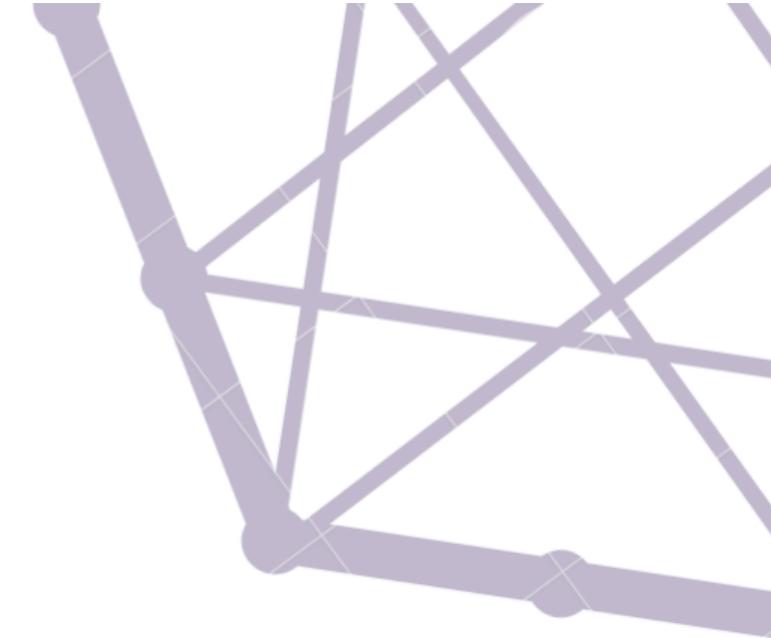


Natural Language Processing: **Text Generation**

:::::::::::::::::::::::::::

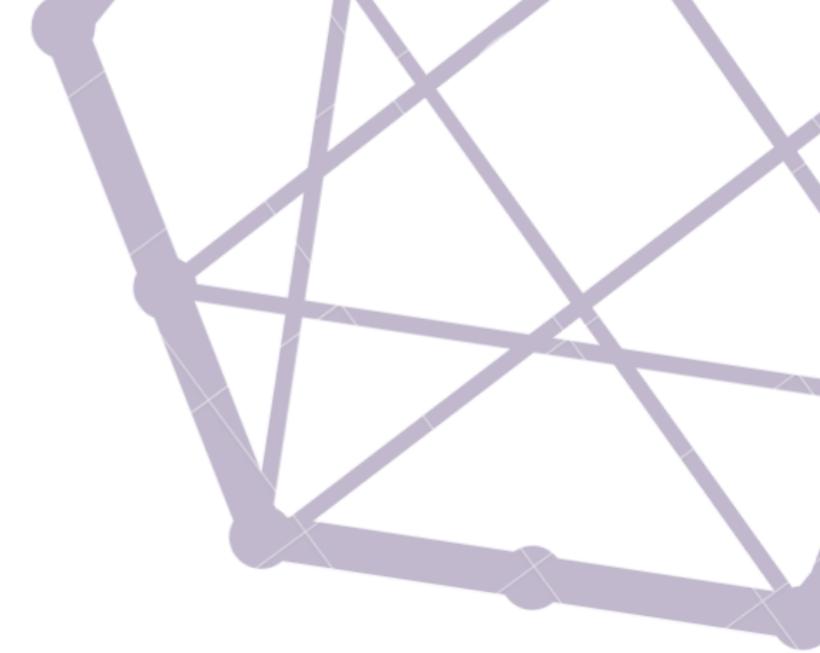
By Maram Fahad

Natural Language Processing



Natural language processing (NLP) is “an area of research and application that explores how computers can be used to understand and manipulate natural language text or speech” (Chowdhury 2003, pp. 51).

Natural Language Understanding



Natural language understanding is the process of teaching computers to comprehend the deeper contextual meanings of human language (Hirschberg and Manning, 2015).

Natural Language Processing

Analysing Techniques

- Tokenization
- Parsing
- Language detection
- Stemming
- Part-of-speech tagging
- Identification of semantic relationships (Akerkar, 2018)

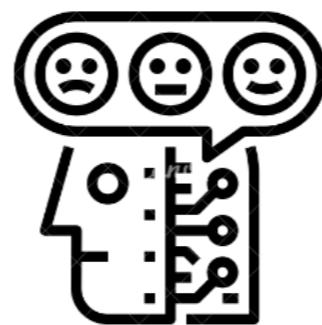
Natural Language Processing Applications



