

**信息与知识获取**



**信息检索系统**

**姓 名 黄链泽 鄭毓恒 刘言**

**学 院 计算机学院**

**专 业 计算机科学与技术**

**班 级 2020211302**

**学 号 2020211246**

**2020211262**

**2020211268**

**任课教师 乔秀全**

**2023年 6 月**

**作业要求**

自己动手设计实现一个信息抽取实验系统，中、英文皆可，可以在作业2信息检索系统的基础上实现，也可以单独实现。特定领域语料根据自己的兴趣选定，规模不低于100篇文档，进行本地存储。对自己感兴趣的特定信息点进行抽取，并将结果展示出来。其中，特定信息点的个数不低于5个。可以调用开源的中英文自然语言处理基本模块，如分句、分词、命名实体识别、句法分析。信息抽取算法可以根据自己的兴趣选择，至少实现正则表达式匹配算法的特定信息点抽取。最好能对抽取结果的准确率进行人工评价。界面不作强制要求，可以是命令行，也可以是可操作的界面。提交作业报告和源代码。鼓励有兴趣和有能力的同学积极尝试优化各模块算法，也可关注各类相关竞赛。

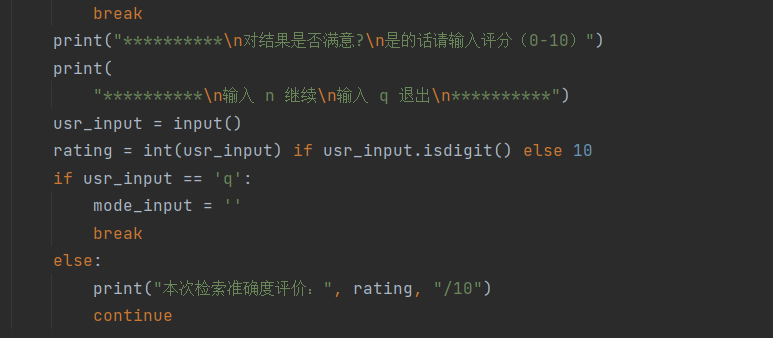
扩展要求：鼓励有兴趣和有能力的同学积极尝试多媒体信息抽取以及优化各模块算法，也可关注各类相关竞赛。自主开展相关文献调研与分析，完成算法评估、优化、论证创新点的过程。

**具体实现**

本次作业使用网上现有的数据源，从[ML Resources - BBC Datasets (ucd.ie)](http://mlg.ucd.ie/datasets/bbc.html)下载。数据源来自BBC网站，包含了由2004至2005年有关于商业、娱乐、政治、运动和科技五个主题的新闻，总共有2225个TXT文档。

本次作业采用了scikit-learn库中的Bag of Word模型构建文档向量，将单词之间的顺序关系全部丢弃，只关注文档中单词出现的次数。Bag of Word模型为每个单词分配一个唯一编号，这样一个文档就能够被编码成与已知词汇数量相同长度的向量。向量中每一个位置上的值就是其编号对应单词在文档中出现的次数。

main.py增加了人工评价系统，在每次查询后对结果进行评价：



代码定义了一个infoExtract类，类的初始化函数如下： 

初始化过程中，遍历 data\_dict 中的每个文件（for cur\_file in data\_dict.keys()）。

对于当前文件中的每个文本（for text in data\_dict[cur\_file]），进行以下处理步骤： 1.使用 nltk.sent\_tokenize() 将文本拆分成句子。

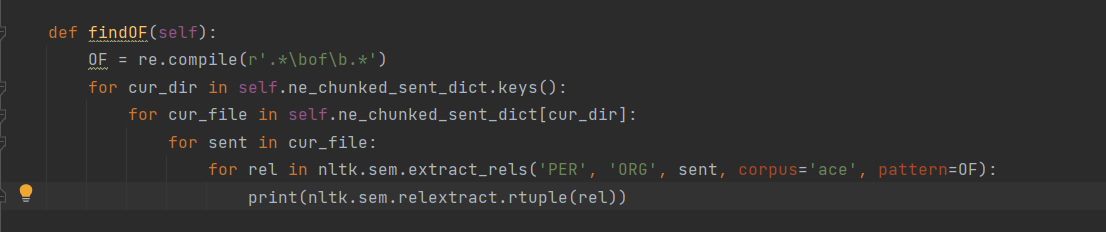
2.对每个句子使用 nltk.word\_tokenize() 进行分词，将句子分割成单词。

3.使用 nltk.pos\_tag() 对每个句子进行词性标注，即为句子中的每个单词标注其词性。

4.使用 nltk.ne\_chunk() 对标注后的句子进行命名实体识别，将句子中的命名实体识别为一个个实体块。

5.将命名实体识别结果 ne\_chunked\_sent 添加到 ne\_chunked\_sent\_dict 字典中的当前文件对应的列表中。

函数findOF目的是在处理过的句子中查找并提取部分人与部分组织信息，关键字：of

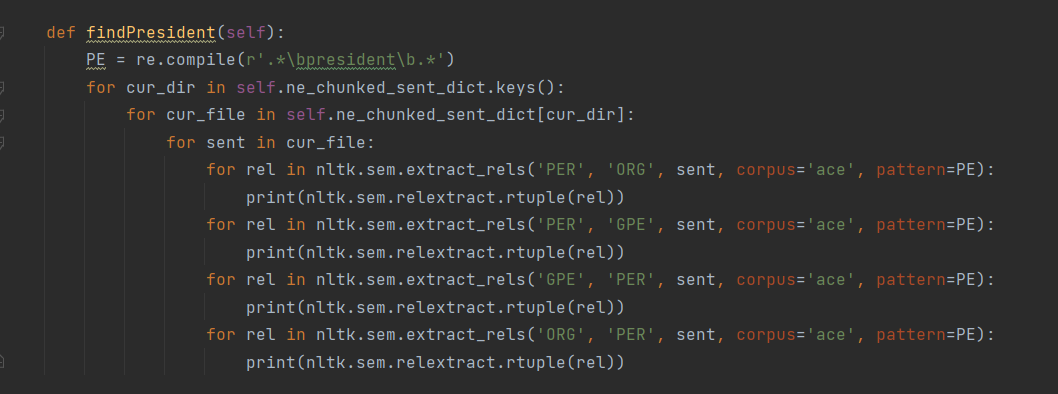


首先，使用正则表达式模式 OF来匹配包含单词 'of' 的字符串。这个模式用于在后续的关系抽取中筛选出具有 'of' 关联的实体关系。接下来，代码遍历 self.ne\_chunked\_sent\_dict 字典的键。对于每个目录（cur\_dir），它遍历与该目录关联的文件列表。对于每个文件，它遍历存储在 cur\_file 中的句子。使用 nltk.sem.extract\_rels() 函数在每个句子中提取 'PER' 和 'ORG' 之间的关系。该函数的参数指定了需要使用 ACE（自动内容提取）语料库进行关系抽取。提取出的关系通过 nltk.sem.relextract.rtuple(rel) 进行打印。

同理，函数finWin目的是在处理过的句子中查找并提取部分人的胜利信息，关键字：Win。



函数finPresident目的是在处理过的句子中查找并提取主席/总统信息，关键字：President。



首先，使用正则表达式模式 来匹配包含单词 "president" 的字符串。这个模式用于在后续的关系抽取中筛选出与 "president" 相关的实体关系。接下来，代码遍历 self.ne\_chunked\_sent\_dict 字典的键。对于每个目录，它遍历与该目录关联的文件列表。对于每个文件，它遍历存储在 cur\_file 中的句子。

