Session 2

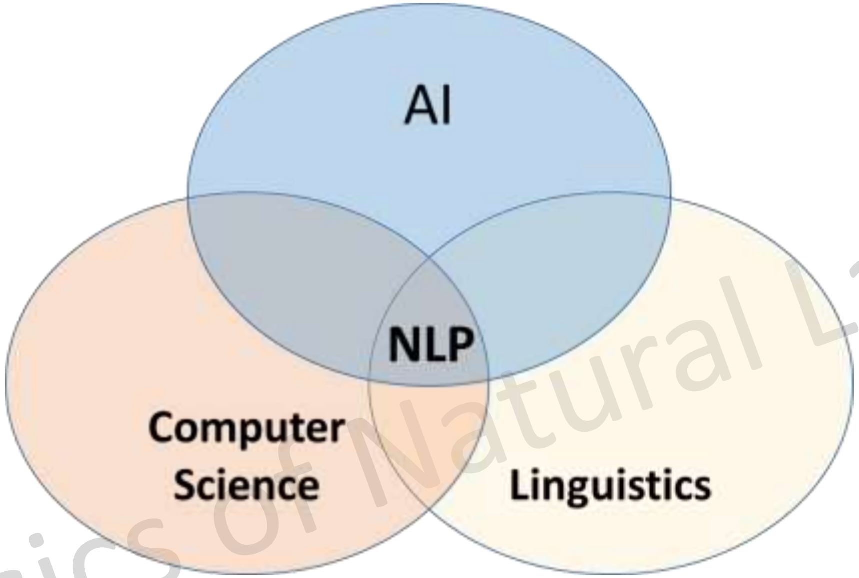
Basics of Linguistics

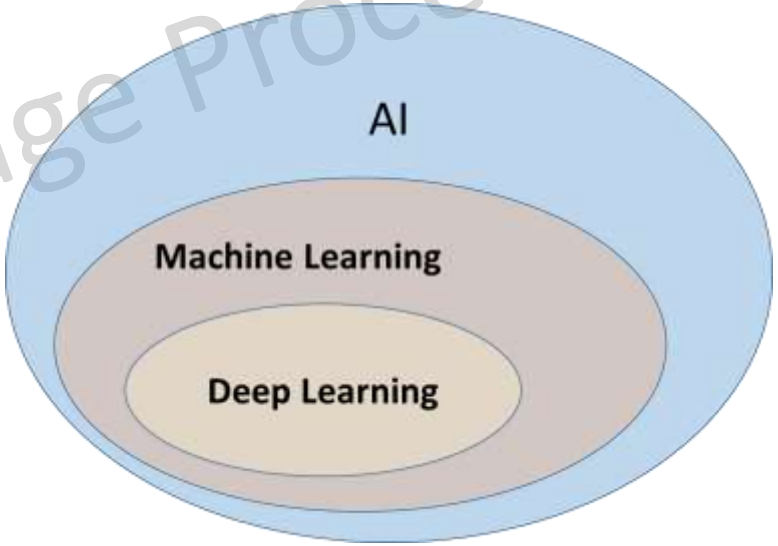
Azam Rabiee, PhD

August 23, 2020

Roadmap

We will review ML/DL methods for NLP





Digikala Academy NLP Events

Basic

- **Session 1.** Introduction
- **Session 2.** Basics of Linguistics
- **Session 3.** Basics of ML
- **Session 4 (Lab).** Effective Word Representation by python

Intermediate

- TBA

Advanced

- TBA

Outline

Session 1: Introduction

- Applications
- Tasks
- Approaches

Session 2. Basics of Linguistics

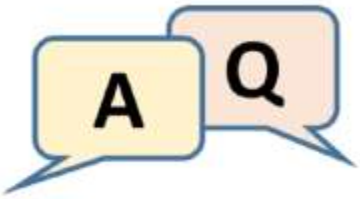
Session 3. Basics of ML

Session 4 (Lab). Effective Word Representation by python

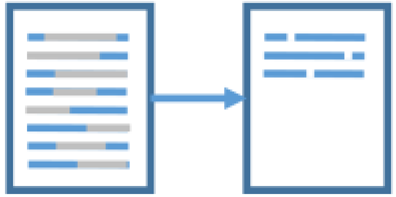
Review: Applications

Machine Translation

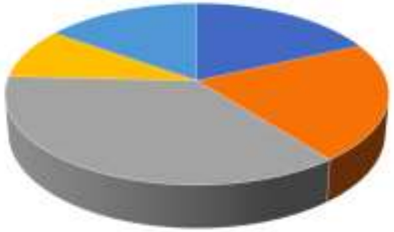
Sentiment Analysis



Question Answering



Automatic Summarization



AI Marketing

Text / Document Classification

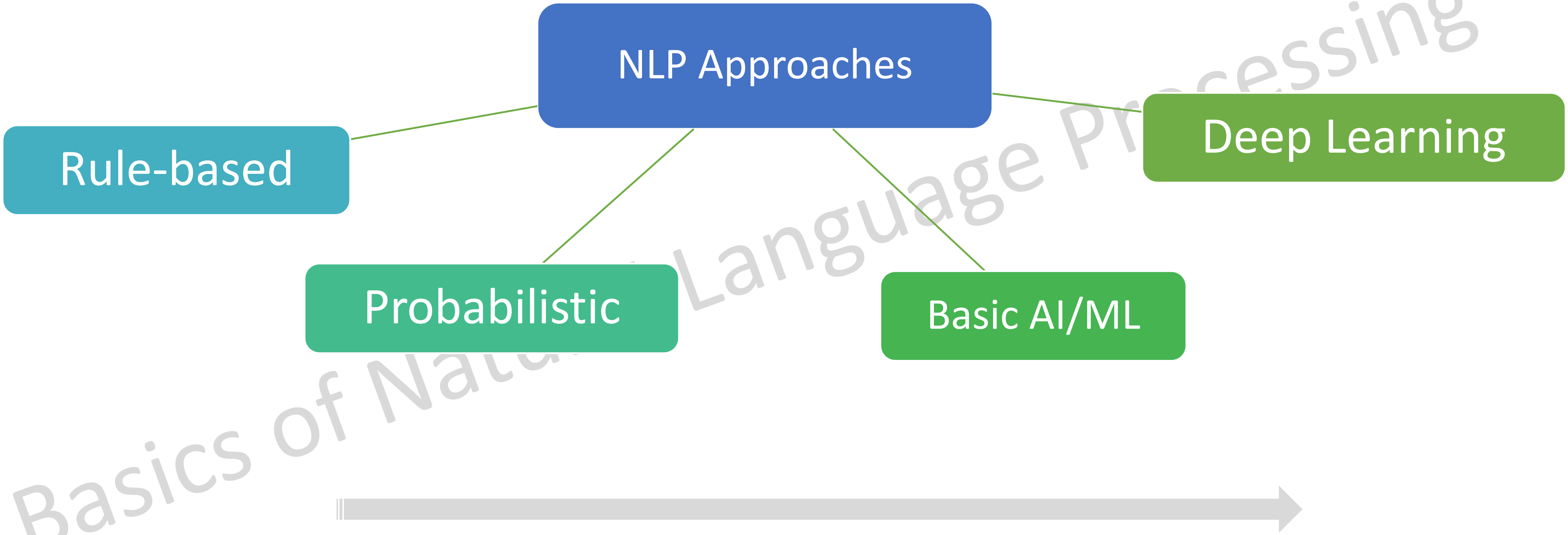
Speech Recognition

Handwritten Character Recognition

Spell Checking

Review: Tasks

Approaches


NLP Approaches
Rule-based
Probabilistic
Basic AI/ML
Deep Learning

Outline

Session 1: Introduction

Session 2. Basics of Linguistics

- Components
- Challenges
- Vectorization

Session 3. Basics of ML

Session 4 (Lab). Effective Word Representation by python

Outline

Session 1: Introduction

Session 2. Basics of Linguistics

- **Components**
- Challenges
- Vectorization

Session 3. Basics of ML

Session 4 (Lab). Effective Word Representation by python

Components

Tokenization

Breaking up the sequence of characters into sentences and words

`A dog is chasing a boy on the playground.`

Components

Tokenization

Tokenizer also extracts token features, such as:

