Natural Language Processing

AIGC 5501

Chunking & Embeddings

Instructor: Ritwick Dutta
Email: ritwick.dutta@humber.ca

This Week

- Chunking
- Named entity recognition (NER)
- Chunk Evaluation
- Semantic Role Labelling (SRL)
- Vector Semantics
- Embedding models

Chunking

Identifying and classifying the flat, non-overlapping segments of a sentence that constitute the basic non-recursive phrases

- Noun phrases
- Verb phrases
- Adverbial phrases (maybe)
- Prepositional phrases (maybe)

Noun phrase + Verb group chunking

When it’s time for their biannual powwow, the nation’s manufacturing titans typically jet off to the sunny confines of resort towns like Boca Raton and Hot Springs.

↓

Chunker

↓

When $[_{NP}$ it] $[_{V}$'s] $[_{NP}$ time] for $[_{NP}$ their biannual powwow] , $[_{NP}$ the nation] 's $[_{NP}$ manufacturing titans] $[_{V}$ typically jet off] to $[_{NP}$ the sunny confines] of $[_{NP}$ resort towns] like $[_{NP}$ Boca Raton] and $[_{NP}$ Hot Springs] .

Why do we care about chunking?

- Much **faster** than full syntactic analysis

- Supports a number of large-scale NLP tasks
  - NER
  - Information extraction
  - Phrase identification for information retrieval
  - Question Answering

Named Entity Recognition - Intro

Identify all:

- Named locations, named persons, named organizations, dates, times, monetary amounts…
- Fixed set of NE types

Type	Tag	Sample Categories	Example sentences
People	PER	people, characters	**Turing** is a giant of computer science.
Organization	ORG	companies, sports teams	The **IPCC** warned about the cyclone.
Location	LOC	regions, mountains, seas	The **Mt. Sanitas** loop is in **Sunshine Canyon**.
Geo-Political Entity	GPE	countries, states, provinces	**Palo Alto** is raising the fees for parking.
Facility	FAC	bridges, buildings, airports	Consider the **Golden Gate Bridge**.
Vehicles	VEH	planes, trains, automobiles	It was a classic **Ford Falcon**.

Figure 17.1 A list of generic named entity types with the kinds of entities they refer to.

Named Entity Recognition

In fact, the Chinese NORP market has the three CARDINAL most influential names of the retail and tech space – Alibaba GPE , Baidu ORG , and Tencent PERSON (collectively touted as BAT ORG), and is betting big in the global AI GPE in retail industry space . The three CARDINAL giants which are claimed to have a cut-throat competition with the U.S. GPE (in terms of resources and capital) are positioning themselves to become the 'future AI PERSON platforms'. The trio is also expanding in other Asian NORP countries and investing heavily in the U.S. GPE based AI GPE startups to leverage the power of AI GPE . Backed by such powerful initiatives and presence of these conglomerates, the market in APAC AI is forecast to be the fastest-growing one CARDINAL , with an anticipated CAGR PERSON of 45% PERCENT over 2018 - 2024 DATE .

To further elaborate on the geographical trends, North America LOC has procured more than 50% PERCENT of the global share in 2017 DATE and has been leading the regional landscape of AI GPE in the retail market. The U.S. GPE has a significant credit in the regional trends with over 65% PERCENT of investments (including M&As, private equity, and venture capital) in artificial intelligence technology. Additionally, the region is a huge hub for startups in tandem with the presence of tech titans, such as Google ORG , IBM ORG , and Microsoft ORG .

Ambiguity in NER

Name	Possible Categories
Washington	Person, Location, Political Entity, Organization, Vehicle
Downing St.	Location, Organization
IRA	Person, Organization, Monetary Instrument
Louis Vuitton	Person, Organization, Commercial Product

Figure 17.2 Common categorical ambiguities associated with various proper names.

[$_{\text{PER}}$ Washington] was born into slavery on the farm of James Burroughs.
[$_{\text{ORG}}$ Washington] went up 2 games to 1 in the four-game series.
Blair arrived in [$_{\text{LOC}}$ Washington] for what may well be his last state visit.
In June, [$_{\text{GPE}}$ Washington] passed a primary seatbelt law.
The [$_{\text{VEH}}$ Washington] had proved to be a leaky ship, every passage I made...

Figure 17.3 Examples of type ambiguities in the use of the name *Washington*.

Goal - NE Recognition

- Identify the text spans that correspond to the proper names (or dates, times, money expressions)

- Assign the correct named entity (NE) type

Manual NER

- Handcrafted finite state patterns
  - <proper noun> + <corporate designator> → <corp>
- Can’t capture typical naming conventions
  - “Boston Power & Light”
- Time-consuming to define
- Expensive to maintain
- Not portable between languages

NER Sequence Models

Features	Label
American	ORG
Airlines	ORG
,	X
a	X
unit	X
of	X
AMR	ORG
Corp.	ORG
,	X
immediately	X
matched	X
the	X
move	X
,	X
spokesman	X
Tim	PER
Wagner	PER
said	X
.	X