使用 nltk.sem.extract\_rels() 函数在每个句子中提取以下类型的关系：

1.从 'PER'（人物）到 'ORG'（组织）的关系，使用模式 PE 进行匹配。

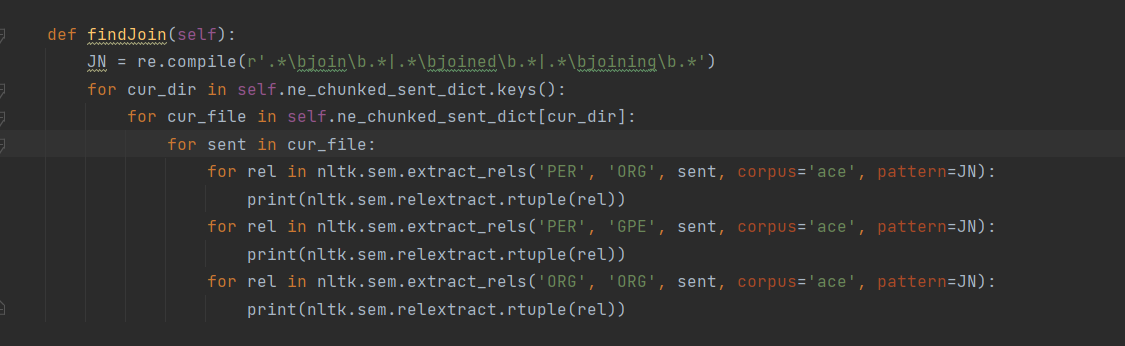
2.从 'PER'（人物）到 'GPE'（地理位置）的关系，使用模式 PE 进行匹配。

