NLP. Text summarization.

There are two approaches to automatic summarization nowadays: extraction and abstraction. Abstraction method tries to generate a summary based on the text. This summary could have words which aren't present in the text itself. This method looks very promising, but currently it is considered to be too complex. As a result extraction methods are more commonly used. They work by selecting certain words or sentences from the text and creating summary using them.

Usually unsupervised approaches are used, as they don't require training data, so that they can summarize a given text without additional information. And their quality is good enough.

In [1]:
```
from nltk import FreqDist
from nltk.tokenize import word_tokenize, sent_tokenize
from nltk.stem.wordnet import WordNetLemmatizer
lemma = WordNetLemmatizer()
from nltk.corpus import stopwords
stop = stopwords.words('english')

from bs4 import BeautifulSoup
from urllib.request import urlopen

from gensim.models import Phrases
from gensim.models.phrases import Phraser

import os
from collections import Counter
import string
punctuations = list(string.punctuation)
#Add some more punctuation, as the list doesn't cover all cases.
punctuations.extend(['”', '—', '``', "''"])
stop = stop + punctuations
```

```
D:\Programs\Anaconda3\lib\site-packages\gensim\utils.py:855: UserWarning: de
tected Windows; aliasing chunkize to chunkize_serial
  warnings.warn("detected Windows; aliasing chunkize to chunkize_serial")
```

The basic idea behind unsupervised summarization is the following:

- split text into sentences;
- tokenize sentences into separate words;
- assign scores to sentences based on importance;
- select several top sentences and display them in original order;

The main point, obviously, is assigning scores to sentences. Here are some of the ways to do this:

- calculate similarity between each pair of sentences and select sentences which are most similar to most sentences;
- calculate word frequences, select most frequent words and select sentences which have most of these words;

In this notebook I'll use the following news article:

In [2]:
```
url = urlopen('http://news.sky.com/story/snap-election-to-be-held-in-march-a
soup = BeautifulSoup(url.read().decode('utf8'), "lxml")
text = '\n\n'.join(map(lambda p: p.text, soup.find_all('p')))

text = text[text.find('An early election'):]
title = soup.find('h1').text.strip()
print(title, '\n', '_' * 60, '\n', text)
```

Snap election to be held in March after Northern Ireland government collapse
s

An early election will be held in Northern Ireland on 2 March after the col
lapse of its government, it has been announced.

Northern Ireland Secretary James Brokenshire said the devolved Northern Irel
and Assembly will sit for the last time on 25 January, before it is dissolve
d the following day.

The break-up of the power-sharing government comes amid a dispute between Si
nn Fein and the DUP over a botched renewable energy scheme that could have c
ost the taxpayer £500m.

The "cash for ash" scandal prompted the resignation of deputy first minister
Martin McGuinness, who called for DUP first minister Arlene Foster to quit.

She refused, calling Mr McGuinness' actions "not principled" and "purely pol
itical".

On Monday afternoon, Sinn Fein announced it would not replace Mr McGuinness
- triggering the snap election.

Despite a last-ditch attempt by Theresa May to urge a resolution, Sinn Fein
MLA Conor Murphy said his party had decided to "call time on the arrogance o
f the DUP".

He said: "We have had scandal after scandal, allegations of corruption need
to be investigated properly and the people responsible need to be held to ac
count."

Mrs Foster, who presided over the controversial renewable energy scheme as e
nterprise minister, claimed Sinn Fein "did not like the election result last
May and are therefore looking to have another go".

:: What does the Northern Ireland crisis mean for Brexit?

Announcing the dissolution of the Northern Ireland Assembly, Mr Brokenshire
urged both parties "to conduct this election with a view to...re-establishin
g a partnership government at the earliest opportunity after that poll."

He said: "This is essential for the operation of devolved government. And th
is means that all must remain open to dialogue."

Sinn Fein and the DUP are expected to remain the two largest parties followi
ng the election, meaning they will still have to hammer out a power-sharing
arrangement.

If they fail to agree terms after three weeks, Mrs May could be forced to su
spend devolution and reinstate direct rule from Westminster.

Sky News Ireland Correspondent David Blevins said the relationship between S
inn Fein and the DUP had been "slowly breaking down for a period of months".

