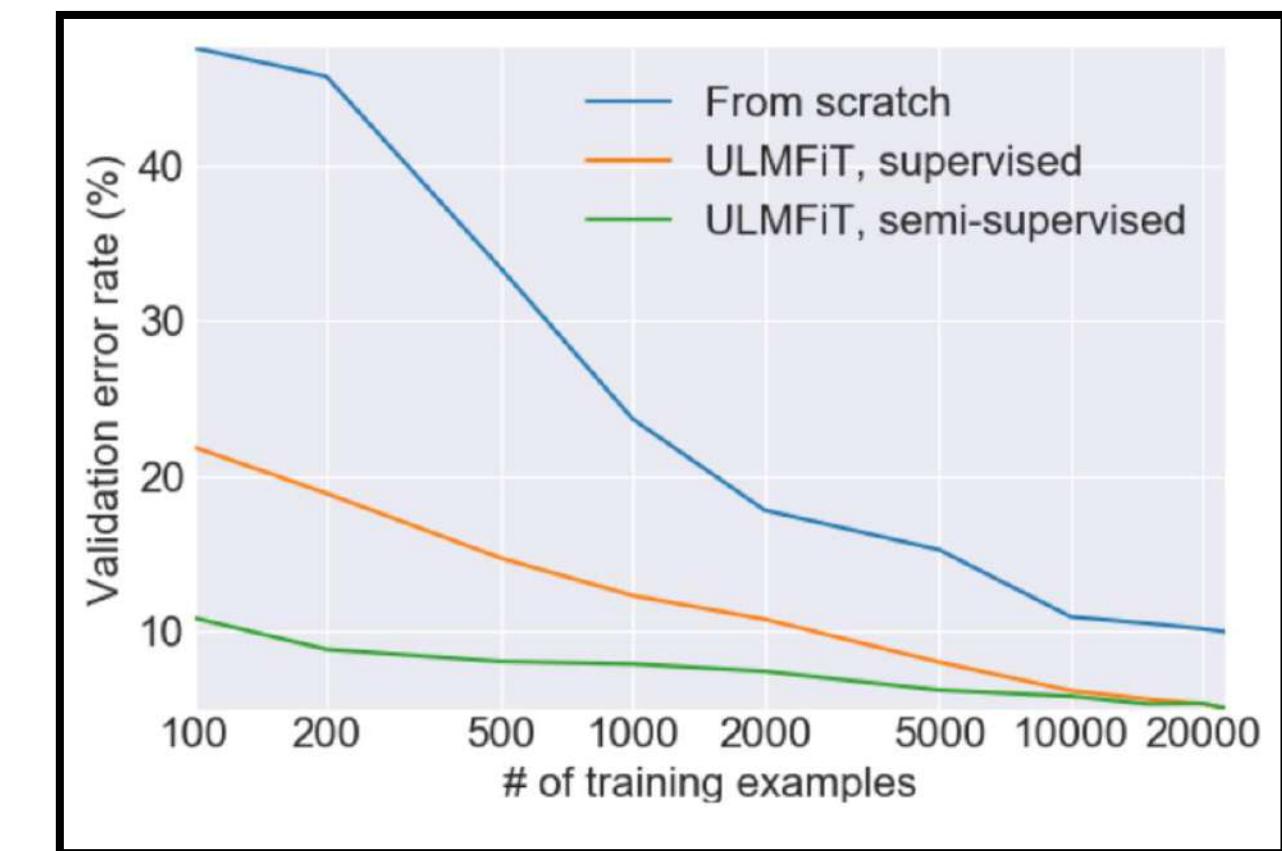
- Traditional learning is isolated and occurs purely based on specific tasks, datasets and training separate isolated models on them
- No knowledge is retained which can be transferred from one model to another
- In transfer learning, you can leverage knowledge (e.g., features, weights) from previously trained models for training newer models
- Tackle problems like having less data for the newer task!

Motivation for transfer learning

- Transfer learning has several benefits, but the main advantages are:
 - Saving training time
 - Better performance
 - Not needing a lot of data

Motivation for transfer learning

- One of the main benefits of pretraining is that it reduces the need for annotated data
- Transfer learning based models could achieve similar performance compared to a simple model with 10x fewer examples (Howard and Ruder, 2018).