3.从 'GPE'（地理位置）到 'PER'（人物）的关系，使用模式 PE 进行匹配。

4.从 'ORG'（组织）到 'PER'（人物）的关系，使用模式 PE 进行匹配。

对于每个提取的关系，使用 nltk.sem.relextract.rtuple(rel) 进行打印。

同理，finJoin目的是在处理过的句子中查找并提取部分人与部分组织信息加入信息，关键字：Join。



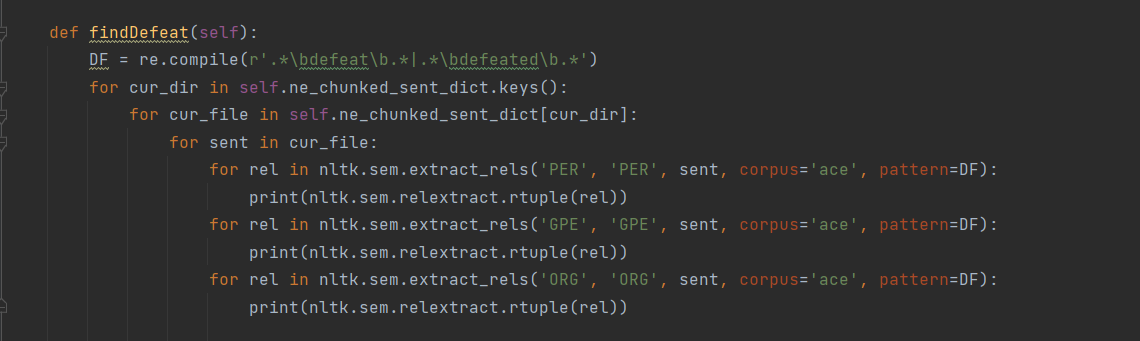