He said: "Some would suggest that the British and Irish governments took the
ir eye off the ball.

```
"The botched renewable energy scheme is being blamed for the collapse of the
devolved government but it was just the tip of the iceberg."

He added that the collapse of the power-sharing government was the "greatest
challenge to face the Northern Ireland peace process in a decade".


© 2017 Sky UK
```

Calculating the similarity between sentences

This method goes through the following steps:

- split text into sentences;
- split sentences into words/tokens - there are several ways to do it, which give various results, I'll show them;
- calculate similarity between sentences - while there are many ways to do it, I'll use a simple one: comparing tokens in each sentence. Similarity between sentences is calculated as number of words which are present in both sentences divided by average length of sentences (for normalization);
- assign scores to sentences based on their similarity with other sentences - for each sentence get a sum of similarity scores with each other sentence;
- select several best sentences and show them in order, in which they appeare in the article;

At first I'll simply split sentences into words, using space as a separator.

In [3]:
```
def intersection(sent1, sent2):
    s1 = sent1.split(' ')
    s2 = sent2.split(' ')

    intersection = [i for i in s1 if i in s2]
    #Normalization
    return len(intersection) / ((len(s1) + len(s2)) / 2)
```

Now creating a matrix of similarities between each pair of sentences. This is a 2D-matrix with a length equal to the number of sentences.

In [4]:
```
sentences = sent_tokenize(text)
matrix = [[intersection(sentences[i], sentences[j]) for i in range(0,len(ser
matrix[:2]
```

Out[4]:
```
[[1.0,
  0.40816326530612246,
  0.1568627450980392,
  0.08695652173913043,
  0.0,
  0.10256410256410256,
  0.15384615384615385,
  0.25,
  0.1111111111111111,
  0.1875,
  0.3018867924528302,
  0.12121212121212122,
  0.0,
  0.16326530612244897,
  0.08888888888888889,
  0.2127659574468085,
  0.10256410256410256,
  0.34782608695652173,
  0.4,
  0.0],
 [0.24489795918367346,
  1.0,
  0.10714285714285714,
  0.11764705882352941,
  0.0,
  0.09090909090909091,
  0.17543859649122806,
  0.03773584905660377,
  0.1016949152542373,
  0.21621621621621623,
  0.20689655172413793,
  0.21052631578947367,
  0.0,
  0.18518518518518517,
  0.0,
  0.19230769230769232,
  0.09090909090909091,
  0.3137254901960784,
  0.24,
  0.0]]
```

Now calculating the score for each sentence, which is a sum of simiarity scores with other sentences.

In [5]:
```
scores = {sentences[i]: sum(matrix[i]) for i in range(len(matrix))}
scores
```

Out[5]:

```
{'"The botched renewable energy scheme is being blamed for the collapse of the devolved government but it was just the tip of the iceberg."': 3.7260340696948515,
 ':: What does the Northern Ireland crisis mean for Brexit?': 3.9061312784591125,
 'An early election will be held in Northern Ireland on 2 March after the collapse of its government, it has been announced.': 4.195413155308382,
 'And this means that all must remain open to dialogue."': 2.0006127492497647,
 'Announcing the dissolution of the Northern Ireland Assembly, Mr Brokenshire urged both parties "to conduct this election with a view to...re-establishing a partnership government at the earliest opportunity after that poll."': 3.814137780904025,
 'Despite a last-ditch attempt by Theresa May to urge a resolution, Sinn Fein MLA Conor Murphy said his party had decided to "call time on the arrogance of the DUP".': 3.894282386960945,
 'He added that the collapse of the power-sharing government was the "greatest challenge to face the Northern Ireland peace process in a decade".': 4.696759739072093,
 'He said: "Some would suggest that the British and Irish governments took their eye off the ball.': 3.4353927892443257,
 'He said: "This is essential for the operation of devolved government.': 4.306593046828432,
 'He said: "We have had scandal after scandal, allegations of corruption need to be investigated properly and the people responsible need to be held to account."': 4.112337244524717,
 'If they fail to agree terms after three weeks, Mrs May could be forced to suspend devolution and reinstate direct rule from Westminster.': 2.204353090231064,
 'Mrs Foster, who presided over the controversial renewable energy scheme as enterprise minister, claimed Sinn Fein "did not like the election result last May and are therefore looking to have another go".': 3.8391866673233834,
 'Northern Ireland Secretary James Brokenshire said the devolved Northern Ireland Assembly will sit for the last time on 25 January, before it is dissolved the following day.': 3.531232869189094,
 'On Monday afternoon, Sinn Fein announced it would not replace Mr McGuinness - triggering the snap election.': 3.393079931658959,
 'She refused, calling Mr McGuinness\' actions "not principled" and "purely political".': 1.50805253635191,
 'Sinn Fein and the DUP are expected to remain the two largest parties following the election, meaning they will still have to hammer out a power-sharing arrangement.': 4.168767988167754,
 'Sky News Ireland Correspondent David Blevins said the relationship between Sinn Fein and the DUP had been "slowly breaking down for a period of months".': 4.635577167457064,
 'The "cash for ash" scandal prompted the resignation of deputy first minister Martin McGuinness, who called for DUP first minister Arlene Foster to quit.': 3.7494398066338452,
 'The break-up of the power-sharing government comes amid a dispute between Sinn Fein and the DUP over a botched renewable energy scheme that could have cost the taxpayer £500m.': 4.399355207918554,
 '© 2017 Sky UK': 1.0689655172413792}
```


