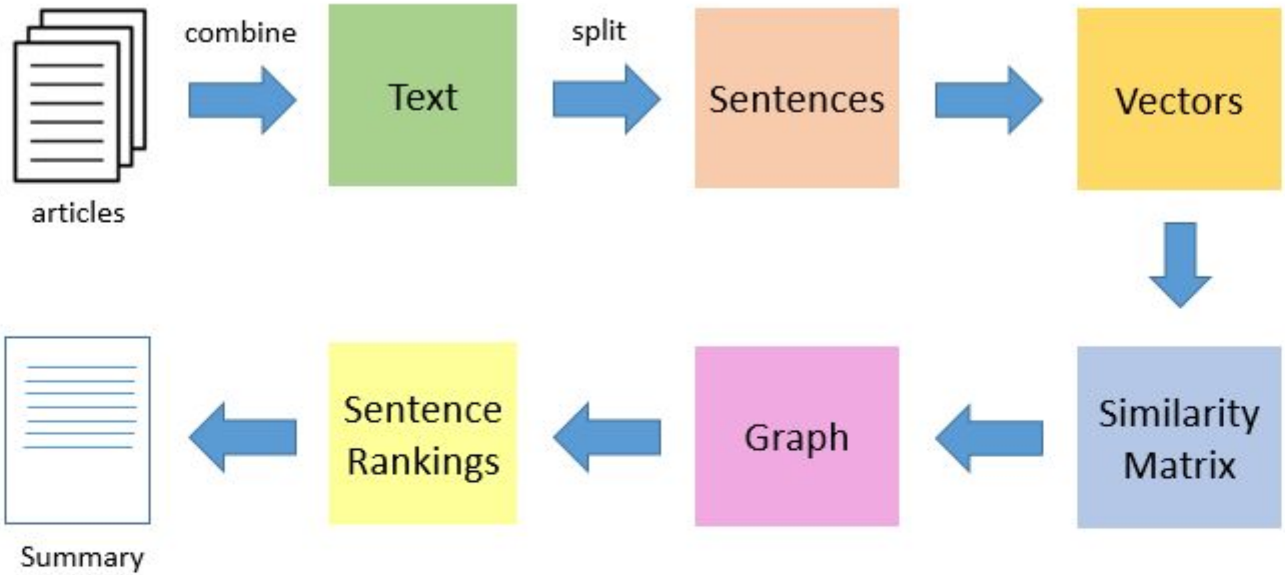
Text Summarization

Luís Filipe da Costa Cunha
lfilipecc1@gmail.com

José João Almeida
jj@di.uminho.pt

Text Summarization

- Extraction-based techniques
    - “Extract some key subset of the content from the original document such that this subset of content contains the core information and acts as the focal point of the entire document. “
- Abstraction-based techniques
    - “Leverage language semantics to create representations and use natural language generation (NLG) techniques where the machine uses knowledge bases and semantic representations to generate text on its own and create summaries just like a human would write them.”


combine
split
articles
Text
Sentences
Vectors
Similarity Matrix
Graph
Sentence Rankings
Summary

Text Processing

- Stop Words
- Lowercase
- Punctuation
- Lemmatization
- Numbers
- ...

```
import nltk
nltk.sent_tokenize(DOC)
nltk.word_tokenize(sentence)
stop_words = nltk.corpus.stopwords.words('portuguese')

import stanza, spacy
```

Creating Normalized Sentences

```
The Pokémon anime series was largely credited for allowing anime to become more popular and familiar
around the world, especially in the United States, where the two highest grossing anime films are
both Pokémon films.
```

```
pokémon anime series largely credited allowing anime become popular familiar around world especially
united states two highest grossing anime films pokémon films
```