- Capitalization
- Inclusion of digits
- Punctuation
- Special characters

```
The MRI, I took on 2019, priced at $500!!
```

Components

Part-of-Speech (PoS) Tagging

A	dog	is	chasing	a	boy	on	the	playground
article	noun	aux	verb	article	noun	preposition	article	noun

Components

Stemming

Normalizing the word to its stem word

(not necessarily the dictionary word)

```
stemmer(`studies`) ➔ `studi`
stemmer(`studied`) ➔ `studi`
stemmer(`studying`) ➔ `studi`
stemmer(`study`) ➔ `studi`
```

Components

Lemmatization

Normalizing the word to its stem dictionary word

```
lemmatizer(`studies`) ➔ `study`
lemmatizer(`studied`) ➔ `study`
lemmatizer(`studying`) ➔ `study`
```

Stemming vs. Lemmatization

```
stemmer(`am`)          ➔ `am`          lemmatizer(`am`,   pos=`v`)  ➔ `be`
stemmer(`was`)         ➔ `was`         lemmatizer(`was`,  pos=`v`)  ➔ `be`
stemmer(`were`)        ➔ `were`        lemmatizer(`were`, pos=`v`)  ➔ `be`
stemmer(`is`)          ➔ `is`          lemmatizer(`is`,   pos=`v`)  ➔ `be`
stemmer(`are`)         ➔ `are`         lemmatizer(`are`,  pos=`v`)  ➔ `be`

stemmer(`studying`) ➔ `studi`          lemmatizer(`studying`, pos=`v`) ➔ `study`
                                       lemmatizer(`studying`, pos=`n`) ➔ `studying`
```

Stemming vs. Lemmatization

Stemming	Lemmatization
Does not consider the context	Considers the context & PoS
Fast	Slow
Less accurate	Accurate

Components

Chunking / Parsing

Extract meaningful phrases, such as:

- noun phrases
- verb phrases
- prepositional phrases
- adjective phrases
- adverbal phrases

to make sure if a sentence **syntactically** is correct

Components

Chunking / Parsing

<table>
<tr><td><u>A</u></td><td><u>dog</u></td><td><u>is</u></td><td><u>chasing</u></td><td><u>a</u></td><td><u>boy</u></td><td><u>on</u></td><td><u>the</u></td><td><u>playground</u></td></tr>
<tr><td>article</td><td>noun</td><td>aux</td><td>verb</td><td>article</td><td>noun</td><td>preposition</td><td>article</td><td>noun</td></tr>
<tr><td colspan="2">noun phrase</td><td colspan="2">complex verb</td><td colspan="2">noun phrase</td><td></td><td colspan="2">noun phrase</td></tr>
<tr><td colspan="2"></td><td colspan="4">verb phrase</td><td colspan="3">prepositional phrase</td></tr>
<tr><td colspan="2"></td><td colspan="7">verb phrase</td></tr>
</table>

Components

Chunking / Parsing

Parsing one word at a time

Parsing two words at a time

N-gram: Parsing N word(s) at a time

Components

Chinking

Removing stop words

A	dog	is	chasing	a	boy	on	the	playground
article	noun	aux	verb	article	noun	preposition	article	noun

dog **chasing** **boy** **playground**

Components

Semantic Analysis

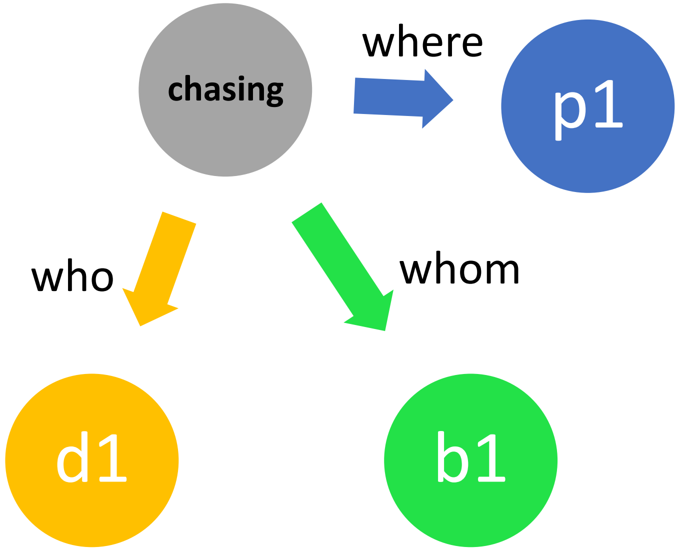
Understanding the meaning, relationships, and interpretation of words

Components

Semantic Analysis

A	dog	is	chasing	a	boy	on	the	playground
	d1				b1			p1

Dog(d1)
Boy(b1)
Playground(p1)
Chasing(d1, b1, p1)



Components

Disclosure Integration

Integrating sentences to take into account the context

```
I know Martin Cook. He works at Google.
```

A sense of the context: Martin Cook works at Google.

Components

Pragmatic Analysis

```
A: - Do you know what time it is?
   - Yes. it’s 6:00 pm.

B: - Do you know what time it is?
   - Sorry. I won’t be late anymore.
```

Components of NLP

Lexical Analysis		
Tokenization	PoS tagging	Stemming/lemmatization

↓

Syntactic Analysis	
Chunking / Parsing	Chinking

↓

Semantic Analysis

↓

Discourse Analysis

↓

Pragmatic Analysis

Which type of analysis can find mistakes in the following sentences?

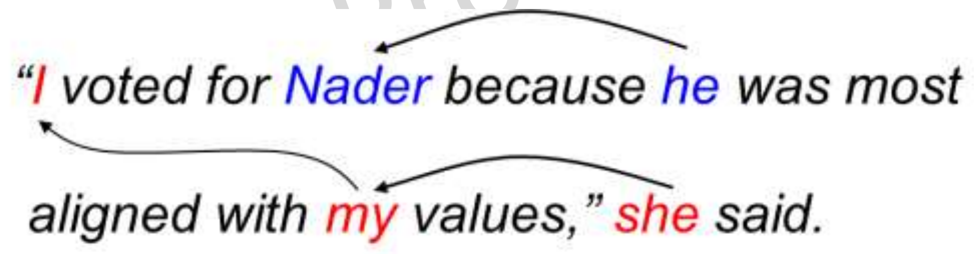
A: Avoid playing when you are **in** tired

B: **Hot** ice-cream

1. A: Syntactic Analysis, B: Semantic Analysis
2. A: Semantic Analysis, B: Syntactic Analysis

Coreference resolution is the task of finding all expressions that refer to the same entity in a text. The task is mostly related to:

1. Tokenization
2. Lemmatization
3. Discourse Analysis



AI-hard problem

Outline

Session 1: Introduction

Session 2. Basics of Linguistics

- Components
- **Challenges**
- Vectorization

Session 3. Basics of ML

Session 4 (Lab). Effective Word Representation by python

Challenges

Tokenization

The main challenge for the sentence tokenizer:

Is it one sentence or two?

```
A dog is chasing Mr. Cook on the playground.
```

Challenges

PoS Tagging

PoS is crucial for syntactic and semantic analysis

```
Can you help me with the can?
```

Challenges

Name Entity Recognition (NER)

```
Cook said: “he is coming.”
```

Challenges

Context Understanding / Pragmatic Analysis

```
A: - Do you know what time it is?
   - Yes. it’s 6:00 pm.


B: - Do you know what time it is?
   - Sorry. I won’t be late anymore.
```

Challenges

Ambiguity

```
I saw a man on a hill with a telescope.
```

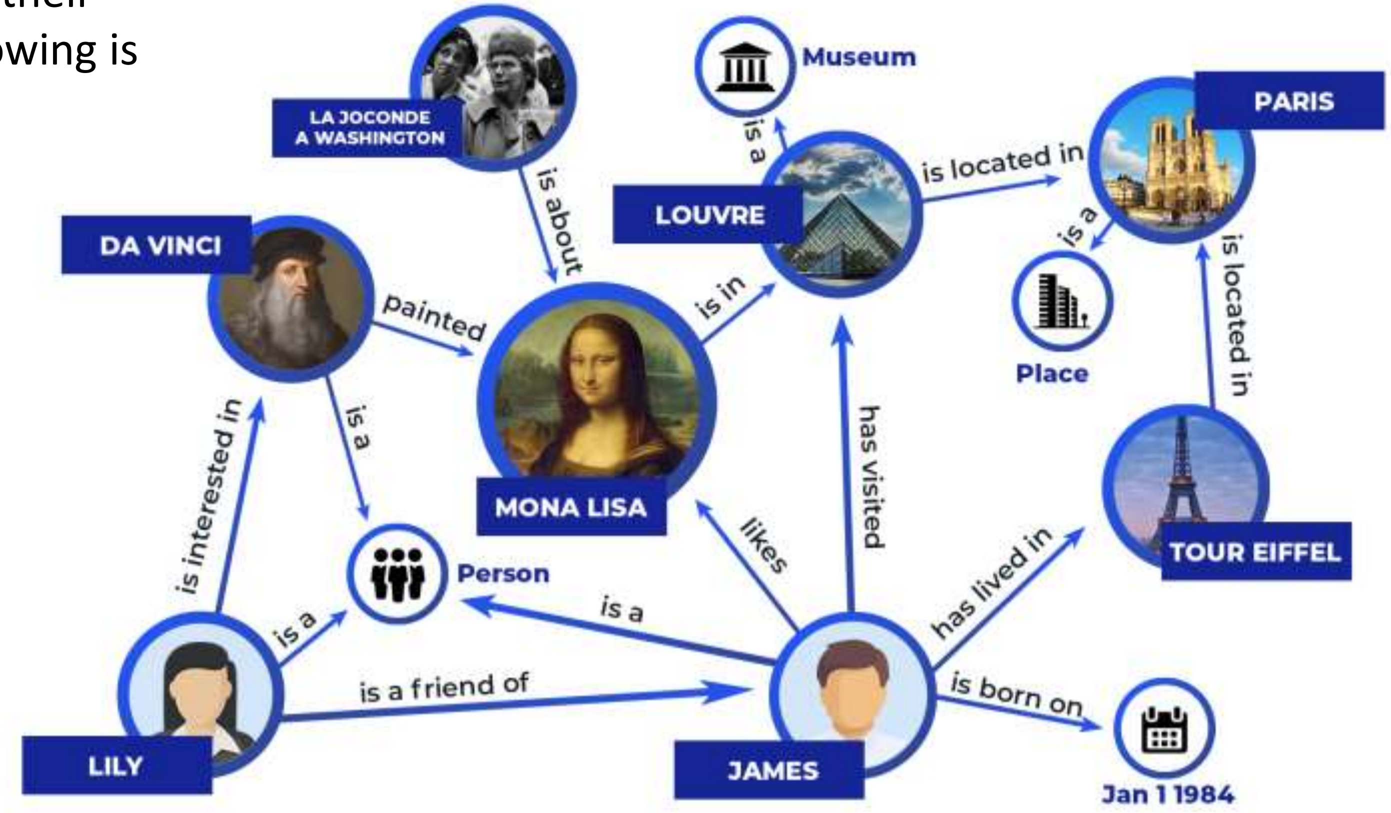
- There is a man on the hill, and I watched him with my telescope.
- There is a man on the hill, and he has a telescope.
- I’m on a hill, and I saw a man using my telescope.
- I’m on a hill, and I saw a man who has a telescope.
- There is a man on a hill, and I saw him something with my telescope.

Current NLP Challenges

- Tokenization
- PoS Tagging
- Named Entity Recognition (NER)
- Context understanding
- Ambiguity in speech
- Semantic analysis
- Building an appropriate vocabulary
- Linking the components of a created vocabulary

Knowledge Graph (KG) is a network of entities, their properties, and relationships. Which of the following is highly related to KG?