Now I'll select five best sentences.

In [6]:
```
sents = sorted(scores, key=scores.__getitem__, reverse=True)[:5]
sents
```

Out[6]:
```
['He added that the collapse of the power-sharing government was the "great
est challenge to face the Northern Ireland peace process in a decade".',
 'Sky News Ireland Correspondent David Blevins said the relationship betwee
n Sinn Fein and the DUP had been "slowly breaking down for a period of mont
hs".',
 'The break-up of the power-sharing government comes amid a dispute between
Sinn Fein and the DUP over a botched renewable energy scheme that could hav
e cost the taxpayer £500m.',
 'He said: "This is essential for the operation of devolved government.',
 'An early election will be held in Northern Ireland on 2 March after the c
ollapse of its government, it has been announced.']
```

Maybe there is a better way to sort sentences based on the order in which they appear in text, but this still works.

In [7]:
```
tuples = [(i, text.find(i)) for i in sents]
sorted_tuples = sorted(tuples, key=lambda x: x[0])
#Leave only sentences.
best_sents = [i[0] for i in sorted_tuples]
best_sents
```

Out[7]:
```
['An early election will be held in Northern Ireland on 2 March after the c
ollapse of its government, it has been announced.',
 'He added that the collapse of the power-sharing government was the "great
est challenge to face the Northern Ireland peace process in a decade".',
 'He said: "This is essential for the operation of devolved government.',
 'Sky News Ireland Correspondent David Blevins said the relationship betwee
n Sinn Fein and the DUP had been "slowly breaking down for a period of mont
hs".',
 'The break-up of the power-sharing government comes amid a dispute between
Sinn Fein and the DUP over a botched renewable energy scheme that could hav
e cost the taxpayer £500m.']
```

Now, I'll put everything together with a nice output.

In [8]:
```
def intersection(sent1, sent2):
    s1 = sent1.split(' ')
    s2 = sent2.split(' ')
    intersection = [i for i in s1 if i in s2]
    return len(intersection) / ((len(s1) + len(s2)) / 2)

def get_summary(text, limit=3):
    sentences = sent_tokenize(text)
    matrix = [[intersection(sentences[i], sentences[j]) for i in range(0,ler
    scores = {sentences[i]: sum(matrix[i]) for i in range(len(matrix))}
    sents = sorted(scores, key=scores.__getitem__, reverse=True)[:limit]
    best_sents = [i[0] for i in sorted([(i, text.find(i)) for i in sents], k
    return best_sents

def summarize(text, limit=3):
    summary = get_summary(text, limit)
```

```
    print(title)
    print()
    print(' '.join(summary))
```

In [9]:
```
summarize(text,5)
```

```
Snap election to be held in March after Northern Ireland government collapse
s

An early election will be held in Northern Ireland on 2 March after the coll
apse of its government, it has been announced. He added that the collapse of
the power-sharing government was the "greatest challenge to face the Norther
n Ireland peace process in a decade". He said: "This is essential for the op
eration of devolved government. Sky News Ireland Correspondent David Blevins
said the relationship between Sinn Fein and the DUP had been "slowly breakin
g down for a period of months". The break-up of the power-sharing government
comes amid a dispute between Sinn Fein and the DUP over a botched renewable
energy scheme that could have cost the taxpayer £500m.
```

So, this is a summary. The number of sentences in summary is arbitrary and can be changed to get the necessary result.

