exploration

March 24, 2022

1 WB-NLP-2022 PD1 - Preprocessing przykładowych danych, eksploracja

Mateusz Krzyziński, Piotr Wilczyński, Artur Żółkowski

Poniższy notebook jest rozwiązaniem pracy domowej numer 1 i dotyczy eksploracji zbioru Database of Parliamentary Speeches in Ireland, 1919-2013, zawierającego teksty przemówień parlamentarzystów irlandzkich.

Ładowanie danych i pakietów

[ ]:
```
! pip install swifter
! pip install pandas
! pip install textacy
! pip install -U kaleido
```

[ ]:
```
!pip install spacy
```

[ ]:
```
!python -m spacy download en_core_web_sm
```

[4]:
```
import spacy
import pandas as pd
from tqdm.auto import tqdm
import swifter
import plotly.express as px
from wordcloud import WordCloud
from matplotlib import pyplot as plt
import numpy as np
import textacy
from IPython.display import Image
pd.options.plotting.backend = "plotly"
```

[ ]:
```
!wget -O data.tar.gz "https://dataverse.harvard.edu/api/access/datafile/:
↪persistentId?persistentId=doi:10.7910/DVN/6MZN76/CRUNFO"
!wget -O Dail_debates_1937-2011_ministers.tab "https://dataverse.harvard.edu/
↪api/access/datafile/:persistentId?persistentId=doi:10.7910/DVN/6MZN76/BFFQKZ"
```

[6]:
```
!tar -xf data.tar.gz
```

[5]:
```
en = spacy.load("en_core_web_sm")
df = pd.read_table('Dail_debates_1919-2013.tab')
```

[6]:
```
df.head(5)
```

[6]:

	speechID	memberID	partyID	constID	title	date	member_name	party_name	const_name	speech
0	1	977	22	158	1. CEANN COMHAIRLE I gCOIR AN LAE.	1919-01-21	Count George Noble, Count Plunkett	Sinn Féin	Roscommon North	Molaimse don Dáil Cathal Brugha, an Teachta ó …
1	2	1603	22	103	1. CEANN COMHAIRLE I gCOIR AN LAE.	1919-01-21	Mr. Pádraic Ó Máille	Sinn Féin	Galway Connemara	Is bród mór damhsa cur leis an dtairgsin sin. …
2	3	116	22	178	1. CEANN COMHAIRLE I gCOIR AN LAE.	1919-01-21	Mr. Cathal Brugha	Sinn Féin	Waterford County	' A cháirde, tá obair thábhachtach le déanamh …
3	4	116	22	178	2. CLEIRIGH I gCOIR AN LAE.	1919-01-21	Mr. Cathal Brugha	Sinn Féin	Waterford County	Tá ceathrar cléireach uainn I gcóir gnótha an …
4	5	116	22	178	3. AN ROLLA.	1919-01-21	Mr. Cathal Brugha	Sinn Féin	Waterford County	Léighfead anois ainmneacha na ndaoine a fuair …

1.0.1 Wstępne informacje

Cała ramka danych zawiera 4 443 713 wierszy (przemówień) i 10 kolumn, w tym 6 kluczowych z punktu widzenia eksploracji danych: - tytuł przemówienia, - data przemówienia, - imię i nazwisko przemawiającego, - nazwa partii politycznej, - nazwa okręgu wyborczego, z którego był wybrany parlamentarzysta, - tekst przemówienia.

Nie ma żadnych braków danych.

[7]:
```
df.isnull().sum()
```

[7]:
```
speechID       0
memberID       0
partyID        0
constID        0
title          0
date           0
member_name    0
party_name     0
```

```
const_name     0
speech         0
dtype: int64
```

[8]:
```
df.date = pd.to_datetime(df.date)
```

1.0.2 Krótka eksploracja danych pozatekstowych

(niezbędne dla kontekstu dalszej eksploracji)

Dane są za okres od 1919 do 2013 roku. W tym czasie swoje wystąpienia w parlamencie miało 1178 parlamentarzystów z 27 różnych partii.

W irlandzkim parlamencie (dokładniej niższej jego izbie - Dáil Éireann) zasiada 160 parlamentarzystów.

W zbiorze danych znajdują się przemówienia parlamentarzystów wybranych z 151 okręgów wyborczych (natomiat okręgi zmieniały się w czasie - obecnie jest ich 39).

[9]:
```
df.date.min(), df.date.max()
```

[9]:
```
(Timestamp('1919-01-21 00:00:00'), Timestamp('2013-03-28 00:00:00'))
```

[10]:
```
df.constID.nunique(), df.partyID.nunique(), df.memberID.nunique()
```

[10]:
```
(151, 27, 1178)
```

Wyraźnie widać, że najwięcej przemówień wygłosili przedstawiciele czterech partii (które miały też najwięcej parlamentarzystów w izbie w tym okresie). Warto przyjrzeć się krótko tym partiom.

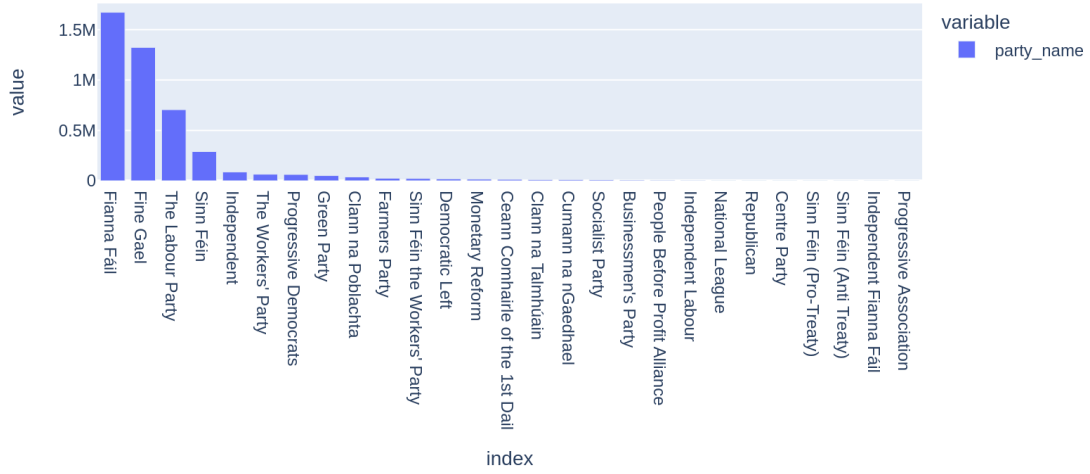
- **Fianna Fail - The Republican Party** (*Żołnierze Losu*) - założona w 1926, partia konserwatywna, chrześcijańsko-demokratyczna, wielokrotnie rządziła sama lub w koalicji, sprzeciwia się silnej integracji europejskiej
- **Fine Gael** (*Rodzina Irlandczyków*) - założona w 1933, liberalno-konserwatywa, chrześcijańsko-demokratyczna, uważana za skłonną do kompromisów z Wielką Brytanią, zwolenniczka integracji europejskiej
- **The Labour Party** - założona w 1912, partia centro-lewicowa, socjaldemokratyczna, proeuropejska, trzecia główna partia, rządząca w koalicjach z Fine Gael
- **Sinn Féin** (*My Sami*) - założona w 1905 (w obecnej formie w 1970), partia lewicowo-nacjonalistyczna, republikańska, jej głównym celem pozostaje zjednoczenie obu części Irlandii w jedną republikę

[11]:
```
fig = df.party_name.value_counts().plot(kind='bar')
img_bytes = fig.to_image(format="png", width=800, height=400, scale=2,␣
↪engine='kaleido')
```