Vocab Numeric Representation

- Token Frequency
- word2Vec
- Doc2Vec
- …

```
from gensim.models import KeyedVectors
#KeyedVectors.load_word2vec_format('models/cbow_s50.txt', binary=False)
word_embeddings = KeyedVectors.load("models/glove-wiki-gigaword-100")
```

```
word_embeddings["pokémon"]

array([ 1.5529e-01, -2.2740e-01,  1.2560e-01, -9.1718e-01, -8.8370e-02,
        5.0716e-01,  5.0123e-01, -5.9726e-01, -3.3003e-04, -5.1133e-01,
       -1.5189e-01, -8.5978e-02,  4.6707e-01, -5.5535e-01, -5.8030e-01,
        2.4302e-01,  6.9896e-01,  2.5054e-01,  1.0198e+00, -3.9722e-03,
        4.6517e-01, -4.8657e-01, -7.2978e-01,  2.1680e-01,  1.4307e+00, …])
```

Sentence Encoding

```
sentence_vectors = []
for frase in norm_sentences:
frase_embedding = [word_embeddings[palavra] if palavra in word_embeddings.key_to_index
                                           else np.zeros((dim,))
                                           for palavra in frase.split()]
   vec = sum(frase_embedding)/(len(frase.split()) + 0.001)
   sentence_vectors.append(vec)
```

```
x = np.array([[1,2,3,4],[4,3,2,1]])
print(sum(x))
print(sum(x)/5)

[5 5 5 5]
[1. 1. 1. 1.]
```

Cosine Similarity

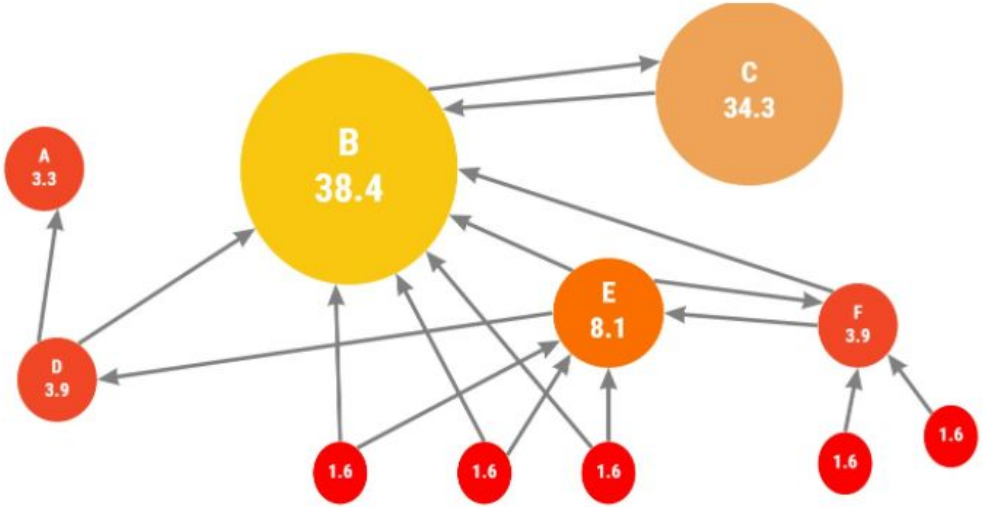
```
from sklearn.metrics.pairwise import cosine_similarity
len_sentences = len(norm_sentences)
sim_mat = np.zeros([len_sentences, len_sentences])
for i in range(len_sentences):
   for j in range(len_sentences):
       if i != j:
               x = sim_mat[i][j] = cosine_similarity(sentence_vectors[i].reshape(1,dim),
                                  sentence_vectors[j].reshape(1,dim))[0,0]
sim_mat = np.round(sim_mat,3)
```

