Capitulo_3

Flavio Codeço Coelho
(Dated: September 12, 2019)

CONTENTS

I. Identificando Entidades Nomeadas 1
I.1. Testando o NER do Spacy 2
I.2. Treinando Um identificador de Entidades a partir do DHBB 3
I.3. Similaridade semantica 18

I. IDENTIFICANDO ENTIDADES NOMEADAS

Neste capítulo vamos treinar um identificador de Entidades (NER) nomeadas usando a biblioteca Spacy. A partir deste capítulo vamos importar também funções que já criamos anteriormente, e que encontram-se reproduzidas em dhbbmining.py

In [1]:
```
import os, glob, pickle
import spacy
from spacy import displacy
from sqlalchemy import create_engine
from dhbbmining import *
import ipywidgets as widgets
```

Para utilizar o spacy em um corpus na lingua portuguesa, vamos primeiro importar o modelo liguístico do português

In [2]:
```
nlp = spacy.load("pt_core_news_sm")
```

Em seguida podemos carregar os verbetes bográficos que salvamos no nosso banco SQLite.

In [3]:
```
eng = create_engine("sqlite:///minha_tabela.sqlite")
#dhbb = pd.read_sql_table('resultados')
biograficos = pd.read_sql_query('select * from resultados', con=eng)
biograficos.head()
```

[3]:

	index	arquivo	title	natureza	sexo	cargos	corpo
0	0	1.text	COELHO, Machado	biográfico	m	\n - dep. fed. DF 1927-1929 \n - dep. fed. DF ...	\n\nńJosé Machado Coelho de Castroż nasceu em ...
1	1	10.text	ABÍLIO, Armando	biográfico	m	\n - dep. fed. PB 1995-1999\n - dep. fed. PB ...	
2	2	100.text	ALEIXO, Pedro	biográfico	m	\n - const. 1934\n - dep. fed. MG 1935-1937\n ...	
3	3	1000.text	CAMPOS, Eduardo	biográfico	m	\n - dep. fed. PE 1995\n - dep. fed. PE 1998-...	
4	4	1001.text	CAMPOS, Eleazar Soares	biográfico	m	\n - magistrado\n - interv. MA 1945-1946\n	

```
1  \n\nńArmando Abílio Vieiraż nasceu em Itaporan...
2  \n\nńPedro Aleixoż nasceu em São Caetano, dist...
3  \n\nńEduardo Henrique Accioly Camposż nasceu e...
4  \n\nńEleazar Soares Camposż nasceu em São Luís...
```

Para começar a utilizar o Spacy, precisamos primeiro precisamos processar o texto. Nesta passagem várias análises linguísticas são realizadas.

```
In [4]: doc = nlp(biograficos.corpo[0].strip())
        type(doc)
```

```
[4]: spacy.tokens.doc.Doc
```

```
In [ ]:
```

```
In [5]: for i, token in enumerate(doc):
            print(token.text, token.lemma_, token.pos_, token.tag_, token.dep_,
                  token.shape_, token.is_alpha, token.is_stop)
            if i>5:
                break
```

```
ń ń PUNCT PU|@PU punct ń False False
José José PROPN PROPN nsubj Xxxx True False
Machado Machado PROPN PROPN flat:name Xxxxx True False
Coelho Coelho PROPN PROPN flat:name Xxxxx True False
de de ADP PRP|@N< case xx True True
Castro Castro PROPN PROP|@P< nmod Xxxxx True False
ż ż PUNCT PU|@PU punct ż False False
```

I.1. Testando o NER do Spacy

Como o Spacy já inclui algum suporte à lingua portuguesa, antes de pensar em treinar nosso próprio NER, podemos avaliar a performance do existente.

Abaixo vamos construir uma visualização interativa da marcação de entidades nos verbetes do DHBB.

```
In [6]: from IPython.display import display,clear_output
        from ipywidgets import interact
```

```
In [7]: @interact(e=(0, len(biograficos)))
        def mostra_ner(e=0):
            text = biograficos.iloc[e].corpo.strip()
            doc = nlp(text)
            displacy.render(doc, style="ent", jupyter=True)
            clear_output(wait=True)
```

```
interactive(children=(IntSlider(value=0, description='e', max=7687), Output()), _dom_classes=('widget-in
```

Além da visualização, podemos extrair as entidades presentes em um verbete:

```
In [8]: for ent in doc.ents:
            print(ent.text, ent.start_char, ent.end_char, ent.label_)
```

```
José Machado Coelho de Castroż 1 31 PER
Lorena 42 48 LOC
SP 50 52 LOC
Ginásio Diocesano de São Paulo 67 97 LOC
Faculdade de Ciências Jurídicas 127 158 ORG
Sociais 161 168 LOC
Cunha 220 225 LOC
SP 227 229 LOC
Rio de
Janeiro 263 277 LOC
Distrito Federal 285 301 LOC
Distrito Federal 357 373 LOC
Câmara 488 494 LOC
Getúlio Vargas 575 589 PER
Ligado 591 597 LOC
Washington Luís 654 669 PER
Guanabara 682 691 LOC
Alcântara 951 960 LOC
Lisboa 965 971 LOC
Getúlio 1050 1057 PER
Vargas 1058 1064 PER
São 1125 1128 LOC
Paulo 1129 1134 LOC
São 1206 1209 LOC
Paulo 1210 1215 LOC
Partido Social Democrático 1231 1257 ORG
PSD 1259 1262 ORG
Assembléia Nacional Constituinte 1284 1316 ORG
ANC 1318 1321 ORG
Carta
 1484 1490 PER
Constituinte 1523 1535 MISC
Congresso 1539 1548 LOC
Comissão Permanente de Obras Públicas da Câmara Federal 1571 1626 ORG
Câmara 1732 1738 ORG
Companhia de Cimento Vale do Paraíba 1784 1820 ORG
Rio de Janeiro 1834 1848 LOC
```

```
In [9]: type(doc.ents[0])
```

```
[9]: spacy.tokens.span.Span
```

I.2. Treinando Um identificador de Entidades a partir do DHBB

Identificadores de entidades são algoritmos treinados em corpora manualmente anotados. Como cada corpora possui um conjunto particular de entidades, para uma performance ótima o ideal é treinarmos o modelo no Próprio DHBB. Para este fim utilizaremos os dicionários já disponíveis no DHBB, juntamente com o índice construído no capítulo 2 para recuperar o contexto de cada entrada dos dicionários.

```
In [10]: from whoosh import index
         import os
         from whoosh.qparser import QueryParser
         from whoosh import qparser
```

O primeiro passo é abrirmos o nosso indice.

```
In [11]: if os.path.exists('indexdir'):
             indice = index.open_dir('indexdir')

In [12]: indice.doc_count()
```

[12]: 6724

```
In [13]: def busca(consulta):
             qp = QueryParser("corpo", indice.schema)
             qp.add_plugin(qparser.EveryPlugin())
             query = qp.parse(consulta)

             with indice.searcher() as searcher:
                 results = [(dict(hit),hit.highlights("corpo")) for hit in searcher.
 ↪search(query,
         limit=None)]
             return results

In [14]: resultados = busca('"filho de"')[0]
```

Agora já temos os ingredientes necessários para treinar um modelo de entidades usando a biblioteca spacy.

```
In [15]: import random, os
         from tqdm import tqdm
         from pathlib import Path
         import spacy
         from spacy.util import minibatch, compounding

In [16]: for verb in biograficos.itertuples():
             print(verb.title)
             break
```

```
COELHO, Machado
```

Primeiro precisamos criar o conjunto de treinamento do modelo. e deve ter a forma de uma lista como a descrita abaixo.

```
TRAIN_DATA = [
    ("nasceu em Itaporanga ( PB ) no dia 29 de dezembro de 1944 , filho de Argemiro Abílio de Sousa
     {"entities": [(10, 20, "LOC"), (69, 93, "PERSON")]}
    ),
]
```

Como primeira abordagem de treinamento, vamos atualizar o modelo que vem com o spacy adicionando mais um tipo de entidade: “eventos”. Posteriormente, o modelo pode continuar a ser incrementado adicionando-se outros tipos de entidade. Vamos também aproveitar as entidades reconhecidas pelo modelo base no corpus do DHBB para reforçar o treinamento do modelo no contexto linguístico do DHBB.

```
In [17]: path = "F:/dhbb-master/dic/evento.txt"
         path = "../../dhbb/dic/evento.txt"
         with open(path, 'r', encoding='utf-8') as f:
             dicio = f.readlines()
         #ent.text, ent.start_char, ent.end_char, ent.label_
         def gera_dados_treinamento(dicionário, tag):
```

```
            data = []
            for verb in tqdm(biograficos.itertuples()):
                texto = verb.corpo
                doc = nlp(texto)
                entdict = {"entities":[(ent.start_char, ent.end_char, ent.label_) for␣
↪ent in
        doc.ents]}
                for evento in dicionário:
                    try:
                        posini =texto.index(evento)
                    except ValueError:
                        continue
                    posfim = posini + len(evento)
                    entdict['entities'].append((posini,posfim, tag))
                data.append((texto, entdict))
            return data
```

```
In [18]: if os.path.exists('ner_training.pickle'):
            dados = pickle.load(open('ner_training.pickle','rb'))
        else:
            dados = gera_dados_treinamento(dicio, "EVT")
```

```
7687it [35:17,  3.43it/s]
```

```
In [19]: pickle.dump(dados,open('ner_training.pickle','wb'))
```

```
In [20]: for d in dados[:10]:
            print(d[1])
```

```
{'entities': [(3, 33, 'PER'), (44, 50, 'LOC'), (52, 54, 'LOC'), (69, 99, 'LOC'), (129,
160, 'ORG'), (163, 170, 'LOC'), (222, 227, 'LOC'), (229, 231, 'LOC'), (265, 279,
'LOC'), (287, 303, 'LOC'), (359, 375, 'LOC'), (490, 496, 'LOC'), (577, 591, 'PER'),
(593, 599, 'LOC'), (656, 671, 'PER'), (684, 693, 'LOC'), (953, 962, 'LOC'), (967, 973,
'LOC'), (1052, 1059, 'PER'), (1060, 1066, 'PER'), (1127, 1130, 'LOC'), (1131, 1136,
'LOC'), (1208, 1211, 'LOC'), (1212, 1217, 'LOC'), (1233, 1259, 'ORG'), (1261, 1264,
'ORG'), (1286, 1318, 'ORG'), (1320, 1323, 'ORG'), (1486, 1492, 'PER'), (1525, 1537,
'MISC'), (1541, 1550, 'LOC'), (1573, 1628, 'ORG'), (1734, 1740, 'ORG'), (1786, 1822,
'ORG'), (1836, 1850, 'LOC')]}
{'entities': [(3, 25, 'PER'), (36, 46, 'LOC'), (48, 50, 'LOC'), (92, 116, 'PER'),
(122, 144, 'PER'), (169, 175, 'LOC'), (180, 212, 'LOC'), (213, 247, 'LOC'), (337, 358,
'LOC'), (418, 444, 'ORG'), (446, 449, 'PER'), (533, 542, 'LOC'), (544, 546, 'LOC'),
(587, 601, 'LOC'), (751, 754, 'PER'), (845, 888, 'PER'), (890, 894, 'ORG'), (927, 938,
'ORG'), (955, 980, 'ORG'), (982, 985, 'PER'), (1077, 1099, 'LOC'), (1237, 1241,
'ORG'), (1420, 1424, 'ORG'), (1466, 1481, 'LOC'), (1495, 1501, 'LOC'), (1540, 1562,
'ORG'), (1616, 1636, 'ORG'), (1688, 1696, 'ORG'), (1725, 1733, 'ORG'), (1748, 1777,
'ORG'), (1780, 1787, 'ORG'), (1884, 1890, 'ORG'), (1892, 1906, 'PER'), (2288, 2297,
'ORG'), (2360, 2386, 'ORG'), (2388, 2391, 'ORG'), (2408, 2437, 'ORG'), (2439, 2442,
'ORG'), (2690, 2707, 'MISC'), (2724, 2727, 'ORG'), (2741, 2760, 'PER'), (2762, 2768,
'PER'), (2797, 2803, 'LOC'), (2861, 2878, 'ORG'), (2928, 2941, 'PER'), (2946, 2971,
'ORG'), (2973, 2975, 'ORG'), (2980, 2983, 'LOC'), (2984, 2989, 'LOC'), (3068, 3080,
'MISC'), (3174, 3186, 'ORG'), (3219, 3248, 'ORG'), (3251, 3259, 'PER'), (3273, 3307,
'ORG'), (3310, 3318, 'ORG'), (3376, 3379, 'ORG'), (3409, 3471, 'ORG'), (3473, 3477,
'ORG'), (3560, 3574, 'PER'), (3633, 3639, 'ORG'), (3699, 3706, 'PER'), (3812, 3823,
'MISC'), (3828, 3832, 'MISC'), (3958, 3973, 'MISC'), (4089, 4109, 'LOC'), (4262, 4270,
'PER'), (4302, 4306, 'MISC'), (4403, 4414, 'LOC'), (4605, 4625, 'LOC'), (4683, 4734,
'ORG'), (4780, 4802, 'LOC'), (4805, 4813, 'LOC'), (4888, 4897, 'LOC'), (4899, 4913,
```