处理的类型关系包括：

1.从 'PER'（人物）到 'ORG'（组织）的关系。

2.从 'PER'（人物）到 'GPE'（地理位置）的关系。

4.从 'ORG'（组织）到 'ORG'（组织）的关系。

函数finDefeat目的是在处理过的句子中查找并提取失败信息，关键字：Defeat。



处理的类型关系包括：

1.从 'PER'（人物）到 'PER'（人物）的关系。

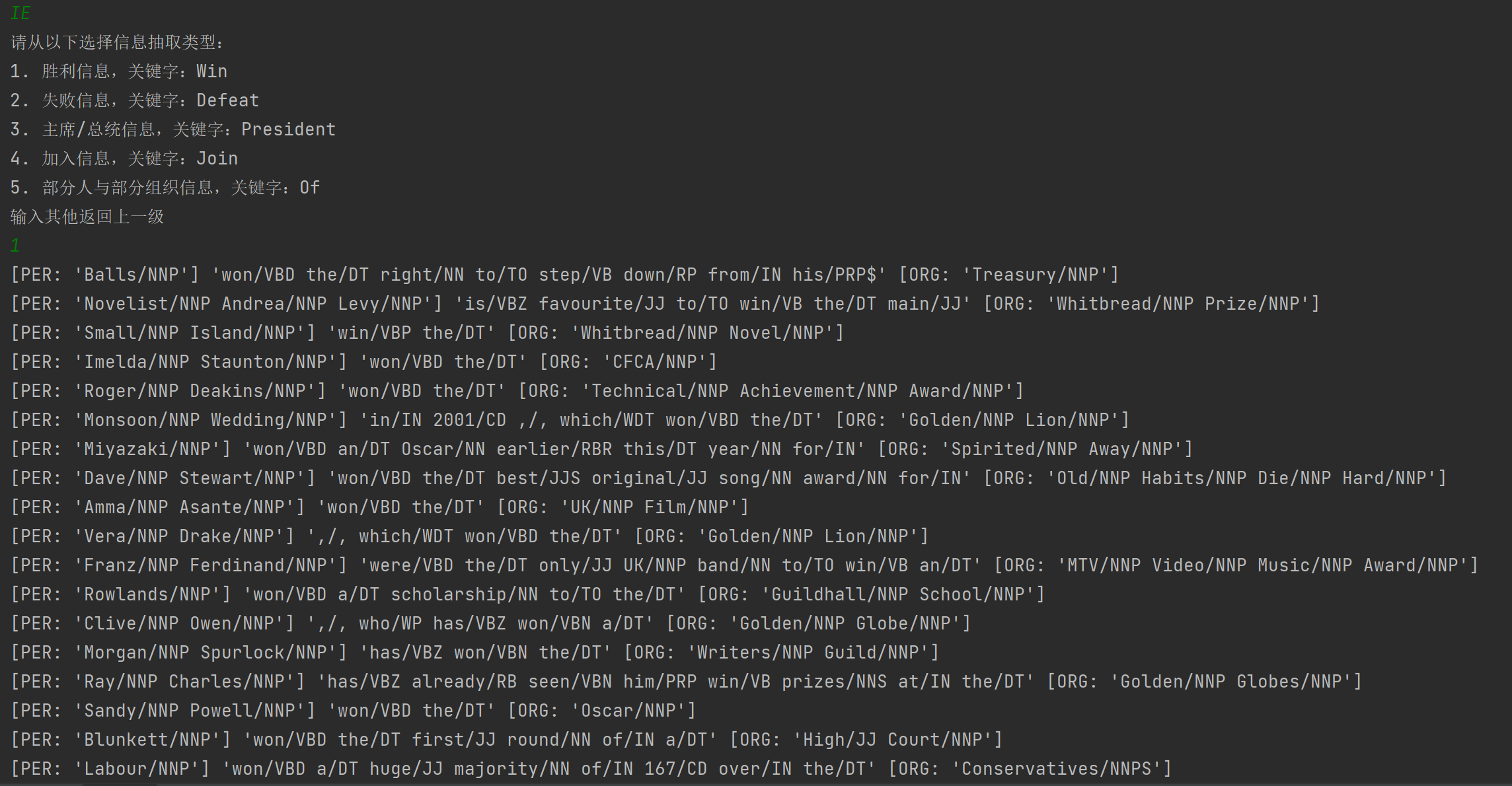