[12]:
```
#w notatniku
fig
```

```
#do pdf
Image(img_bytes)
```

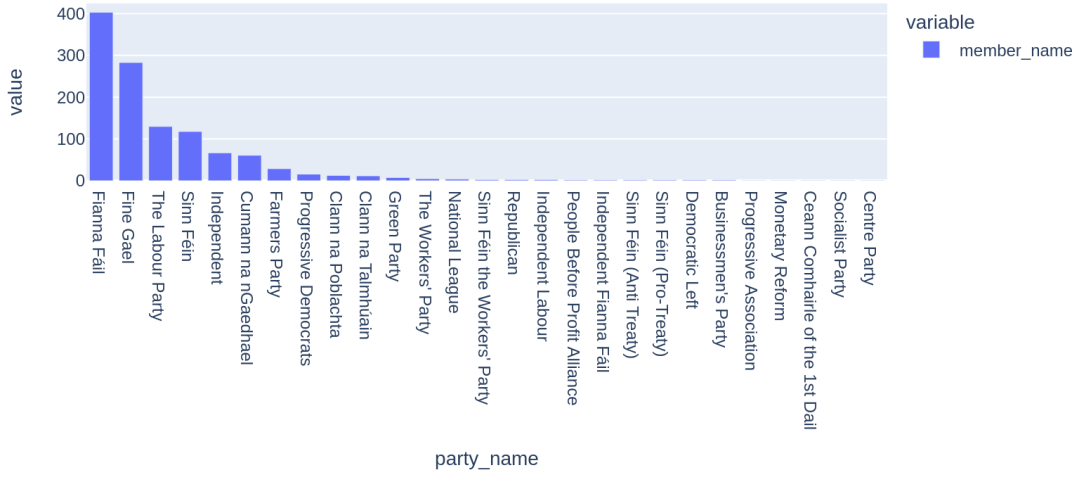
[12]:



[17]:

```
fig = df.groupby("party_name").member_name.nunique().
 ↪sort_values(ascending=False).plot(kind='bar')
img_bytes = fig.to_image(format="png", width=800, height=400, scale=2,␣
 ↪engine='kaleido')
Image(img_bytes)
```

[17]:



Dla zmniejszenia rozmiaru danych ograniczamy się do przemów w parlamencie

począwszy od 6 czerwca 2002 roku, kiedy miało miejsce zaprzysiężenie parlamentarzystów 29-tego Dáil Éireann (Zgromadzenia). Dane z tego okresu stanowią ponad 25% wszystkich danych.

Ponadto dla tego okresu podział na partie wygląda podobnie, choć przewaga dwóch największych partii jest bardziej widoczna, a Sinn Fein straciło popularność wśród wyborców.

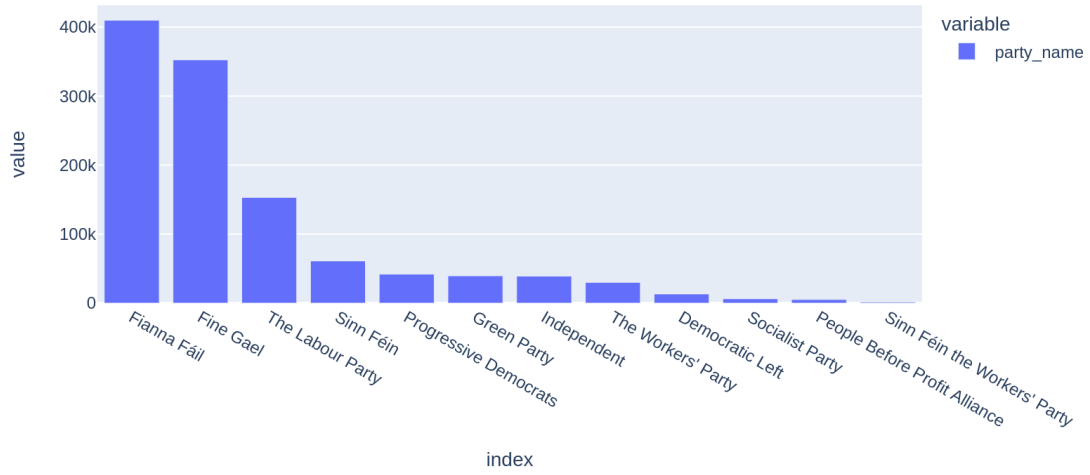
[18]:
```
df2 = df[df.date >= np.datetime64('2002-06-06')]
```

[19]:
```
len(df2)/len(df)
```

[19]: 0.2597015153768932

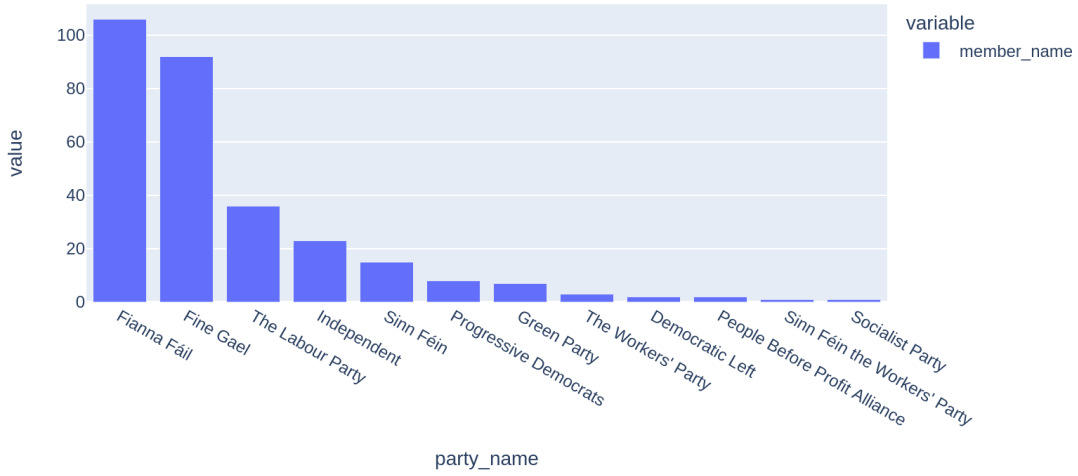
[20]:
```
fig = df2.party_name.value_counts().plot(kind='bar')
img_bytes = fig.to_image(format="png", width=800, height=400, scale=2,␣
↪engine='kaleido')
Image(img_bytes)
```

[20]:



[21]:
```
fig = df2.groupby("party_name").member_name.nunique().
↪sort_values(ascending=False).plot(kind='bar')
img_bytes = fig.to_image(format="png", width=800, height=400, scale=2,␣
↪engine='kaleido')
Image(img_bytes)
```

[21]:



[22]:
```
df2.constID.nunique(), df2.partyID.nunique(), df2.memberID.nunique()
```

[22]: (58, 12, 296)

Ponadto z wybranego okresu bierzemy losowy sample przemówień. Ostatecznie będziemy pracować na około 23 tys. przemówień.