IOB/BIO tag set for NER

<u>BIO Tags</u>

Allows distinguishing adjacent NEs

- We'll fly to **New Orleans** **Friday**

- **Bxxx**: First (ie. Beginning) token in an NE of type XXX

- **Ixxx**: Inside of an entity type XXX

- **O**: Outside of all NEs

NER Sequence Models

Features				Label
American	NNP	B_{NP}	cap	B_{ORG}
Airlines	NNPS	I_{NP}	cap	I_{ORG}
,	PUNC	O	punc	O
a	DT	B_{NP}	lower	O
unit	NN	I_{NP}	lower	O
of	IN	B_{PP}	lower	O
AMR	NNP	B_{NP}	upper	B_{ORG}
Corp.	NNP	I_{NP}	cap_punc	I_{ORG}
,	PUNC	O	punc	O
immediately	RB	B_{ADVP}	lower	O
matched	VBD	B_{VP}	lower	O
the	DT	B_{NP}	lower	O
move	NN	I_{NP}	lower	O
,	PUNC	O	punc	O
spokesman	NN	B_{NP}	lower	O
Tim	NNP	I_{NP}	cap	B_{PER}
Wagner	NNP	I_{NP}	cap	I_{PER}
said	VBD	B_{VP}	lower	O
.	PUNC	O	punc	O

HMMs for NE detection

Just like in POS tagging

- States ***Q***
    - BIO tags
- Observations ***O***
    - Word tokens
- Transition Probabilities ***A***
    - P $(BIOtag_i \mid BIOtag_{i-1})$
- Lexical generation Probabilities ***B***
    - P $(w_i \mid BIOtag_i)$

Find most likely BIO tag sequence using Viterbi

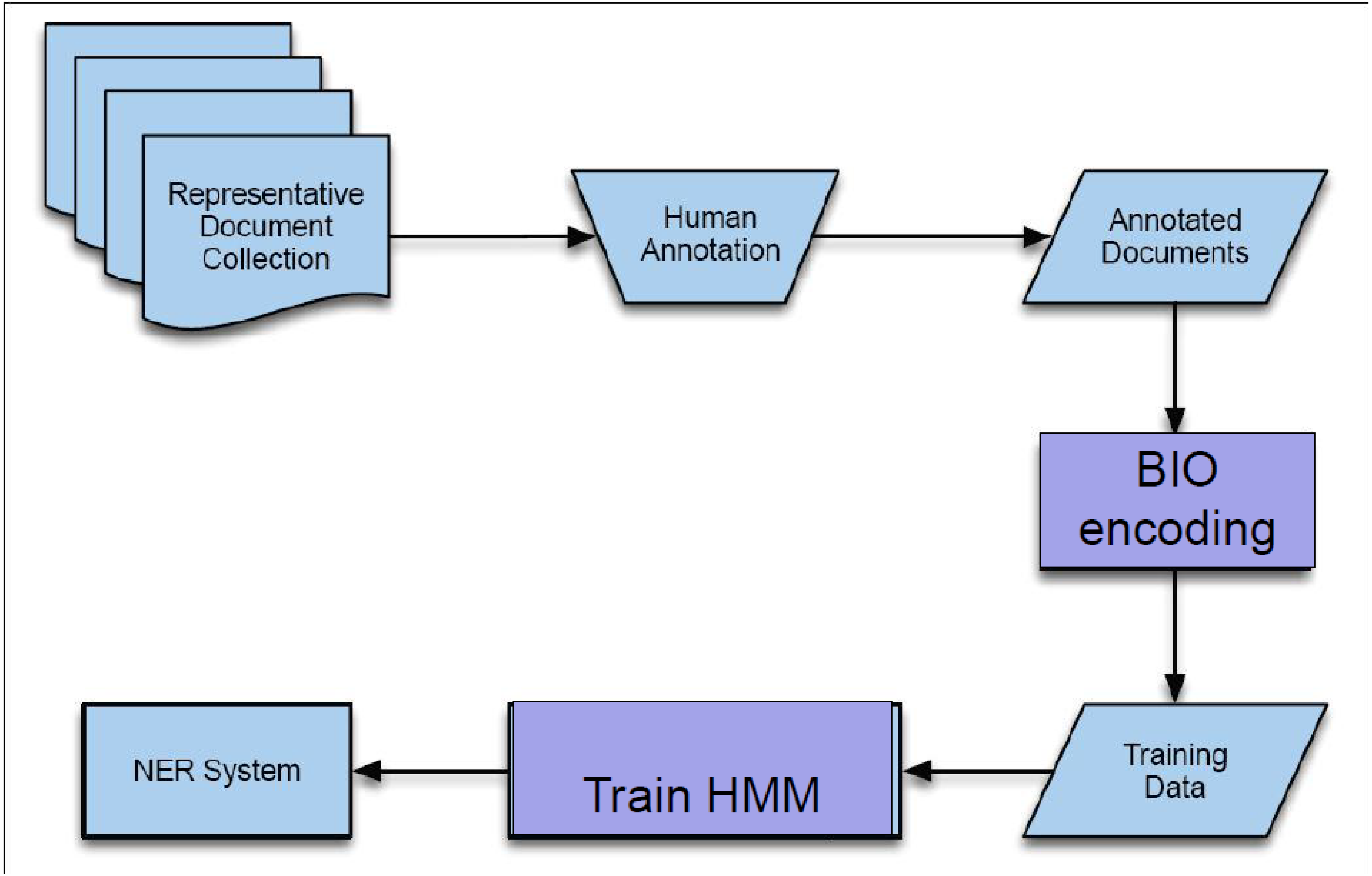
Reconstruct the NEs from the BIO tags

Alternative tag set for NER

<u>**BILOU**</u>

- **Bxxx**: First (ie. Beginning) token in an NE of type XXX
- **Ixxx**: Inside of an entity type XXX
- **Lxxx**: Last token of entity type XXX
- **O**: Outside of all Nes
- **Uxxx**: Single-token (ie. unit) of entity type XXX

End to end process



What kinds of cues are useful for NER?