2.从 'GPE'（地理位置）到 'GPE'（地理位置）的关系。

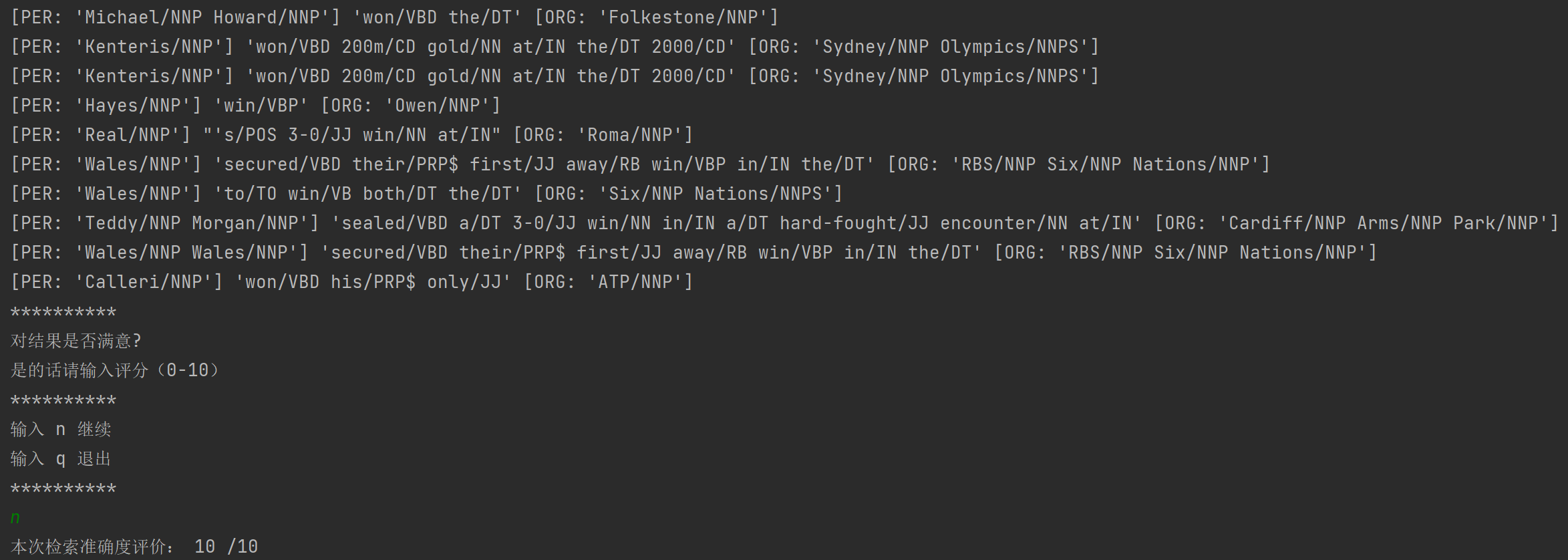
4.从 'ORG'（组织）到 'ORG'（组织）的关系。

**运行结果**

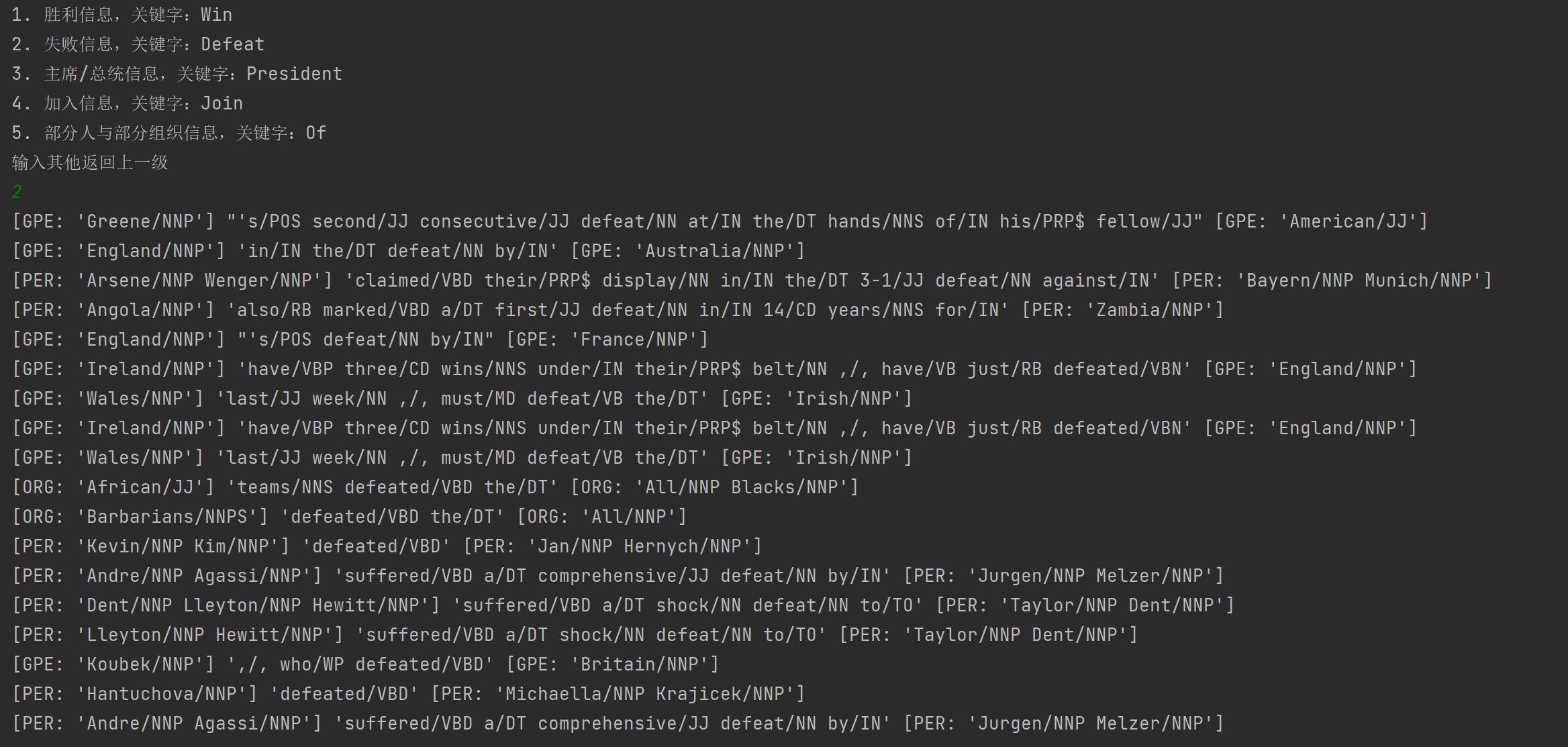
在信息抽取IE模式下，分别查看五个关键词对应抽取情况：

1. Win

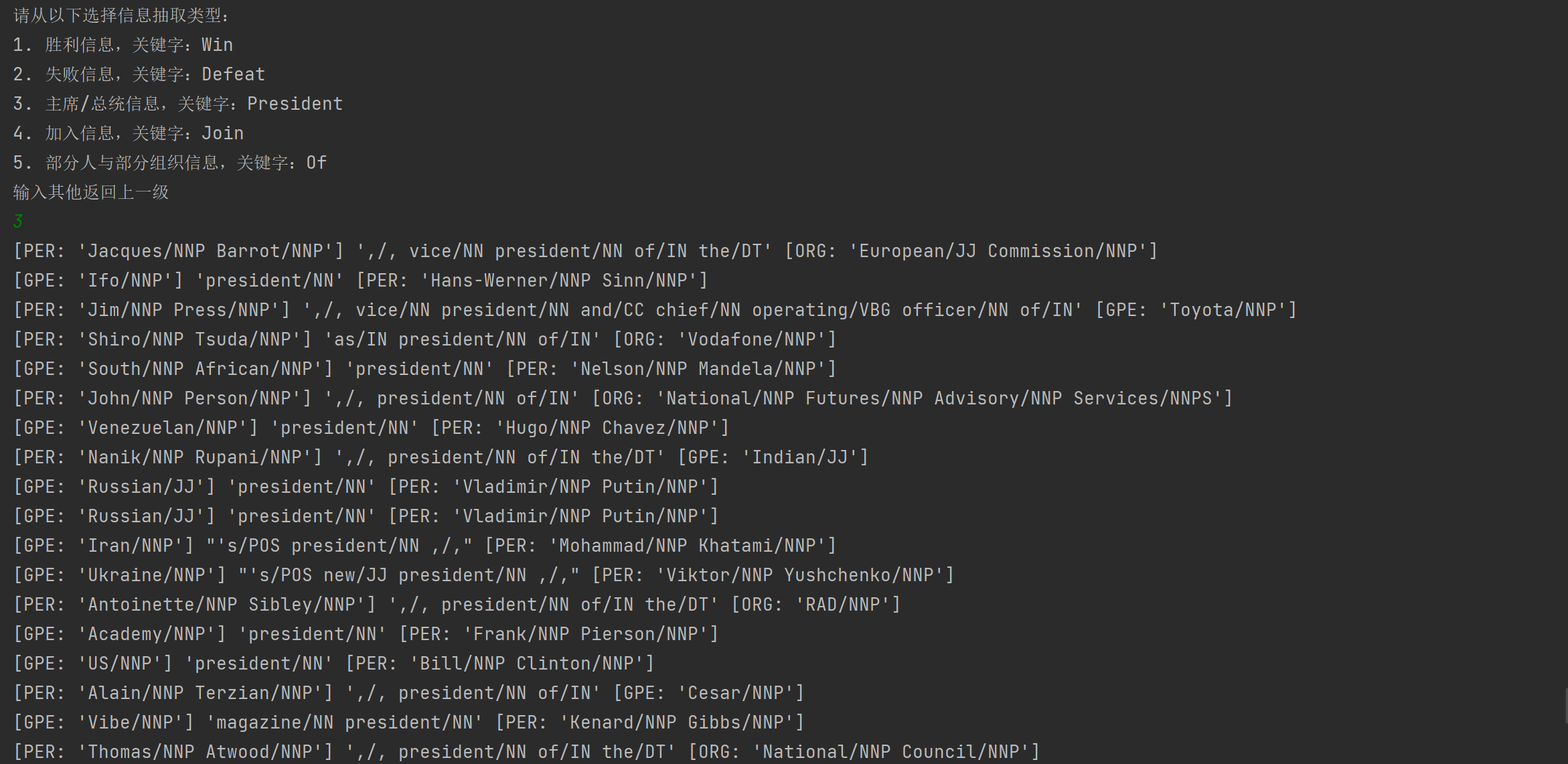


