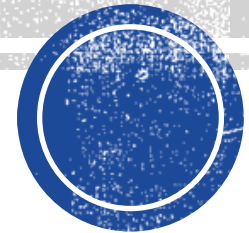
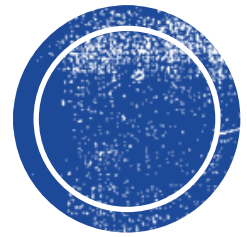


Text Processing

Professional Program in Artificial Intelligence and Emerging
Technologies

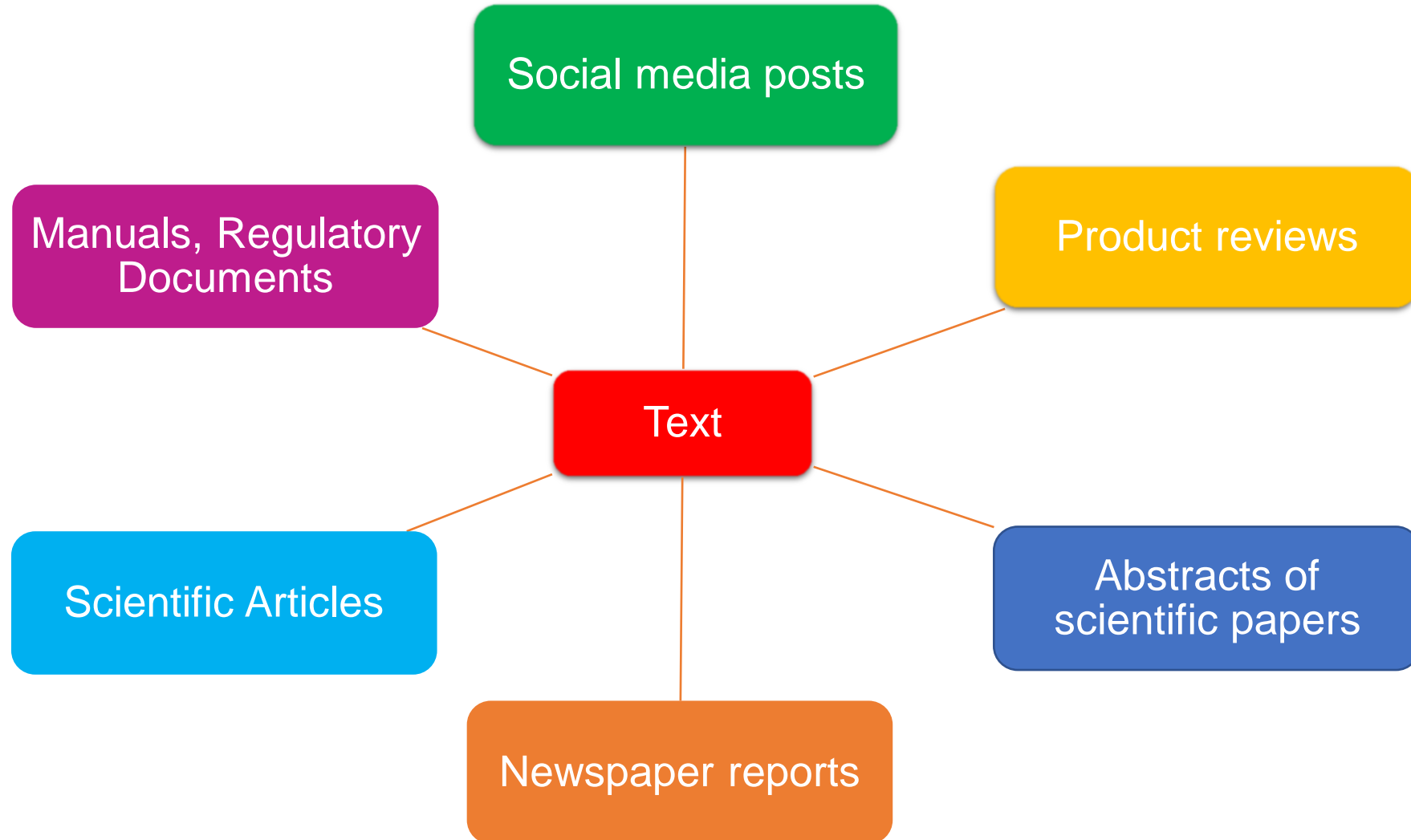




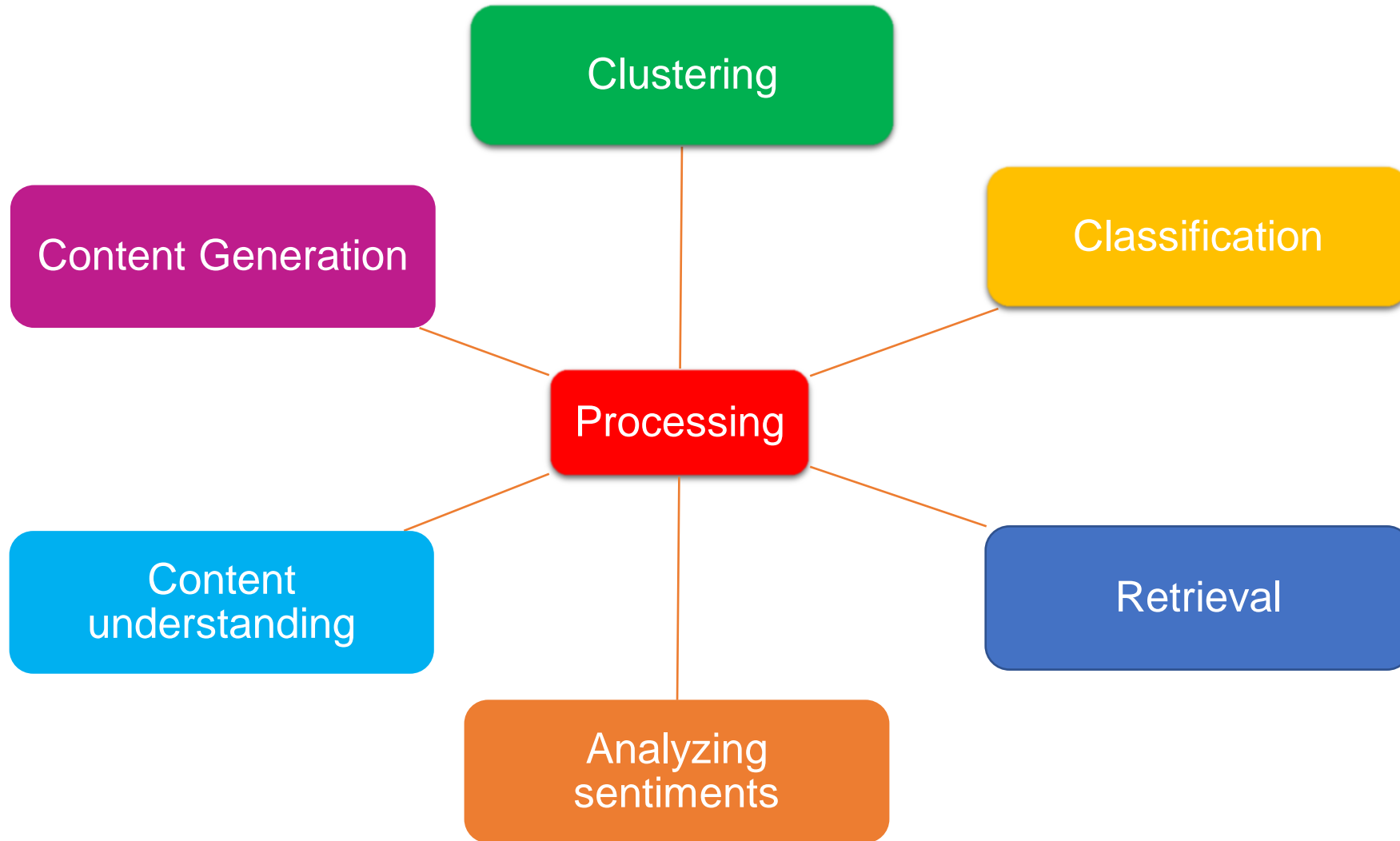
Text processing

What is it?

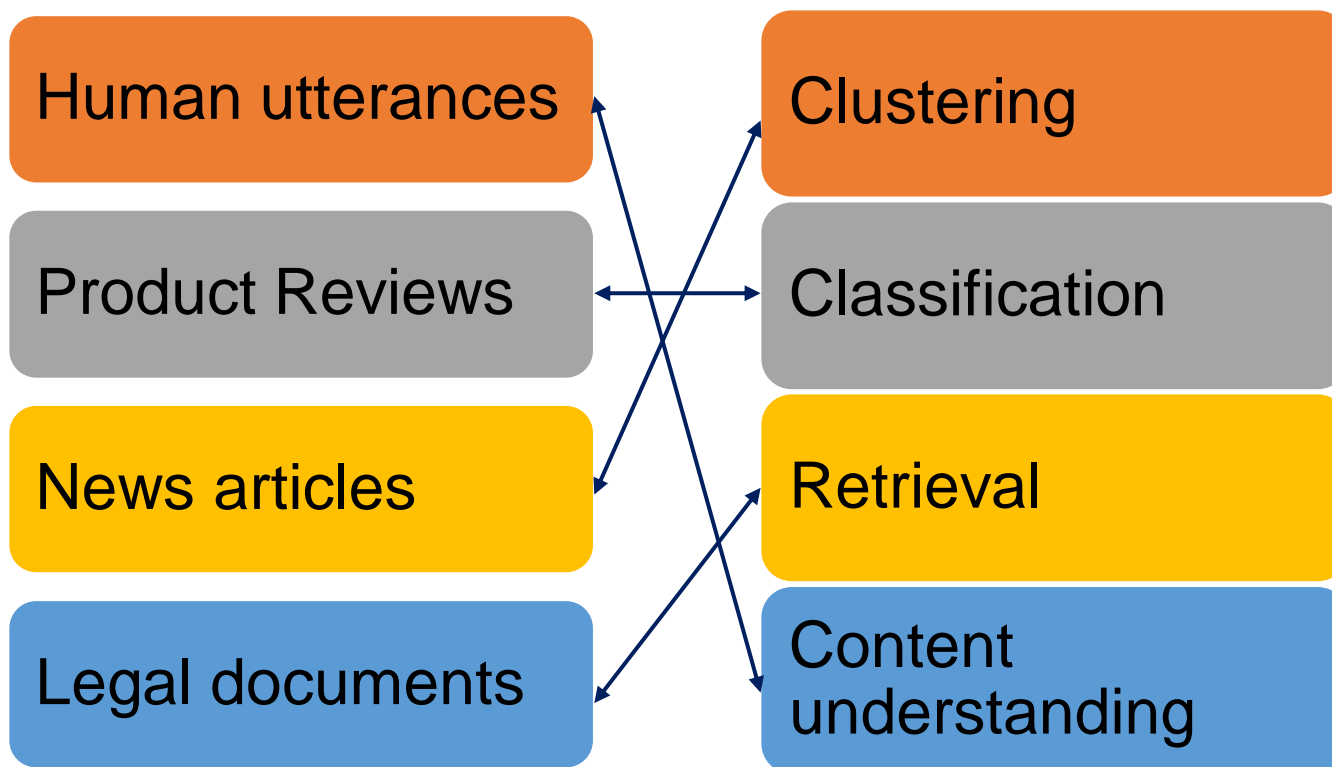
Text Processing



Text Processing



Text Processing



mohit sud



Poor Finish, Peeling Paint

Reviewed in India on 4 August 2020

Size name: 24" | **Verified Purchase**

New monitor.