[23]:
```
df3 = df2.sample(frac=.02, random_state=42)
len(df3)
```

[23]: 23081

1.1 Analiza ze względu na długość przemówień

[24]:
```
tqdm.pandas()
docs = df3['speech'].swifter.apply(en)
```

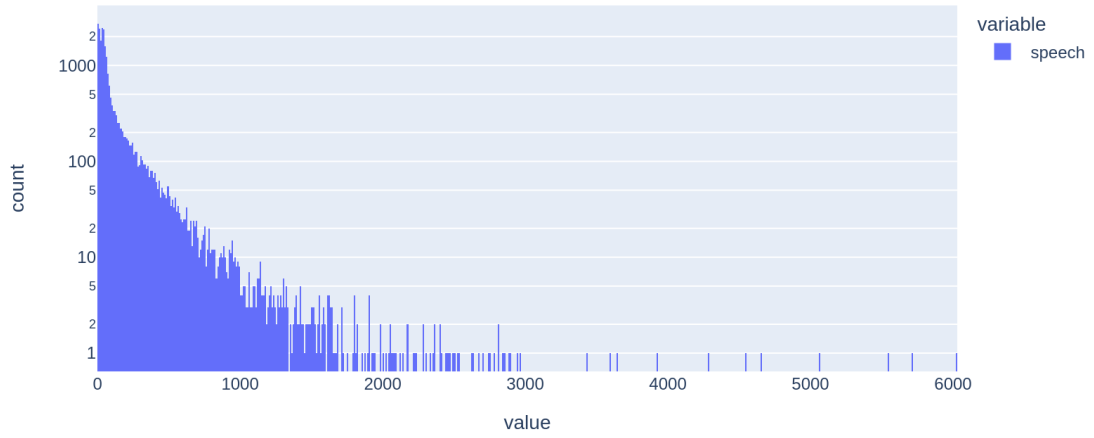
```
Pandas Apply:   0%|          | 0/23081 [00:00<?, ?it/s]
```

Rozkład długości wystąpień - najwięcej jest krótkich przemówień, do 1000 słów. Te powyżej 2000, a w szczególności 3000 tysięcy sprawiają wrażenie outlierów. Spróbujemy się coś o nich dowiedzieć.

[25]:
```
doc_lens = docs.str.len()
fig = doc_lens.hist(log_y=True)
img_bytes = fig.to_image(format="png", width=800, height=400, scale=2,␣
 ↪engine='kaleido')
```

```
Image(img_bytes)
```

[25]:



[26]:
```
ministers_df = pd.read_table("Dail_debates_1937-2011_ministers.tab")
ministers_df = ministers_df[~ministers_df.memberID.isna()]
ministers_df["memberID"] = ministers_df["memberID"].astype(int)
```

Najdłuższa z wypowiedzi dotyczyła ustawy o napojach alkoholowych. Wygłosił ją w czerwcu 2003 roku Michael McDowell, który w ówczesnym rządzie pełnił funkcję Ministra Sprawiedliwości, Równości i Reform Prawnych. Widzimy również, że część z najdłuższych wystąpień jest powiązana ze sprawami budżetu, czysto finansowymi. Warto sprawdzić, jak często to właśnie ministrowie są "odpowiedzialni" za najdłuższe przemówienia.

[27]:
```
df3.title[doc_lens[doc_lens > 3000].sort_values(ascending=False).index]
```

[27]:
```
3378357    Intoxicating Liquor Bill 2003 [ Seanad ] : Sec…
3631498                    Finance Bill 2006: Second Stage.
3968154                Supplementary Budget Statement 2009.
3408801                          Financial Resolutions 2003.
3964322    Industrial Development Bill 2008 [Seanad]: Sec…
4032421           Adoption Bill 2009 [Seanad]: Second Stage.
4247409      Competition (Amendment) Bill 2011: Second Stage
3419352    Private Members' Business. - Protection of Chi…
3608659    Sea-Fisheries and Maritime Jurisdiction Bill 2…
3350530    Broadcasting (Major Events Television Coverage…
3404005    Industrial Relations (Amendment) Bill 2003: Se…
Name: title, dtype: object
```

[28]:
```
df3.loc[3378357]
```

[28]:
```
speechID                                                  3378456
memberID                                                      719
partyID                                                        21
constID                                                        92
title          Intoxicating Liquor Bill 2003 [ Seanad ] : Sec…
date                                           2003-06-24 00:00:00
member_name                                    Mr. Michael McDowell
party_name                                     Progressive Democrats
const_name                                        Dublin South-East
speech         I move: 'That the Bill be now read a Second Ti…
Name: 3378357, dtype: object
```

[29]:
```
ministers_df[ministers_df.memberID == 719]
```

[29]:

	govt_number	start_day	start_month	start_year	end_day	end_month	\
741	26	6	6	2002	14.0	6.0	
742	26	13	9	2006	14.0	6.0	

	end_year	position	department	name	\
741	2007.0	Minister	Justice, Equality and Law Reform	Michael McDowell	
742	2007.0	Tánaiste	Tánaiste	Michael McDowell	

	memberName	memberID	start_date	end_date
741	Mr. Michael McDowell	719	2002-06-06	2007-06-14
742	Mr. Michael McDowell	719	2006-09-13	2007-06-14

[30]:
```
long_speeches_idx = doc_lens[doc_lens >= 2000].sort_values(ascending=False).
 ↪index
short_speeches_idx = doc_lens[doc_lens < 2000].sort_values(ascending=False).
 ↪index
```

[31]:
```
ministers_ids = ministers_df.memberID.unique()
```

[32]:
```
ministers_ratio_long = np.mean(df3.loc[long_speeches_idx].memberID.
 ↪isin(ministers_ids))
ministers_ratio_short = np.mean(df3.loc[short_speeches_idx].memberID.
 ↪isin(ministers_ids))
ministers_ratio = np.mean(df3.memberID.isin(ministers_ids))

ministers_ratio, ministers_ratio_short, ministers_ratio_long
```

[32]:
```
(0.6987565530089684, 0.698927066591373, 0.6333333333333333)
```

Ta hipoteza się nie potwierdziła - nie tylko przemówienia ministrów są długie (dłuższe niż 2000 słów) szczególnie często. Jest wręcz odwrotnie - ich procentowy udział w dłuższych przemówieniach jest mniejszy aniżeli w ogóle i w krótszych. Natomiast widać wyraźną różnice pomiędzy poszczególnymi politykami, jeżeli chodzi

o ich średnią długość, jak i ilość wystąpień.

[33]:
```
df3["speech_length"] = doc_lens
```

[34]:
```
speakers = df3.groupby(["memberID", "member_name", "party_name"]).
↪agg({'speech_length': ['mean', 'size']}).reset_index()
```

[35]:
```
longest_speaches = speakers.sort_values([('speech_length', 'mean')],␣
↪ascending=False)
longest_speaches
```

[35]:

	memberID	member_name	party_name	speech_length	
				mean	size
168	1842	Mr. Ollie Wilkinson	Fianna Fáil	710.000000	2
286	2342	Mr. Derek Nolan	The Labour Party	533.250000	4
263	2317	Mr. Tom Barry	Fine Gael	514.600000	5
132	1768	Mr. Joe Callanan	Fianna Fáil	472.000000	2
233	2286	Mr. Martin Heydon	Fine Gael	445.750000	4
..	…	…	…	…	…
171	1928	Mr. Jim Glennon	Fianna Fáil	24.000000	3
91	888	Dr. Rory O'Hanlon	Fianna Fáil	20.109015	477
250	2303	Mr. Ray Butler	Fine Gael	18.000000	5
90	882	Mr. Noel O'Flynn	Fianna Fáil	17.250000	16
116	1093	Mr. Dan Wallace	Fianna Fáil	9.000000	1

[289 rows x 5 columns]

[36]:
```
most_speaches = speakers.sort_values([('speech_length', 'size')],␣
↪ascending=False)
most_speaches
```

[36]:

	memberID	member_name	party_name	speech_length	
				mean	size
34	355	Mr. Bernard Durkan	Fine Gael	63.149425	783
47	486	Ms. Mary Harney	Fianna Fáil	125.224138	580
58	580	Mr. Enda Kenny	Fine Gael	113.787056	479
46	484	Ms. Mary Hanafin	Fianna Fáil	192.207113	478
91	888	Dr. Rory O'Hanlon	Fianna Fáil	20.109015	477
..	…	…	…	…	…
54	546	Mr. Joe Jacob	Fianna Fáil	379.000000	1
275	2329	Mr. Noel Harrington	Fine Gael	64.000000	1
116	1093	Mr. Dan Wallace	Fianna Fáil	9.000000	1
41	432	Ms. Mildred Fox	Independent	37.000000	1
88	861	Ms. Liz O'Donnell	Progressive Democrats	59.000000	1

[289 rows x 5 columns]

Spośród 25 najczęściej wypowiadających się polityków tylko jeden nie był

ministrem. Jest to Caoimhghín Ó Caoláin - polityk Sinn Féin obecny w każdej z analizowanych kadencji.