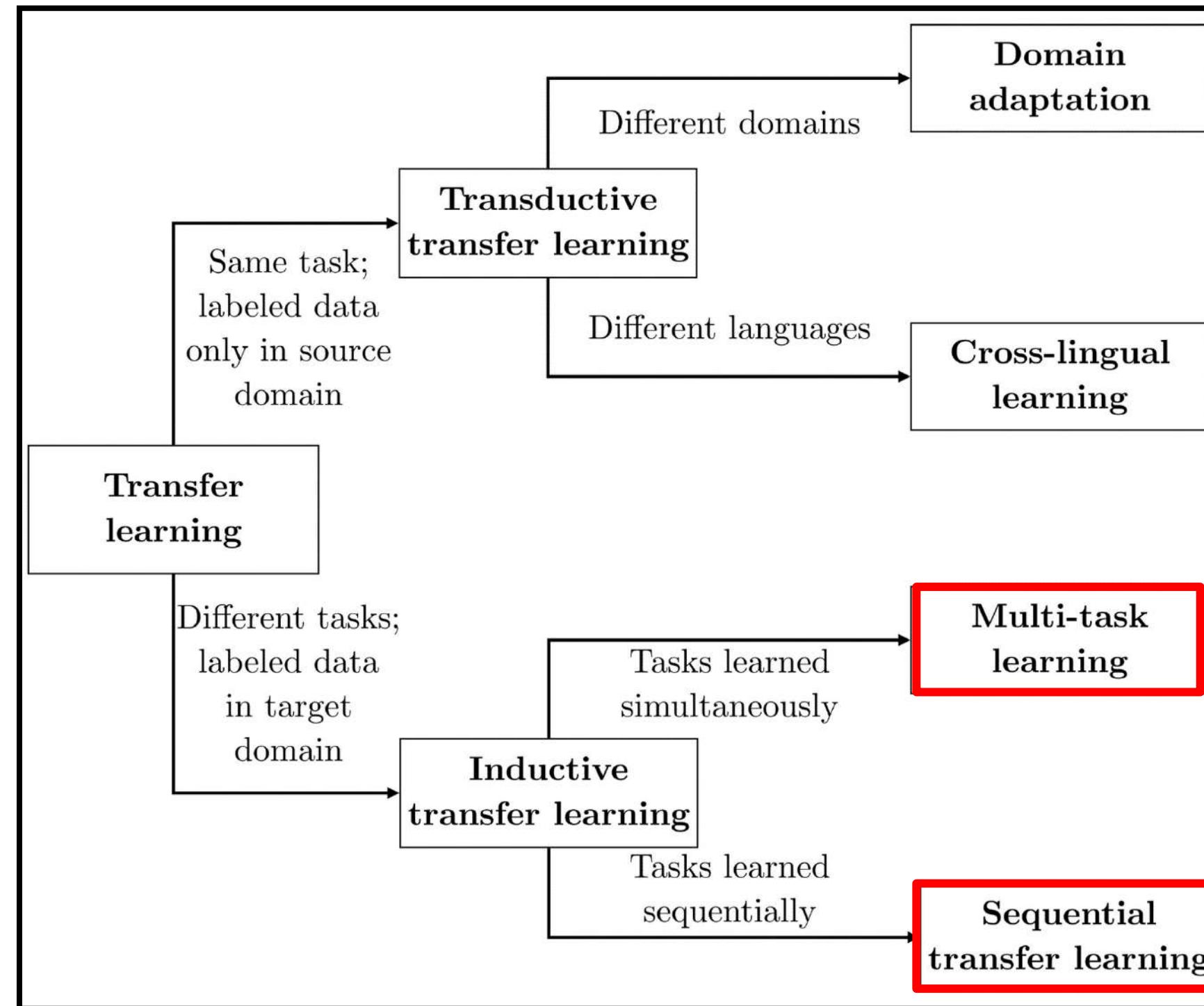
Motivation for transfer learning

- Most of the labeled text datasets are not big enough to train deep neural networks because these networks have a huge number of parameters and training such networks on small datasets will cause overfitting

Transfer learning

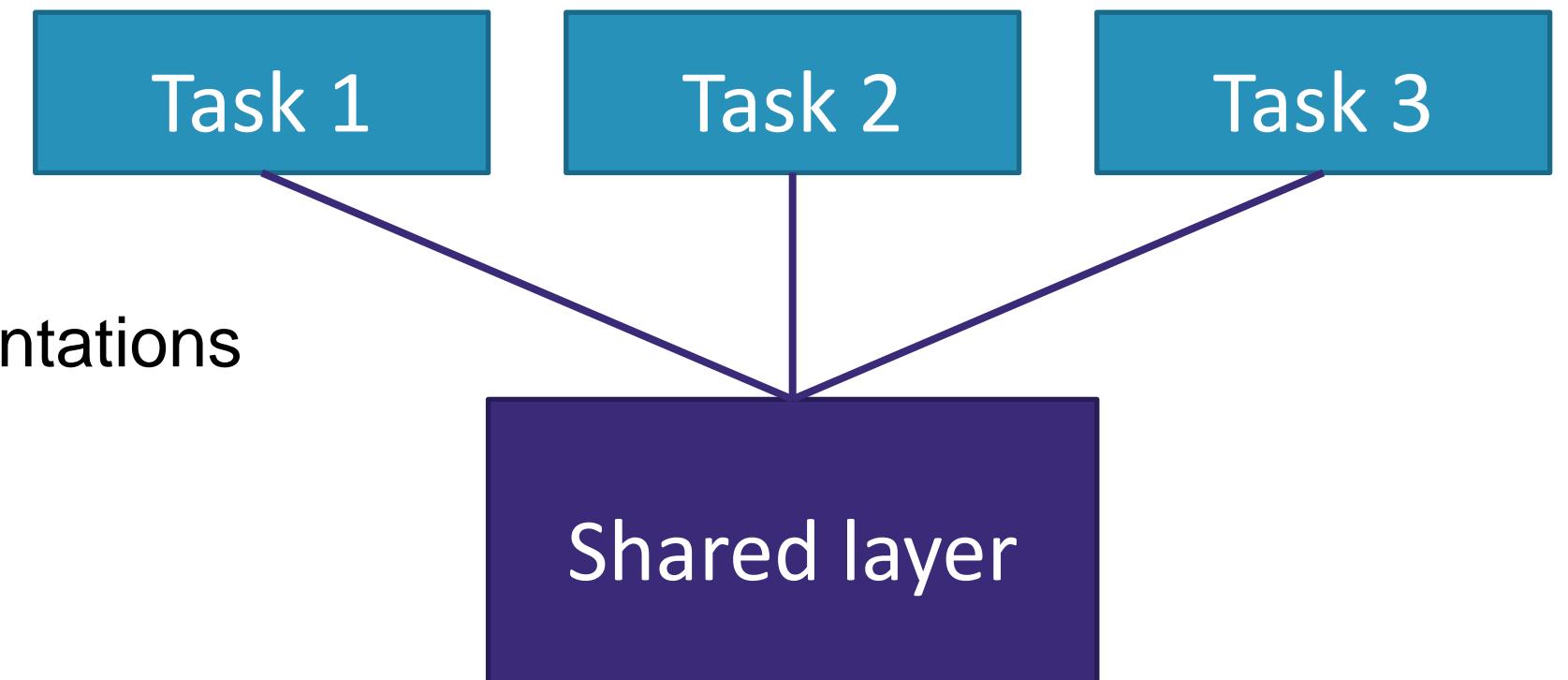
- What is transfer learning
- Motivation for transfer learning
- Different approaches for transfer learning
- BERT