'PER'), (4954, 4969, 'PER'), (4971, 4977, 'PER'), (5006, 5026, 'ORG'), (5100, 5114,
'PER'), (5130, 5170, 'ORG'), (5342, 5367, 'ORG'), (5369, 5371, 'ORG'), (5378, 5405,
'ORG'), (5407, 5412, 'ORG'), (5477, 5481, 'ORG'), (5487, 5491, 'ORG'), (5557, 5575,
'PER'), (5578, 5591, 'PER'), (5656, 5666, 'PER'), (5674, 5678, 'ORG'), (5680, 5687,
'PER'), (5688, 5694, 'PER'), (5777, 5791, 'LOC'), (5899, 5921, 'MISC'), (5936, 5953,
'PER'), (5969, 5978, 'ORG'), (6098, 6112, 'LOC'), (6137, 6141, 'ORG'), (6147, 6154,
'ORG'), (6179, 6182, 'ORG'), (6228, 6248, 'PER'), (6508, 6522, 'PER'), (6557, 6560,
'LOC'), (6587, 6591, 'ORG'), (6740, 6744, 'MISC'), (6808, 6822, 'PER'), (6841, 6846,
'ORG'), (6847, 6851, 'MISC'), (6853, 6856, 'ORG'), (6858, 6861, 'ORG'), (6863, 6866,
'ORG'), (6926, 6929, 'ORG'), (6933, 6939, 'LOC'), (7008, 7028, 'LOC'), (7076, 7082,
'MISC'), (7113, 7124, 'PER'), (7243, 7249, 'MISC'), (7251, 7269, 'PER'), (7495, 7500,
'PER'), (7504, 7513, 'PER'), (7528, 7548, 'PER'), (7559, 7570, 'PER'), (7571, 7577,
'PER'), (7608, 7613, 'LOC'), (7644, 7658, 'MISC'), (7727, 7736, 'PER'), (7761, 7764,
'ORG'), (7799, 7813, 'PER'), (7911, 7928, 'PER'), (7933, 7953, 'ORG'), (7955, 7957,
'ORG'), (7997, 8004, 'LOC'), (8060, 8117, 'ORG'), (8123, 8131, 'LOC'), (8133, 8139,
'PER'), (8176, 8179, 'ORG'), (8210, 8246, 'MISC'), (8249, 8263, 'MISC'), (8368, 8381,
'MISC'), (8384, 8399, 'MISC'), (8437, 8445, 'LOC'), (8562, 8579, 'LOC'), (8647, 8650,
'ORG'), (8654, 8661, 'LOC'), (8677, 8702, 'PER'), (8717, 8731, 'PER'), (8763, 8770,
'LOC')]}
{'entities': [(3, 16, 'PER'), (27, 38, 'LOC'), (65, 72, 'LOC'), (74, 76, 'LOC'), (129,
141, 'PER'), (142, 148, 'PER'), (154, 175, 'PER'), (218, 233, 'PER'), (236, 245,
'LOC'), (250, 257, 'LOC'), (290, 300, 'LOC'), (302, 304, 'LOC'), (350, 365, 'LOC'),
(393, 407, 'LOC'), (412, 427, 'LOC'), (454, 474, 'LOC'), (478, 506, 'LOC'), (508, 511,
'ORG'), (541, 557, 'LOC'), (563, 581, 'PER'), (649, 658, 'LOC'), (677, 695, 'MISC'),
(735, 750, 'PER'), (853, 867, 'PER'), (1008, 1022, 'LOC'), (1077, 1094, 'PER'), (1097,
1119, 'PER'), (1140, 1155, 'LOC'), (1231, 1269, 'LOC'), (1301, 1316, 'LOC'), (1361,
1376, 'ORG'), (1432, 1446, 'PER'), (1452, 1463, 'PER'), (1516, 1525, 'LOC'), (1567,
1585, 'ORG'), (1707, 1715, 'ORG'), (1881, 1898, 'MISC'), (1901, 1913, 'MISC'), (1953,
1966, 'PER'), (1978, 2001, 'MISC'), (2014, 2028, 'PER'), (2211, 2223, 'PER'), (2285,
2290, 'LOC'), (2384, 2390, 'PER'), (2500, 2505, 'LOC'), (2524, 2539, 'PER'), (2633,
2650, 'PER'), (2720, 2721, 'PER'), (2775, 2795, 'MISC'), (2809, 2825, 'PER'), (2827,
2843, 'PER'), (2846, 2858, 'PER'), (2860, 2865, 'PER'), (2866, 2872, 'PER'), (2916,
2928, 'LOC'), (2954, 2969, 'PER'), (3014, 3041, 'ORG'), (3043, 3046, 'ORG'), (3097,
3115, 'MISC'), (3178, 3200, 'PER'), (3234, 3256, 'PER'), (3262, 3265, 'ORG'), (3344,
3358, 'MISC'), (3379, 3393, 'PER'), (3427, 3442, 'PER'), (3448, 3451, 'ORG'), (3529,
3544, 'PER'), (3713, 3722, 'LOC'), (3741, 3755, 'ORG'), (3756, 3768, 'ORG'), (3770,
3773, 'ORG'), (3876, 3888, 'PER'), (3920, 3955, 'LOC'), (3984, 4001, 'MISC'), (4041,
4051, 'LOC'), (4052, 4073, 'ORG'), (4318, 4330, 'PER'), (4356, 4376, 'ORG'), (4378,
4380, 'ORG'), (4385, 4397, 'LOC'), (4410, 4429, 'LOC'), (4488, 4490, 'ORG'), (4564,
4578, 'PER'), (4579, 4597, 'PER'), (4643, 4650, 'PER'), (4651, 4659, 'PER'), (4661,
4684, 'PER'), (4686, 4712, 'PER'), (4725, 4737, 'PER'), (4808, 4811, 'ORG'), (4859,
4874, 'PER'), (4883, 4895, 'MISC'), (4909, 4929, 'LOC'), (4951, 4963, 'ORG'), (4967,
4969, 'ORG'), (4999, 5002, 'ORG'), (5044, 5056, 'PER'), (5134, 5149, 'PER'), (5168,
5174, 'PER'), (5212, 5230, 'PER'), (5234, 5236, 'ORG'), (5390, 5401, 'PER'), (5404,
5427, 'PER'), (5429, 5441, 'PER'), (5465, 5479, 'PER'), (5502, 5512, 'LOC'), (5539,
5551, 'MISC'), (5617, 5622, 'LOC'), (5652, 5669, 'MISC'), (5690, 5705, 'ORG'), (5754,
5773, 'ORG'), (5774, 5781, 'MISC'), (5783, 5786, 'ORG'), (5789, 5817, 'PER'), (5835,
5847, 'PER'), (5872, 5916, 'ORG'), (5972, 5992, 'PER'), (6136, 6170, 'ORG'), (6174,
6194, 'LOC'), (6248, 6253, 'LOC'), (6268, 6270, 'ORG'), (6313, 6327, 'LOC'), (6341,
6344, 'ORG'), (6399, 6411, 'PER'), (6453, 6459, 'LOC'), (6517, 6535, 'PER'), (6608,
6620, 'PER'), (6867, 6895, 'MISC'), (6960, 6973, 'PER'), (6975, 6991, 'PER'), (6993,
7008, 'PER'), (7011, 7029, 'PER'), (7047, 7060, 'PER'), (7103, 7122, 'ORG'), (7141,
7150, 'LOC'), (7235, 7249, 'PER'), (7254, 7273, 'ORG'), (7421, 7427, 'LOC'), (7470,
7482, 'PER'), (7485, 7502, 'LOC'), (7592, 7598, 'LOC'), (7786, 7798, 'PER'), (7912,
7926, 'LOC'), (7960, 7974, 'PER'), (8056, 8063, 'PER'), (8065, 8083, 'PER'), (8176,
8185, 'LOC'), (8200, 8214, 'PER'), (8288, 8299, 'ORG'), (8374, 8388, 'PER'), (8484,

8494, 'PER'), (8497, 8510, 'LOC'), (8560, 8572, 'PER'), (8630, 8637, 'PER'), (8641,
8655, 'PER'), (8670, 8684, 'PER'), (8716, 8739, 'PER'), (8753, 8759, 'PER'), (8778,
8789, 'ORG'), (8844, 8853, 'LOC'), (8958, 8970, 'PER'), (9051, 9057, 'PER'), (9111,
9131, 'ORG'), (9313, 9319, 'PER'), (9349, 9361, 'PER'), (9382, 9396, 'LOC'), (9444,
9470, 'ORG'), (9474, 9489, 'LOC'), (9490, 9496, 'LOC'), (9525, 9556, 'LOC'), (9602,
9650, 'ORG'), (9738, 9751, 'PER'), (9790, 9808, 'PER'), (9837, 9851, 'LOC'), (9870,
9893, 'PER'), (10055, 10075, 'PER'), (10106, 10107, 'ORG'), (10132, 10144, 'PER'),
(10173, 10187, 'LOC'), (10195, 10211, 'LOC'), (10277, 10300, 'PER'), (10302, 10308,
'PER'), (10309, 10330, 'PER'), (10332, 10344, 'PER'), (10347, 10370, 'PER'), (10442,
10448, 'LOC'), (10449, 10453, 'LOC'), (10491, 10515, 'LOC'), (10516, 10524, 'LOC'),
(10541, 10564, 'ORG'), (10587, 10592, 'LOC'), (10598, 10614, 'LOC'), (10641, 10653,
'PER'), (10655, 10659, 'PER'), (10660, 10678, 'PER'), (10681, 10708, 'PER'), (10749,
10769, 'PER'), (10959, 10971, 'PER'), (11155, 11177, 'MISC'), (11177, 11178, 'MISC'),
(11241, 11252, 'ORG'), (11268, 11280, 'PER'), (11282, 11308, 'PER'), (11311, 11334,
'PER'), (11373, 11385, 'PER'), (11388, 11401, 'PER'), (11679, 11691, 'LOC'), (12094,
12101, 'LOC'), (12103, 12123, 'PER'), (12144, 12162, 'PER'), (12190, 12210, 'MISC'),
(12473, 12485, 'PER'), (12527, 12544, 'ORG'), (12547, 12571, 'PER'), (12580, 12583,
'PER'), (12619, 12630, 'ORG'), (12744, 12750, 'PER'), (13051, 13064, 'MISC'), (13129,
13140, 'ORG'), (13195, 13208, 'PER'), (13226, 13235, 'LOC'), (13260, 13286, 'ORG'),
(13288, 13291, 'PER'), (13359, 13375, 'LOC'), (13390, 13402, 'PER'), (13466, 13472,
'PER'), (13510, 13523, 'PER'), (13638, 13664, 'ORG'), (13666, 13669, 'ORG'), (13697,
13716, 'PER'), (13730, 13737, 'MISC'), (13740, 13746, 'PER'), (13770, 13788, 'PER'),
(13826, 13837, 'ORG'), (13857, 13869, 'PER'), (13905, 13924, 'LOC'), (13929, 13938,
'LOC'), (13967, 13980, 'PER'), (14104, 14113, 'LOC'), (14138, 14149, 'ORG'), (14172,
14178, 'PER'), (14369, 14375, 'PER'), (14415, 14445, 'ORG'), (14447, 14450, 'ORG'),
(14457, 14464, 'ORG'), (14465, 14485, 'LOC'), (14493, 14520, 'ORG'), (14522, 14525,
'ORG'), (14806, 14812, 'PER'), (14919, 14925, 'MISC'), (14935, 14956, 'PER'), (14957,
14965, 'LOC'), (14983, 14988, 'PER'), (15068, 15077, 'LOC'), (15093, 15096, 'ORG'),
(15106, 15125, 'PER'), (15154, 15157, 'MISC'), (15193, 15205, 'PER'), (15225, 15228,
'PER'), (15264, 15269, 'LOC'), (15340, 15350, 'LOC'), (15426, 15434, 'LOC'), (15437,
15444, 'PER'), (15449, 15462, 'PER'), (15486, 15491, 'LOC'), (15567, 15577, 'LOC'),
(15713, 15720, 'PER'), (15770, 15783, 'PER'), (15796, 15812, 'MISC'), (15833, 15845,
'PER'), (15884, 15921, 'ORG'), (15942, 15950, 'LOC'), (16118, 16127, 'LOC'), (16275,
16278, 'PER'), (16329, 16342, 'PER'), (16344, 16356, 'PER'), (16371, 16393, 'MISC'),
(16447, 16469, 'LOC'), (16497, 16500, 'PER'), (16545, 16552, 'PER'), (16553, 16559,
'LOC'), (16608, 16620, 'PER'), (16665, 16671, 'PER'), (16693, 16699, 'PER'), (16712,
16726, 'PER'), (16771, 16774, 'MISC'), (16779, 16782, 'PER'), (16786, 16798, 'LOC'),
(16903, 16923, 'PER'), (16926, 16940, 'PER'), (17004, 17031, 'LOC'), (17036, 17039,
'ORG'), (17043, 17056, 'PER'), (17061, 17064, 'PER'), (17087, 17101, 'PER'), (17131,
17134, 'MISC'), (17137, 17158, 'MISC'), (17187, 17193, 'PER'), (17344, 17371, 'LOC'),
(17506, 17518, 'LOC'), (17533, 17536, 'PER'), (17617, 17620, 'PER'), (17624, 17630,
'PER'), (17688, 17708, 'PER'), (17776, 17785, 'LOC'), (17787, 17800, 'PER'), (17822,
17828, 'ORG'), (17968, 17977, 'LOC'), (17979, 17993, 'PER'), (18059, 18072, 'PER'),
(18092, 18099, 'PER'), (18101, 18120, 'PER'), (18149, 18169, 'ORG'), (18213, 18225,
'PER'), (18270, 18287, 'PER'), (18347, 18354, 'PER'), (18357, 18363, 'LOC'), (18421,
18428, 'PER'), (18430, 18437, 'LOC'), (18438, 18443, 'LOC'), (18550, 18555, 'PER'),
(18630, 18639, 'LOC'), (18843, 18855, 'PER'), (19084, 19105, 'LOC'), (19238, 19245,
'PER'), (19289, 19301, 'PER'), (19630, 19639, 'LOC'), (19684, 19700, 'PER'), (19773,
19785, 'PER'), (19956, 19962, 'PER'), (20139, 20146, 'PER'), (20186, 20189, 'PER'),
(20301, 20315, 'PER'), (20332, 20339, 'PER'), (20352, 20366, 'PER'), (20463, 20470,
'PER'), (20557, 20576, 'PER'), (20577, 20591, 'LOC'), (20593, 20613, 'PER'), (20616,
20642, 'PER'), (20667, 20679, 'PER'), (20728, 20734, 'LOC'), (20738, 20758, 'PER'),
(20760, 20777, 'PER'), (20779, 20802, 'PER'), (20804, 20818, 'PER'), (20821, 20835,
'PER'), (21017, 21029, 'PER'), (21091, 21105, 'PER'), (21171, 21179, 'ORG'), (21391,
21398, 'PER'), (21428, 21436, 'LOC'), (21470, 21479, 'LOC'), (21499, 21515, 'PER'),
(21531, 21537, 'ORG'), (21612, 21624, 'PER'), (21627, 21638, 'PER'), (21654, 21668,