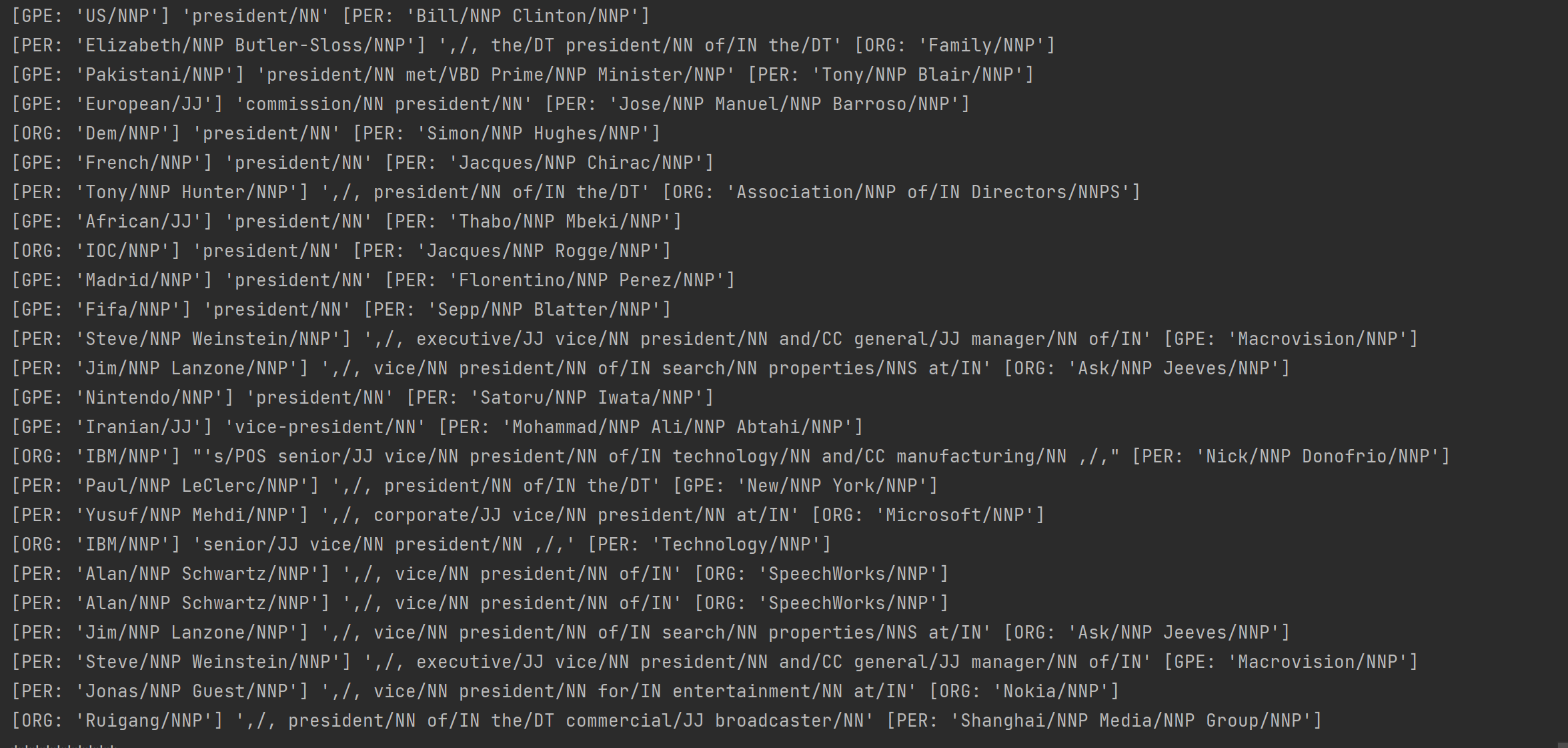


1. Defeat

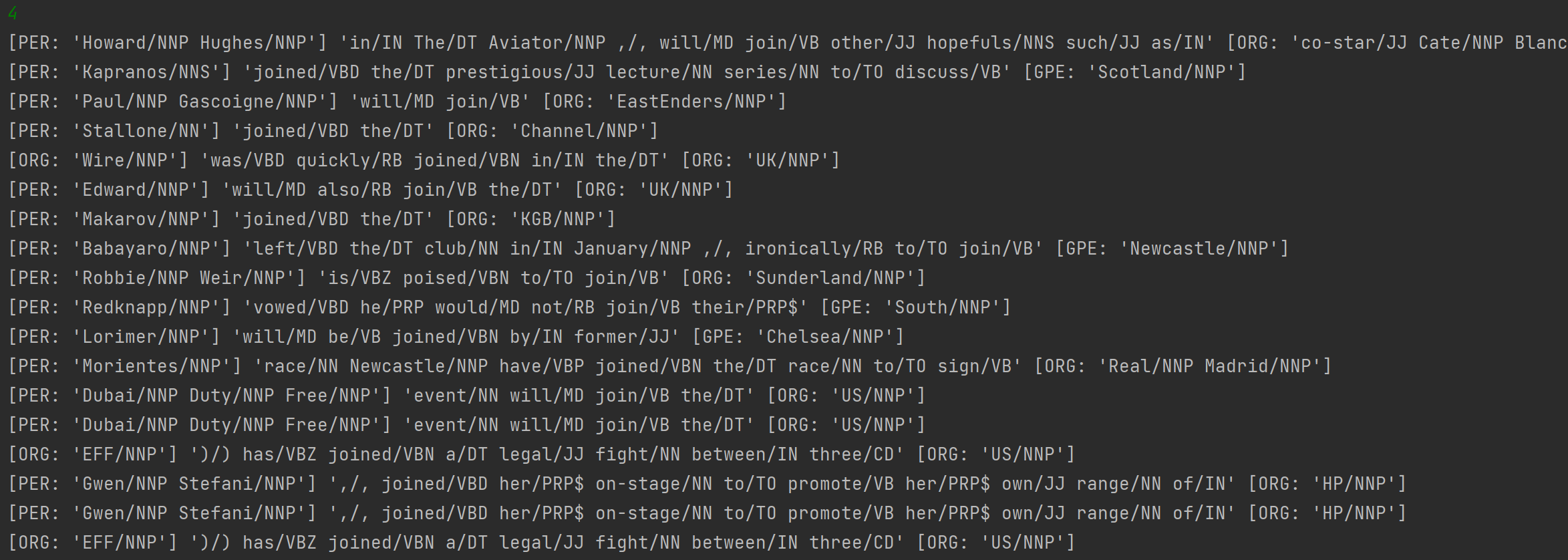


1. President

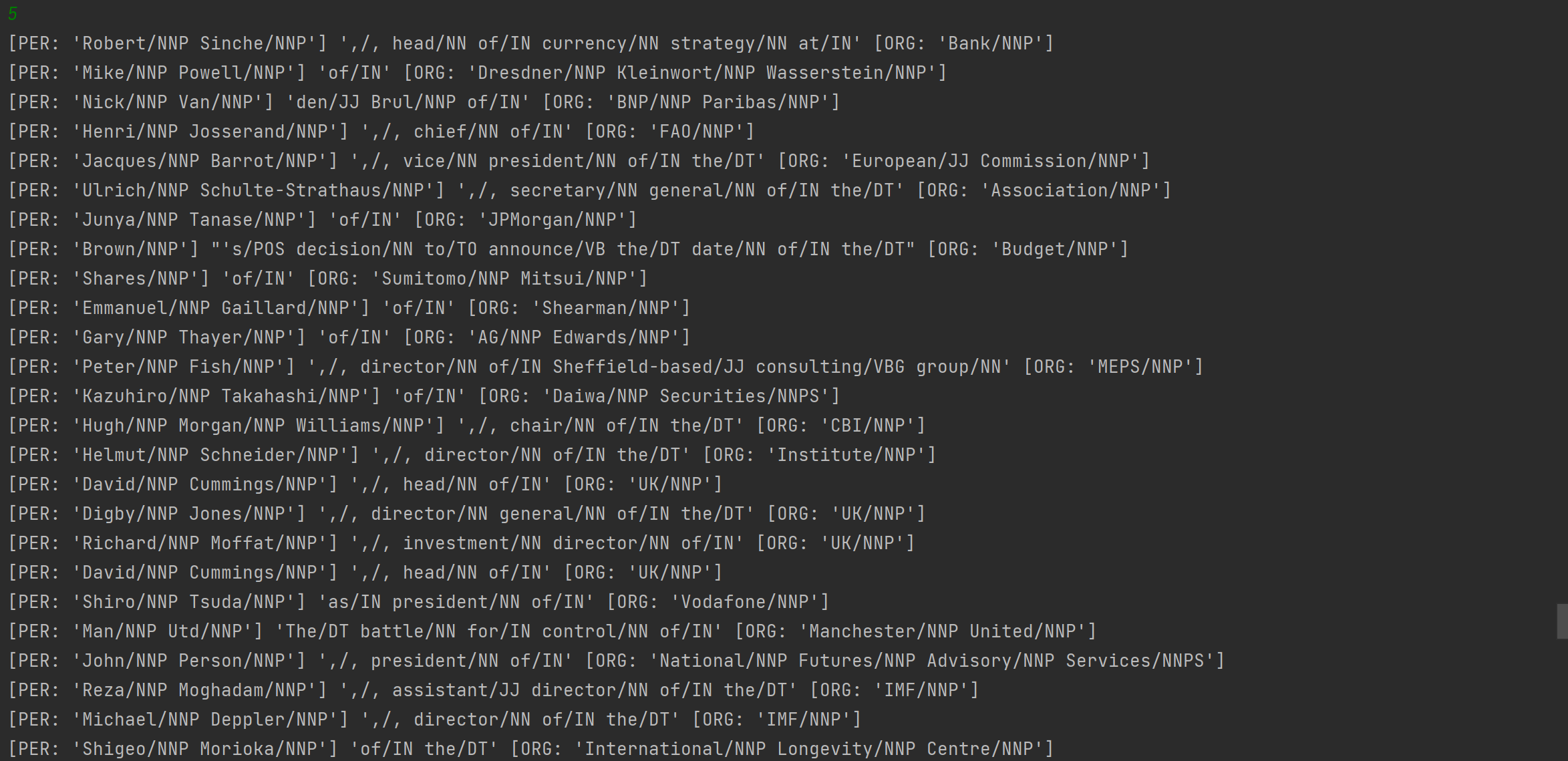


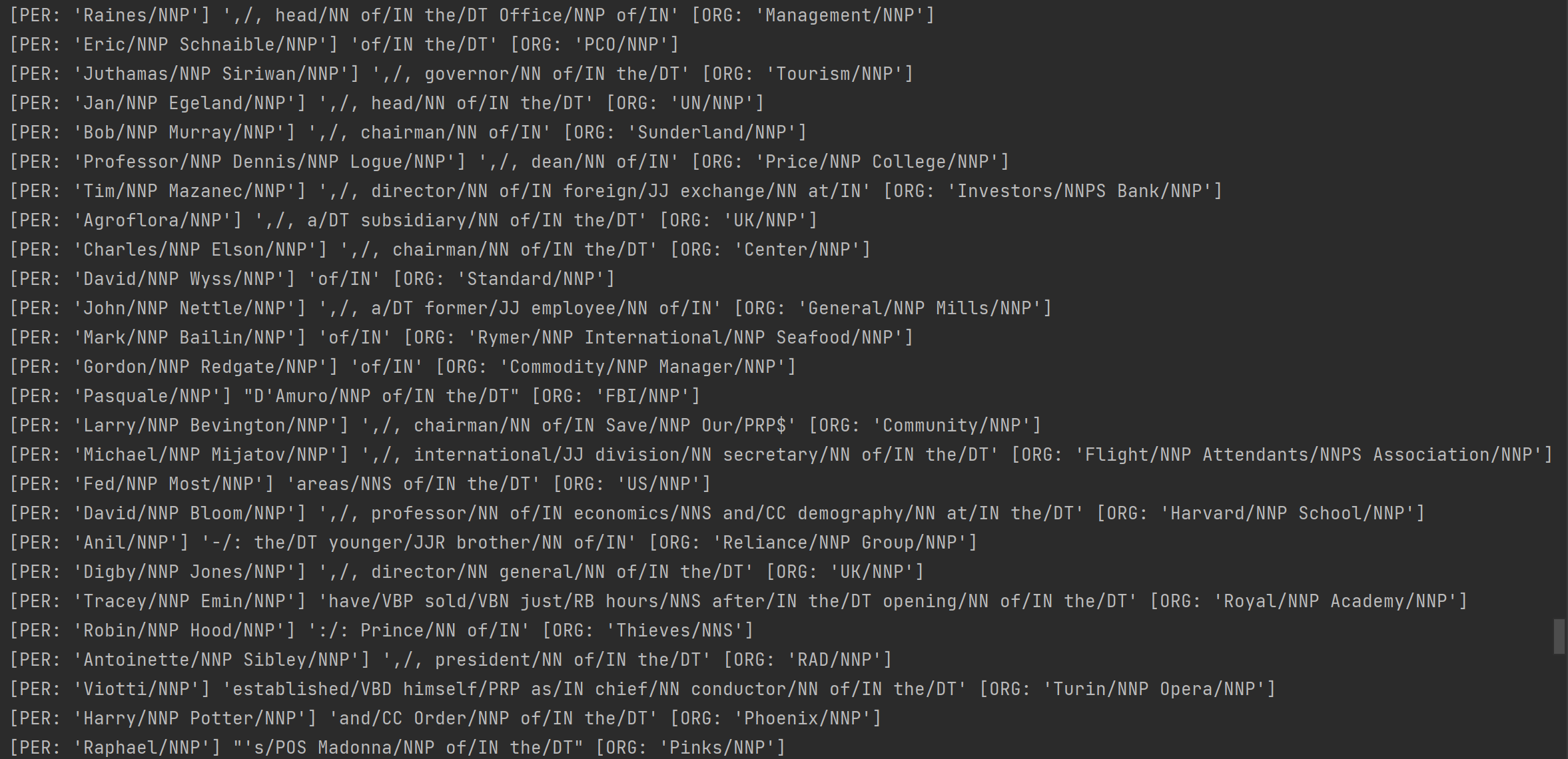