1. Stemming
2. Semantic Analysis
3. Tokenization



Outline

Session 1: Introduction

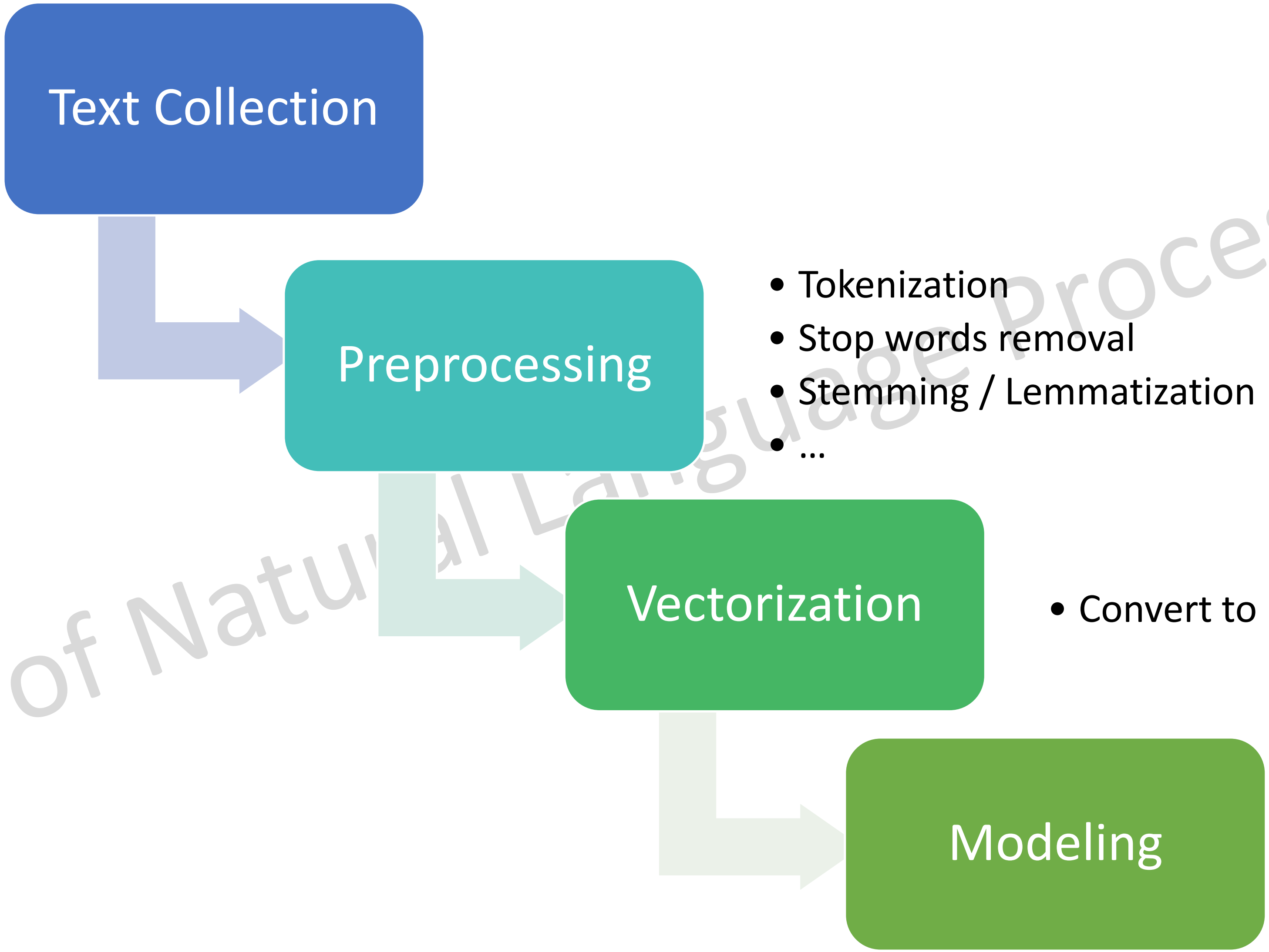
Session 2. Basics of Linguistics

- Components
- Challenges
- **Vectorization**

Session 3. Basics of ML

Session 4 (Lab). Effective Word Representation by python

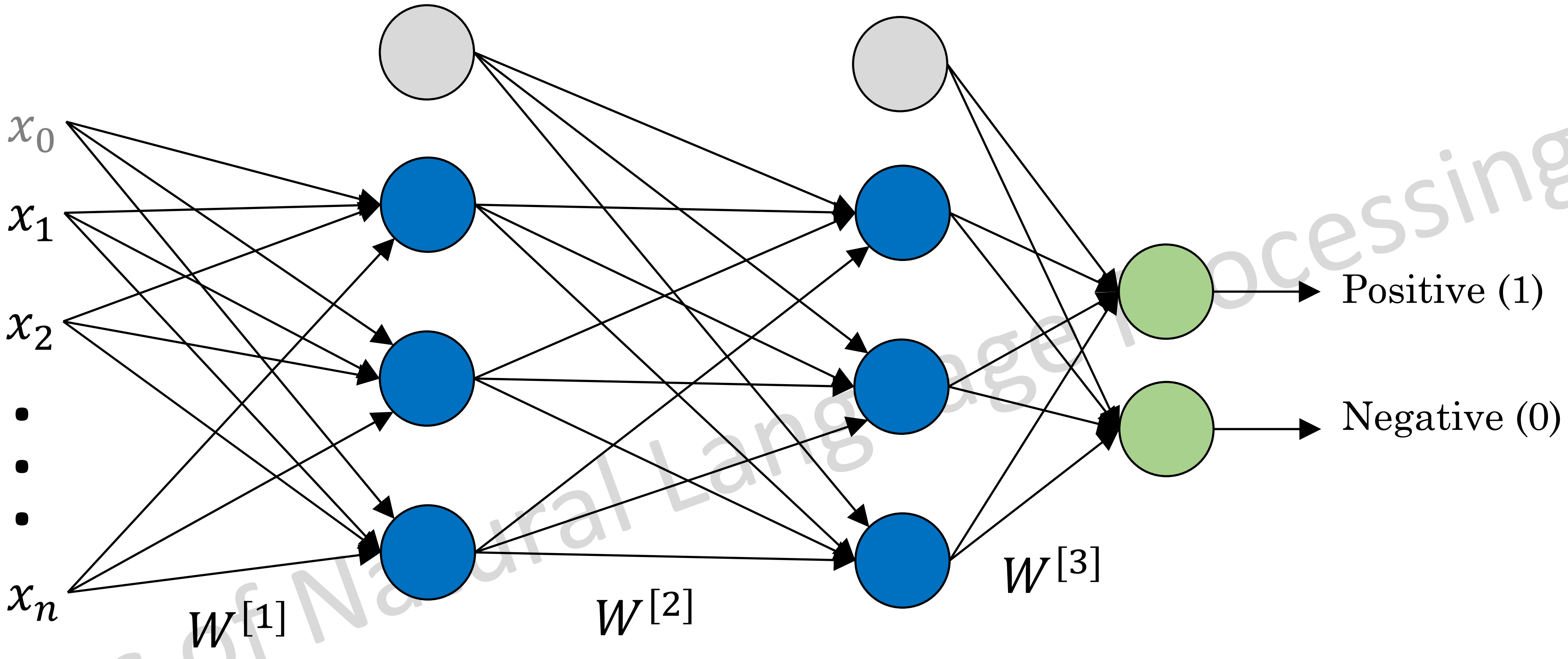
Steps of ML/DL Projects with Text Data



- Tokenization
- Stop words removal
- Stemming / Lemmatization
- …

- Convert to numbers

ANN Example for Sentiment Analysis



$$\boldsymbol{z}^{[i]} = \boldsymbol{W}^{[i]}\boldsymbol{a}^{[i-1]}$$

$$\boldsymbol{a}^{[i]} = f(\boldsymbol{z}^{[i]})$$

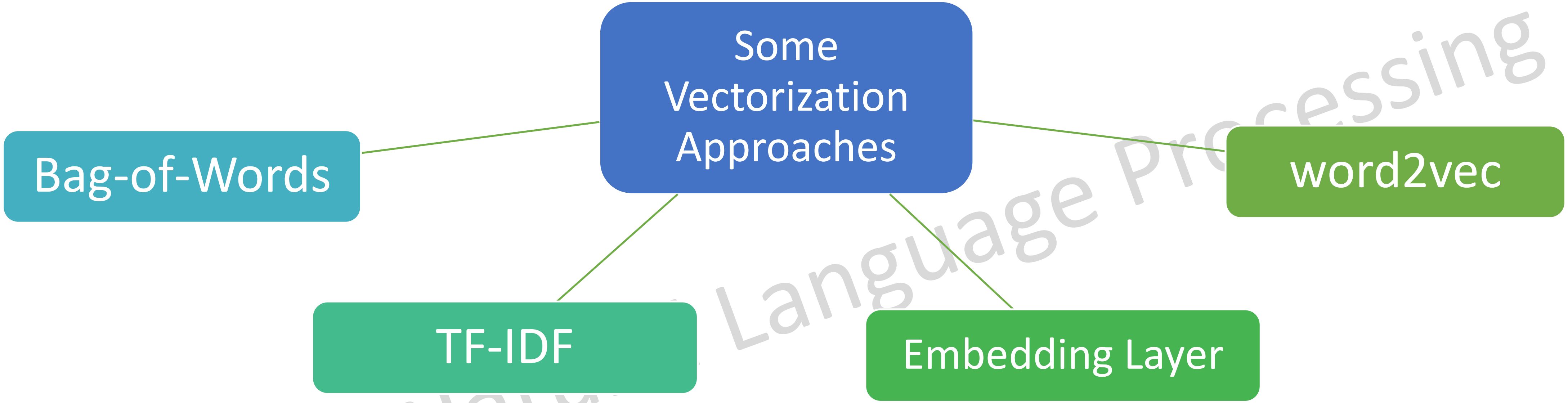
$\boldsymbol{x} = [x_1, x_2, \dots, x_n]$: tweets

Example: "This movie was almost good"

QUICK TIPS

Vectorization (converting words to numbers) is unavoidable for ML

Vectorization



Bag-of-Words (BoW)

S1: Jim and Pam traveled by bus.