How can this algorithm be improved? I think that splitting sentences while calculating intersections should be changed. Splitting by spaces leaves punctuation attached to the words, which leads to mistakes when evaluating similarity between sentences. So I'll tokenize sentences using nltk and remove stopwords and punctuation. Also taking lemmas of words could help (but didn't help in this case - I tried).

In [10]:
```
def intersection(sent1, sent2):
    s1 = [i for i in word_tokenize(sent1) if i not in punctuations and i not
    s2 = [i for i in word_tokenize(sent2) if i not in punctuations and i not
    intersection = [i for i in s1 if i in s2]
    return len(intersection) / ((len(s1) + len(s2)) / 2)
```

In [11]:
```
summarize(text,5)
```

```
Snap election to be held in March after Northern Ireland government collapse
s

An early election will be held in Northern Ireland on 2 March after the coll
apse of its government, it has been announced. Announcing the dissolution of
the Northern Ireland Assembly, Mr Brokenshire urged both parties "to conduct
this election with a view to...re-establishing a partnership government at t
he earliest opportunity after that poll." He added that the collapse of the
power-sharing government was the "greatest challenge to face the Northern Ir
eland peace process in a decade". Sky News Ireland Correspondent David Blevi
ns said the relationship between Sinn Fein and the DUP had been "slowly brea
king down for a period of months". The break-up of the power-sharing governm
ent comes amid a dispute between Sinn Fein and the DUP over a botched renewa
ble energy scheme that could have cost the taxpayer £500m.
```

We see that the summary changed. And in one last change I'll increase the complexity of the model even further. Tokenizing sentences is good, but a better idea would be to use n_grams. For this I use gensim's Phrases. Phrases detects collocations in text and can be used for finding n_grams in text.

In [12]:
```
sents = sent_tokenize(text)
#Phrases need input as list of lists of tokens.
sentence_stream = [[i for i in word_tokenize(sent) if i not in stop] for ser
bigram = Phrases(sentence_stream, min_count=2, threshold=2, delimiter=b' ')
#Create Phraser object.
bigram_phraser = Phraser(bigram)
bigram_tokens = bigram_phraser[sentence_stream]
trigram = Phrases(bigram_tokens, min_count=2, threshold=2, delimiter=b' ')
trigram_phraser = Phraser(trigram)
trigram_tokens = trigram_phraser[bigram_tokens]
all_words = [i for j in trigram_tokens for i in j]

Counter(all_words).most_common(20)
```

Out[12]:
```
[('government', 6),
 ('Northern Ireland', 6),
 ('election', 5),
 ('scandal', 3),
 ('devolved', 3),
 ('Sinn Fein DUP', 3),
 ('Mr', 3),
 ('McGuinness', 3),
 ('Sinn Fein', 3),
 ('May', 3),
 ('renewable energy scheme', 3),
 ('He said', 3),
 ('power-sharing', 3),
 ('minister', 3),
 ('said', 3),
 ('collapse', 3),
 ('The', 3),
 ('need', 2),
 ('held', 2),
 ('Mrs', 2)]
```

We can see that there are bigrams and trigrams among the most common words. Now I'll use this.

In [13]:
```
def intersection(sent1, sent2):
    #As sentences are lists of tokens, there is no need to split them.
    intersection = [i for i in sent1 if i in sent2]
    return len(intersection) / ((len(sent1) + len(sent2)) / 2)

def split_sentences(sents):
    sentence_stream = [[i for i in word_tokenize(sent) if i not in stop] for
    bigram = Phrases(sentence_stream, min_count=2, threshold=2, delimiter=b'
    bigram_phraser = Phraser(bigram)
    bigram_tokens = bigram_phraser[sentence_stream]
```

```
    trigram = Phrases(bigram_tokens,min_count=2, threshold=2, delimiter=b'_'
    trigram_phraser = Phraser(trigram)
    trigram_tokens = trigram_phraser[bigram_tokens]
    return [i for i in trigram_tokens]

def get_summary(text, limit=3):
    sents = sent_tokenize(text)
    sentences = split_sentences(sents)
    matrix = [[intersection(sentences[i], sentences[j]) for i in range(0,ler
    scores = {sents[i]: sum(matrix[i]) for i in range(len(matrix))}
    sents = sorted(scores, key=scores.__getitem__, reverse=True)[:limit]
    best_sents = [i[0] for i in sorted([(i, text.find(i)) for i in sents], k
    return best_sents
```