'LOC'), (21725, 21740, 'PER'), (21743, 21759, 'PER'), (21831, 21854, 'MISC'), (21983,
21990, 'PER'), (22071, 22088, 'MISC'), (22095, 22099, 'MISC'), (22355, 22364, 'LOC'),
(22386, 22400, 'LOC'), (22420, 22429, 'LOC'), (22454, 22463, 'LOC'), (22518, 22530,
'PER'), (22551, 22554, 'PER'), (22558, 22564, 'LOC'), (22567, 22576, 'LOC'), (22685,
22697, 'PER'), (22764, 22773, 'LOC'), (22848, 22853, 'LOC'), (22903, 22915, 'PER'),
(22955, 22961, 'PER'), (23042, 23056, 'LOC'), (23124, 23127, 'PER'), (23170, 23173,
'ORG'), (23179, 23182, 'ORG'), (23218, 23227, 'LOC'), (23303, 23310, 'PER'), (23312,
23325, 'PER'), (23327, 23347, 'PER'), (23402, 23414, 'PER'), (23440, 23446, 'LOC'),
(23764, 23769, 'LOC'), (23771, 23786, 'PER'), (23808, 23830, 'LOC'), (23881, 23893,
'PER'), (23896, 23909, 'PER'), (23984, 23998, 'PER'), (24124, 24136, 'ORG'), (24183,
24191, 'ORG'), (24196, 24199, 'PER'), (24251, 24276, 'PER'), (24280, 24285, 'LOC'),
(24293, 24309, 'PER'), (24313, 24322, 'LOC'), (24392, 24402, 'PER'), (24405, 24412,
'PER'), (24470, 24477, 'PER'), (24548, 24561, 'PER'), (24564, 24576, 'PER'), (24627,
24630, 'PER'), (24730, 24737, 'PER'), (24865, 24881, 'PER'), (24904, 24916, 'PER'),
(24918, 24938, 'PER'), (24941, 24966, 'PER'), (25021, 25036, 'PER'), (25284, 25296,
'PER'), (25390, 25417, 'ORG'), (25419, 25422, 'ORG'), (25623, 25632, 'LOC'), (25635,
25647, 'LOC'), (25667, 25691, 'PER'), (25694, 25700, 'LOC'), (25701, 25709, 'LOC'),
(25796, 25804, 'ORG'), (25925, 25934, 'ORG'), (26044, 26051, 'LOC'), (26069, 26086,
'MISC'), (26093, 26097, 'MISC'), (26243, 26252, 'MISC'), (26390, 26406, 'MISC'),
(26457, 26469, 'PER'), (26471, 26487, 'PER'), (26489, 26494, 'PER'), (26495, 26503,
'PER'), (26505, 26515, 'PER'), (26518, 26542, 'PER'), (26702, 26709, 'LOC'), (26731,
26736, 'PER'), (26753, 26785, 'PER'), (26787, 26790, 'LOC'), (26820, 26825, 'PER'),
(26869, 26881, 'PER'), (26924, 26930, 'LOC'), (27118, 27132, 'PER'), (27140, 27152,
'PER'), (27165, 27173, 'LOC'), (27193, 27199, 'PER'), (27200, 27218, 'PER'), (27376,
27382, 'MISC'), (27392, 27406, 'PER'), (27433, 27442, 'LOC'), (27482, 27487, 'PER'),
(27496, 27508, 'PER'), (27585, 27592, 'LOC'), (27595, 27608, 'PER'), (27755, 27767,
'PER'), (27847, 27871, 'PER'), (27992, 27995, 'LOC'), (28029, 28036, 'LOC'), (28456,
28474, 'ORG'), (28513, 28530, 'MISC'), (28537, 28541, 'MISC'), (28556, 28565, 'LOC'),
(28676, 28681, 'PER'), (28950, 28962, 'PER'), (29004, 29025, 'PER'), (29041, 29047,
'MISC'), (29049, 29055, 'LOC'), (29092, 29104, 'ORG'), (29206, 29218, 'PER'), (29471,
29486, 'ORG'), (29492, 29520, 'MISC'), (29550, 29563, 'PER'), (29566, 29578, 'PER'),
(29639, 29648, 'LOC'), (29733, 29739, 'PER'), (29889, 29902, 'PER'), (29906, 29913,
'LOC'), (30230, 30236, 'ORG'), (30304, 30307, 'LOC'), (30361, 30378, 'MISC'), (30385,
30389, 'MISC'), (30409, 30432, 'MISC'), (30457, 30466, 'LOC'), (30467, 30475, 'LOC'),
(30576, 30588, 'ORG'), (30763, 30764, 'ORG'), (30797, 30813, 'PER'), (30814, 30822,
'LOC'), (30840, 30858, 'MISC'), (30887, 30896, 'LOC'), (30922, 30934, 'PER'), (30952,
30956, 'MISC'), (31120, 31129, 'MISC'), (31163, 31176, 'PER'), (31248, 31260, 'ORG'),
(31270, 31276, 'PER'), (31444, 31453, 'LOC'), (31579, 31588, 'LOC'), (31657, 31670,
'PER'), (31711, 31720, 'LOC'), (31722, 31735, 'PER'), (31738, 31750, 'PER'), (31879,
31888, 'LOC'), (32055, 32068, 'PER'), (32196, 32205, 'LOC'), (32215, 32228, 'PER'),
(32268, 32273, 'PER'), (32276, 32283, 'PER'), (32308, 32314, 'PER'), (32370, 32400,
'ORG'), (32430, 32446, 'PER'), (32456, 32469, 'PER'), (32603, 32608, 'PER'), (32611,
32616, 'PER'), (32696, 32709, 'PER'), (32806, 32829, 'PER'), (32831, 32839, 'ORG'),
(32852, 32869, 'PER'), (32871, 32878, 'ORG'), (32893, 32915, 'PER'), (32994, 33007,
'PER'), (33092, 33104, 'PER'), (33136, 33150, 'LOC'), (33255, 33264, 'PER'), (33383,
33396, 'PER'), (33499, 33508, 'PER'), (33568, 33572, 'MISC'), (33574, 33580, 'PER'),
(33655, 33668, 'PER'), (33698, 33707, 'LOC'), (33746, 33755, 'PER'), (33768, 33780,
'PER'), (33821, 33824, 'LOC'), (33914, 33926, 'ORG'), (33930, 33938, 'ORG'), (33948,
33969, 'PER'), (33982, 33994, 'LOC'), (33998, 34004, 'MISC'), (34016, 34041, 'LOC'),
(34054, 34081, 'ORG'), (34094, 34133, 'PER'), (34146, 34177, 'MISC'), (34187, 34201,
'PER'), (34215, 34228, 'PER'), (34383, 34395, 'PER'), (34711, 34728, 'MISC'), (34736,
34741, 'MISC'), (34899, 34912, 'PER'), (34958, 34963, 'MISC'), (35045, 35058, 'PER'),
(35138, 35147, 'LOC'), (35193, 35205, 'PER'), (35230, 35246, 'ORG'), (35253, 35261,
'LOC'), (35306, 35314, 'LOC'), (35316, 35322, 'LOC'), (35430, 35443, 'LOC'), (35480,
35492, 'LOC'), (35502, 35526, 'PER'), (35550, 35559, 'LOC'), (35566, 35583, 'MISC'),
(35591, 35596, 'MISC'), (35693, 35702, 'LOC'), (35717, 35730, 'PER'), (35984, 36005,

'LOC'), (36037, 36049, 'PER'), (36174, 36183, 'LOC'), (36290, 36320, 'ORG'), (36322,
36325, 'ORG'), (36330, 36334, 'MISC'), (36445, 36448, 'ORG'), (36563, 36569, 'PER'),
(36584, 36593, 'PER'), (36676, 36685, 'LOC'), (36757, 36760, 'LOC'), (36840, 36843,
'ORG'), (36845, 36857, 'PER'), (36873, 36878, 'PER'), (36996, 37027, 'ORG'), (37029,
37032, 'ORG'), (37449, 37452, 'ORG'), (37652, 37664, 'ORG'), (37778, 37790, 'PER'),
(37971, 37976, 'PER'), (37982, 37985, 'LOC'), (37988, 37994, 'LOC'), (38110, 38131,
'PER'), (38319, 38322, 'ORG'), (38418, 38430, 'PER'), (38491, 38495, 'PER'), (38568,
38582, 'PER'), (38679, 38691, 'PER'), (38745, 38779, 'ORG'), (38782, 38795, 'LOC'),
(38842, 38848, 'LOC'), (38962, 38988, 'PER'), (39033, 39055, 'PER'), (39129, 39132,
'ORG'), (39264, 39276, 'PER'), (39328, 39329, 'MISC'), (39339, 39349, 'PER'), (39363,
39364, 'PER'), (39388, 39407, 'ORG'), (39514, 39539, 'PER'), (39542, 39555, 'PER'),
(39565, 39577, 'PER'), (39599, 39600, 'MISC')]}
{'entities': [(3, 35, 'PER'), (46, 52, 'LOC'), (93, 117, 'PER'), (123, 150, 'PER'),
(162, 175, 'PER'), (254, 264, 'LOC'), (375, 385, 'PER'), (465, 492, 'LOC'), (494, 497,
'ORG'), (510, 524, 'PER'), (556, 591, 'LOC'), (593, 597, 'LOC'), (638, 682, 'ORG'),
(684, 688, 'ORG'), (734, 769, 'ORG'), (771, 777, 'LOC'), (877, 907, 'LOC'), (1005,
1040, 'PER'), (1045, 1051, 'LOC'), (1073, 1091, 'PER'), (1158, 1162, 'ORG'), (1195,
1208, 'PER'), (1268, 1278, 'LOC'), (1351, 1380, 'ORG'), (1382, 1385, 'ORG'), (1428,
1464, 'ORG'), (1577, 1580, 'ORG'), (1705, 1723, 'ORG'), (1728, 1741, 'LOC'), (1758,
1778, 'ORG'), (1780, 1789, 'ORG'), (1792, 1800, 'LOC'), (1884, 1887, 'ORG'), (1891,
1901, 'ORG'), (2022, 2036, 'PER'), (2070, 2084, 'LOC'), (2092, 2098, 'LOC'), (2146,
2156, 'LOC'), (2281, 2302, 'PER'), (2359, 2366, 'LOC'), (2391, 2404, 'PER'), (2434,
2448, 'PER'), (2515, 2529, 'MISC'), (2676, 2704, 'ORG'), (2706, 2709, 'LOC'), (2713,
2723, 'LOC'), (2928, 2935, 'LOC'), (3031, 3049, 'LOC'), (3160, 3171, 'ORG'), (3237,
3242, 'PER'), (3381, 3387, 'MISC'), (3443, 3457, 'PER'), (3924, 3941, 'LOC'), (3944,
3951, 'LOC'), (3954, 3963, 'LOC'), (4011, 4014, 'ORG'), (4283, 4335, 'MISC'), (4337,
4341, 'ORG'), (4432, 4441, 'LOC'), (4498, 4502, 'MISC'), (4524, 4539, 'LOC'), (4547,
4554, 'ORG'), (4556, 4560, 'ORG'), (4561, 4564, 'ORG'), (4692, 4710, 'ORG'), (4714,
4734, 'LOC'), (4736, 4745, 'PER'), (4747, 4756, 'PER'), (4759, 4777, 'PER'), (4841,
4847, 'ORG'), (4919, 4938, 'ORG'), (4993, 5018, 'PER'), (5032, 5046, 'PER'), (5142,
5153, 'LOC'), (5156, 5166, 'LOC'), (5219, 5226, 'LOC'), (5229, 5239, 'LOC'), (5259,
5280, 'LOC'), (5408, 5422, 'PER'), (5450, 5459, 'LOC'), (5475, 5503, 'MISC'), (5756,
5776, 'PER'), (5795, 5810, 'PER'), (5996, 5999, 'ORG'), (6048, 6055, 'ORG'), (6057,
6070, 'PER'), (6091, 6097, 'PER'), (6139, 6142, 'ORG'), (6183, 6193, 'LOC'), (6199,
6227, 'ORG'), (6255, 6269, 'PER'), (6287, 6298, 'MISC'), (6298, 6310, 'LOC'), (6354,
6370, 'PER'), (6376, 6389, 'ORG'), (6391, 6396, 'MISC'), (6495, 6502, 'PER'), (6527,
6531, 'PER'), (6534, 6540, 'PER'), (6654, 6660, 'PER'), (6785, 6795, 'LOC'), (6954,
6968, 'PER'), (6997, 7022, 'ORG'), (7024, 7027, 'PER'), (7072, 7086, 'PER'), (7131,
7160, 'ORG'), (7162, 7165, 'ORG'), (7409, 7419, 'MISC'), (7442, 7461, 'PER'), (7589,
7612, 'MISC'), (7698, 7727, 'ORG'), (7729, 7732, 'ORG'), (7767, 7801, 'MISC'), (7809,
7831, 'ORG'), (7855, 7865, 'LOC'), (8090, 8100, 'LOC'), (8102, 8109, 'PER'), (8111,
8117, 'PER'), (8257, 8266, 'LOC'), (8268, 8293, 'PER'), (8341, 8345, 'PER'), (8412,
8422, 'LOC'), (8425, 8439, 'MISC'), (8504, 8507, 'ORG'), (8519, 8547, 'ORG'), (8549,
8553, 'ORG'), (8555, 8574, 'PER'), (8576, 8591, 'ORG'), (8593, 8626, 'ORG'), (8628,
8647, 'ORG'), (8649, 8663, 'ORG'), (8665, 8688, 'ORG'), (8690, 8722, 'ORG'), (8725,
8751, 'ORG'), (8754, 8785, 'ORG'), (8856, 8859, 'ORG'), (8902, 8912, 'LOC'), (8914,
8920, 'LOC'), (9025, 9044, 'PER'), (9049, 9053, 'ORG'), (9119, 9133, 'PER'), (9339,
9349, 'LOC'), (9440, 9465, 'MISC'), (9470, 9483, 'ORG'), (9485, 9490, 'ORG'), (9802,
9836, 'LOC'), (9838, 9841, 'ORG'), (9862, 9867, 'MISC'), (9889, 9903, 'PER'), (10277,
10286, 'LOC'), (10306, 10320, 'PER'), (10347, 10357, 'LOC'), (10376, 10390, 'PER'),
(10402, 10405, 'ORG'), (10438, 10444, 'LOC'), (10471, 10483, 'PER'), (10502, 10505,
'ORG'), (10574, 10583, 'LOC'), (10585, 10591, 'LOC'), (10698, 10709, 'PER'), (10714,
10753, 'ORG'), (10755, 10759, 'ORG'), (10797, 10811, 'PER'), (10887, 10901, 'PER'),
(10935, 10941, 'LOC'), (10943, 10945, 'LOC'), (11034, 11042, 'LOC'), (11042, 11049,
'LOC'), (11053, 11060, 'LOC'), (11147, 11165, 'MISC'), (11182, 11197, 'LOC'), (11287,
11296, 'ORG'), (11324, 11331, 'PER'), (11333, 11339, 'LOC'), (11437, 11450, 'LOC'),