Table 3.1 Word features, examples and intuition behind them.[2]

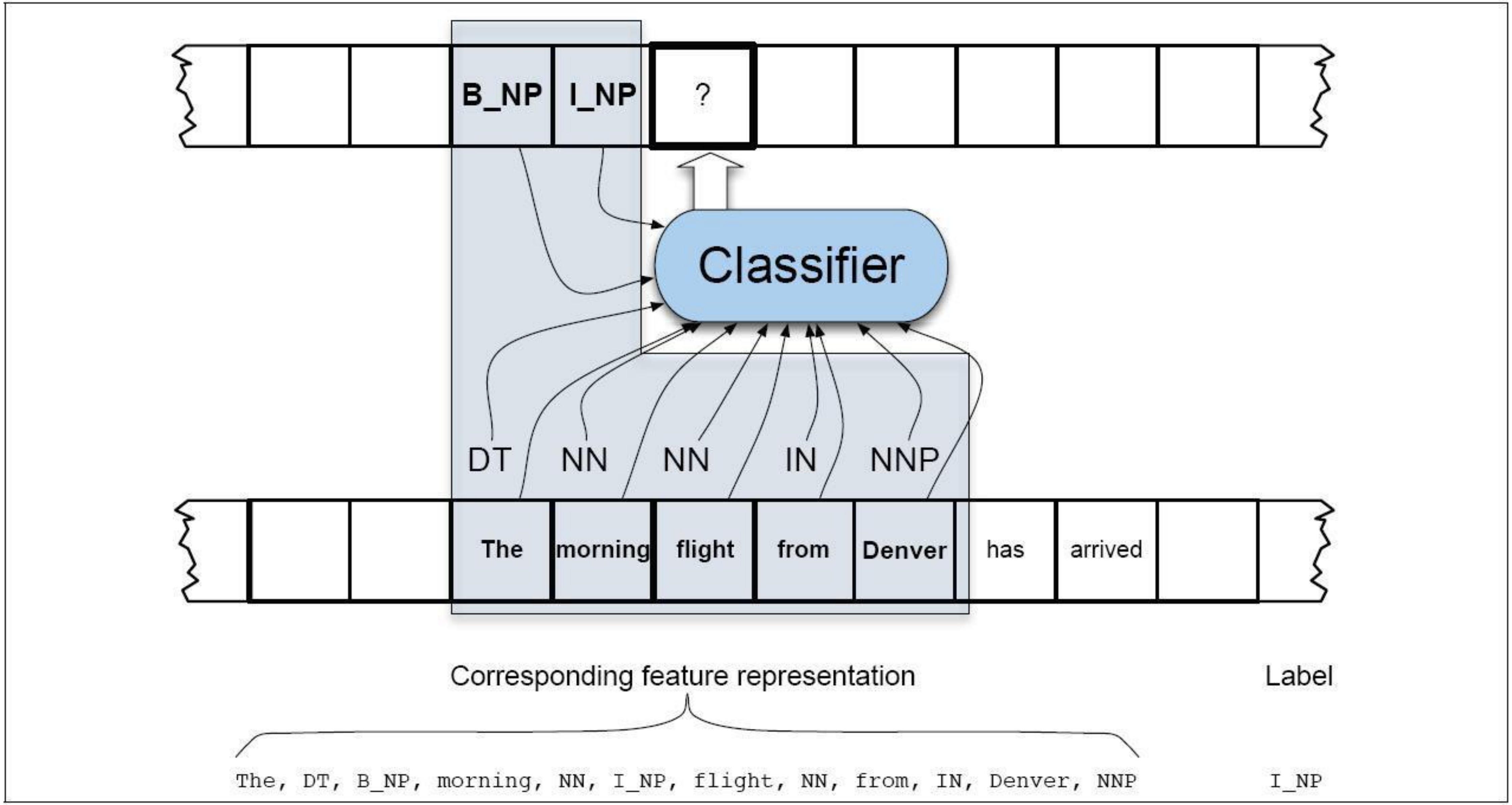
Word Feature	Example Text	Intuition
twoDigitNum	90	Two-digit year
fourDigitNum	1990	Four digit year
containsDigitAndAlpha	A8956-67	Product code
containsDigitAndDash	09-96	Date
containsDigitAndSlash	11/9/89	Date
containsDigitAndComma	23,000.00	Monetary amount
containsDigitAndPeriod	1.00	Monetary amount, percentage
otherNum	456789	Other number
allCaps	BBN	Organization
capPeriod	M.	Person name initial
firstWord	*first word of sentence*	No useful capitalization information
initCap	Sally	Capitalized word
lowerCase	can	Uncapitalized word
other	,	Punctuation marks, all other words

- Part of speech of current word
- Part of speech of preceding word
- Part of speech of the following word
- ….