S2: The train was late.

S3: The flight was full; Traveling by flight is expensive.

	and	bus	by	expensive	flight	full	is	jim	late	pam	the	train	travel	was
S1	1	1	1	0	0	0	0	1	0	1	0	0	1	0
S2	0	0	0	0	0	0	0	0	1	0	1	1	0	1
S3	0	0	1	1	2	1	1	0	0	0	1	0	1	1

Bag-of-Words (BoW)

Pros

- Simple
- Fast

Cons

- Sparse
- High dimension
- No distinction between rare and common words
- No word relation (context)

TF-IDF

Term Frequency	—	Inverse Document Frequency
Frequency of the word	×	Importance of the word

TF-IDF

Term Frequency — Inverse Document Frequency

Frequency of the word × Importance of the word

$$\frac{no.\ of\ times\ it\ appears}{total\ no.\ of\ words} \times \log_{10} \frac{total\ no.of\ docs}{no.of\ docs\ in\ which\ word\ appears}$$

TF-IDF

Term Frequency — Inverse Document Frequency

Frequency of the word × Importance of the word

$$\frac{no.\ of\ times\ it\ appears}{total\ no.\ of\ words} \times \log_{10} \frac{total\ no.\ of\ docs}{no.\ of\ docs\ in\ which\ word\ appears}$$

Examples:

TF("the") = 0.1

Q1: What does it mean?

Q2: If "the" appears in all 10 documents, what is IDF("the")?

TF-IDF

Term Frequency — Inverse Document Frequency

Frequency of the word × Importance of the word

$$\frac{no.\ of\ times\ it\ appears}{total\ no.\ of\ words} \times \log_{10} \frac{total\ no.\ of\ docs}{no.\ of\ docs\ in\ which\ word\ appears}$$

Examples:

TF(“the”) = 0.1

IDF(“the”) = 0

➔ TF-IDF(“the”) = 0

TF-IDF

Term Frequency — Inverse Document Frequency

Frequency of the word × Importance of the word

$$\frac{no.\ of\ times\ it\ appears}{total\ no.\ of\ words} \times \log_{10} \frac{total\ no.\ of\ docs}{no.\ of\ docs\ in\ which\ word\ appears}$$

Examples:

TF(“NLP”) = 0.1

Q: If “NLP” appears in only one document, what is IDF(“NLP”)?

TF-IDF

Term Frequency — Inverse Document Frequency

Frequency of the word × Importance of the word

$$\frac{no.\ of\ times\ it\ appears}{total\ no.\ of\ words} \times \log_{10} \frac{total\ no.of\ docs}{no.of\ docs\ in\ which\ word\ appears}$$

Examples:

TF(“NLP”) = 0.1

IDF(“NLP”) = 1

➔ TF-IDF(“NLP”) = 0.1

TF-IDF

D1: Jim and Pam traveled by bus.

D2: The train was late.

D3: The flight was full. Traveling by flight is expensive.

	and	bus	by	expensive	flight	full	is	jim	late	pam	the	train	travel	was
D1	0.4	0.4	0.3	0	0	0	0	0.4	0	0.4	0	0	0.3	0
D2	0	0	0	0	0	0	0	0	0.5	0	0.4	0.5	0	0.4
D3	0	0	0.2	0.3	0.6	0.3	0.3	0	0	0	0.2	0	0.2	0.2

TF-IDF vs. BoW

BoW

	and	bus	by	expensive	flight	full	is	jim	late	pam	the	train	travel	was
S1	1	1	1	0	0	0	0	1	0	1	0	0	1	0
S2	0	0	0	0	0	0	0	0	1	0	1	1	0	1
S3	0	0	1	1	2	1	1	0	0	0	1	0	1	1

TF-IDF

	and	bus	by	expensive	flight	full	is	jim	late	pam	the	train	travel	was
D1	0.4	0.4	0.3	0	0	0	0	0.4	0	0.4	0	0	0.3	0
D2	0	0	0	0	0	0	0	0	0.5	0	0.4	0.5	0	0.4
D3	0	0	0.2	0.3	0.6	0.3	0.3	0	0	0	0.2	0	0.2	0.2