Delivered yesterday

I peeled off the plastic cover on the speaker today and the paint peeled off

U_0 : Can you help me find some attractions in the **east** part of town?

S_1 : Definitely! My favorite place in the east is the **Funky Fun House**. It's funky and fun!

U_1 : Can I have the number please?

2.The petitioner is the owner of the vehicle, namely, TATA BOLT with Registration No.TN-79-F-0796. On 05.10.2021, the said vehicle was intercepted and seized by the second respondent police on the alleged reason that the vehicle was involved in transporting the illicit liquor bottles and therefore, a case has been registered in Crime No. 245/2021 on the file of the second respondent police for the alleged offence punishable under Section 4(1)(a) of the Tamilnadu Prohibition Act, with the result, the vehicle of the petitioner, seized by the second <https://www.mhc.tn.gov.in/judis/> W.P.(MD) No.18894 of 2021 respondent, has been kept at the custody of the second respondent from 05.10.2021.



This company offered Okinawa e-scooters as Diwali gift to employees

8 hours ago



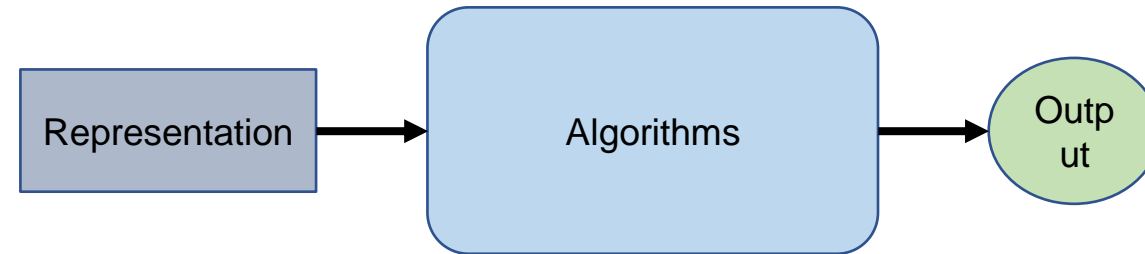
Challenges with Textual Content

- What are the major challenges in text processing/language processing?
- Semantics/Meaning
 - I went to the bank yesterday
- Syntax/Structure
 - I saw a man with a telescope



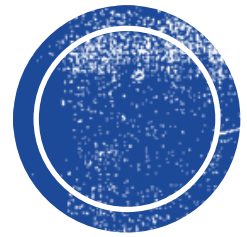
Text Processing Pipeline

- Input Representation, Processing and Output



- Algorithm and output depend on the task
- Let us focus on representations, which can be generic (to the extent possible)
- What does it require?





Text processing

Identifying individual words

Processing text

- Assume collection of text “documents”
- Each document has an ID
- Identify individual units – the “terms”
- Store the “term occurrence information”
- Size????

Term	Doc1	Doc2	Doc3	Doc4	Doc5	Doc6	Doc7	Doc8
aid	0	0	0	1	0	0	0	1
all	0	1	0	1	0	1	0	0
back	1	0	1	0	0	0	1	0
brown	1	0	1	0	1	0	1	0
come	0	1	0	1	0	1	0	1
dog	0	0	1	0	1	0	0	0
fox	0	0	1	0	1	0	1	0
good	0	1	0	1	0	1	0	1
jump	0	0	1	0	0	0	0	0
lazy	1	0	1	0	1	0	1	0
men	0	1	0	1	0	0	0	1
now	0	1	0	0	0	1	0	1
over	1	0	1	0	1	0	1	1
party	0	0	0	0	0	1	0	1
quick	1	0	1	0	0	0	0	0
their	1	0	0	0	1	0	1	0
time	0	1	0	1	0	1	0	0



Bigger collections

- Consider $N = 10^6$ documents, each with about 1000 **tokens**
 - Total of $10^6 * 10^3 = 10^9$ tokens
- On average 6 bytes per token, including spaces and punctuation
 - **Size of the corpus:** $6 \cdot 10^9 = 6 \text{ GB}$
- Assume there are $M = 500,000$ distinct **terms** in the collection
- $M = 500,000 \times 10^6 = \text{half a trillion 0s and 1s.}$
- But the matrix has no more than one billion 1s.
 - Matrix is extremely sparse.
- What is a better representation?
 - We only record the 1s.

Identifying terms from a document

1. **Read** the document:

The value of each vote is one

2. **Tokenize** the text, turning each document into a list of tokens:

The, value, of, each, vote, is, one

3. **Store** the information *appropriately*





Challenges?

- **File format**

- Text, CSV, Excel, HTML

- **Should we lowercase everything?**

- Information to information, IITH to iith, WHO to who

- **Which delimiters to use?**

- State-of-the-art, Forbes-500

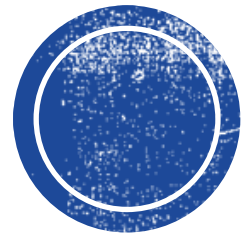
- **Date Format**

- 12/09/2020, 12-Sep-2020, September 12th, 2020, 3 pm, 3:00 pm, 15:00 hours

- **Numbers**

- Two thousand, 2000, 20.04





Text processing

The terms, their forms and other decisions

Linguistic Preprocessing

- Stemming

- Institution -> Institut
- Terrible -> Terribl
- Requiring -> Requir

- Lemmatization

- Institution -> Institute
- Terrible -> terrible
- Requiring -> Require

Vodafone Idea Ltd on Monday rebranded itself as 'Vi', creating a unified identity two years after the merger of erstwhile Vodafone India Ltd and Idea Cellular Ltd.

Porter: Vodafone Idea Ltd on Monday rebranded itself as Vi creating a unified identity two years after the merger of erstwhile Vodafone India Ltd and Idea Cellular Ltd

Lancaster: Vodafone Idea Ltd on Monday rebranded itself as 'Vi', creating a unified identity two years after the merger of erstwhile Vodafone India Ltd and Idea Cellular Ltd.

Wordnet Lemmatizer: Vodafone Idea Ltd on Monday rebranded itself as 'Vi', creating a unified identity two years after the merger of erstwhile Vodafone India Ltd and Idea Cellular Ltd.

<http://text-processing.com/demo/stem/>



Linguistic Preprocessing

According to the Russian Health Ministry, as reported by news agency ANI, the authorities will soon start the regional deliveries of the much-awaited Sputnik V vaccine for the highly contagious coronavirus caused by SARS-CoV-2.

accord to the Russian Health Ministry , as report by news agency ANI , the authority will soon start the
regional delivery of the much - await Sputnik V vaccine for the highly contagious coronavirus cause by SARS
- CoV - 2 .

From <http://stanza.run/>



Identifying terms from a document

- 1 **Collect** the documents to be indexed:

The value of each vote is one. The EVM used for the election will have the names of the contesting candidates

- 2 **Tokenize** the text, turning each document into a list of tokens:

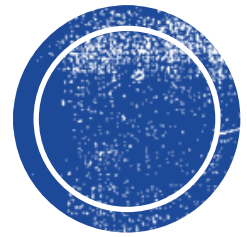
the, value, of, each, vote, is, one, the, EVM, used, for, the, election, will, have, the, names, of, the, contesting, candidates

- 3 Do **linguistic preprocessing**, producing a list of normalized tokens, which are the indexing terms:

the, value, of, each, vote, is, one, EVM, **use**, for, the, election, will, have, the, **name**, of, the, **contest**, **candidate**

- 4 **Store/Use** the tokens appropriately





Text processing

More Information from/about the Tokens

Some examples from NLP

- **Word Sense Disambiguation** (e.g. for translation)
 - He wrote his last sentence two years ago (legal judgement vs grammatical sentence?)
 - Look at surrounding context
 - Needs to build a classifier for these two meanings for the word “letter”

Translation	Context
(1) sentencia	... for a maximum <i>sentence</i> for a young offender ...
” ”	... of the minimum <i>sentence</i> of seven years in jail ...
” ”	... were under the <i>sentence</i> of death at that time ...
(2) frase	... read the second <i>sentence</i> because it is just as ...
” ”	... The next <i>sentence</i> is a very important ...
” ”	... It is the second <i>sentence</i> which I think is at ...

- **Spelling Correction**
 - Words to the left and right, their POS,



Named Entity Recognition

- The IOB encoding (Ramshaw & Marcus 1995):
 - B_X = “beginning” (first word of an X)
 - I_X = “inside” (non-first word of an X)
 - O = “outside” (not in any phrase)
 - Does not allow overlapping or recursive phrases

...**United Airlines** said Friday it has increased ...

B_ORG I_ORG O O O

... spokesman **Tim Wagner** said ...

O B_PER I_PER O

Example features:

- Original token
- Lowercased?
- First letter capitalized?
- All letters in capital?
- First c characters of the token
- Last c characters of the token
- Lemma
- POS
- In Known List?
- k surrounding words



Some examples from NLP

- Information extraction

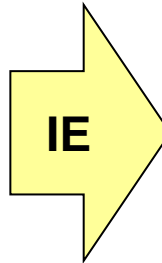
October 14, 2002, 4:00 a.m. PT

For years, [Microsoft Corporation CEO Bill Gates](#) railed against the economic philosophy of open-source software with Orwellian fervor, denouncing its communal licensing as a "cancer" that stifled technological innovation.

Today, Microsoft claims to "love" the open-source concept, by which software code is made public to encourage improvement and development by outside programmers. Gates himself says Microsoft will gladly disclose its crown jewels--the coveted code behind the Windows operating system--to select customers.

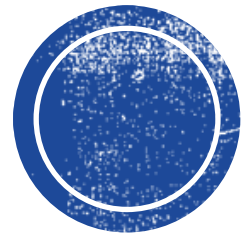
"We can be open source. We love the concept of shared source," said [Bill Veghte](#), a [Microsoft VP](#). "That's a super-important shift for us in terms of code access."