[37]:
```
most_speaches.memberID[:25].isin(ministers_ids)
```

[37]:
```
34      True
47      True
58      True
46      True
91      True
25      True
72      True
14      True
68      True
1       True
23      True
32      True
64      True
53      True
0       True
102     True
28      True
85     False
101     True
103     True
13      True
89      True
43      True
106     True
7       True
Name: memberID, dtype: bool
```

[38]:
```
most_speaches.loc[85]
```

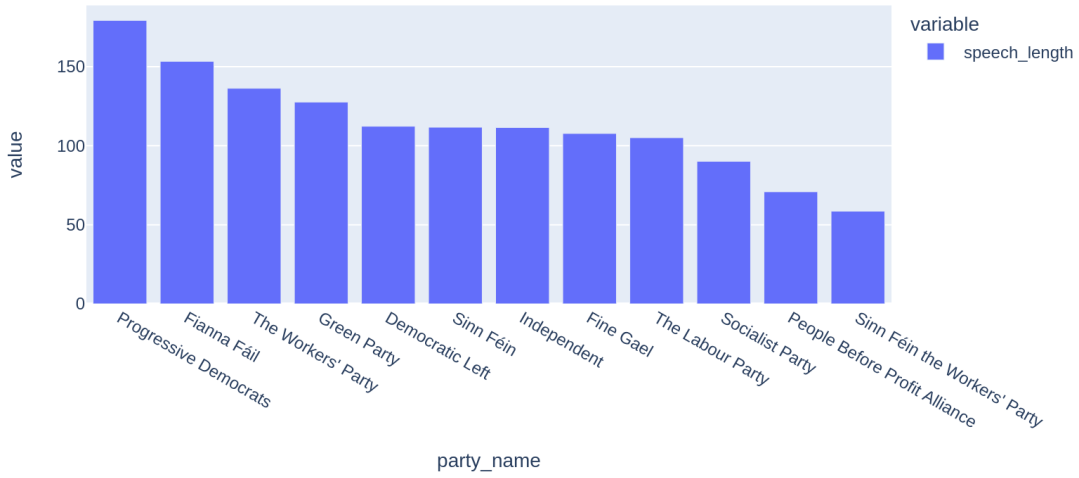
[38]:
```
memberID                                    837
member_name            Mr. Caoimhghín Ó Caoláin
party_name                             Sinn Féin
speech_length  mean                   108.221854
               size                          302
Name: 85, dtype: object
```

Wyraźne są również dość wyraźne różnice pomiędzy długościami wypowiedzi przedstawicieli poszczególnych partii.

[40]:
```
fig = df3.groupby(["party_name"]).agg({'speech_length': 'mean'}).
 ↪sort_values("speech_length", ascending=False).plot(kind='bar')
img_bytes = fig.to_image(format="png", width=800, height=400, scale=2,␣
 ↪engine='kaleido')
```

```
Image(img_bytes)
```

[40]:



[41]:
```
df3.party_name.value_counts()
```

[41]:
```
Fianna Fáil                    8285
Fine Gael                      6976
The Labour Party               3037
Sinn Féin                      1233
Progressive Democrats           882
Independent                     775
Green Party                     754
The Workers' Party              596
Democratic Left                 280
Socialist Party                 133
People Before Profit Alliance    96
Sinn Féin the Workers' Party     34
Name: party_name, dtype: int64
```

1.2 Analiza ze względu na treść przemówień

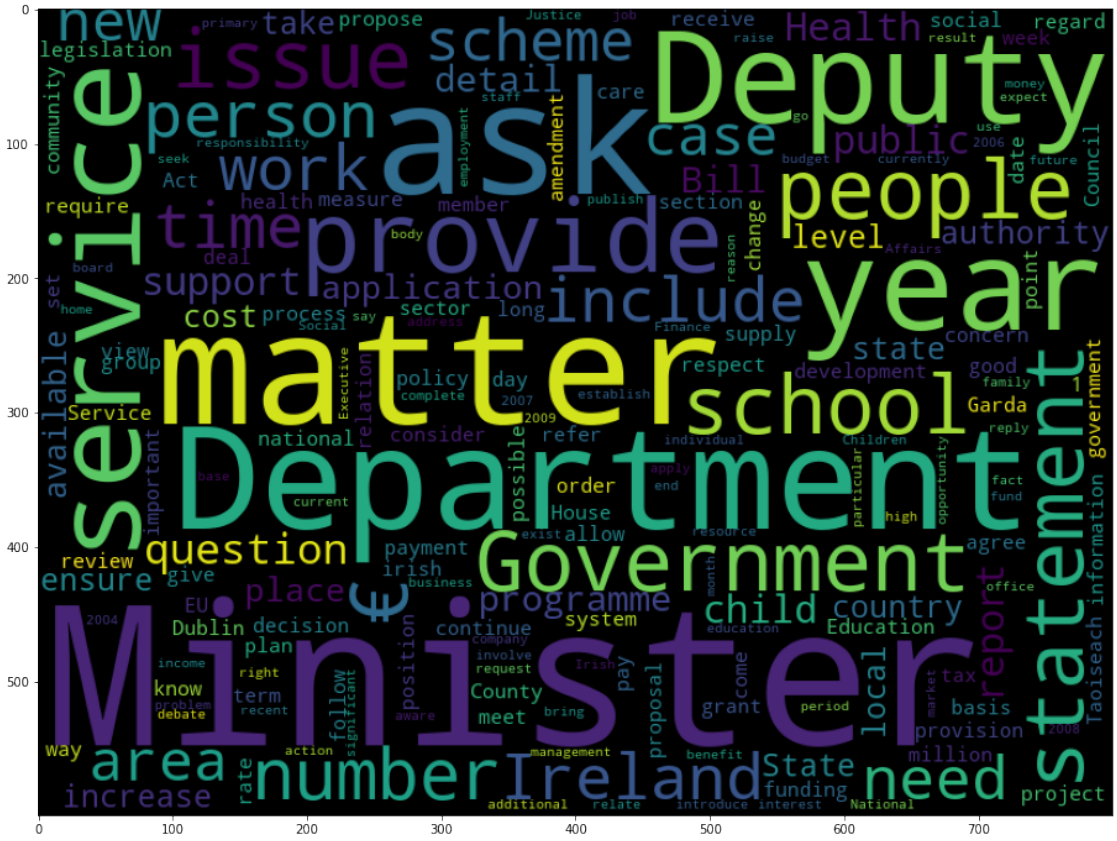
[42]:
```
lemmas = docs.apply(lambda doc: [token.lemma_ for token in doc if not token.
↪is_stop if not token.is_punct])
```

[43]:
```
from collections import Counter
word_counts = Counter(lemmas.sum())
```

W chmurze słów dla wszystkich analizowanych przemówień pojawiają się słowa

związane z politycznym dyskursem, np. powiązane z pozycjami/strukturami: Minister, Department, Government, Deputy, House.