Different approaches for transfer learning



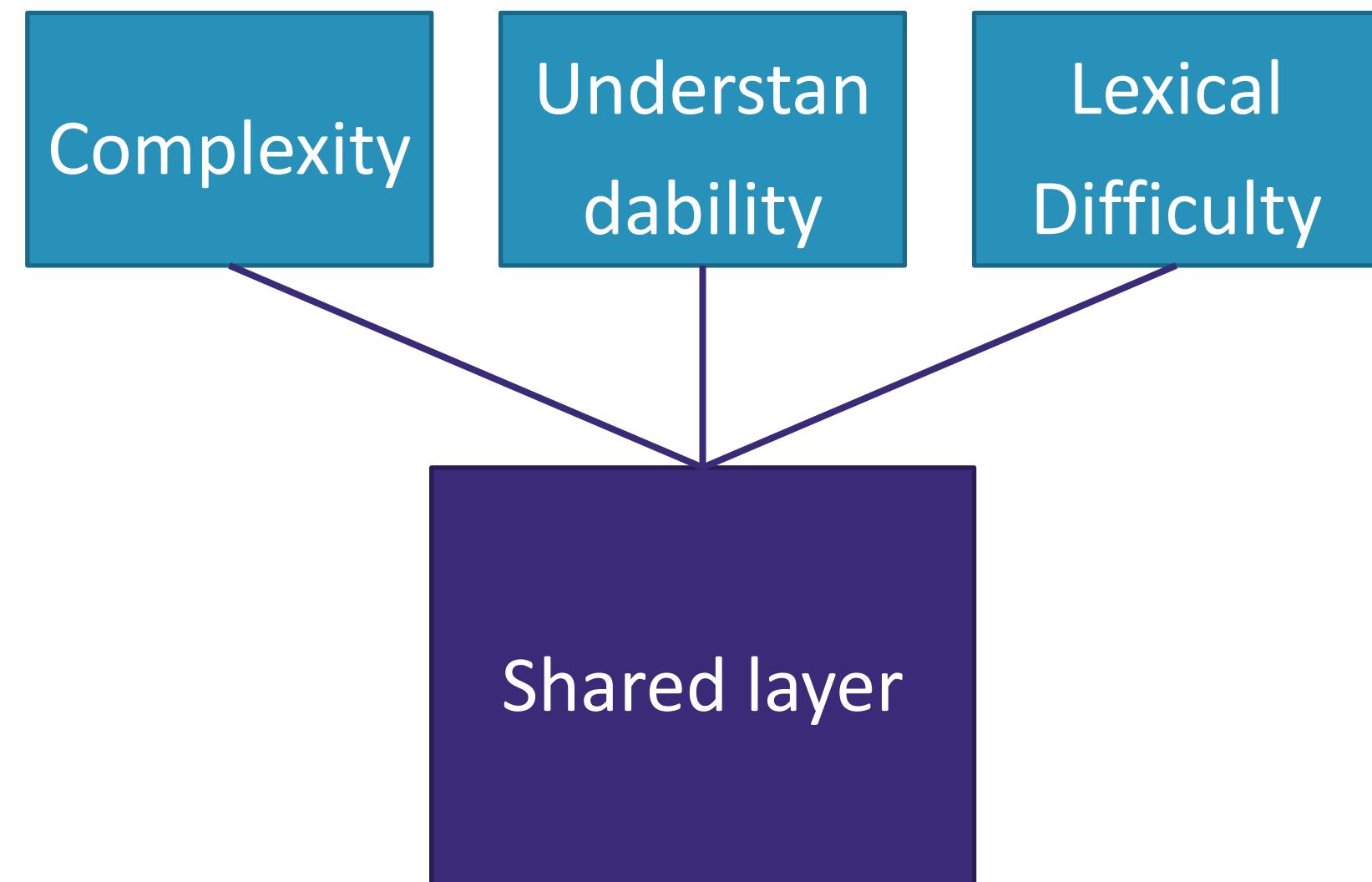
Different approaches for transfer learning

- Multi-task learning
 - Multi-task learning (MTL) is a subfield of machine learning in which multiple tasks are simultaneously learned by a shared model
 - Such approaches offer advantages like:
 - Improved data efficiency
 - Reduced overfitting through shared representations
 - fast learning



Different approaches for transfer learning

- Subjective assessment of German text complexity



Different approaches for transfer learning

- Different approaches for sequential transfer learning
 - Training a model to reuse it
 - Fine-tuning a pre-trained model
 - Feature extraction

Different approaches for transfer learning

- Training a model to reuse it
 - Imagine you want to solve the task of sentiment analysis for a specific domain
 - But you don't have enough data to train a deep neural network
 - One way around this is to find a related task B with an abundance of data
 - Train a model on task B and use the model as a starting point for solving the task of sentiment analysis for the target domain
 - The trained model can be fine-tuned on the task B

Different approaches for transfer learning

- Fine-tuning a pre-trained model
 - There are a lot of pre-trained models for NLP which are trained for different tasks
 - One popular approach for transfer learning would be using one of these pre-trained models
 - The model can fine-tuned based on a small data which is available for the target task

Different approaches for transfer learning

- Feature extraction
 - Pre-trained models may be used as feature extraction models
 - Here, the output of the model is used as input to a new classifier model
 - Here there is no fine-tuning of the weights

Different approaches for transfer learning

- Using a pre-trained model
 - Empirically, language modelling works better than other pretraining tasks
 - Language modelling is a very difficult task, even for humans
 - To have any chance at solving this task, a model is required to learn about syntax, semantics, as well as certain facts about the world
 - Given enough data, a large number of parameters, and enough compute, a model can do a reasonable job

Different approaches for transfer learning

- Advantages of language modelling is that
 - It does not require any human annotation
 - Many languages have enough text available to learn reasonable models
 - Language model is enable to learn both sentence and word representations

Different approaches for transfer learning

- Using a pre-trained model
 - **BERT**
 - GPT-3
 - ELMo
 - XLNet
 - ALBERT
 - ULMFiT
 - RoBERTa