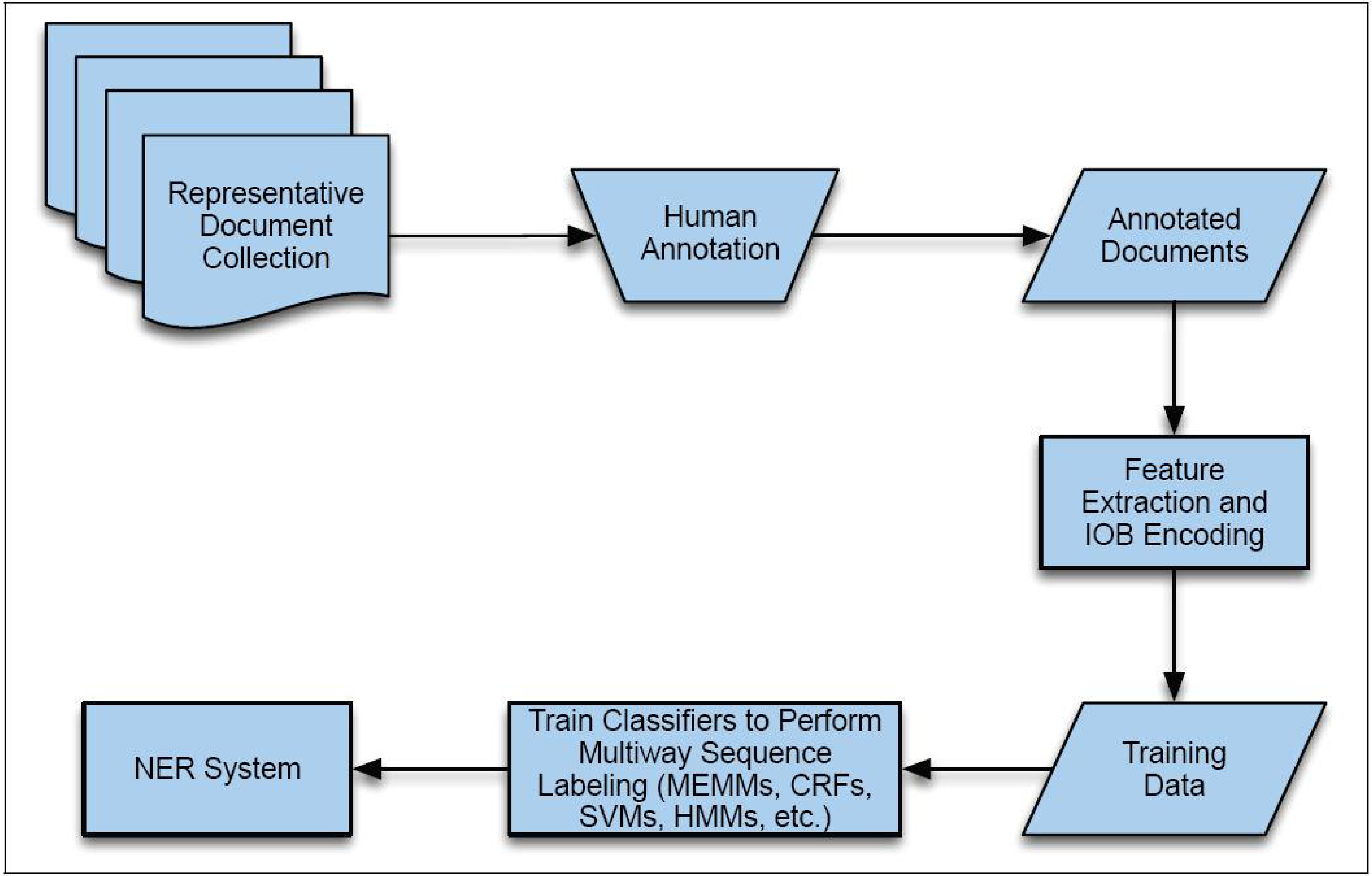
NER data format (with features!)

Features				Label
American	NNP	B_{NP}	cap	B_{ORG}
Airlines	NNPS	I_{NP}	cap	I_{ORG}
,	PUNC	O	punc	O
a	DT	B_{NP}	lower	O
unit	NN	I_{NP}	lower	O
of	IN	B_{PP}	lower	O
AMR	NNP	B_{NP}	upper	B_{ORG}
Corp.	NNP	I_{NP}	cap_punc	I_{ORG}
,	PUNC	O	punc	O
immediately	RB	B_{ADVP}	lower	O
matched	VBD	B_{VP}	lower	O
the	DT	B_{NP}	lower	O
move	NN	I_{NP}	lower	O
,	PUNC	O	punc	O
spokesman	NN	B_{NP}	lower	O
Tim	NNP	I_{NP}	cap	B_{PER}
Wagner	NNP	I_{NP}	cap	I_{PER}
said	VBD	B_{VP}	lower	O
.	PUNC	O	punc	O

Window-based classification



End to end process



Chunk Evaluation

Precision:
#Correct NEs / # Predicted NEs

Recall:
#Correct NEs / #NEs in answer key

F-Measure (F1):
2PR / (P+R)

Note: Evaluation is over NEs, NOT tokens

In [3]:

```
import spacy
nlp = spacy.load("en_core_web_sm")

doc = nlp("NASA awarded Elon Musk’s SpaceX a $2.9 billion contract to build the
lunar lander.")
for token in doc:
    print(token.text, token.ent_iob_, token.ent_type_)
```

[Out] :

```
NASA B ORG
awarded O
Elon B ORG
Musk I ORG
’s I ORG
SpaceX B CARDINAL
a O
$ B MONEY
```