[44]:
```
wc = WordCloud(width=800, height=600)
wc.generate_from_frequencies(frequencies=word_counts)
plt.figure(figsize=(16,12))
plt.imshow(wc)
plt.show()
```



Dokładniejszy insight w najczęściej używane słowa daje wykres słupkowy. Tutaj możemy dostrzec kilka ciekawych rzeczy, niezwiązanych już tylko z samym dyskursem.

- Wśród najpopularniejszych słów są np. 'school' i 'child', co może wskazywać na to, że często poruszany był temat edukacji.
- Dość zaskakujące jest też częstotliwość występowania '€' - z pozoru mogłoby się wydawać, że politycy rzadko w wystąpieniach operują konkretnymi kwotami pieniężnymi.

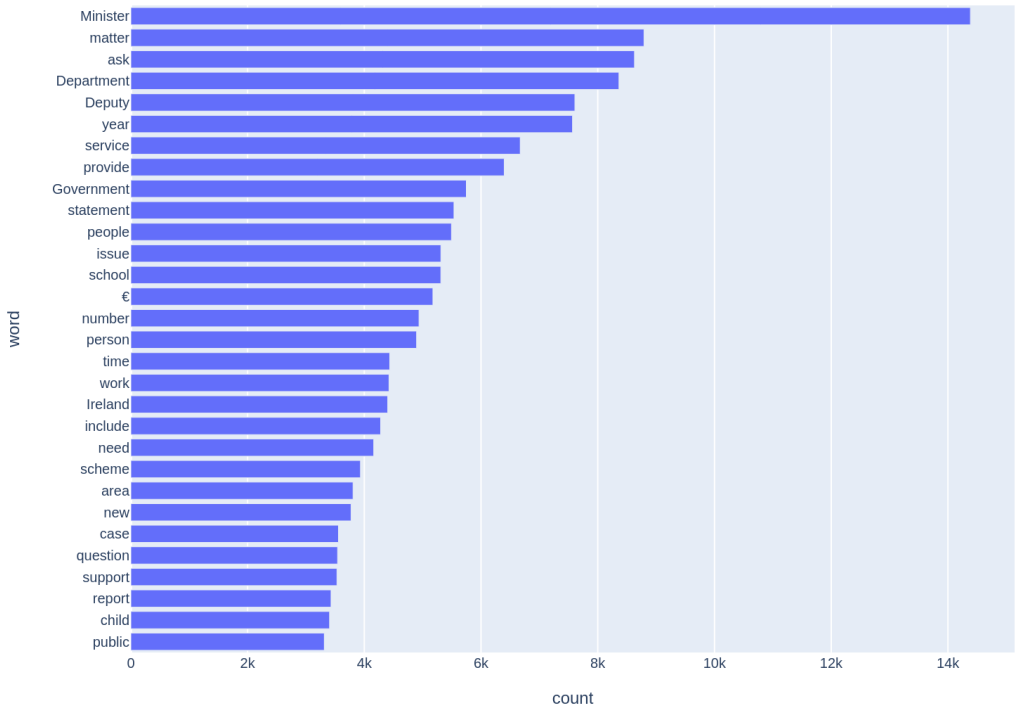
- Ciekawym jest też częste występowanie słowa 'person' będącego bardziej osobistym niż grupowe 'people'. Wartym uwagi pytaniem badawczym byłaby analiza, jak wyglądają zależności względem częstości użycia tych właśnie

słów.

[45]:
```
counts = pd.DataFrame(word_counts.most_common(30), columns=['word', 'count'])
```

[66]:
```
fig = px.bar(counts,orientation='h', y='word', x='count')
fig['layout']['yaxis']['autorange'] = "reversed"
fig.update_layout(bargap=0.2, font={'size':10})
img_bytes = fig.to_image(format="png", width=800, height=600, scale=2,␣
↪engine='kaleido')
Image(img_bytes)
```

[66]:



[47]:
```
party_names = ["Fianna Fáil", "Fine Gael", "The Labour Party", "Sinn Féin"]
FF_indices = df3.index[df3.party_name == party_names[0]]
FG_indices = df3.index[df3.party_name == party_names[1]]
LP_indices = df3.index[df3.party_name == party_names[2]]
SF_indices = df3.index[df3.party_name == party_names[3]]
```

[48]:
```
FF_word_counts = Counter(lemmas[FF_indices].sum())
FG_word_counts = Counter(lemmas[FG_indices].sum())
```

[49]:
```
LP_word_counts = Counter(lemmas[LP_indices].sum())
SF_word_counts = Counter(lemmas[SF_indices].sum())
```

[50]:
```
FF_top_100_words = pd.DataFrame(FF_word_counts.most_common(100),␣
↪columns=['word', 'count'])
FG_top_100_words = pd.DataFrame(FG_word_counts.most_common(100),␣
↪columns=['word', 'count'])
LP_top_100_words = pd.DataFrame(LP_word_counts.most_common(100),␣
↪columns=['word', 'count'])
SF_top_100_words = pd.DataFrame(SF_word_counts.most_common(100),␣
↪columns=['word', 'count'])
```

[51]:
```
top_100_words_by_parties = pd.merge(FF_top_100_words, FG_top_100_words, on =␣
↪'word', how = 'outer').merge(LP_top_100_words, on = 'word', how = 'outer').
↪merge(SF_top_100_words, on = 'word', how = 'outer')
```

[52]:
```
top_100_words_by_parties.columns = ["word"] + party_names
top_100_words_by_parties.set_index("word", inplace=True)
top_100_words_by_parties = top_100_words_by_parties.fillna(0)
```

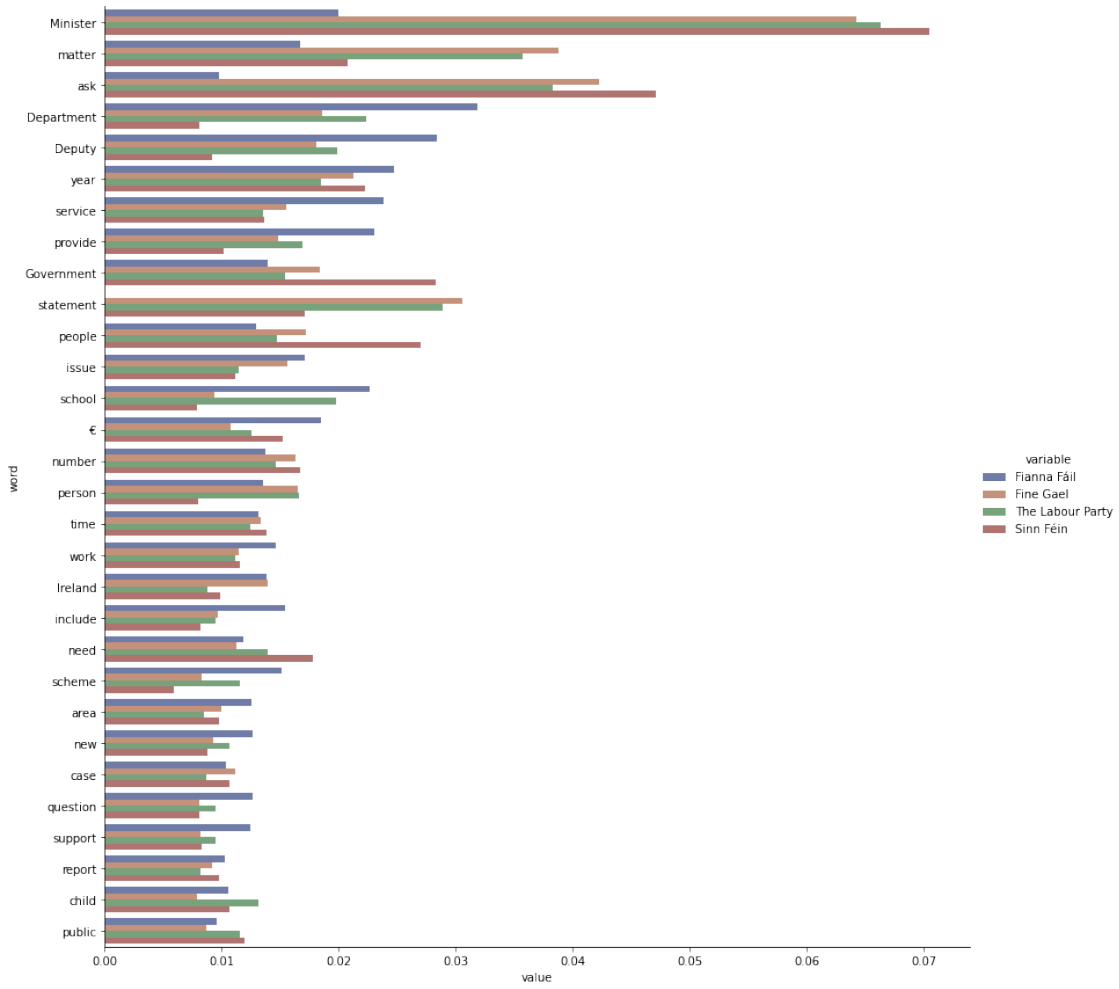
[53]:
```
top_100_words_by_parties_ratios = top_100_words_by_parties.
↪div(top_100_words_by_parties.sum(axis=0), axis=1)
```