[Richard Stallman](#), [founder](#) of the [Free Software Foundation](#), countered saying...



NAME	TITLE	ORGANIZATION
Bill Gates	CEO	Microsoft
Bill Veghte	VP	Microsoft
Richard Stallman	founder	Free Soft..





Text processing

Beyond individual words

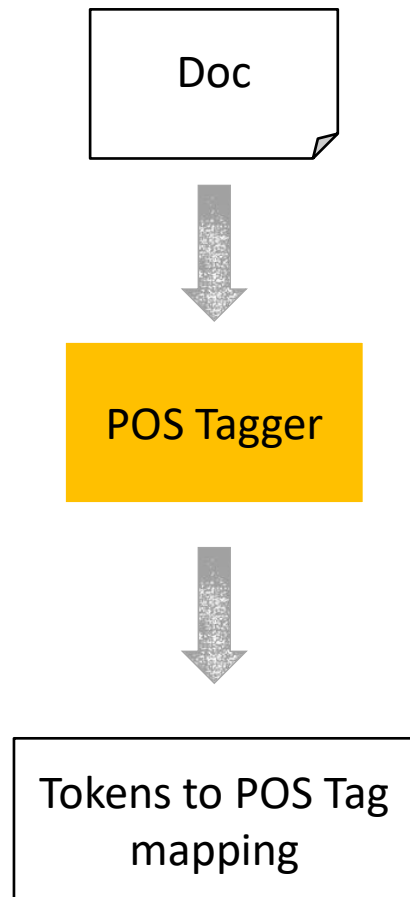
Biwords

- Consecutive pair of terms in text (2-word phrase, word bigram)
- Text: *Natural language processing*
- *Corresponding bi-words:*
 - “*natural language*” and “*language processing*”
- Each of these bi-words is now a vocabulary term.



Extended biwords

Parse each document and perform part-of-speech tagging



Apache OpenNLP

(<https://opennlp.apache.org/>)

Stanford NLP

(<https://nlp.stanford.edu/software/tagger.html>)

NLTK POS Tagger

(<https://www.nltk.org/api/nltk.tag.html>)

ProLNat@GE

(<https://gramatica.usc.es/pln/tools/CitiusTools.html>)



Few more resources for POS Tagging

- List of few online tools and APIs:
 - <http://nlp.stanford.edu:8080/parser/index.jsp>
 - http://cogcomp.org/page/demo_view/pos
 - <http://textanalysisonline.com/spacy-pos-tagging>
- **Sample text:** "India has been ranked 47th in the overall "Inclusive Internet Index 2019" score while Sweden topped the chart, followed by Singapore and the US, a Facebook-led study has revealed."



POS Tagging Example

- Idukki dam full to brim, army steps in for rescue ops
 - [NNP] Idukki [NN] dam [JJ] full [TO] to [NN] brim [,] , [NN] army [NNS] steps [IN] in [IN] for [NN] rescue [NNS] ops
- **NN** Singular noun
 - **NNS** Plural noun
 - **NNP** Proper singular noun
 - **JJ** Adjective
 - **TO** To
 - **IN** Preposition



POS Tagging Example

- Lord of the rings is a damn good movie
- [NNP] Lord [IN] of [DT] the [NNS] rings [VBZ] is [DT] a [RB] damn [JJ] good [NN] movie

- **NN** Singular noun
- **NNP** Proper singular noun
- **NNS** Plural noun
- **JJ** Adjective
- **DT** Determine
- **IN** Preposition
- **VBZ** Verb, 3rd ps. sing. Present
- **RB** Adverb



Sample Python Code for POS Tagging

```
import nltk
nltk.download('punkt')
nltk.download('averaged_perceptron_tagger')
from nltk import word_tokenize
sentence = "I am attending the class"
tokens=word_tokenize(sentence)
```

```
import spacy
sp = spacy.load('en_core_web_sm')
sen = sp(u"I like to play football. I hated it in my childhood though")
print(sen.text)
print(sen[7].pos_)
print(sen[7].tag_)
print(spacy.explain(sen[7].tag_))
```

<https://stackabuse.com/python-for-nlp-parts-of-speech-tagging-and-named-entity-recognition/>



Extended biwords

- Parse each document and perform part-of-speech tagging
- Bucket the terms into (say) nouns (N) and articles/prepositions (X)
- Extended Biwords: terms of the form NX*N
- Examples:
 - Lord of the rings
N X X N
 - President of India
N X N
- Include extended biwords in the term vocabulary
- Queries are processed accordingly



Issues with biwords

- Why are biword indexes rarely used?
 - More than 2 words in phrase query -> Possible false positives
 - Index blowup due to very large term vocabulary
-
- Solution/Remedy?

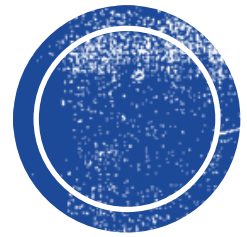


Named Entities

- Phrases/Sequences of words that talk about a name (of Location, Organization, Place, ...)
 - <https://dandelion.eu/semantic-text/entity-extraction-demo/>
 - <http://textanalysisonline.com/spacy-named-entity-recognition-ner>

Sample text: “It is fuelled by a 4,000 mAh battery and supports 30 W Warp charge technology. Powering the phone is Qualcomm Snapdragon 855 platform.”





Text processing

Representing the documents

Documents and words

- A text **document is composed by words**
- **Meaning** of the document is **implied by the words** it contains
- A **document** can be represented as a **set of terms** (words, phrases)
- Can be **represented as a vector**
- **Effective for document categorization, clustering, ...**



How to represent documents?

- Many ML algorithms have to deal with real life objects/subjects
 - Users, Products, Videos, Movies,...
- How do we represent them?
 - As vectors
- We can represent documents as vectors too
- The concept is used in various problem settings
 - Web Search
 - Document classification
 - Document clustering



Vectors

- **Dimensions:** Words (n-grams)
- **Values:** Weight of the word in the document
 - Boolean (Presence/Absence)
 - Integer (Occurrence count)
 - Real (Functions of occurrence count)

If you want to shine like a sun. First burn like a sun

if	you	want	to	shine	like	a	sun	first	burn
1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	2	2	2	1	1



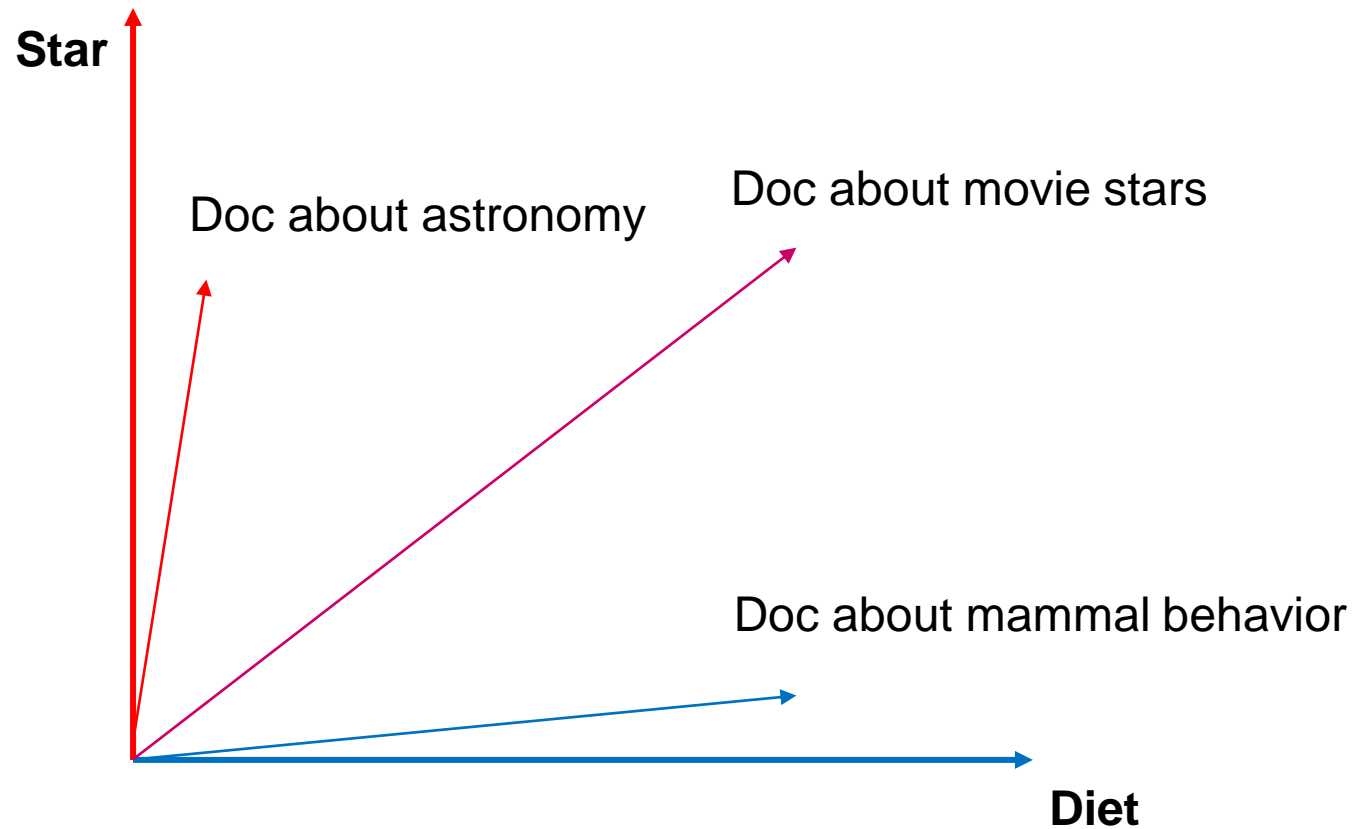
Sample collection

- D1: Today's class is in the evening
- D2: The evening class was delayed
- D3: Class-X exams start today
- D4: Let's meet today evening





Documents in Vector Space



Vectors Space Model: Observations

- **No semantic information**
 - Polysemy (E.g.: Set, Goal, Spears, Apple)
 - Synonymy (E.g.: Auto vs Car, light vs bulb)
 - Remedies: Context based Semantic Indexing
- **Bag of words – No sequence information**
 - Different meaning but same representation (E.g.: A is better than B vs B is better than A)
 - Remedies: Use of Positional index, phrase information
- **Q: Should all words be equally important?**