Chatbots



Sentiment Analysis



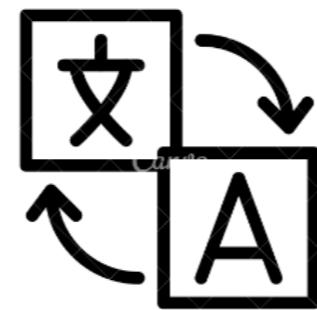
Spell Checking



Speech Recognition



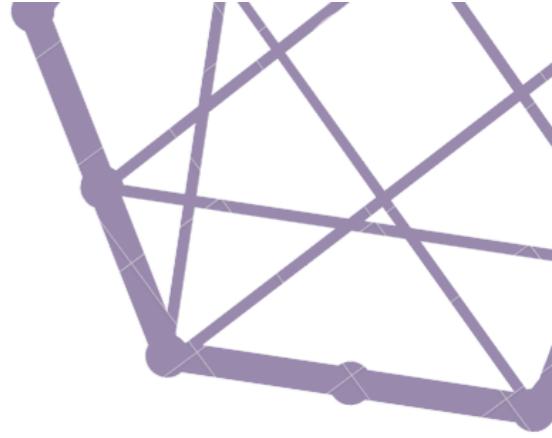
Machine Translation



Text Generation



Natural Language Processing: Text Generation



Text generation is the generation of natural language texts from structured data (Shao et al., 2019; Gatt and Krahmer, 2018).

It has a wide range of applications, including weather forecasts, news reports, etc.

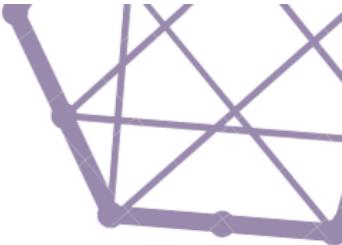
Markov Chain



It is named after the Russian mathematician Andrey Markov.

A Markov chain is a stochastic model that “describe[s] a sequence of possible events in which the probability of each event depends only on the state attained in the previous event” (Gagniuc, 2017; Oxford Dictionaries, 2017).

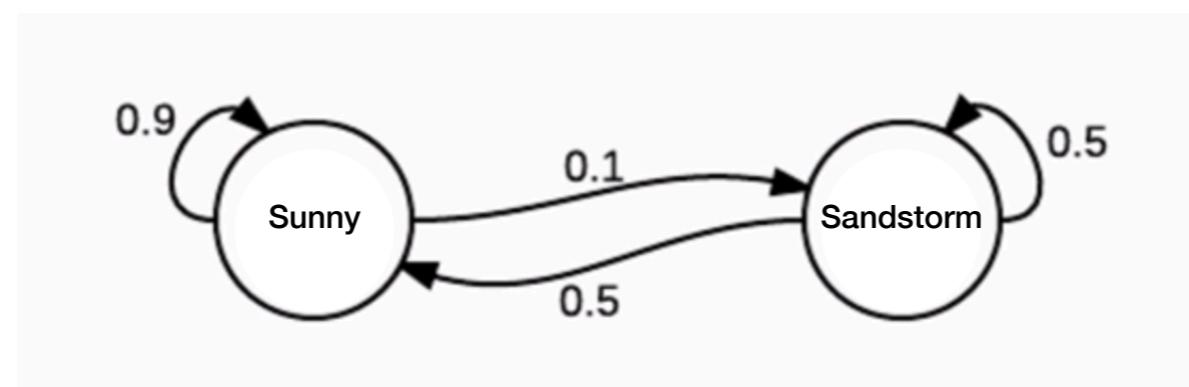
Markov Chain



Using Markov chains, tomorrow's weather is based on what happens today and nothing before that.

In a sunny state, there is a 10% chance that there will be a sandstorm tomorrow and a 90% chance that it will remain sunny.