PageRank

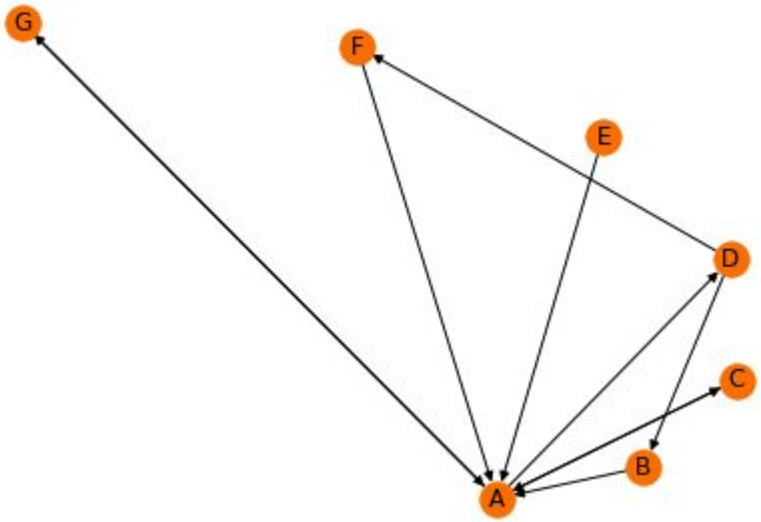
Google interprets a link from page A to page B as a vote from page A to page B. Incoming links can be interpreted as votes.

It takes into consideration the “importance” of the page that is “giving” out the vote.

Page’s importance is equal to the sum of the votes of its incoming links.

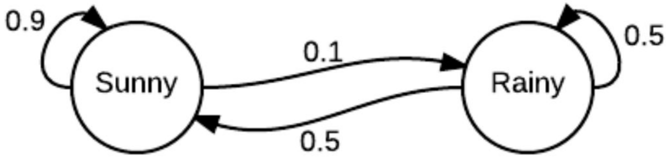


PageRank



Page rank value:
'A': 0.408,
'C': 0.137,
'G': 0.137
'D': 0.137,
'B': 0.079,
'F': 0.079,
'E': 0.021,

Markov Chain



Transition matrix

$$P = \begin{bmatrix} 0.9 & 0.1 \\ 0.5 & 0.5 \end{bmatrix}$$

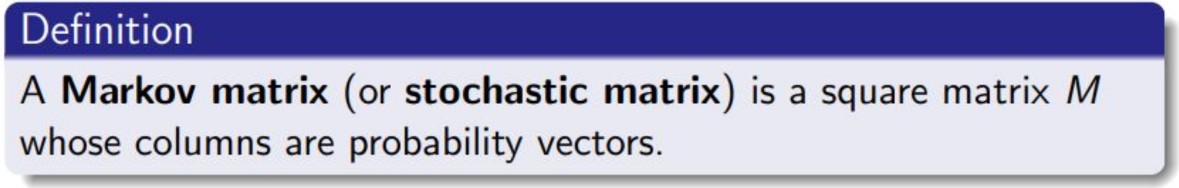
$$\mathbf{x}^{(0)} = [1 \quad 0]$$

$$\mathbf{x}^{(1)} = \mathbf{x}^{(0)} P = [1 \quad 0] \begin{bmatrix} 0.9 & 0.1 \\ 0.5 & 0.5 \end{bmatrix} = [0.9 \quad 0.1]$$

$$\mathbf{x}^{(2)} = \mathbf{x}^{(1)} P = [0.9 \quad 0.1] \begin{bmatrix} 0.9 & 0.1 \\ 0.5 & 0.5 \end{bmatrix} = [0.86 \quad 0.14]$$

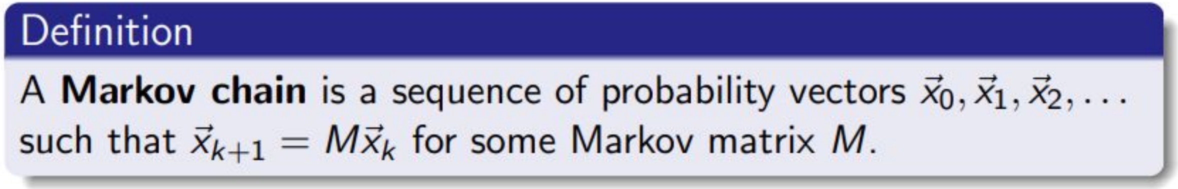
$$\mathbf{x}^{(n)} = \mathbf{x}^{(0)} P^n$$