TF-IDF

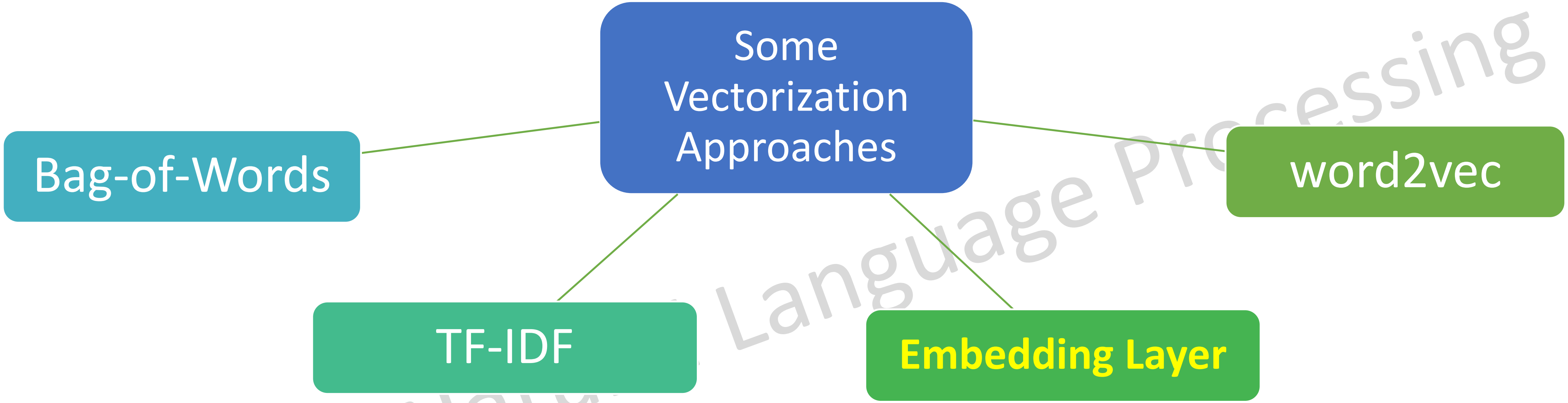
Pros

- Simple
- Fast
- Considers the importance of words

Cons

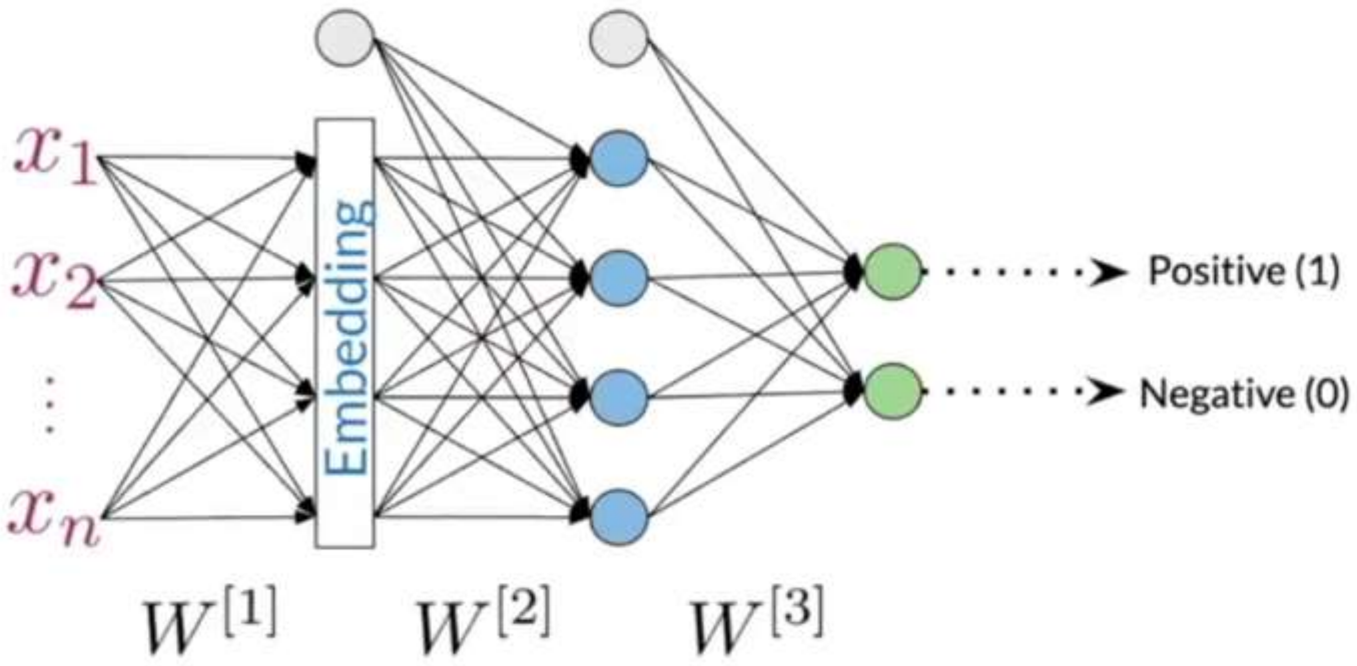
- Sparse
- High dimension
- No word relation (context)

Vectorization



Embedding Layer

Example: ANN for Sentiment Analysis



$\boldsymbol{x} = [x_1, x_2, \dots, x_n]$: tweets

Example: "This movie was almost good"

Embedding Layer

Input Representation

Word	Number
a	1
able	2
about	3
…	…
movie	680
…	…
zebra	1000

Embedding Layer

Input Representation

Word	Number
a	1
able	2
about	3
…	…
movie	680
…	…
zebra	1000

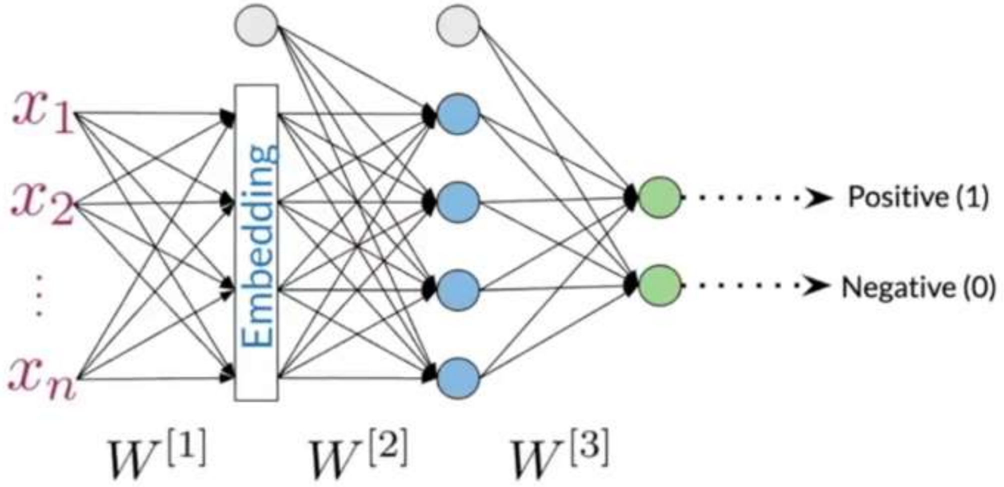
$\boldsymbol{x} = [x_1, x_2, \dots, x_n]$: tweets

Example: "This movie was almost good"

$$\boldsymbol{x} = [700, 680, 720, 20, 55]$$

In the ANN for sentiment analysis, how can we handle variable-length tweets?

Tweet 1: “this movie was almost good”
Tweet 2: “it was terrible”



Embedding Layer

Input Representation

Word	Number
a	1
able	2
about	3
…	…
movie	680
…	…
zebra	1000

$\boldsymbol{x} = [x_1, x_2, \dots, x_n]$: tweets

Example: "This movie was almost good"

$$\boldsymbol{x} = [700, 680, 720, 20, 55]$$

Zero-padding to match size of largest tweet

$$\boldsymbol{x} = [700, 680, 720, 20, 55, 0, 0, 0, 0, 0, 0, 0, 0, 0]$$

Embedding Layer

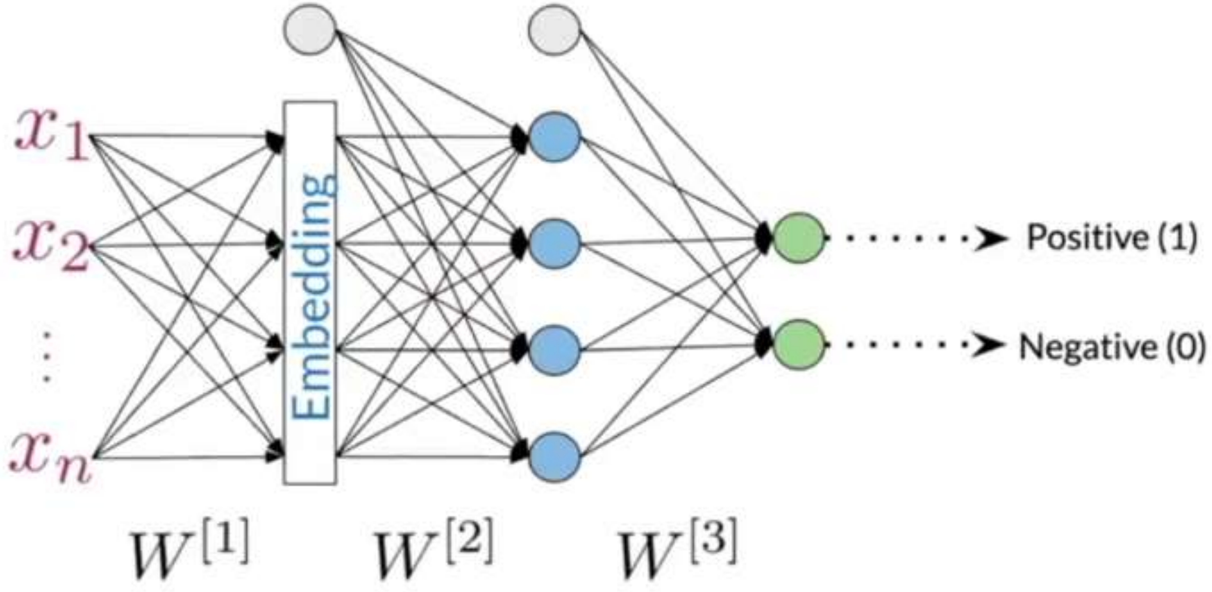
ANN learns the 4-D embedding representation of sentences