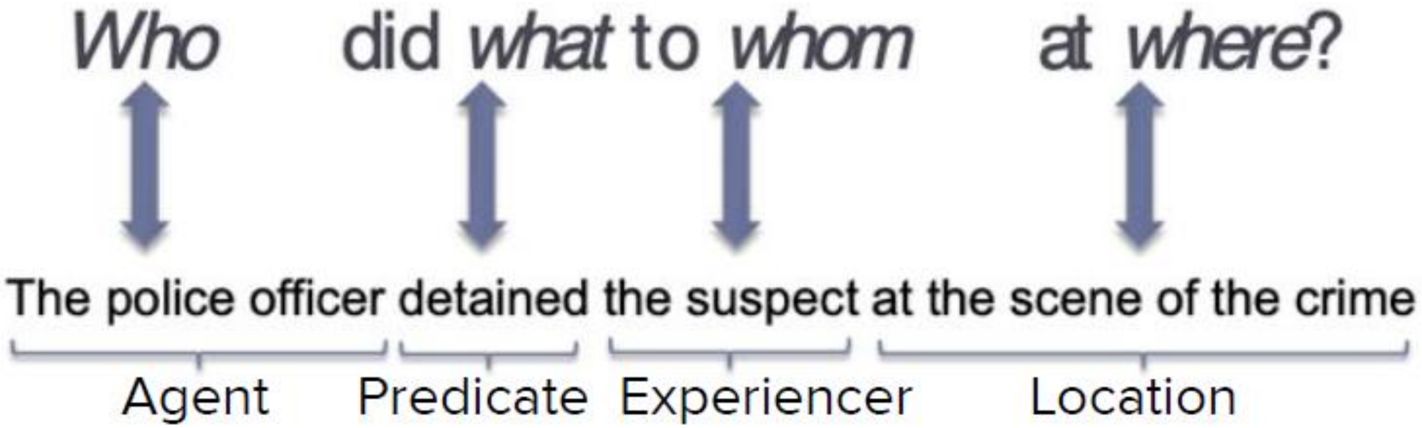
Semantic Role Labeling

Can we figure out that these have the same meaning?

- XYZ corporation bought the stock.
- They sold the stock to XYZ corporation.
- The stock was bought by XYZ corporation.
- The purchase of the stock by XYZ corporation...
- The stock purchase by XYZ corporation...

Semantic Role Labeling

- Predicates (e.g., *bought, sell, purchasing*) represent **events**.
- **Semantic roles** express the abstract roles that predicate arguments can take in the event



J&M Chapter 19

Semantic Role Labeling

- Allows us to make inferences that aren’t possible from purely surface text.

- Useful for machine translation, question answering, summarization, information extraction

- Semantic roles are also called **Thematic roles** or **Theta roles**

A Few Semantic Roles

Thematic Role	Definition	Example
AGENT	The volitional causer of an event	*The waiter* spilled the soup.
EXPERIENCER	The experiencer of an event	*John* has a headache.
FORCE	The non-volitional causer of the event	*The wind* blows debris from the mall into our yards.
THEME	The participant most directly affected by an event	Only after Benjamin Franklin broke *the ice*...
RESULT	The end product of an event	The city built a *regulation-size baseball diamond*...
CONTENT	The proposition or content of a propositional event	Mona asked *"You met Mary Ann at a supermarket?"*
INSTRUMENT	An instrument used in an event	He poached catfish, stunning them *with a shocking device*...
BENEFICIARY	The beneficiary of an event	Whenever Ann Callahan makes hotel reservations *for her boss*...
SOURCE	The origin of the object of a transfer event	I flew in *from Boston*.
GOAL	The destination of an object of a transfer event	I drove *to Portland*.

Semantic Role Labeling

Semantic Role Labeling (SRL) is the task of automatically labelling the semantic roles of each argument according to each predicate in a passage.

[$_{\text{AGENT}}$ John] **broke** [$_{\text{THEME}}$ the window]

John	B-AGENT
broke	B-PREDICATE
the	B-THEME
window	I-THEME

Human Annotated
Deep Learning

Similarity between words

"**fast**" is similar to "**rapid**"
"**tall**" is similar to "**height**"

Question answering:
Q: "How **tall** is Mt. Everest?"
Candidate A: "The official **height** of Mount Everest is 29029 feet"

Similar words in plagiarism detection

MAINFRAMES

Mainframes are primarily referred to large computers with rapid, advanced processing capabilities that can execute and perform tasks equivalent to many Personal Computers (PCs) machines networked together. It is characterized with high quantity Random Access Memory (RAM), very large secondary storage devices, and high-speed processors to cater for the needs of the computers under its service.

Consisting of advanced components, mainframes have the capability of running multiple large applications required by many and most enterprises and organizations. This is one of its advantages. Mainframes are also suitable to cater for those applications (programs) or files that are of very high demand by its users (clients). Examples of such organizations and enterprises using mainframes are online shopping websites such as Ebay, Amazon, and computing-giant

MAINFRAMES

Mainframes usually are referred those computers with fast, advanced processing capabilities that could perform by itself tasks that may require a lot of Personal Computers (PC) Machines. Usually mainframes would have lots of RAMs, very large secondary storage devices, and very fast processors to cater for the needs of those computers under its service.