Markov Chain



Definition

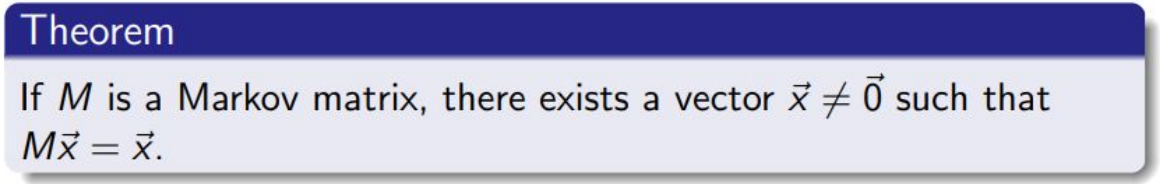
A **Markov matrix** (or **stochastic matrix**) is a square matrix M whose columns are probability vectors.



Definition

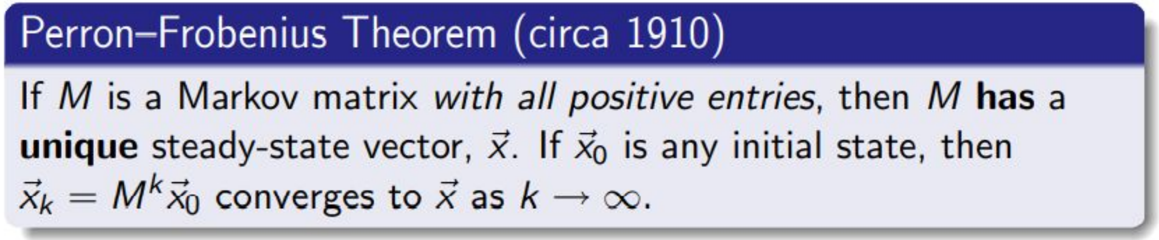
A **Markov chain** is a sequence of probability vectors $\vec{x}_0, \vec{x}_1, \vec{x}_2, \ldots$ such that $\vec{x}_{k+1} = M\vec{x}_k$ for some Markov matrix M.

Markov Chain



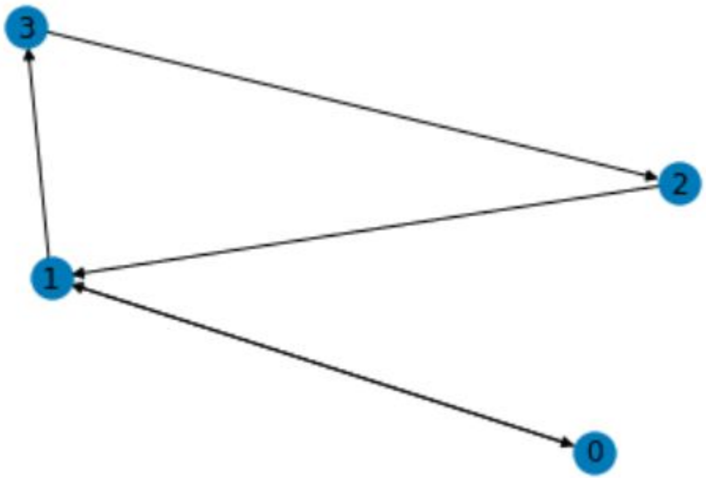
Theorem

If M is a Markov matrix, there exists a vector $\vec{x} \neq \vec{0}$ such that $M\vec{x} = \vec{x}$.