Example: “This movie was almost good”

$$\boldsymbol{x} = [700, 680, 720, 20, 55, 0, 0, 0, 0, 0, 0, 0, 0, 0]$$

$$\downarrow$$

$$\boldsymbol{a}^{[1]} = [0.4\ \ 0.01\ \ 0.3\ \ 0.21]$$



Embedding Layer

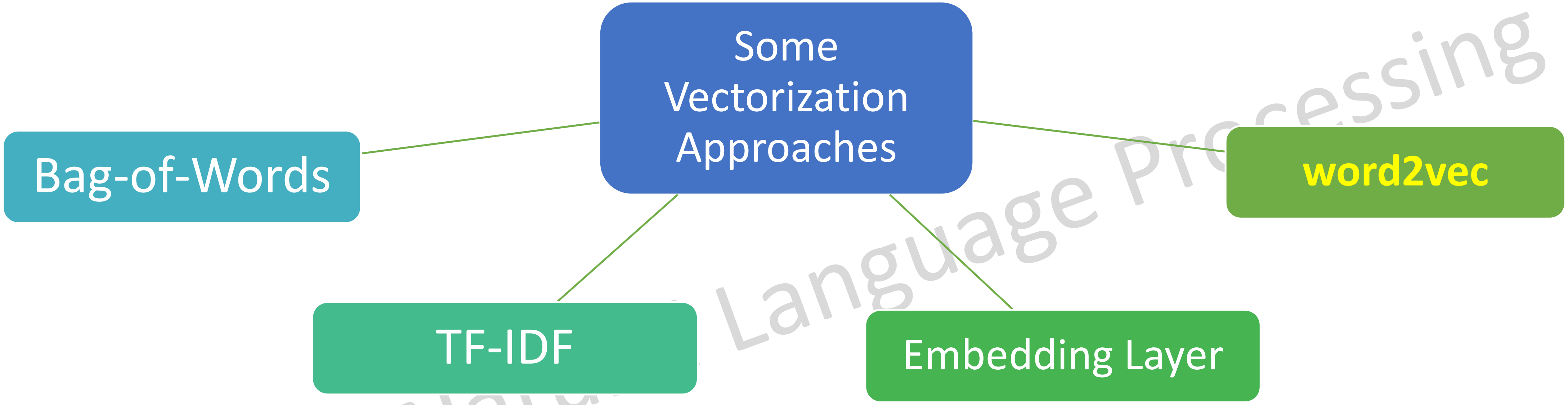
Pros

- Simple
- Fast
- Considers the importance of words
- Condense and low dimension

Cons

- No word relation
- Meaningless math operation between words

Vectorization



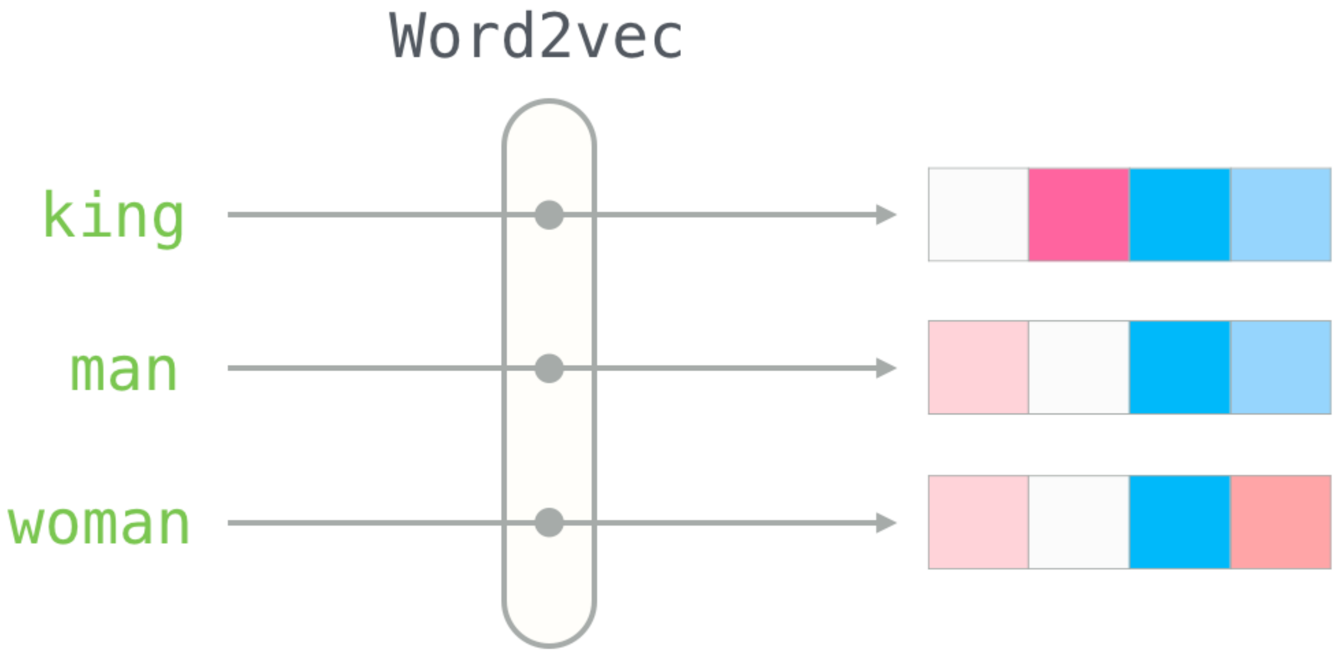
word2vec

Word2vec is a model provided by Google in 2013
for the **effective word embedding**.

[Mikolov, Tomas, et al. "Distributed representations of words and phrases and their compositionality." *Advances in neural information processing systems*. 2013]

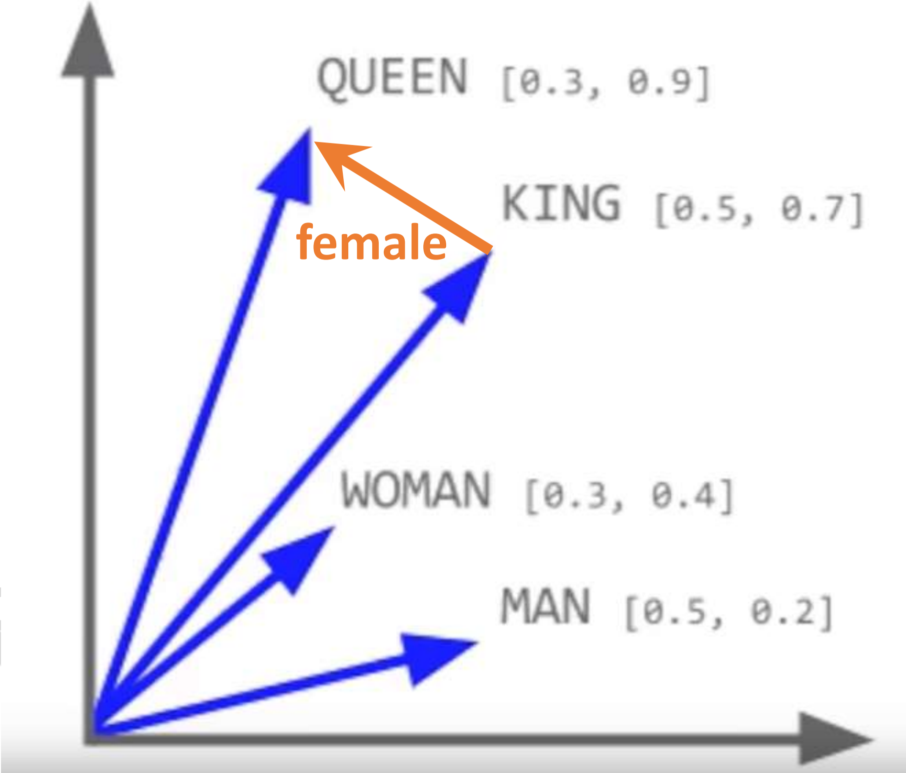
word2vec

Word2vec is a model provided by Google in 2013 for the **effective word embedding**.