[54]:
```
top_100_wbp_plot = top_100_words_by_parties_ratios.loc[counts.word.to_list()].
↪reset_index().melt(id_vars="word", value_vars=party_names)
```

Porównanie jaką część z sumy 100 najczęściej używanych słów przez każdą z 4 głównych partii stanowi 30 najpopularniejszych słów ogółem również dostarcza nam ciekawych informacji: - politcy Fianna Fail stosunkowo rzadziej mówią o ministrach i nie używają słowa 'ask', co jest powiązane z tym, że to politycy partii rządzącej przez większość analizowanego zakresu czasowego, a to politycy opozycyjni częściej zwracają się do ministrów rządu, - politycy Fianna Fail i Sinn Fein zdecydowanie rzadziej używają słowa 'matter' od pozostałych partii, - słowo 'statement' nie znalazło się w ogóle wśród 100 najpopularniejszych słów dla partii Fianna Fail, - politycy Sinn Fein zdecydowanie częściej używają określenia ogółu - 'people' aniżeli słowa 'person'; dla innych partii stosunek ten jest porównywalny, - tematem szkół zdecydowanie częściej zajmowali się w swoich przemówieniach politycy FF i The Labour Party, - zauważalne jest podobieństwo między częstością występowania danych słów (poruszania pewnych tematów?) przez polityków Fine Gael i The Labour Party, które wchodzą ze sobą w koalicje.

[55]:
```
import seaborn as sns
sns.catplot(
    data=top_100_wbp_plot, kind="bar",
    y="word", x="value", hue="variable",
    palette="dark", alpha=.6, height=12, orientation="horizontal"
)
```

```
plt.show()
```



[56]:
```
top_100_words_by_parties_ratios['sum'] = top_100_words_by_parties_ratios.
↪sum(axis=1)
```

Okazuje się, że partia Fianna Fail stosunkowo częściej od innych mówi chociażby o UE (przypomnijmy, że jest przeciwna silnej integracji). W ich retoryce pojawia się też częściej zdrowie.

Fine Gael często mówi o Gardzie - irlandzkiej policji. Pojawienie się słowa 'Affairs' może być związane z Department of Foreign Affairs.

Sinn Fein mówi częściej o podatkach, gospodarce, bankach, pieniądzach, ale również o wspólnocie (przypomnijmy, że dążą do zjednoczenia z Irlandią Północną).

Partia Pracy odróżnia się natomiast tym, że mówi o edukacji, sprawach socjalnych i zasiłkach.

[57]:
```
party_specific_words = dict(zip(party_names, [[] for i in range(4)]))
for party in party_names:
  for word, row in top_100_words_by_parties_ratios.iterrows():
    if row[party] >= row['sum'] - row[party]:
      party_specific_words[party].append(word)
party_specific_words
```

[57]:
```
{'Fianna Fáil': ['Act',
  'information',
  'health',
  'continue',
  'million',
  'national',
  'EU',
  'follow',
  'refer',
  'grant',
  'require',
  'Service',
  'set',
  'section',
  'basis',
  'group',
  'policy',
  '1',
  'meet',
  'primary',
  'Council',
  'good',
  'responsibility',
  'Executive'],
 'Fine Gael': ['date', 'Garda', 'important', 'Affairs', 'staff'],
 'Sinn Féin': ['tax',
  'measure',
  'community',
  'come',
  'job',
  'Taoiseach',
  'know',
  'money',
  'amendment',
  'Finance',
  'Reform',
  'proposal',
  'address',
  'cut',
  'agus',
```

```
 'bank',
 'right',
 'economy',
 'fact',
 'past',
 'family',
 'week',
 'Equality',
 'Tánaiste',
 'company',
 'na',
 'budget',
 'month',
 'party',
 'bring',
 'long'],
'The Labour Party': ['Education',
 '2011',
 '2012',
 'appeal',
 'Social',
 'education',
 'welfare']}
```

Analiza bigramów i trigramów, które najczęściej pojawiają się w tekstach dostarcza kolejnych informacji o tym, co jest poruszane w przemówieniach parlamentarzystów. W tym przypadku bardziej widać zakresy tematyczne niż analizując pojedyncze słowa. Widzimy np. że ważnym tematem jest irlandzka policja, ale bardzo często pojawia się też temat zasiłków społecznych, a nawet praw człowieka.

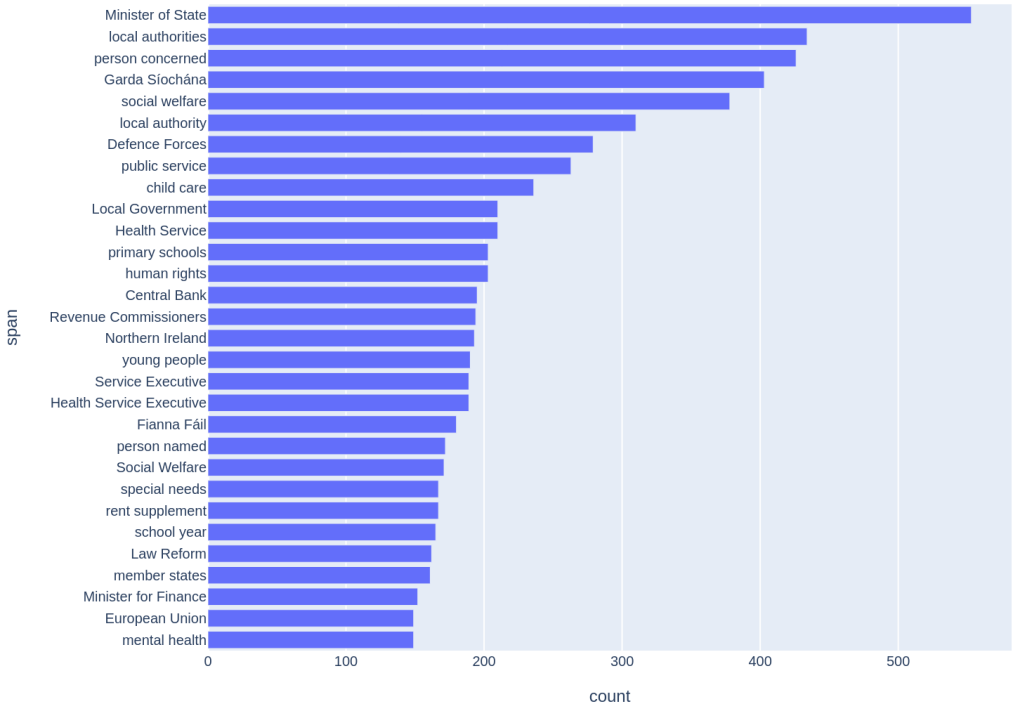
[58]:
```
bigrams = docs.apply(lambda doc: [span.text for span in textacy.extract.
↪ngrams(doc, n=(2,3), min_freq=2)])
```