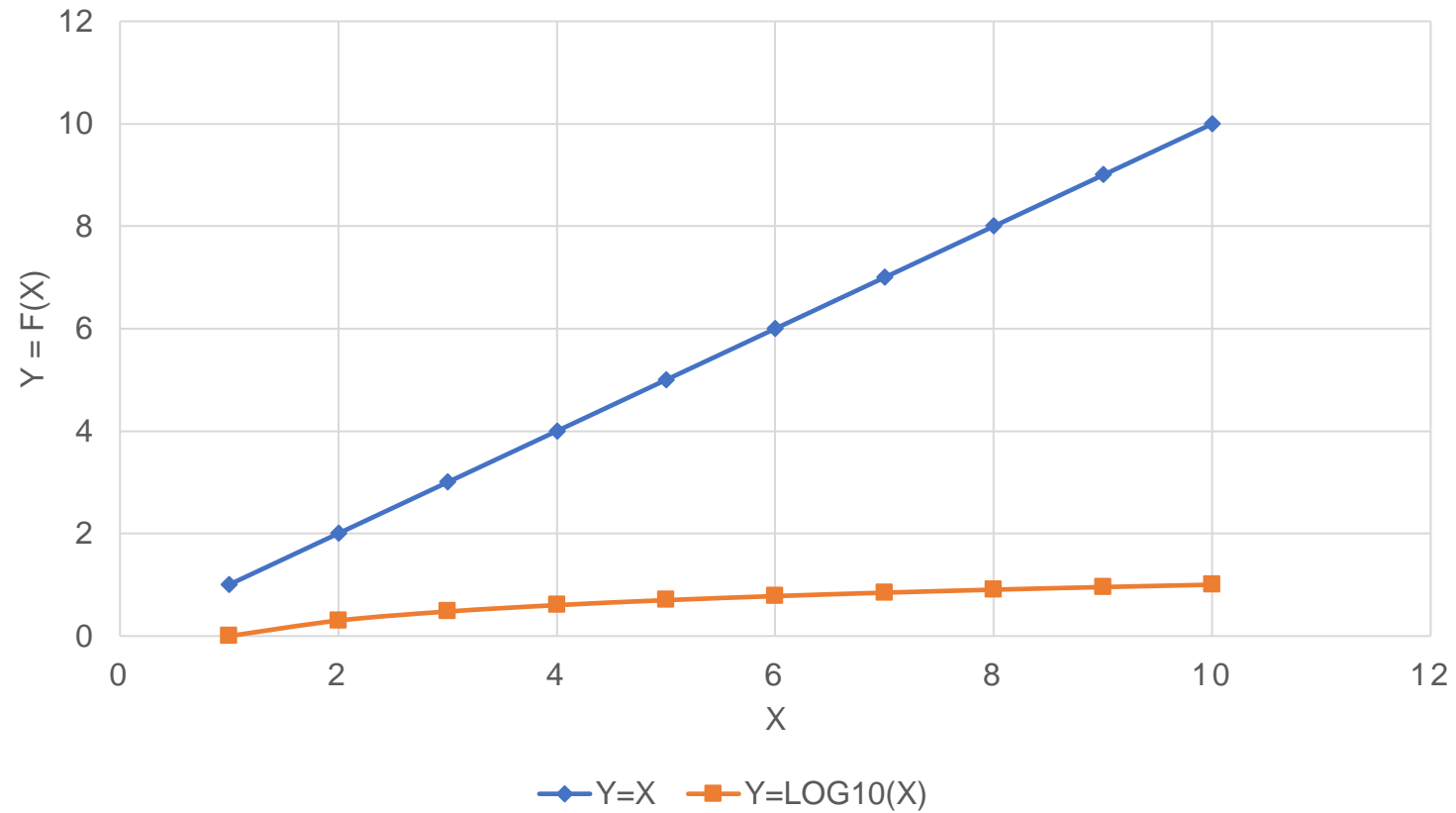


Term frequency tf

- The term frequency $tf_{t,d}$ of term t in document d is defined as the number of times that t occurs in d .
- Raw term frequency is not what we want
 - A document with $tf = 10$ occurrences of the term is more relevant than a document with $tf = 1$ occurrence of the term.
 - But not 10 times more relevant.
- Relevance does not increase proportionally with term frequency.



Damping in Term Frequency Function





Raw frequency to Log frequency weighting

- The log frequency weight of term t in d is defined as follows

$$w_{t,d} = 1 + \log_{10} tf_{t,d} \quad \text{if } tf_{t,d} > 0$$

0 otherwise



Desired weight for rare terms

- Rare terms are more informative than frequent terms.
- Frequent terms:
 - Terms appearing in many documents
- Weight of term (t) inversely proportional to document frequency of the term
 - $idf_t = \log_{10} \frac{N}{df_t}$

Term	Document Frequency	IDF weight
The	100000	0
What	10000	1
Simple	1000	2
Weather	100	3
Update	10	4

N = 100000

Each field of the vector is now $d_t = (1 + \log(tf_{t,d})) \log \frac{N}{df_t}$



TF-IDF Weight: Exercise

- Consider a collection with three documents:
 - d1 = The highway is empty
 - d2 = Empty glass empty plate
 - d3 = Highway on my plate
- Find the idfs of the different terms in the collection
- What is the tf-idf weight of the term “plate” in d2?
- What is the tf-idf weight of the term “on” in d3?
- Using tf-idf values, what is the vector representation of d1, d2 and d3?
- What is the similarity of the q=“empty plate” with d1?
- How many terms will be there in the vocabulary, if it stores both unigrams and bigrams? Consider no stopword removal, no morphological analysis is done.



For more, <https://janav.wordpress.com/2013/10/27/tf-idf-and-cosine-similarity/>



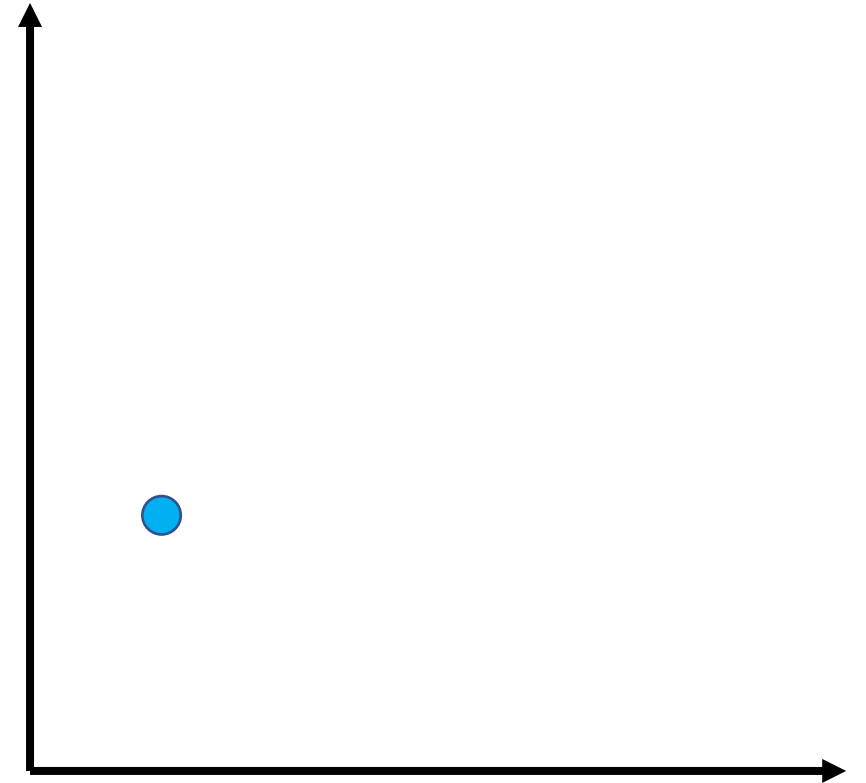
How do we formalize vector space similarity?

- Using distance?
 - Less the distance, higher the similarity
- First cut: (negative) distance between two points
 - Euclidean distance?
 - Consideration: **large** for vectors of different lengths.
- Any other measure that does not have this issue?
 - Angle?



Use angle instead of distance

- Rank documents according to angle with query
- Experiment:
 - Take a document d and append it to itself. Call this document d' . d' is twice as long as d . (according to raw tf)
 - “Semantically” d and d' have the same content.
- The angle between the two documents is 0, corresponding to maximal similarity
- Although the Euclidean distance between the two documents can be quite large.



From angles to cosines

- The following two notions are equivalent.
 - Rank documents according to the **angle** between query and document in increasing order
 - Rank documents according to **cosine** (query,document) in decreasing order
- Why?
 - Cosine is a monotonically decreasing function of the angle for the interval $[0^\circ, 180^\circ]$



Length normalization

- A vector can be (length-) normalized by dividing each of its components by its length – here we use the L_2 norm: $||\vec{x}|| = \sqrt{\sum_i x_i^2}$
- As a result, longer documents and shorter documents have weights of the same order of magnitude.
- Effect on the two documents d and d' (d appended to itself) from earlier slide: they have **identical vectors** after length-normalization

Exercise:

1. Consider the vector $v = [1, 0, 2, 1]$. Normalize it using the procedure discussed above. Verify that it has unit length
2. Consider the vector $c = [2, 0, 4, 2]$. Normalize it using the procedure discussed above. Verify that it has unit length





Relationship between Cosine similarity and Dot Product

- Earlier, we said that we will rank using inner product
- Then we said, we will rank by angle
- Then we said, we will rank by cosine of the angle
- How are these things related?

$$\begin{aligned}\cos(\vec{q}, \vec{d}) &= \frac{\vec{q} \cdot \vec{d}}{|\vec{q}| |\vec{d}|} \\ &= \frac{\sum_i q_i d_i}{\sqrt{\sum_i q_i^2} \sqrt{\sum_i d_i^2}} \\ &= \sum_i \left(\frac{q_i}{\sqrt{\sum_i q_i^2}} \frac{d_i}{\sqrt{\sum_i d_i^2}} \right) \\ &= \sum_i (q'_i d'_i) \\ &= \vec{q'} \cdot \vec{d'}\end{aligned}$$

Cosine of the angle between two vectors
= inner product between the normalized versions of the corresponding vectors (our initial scoring scheme)



Normalization


- What **extra operation** are we doing in the dot product?
 - Normalizing the vectors
- What are the **benefits** of normalization?
 - Bringing down the values in the vectors' fields to similar range
 - Helps to remove the undue advantage received otherwise by the larger documents
- **Is this the only way** of doing normalization?
 - Definitely not





Using these representations

- Document classification

 mohit sud

☆☆☆☆☆ **Poor Finish, Peeling Paint**

Reviewed in India on 4 August 2020

Size name: 24" | **Verified Purchase**

New monitor.