[Mikolov, Tomas, et al. "Distributed representations of words and phrases and their compositionality." *Advances in neural information processing systems*. 2013]

Word Embedding

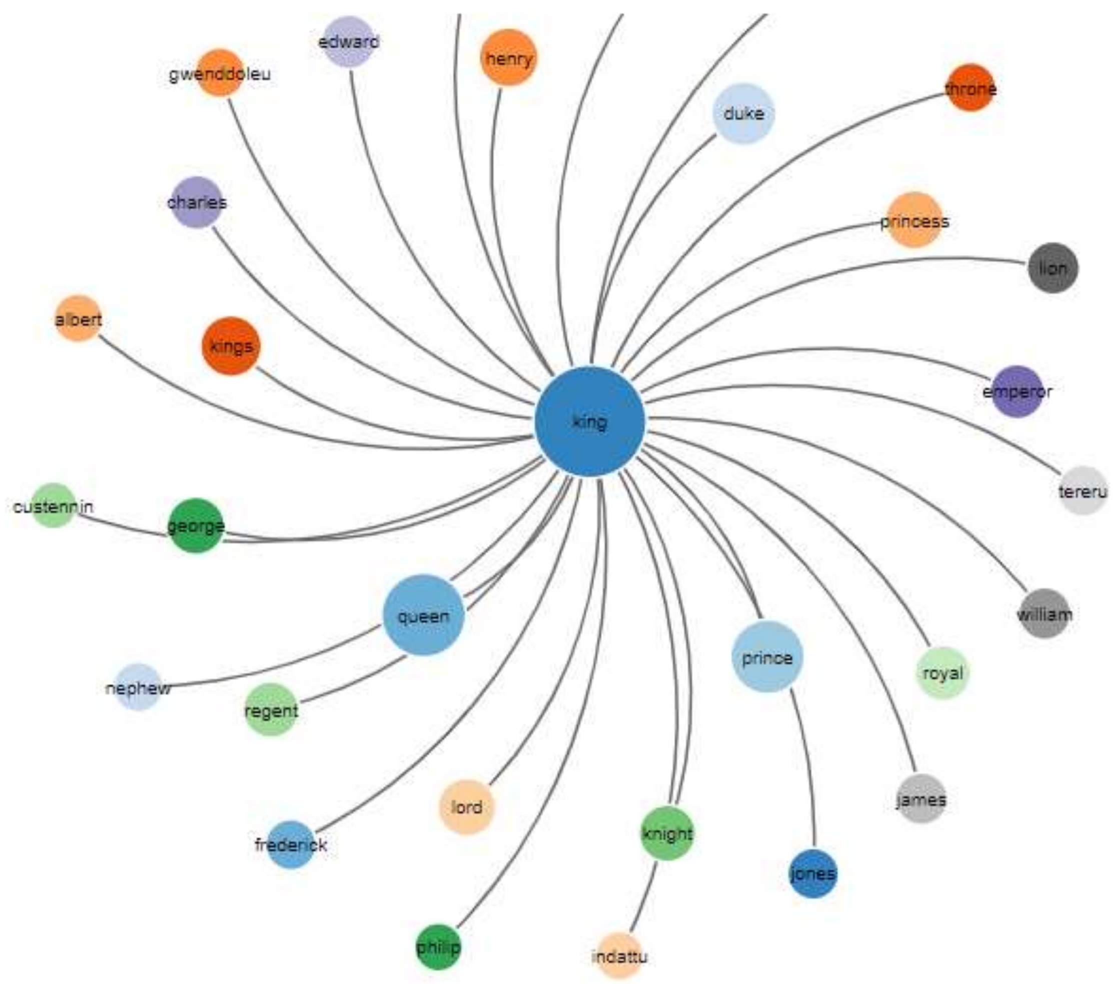


King + female = Queen
Man + female = Woman

Queen – royal = Woman

Word Similarity

Top 30 analogous words or synonyms for king



Top 10 similar words or synonyms for king

queen	0.745687
prince	0.659387
duke	0.574773
kings	0.544352
henry	0.532862
princess	0.524547
lord	0.518185
george	0.512252
knight	0.505063
regent	0.499793

https://wordsimilarity.com/

Vectorization: Wrap up

Some Vectorization Approaches

Bag-of-Words

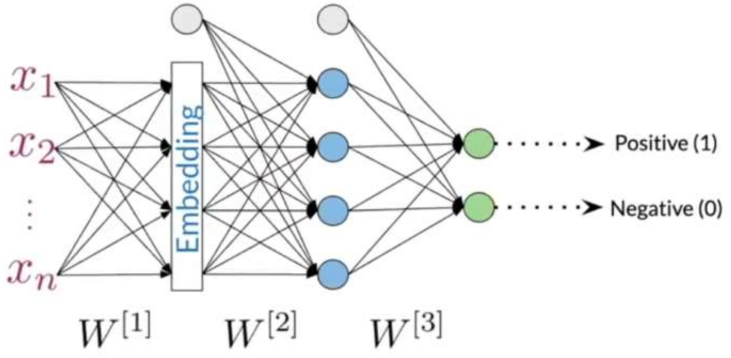
TF-IDF

Embedding Layer

word2vec

	and	bus	by	expensive	flight	full	is	jim	late	pam	the	train	travel	was
S1	1	1	1	0	0	0	0	1	0	1	0	0	1	0
S2	0	0	0	0	0	0	0	0	1	0	1	1	0	1
S3	0	0	1	1	2	1	1	0	0	0	1	0	1	1

	and	bus	by	expensive	flight	full	is	jim	late	pam	the	train	travel	was
D1	0.4	0.4	0.3	0	0	0	0	0.4	0	0.4	0	0	0.3	0
D2	0	0	0	0	0	0	0	0	0.5	0	0.4	0.5	0	0.4
D3	0	0	0.2	0.3	0.6	0.3	0.3	0	0	0	0.2	0	0.2	0.2



Wrap up

Session 1: Introduction

Session 2. Basics of Linguistics

- Components
- Challenges
- Vectorization

Session 3. Basics of ML

Session 4 (Lab). Effective Word Representation by python

Digikala Academy NLP Events

Basic

- **Session 1.** Introduction
- **Session 2.** Basics of Linguistics
- **Session 3.** Basics of ML
- **Session 4 (Lab).** Effective Word Representation by python

Intermediate

- TBA

Advanced

- TBA

References of this session

Natural Language Processing (NLP) with Python — Tutorial

https://medium.com/towards-artificial-intelligence/natural-language-processing-nlp-with-python-tutorial-for-beginners-1f54e610a1a0#2847