Different approaches for transfer learning

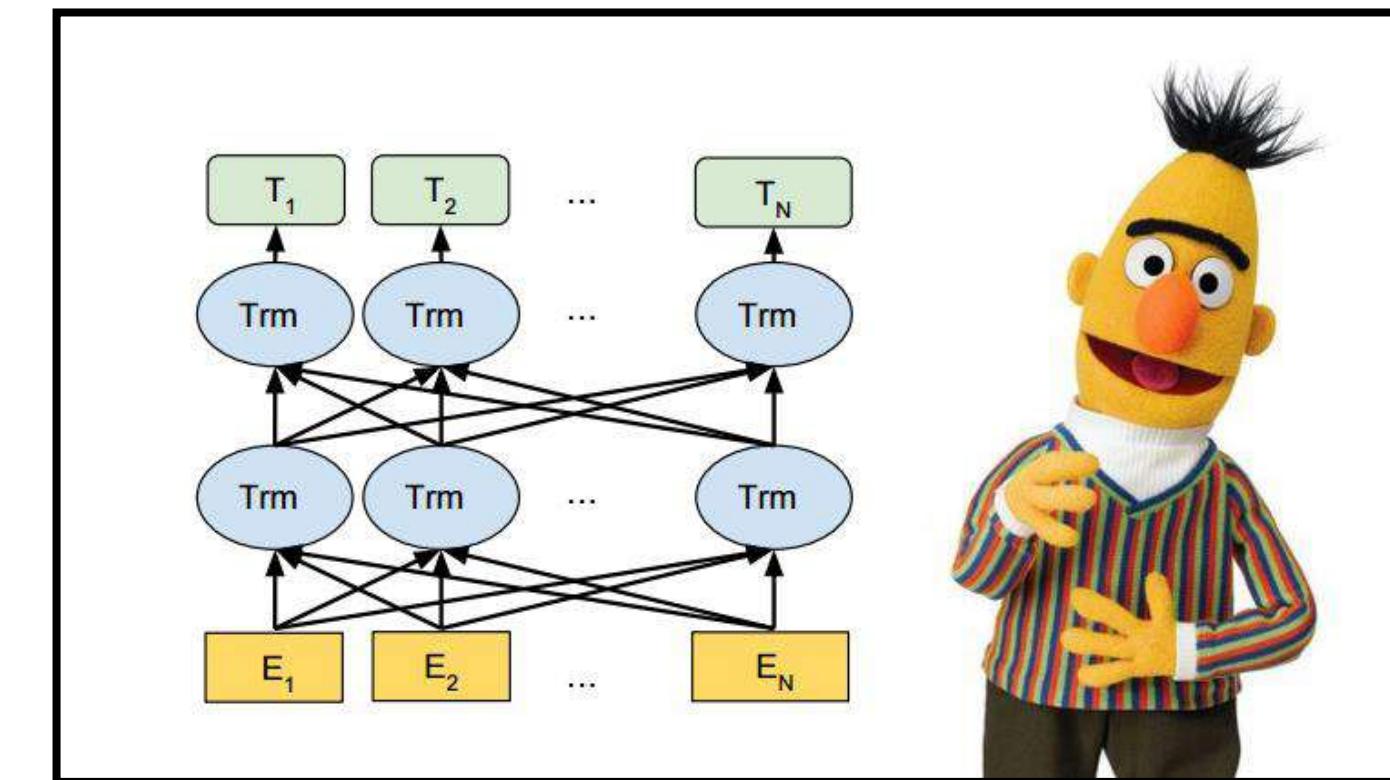
- Important questions on using pre-trained models:
 - What to transfer
 - Which part of the knowledge can be transferred from the source to the target
 - Which portion of knowledge is source-specific and what is common between the source and the target
 - When to transfer
 - How to transfer

Transfer learning

- What is transfer learning
- Motivation for transfer learning
- Different approaches for transfer learning
- BERT

BERT

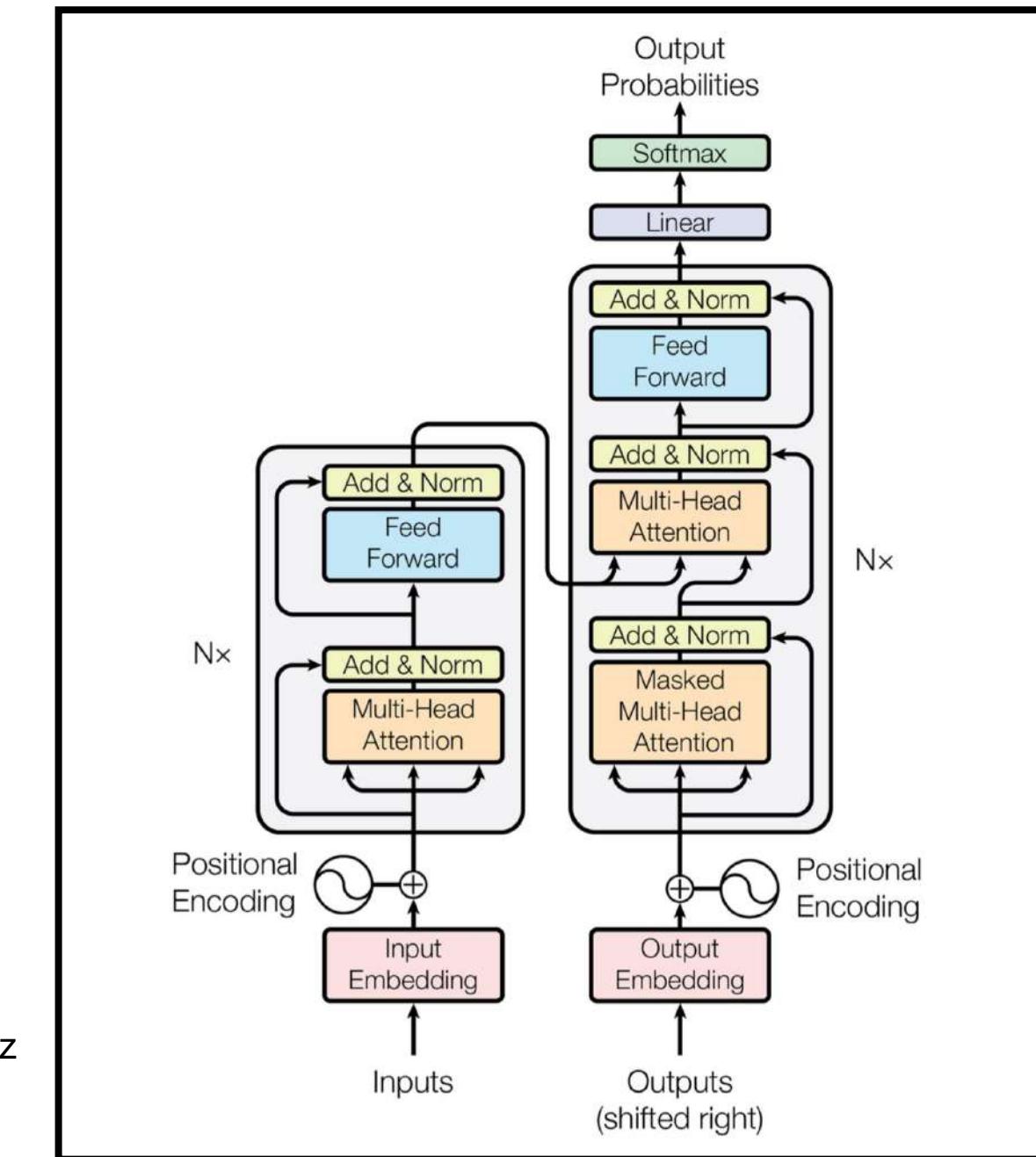
- BERT (Bidirectional Encoder Representations from Transformers)
 - A language model which is bidirectionally trained
 - Have a deeper sense of language context and flow compared to the single-direction language models



BERT

- The transformer
 - It a new architecture that uses the attention-mechanism
 - Like LSTM, Transformer is an seq2seq architecture but it differs because it does not imply any Recurrent Networks (GRU, LSTM, etc.)
 - The architecture which is only with attention-mechanisms without any RNN (Recurrent Neural Networks) improved the results in many NLP tasks includes translation task

Vaswani, Ashish, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. "Attention is all you need." *arXiv preprint arXiv:1706.03762* (2017).