Delivered yesterday

I peeled off the plastic cover on the speaker today and the paint peeled off

- Document Clustering



This company offered Okinawa e-scooters as Diwali gift to employees

8 hours ago



- Finding documents similar to a given document

2.The petitioner is the owner of the vehicle, namely, TATA BOLT with Registration No.TN-79-F-0796. On 05.10.2021, the said vehicle was intercepted and seized by the second respondent police on the alleged reason that the vehicle was involved in transporting the illicit liquor bottles and therefore, a case has been registered in Crime No. 245/2021 on the file of the second respondent police for the alleged offence punishable under Section 4(1)(a) of the Tamilnadu Prohibition Act, with the result, the vehicle of the petitioner, seized by the second <https://www.mhc.tn.gov.in/judis/> W.P.(MD) No.18894 of 2021 respondent, has been kept at the custody of the second respondent from 05.10.2021.

- May require/benefit-from additional information



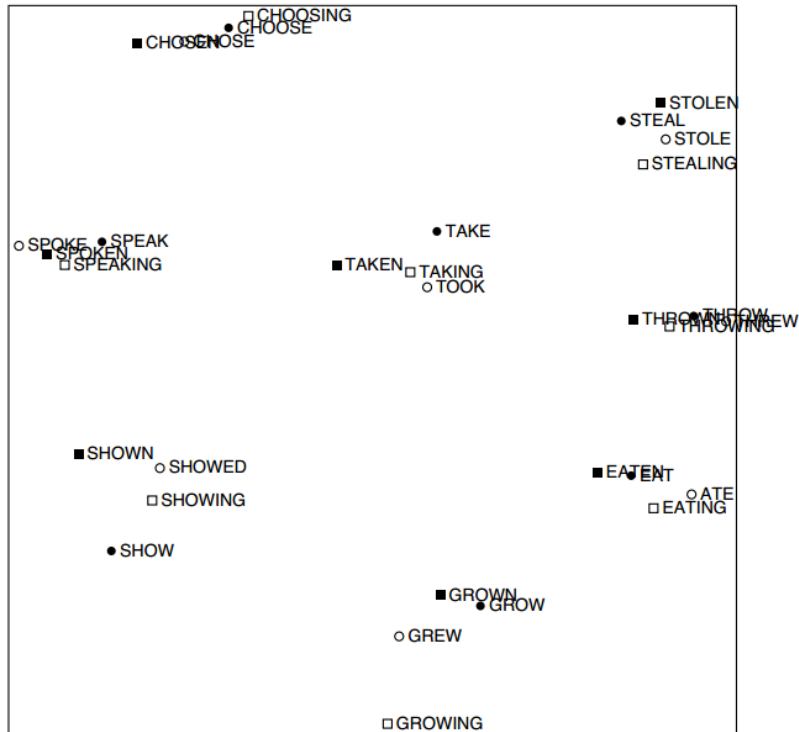


Word and it's company

- Let few words w_1, w_2, w_3, w_4 tend to cooccur together many times
- Consider $w_1 = \text{programming}$ $w_2 = \text{python}$ $w_3 = \text{script}$ $w_4 = \text{execute}$
- Can we pass on the following information to a learning **method** ..
 - w_1 and w_2 tend to occur together many times
 - w_1 and w_3 tend to occur together many times
 - w_1 and w_4 tend to occur together many times
 - w_2 and w_3 tend to occur together many times
 - ...
- ... and expect that the above information can be incorporated in the learning process – learning of word representations?



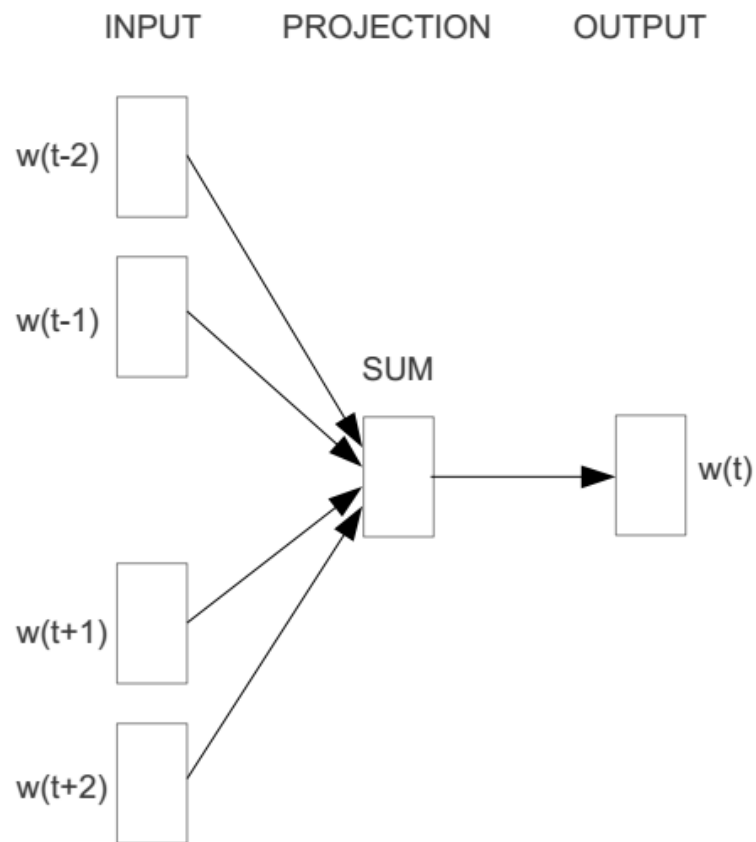
Words as vectors: Word2Vec



- Representation of a word is dictated by other surrounding words
- Assume a fixed length context window
- For example:
 - $[w_{-2} \ w_{-1} \ c \ w_1 \ w_2]$
- Start with random initialization
- Iterate till convergence



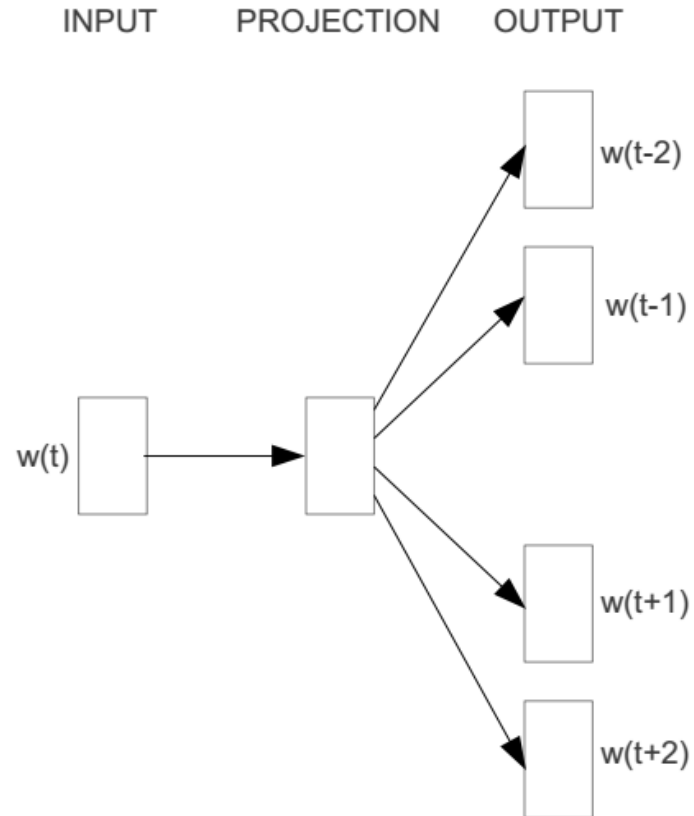
Word2Vec Models: SkipGram (SG)



- Training sentence:
- ... the algorithm's asymptotic complexity is quadratic...
- w_{-2} w_{-1} c w_1 w_2
- Considering words in a context window of length 5
 - $P(\text{target} \mid [w_{-2} \ w_{-1} \ w_1 \ w_2]) = ?$
- Training data:
 - (c, w_{-2})
 - (c, w_{-1})
 - (c, w_1)
 - (c, w_2)



Word2Vec Models: CBOW



Skip-gram

- Training sentence:
- ... the algorithm's asymptotic complexity is quadratic...
- Considering words in a context window of length 5
 - w_{-2} w_{-1} c w_1 w_2
 - $P([w_{-2} \ w_{-1} \ w_1 \ w_2 | target) = ?$
- Training data:
 - (c, w_{-2})
 - (c, w_{-1})
 - (c, w_1)
 - (c, w_2)



More examples of target and context

#1	natural	language	processing	and	machine	learning	is	fun	and	exciting	#1
	X _k	Y(c=1)	Y(c=2)								
#2	natural	language	processing	and	machine	learning	is	fun	and	exciting	#2
	Y(c=1)	X _k	Y(c=2)	Y(c=3)							
#3	natural	language	processing	and	machine	learning	is	fun	and	exciting	#3
	Y(c=1)	Y(c=2)	X _k	Y(c=3)	Y(c=4)						
#4	natural	language	processing	and	machine	learning	is	fun	and	exciting	#4
		Y(c=1)	Y(c=2)	X _k	Y(c=3)	Y(c=4)					
#5	natural	language	processing	and	machine	learning	is	fun	and	exciting	#5
			Y(c=1)	Y(c=2)	X _k	Y(c=3)	Y(c=4)				
#6	natural	language	processing	and	machine	learning	is	fun	and	exciting	#6
				Y(c=1)	Y(c=2)	X _k	Y(c=3)	Y(c=4)			
#7	natural	language	processing	and	machine	learning	is	fun	and	exciting	#7
					Y(c=1)	Y(c=2)	X _k	Y(c=3)	Y(c=4)		
#8	natural	language	processing	and	machine	learning	is	fun	and	exciting	#8
						Y(c=1)	Y(c=2)	X _k	Y(c=3)	Y(c=4)	
#9	natural	language	processing	and	machine	learning	is	fun	and	exciting	#9
							Y(c=1)	Y(c=2)	X _k	Y(c=3)	
#10	natural	language	processing	and	machine	learning	is	fun	and	exciting	#10
								Y(c=1)	Y(c=2)	X _k	