(11465, 11484, 'PER'), (11636, 11642, 'LOC'), (11650, 11656, 'PER'), (11658, 11663, 'PER'), (11740, 11742, 'PER'), (11799, 11806, 'PER'), (11808, 11814, 'PER'), (11933, 11951, 'ORG'), (11991, 12006, 'LOC'), (12022, 12051, 'PER'), (12078, 12086, 'LOC'), (12091, 12111, 'LOC'), (12158, 12162, 'PER'), (12164, 12172, 'PER'), (12186, 12219, 'MISC'), (12223, 12226, 'ORG'), (12230, 12240, 'LOC'), (12337, 12348, 'ORG'), (12414, 12418, 'PER'), (12478, 12484, 'PER'), (12510, 12518, 'MISC'), (12621, 12630, 'LOC')]}
{'entities': [(3, 25, 'PER'), (36, 44, 'LOC'), (82, 113, 'ORG'), (134, 140, 'PER'), (224, 240, 'PER'), (265, 281, 'PER'), (318, 326, 'LOC'), (405, 419, 'PER'), (433, 447, 'LOC')]}
{'entities': [(3, 31, 'PER'), (42, 52, 'LOC'), (89, 115, 'PER'), (121, 145, 'PER'), (169, 190, 'PER'), (193, 209, 'PER'), (214, 219, 'LOC'), (247, 268, 'LOC'), (271, 303, 'PER'), (382, 393, 'ORG'), (487, 506, 'ORG'), (525, 529, 'LOC'), (544, 570, 'ORG'), (572, 575, 'PER'), (764, 806, 'ORG'), (811, 847, 'ORG'), (1080, 1126, 'ORG'), (1128, 1132, 'ORG'), (1389, 1403, 'LOC'), (1589, 1610, 'LOC'), (1614, 1634, 'LOC'), (1666, 1686, 'LOC'), (1728, 1737, 'LOC'), (1792, 1796, 'LOC'), (1812, 1842, 'MISC'), (1857, 1860, 'PER'), (1864, 1879, 'ORG'), (1881, 1883, 'PER'), (1888, 1914, 'ORG'), (1916, 1919, 'ORG'), (1925, 1952, 'ORG'), (1954, 1957, 'ORG'), (2028, 2048, 'LOC'), (2099, 2102, 'PER'), (2160, 2190, 'ORG'), (2231, 2238, 'ORG'), (2259, 2262, 'ORG'), (2275, 2307, 'PER'), (2381, 2384, 'PER'), (2434, 2445, 'ORG'), (2557, 2563, 'ORG'), (2682, 2713, 'PER'), (2714, 2736, 'LOC'), (2780, 2784, 'PER'), (2785, 2792, 'PER'), (2893, 2910, 'MISC'), (2986, 3013, 'MISC'), (3015, 3020, 'PER'), (3074, 3080, 'LOC'), (3086, 3090, 'LOC'), (3195, 3201, 'LOC'), (3226, 3248, 'PER'), (3284, 3315, 'ORG'), (3344, 3351, 'LOC'), (3353, 3356, 'LOC'), (3552, 3573, 'PER'), (3644, 3661, 'MISC'), (3668, 3672, 'MISC'), (3733, 3776, 'ORG'), (3781, 3787, 'LOC'), (3788, 3832, 'LOC'), (3837, 3859, 'MISC'), (3862, 3879, 'PER'), (3882, 3903, 'PER'), (3909, 3929, 'ORG'), (3933, 3982, 'LOC'), (4020, 4039, 'LOC'), (4044, 4063, 'LOC'), (4065, 4067, 'LOC'), (4101, 4108, 'LOC'), (4112, 4143, 'PER'), (4147, 4161, 'LOC'), (4222, 4232, 'LOC'), (4233, 4254, 'ORG'), (4276, 4295, 'ORG'), (4296, 4306, 'ORG'), (4308, 4311, 'ORG'), (4363, 4399, 'ORG'), (4404, 4437, 'ORG'), (4443, 4482, 'ORG'), (4484, 4494, 'PER'), (4509, 4523, 'LOC'), (4528, 4533, 'LOC')]}
{'entities': [(3, 24, 'PER'), (35, 45, 'LOC'), (47, 49, 'LOC'), (102, 121, 'PER'), (127, 146, 'LOC'), (170, 178, 'PER'), (181, 190, 'LOC'), (195, 209, 'LOC'), (217, 233, 'LOC'), (270, 298, 'LOC'), (331, 359, 'LOC'), (360, 366, 'LOC'), (407, 427, 'LOC'), (431, 459, 'LOC'), (461, 464, 'ORG'), (473, 502, 'LOC'), (503, 509, 'LOC'), (511, 515, 'LOC'), (646, 669, 'ORG'), (673, 685, 'LOC'), (800, 822, 'LOC'), (851, 865, 'ORG'), (879, 882, 'ORG'), (982, 992, 'LOC'), (1009, 1033, 'ORG'), (1035, 1055, 'PER'), (1080, 1083, 'ORG'), (1087, 1101, 'LOC'), (1211, 1223, 'LOC'), (1281, 1291, 'LOC'), (1302, 1319, 'MISC'), (1360, 1374, 'PER'), (1414, 1431, 'ORG'), (1432, 1440, 'ORG'), (1442, 1445, 'PER'), (1489, 1495, 'ORG'), (1594, 1597, 'ORG'), (1601, 1607, 'LOC'), (1715, 1718, 'ORG'), (1722, 1752, 'ORG'), (1754, 1757, 'ORG'), (1762, 1776, 'ORG'), (1777, 1789, 'ORG'), (1791, 1794, 'ORG'), (1800, 1819, 'ORG'), (1821, 1823, 'ORG'), (1863, 1869, 'LOC'), (1873, 1902, 'ORG'), (1904, 1907, 'ORG'), (1993, 2013, 'PER'), (2061, 2067, 'ORG'), (2104, 2112, 'LOC'), (2114, 2124, 'LOC'), (2457, 2466, 'PER'), (2723, 2733, 'PER'), (2760, 2778, 'LOC'), (2782, 2798, 'LOC'), (2845, 2859, 'LOC'), (2901, 2919, 'PER'), (2950, 2968, 'MISC'), (2974, 2975, 'MISC'), (2985, 2993, 'LOC'), (3018, 3019, 'ORG'), (3023, 3042, 'MISC')]}
{'entities': [(3, 31, 'PER'), (42, 49, 'LOC'), (51, 53, 'LOC'), (91, 116, 'PER'), (122, 142, 'PER'), (144, 153, 'LOC'), (229, 238, 'LOC'), (279, 310, 'ORG'), (319, 368, 'ORG'), (370, 373, 'LOC'), (418, 447, 'ORG'), (451, 456, 'PER'), (482, 531, 'ORG'), (629, 637, 'LOC'), (640, 644, 'LOC'), (645, 655, 'LOC'), (659, 678, 'LOC'), (752, 785, 'ORG'), (808, 824, 'LOC'), (906, 933, 'LOC'), (936, 939, 'LOC'), (955, 973, 'ORG'), (996, 1013, 'MISC'), (1051, 1067, 'PER'), (1116, 1128, 'LOC'), (1155, 1161, 'LOC'), (1164, 1178, 'LOC'), (1237, 1262, 'PER'), (1305, 1327, 'ORG'), (1329, 1333, 'ORG'), (1387, 1403, 'LOC'), (1441, 1464, 'LOC'), (1478, 1523, 'ORG'), (1567, 1576, 'LOC'), (1590, 1626, 'PER')]}
{'entities': [(3, 34, 'PER'), (45, 60, 'LOC'), (62, 64, 'LOC'), (117, 141, 'PER'),

(142, 148, 'PER'), (154, 179, 'PER'), (228, 254, 'PER'), (258, 278, 'PER'), (285, 303,
'MISC'), (316, 338, 'PER'), (360, 383, 'PER'), (406, 414, 'LOC'), (416, 418, 'LOC'),
(447, 465, 'PER'), (497, 509, 'LOC'), (680, 697, 'MISC'), (733, 751, 'PER'), (753,
769, 'PER'), (774, 785, 'PER'), (787, 810, 'PER'), (812, 827, 'PER'), (830, 845,
'PER'), (861, 877, 'PER'), (883, 901, 'PER'), (903, 910, 'PER'), (911, 934, 'PER'),
(952, 959, 'LOC'), (973, 1005, 'ORG'), (1108, 1122, 'LOC'), (1175, 1181, 'LOC'),
(1192, 1208, 'PER'), (1292, 1329, 'LOC'), (1330, 1335, 'PER'), (1362, 1377, 'PER'),
(1459, 1465, 'LOC'), (1468, 1472, 'LOC'), (1473, 1478, 'LOC'), (1506, 1520, 'LOC'),
(1524, 1550, 'LOC'), (1870, 1881, 'PER'), (2119, 2163, 'MISC'), (2344, 2358, 'LOC'),
(2643, 2657, 'PER'), (2715, 2720, 'LOC'), (2921, 2936, 'PER'), (2940, 2945, 'LOC'),
(3010, 3019, 'PER'), (3040, 3048, 'LOC'), (3050, 3061, 'PER'), (3250, 3255, 'LOC'),
(3280, 3292, 'LOC'), (3419, 3434, 'PER'), (3459, 3486, 'ORG'), (3488, 3491, 'ORG'),
(3551, 3573, 'PER'), (3598, 3638, 'MISC'), (3845, 3857, 'PER'), (3859, 3876, 'PER'),
(3878, 3896, 'PER'), (4013, 4022, 'PER'), (4023, 4029, 'LOC'), (4140, 4156, 'PER'),
(4184, 4195, 'PER'), (4222, 4225, 'ORG'), (4320, 4343, 'PER'), (4360, 4384, 'ORG'),
(4386, 4416, 'PER'), (4507, 4512, 'PER'), (4513, 4522, 'PER'), (4634, 4647, 'PER'),
(4648, 4656, 'LOC'), (5007, 5023, 'PER'), (5065, 5068, 'ORG'), (5185, 5205, 'LOC'),
(5344, 5350, 'PER'), (5351, 5362, 'PER'), (5364, 5386, 'PER'), (5388, 5421, 'PER'),
(5423, 5456, 'PER'), (5458, 5478, 'PER'), (5480, 5494, 'PER'), (5495, 5506, 'LOC'),
(5509, 5539, 'PER'), (5592, 5598, 'LOC'), (5706, 5721, 'PER'), (5880, 5901, 'PER'),
(6309, 6315, 'LOC'), (6337, 6343, 'LOC'), (6376, 6384, 'LOC'), (6792, 6804, 'PER'),
(6827, 6836, 'LOC'), (6845, 6860, 'PER'), (7093, 7101, 'PER'), (7256, 7265, 'LOC'),
(7308, 7316, 'PER'), (7414, 7429, 'PER'), (7444, 7459, 'PER'), (7492, 7507, 'MISC'),
(7793, 7826, 'PER'), (7898, 7915, 'MISC'), (7980, 7995, 'PER'), (8047, 8052, 'LOC'),
(8140, 8155, 'PER'), (8517, 8522, 'LOC'), (8693, 8705, 'PER'), (8716, 8723, 'MISC'),
(8724, 8735, 'LOC'), (9210, 9216, 'LOC'), (9595, 9604, 'PER'), (9649, 9652, 'MISC'),
(9727, 9736, 'MISC'), (9771, 9783, 'ORG'), (9806, 9815, 'PER'), (9841, 9850, 'ORG'),
(9856, 9879, 'MISC'), (9883, 9888, 'LOC'), (9990, 10006, 'PER'), (10528, 10538,
'LOC'), (10588, 10597, 'PER'), (10601, 10621, 'LOC'), (10623, 10637, 'PER'), (10658,
10661, 'ORG'), (10687, 10692, 'LOC'), (10760, 10767, 'PER'), (10768, 10774, 'PER'),
(10827, 10837, 'PER'), (10841, 10866, 'PER'), (10881, 10887, 'LOC'), (10942, 10951,
'ORG'), (10970, 10985, 'PER'), (10987, 11003, 'PER'), (11220, 11226, 'LOC'), (11484,
11485, 'MISC'), (11567, 11581, 'PER'), (11585, 11590, 'LOC'), (11776, 11807, 'PER'),
(11899, 11906, 'PER'), (11907, 11913, 'PER'), (11944, 11959, 'PER'), (11991, 12000,
'LOC'), (12069, 12085, 'PER'), (12104, 12118, 'PER'), (12180, 12194, 'LOC'), (12197,
12225, 'LOC'), (12233, 12269, 'LOC'), (12383, 12399, 'PER'), (12495, 12501, 'LOC'),
(12530, 12545, 'PER'), (12548, 12567, 'PER'), (12576, 12599, 'MISC'), (12602, 12618,
'PER'), (12653, 12668, 'PER'), (13225, 13240, 'PER'), (13284, 13300, 'PER'), (13388,
13413, 'ORG'), (13632, 13646, 'LOC'), (13649, 13659, 'LOC'), (13691, 13696, 'LOC'),
(13698, 13704, 'LOC'), (13707, 13714, 'LOC'), (13753, 13767, 'LOC'), (13886, 13901,
'ORG'), (13916, 13928, 'LOC'), (14038, 14054, 'PER'), (14301, 14316, 'ORG'), (14368,
14385, 'MISC'), (14391, 14395, 'LOC'), (14416, 14431, 'ORG'), (14510, 14527, 'LOC'),
(14529, 14543, 'PER'), (14550, 14557, 'LOC'), (14559, 14570, 'LOC'), (14615, 14624,
'LOC'), (14678, 14684, 'PER'), (14685, 14692, 'PER'), (14702, 14717, 'PER'), (14807,
14819, 'LOC'), (14822, 14827, 'LOC'), (14828, 14834, 'LOC'), (14857, 14874, 'LOC'),
(14991, 15006, 'LOC'), (15327, 15342, 'PER'), (15354, 15369, 'PER'), (15387, 15396,
'LOC'), (15478, 15491, 'PER'), (15507, 15510, 'LOC'), (15511, 15516, 'PER'), (15532,
15546, 'PER'), (15601, 15616, 'PER'), (15865, 15883, 'LOC'), (15903, 15920, 'LOC'),
(15931, 15957, 'PER'), (16038, 16043, 'LOC'), (16048, 16065, 'LOC'), (16148, 16160,
'LOC'), (16166, 16180, 'LOC'), (16182, 16198, 'PER'), (16231, 16251, 'LOC'), (16253,
16286, 'PER'), (16297, 16311, 'PER'), (16349, 16355, 'PER'), (16369, 16402, 'ORG'),
(16404, 16407, 'ORG'), (16410, 16432, 'PER'), (16436, 16444, 'LOC'), (16497, 16500,
'ORG'), (16502, 16524, 'PER'), (16623, 16638, 'PER'), (16712, 16717, 'LOC'), (16761,
16779, 'PER'), (16783, 16797, 'PER'), (16864, 16878, 'PER'), (16937, 16950, 'PER'),
(16989, 17004, 'PER'), (17020, 17034, 'PER'), (17037, 17051, 'PER'), (17099, 17114,
'ORG'), (17351, 17365, 'PER'), (17369, 17384, 'PER'), (17447, 17461, 'LOC'), (17494,