BERT

- Instead of predicting the next word in a sequence, BERT makes use of a novel technique called masked LM
 - From each input sequence 15% of the tokens are processed as follows:
 - with 0.8 probability the token is replaced by [MASK]
 - with 0.1 probability the token is replaced by another random token
 - with 0.1 probability the token is unchanged

BERT

- The input is composed of two sentences
- These two sentences A and B are separated with the special token [SEP]
- 50% of the time B is the actual next sentence and 50% of the time is a random sentence

Input= [CLS] the man went to [MASK] store [SEP] he bought a gallon [MASK] milk [SEP]

Label=IsNext

Input=[CLS] the man [MASK] to the store [SEP] penguin [MASK] are flight ##less birds [SEP]

Label=NotNext

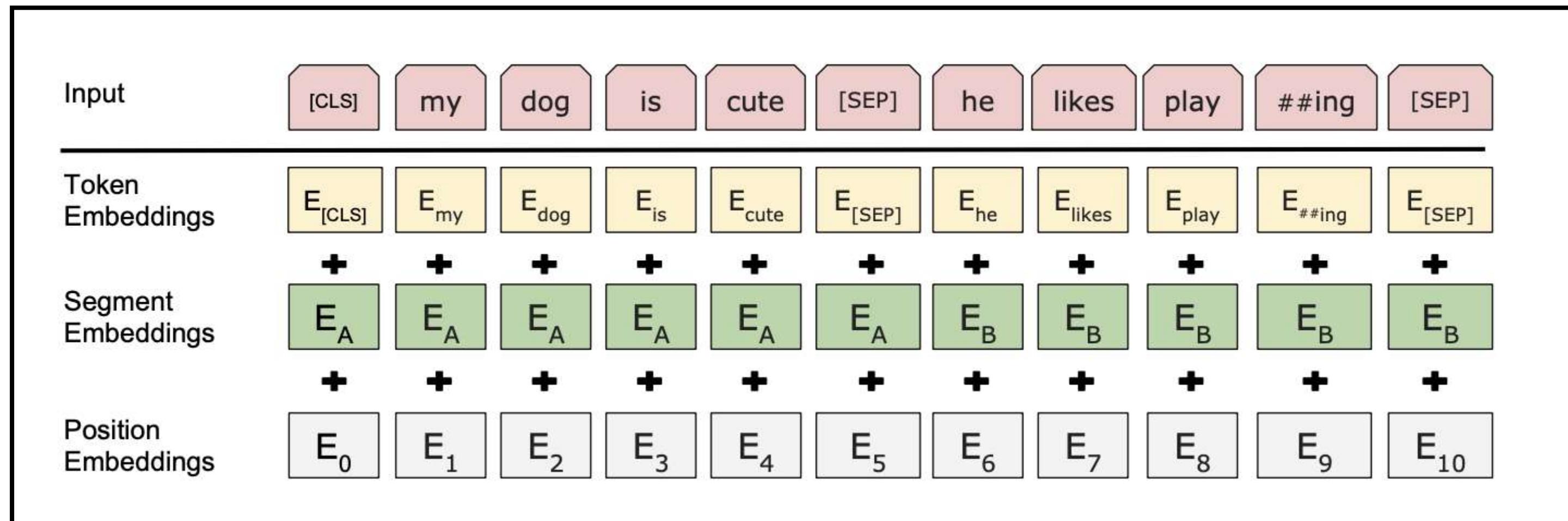
Input= [CLS] the man went to [MASK] store [SEP] he bought a gallon [MASK] milk [SEP]

Label=IsNext

Input=[CLS] the man [MASK] to the store [SEP] penguin [MASK] are flight ##less birds [SEP]