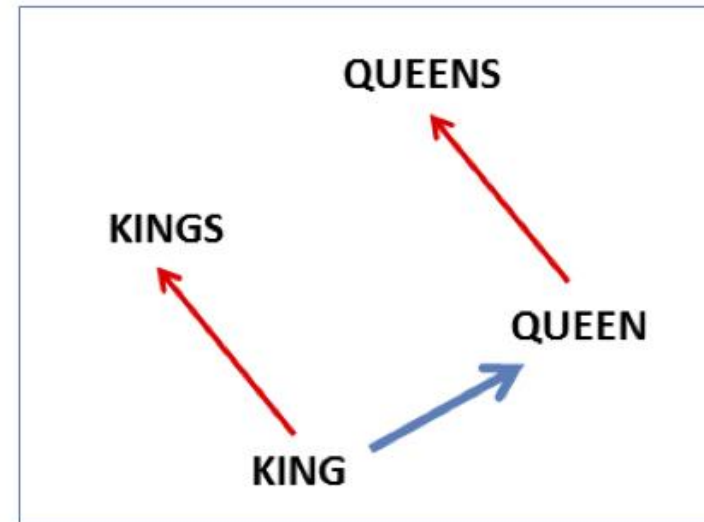
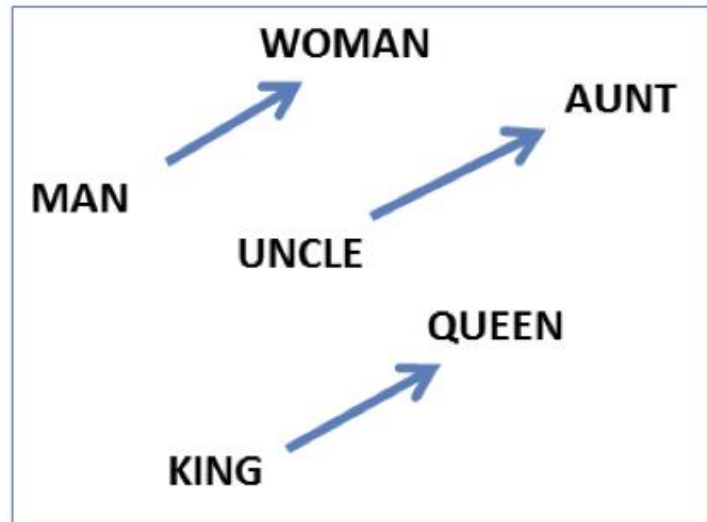
Ref: <https://towardsdatascience.com/an-implementation-guide-to-word2vec-using-numpy-and-google-sheets-13445eebd281>



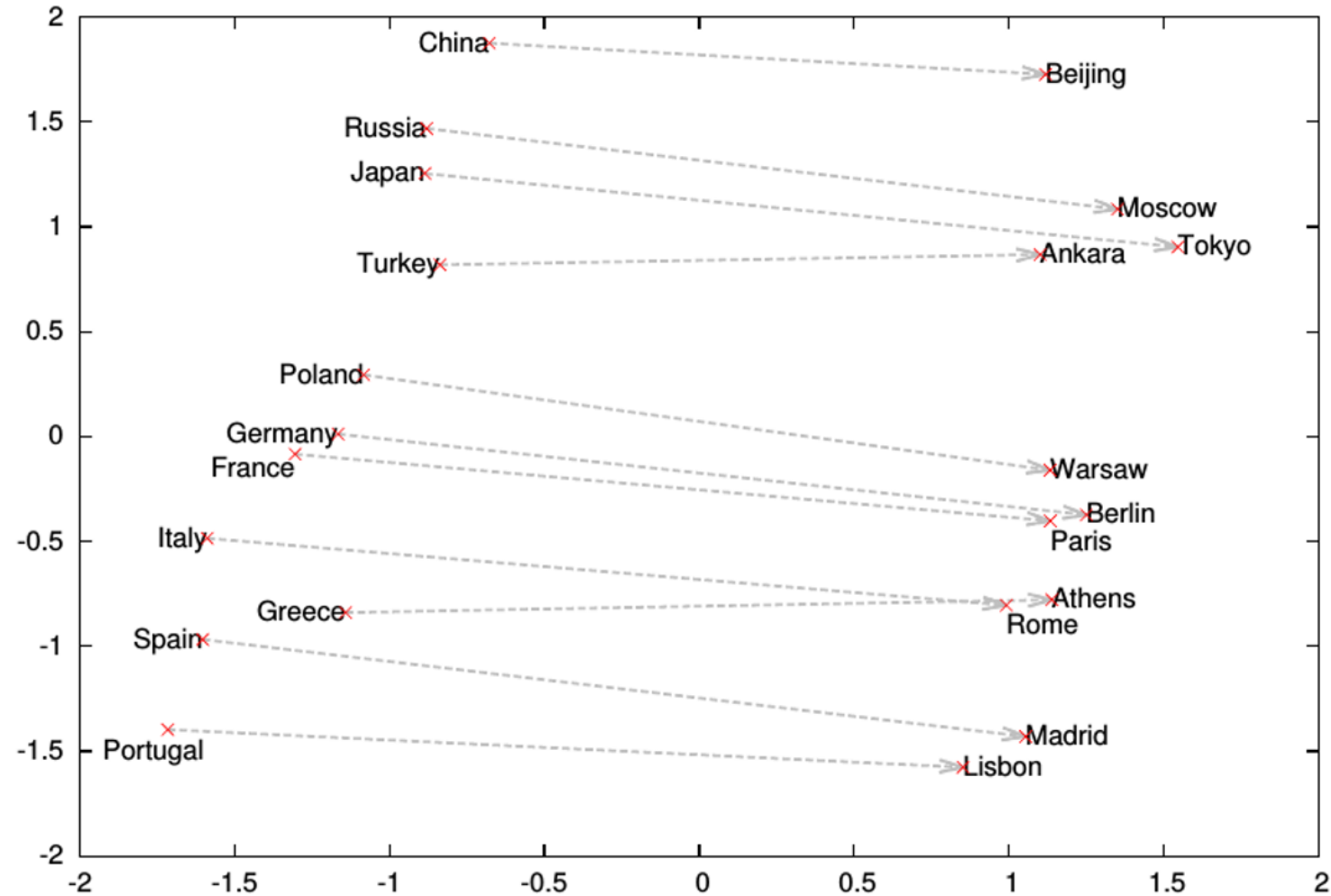
Analogy: Embeddings capture relational meaning!

$\text{vector}('king') - \text{vector}('man') + \text{vector}('woman') \approx \text{vector}('queen')$

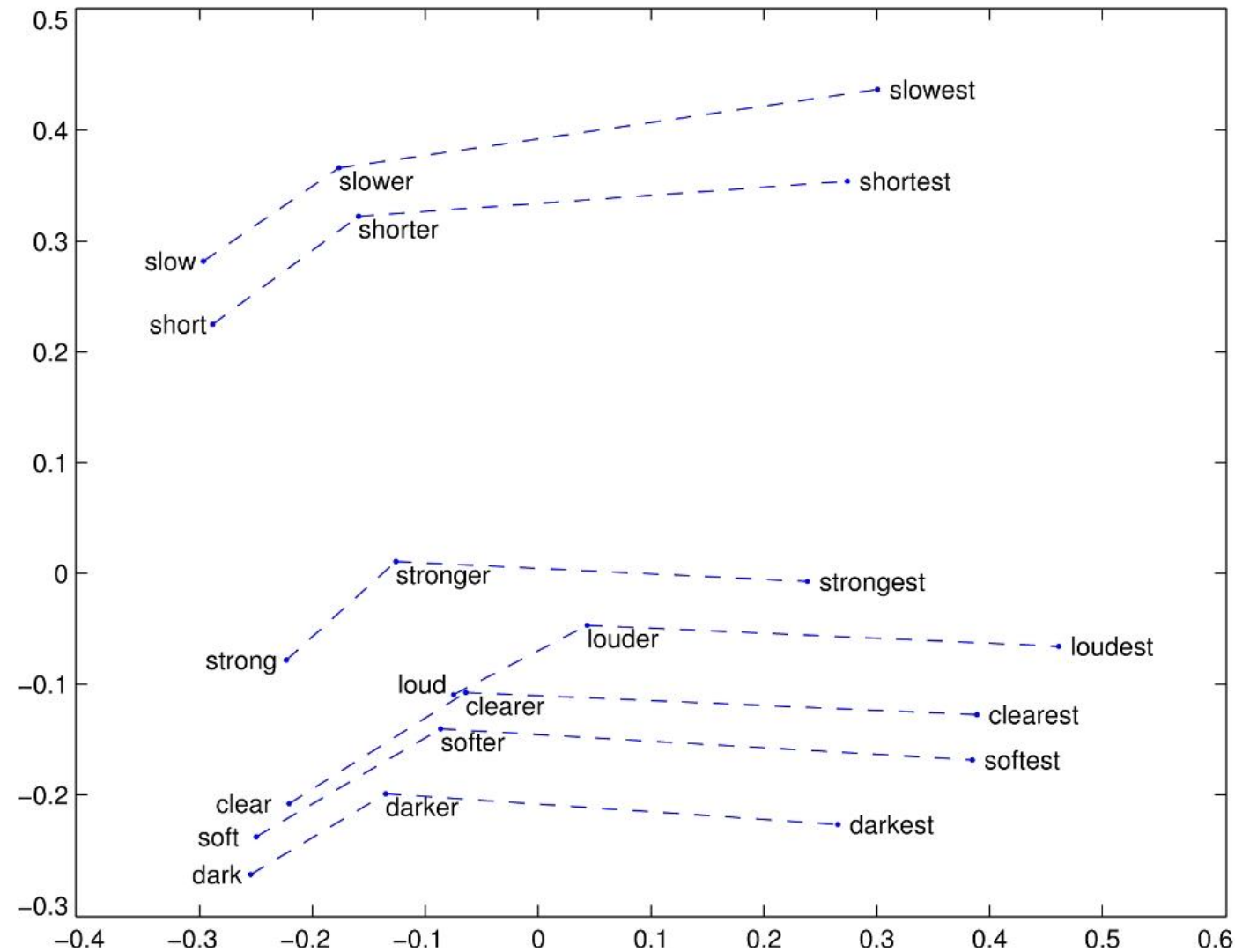
$\text{vector}('Paris') - \text{vector}('France') + \text{vector}('Italy') \approx \text{vector}('Rome')$



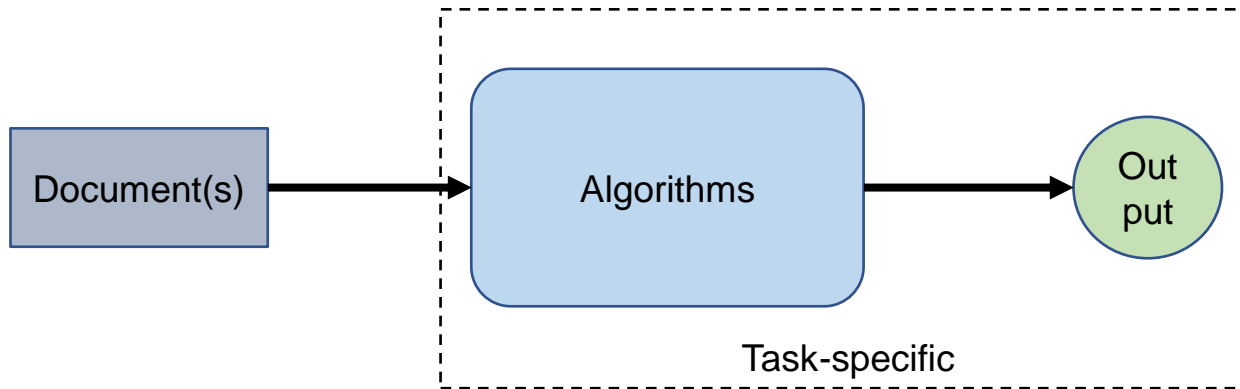
Word analogies



More on embeddings



What About Context???