17500, 'PER'), (17503, 17514, 'PER'), (17595, 17610, 'PER'), (17628, 17651, 'PER'),
(17665, 17675, 'PER'), (17677, 17705, 'PER'), (17707, 17719, 'PER'), (17720, 17727,
'PER'), (17730, 17744, 'PER'), (17760, 17768, 'LOC'), (17772, 17778, 'PER'), (17817,
17832, 'PER'), (18100, 18119, 'PER'), (18120, 18126, 'LOC'), (18128, 18148, 'PER'),
(18150, 18177, 'PER'), (18179, 18185, 'LOC'), (18186, 18192, 'LOC'), (18194, 18218,
'PER'), (18220, 18233, 'PER'), (18236, 18262, 'PER'), (18356, 18379, 'PER'), (18410,
18415, 'LOC'), (18615, 18638, 'PER'), (18794, 18808, 'PER'), (18918, 18933, 'PER'),
(18953, 18966, 'PER'), (19007, 19012, 'LOC'), (19013, 19019, 'LOC'), (19068, 19083,
'ORG'), (19163, 19178, 'PER'), (19196, 19219, 'PER'), (19238, 19253, 'PER'), (19258,
19268, 'LOC'), (19270, 19272, 'LOC'), (19277, 19291, 'PER'), (19296, 19301, 'LOC'),
(19324, 19336, 'LOC'), (19422, 19436, 'PER'), (19438, 19446, 'PER'), (19461, 19464,
'LOC'), (19476, 19484, 'PER'), (19528, 19542, 'PER'), (19634, 19642, 'PER'), (19644,
19651, 'PER'), (19654, 19665, 'PER'), (19688, 19702, 'PER'), (19706, 19721, 'PER'),
(19723, 19738, 'PER'), (19741, 19755, 'PER'), (19757, 19771, 'PER'), (19813, 19828,
'LOC'), (19893, 19910, 'LOC'), (19981, 19986, 'LOC'), (20000, 20007, 'LOC'), (20010,
20024, 'PER'), (20069, 20078, 'PER'), (20079, 20085, 'LOC'), (20100, 20111, 'PER'),
(20115, 20132, 'LOC'), (20157, 20158, 'LOC'), (20202, 20219, 'LOC'), (20282, 20290,
'LOC'), (20294, 20297, 'ORG'), (20304, 20322, 'PER'), (20328, 20334, 'PER'), (20337,
20351, 'PER'), (20393, 20405, 'LOC'), (20626, 20635, 'LOC'), (20690, 20707, 'LOC'),
(20747, 20756, 'LOC'), (20771, 20774, 'LOC'), (20795, 20810, 'PER'), (20825, 20830,
'LOC'), (20832, 20846, 'PER'), (20946, 20949, 'ORG'), (20989, 20994, 'LOC'), (21025,
21031, 'PER'), (21297, 21311, 'PER'), (21607, 21630, 'PER'), (21715, 21727, 'LOC'),
(21754, 21770, 'PER'), (21781, 21789, 'PER'), (21906, 21913, 'MISC'), (21937, 21951,
'PER'), (21962, 21978, 'PER'), (22012, 22026, 'PER'), (22057, 22062, 'LOC'), (22207,
22212, 'LOC'), (22217, 22234, 'LOC'), (22270, 22284, 'PER'), (22372, 22388, 'PER'),
(22415, 22429, 'PER'), (22446, 22460, 'PER'), (22523, 22534, 'PER'), (22570, 22584,
'PER'), (22683, 22689, 'PER'), (22714, 22719, 'LOC'), (22812, 22826, 'PER'), (22892,
22897, 'LOC'), (22900, 22915, 'PER'), (22989, 22996, 'PER'), (23026, 23040, 'PER'),
(23074, 23088, 'PER'), (23105, 23127, 'MISC'), (23225, 23236, 'PER'), (23240, 23246,
'LOC'), (23422, 23427, 'LOC'), (23533, 23547, 'PER'), (23599, 23614, 'PER'), (23662,
23668, 'PER'), (23671, 23685, 'PER'), (23940, 23954, 'LOC'), (24022, 24049, 'LOC'),
(24055, 24057, 'ORG'), (24143, 24156, 'ORG'), (24198, 24210, 'PER'), (24235, 24244,
'LOC'), (24324, 24340, 'PER'), (24363, 24385, 'LOC'), (24390, 24407, 'LOC'), (24544,
24561, 'LOC'), (24589, 24595, 'LOC'), (24600, 24609, 'LOC'), (24611, 24626, 'PER'),
(24792, 24798, 'PER'), (24891, 24898, 'PER'), (24955, 24973, 'MISC'), (25094, 25110,
'LOC'), (25116, 25132, 'MISC'), (25134, 25154, 'MISC'), (25234, 25240, 'PER'), (25272,
25310, 'MISC'), (25322, 25338, 'PER'), (25363, 25375, 'PER'), (25397, 25403, 'PER'),
(25454, 25471, 'LOC'), (25480, 25485, 'LOC'), (25493, 25500, 'LOC'), (25548, 25555,
'ORG'), (25674, 25688, 'LOC'), (25727, 25742, 'PER'), (25743, 25749, 'PER'), (25811,
25826, 'PER'), (25832, 25847, 'PER'), (25850, 25865, 'PER'), (25912, 25917, 'LOC'),
(25964, 25971, 'PER'), (26048, 26059, 'PER'), (26062, 26078, 'PER'), (26096, 26107,
'PER'), (26134, 26149, 'ORG'), (26214, 26220, 'LOC'), (26312, 26324, 'LOC'), (26342,
26357, 'MISC'), (26444, 26447, 'ORG'), (26518, 26533, 'PER'), (26676, 26690, 'PER'),
(26702, 26717, 'PER'), (26865, 26880, 'ORG'), (26887, 26901, 'PER'), (26904, 26920,
'PER'), (27007, 27015, 'LOC'), (27017, 27026, 'PER'), (27029, 27043, 'PER'), (27152,
27161, 'LOC'), (27383, 27396, 'PER'), (27430, 27442, 'MISC'), (27467, 27468, 'MISC'),
(27520, 27523, 'ORG'), (27555, 27570, 'PER'), (27678, 27687, 'LOC'), (27715, 27737,
'PER'), (27766, 27791, 'ORG'), (27860, 27865, 'LOC'), (27917, 27932, 'PER'), (28004,
28007, 'ORG'), (28096, 28105, 'PER'), (28244, 28252, 'PER'), (28376, 28385, 'PER'),
(28428, 28431, 'ORG'), (28568, 28582, 'PER'), (28596, 28625, 'MISC'), (28629, 28645,
'PER'), (28781, 28802, 'PER'), (28803, 28811, 'LOC'), (28894, 28906, 'PER'), (28908,
28920, 'PER'), (28923, 28936, 'LOC'), (28948, 28965, 'PER'), (28982, 29003, 'PER'),
(29062, 29071, 'LOC'), (29127, 29141, 'PER'), (29143, 29159, 'PER'), (29161, 29191,
'PER'), (29193, 29199, 'MISC'), (29213, 29220, 'LOC'), (29222, 29229, 'ORG'), (29240,
29255, 'PER'), (29277, 29281, 'PER'), (29282, 29290, 'LOC'), (29316, 29331, 'PER'),
(29355, 29372, 'MISC'), (29376, 29381, 'LOC'), (29448, 29464, 'PER'), (29535, 29541,