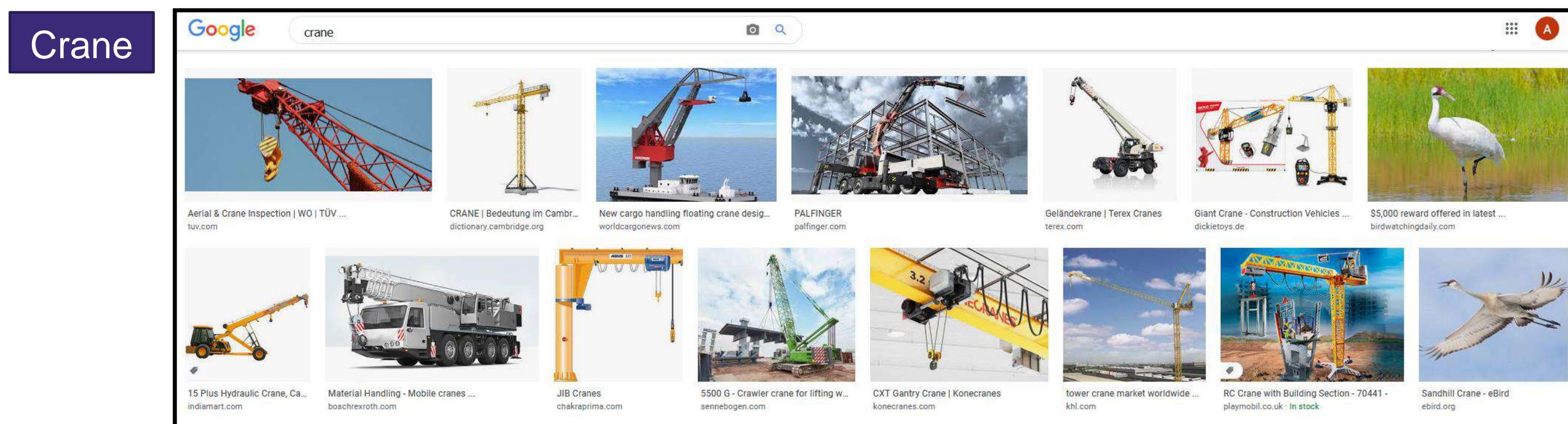
Label=NotNext

BERT



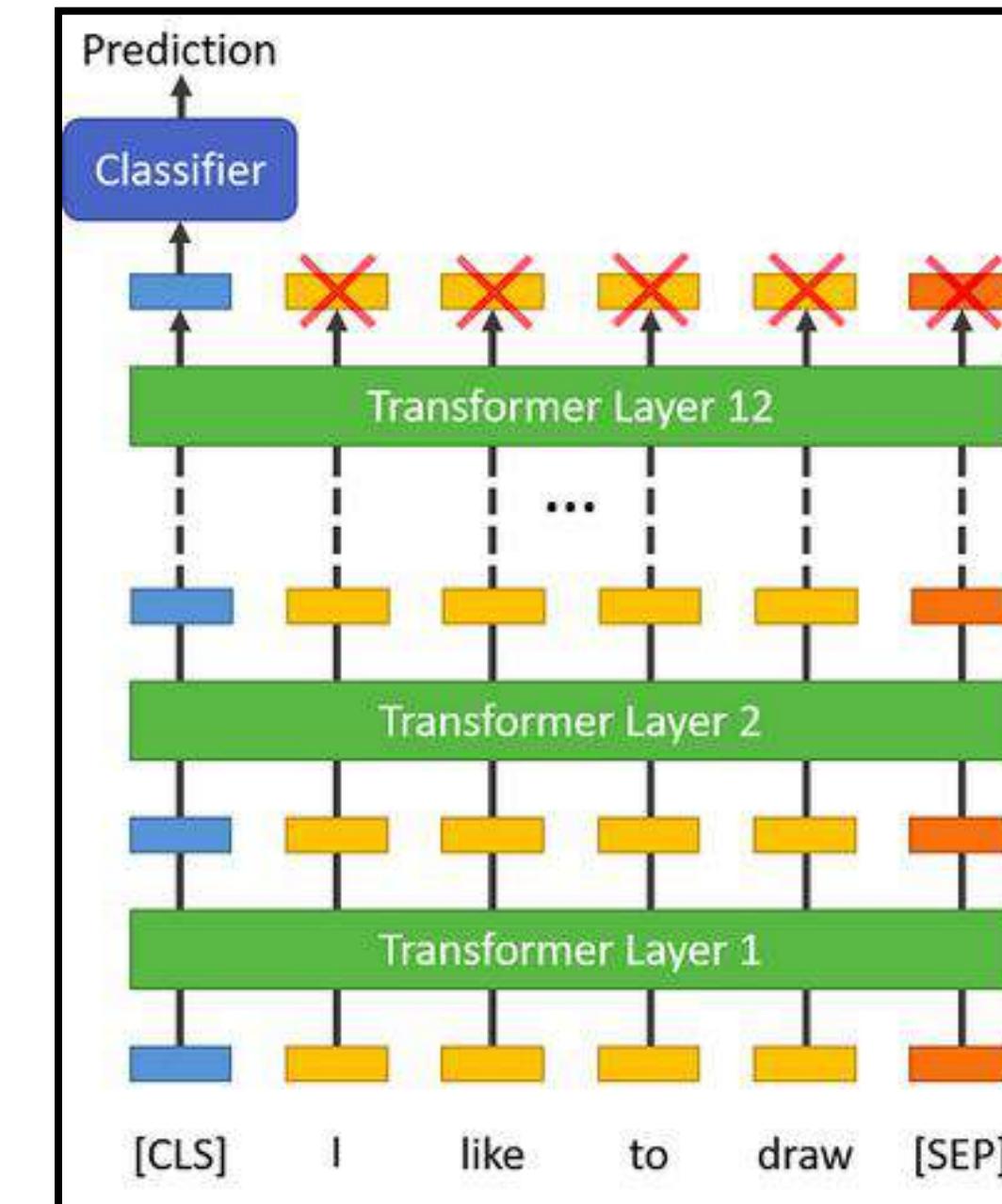
BERT

- Context based embedding
 - Unlike context-free models which generate a single vector (word embedding) representation for each token
 - Context based embeddings like BERT generate word vectors based on the context



BERT

- BERT-Base
 - 12-layer
 - 768-hidden-nodes
 - 12-attention-heads
 - 110M parameters
- BERT-Large
 - 24-layer
 - 1024-hidden-nodes
 - 16-attention-heads
 - 340M parameters