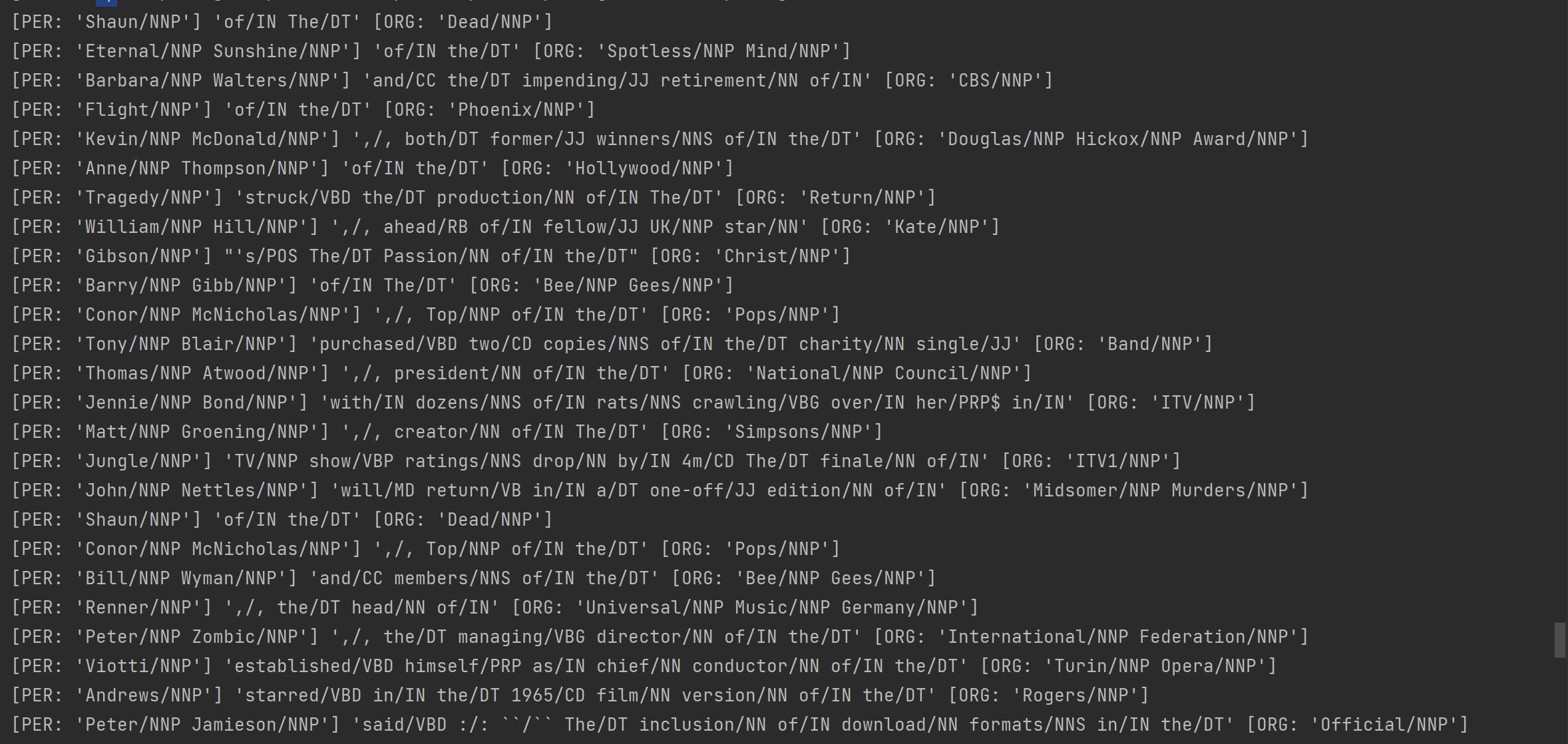
1. Join

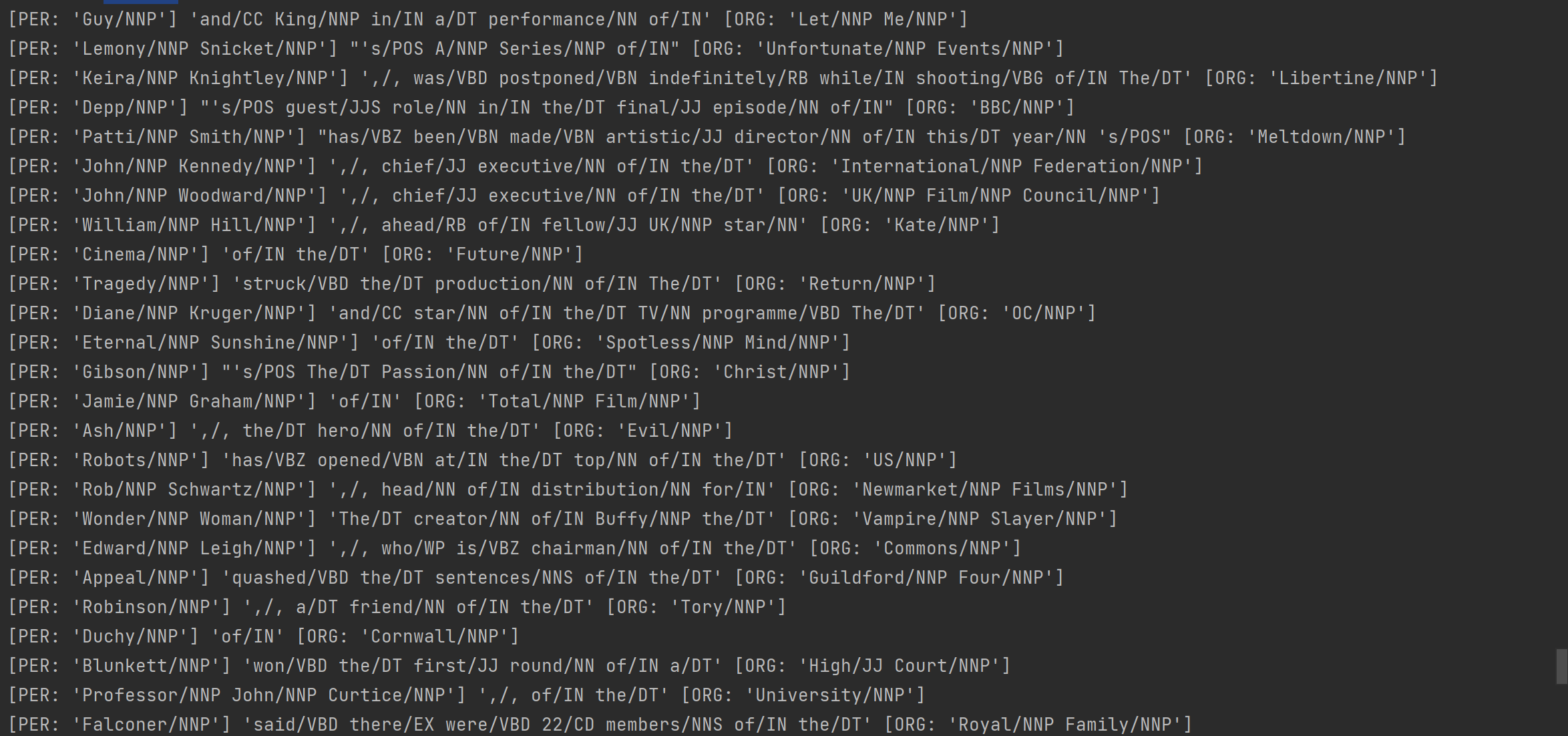


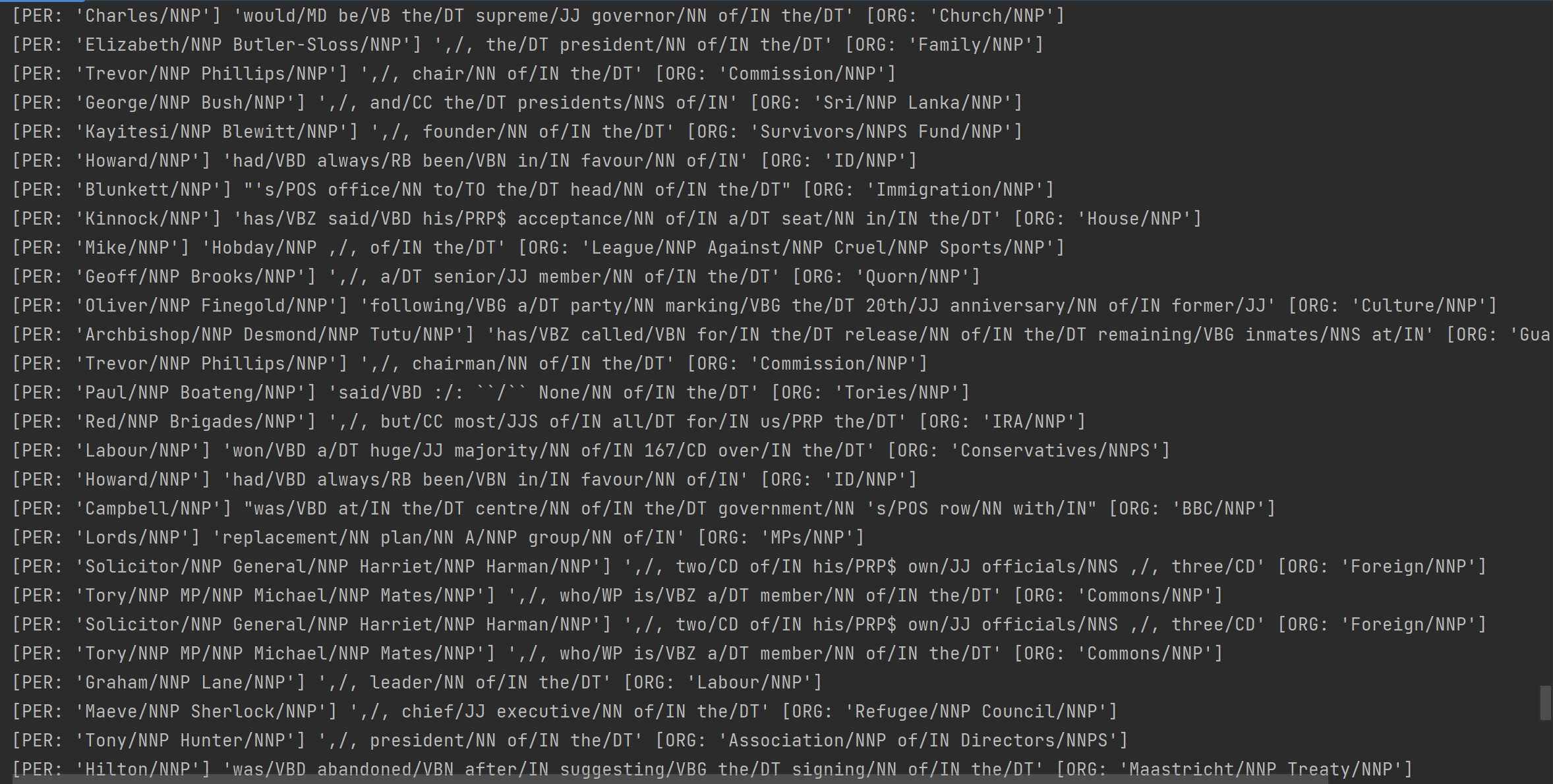
1. Of

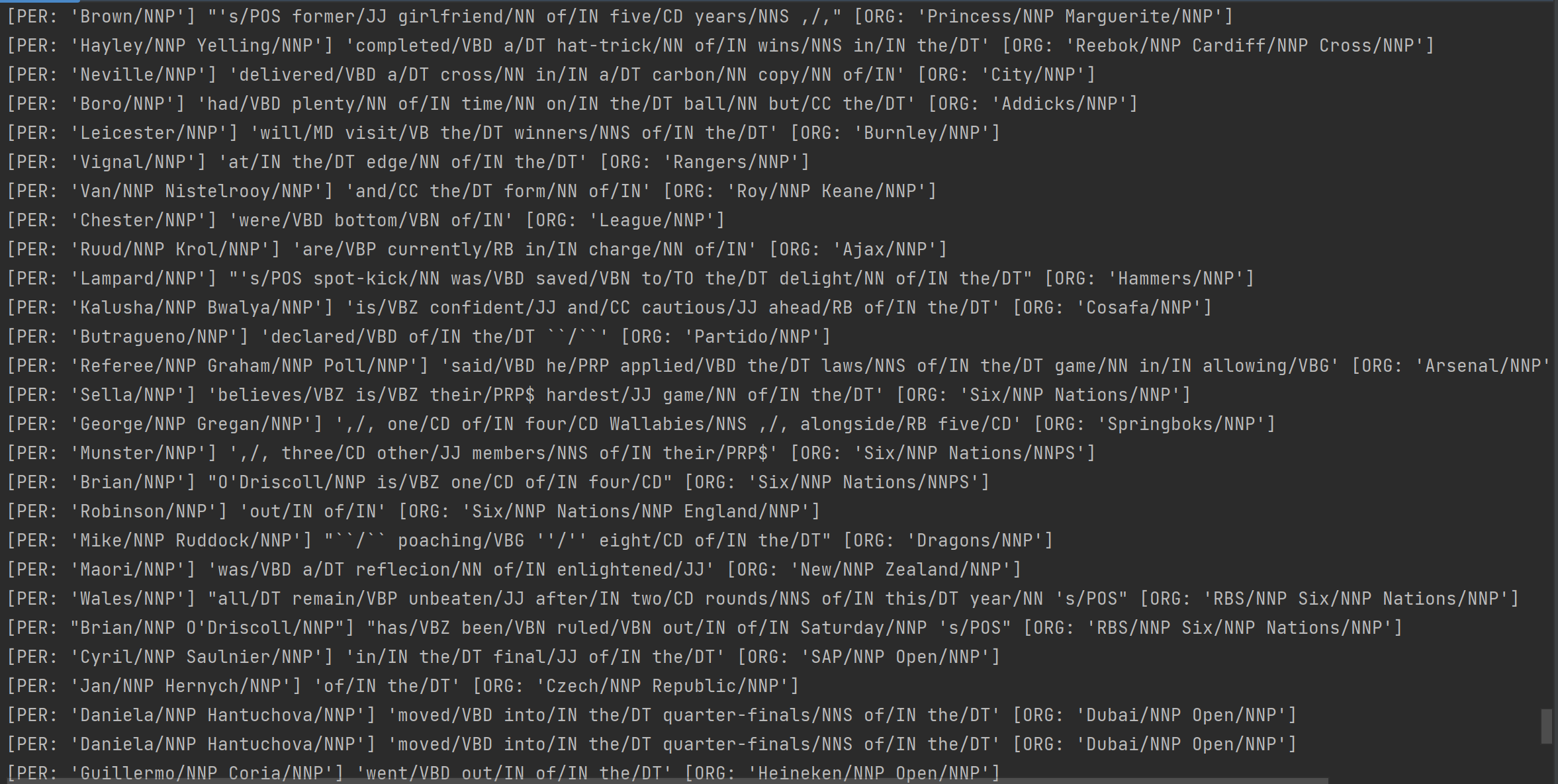


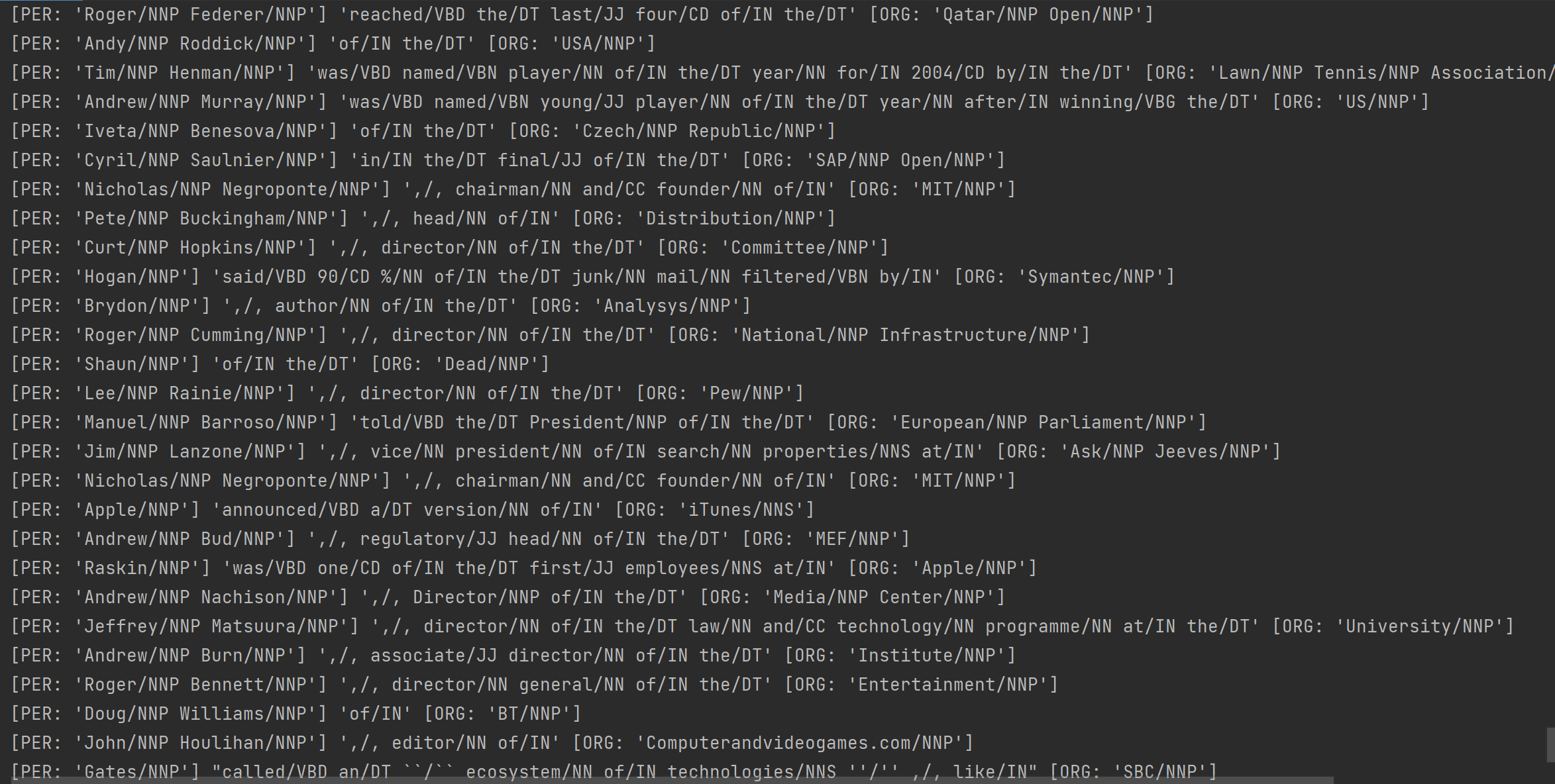