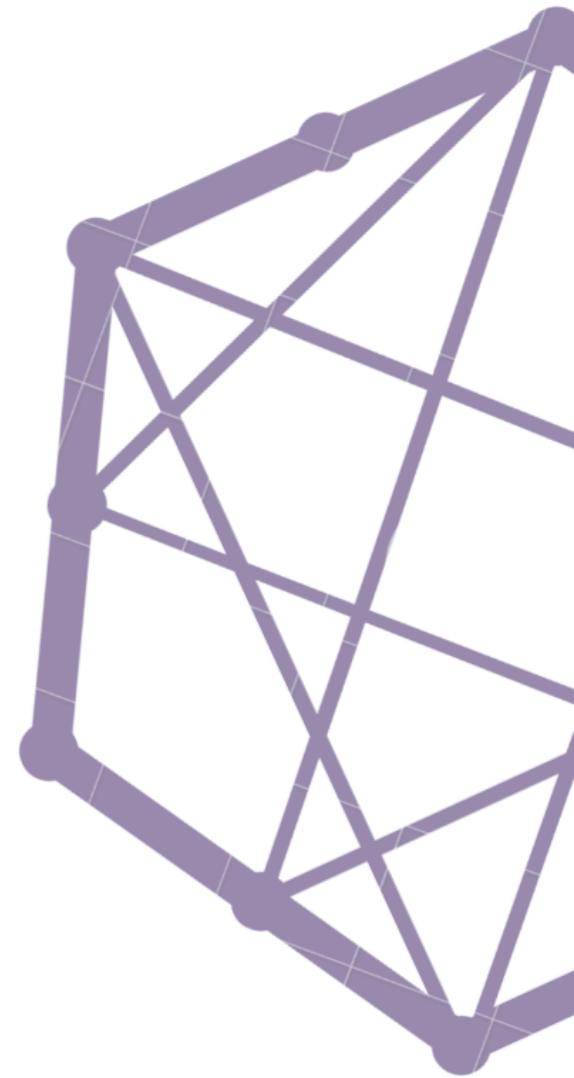
In a sandstorm state, there is a 50% chance that there will continue to be a sandstorm tomorrow and a 50% chance that it will remain sunny.



source: edited image from Zhao, 2017

Markov Chain

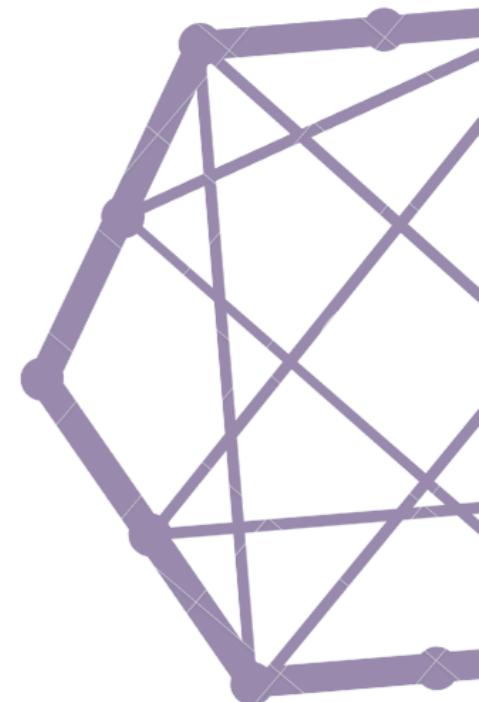
Text Generation



Markov Chain

Text Generation

- Markov chains create word dependencies without any knowledge of a language's syntax or semantics.
- Markov chains are created based on statistical knowledge that has been extracted from a corpus of information.



```
In [1]: import random
```

```
In [2]: corpus = [line.rstrip('\n') for line in open("Markov_Chain.txt")]
```

```
In [3]: corpus
```

```
Out[3]: ['Keep calm and read the lessons.',  
 'Keep calm and commit your labs.',  
 'Keep calm and practice coding in python.',  
 'Keep calm and work on lightning talk 2.',  
 'Keep calm and help your classmates.',  
 'Keep calm and work on your capstone.',  
 'Keep calm and review the lessons we sat through.',  
 'Keep calm and finish working on project 3.',  
 'Keep calm and discuss your presentation ideas.',  
 'Keep calm and submit your presentation.',  
 'Keep calm and ask for help when needed.',  
 'Keep calm and think about your lab answers.',  
 'Keep calm and enjoy the program.',  
 'Keep calm and read ahead.',  
 'Keep calm and enjoy the learning experience.',  
 'Keep calm and keep being awesome.',  
 'Keep calm and keep being motivated.',  
 'Keep calm and maintain connections with your classmates.']
```

```
In [4]: #Build a Markov Chain model
```

```
model = {}

for line in corpus:
    line = line.lower().split()
    for i, word in enumerate(line):
        if i == len(line)-1:
            model['END'] = model.get('END', []) + [word]
        else:
            if i == 0:
                model['START'] = model.get('START', []) + [word]
            model[word] = model.get(word, []) + [line[i+1]]
```

```
In [5]: print(model)
```

```
In [6]: # Use model to generate sayings
def generator():
    motivational_sayings = []

    while True:
        if not motivational_sayings:
            words = model['START']
        elif motivational_sayings[-1] in model['END']:
            break
        else:
            words = model[motivational_sayings[-1]]
        motivational_sayings.append(random.choice(words))
    return motivational_sayings
```

```
In [7]: sayingsList = []
```

```
In [8]: for i in range(3):
    sayingsList.append(" ".join(generator()))
```

```
In [9]: sayingsList
```