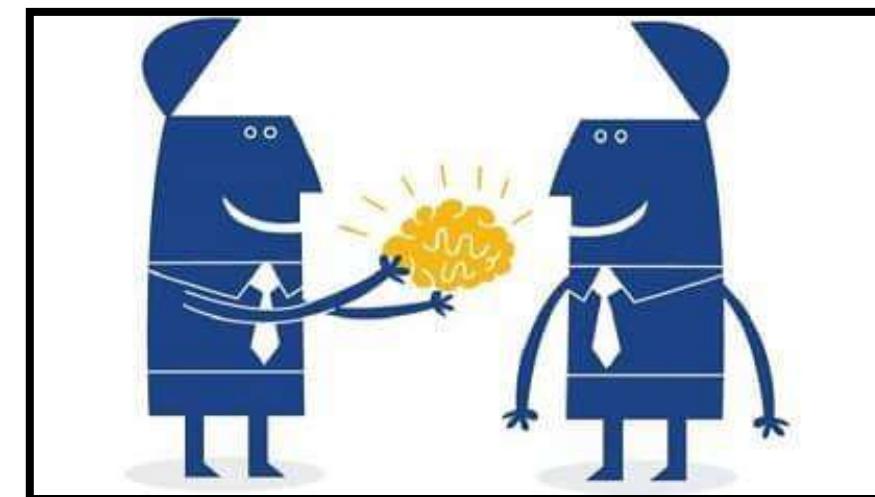


BERT

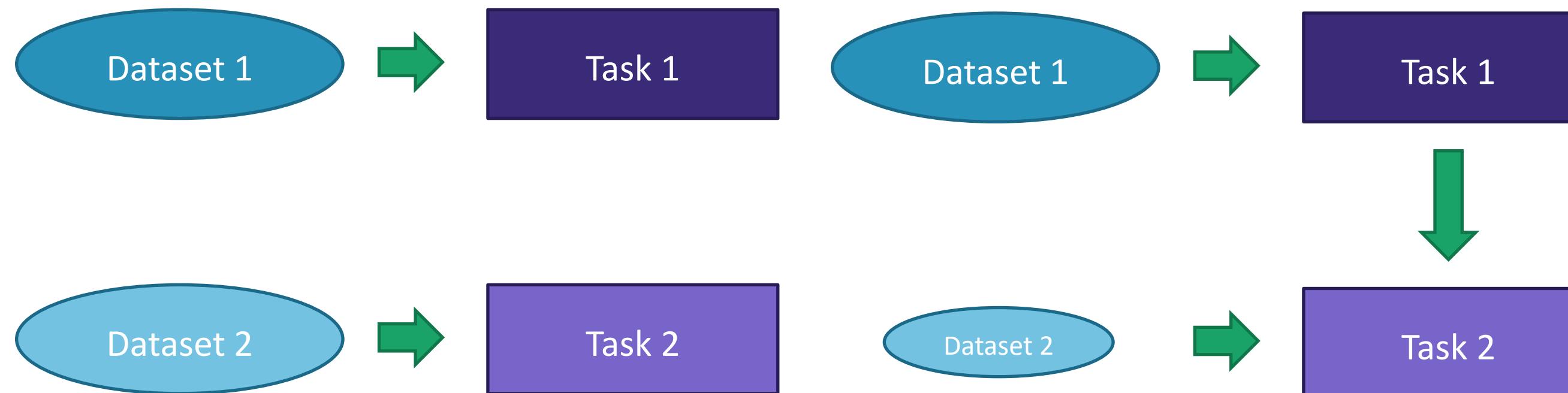
- Updating the weights
 - Not tune (feature extraction)
 - Pretrained representations can be used as features in a downstream model
 - Tune (fine-tuning)
 - The pretrained weights are used as initialization for parameters of the downstream model
 - The whole pretrained architecture is then trained during the adaptation phase

Summary

- Transfer learning is a learning procedure in which representations learned on a source task are **transmitted** to improve learning on the target task