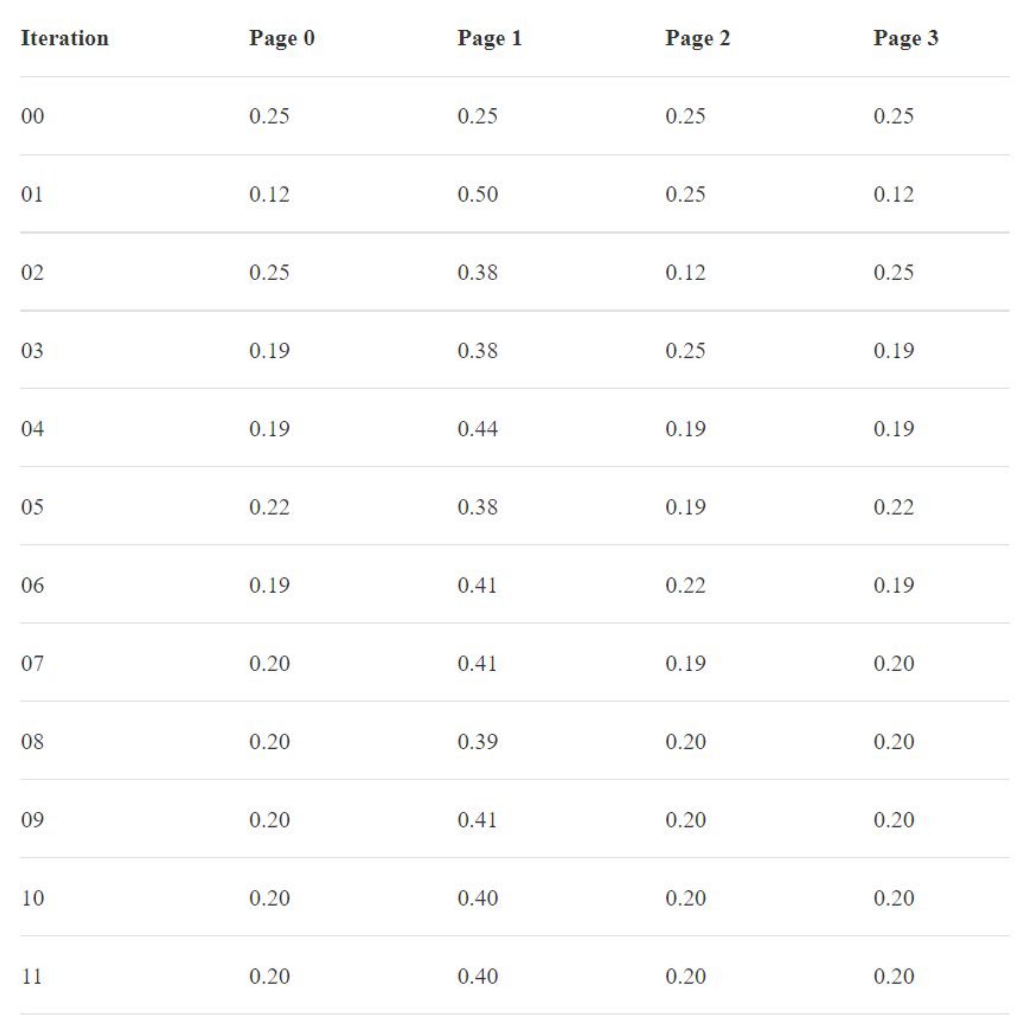
Perron–Frobenius Theorem (circa 1910)

If M is a Markov matrix *with all positive entries*, then M **has** a **unique** steady-state vector, $\vec{x}$. If $\vec{x}_0$ is any initial state, then $\vec{x}_k = M^k \vec{x}_0$ converges to $\vec{x}$ as $k \to \infty$.



$$A = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 \end{pmatrix}$$

$$\begin{pmatrix} 0 & 1/2 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 1/2 & 0 & 0 \end{pmatrix}$$



Iteration	Page 0	Page 1	Page 2	Page 3
00	0.25	0.25	0.25	0.25
01	0.12	0.50	0.25	0.12
02	0.25	0.38	0.12	0.25
03	0.19	0.38	0.25	0.19
04	0.19	0.44	0.19	0.19
05	0.22	0.38	0.19	0.22
06	0.19	0.41	0.22	0.19
07	0.20	0.41	0.19	0.20
08	0.20	0.39	0.20	0.20
09	0.20	0.41	0.20	0.20
10	0.20	0.40	0.20	0.20
11	0.20	0.40	0.20	0.20

“It is equivalent to calculating the eigenvector corresponding to the eigenvalue 1”