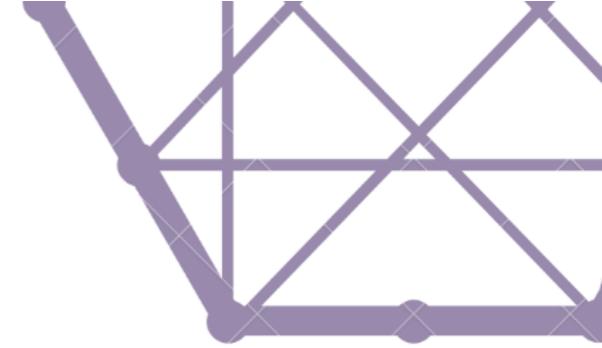
```
Out[9]: ['keep calm and work on your presentation ideas.',
         'keep calm and commit your presentation.',
         'keep calm and work on project 3. ']
```

```
In [10]: f=open('Motivational_Sayings.txt','a')
for ele in sayingsList:
    f.write(ele+'\n')

f.close()
```

- 1 keep calm and finish working on lightning talk 2.
- 2 keep calm and help your presentation.
- 3 keep calm and review the program.
- 4 keep calm and finish working on lightning talk 2.
- 5 keep calm and help your presentation.
- 6 keep calm and review the program.
- 7 keep calm and submit your classmates.
- 8 keep calm and enjoy the program.
- 9 keep calm and enjoy the program.
- 10 keep calm and read the lessons we sat through.
- 11 keep calm and keep calm and enjoy the learning experience.
- 12 keep calm and enjoy the program.

Markov Chain Text Generation Project



Objective: To lift the spirits of the General Assembly community, specifically the data science immersive group.

Method: Post motivational quotes on Slack to keep morale high.

Oh, no!

Human



Name: Maram

Bio: wasting precious time writing
Evil Robot's bio

VS

Machine



Name: Evil Robot

Bio: workforce domination and
automating Maram's motivational
process

Human vs. Machine

Is it or is it not a human-generated motivational saying? React with the (🤖) emoji if you think it is a machine or the (👉) emoji if you think it was written by a human.

Keep calm and review the lessons we sat through.



Keep calm and work on the learning experience.



Keep calm and enjoy coding in python.



Keep calm and help your classmates.



Keep calm and work on your capstone.



Keep calm and think about being awesome.



Keep calm and review the lessons.



Keep calm and finish working on project 3.



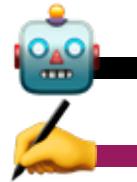
Keep calm and keep being motivated.



keep calm and commit your classmates.



Human Wins



Keep calm and review the lessons we sat through.

✍ 8 🤖 14 THANK YOU! 1



Keep calm and work on the learning experience.

? 1 THANK YOU! 3 ✍ 4 🤖 14



Keep calm and enjoy coding in python.

✍ 7 🤖 5 THANK YOU! 1



Keep calm and help your classmates.

✍ 7 🤖 2 THANK YOU! 1



Keep calm and work on your capstone.

✍ 11 THANK YOU! 1



Keep calm and think about being awesome.

🤖 8 ✓ 1 THANK YOU! 1



Keep calm and review the lessons.

✍ 7 THANK YOU! 1



Keep calm and finish working on project 3.

✍ 8 🤖 4 THANK YOU! 1



Keep calm and keep being motivated.

🤔 3 🤖 5 ✍ 2 THANK YOU! 1



keep calm and commit your classmates.

🤖 10 ⚡ 1 ✍ 3 THANK YOU! 1

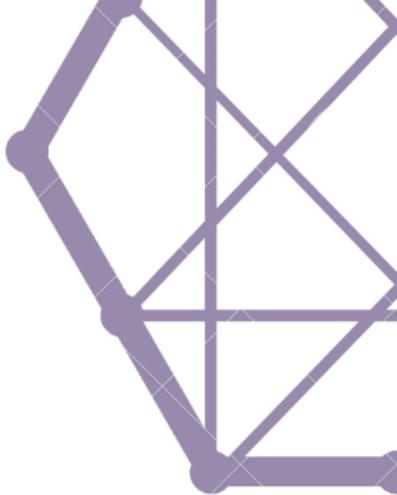


Disclaimer: the last motivational saying was generated by Evil Robot (see image below)