[59]:
```
bitrigrams_counts = Counter(bigrams.sum())
```

[60]:
```
top_30_bitrigrams = pd.DataFrame(bitrigrams_counts.most_common(30),␣
↪columns=['span', 'count'])
```

[65]:
```
fig = px.bar(top_30_bitrigrams,orientation='h', y='span', x='count')
fig['layout']['yaxis']['autorange'] = "reversed"
fig.update_layout(bargap=0.2, font={'size':10})
img_bytes = fig.to_image(format="png", width=800, height=600, scale=2,␣
↪engine='kaleido')
Image(img_bytes)
```

[65]:



1.3 Analiza najważniejszych terminów w przemówieniach przedstawicieli danych ministerstw

[67]:
```
ministers_df.head()
```

[67]:

	govt_number	start_day	start_month	start_year	end_day	end_month	end_year	position	department	name	memberName
0	18	9	3	1982	14.0	12.0	1982.0	Minister of State	Taoiseach	Bertie Ahern	Mr. Bertie Ahern
1	18	23	3	1982	14.0	12.0	1982.0	Minister of State	Defence	Bertie Ahern	Mr. Bertie Ahern
2	20	10	3	1987	12.0	7.0	1989.0	Minister	Labour	Bertie Ahern	Mr. Bertie Ahern
3	21	12	7	1989	14.0	11.0	1991.0	Minister	Labour	Bertie Ahern	Mr. Bertie Ahern
4	21	14	11	1991	11.0	2.0	1992.0	Minister	Finance	Bertie Ahern	Mr. Bertie Ahern

```
   memberID  start_date    end_date
0         5  1982-03-09  1982-12-14
1         5  1982-03-23  1982-12-14
2         5  1987-03-10  1989-07-12
3         5  1989-07-12  1991-11-14
4         5  1991-11-14  1992-02-11
```

[68]:
```
ministers_df["position"].unique()
```

[68]:
```
array(['Minister of State', 'Minister', 'Tánaiste', 'Taoiseach',
       'Secretary'], dtype=object)
```

[69]:
```
ministers_df["department"].unique()
```

[69]:
```
array(['Taoiseach', 'Defence', 'Labour', 'Finance',
       'Industry and Commerce', 'Arts, Culture and the Gaeltacht',
       'Tánaiste', 'Foreign Affairs', 'Social Welfare',
       'Marine and Natural Resources', 'Justice, Equality and Law Reform',
       'Enterprise, Trade and Employment', 'Education and Science',
       'Environment and Local Government',
       'Community, Rural and Gaeltacht Affairs', 'Transport',
       'Co-ordination of Defence Measures', 'External Affairs',
       'Agriculture', 'Education', 'Environment', 'Justice', 'Marine',
       'Energy', 'Agriculture and Food', 'Local Government',
       'Transport and Power', 'Health', 'Gaeltacht',
       'Industry, Commerce and Tourism', 'Posts and Telegraphs',
       'Agriculture and Fisheries', 'Lands', 'Public Service',
       'Communications, Marine and Natural Resources',
       'Industry and Energy', 'Industry, Trade, Commerce and Tourism',
       'Transport, Energy and Communications',
       'Enterprise and Employment', 'Jobs, Enterprise and Innovation',
       'Industry, Commerce and Energy', 'Communications',
       'Social Protection', 'Trade, Commerce and Tourism',
       'Health and Children',
       'Communications, Energy and Natural Resources',
       'Local Government and Public Health', 'Minister without Portfolio',
       'Tourism and Transport', 'Foreign Affairs and Trade',
       'Arts, Heritage, Gaeltacht and the Islands',
       'Social, Community and Family Affairs', 'Education and Skills',
       'Agriculture, Food and the Marine', 'Social and Family Affairs',
       'Arts, Sport and Tourism', 'Fisheries and Forestry',
       'Tourism, Fisheries and Forestry',
       'Agriculture, Food and Forestry', 'Arts, Heritage and Gaeltacht',
       'Health and Social Welfare',
       'Tourism, Transport and Communications',
       'Children and Youth Affairs', 'Tourism, Sport and Recreation',
       'Equality and Law Reform',
```

```
       'Environment, Heritage and Local Government',
       'Tourism, Culture and Sport', 'Public Expenditure and Reform',
       'Environment, Community and Local Government', 'Public Enterprise',
       'Tourism and Trade', 'Agriculture, Fisheries and Food', 'Supplies',
       'Fisheries', 'Government', 'Justice and Equality',
       'Finance and Public Service',
       'Agriculture, Food and Rural Development',
       'Economic Planning and Development', 'Labour and Public Service',
       'Transport, Tourism and Sport', 'Justice and Law Reform',
       'Education and Public Service',
       'Community, Equality and Gaeltacht Affairs'], dtype=object)
```

Usuwamy wiersze dotyczące premierów, wicepremierów i sekretarzy oraz zbędne kolumny. Zostawimy też dane o ministrach tylko od 6 czerwca 2002 roku tak jak w ramce danych dotyczącej przemówień.

[70]:
```
ministers_df_prepoc = ministers_df.loc[(ministers_df["department"] !=␣
↪"Taoiseach") & (ministers_df["department"] != "Tánaiste"),
                                       ["position", "department", "name",␣
↪"memberID", "start_date", "end_date"]]
ministers_df_prepoc.start_date = pd.to_datetime(ministers_df_prepoc.start_date)
ministers_df_prepoc.end_date = pd.to_datetime(ministers_df_prepoc.end_date)
```

[71]:
```
ministers_df_prepoc.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 1058 entries, 1 to 1184
Data columns (total 6 columns):
 #   Column      Non-Null Count  Dtype
---  ------      --------------  -----
 0   position    1058 non-null   object
 1   department  1058 non-null   object
 2   name        1058 non-null   object
 3   memberID    1058 non-null   int64
 4   start_date  1058 non-null   datetime64[ns]
 5   end_date    1025 non-null   datetime64[ns]
dtypes: datetime64[ns](2), int64(1), object(3)
memory usage: 57.9+ KB
```

[72]:
```
pd.isnull(ministers_df_prepoc["end_date"])
```

[72]:
```
1       False
2       False
3       False
4       False
5       False
        …
```

```
1180    False
1181    False
1182     True
1183     True
1184     True
Name: end_date, Length: 1058, dtype: bool
```

[73]:
```
ministers_df_prepoc.loc[pd.isnull(ministers_df_prepoc["end_date"])].start_date.
↪unique()
```

[73]:
```
array(['2011-03-09T00:00:00.000000000'], dtype='datetime64[ns]')
```

W ramce są braki danych. Nie ma wpisanej daty zakończenia piastowania stanowiska, jeżeli minister zaczął pełnić rolę 9 marca 2011. Będziemy zatem rozważali okres do 2011-03-09.