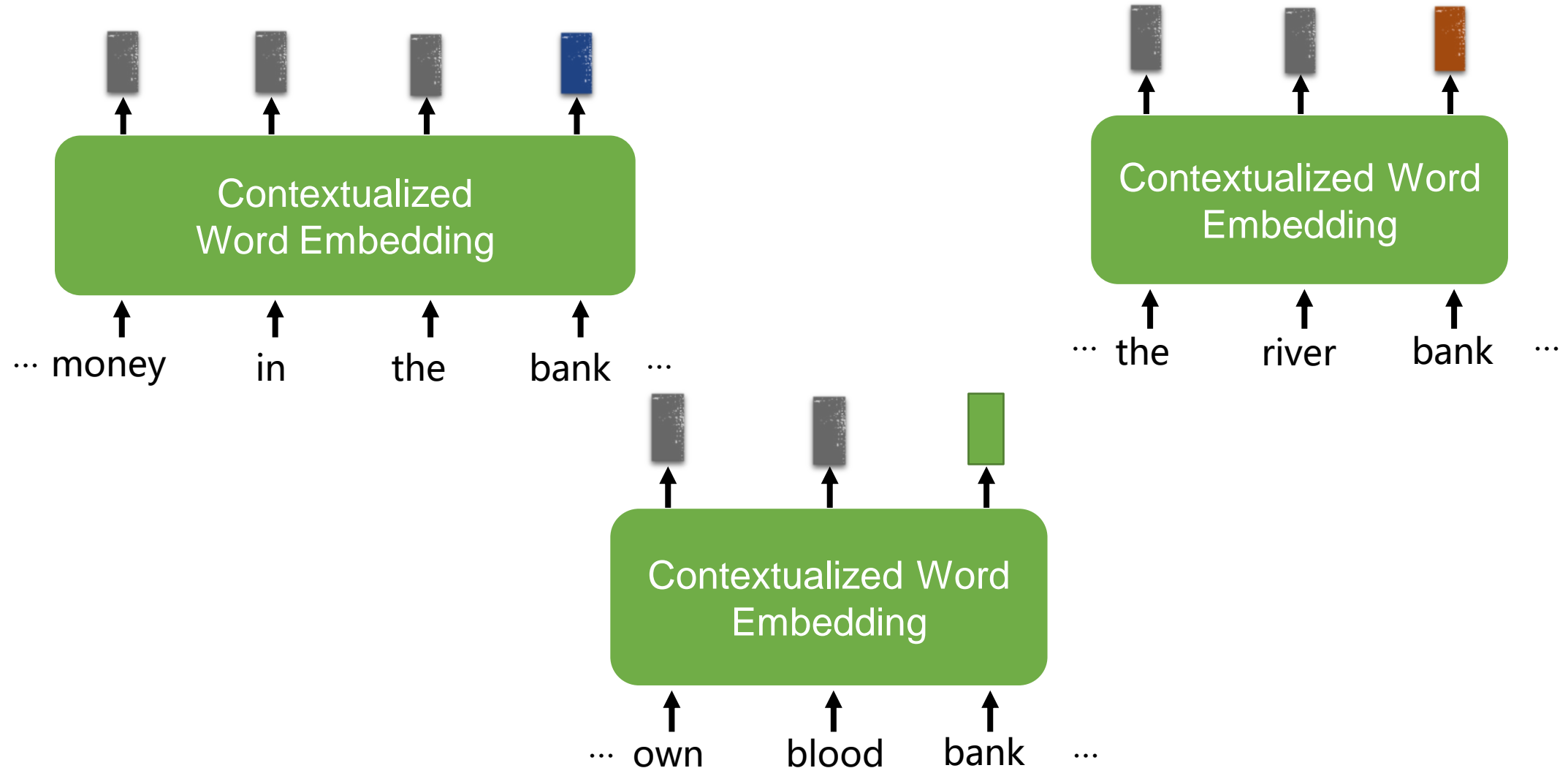
- Banks lower home loan interest rates
- Migratory birds on river bank
- Do not bank on strangers

- CEOs set the target for next quarter
- Nadal wins the first set
- Let R be the set of real numbers

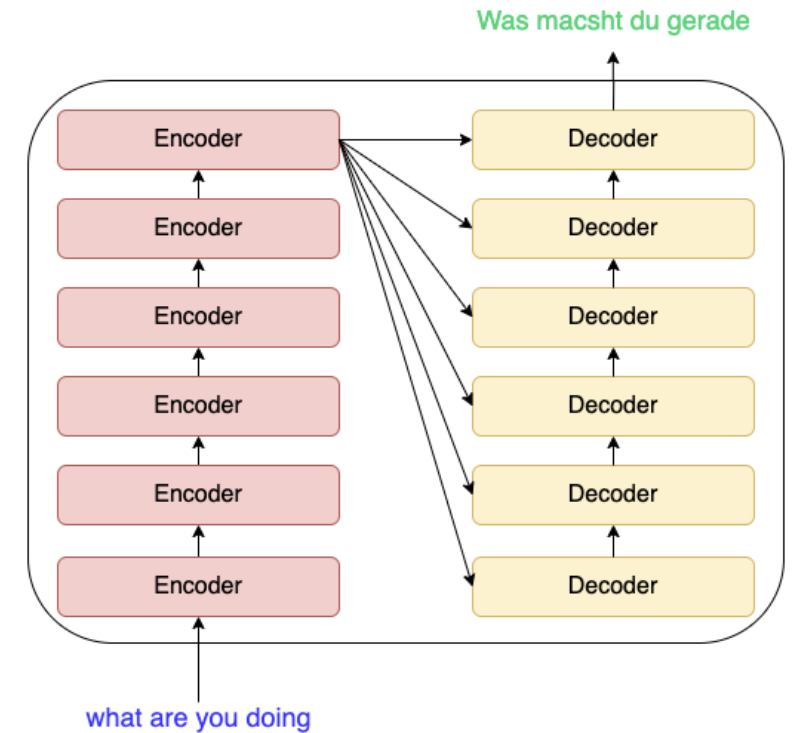
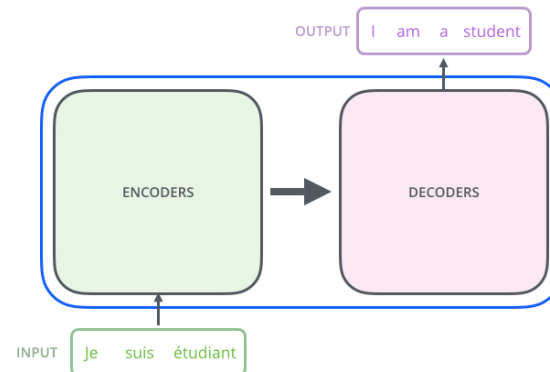
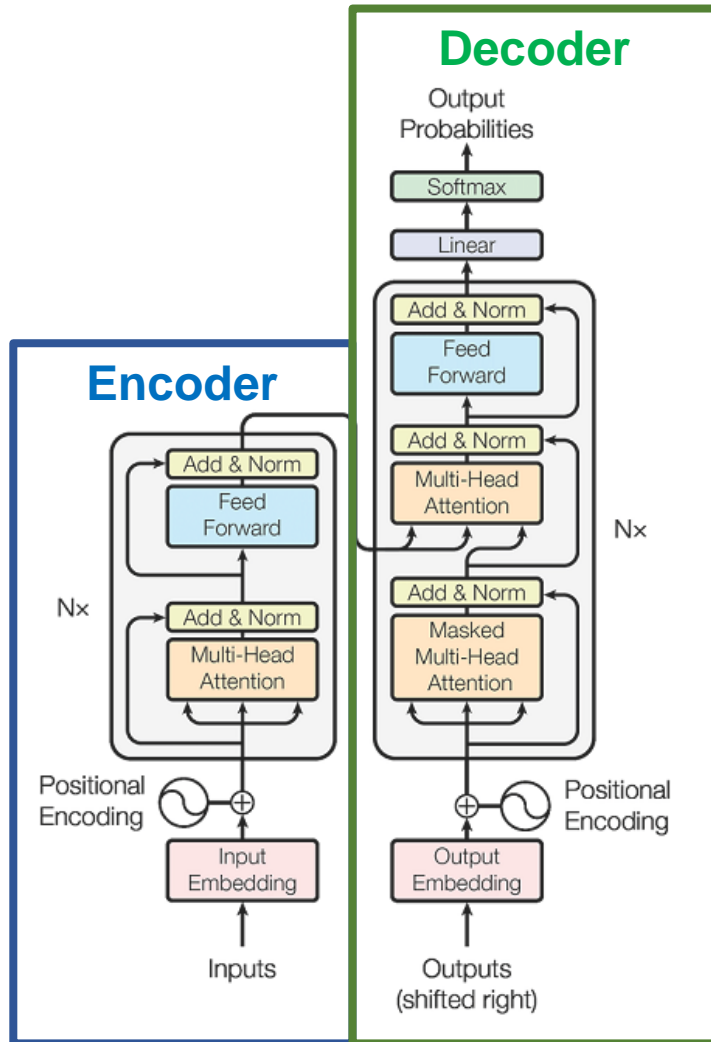
- Word2Vec, GloVe, Fasttext
 - Problem of Synonymy is addressed
- But, polysemy?
- Context-dependent meaning
- Context-dependent word vectors
 - [ELMO](#)
 - [BERT](#)
 - ...