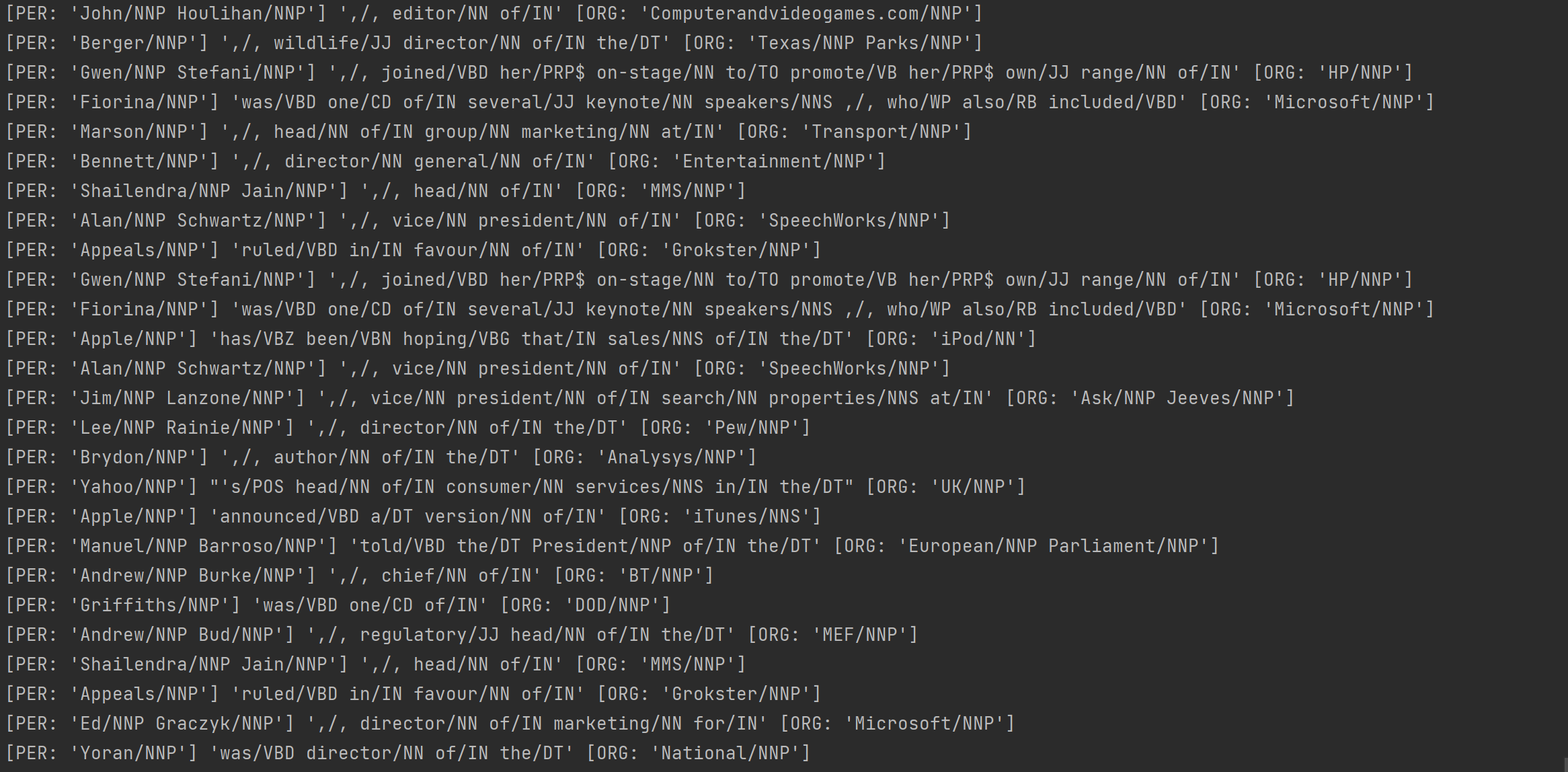












在每次抽取完成之后都可以对抽取结果进行1-10分的人工评价

**遇到的问题**

由于网络连接问题，无法通过远程下载nltk库的组件，通过手动下载，解压chunkers中的maxent\_ne\_chunker，corpora中的words，taggers中的averaged\_perceptron\_tagger以及中tokenizers中的punkt压缩包，并复制到报错中的遍历路径中。