In [14]:
```
summarize(text,5)
```

```
Snap election to be held in March after Northern Ireland government collapse
s

"The botched renewable energy scheme is being blamed for the collapse of the
devolved government but it was just the tip of the iceberg." An early electi
on will be held in Northern Ireland on 2 March after the collapse of its gov
ernment, it has been announced. Announcing the dissolution of the Northern I
reland Assembly, Mr Brokenshire urged both parties "to conduct this election
with a view to...re-establishing a partnership government at the earliest op
portunity after that poll." He added that the collapse of the power-sharing
government was the "greatest challenge to face the Northern Ireland peace pr
ocess in a decade". The break-up of the power-sharing government comes amid
a dispute between Sinn Fein and the DUP over a botched renewable energy sche
me that could have cost the taxpayer £500m.
```

The summary changed again. Various ways to split sentences may work better on some types of texts and worse on others.

Calculating words' frequencies

This method goes through the following steps:

- split text into sentences and sentences into tokens;
- assign scores to sentences based on the frequency of words in these sentences. It could be simply a point for each frequent word in sentence or some score based on frequency of certain words. Maybe add additional points if word is also in a title (as word in title should be more important). There are other options, but the main idea is that sentences's score is based on words in this sentence. It is similar to the idea of assigning scores based on similarities between sentences, but works in another way;
- select several best sentences and show them in order, in which their appear in the article;

In [15]:
```
def score_sentences(words, sentences):
    #Return scores for sentences.
```

```
    scores = Counter()
    #Words - list of words and their scores, first element is the word, secc
    for word in words:
        for i in range(0, len(sentences)):
            #If word is also in title, then add double score to the sentence
            if word[0] in sentences[i] and word[0] in title:
                scores[i] += 2 * word[1]
            elif word[0] in sentences[i]:
                scores[i] += word[1]
    sentence_scores = sorted(scores.items(), key=scores.__getitem__, reverse
    return sentence_scores

def split_sentences(sents):

    sentence_stream = [[i for i in word_tokenize(sent) if i not in stop] for
    bigram = Phrases(sentence_stream, min_count=2, threshold=2, delimiter=b'
    bigram_phraser = Phraser(bigram)
    bigram_tokens = bigram_phraser[sentence_stream]
    trigram = Phrases(bigram_tokens,min_count=2, threshold=2, delimiter=b'_'
    trigram_phraser = Phraser(trigram)
    trigram_tokens = trigram_phraser[bigram_tokens]

    all_words = [i for j in trigram_tokens for i in j]
    frequent_words = [i for i in Counter(all_words).most_common() if i[1] >
    sentences = [i for i in trigram_tokens]

    return frequent_words, sentences

def get_summary(text, limit=3):
    sents = sent_tokenize(text)
    frequent_words, sentences = split_sentences(sents)
    sentence_scores = score_sentences(frequent_words, sentences)

    limited_sents = [sents[num] for num, count in sentence_scores[:limit]]
    best_sents = [i[0] for i in sorted([(i, text.find(i)) for i in limited_s
    return best_sents

def summarize(text, limit=3):
    summary = get_summary(text, limit)
    print(title)
    print()
    print(' '.join(summary))
```

In [16]:
```
summarize(text, 5)
```

```
Snap election to be held in March after Northern Ireland government collapse
s

An early election will be held in Northern Ireland on 2 March after the coll
apse of its government, it has been announced. Northern Ireland Secretary Ja
mes Brokenshire said the devolved Northern Ireland Assembly will sit for the
last time on 25 January, before it is dissolved the following day. She refus
ed, calling Mr McGuinness' actions "not principled" and "purely political".
The "cash for ash" scandal prompted the resignation of deputy first minister
Martin McGuinness, who called for DUP first minister Arlene Foster to quit.
The break-up of the power-sharing government comes amid a dispute between Si
nn Fein and the DUP over a botched renewable energy scheme that could have c
ost the taxpayer £500m.
```

Conclusions

As I have shown, there are many ways to summarize articles with extraction methods. Of course, there are many other ideas which could improve the algorithms. And it is difficult to measure the accuracy of summaries - often there are many "meaningful" sentences and choosing one best combination of them isn't possible. So we just try several ways and choose the best implementation for a particular case. And try developing abstraction methods, as extraction methods are limited.