Due to the advanced components mainframes have, these computers have the capability of running multiple large applications required by most enterprises, which is one of its advantage. Mainframes are also suitable to cater for those applications or files that are of very large demand by its users (clients). Examples of these include the large online shopping websites -i.e. : Ebay, Amazon, Microsoft, etc.

Vector semantics

- **Goal**: Learning **representations** (embeddings) of the meaning of words, directly from their **distributions** in text

- Important for NLP applications that make use of meaning
  - Question Answering, Summarization, Detecting paraphrases or plagiarism and dialogue

Approaches to Convert Text Into Vector

- Label Encoding
- One Hot Encoding
- Bag of Words
Bag of n-grams
- TF-IDF
- Word Embeddings

Term-document matrix

Bag of Words
Bag of n-grams

- Count of word w in a document d:
  - Each document is a count vector in N^V

Document

	As You Like It	Twelfth Night	Julius Caesar	Henry V
battle	1	0	7	13
good	114	80	62	89
fool	36	58	1	4
wit	20	15	2	3

Word / Term

Text Representation Using TF-IDF

TF-IDF

	musk	that	price	market	investor	iphone	itunes	gigafactory	...
Apple article 1	[0	32	45	48	26	7	3	0	...]
Apple article 2	[0	4	3	7	8	6	3	0	...]
Tesla article 3	[15	31	44	43	25	0	0	0	...]
Tesla article 4	[3	0	0	0	0	0	0	1	...]

that → 4/3 -> 1.33 gigafactory → 4/1 → 4 iphone → 4/2 → 2

$$IDF(t) = log\left(\frac{Total\ Documents}{Number\ of\ documents\ term\ t\ is\ present\ in}\right)$$

$$TF(t,d) = \left(\frac{Total\ Number\ of\ time\ term\ t\ is\ present\ in\ doc\ A}{Total\ number\ of\ tokens\ in\ doc\ A}\right)$$

$$IDF(t) = log\left(\frac{Total\ Documents}{Number\ of\ documents\ term\ t\ is\ present\ in}\right)$$

Text Representation (or Vectorizer)

TF-IDF

$$TF-IDF = TF(t,d) * IDF(t)$$

	musk	that	price	market	investor	iphone	itunes	gigafactory	...
Apple article 1	[0	0.05	0.01	0.05	0.05	0.9	0.8	0	...]
Apple article 2	[0	0.002	0.008	0.01	0.02	0.9	0.8	0	...]
Tesla article 3	[0.99	0.05	0.01	0.05	0.05	0	0	0	...]
Tesla article 4	[0.95	0	0	0	0	0	0	0.87	...]

Limitations of tf-idf model

- As n increased, dimensionality, sparsity increases
- Doesn't capture relationship between words
- Doesn't address out of vocabulary (OOV) problem

Word Embeddings

Cosine Similarity: The **cosine** values range from 1 for vectors pointing in the same directions to 0 for orthogonal vectors.

Limitations of tf-idf model

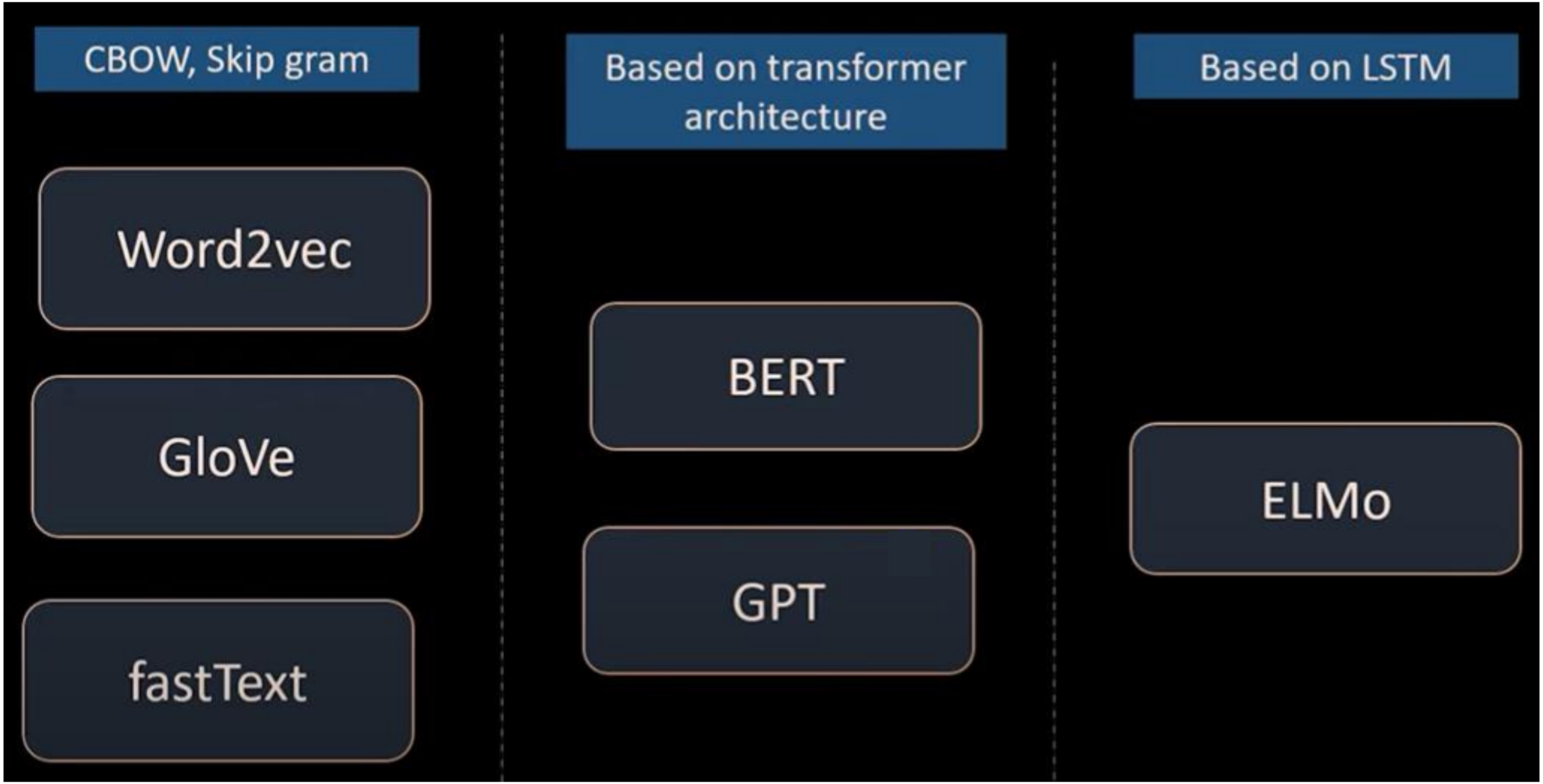
As n increased, dimensionality, sparsity increases

