How Does Textrank Work?

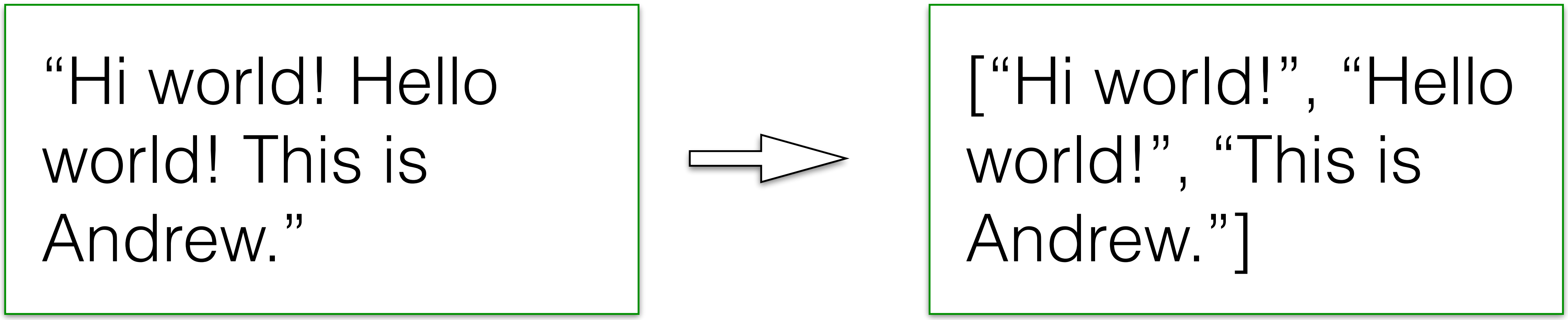
Andrew Koo - Insight Data Science

Textrank

- Separate the text into sentences based on a trained model
- Build a sparse matrix of words and the count it appears in each sentence
- Normalize each word with tf-idf
- Construct the similarity matrix between sentences
- Use Pagerank to score the sentences in graph

1. Separate the Text into Sentences

- Apply PunktSentenceTokenizer from the Python NLTK Library



“Hi world! Hello world! This is Andrew.”

[“Hi world!”, “Hello world!”, “This is Andrew.”]

2. Build a sparse matrix of words and the count it appears in each sentence

[“Hi world!”, “Hello world!”, “This is Andrew.”]

→

(Sen , word)	Count
(0 , 2)	1
(0 , 5)	1
(1 , 5)	1
(1 , 1)	1
(2 , 4)	1
(2 , 3)	1
(2 , 0)	1

3. Normalize each word with tf-idf

- **tf: term frequency** - how frequent a term occurs in a document
- **idf: inverse doc frequency** - how important a word is (weigh down the frequent terms, ex: is, does, how)

(Sen , word)	Count
(0 , 2)	1
(0 , 5)	1
(1 , 5)	1
(1 , 1)	1
(2 , 4)	1
(2 , 3)	1
(2 , 0)	1

→

(Sen , word)	Count
(0 , 2)	0.796
(0 , 5)	0.605
(1 , 5)	0.605
(1 , 1)	0.796
(2 , 4)	0.577
(2 , 3)	0.577
(2 , 0)	0.577

4. Construct the similarity matrix between sentences

(Sen , word)	Count
(0 , 2)	0.796
(0 , 5)	0.605
(1 , 5)	0.605
(1 , 1)	0.796
(2 , 4)	0.577
(2 , 3)	0.577
(2 , 0)	0.577

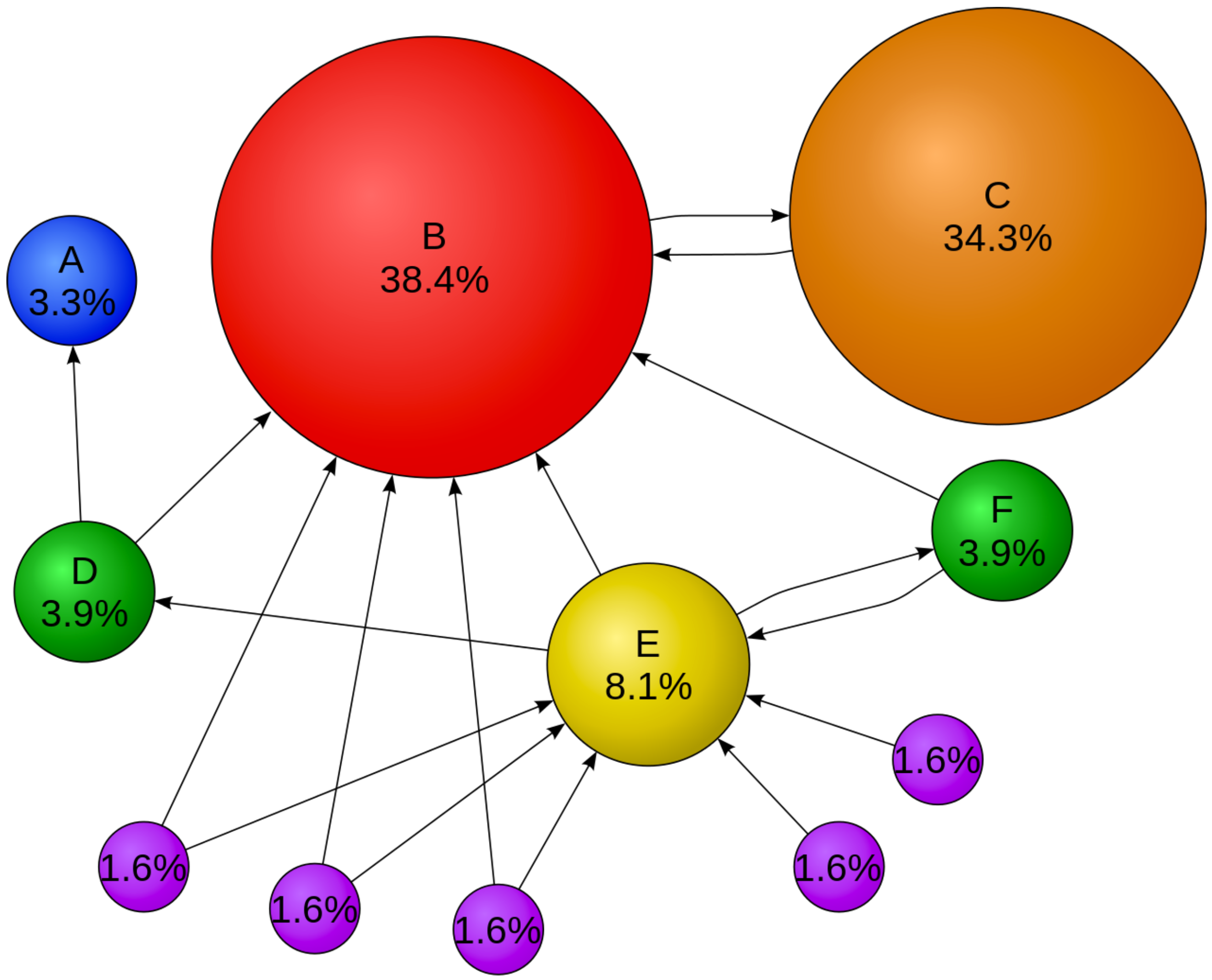
matrix * matrix.T

→

1	0.366	0
0.366	1	0
0	0	1

similarity matrix

5. Use Pagerank to score the sentences in graph



- Rank the sentences with underlying assumption that “summary sentences” are similar to most other sentences