sayingsList

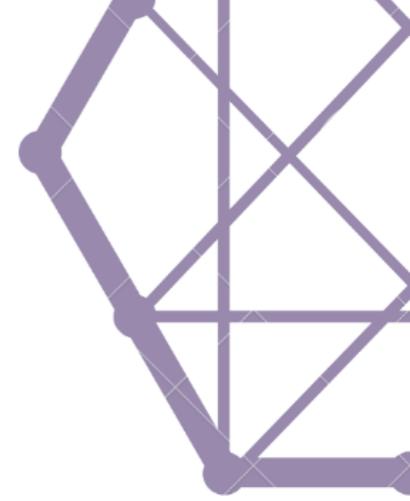
```
[ 'keep calm and practice coding in python.',  
  'keep being motivated.',  
  'keep being awesome.',  
  'keep calm and practice coding in python.',  
  'keep calm and commit your classmates.',
```

Natural Language Processing **Conclusion**



NLP plays a significant role in advancing human-machine interactions. With more research in this field, there is a significant chance of a breakthrough that will allow machines to recognize and understand human language, and perhaps, automate the workforce (Dun Dun Duuuuun!).

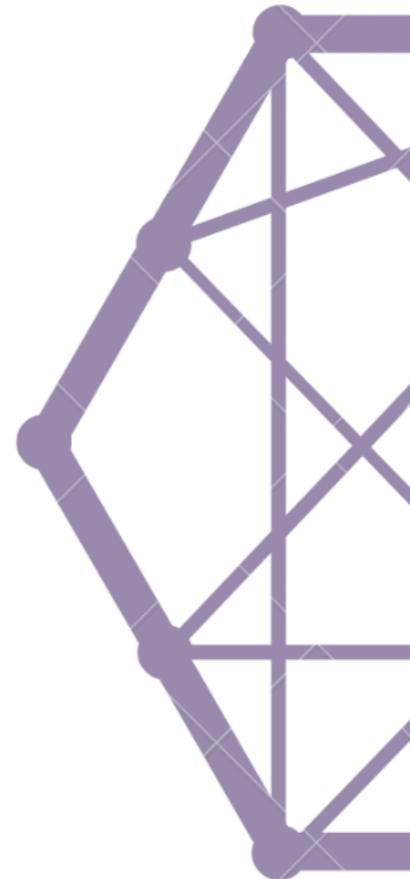
Natural Language Processing References



- Akerkar, R., 2018. Artificial intelligence for business. Springer.
- Chowdhury, G.G., 2003. Natural language processing. Annual review of information science and technology, 37(1), pp.51-89.
- Gagniuc, P.A., 2017. Markov chains: from theory to implementation and experimentation. John Wiley & Sons.
- Gatt, A. and Krahmer, E., 2018. Survey of the state of the art in natural language generation: Core tasks, applications and evaluation. J. Artif. Intell. Res., 61, pp. 65–170.
- Hirschberg, J. and Manning, C.D., 2015. Advances in natural language processing. Science, 349(6245), pp.261-266.
- Oxford Dictionaries, 2017. Markov chain | Definition of Markov chain in US English. Oxford Dictionaries | English. Accessed March 10, 2020: (https://www.lexico.com/en/definition/markov_chain)
- Ratnaparkhi, A., 2006. International Business Machines Corp. Trainable dynamic phrase reordering for natural language generation in conversational systems. U.S. Patent 7,043,420. Song, B. and Suh, Y., 2019. Narrative texts-based anomaly detection using accident report documents: The case of chemical process safety. Journal of Loss Prevention in the Process Industries, 57, pp.47-54.
- Shao, Z., Huang, M., Wen, J., Xu, W. and Zhu, X., 2019. Long and Diverse Text Generation with Planning-based Hierarchical Variational Model. pp.1-16
- Pati, B., Panigrahi, C.R., Buyya, R. and Li, K.C., 2018. Advanced Computing and Intelligent Engineering. Proceedings of ICACIE, 1.

**Natural
Language
Processing
Inspired by**

- Andrew's Adventures in Technology, 2020. Text generation using Markov chains. Accessed March 10, 2020: (<http://awalsh128.blogspot.com/2013/01/text-generation-using-markov-chains.html>).
- Alami, R., 2018. How I generated inspirational quotes with less than 20 lines of python code. Accessed March 10, 2020: (<https://codeburst.io/how-i-generated-inspirational-quotes-with-less-than-20-lines-of-code-38273623c905>).
- Zhao, A., 2017. Comparing Stand up Comedians Using Natural Language Processing . Accessed March 10, 2020: (<https://github.com/adashofdata/nlp-in-python-tutorial>).





Thank you for
listening.

شكرا على收听