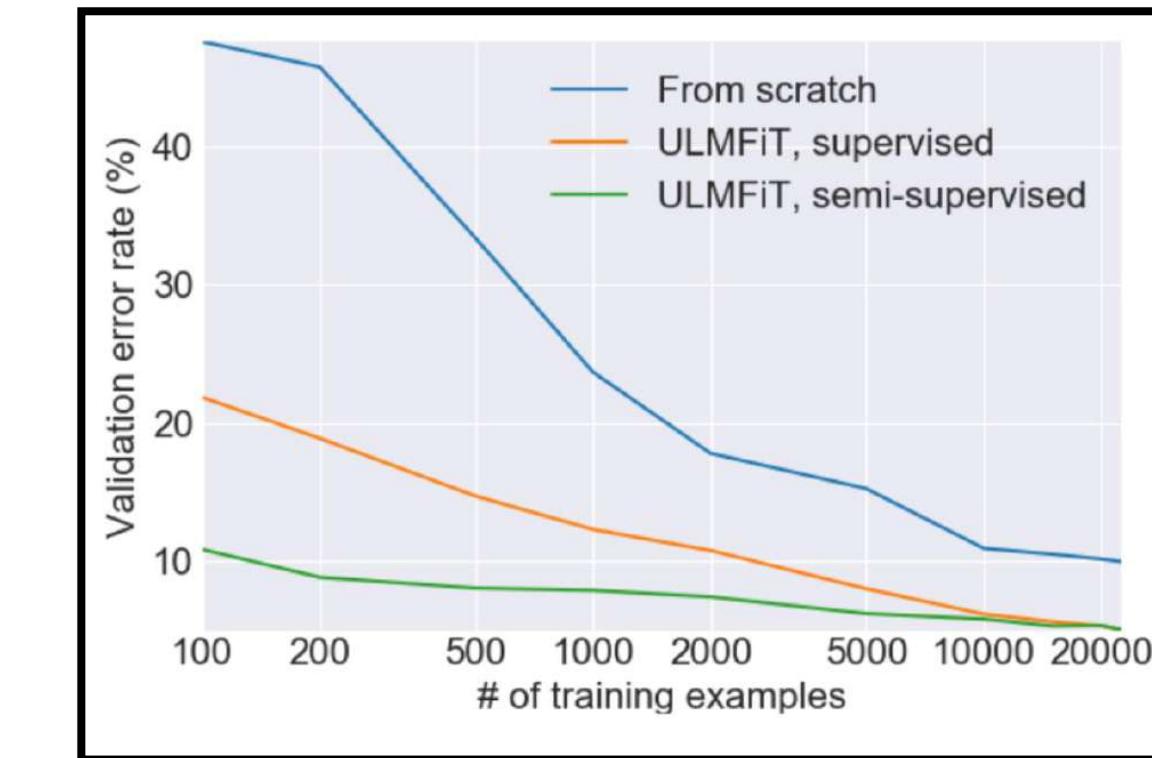


<https://jeanvitor.com>

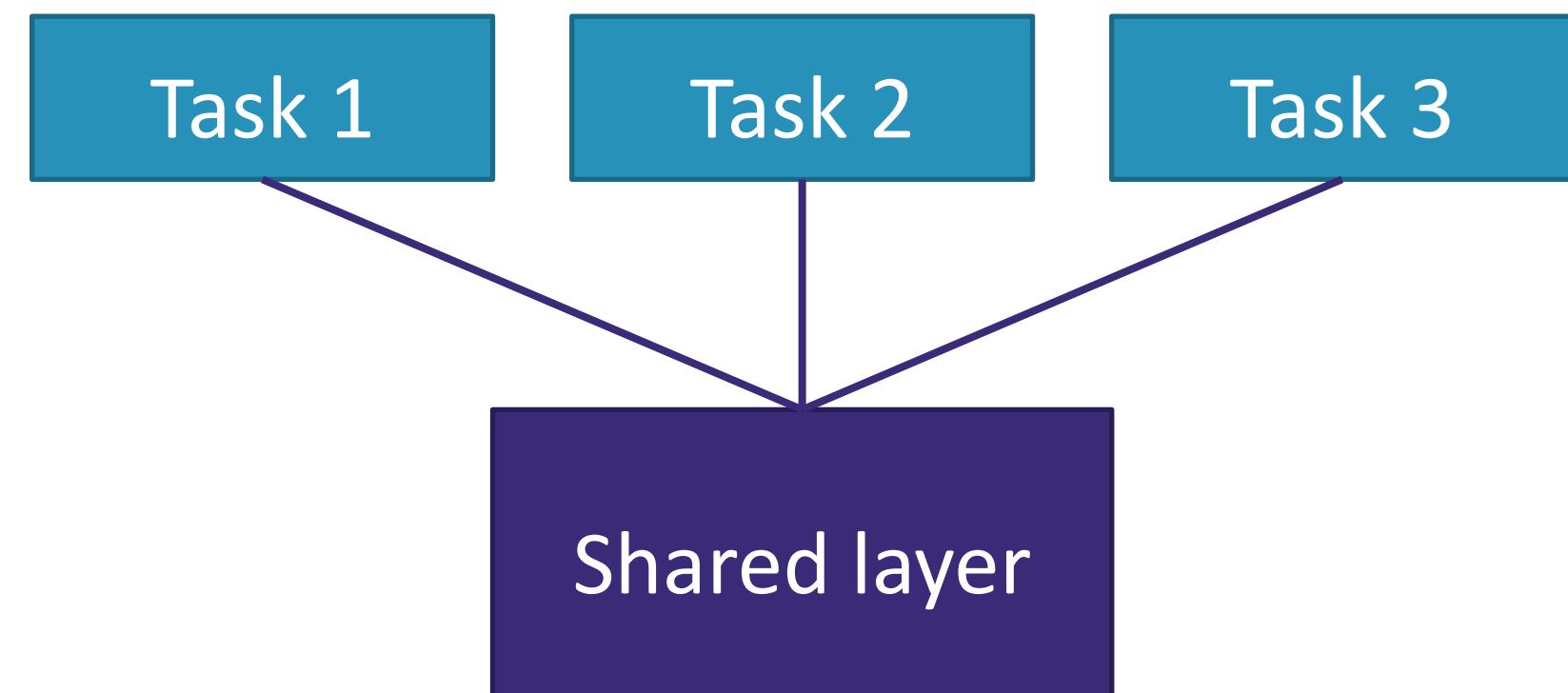
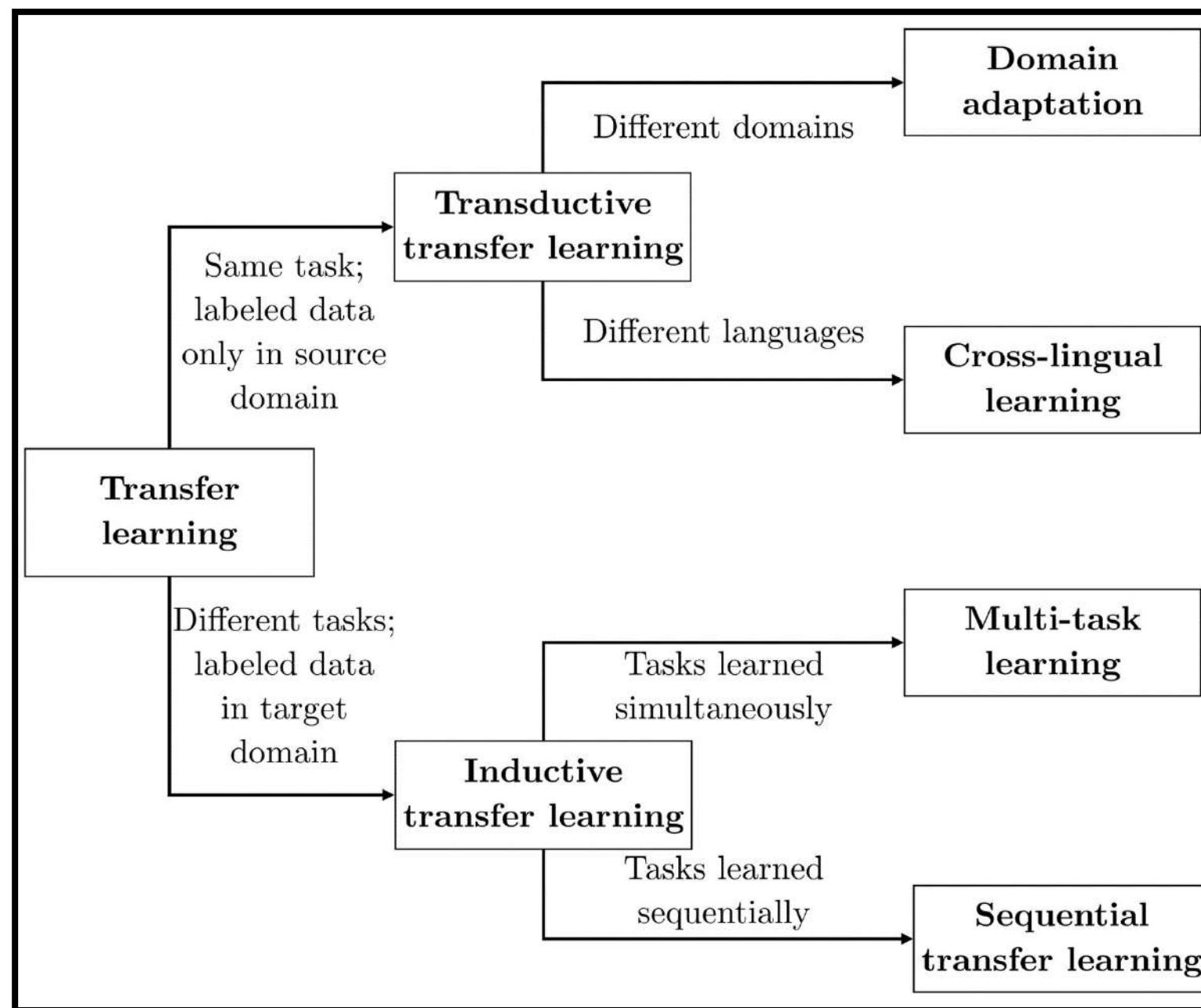


Summary

- Transfer learning has several benefits, but the main advantages are:
 - Saving training time
 - Better performance
 - Not needing a lot of data



Summary



Summary

- Different approaches for sequential transfer learning
 - Training a model to reuse it
 - Fine-tuning a pre-trained model
 - Feature extraction