Doesn't capture relationship between words

Doesn't address out of vocabulary (OOV) problem

Word Embeddings

Word Embedding Techniques



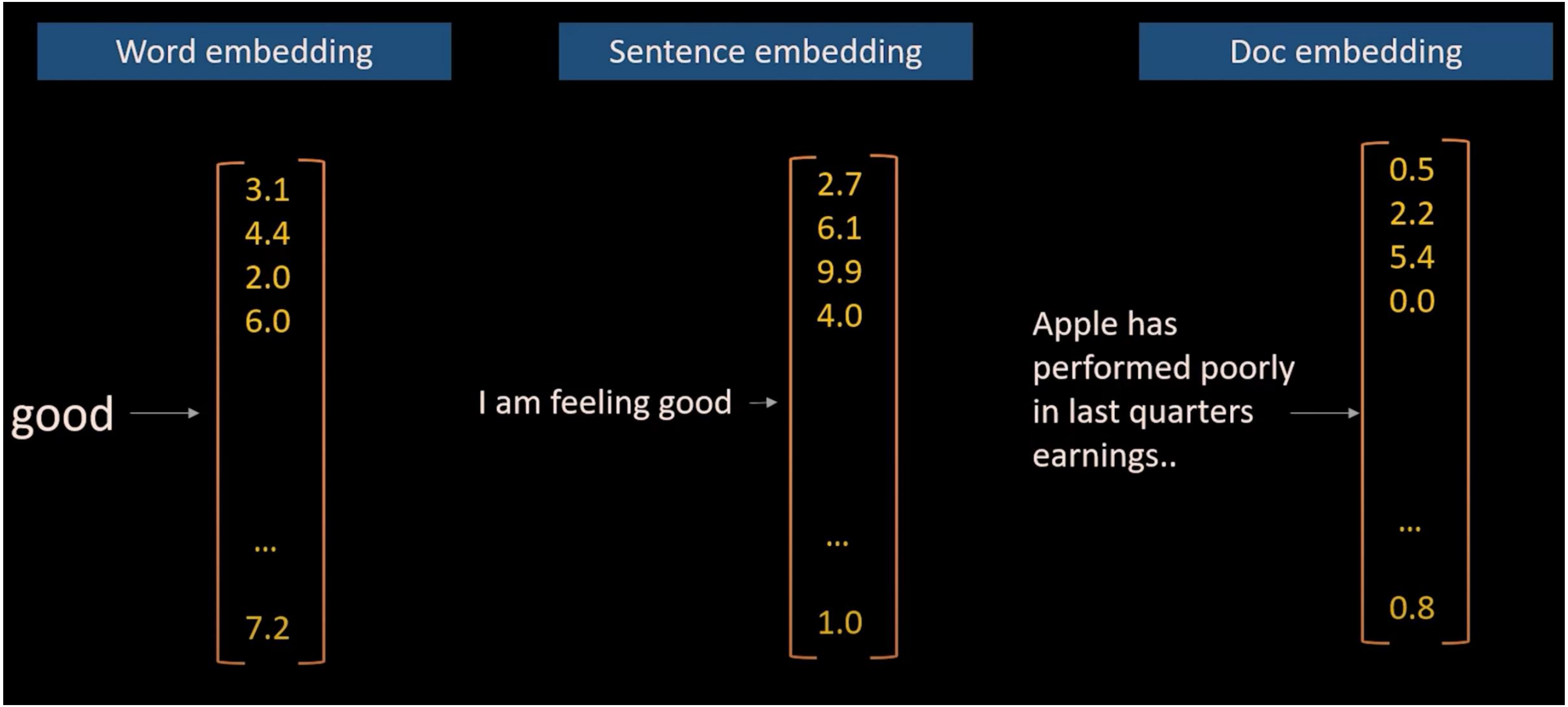
Word2Vec Examples

King – man + woman = Queen

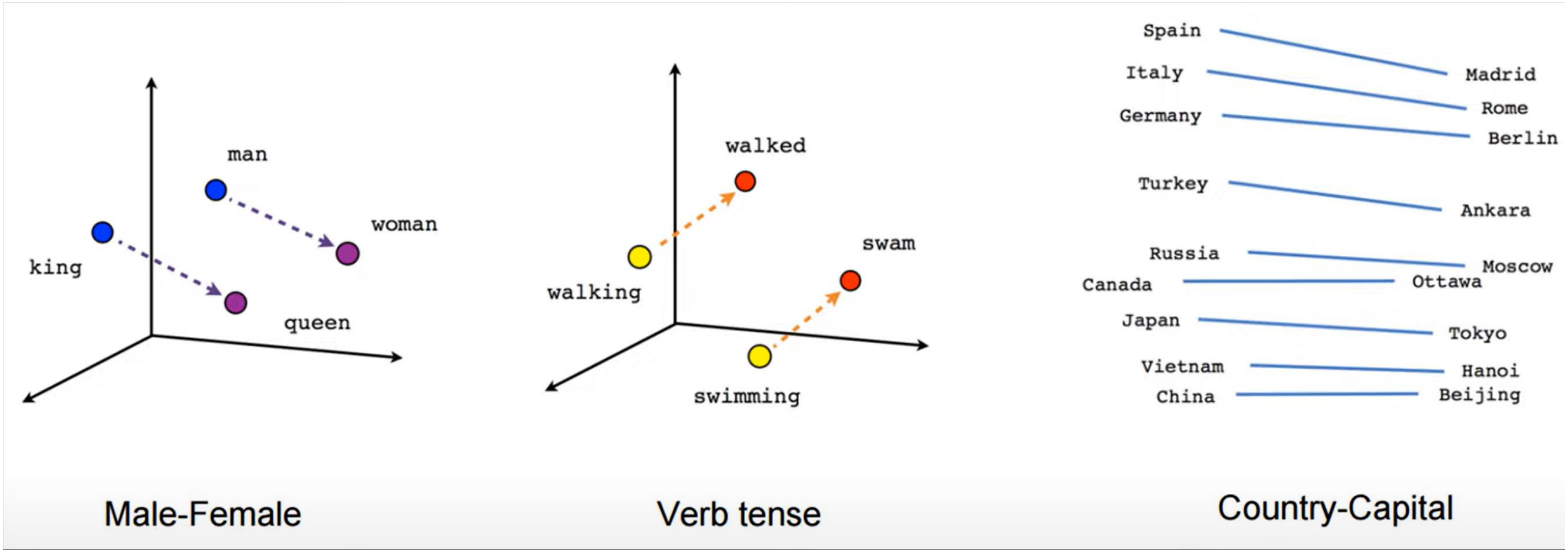
USA – Washington D.C + Delhi = India

Samsun – Galaxy + iPhone = Apple

These Techniques Produce…



Vector Space Representations



Gensim

Gensim is a free Python library designed to automatically extract semantic topics from documents, as efficiently (computer-wise) and painlessly (human-wise) possible.

Gensim library was developed and is maintained by the Czech digital NLP (natural language processing) scientist Radim Řehůřek and his company named RaRe Technologies.

Gensim is designed to process raw, unstructured digital texts (“plain text”).

Features of Gensim Library

Gensim library includes streamed parallelized implementations of the following:

fastText: This feature uses a neural network for word embedding purposes, which is a library for learning word embedding and text classification as well. The library has developed by the Lab of Facebook AI Research known as FAIR. Basically, this model allows us to create or develop a supervised or unsupervised algorithm to obtain vector representations of words.

word2vec: Word2vec is used to create word embedding which is also group of shallow and two-layer neural network models.

doc2vec algorithms: Doc2Vec model is just opposite to the Word2Vec model that is used to develop a vectorized representation of a group of words taken collectively as a single unit.