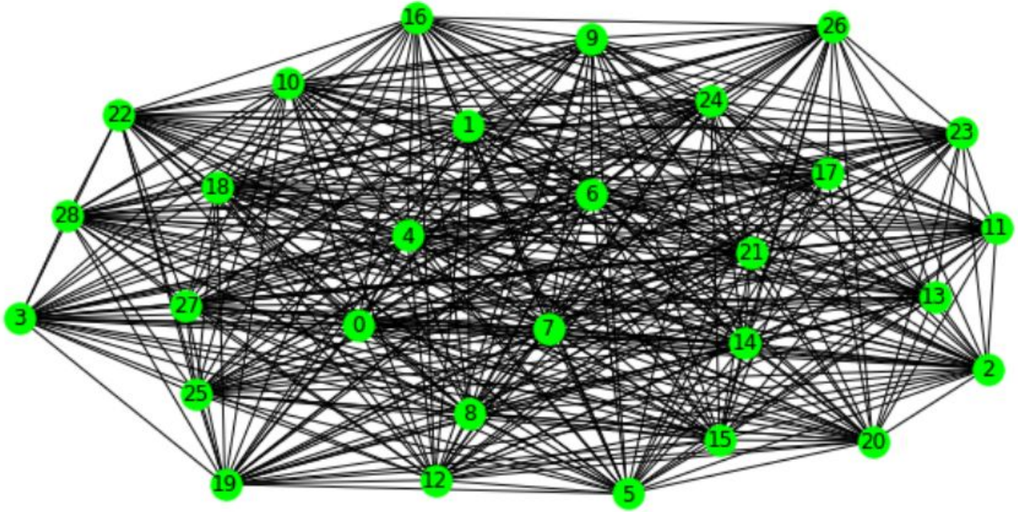
$$v = Mv$$

$$\mathrm{A}\nu = \lambda\nu$$

“it finds how similar each sentence is to all other sentences in the text. The most important sentence is the one that is most similar to all the others.”

Get Sentence Importance Scores

```
import networkx
similarity_graph = networkx.from_numpy_array(sim_mat)



import matplotlib.pyplot as plt

plt.figure(figsize=(12, 6))
networkx.draw_networkx(similarity_graph,
                       node_color='lime')

scores = networkx.pagerank(similarity_graph)
{index: similarity}

{0: 0.0357784643575689, 1: 0.03579266319732,
 2: 0.034600532999727665, ….}
```



Result...

The Pokémon anime series was largely credited for allowing anime to become more popular and familiar around the world, especially in the United States, where the two highest-grossing anime films are both Pokémon films.

It is also considered to be one of the first anime series on television to reach this level of mainstream success with Western audiences, as well as being credited with allowing the game series to reach such a degree of popularity and vice versa.

The series, originally produced for the company’s Game Boy line of handheld consoles, was introduced in 1998 to the United States with two titles, known to fans as Red and Blue.

Pokémon became one of the most successful video game franchises in the world, second only to Nintendo’s Super Mario Bros.

The original Pokémon is a role-playing game based around building a small team of monsters to battle other monsters in a quest to become the best.

Result...

A Universidade do Minho (abreviado como UMinho ou UM) é uma instituição pública de ensino superior fundada em 1973 na cidade de Braga (Portugal), integrando-se no grupo das "Novas Universidades" (que alteraram o panorama do ensino superior no país), iniciando as suas actividades académicas em 1975/76.

A Universidade passa a reger-se pelo direito privado, nomeadamente no que respeita à sua gestão financeira, patrimonial e do pessoal.

O governo determinou ainda que ao fim de um "período experimental de cinco anos" será feita uma "avaliação independente" da aplicação do regime fundacional mesmo.

Em consequência desta avaliação, o Conselho Geral da Universidade "pode propor, justificadamente, o regresso da instituição ao regime não fundacional".

[3] Na edição de 2017 do Ranking de Xangai, a instituição ficou classificada no intervalo [401-500], sendo a quinta universidade portuguesa naquela classificação, após a Universidade de Lisboa, a Universidade do Porto, a Universidade de Aveiro e a Universidade de Coimbra.

https://we.tl/t-dWcUx3KsMi

Text Summarization

Luís Filipe da Costa Cunha
lfilipecc1@gmail.com

José João Almeida
jj@di.uminho.pt