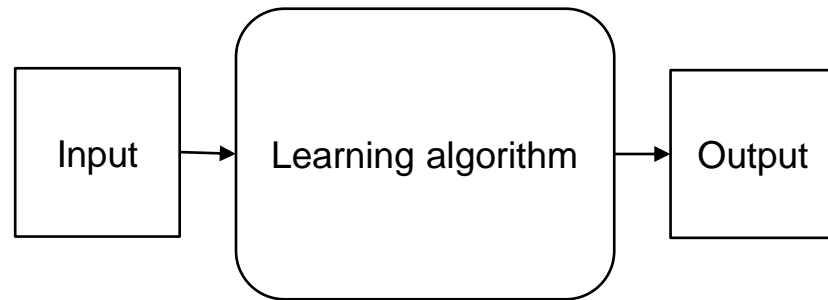
Contextualized Word Embedding



Transformers and BERT

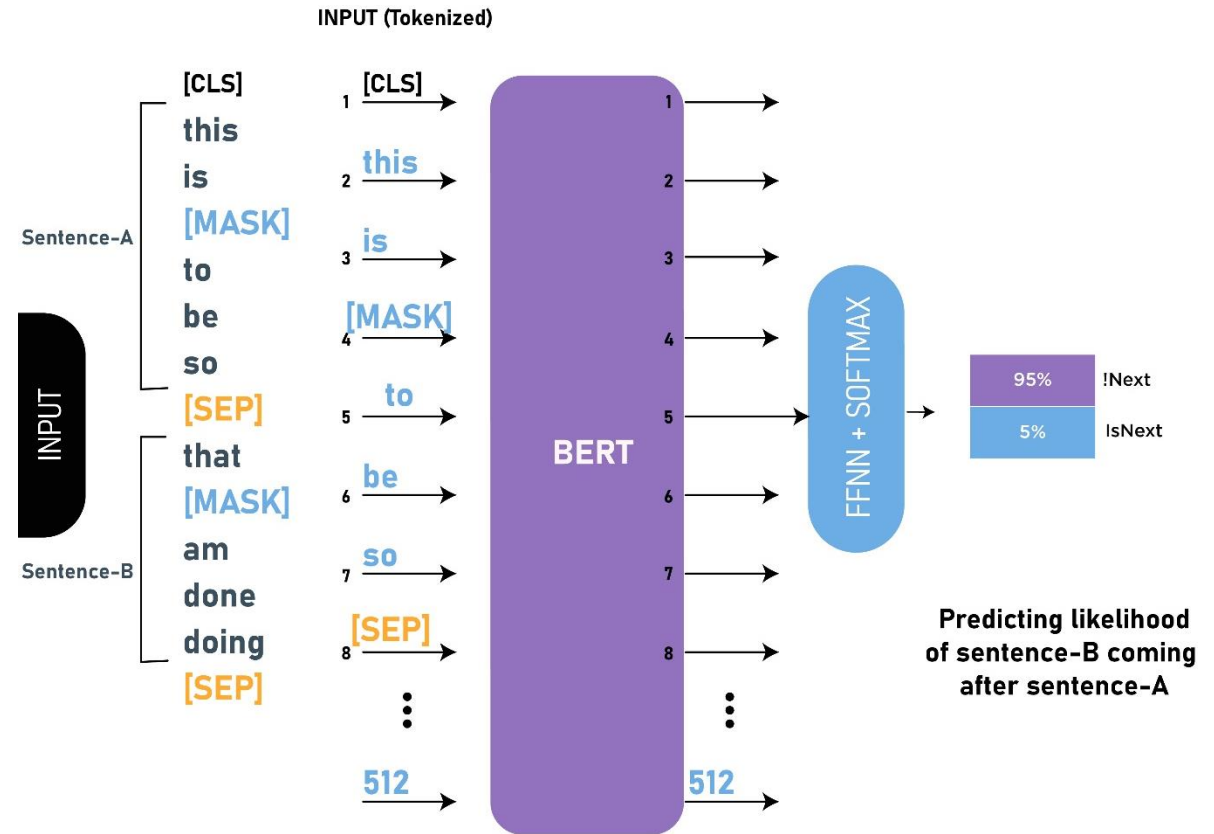


Pretraining Objective

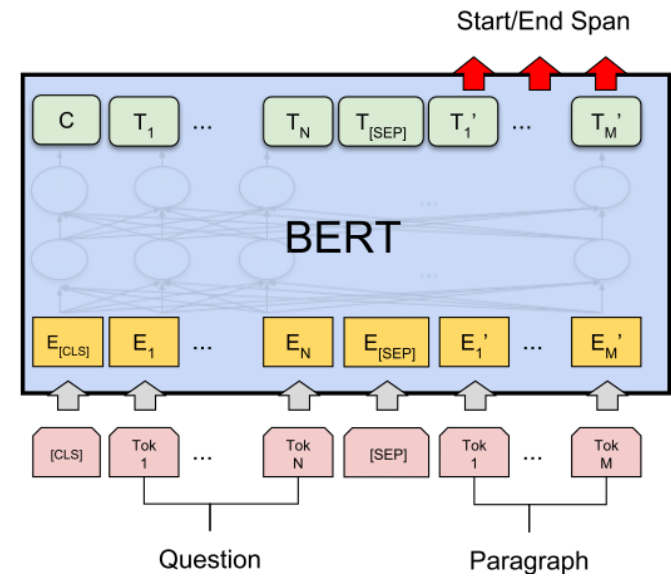
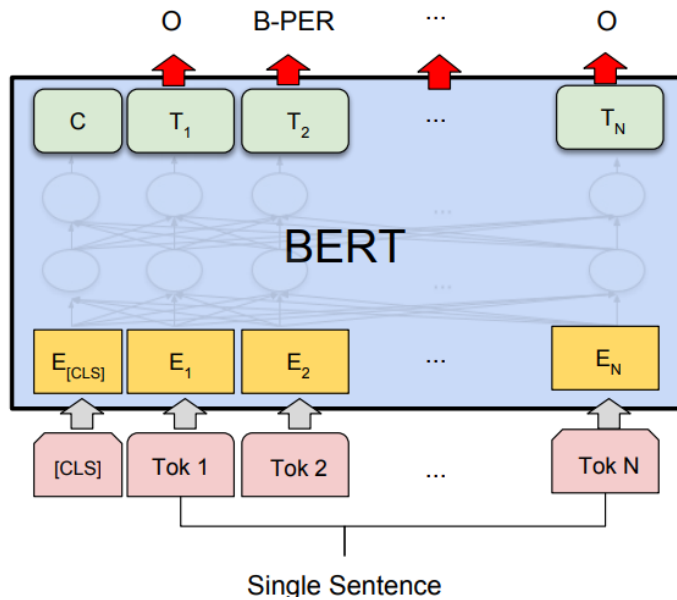


Task 1: Given sentence A with masked words, predict masked words

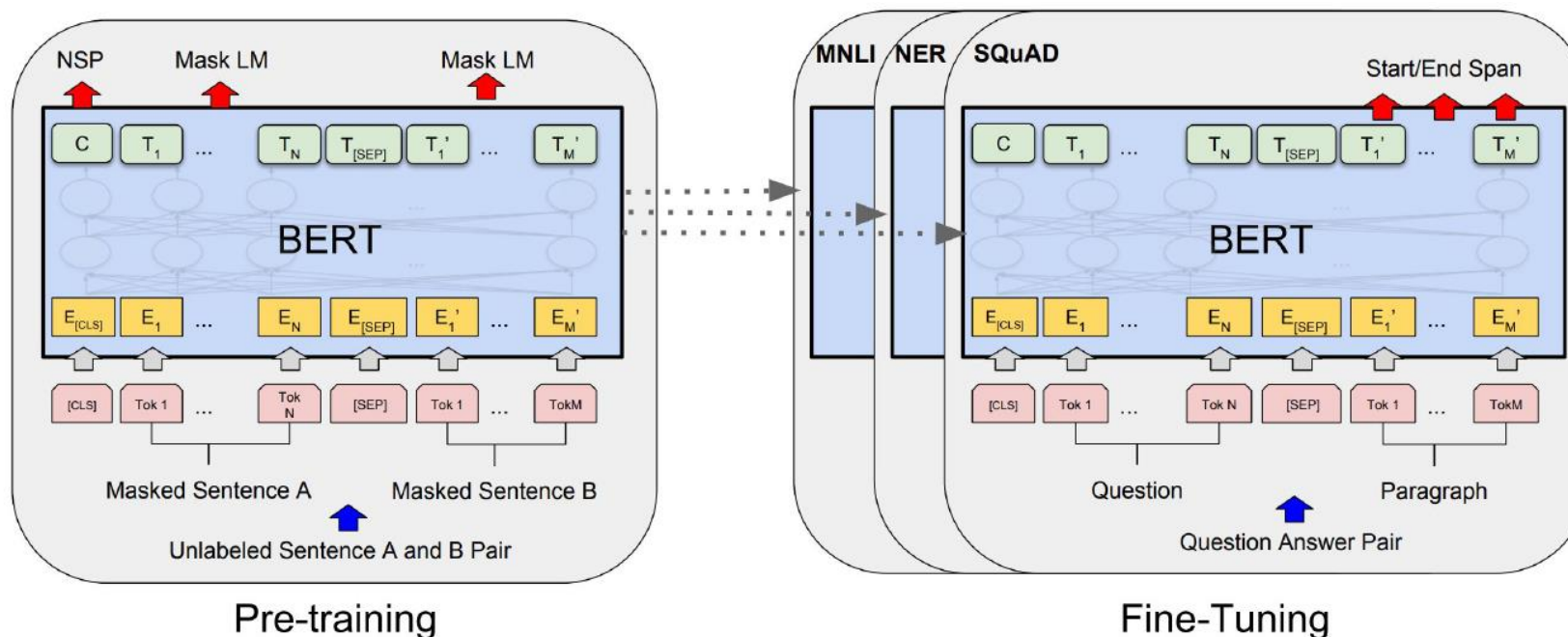
Task 2: Given A and B, predict whether B is the next sentence of A



Fine Tuning



Adapting to Multiple Tasks



System	MNLI-(m/mm)	QQP	QNLI	SST-2	CoLA	STS-B	MRPC	RTE	Average
	392k	363k	108k	67k	8.5k	5.7k	3.5k	2.5k	-
Pre-OpenAI SOTA	80.6/80.1	66.1	82.3	93.2	35.0	81.0	86.0	61.7	74.0
BiLSTM+ELMo+Attn	76.4/76.1	64.8	79.9	90.4	36.0	73.3	84.9	56.8	71.0
OpenAI GPT	82.1/81.4	70.3	88.1	91.3	45.4	80.0	82.3	56.0	75.2
BERT _{BASE}	84.6/83.4	71.2	90.1	93.5	52.1	85.8	88.9	66.4	79.6
BERT _{LARGE}	86.7/85.9	72.1	91.1	94.9	60.5	86.5	89.3	70.1	81.9



Feeding Representations to Tasks

- **Identifying hostile posts**

- Take BERT representation of the document
- Pass it to classifier (SVM/ANN/GBT/...)

- **Document clustering**

- Take BERT representation of the document
- K-Means Clustering

- **Headline generation**

- Take BERT representation of the document
- Pass it to a decoder (sequence generator) model

- **Finding similar prior legal cases**

- Take BERT representation of the case description
- Match it with BERT representations of past case descriptions



What Next?

- Can there be better pretraining tasks?
- Can there be better architectures?
- What about languages other than English?
- Can we align embeddings of terms across languages?
- Application-specific fine tuning
- Single fine-tuning for multiple tasks
- Newer applications to leverage such representations