'PER'), (29543, 29557, 'PER'), (29560, 29564, 'PER'), (29565, 29573, 'LOC'), (29612,
29621, 'PER'), (29625, 29630, 'LOC'), (29668, 29671, 'ORG'), (29674, 29705, 'MISC'),
(29755, 29777, 'PER'), (29780, 29796, 'PER'), (30006, 30014, 'LOC'), (30061, 30070,
'PER'), (30073, 30097, 'PER'), (30099, 30107, 'PER'), (30110, 30121, 'PER'), (30123,
30134, 'PER'), (30138, 30155, 'LOC'), (30157, 30165, 'LOC'), (30203, 30215, 'PER'),
(30217, 30242, 'PER'), (30245, 30252, 'PER'), (30253, 30261, 'LOC'), (30278, 30294,
'PER'), (30301, 30306, 'LOC'), (30385, 30402, 'ORG'), (30537, 30544, 'LOC'), (30570,
30573, 'LOC'), (30592, 30608, 'PER'), (30624, 30637, 'PER'), (30651, 30667, 'PER'),
(30751, 30760, 'MISC'), (30783, 30789, 'LOC'), (30842, 30847, 'LOC'), (30894, 30897,
'MISC'), (30918, 30927, 'MISC'), (30929, 30934, 'PER'), (31053, 31056, 'LOC'), (31079,
31085, 'LOC'), (31150, 31166, 'PER'), (31197, 31218, 'LOC'), (31247, 31261, 'PER'),
(31274, 31289, 'PER'), (31309, 31315, 'PER'), (31327, 31343, 'LOC'), (31361, 31373,
'LOC'), (31383, 31390, 'LOC'), (31391, 31401, 'LOC'), (31404, 31423, 'MISC'), (31439,
31444, 'LOC'), (31524, 31538, 'LOC'), (31552, 31568, 'PER'), (31570, 31586, 'PER'),
(31589, 31601, 'PER'), (31837, 31843, 'LOC'), (31848, 31865, 'MISC'), (31967, 31975,
'PER'), (32673, 32692, 'MISC'), (32757, 32773, 'PER'), (32808, 32822, 'LOC'), (32947,
32962, 'PER'), (33143, 33165, 'PER'), (33195, 33209, 'PER'), (33318, 33331, 'PER'),
(33371, 33377, 'PER'), (33628, 33639, 'MISC'), (33657, 33673, 'PER'), (33712, 33726,
'PER'), (33753, 33761, 'PER'), (33765, 33772, 'PER'), (33802, 33808, 'PER'), (33839,
33857, 'MISC'), (33860, 33870, 'LOC'), (33959, 33965, 'LOC'), (33975, 33983, 'LOC'),
(34092, 34104, 'MISC'), (34162, 34180, 'MISC'), (34192, 34209, 'MISC'), (34324, 34339,
'PER'), (34382, 34398, 'LOC'), (34429, 34444, 'PER'), (34522, 34536, 'PER'), (34538,
34554, 'PER'), (34557, 34572, 'PER'), (34588, 34604, 'PER'), (34641, 34677, 'ORG'),
(34723, 34737, 'PER'), (34831, 34847, 'PER'), (34860, 34868, 'LOC'), (34871, 34884,
'LOC'), (35273, 35280, 'LOC'), (35282, 35290, 'LOC'), (35293, 35303, 'LOC'), (35335,
35356, 'ORG'), (35358, 35375, 'ORG'), (35442, 35472, 'LOC'), (35489, 35511, 'LOC'),
(35520, 35558, 'LOC'), (35742, 35748, 'LOC'), (36392, 36408, 'PER'), (37367, 37397,
'LOC'), (37699, 37706, 'PER'), (38031, 38040, 'PER'), (38906, 38918, 'LOC'), (39018,
39032, 'PER'), (39062, 39077, 'ORG'), (39278, 39291, 'PER'), (39352, 39358, 'ORG'),
(39474, 39490, 'PER'), (39653, 39669, 'PER'), (39673, 39695, 'LOC'), (39743, 39755,
'LOC'), (39786, 39804, 'MISC'), (39828, 39833, 'LOC'), (39837, 39844, 'LOC'), (39845,
39855, 'LOC'), (39857, 39863, 'LOC'), (39951, 39964, 'PER'), (39966, 39967, 'PER'),
(40164, 40179, 'PER'), (40210, 40224, 'PER'), (40271, 40287, 'PER'), (40303, 40311,
'LOC'), (40339, 40355, 'PER'), (40359, 40373, 'PER'), (40481, 40495, 'MISC'), (40535,
40538, 'ORG'), (40553, 40562, 'PER'), (40687, 40701, 'LOC'), (40759, 40773, 'PER'),
(40857, 40872, 'PER'), (40883, 40906, 'PER'), (41004, 41026, 'PER'), (41067, 41075,
'LOC'), (41093, 41106, 'ORG'), (41121, 41137, 'PER'), (41139, 41153, 'PER'), (41156,
41170, 'PER'), (41245, 41259, 'PER'), (41269, 41285, 'PER'), (41294, 41300, 'PER'),
(41341, 41349, 'LOC'), (41353, 41371, 'MISC'), (41373, 41389, 'PER'), (41430, 41444,
'PER'), (41460, 41466, 'PER'), (41493, 41499, 'PER'), (41517, 41526, 'PER'), (41527,
41533, 'LOC'), (41603, 41617, 'PER'), (41630, 41668, 'ORG'), (41728, 41733, 'LOC'),
(41735, 41751, 'PER'), (41984, 41989, 'LOC'), (42056, 42070, 'PER'), (42208, 42215,
'PER'), (42241, 42257, 'PER'), (42295, 42309, 'PER'), (42314, 42317, 'ORG'), (42448,
42462, 'MISC'), (42490, 42517, 'ORG'), (42519, 42522, 'ORG'), (42531, 42545, 'PER'),
(42547, 42556, 'PER'), (42558, 42572, 'PER'), (42575, 42598, 'PER'), (42643, 42671,
'MISC'), (42735, 42753, 'MISC'), (42859, 42901, 'PER'), (42917, 42938, 'LOC'), (43060,
43078, 'MISC'), (43095, 43111, 'PER'), (43129, 43143, 'PER'), (43156, 43163, 'PER'),
(43179, 43207, 'MISC'), (43209, 43224, 'PER'), (43236, 43242, 'PER'), (43252, 43268,
'PER'), (43354, 43369, 'PER'), (43392, 43397, 'LOC'), (43433, 43442, 'LOC'), (43529,
43535, 'PER'), (43610, 43626, 'PER'), (43683, 43699, 'LOC'), (43720, 43736, 'PER'),
(43762, 43767, 'LOC'), (43802, 43820, 'MISC'), (43839, 43854, 'PER'), (43867, 43874,
'ORG'), (43912, 43934, 'PER'), (43982, 44006, 'PER'), (44066, 44071, 'PER'), (44075,
44082, 'PER'), (44084, 44100, 'PER'), (44138, 44167, 'ORG'), (44284, 44287, 'LOC'),
(44311, 44340, 'ORG'), (44469, 44487, 'MISC'), (44618, 44624, 'PER'), (44648, 44662,
'MISC'), (44701, 44704, 'ORG'), (44708, 44711, 'ORG'), (44755, 44762, 'LOC'), (44848,
44862, 'PER'), (44867, 44884, 'ORG'), (44885, 44895, 'MISC'), (44925, 44940, 'PER'),

(44942, 44956, 'PER'), (44958, 44989, 'PER'), (44991, 45007, 'PER'), (45010, 45026,
'PER'), (45027, 45033, 'PER'), (45045, 45065, 'ORG'), (45067, 45069, 'ORG'), (45074,
45086, 'LOC'), (45109, 45112, 'ORG'), (45168, 45180, 'MISC'), (45249, 45259, 'MISC'),
(45270, 45283, 'PER'), (45284, 45292, 'PER'), (45320, 45326, 'LOC'), (45449, 45455,
'PER'), (45507, 45519, 'LOC'), (45675, 45678, 'LOC'), (45866, 45871, 'LOC'), (45901,
45916, 'MISC'), (45941, 45964, 'PER'), (45975, 45981, 'PER'), (46019, 46028, 'LOC'),
(46065, 46071, 'LOC'), (46113, 46119, 'LOC'), (46123, 46126, 'ORG'), (46156, 46166,
'LOC'), (46304, 46320, 'ORG'), (46504, 46515, 'ORG'), (46595, 46608, 'MISC'), (46628,
46642, 'PER'), (46656, 46665, 'ORG'), (46953, 46960, 'PER'), (47087, 47103, 'LOC'),
(47105, 47126, 'PER'), (47198, 47206, 'LOC'), (47208, 47223, 'PER'), (47234, 47250,
'PER'), (47288, 47301, 'PER'), (47344, 47358, 'PER'), (47369, 47384, 'PER'), (47421,
47437, 'PER'), (47493, 47508, 'PER'), (47565, 47597, 'LOC'), (47710, 47726, 'PER'),
(47816, 47825, 'PER'), (47826, 47848, 'PER'), (47854, 47884, 'PER'), (47946, 47967,
'LOC'), (48042, 48056, 'PER'), (48259, 48265, 'LOC'), (48425, 48438, 'PER'), (48455,
48456, 'LOC'), (48504, 48520, 'PER'), (48753, 48758, 'PER'), (49351, 49356, 'PER'),
(49399, 49405, 'PER'), (49411, 49417, 'PER'), (49433, 49465, 'LOC'), (50100, 50116,
'LOC'), (50165, 50180, 'PER'), (50264, 50275, 'ORG'), (50718, 50725, 'MISC'), (50745,
50751, 'LOC'), (50765, 50771, 'LOC'), (50808, 50814, 'LOC'), (50898, 50905, 'LOC'),
(51627, 51638, 'ORG'), (51710, 51726, 'PER'), (52062, 52069, 'PER'), (52220, 52243,
'PER'), (52376, 52384, 'ORG'), (52398, 52411, 'PER'), (52422, 52434, 'ORG'), (52438,
52446, 'ORG'), (52450, 52469, 'PER'), (52483, 52489, 'MISC'), (52513, 52529, 'PER'),
(52544, 52550, 'PER'), (52579, 52599, 'ORG'), (52662, 52674, 'MISC'), (52711, 52723,
'ORG'), (52756, 52762, 'LOC'), (52841, 52852, 'PER'), (52874, 52881, 'PER'), (52921,
52929, 'LOC'), (52979, 52993, 'PER'), (53094, 53100, 'PER'), (53112, 53124, 'PER'),
(53168, 53175, 'PER'), (53209, 53225, 'PER'), (53261, 53273, 'PER'), (53303, 53309,
'PER'), (53328, 53334, 'PER'), (53464, 53476, 'PER'), (53478, 53484, 'PER'), (53539,
53545, 'LOC'), (53602, 53614, 'ORG'), (53650, 53663, 'MISC'), (53748, 53754, 'LOC'),
(54009, 54015, 'PER'), (54034, 54042, 'ORG'), (54050, 54057, 'ORG'), (54207, 54216,
'LOC'), (54272, 54282, 'LOC'), (54319, 54335, 'PER'), (54348, 54354, 'PER'), (54380,
54391, 'ORG'), (54511, 54517, 'PER'), (54527, 54533, 'PER'), (54573, 54585, 'PER'),
(54664, 54678, 'PER'), (54693, 54705, 'ORG'), (54808, 54809, 'PER'), (54872, 54886,
'ORG'), (55099, 55115, 'PER'), (55165, 55171, 'PER'), (55226, 55232, 'PER'), (55306,
55312, 'LOC'), (55404, 55420, 'PER'), (55427, 55433, 'PER'), (55477, 55478, 'PER'),
(55705, 55717, 'MISC'), (55721, 55729, 'ORG'), (55783, 55806, 'ORG'), (55808, 55817,
'ORG'), (55850, 55856, 'LOC'), (55920, 55931, 'PER'), (55967, 55974, 'PER'), (55975,
55987, 'PER'), (56042, 56054, 'MISC'), (56058, 56066, 'ORG'), (56110, 56126, 'PER'),
(56129, 56143, 'PER'), (56261, 56281, 'LOC'), (56302, 56311, 'MISC'), (56416, 56444,
'PER'), (56460, 56468, 'ORG'), (56474, 56481, 'PER'), (56541, 56551, 'LOC'), (56611,
56624, 'PER'), (56707, 56728, 'PER'), (56759, 56768, 'PER'), (56769, 56783, 'PER'),
(56850, 56855, 'LOC'), (56858, 56866, 'LOC'), (57074, 57080, 'PER'), (57095, 57111,
'PER'), (57211, 57217, 'PER'), (57270, 57273, 'LOC'), (57316, 57321, 'LOC'), (57342,
57360, 'PER'), (57497, 57513, 'PER'), (57570, 57577, 'PER'), (57625, 57646, 'MISC'),
(57676, 57684, 'MISC'), (57687, 57693, 'ORG'), (57698, 57712, 'LOC'), (57740, 57747,
'PER'), (57769, 57787, 'PER'), (57826, 57842, 'PER'), (57995, 58002, 'PER'), (58004,
58032, 'PER'), (58045, 58051, 'PER'), (58075, 58080, 'PER'), (58089, 58096, 'PER'),
(58160, 58176, 'PER'), (58200, 58212, 'ORG'), (58254, 58270, 'PER'), (58275, 58288,
'PER'), (58310, 58316, 'MISC'), (58326, 58338, 'PER'), (58345, 58352, 'ORG'), (58364,
58381, 'PER'), (58448, 58453, 'MISC'), (58567, 58587, 'LOC'), (58604, 58617, 'PER'),
(58716, 58723, 'PER'), (58828, 58840, 'PER'), (58854, 58865, 'LOC'), (58867, 58883,
'PER'), (58899, 58920, 'LOC'), (58923, 58937, 'PER'), (59023, 59032, 'LOC'), (59033,
59041, 'LOC'), (59121, 59133, 'ORG'), (59158, 59165, 'PER'), (59247, 59267, 'PER'),
(59358, 59372, 'LOC'), (59389, 59395, 'LOC'), (59399, 59409, 'LOC'), (59411, 59425,
'PER'), (59442, 59449, 'PER'), (59453, 59469, 'PER'), (59485, 59500, 'MISC'), (59595,
59601, 'PER'), (59864, 59884, 'PER'), (59914, 59931, 'MISC'), (60096, 60102, 'LOC'),
(60135, 60141, 'LOC'), (60160, 60166, 'LOC'), (60185, 60191, 'LOC'), (60208, 60214,
'LOC'), (60303, 60315, 'ORG'), (60349, 60361, 'PER'), (60412, 60417, 'MISC'), (61037,