TF-IDF: Term frequency-inverse document frequency is a numeric statistic in information renewal, throwback how important a word is to a document in a corpus. **It is frequently used by search engines to score and rank a document's relevance as per given a user query. It also used for stop-word refining in text summarization and classification.**

Word2Vec

Word2vec is a group of models that are used to develop word embeddings.

- Word2vec models are generally shallow, two-layer neural networks that are trained to reconstruct semantic contexts of words.

- Word2vec was created by a team of researchers led by Tomas Mikolov at Google and patented.

- There are two main algorithms on which we can train with Word2Vec namely, CBOW (Continuous Bag of Words) and Skip-Grams.

- We will be using pre-trained algorithms

- **Gensim** provides the Word2Vec class for working with a Word2Vec model.

GloVe

GloVe(Global Vectors for Word Representation) is an alternative method to develop word embeddings.

It is purely based on matrix factorization techniques on the “word-context matrix”.

Normally, we can scan our corpus in the following manner:
For every term, we look for context terms within the area defined by a *window size before the term* and *a window size after the term*.

And hence, we give less amount of weight to more distant words.

Lab -4

Exercise 1: https://machinelearningknowledge.ai/beginners-guide-to-named-entity-recognition-ner-in-nltk-library-python/

Exercise 2:

a) Find 2 new text datasets – *big paragraphs*
b) Redo the same exercise
c) Write explanations

Exercise 3: https://melaniewalsh.github.io/Intro-Cultural-Analytics/05-Text-Analysis/03-TF-IDF-Scikit-Learn.html#visualize-tf-idf

Reminder: Delay Penalty – 20% each day