[74]:
```
ministers_df_prepoc.loc[ministers_df_prepoc["start_date"] != np.
↪datetime64('2011-03-09')].info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 1025 entries, 1 to 1181
Data columns (total 6 columns):
 #   Column      Non-Null Count  Dtype
---  ------      --------------  -----
 0   position    1025 non-null   object
 1   department  1025 non-null   object
 2   name        1025 non-null   object
 3   memberID    1025 non-null   int64
 4   start_date  1025 non-null   datetime64[ns]
 5   end_date    1025 non-null   datetime64[ns]
dtypes: datetime64[ns](2), int64(1), object(3)
memory usage: 56.1+ KB
```

[75]:
```
ministers_df_prepoc = ministers_df_prepoc.loc[(ministers_df_prepoc["end_date"] !
↪= np.datetime64('2011-03-09')) & (ministers_df_prepoc["end_date"] >= np.
↪datetime64('2002-06-06'))]
```

Wyciągnijmy informacje o przemówieniach, które były prowadzone przez ministrów.

[76]:
```
df4 = df3.copy()
df4["department"] = "False"
```

[77]:
```
for index, row in df3.iterrows():
    member_id = row["memberID"]
    speech_date = row["date"]
    temp_df = ministers_df_prepoc.loc[(ministers_df_prepoc["memberID"] ==␣
↪member_id) &\
```

```
                                        (ministers_df_prepoc["start_date"] <=␣
↪speech_date) &\
                                        (ministers_df_prepoc["end_date"] >=␣
↪speech_date)]
    if(len(temp_df)):
      df4.at[index, "department"] = temp_df["department"].iloc[0]
df4.department.value_counts()
```

[77]:
```
False                                          17851
Health and Children                              846
Justice, Equality and Law Reform                 794
Education and Science                            709
Finance                                          312
Environment, Heritage and Local Government       311
Transport                                        259
Defence                                          236
Foreign Affairs                                  222
Enterprise, Trade and Employment                 203
Environment and Local Government                 187
Agriculture and Food                             176
Communications, Marine and Natural Resources     143
Public Enterprise                                143
Communications, Energy and Natural Resources      81
Agriculture, Food and Rural Development           79
Arts, Heritage, Gaeltacht and the Islands         75
Tourism, Sport and Recreation                     75
Arts, Sport and Tourism                           75
Enterprise and Employment                         73
Social, Community and Family Affairs              69
Community, Rural and Gaeltacht Affairs            63
Marine and Natural Resources                      52
Social and Family Affairs                         31
Agriculture, Fisheries and Food                   15
Justice and Law Reform                             1
Name: department, dtype: int64
```

[78]:
```
df4 = df4.loc[df4["department"] != "False"]
```

Dla każdego przemówienia będziemy wyciągać 3 najważniejsze terminy. Następnie będziemy je zliczać, aby stwierdzić, które występują w największej liczbie przemówień.

[79]:
```
department_keywords = {dep : np.array([]) for dep in df4.department.unique()}
```

[80]:
```
for index, row in df4.iterrows():
```

```
    department_keywords[row["department"]] = np.
↪append(department_keywords[row["department"]], [term[0] for term in textacy.
↪extract.keyterms.textrank(docs[index])[0:3]])
```

[81]:
```
df_keywords = pd.DataFrame(columns=['depatament', 'most_frequent',␣
↪'second_most_frequent', 'third_most_frquent'])
for key in department_keywords:
  unique, counts = np.unique(department_keywords[key], return_counts=True)
  count_sort_idx = np.argsort(-counts)
  df_keywords = df_keywords.append({'depatament' : key, 'most_frequent' :␣
↪unique[count_sort_idx][0],'second_most_frequent' :␣
↪unique[count_sort_idx][1],'third_most_frquent' : unique[count_sort_idx][2]},␣
↪ignore_index=True)
```

[82]:
```
df_keywords
```

[82]:

	depatament	most_frequent
0	Agriculture and Food	Single Payment Scheme
1	Communications, Marine and Natural Resources	Deputy
2	Education and Science	primary school
3	Communications, Energy and Natural Resources	
4	Transport	
5	Justice, Equality and Law Reform	
6	Public Enterprise	
7	Health and Children	
8	Finance	
9	Defence	
10	Foreign Affairs	
11	Enterprise and Employment	
12	Environment, Heritage and Local Government	
13	Social and Family Affairs	
14	Arts, Heritage, Gaeltacht and the Islands	
15	Environment and Local Government	
16	Enterprise, Trade and Employment	
17	Tourism, Sport and Recreation	
18	Social, Community and Family Affairs	
19	Arts, Sport and Tourism	
20	Marine and Natural Resources	
21	Agriculture, Food and Rural Development	
22	Community, Rural and Gaeltacht Affairs	
23	Agriculture, Fisheries and Food	
24	Justice and Law Reform	

3	amendment
4	Deputy
5	Deputy
6	supplementary welfare allowance scheme
7	Health Service Executive
8	Revenue Commissioners
9	question
10	Deputy
11	Deputy
12	Water Services Investment Programme
13	1953]an organisational structure
14	CLÁR area
15	Deputy
16	Deputy
17	sport capital programme
18	reduced fee
19	sport capital programme
20	Deputy
21	area aid application
22	Deputy
23	issue
24	Midland Traveller Conflict

	second_most_frequent	\
0	application	
1	Question	
2	school authority	
3	Deputy	
4	matter	
5	Refugee Applications Commissioner	
6	social welfare payment	
7	personal social service	
8	Deputy	
9	Dáil Éireann	
10	EU member state	
11	Bill	
12	Deputy	
13	qualified adult payment	
14	Waterways Ireland	
15	local authority	
16	work permit section	
17	national level	
18	record	
19	Local Authority Swimming Pool Programme	
20	EU Commission	
21	suckler cow premium scheme	
22	information	

```
23                                  Chair
24            Pavee Point Mediation Service

                         third_most_frquent
0                     single payment scheme
1                              14th February
2                 school building programme
3                 energy efficiency measure
4                     Road Safety Authority
5                           Garda authority
6             social welfare pension system
7                               direct reply
8                               Public Works
9                                    Deputy
10                    UN Secretary General
11                                question
12       water service investment programme
13                  proper financial system
14                      rural social scheme
15                relevant local authority
16                              Department
17  local authority swimming pool programme
18                    disability allowance
19                    Irish Sports Council
20                     Mayo County Council
21            compensatory allowance scheme
22                              staff action
23                          provide revenue
24                Traveller policy division
```

Przykładowe wnioski: - Ministerstwo Agrokultury i Żywności zajmowało się 'Single Payment Scheme'. Jest to wsparcie finansowe wypłacane rolnikom. Otrzymują oni dopłatę za hektar ziemi wykorzystywanej pod uprawę. - Ministerstwo Edukacji zajmowało się głównie problemami szkół podstawowych. - Ministerstwo Sprawiedliwości i Równości częstwo odwoływało się do pozycji Komisarza ds. Wniosków Uchodźców. - Ministerstwo zdrowia poruszało temat usług socjalnych. Często odwoływało się do roli Kierownika służby zdrowia - Ministerstwo spraw zewnętrznych często wspominało o członkostwie w Unii Europejskiej. - Ministerstwo środowiska przemawiało na temat programu 'Water Services Investment Programme', czyli planu inwestycji w szeroko pojętą infrastrukturę wodną.

[ ]:
```
!jupyter nbconvert --to PDF "/content/drive/MyDrive/Colab Notebooks/exploration.
↪ipynb"
```