61049, 'ORG'), (61183, 61199, 'PER'), (61769, 61775, 'LOC'), (62415, 62419, 'PER'),
(62459, 62474, 'PER'), (62511, 62527, 'ORG'), (62555, 62568, 'LOC'), (62624, 62639,
'PER'), (62667, 62673, 'LOC'), (62711, 62717, 'LOC'), (62840, 62852, 'ORG'), (62856,
62861, 'MISC'), (62928, 62951, 'MISC'), (62958, 62968, 'ORG'), (62971, 62991, 'LOC'),
(62994, 63010, 'LOC'), (63017, 63046, 'ORG'), (63208, 63221, 'MISC'), (63583, 63595,
'ORG'), (63834, 63839, 'MISC'), (63947, 63962, 'PER'), (63967, 63979, 'ORG'), (64026,
64049, 'MISC'), (64057, 64081, 'PER'), (64263, 64273, 'ORG'), (64354, 64365, 'ORG'),
(64436, 64442, 'PER'), (64453, 64464, 'PER'), (64467, 64481, 'PER'), (64496, 64503,
'ORG'), (64539, 64547, 'PER'), (64552, 64579, 'ORG'), (64608, 64614, 'PER'), (64656,
64668, 'ORG'), (64902, 64910, 'PER'), (64958, 64970, 'ORG'), (64987, 65003, 'PER'),
(65155, 65161, 'LOC'), (65307, 65314, 'PER'), (65315, 65321, 'PER'), (65324, 65342,
'LOC'), (65351, 65374, 'ORG'), (65384, 65395, 'PER'), (65412, 65430, 'ORG'), (65439,
65464, 'ORG'), (65474, 65486, 'PER'), (65488, 65503, 'PER'), (65506, 65520, 'PER'),
(65529, 65549, 'ORG'), (65551, 65567, 'PER'), (65683, 65689, 'LOC'), (65907, 65913,
'PER'), (65971, 65976, 'PER'), (66028, 66039, 'ORG'), (66060, 66071, 'ORG'), (66159,
66182, 'MISC'), (66213, 66229, 'PER'), (66243, 66253, 'LOC'), (66312, 66323, 'ORG'),
(66387, 66393, 'LOC'), (66498, 66504, 'LOC'), (66543, 66549, 'LOC'), (66560, 66566,
'LOC'), (66689, 66697, 'ORG'), (66734, 66740, 'LOC'), (66765, 66776, 'ORG'), (66894,
66900, 'LOC'), (66962, 66976, 'PER'), (66998, 67004, 'LOC'), (67027, 67036, 'LOC'),
(67052, 67058, 'PER'), (67177, 67182, 'PER'), (67196, 67202, 'MISC'), (67204, 67209,
'LOC'), (67317, 67324, 'PER'), (67689, 67712, 'MISC'), (67762, 67776, 'PER'), (67885,
67891, 'PER'), (67932, 67953, 'PER'), (67983, 67989, 'PER'), (68051, 68057, 'PER'),
(68103, 68109, 'PER'), (68205, 68226, 'PER'), (68318, 68334, 'PER'), (68371, 68387,
'PER'), (68478, 68485, 'PER'), (68489, 68500, 'MISC'), (68596, 68624, 'PER'), (68627,
68641, 'PER'), (68699, 68710, 'ORG'), (68825, 68838, 'MISC'), (68840, 68856, 'PER'),
(68888, 68906, 'MISC'), (68924, 68936, 'MISC'), (68949, 68973, 'MISC'), (69022, 69034,
'MISC'), (69047, 69058, 'MISC'), (69062, 69097, 'MISC'), (69101, 69113, 'MISC'),
(69117, 69143, 'LOC'), (69147, 69163, 'PER'), (69196, 69217, 'LOC'), (69288, 69316,
'MISC'), (69329, 69351, 'PER'), (69356, 69372, 'PER'), (69377, 69387, 'LOC'), (69392,
69405, 'MISC'), (69410, 69447, 'MISC'), (69452, 69461, 'LOC'), (69466, 69502, 'LOC'),
(69508, 69531, 'PER'), (69532, 69541, 'MISC'), (69552, 69567, 'PER'), (69817, 69833,
'PER'), (69835, 69836, 'PER'), (69933, 69939, 'LOC'), (69952, 69958, 'LOC'), (70156,
70172, 'PER'), (70235, 70250, 'PER'), (70265, 70280, 'MISC'), (70283, 70288, 'LOC'),
(70296, 70302, 'LOC'), (70821, 70842, 'LOC'), (71104, 71118, 'PER'), (71340, 71367,
'MISC'), (71390, 71396, 'LOC'), (71443, 71447, 'ORG'), (71451, 71459, 'LOC'), (71463,
71469, 'LOC'), (71474, 71479, 'LOC'), (71633, 71647, 'LOC'), (71775, 71786, 'MISC'),
(71969, 71980, 'ORG'), (72009, 72023, 'PER'), (72050, 72069, 'MISC'), (72091, 72104,
'LOC'), (72106, 72127, 'PER'), (72149, 72163, 'PER'), (72210, 72231, 'PER'), (72315,
72322, 'PER'), (72427, 72443, 'PER'), (72512, 72533, 'PER'), (72535, 72549, 'PER'),
(72552, 72567, 'PER'), (72578, 72615, 'ORG'), (72617, 72620, 'ORG'), (72625, 72646,
'PER'), (72693, 72701, 'PER'), (72703, 72728, 'PER'), (72853, 72875, 'PER'), (72877,
72886, 'PER'), (72887, 72893, 'LOC'), (72910, 72917, 'PER'), (72954, 72986, 'ORG'),
(73023, 73029, 'LOC'), (73167, 73196, 'LOC'), (73240, 73245, 'LOC'), (73260, 73268,
'LOC'), (73294, 73300, 'LOC'), (73353, 73371, 'PER'), (73415, 73426, 'ORG'), (73471,
73487, 'PER'), (73505, 73512, 'PER'), (73676, 73682, 'MISC'), (73689, 73702, 'MISC'),
(73763, 73775, 'ORG'), (73821, 73826, 'PER'), (73905, 73920, 'PER'), (73923, 73936,
'PER'), (74088, 74094, 'PER'), (74144, 74151, 'PER'), (74215, 74238, 'PER'), (74243,
74251, 'MISC'), (74254, 74260, 'ORG'), (74328, 74334, 'LOC'), (74398, 74416, 'MISC'),
(74438, 74451, 'MISC'), (74519, 74532, 'MISC'), (74620, 74639, 'ORG'), (74643, 74649,
'LOC'), (74714, 74743, 'ORG'), (74753, 74773, 'ORG'), (74824, 74842, 'MISC'), (74864,
74880, 'PER'), (74946, 74953, 'MISC'), (74958, 74972, 'LOC'), (75026, 75037, 'ORG'),
(75069, 75089, 'ORG'), (75205, 75211, 'LOC'), (75262, 75274, 'ORG'), (75404, 75417,
'MISC'), (75442, 75454, 'ORG'), (76087, 76099, 'ORG'), (76667, 76677, 'LOC'), (76692,
76712, 'ORG'), (76830, 76842, 'ORG'), (76854, 76867, 'MISC'), (77014, 77026, 'PER'),
(77052, 77065, 'MISC'), (77151, 77154, 'PER'), (77485, 77492, 'PER'), (77493, 77499,
'PER'), (77894, 77908, 'PER'), (77969, 77975, 'LOC'), (78077, 78093, 'PER'), (78310,

78324, 'PER'), (78361, 78367, 'LOC'), (78609, 78615, 'PER'), (78677, 78683, 'PER'),
(78700, 78720, 'PER'), (78722, 78735, 'PER'), (78738, 78750, 'PER'), (78808, 78813,
'MISC'), (78891, 78907, 'PER'), (78937, 78969, 'ORG'), (79022, 79051, 'PER'), (79083,
79099, 'PER'), (79101, 79117, 'LOC'), (79136, 79147, 'PER'), (79285, 79296, 'PER'),
(79445, 79461, 'PER'), (79941, 79947, 'LOC'), (80012, 80038, 'MISC'), (80040, 80056,
'PER'), (80079, 80094, 'PER'), (80152, 80158, 'PER'), (80195, 80206, 'MISC'), (80681,
80687, 'PER'), (80731, 80742, 'MISC'), (80969, 80980, 'MISC'), (81001, 81016, 'PER'),
(81343, 81349, 'PER'), (81726, 81733, 'LOC'), (81749, 81764, 'PER'), (81919, 81925,
'LOC'), (82006, 82026, 'PER'), (82042, 82052, 'LOC'), (82054, 82056, 'LOC'), (82087,
82097, 'PER'), (82885, 82900, 'PER'), (83590, 83596, 'LOC'), (84018, 84030, 'MISC'),
(84156, 84169, 'PER'), (84180, 84192, 'PER'), (84204, 84215, 'PER'), (84229, 84241,
'PER'), (84242, 84246, 'PER'), (84290, 84302, 'PER'), (84343, 84354, 'PER'), (84417,
84422, 'PER'), (84494, 84504, 'LOC'), (84506, 84508, 'LOC'), (84523, 84533, 'LOC'),
(84605, 84618, 'PER'), (84620, 84636, 'PER'), (84639, 84652, 'PER'), (84683, 84707,
'LOC'), (84710, 84721, 'PER'), (84724, 84733, 'PER'), (84773, 84783, 'LOC'), (84827,
84853, 'PER'), (84856, 84870, 'PER'), (84935, 84941, 'LOC'), (85005, 85014, 'LOC'),
(85052, 85074, 'PER'), (85156, 85173, 'ORG'), (85197, 85220, 'MISC'), (85235, 85259,
'PER'), (85314, 85327, 'PER'), (85340, 85346, 'MISC'), (85386, 85393, 'ORG'), (85399,
85410, 'ORG'), (85424, 85441, 'PER'), (85457, 85491, 'PER'), (85532, 85550, 'LOC'),
(85580, 85592, 'PER'), (85611, 85620, 'LOC'), (85653, 85656, 'ORG'), (85658, 85684,
'ORG'), (85707, 85710, 'PER'), (85712, 85738, 'ORG'), (85865, 85880, 'PER'), (86022,
86028, 'PER'), (86039, 86050, 'PER'), (86101, 86109, 'ORG'), (86207, 86216, 'LOC'),
(86270, 86298, 'PER'), (86546, 86559, 'PER'), (86586, 86589, 'PER'), (86612, 86620,
'LOC'), (86628, 86631, 'LOC'), (86690, 86706, 'PER'), (86709, 86730, 'PER'), (86820,
86837, 'MISC'), (86986, 87002, 'PER'), (87018, 87036, 'MISC'), (87042, 87059, 'MISC'),
(87120, 87126, 'LOC'), (87291, 87307, 'PER'), (87473, 87484, 'ORG'), (87490, 87500,
'LOC'), (87631, 87640, 'MISC'), (87688, 87692, 'MISC'), (87830, 87848, 'MISC'),
(87957, 87975, 'MISC'), (88014, 88032, 'MISC'), (88128, 88144, 'PER'), (88146, 88147,
'PER'), (88351, 88363, 'ORG'), (88443, 88466, 'MISC'), (88525, 88531, 'LOC'), (88797,
88815, 'LOC'), (89001, 89010, 'LOC'), (89036, 89053, 'MISC'), (89082, 89087, 'MISC'),
(89297, 89306, 'MISC'), (89362, 89371, 'LOC'), (89700, 89708, 'LOC'), (90229, 90252,
'MISC'), (90277, 90307, 'ORG'), (90387, 90398, 'PER'), (90420, 90429, 'PER'), (90430,
90436, 'LOC'), (90533, 90561, 'LOC'), (90647, 90656, 'LOC'), (90657, 90665, 'LOC'),
(90852, 90868, 'PER'), (90869, 90875, 'PER'), (90896, 90908, 'ORG'), (90912, 90920,
'ORG'), (90933, 90956, 'MISC'), (90998, 91014, 'PER'), (91145, 91177, 'ORG'), (91229,
91240, 'PER'), (91257, 91264, 'PER'), (91302, 91319, 'MISC'), (91338, 91352, 'LOC'),
(91387, 91397, 'PER'), (91398, 91403, 'PER'), (91405, 91411, 'PER'), (91425, 91429,
'MISC'), (91490, 91504, 'LOC'), (91557, 91569, 'ORG'), (91694, 91753, 'ORG'), (92190,
92204, 'LOC'), (92252, 92277, 'PER'), (92398, 92413, 'PER'), (92460, 92470, 'PER'),
(92501, 92502, 'MISC'), (92512, 92520, 'PER'), (92530, 92531, 'MISC'), (92565, 92566,
'MISC'), (92576, 92581, 'PER'), (92585, 92592, 'PER'), (92610, 92620, 'PER'), (92719,
92727, 'MISC'), (92741, 92742, 'ORG'), (92746, 92754, 'LOC'), (92764, 92765, 'MISC'),
(92788, 92794, 'LOC'), (92825, 92837, 'PER'), (92856, 92857, 'PER'), (92878, 92904,
'MISC'), (92923, 92939, 'PER'), (92964, 92965, 'PER'), (93157, 93168, 'ORG'), (93172,
93175, 'ORG')]}
{'entities': [(3, 34, 'PER'), (45, 51, 'LOC'), (85, 105, 'PER'), (108, 129, 'PER'),
(148, 173, 'PER'), (188, 200, 'MISC'), (235, 241, 'PER'), (244, 264, 'PER'), (292,
303, 'MISC'), (330, 355, 'LOC'), (391, 400, 'LOC'), (417, 431, 'LOC'), (469, 491,
'LOC'), (529, 535, 'LOC'), (660, 687, 'LOC'), (689, 691, 'LOC'), (718, 745, 'LOC'),
(817, 837, 'MISC'), (895, 932, 'ORG'), (972, 1022, 'ORG'), (1026, 1034, 'LOC'), (1036,
1041, 'ORG'), (1075, 1082, 'LOC'), (1086, 1112, 'ORG'), (1114, 1117, 'ORG'), (1163,
1180, 'MISC'), (1253, 1280, 'ORG'), (1282, 1287, 'PER'), (1402, 1449, 'MISC'), (1451,
1456, 'LOC'), (1542, 1548, 'LOC'), (1605, 1621, 'PER'), (1645, 1652, 'LOC'), (1654,
1656, 'LOC'), (1725, 1731, 'LOC'), (1758, 1765, 'PER'), (1767, 1781, 'MISC'), (1796,
1807, 'PER'), (1886, 1921, 'LOC'), (2047, 2052, 'PER'), (2084, 2101, 'MISC'), (2139,
2150, 'PER'), (2172, 2188, 'PER'), (2200, 2206, 'LOC'), (2237, 2243, 'MISC'), (2569,

```
2587, 'LOC'), (2595, 2607, 'LOC'), (2624, 2635, 'LOC'), (2664, 2670, 'LOC'), (2796,
2807, 'LOC'), (2922, 2948, 'ORG'), (2950, 2953, 'PER'), (3073, 3081, 'LOC'), (3110,
3128, 'LOC'), (3168, 3185, 'PER'), (3275, 3283, 'PER'), (3571, 3588, 'MISC'), (3593,
3603, 'LOC'), (3767, 3779, 'PER'), (3863, 3921, 'MISC'), (3923, 3929, 'LOC'), (3933,
3940, 'LOC'), (3944, 3975, 'ORG'), (3977, 3980, 'ORG'), (4000, 4006, 'PER'), (4038,
4070, 'ORG'), (4072, 4075, 'ORG'), (4088, 4094, 'LOC'), (4120, 4123, 'ORG'), (4169,
4185, 'PER'), (4313, 4316, 'PER'), (4323, 4353, 'ORG'), (4355, 4358, 'ORG'), (4367,
4382, 'PER'), (4384, 4386, 'PER'), (4429, 4454, 'ORG'), (4456, 4459, 'PER'), (4507,
4513, 'LOC'), (4597, 4600, 'MISC'), (4732, 4738, 'LOC'), (4840, 4843, 'MISC'), (5057,
5090, 'ORG'), (5092, 5094, 'LOC'), (5095, 5102, 'LOC'), (5139, 5145, 'LOC'), (5200,
5234, 'ORG'), (5236, 5243, 'LOC'), (5308, 5330, 'PER'), (5388, 5396, 'PER'), (5398,
5410, 'PER'), (5416, 5428, 'PER'), (5458, 5459, 'PER'), (5461, 5467, 'LOC'), (5469,
5479, 'PER')]}
```

In [21]:
```
TEST_DATA = dados[-1000:]
TRAIN_DATA = dados[:-1000]
```

In [22]:
```
def main(model="pt_core_news_sm", output_dir="dhbb_nlp", n_iter=100):
    """Load the model, set up the pipeline and train the entity recognizer."""
    if model is not None:
        nlp = spacy.load(model)  # load existing spaCy model
        print("Loaded model '%s'" % model)
    else:
        nlp = spacy.blank("pt")  # create blank Language class
        print("Created blank 'pt' model")

    # create the built-in pipeline components and add them to the pipeline
    # nlp.create_pipe works for built-ins that are registered with spaCy
    if "ner" not in nlp.pipe_names:
        ner = nlp.create_pipe("ner")
        nlp.add_pipe(ner, last=True)
    # otherwise, get it so we can add labels
    else:
        ner = nlp.get_pipe("ner")

    # add labels
    for _, annotations in TRAIN_DATA:
        for ent in annotations.get("entities"):
            ner.add_label(ent[2])

    # get names of other pipes to disable them during training
    other_pipes = [pipe for pipe in nlp.pipe_names if pipe != "ner"]
    with nlp.disable_pipes(*other_pipes):  # only train NER
        # reset and initialize the weights randomly  but only if we're
        # training a new model
        if model is None:
            nlp.begin_training()
        for itn in range(n_iter):
            random.shuffle(TRAIN_DATA)
            losses = {}
            # batch up the examples using spaCy's minibatch
            batches = minibatch(TRAIN_DATA, size=compounding(4.0, 32.0, 1.001))
            for batch in batches:
                texts, annotations = zip(*batch)
                nlp.update(
                    texts,  # batch of texts
```

```
                            annotations,  # batch of annotations
                            drop=0.5,  # dropout - make it harder to memorise data
                            losses=losses,
                        )
                    print("Losses", losses)

            # test the trained model
            for text, _ in TRAIN_DATA:
                doc = nlp(text)
                print("Entities", [(ent.text, ent.label_) for ent in doc.ents])
                print("Tokens", [(t.text, t.ent_type_, t.ent_iob) for t in doc])

            # save model to output directory
            if output_dir is not None:
                output_dir = Path(output_dir)
                if not output_dir.exists():
                    output_dir.mkdir()
                nlp.to_disk(output_dir)
                print("Saved model to", output_dir)

                # test the saved model
                print("Loading from", output_dir)
                nlp2 = spacy.load(output_dir)
                for text, _ in TRAIN_DATA:
                    doc = nlp2(text)
                    print("Entities", [(ent.text, ent.label_) for ent in doc.ents])
                    print("Tokens", [(t.text, t.ent_type_, t.ent_iob) for t in doc])
```

```
In [ ]: main()
```

```
Loaded model 'pt_core_news_sm'
Losses {'ner': 6148187.004867554}
```

I.3. Similaridade semantica

```
In [10]: import numpy as np
         import networkx as nx
         %pylab inline
```

```
Populating the interactive namespace from numpy and matplotlib
```

```
In [24]: G = nx.Graph()

         for i,verb1 in tqdm(enumerate(biograficos.itertuples())):
             for j,verb2 in enumerate(biograficos.itertuples()):
                 if j>=i:
                     continue
                 doc1 = nlp(verb1.corpo)
                 doc2 = nlp(verb2.corpo)
                 sim = doc1.similarity(doc2)
                 print(sim)
                 if sim > 0.9:
                     G.add_edge(verb1.title, verb2.title, weight=sim)
             if len(G.nodes)>20:
                 break
```

```
0it [00:00, ?it/s]

0.9736412094412655



2it [00:00,  4.01it/s]

0.9692314855554565
0.9887844127611848



3it [00:04,  1.37s/it]

0.9690453424512796
0.9877588312193162
0.988704284126218



4it [00:08,  2.15s/it]

0.9312939874386111
0.8880303304435069
0.8871060491758661
0.8972676949479885



5it [00:11,  2.44s/it]

0.9778979879718187
0.9753462364899442
0.9714444980576168
0.9663467679632937
0.9262295144401926



6it [00:15,  2.88s/it]

0.9592603041086275
0.975532591880615
0.9702362581707359
0.9769561741537794
0.8732849984678037
0.9692377929472659



7it [00:19,  3.23s/it]
```

```
0.9321373688948714
0.8950161053840271
0.8954882026612847
0.9071147980548291
0.9237984481520677
0.9481836081569859
0.9120318875779666



8it [00:23,  3.43s/it]

0.9436484836235557
0.9782216219485189
0.9891414470255626
0.9796892200211428
0.8401242902069572
0.9507064622669714
0.9688882565409441
0.8619679527387331



9it [00:59, 13.15s/it]

0.9748167121226012
0.9843483944663809
0.9743991632302015
0.9821984001579731
0.886910283577205
0.96882818445918
0.9887274101027633
0.9074350832940326
0.9674930933492911



10it [01:08, 12.08s/it]

0.9407843243035253
0.962330942146688
0.9644607979500517
0.9764554327746553
0.8712466335069049
0.954183205104085
0.9578722292318408
0.9198054075661106
0.9572893878546672
0.9556852750672585



11it [01:17, 11.20s/it]
```

```
0.9709523952805834
0.9824180161628335
0.9813879056029056
0.9905682753054265
0.9059016520110134
0.9670929594732104
0.9696390158818233
0.9193160302814072
0.9675528280338189
0.9807619758015435
0.9804897469129027



12it [01:28, 11.05s/it]

0.974129091747883
0.98092806540998
0.9730809212398651
0.9734594355181105
0.9010761873907738
0.9734884673675149
0.9821564230710962
0.9152155548734985
0.9550457721425507
0.9875564458204222
0.940909910541132
0.9702688337937283



13it [01:39, 10.87s/it]

0.9827894999816637
0.9720316060912974
0.9668176915933375
0.9698890308031304
0.9354013635964981
0.9842963267406089
0.956892814617154
0.9473540475908763
0.9444062830038539
0.9669185878890049
0.950198265833854
0.9723349338711532
0.966422188470771



14it [01:50, 11.13s/it]

0.981132586304402
0.9815047779409996
0.9783434893384498
0.97677772538701
```

```
0.9101473003305437
0.9721315341369298
0.9731784459982892
0.9151609734377093
0.9548278413255117
0.9841427013827381
0.9414463814057725
0.9729626095236181
0.9954757296113066
0.9697355226716522



15it [02:02, 11.38s/it]

0.9750118779764058
0.9826250405817543
0.978151388734338
0.9852927523365185
0.9121888789328934
0.9862560542668121
0.9827794680472961
0.9427137514629921
0.9612477605606814
0.9842285427085684
0.9755486915900998
0.9848907434531723
0.9821222880320676
0.978869223936925
0.9814983394217695



16it [02:17, 12.22s/it]

0.9656398823983473
0.969678042772926
0.970864767294153
0.9774371578737523
0.9080779275061355
0.9580085885467087
0.9731642283364211
0.9004763446067745
0.9572270317214112
0.9767716452428961
0.9403238456020123
0.9706329435270021
```

KeyboardInterrupt Traceback (most recent call last)

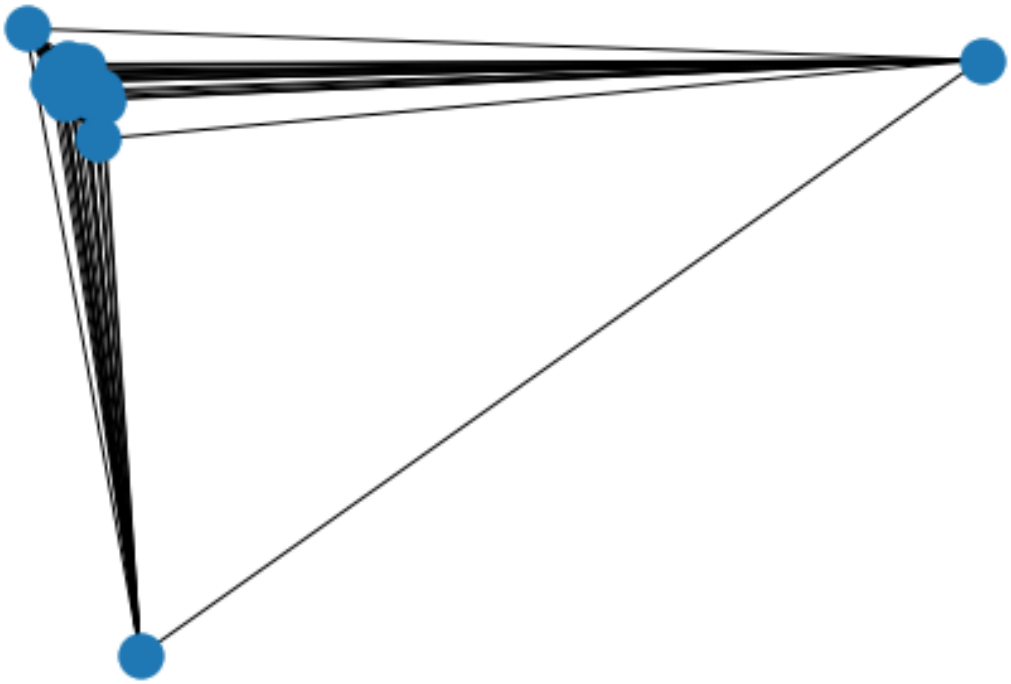
<ipython-input-24-f913f04d0c79> in <module>

```
         6             continue
         7         doc1 = nlp(verb1.corpo)
   ----> 8         doc2 = nlp(verb2.corpo)
         9         sim = doc1.similarity(doc2)
        10         print(sim)


       ~\.conda\envs\curso\lib\site-packages\spacy\language.py in __call__(self, text,␣
↪disable)
       344             if not hasattr(proc, '__call__'):
       345                 raise ValueError(Errors.E003.format(component=type(proc),␣
↪name=name))
   --> 346             doc = proc(doc)
       347             if doc is None:
       348                 raise ValueError(Errors.E005.format(name=name))


       pipeline.pyx in spacy.pipeline.Tagger.__call__()


       pipeline.pyx in spacy.pipeline.Tagger.predict()


       ~\.conda\envs\curso\lib\site-packages\thinc\neural\_classes\model.py in␣
↪__call__(self, x)
       159             Must match expected shape
       160         '''
   --> 161         return self.predict(x)
       162
       163     def pipe(self, stream, batch_size=128):


       ~\.conda\envs\curso\lib\site-packages\thinc\api.py in predict(seqs_in)
       291     def predict(seqs_in):
       292         lengths = layer.ops.asarray([len(seq) for seq in seqs_in])
   --> 293         X = layer(layer.ops.flatten(seqs_in, pad=pad))
       294         return layer.ops.unflatten(X, lengths, pad=pad)
       295


       ~\.conda\envs\curso\lib\site-packages\thinc\neural\_classes\model.py in␣
↪__call__(self, x)
       159             Must match expected shape
       160         '''
   --> 161         return self.predict(x)
       162
       163     def pipe(self, stream, batch_size=128):


       ~\.conda\envs\curso\lib\site-packages\thinc\check.py in␣
↪checked_function(wrapped, instance, args, kwargs)
       144                     raise ExpectedTypeError(check, ['Callable'])
       145                 check(arg_id, fix_args, kwargs)
   --> 146         return wrapped(*args, **kwargs)
       147
       148     def arg_check_adder(func):
```

```
       ~\.conda\envs\curso\lib\site-packages\thinc\neural\_classes\softmax.py in␣
↪predict(self, input__BI)
        16     def predict(self, input__BI):
        17         output__BO = self.ops.affine(self.W, self.b, input__BI)
   ---> 18         output__BO = self.ops.softmax(output__BO, inplace=False)
        19         return output__BO
        20


       ops.pyx in thinc.neural.ops.Ops.softmax()


       ~\.conda\envs\curso\lib\site-packages\numpy\core\_methods.py in _sum(a, axis,␣
↪dtype, out, keepdims, initial)
        34 def _sum(a, axis=None, dtype=None, out=None, keepdims=False,
        35          initial=_NoValue):
   ---> 36     return umr_sum(a, axis, dtype, out, keepdims, initial)
        37
        38 def _prod(a, axis=None, dtype=None, out=None, keepdims=False,


       KeyboardInterrupt:
```

```
In [21]: nx.draw(G, pos=nx.spectral_layout(G, weight='similarity'));
```

```
C:\Users\flavio.codeco.coelho\.conda\envs\curso\lib\site-
packages\networkx\drawing\nx_pylab.py:579: MatplotlibDeprecationWarning:
The iterable function was deprecated in Matplotlib 3.1 and will be removed in 3.3. Use
np.iterable instead.
  if not cb.iterable(width):
```



In [23]: nx.write_gexf(G,'dhbb.gexf')

In [ ]: