

**From knowledge extraction to knowledge representation**

**HaoranHU**

**M2 BI**

1. **Abstract**

**RDF (Resource Description Framework) is a data description framework advocated by the W3C to describe the resources on the earth and their relationships.**

**The core of the RDF data model includes resources (resources), attributes (attributes), RDF statements (RDF statements), etc. The core is the triples, resources-relationships-resources, which are briefly introduced next.**

1. **Introduction**

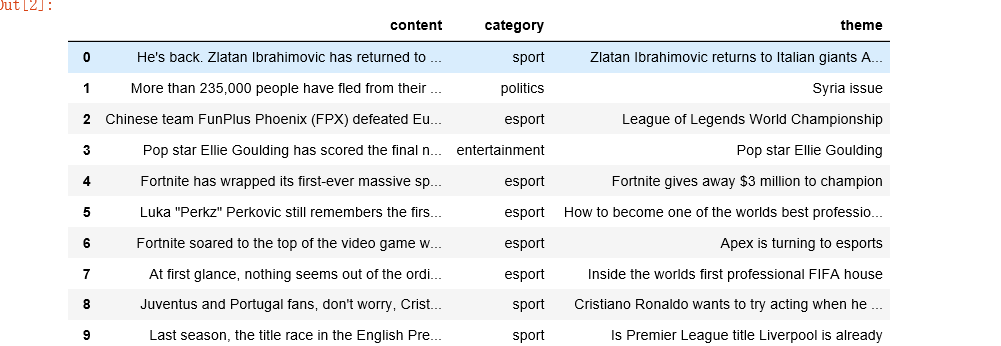
**Represent knowledge and use RDF, RDFS and OWL to enrich knowledge, use NaîveBayes and SVM to extract knowledge from text**

**Methods:Naîve Bayes and SVM,LDA,Nltk,RDF,Sparql**

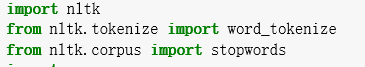
1. **Approach**
2. **I download different types of news documents online, and then use python to import the news and store it in a dataframe**

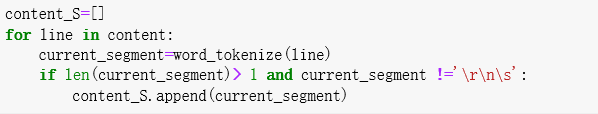


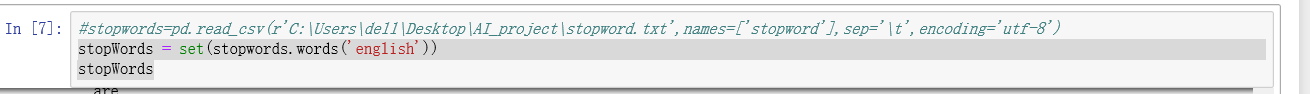


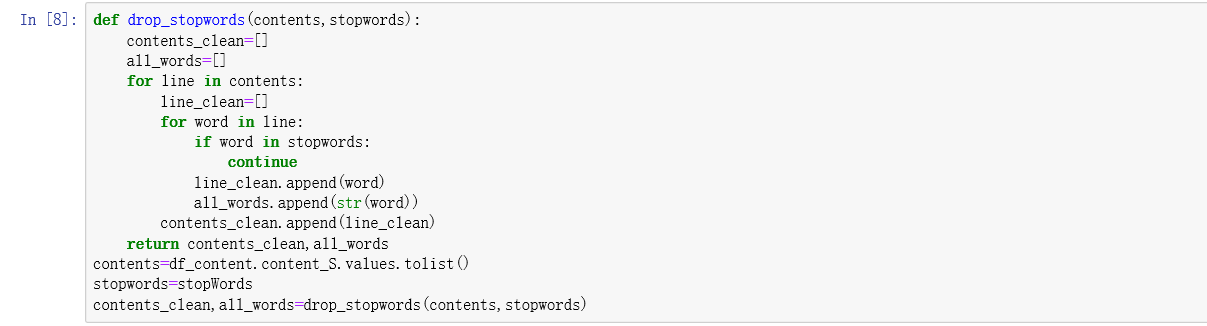


1. **Use the nltk package to tokenize news articles and remove stop words**



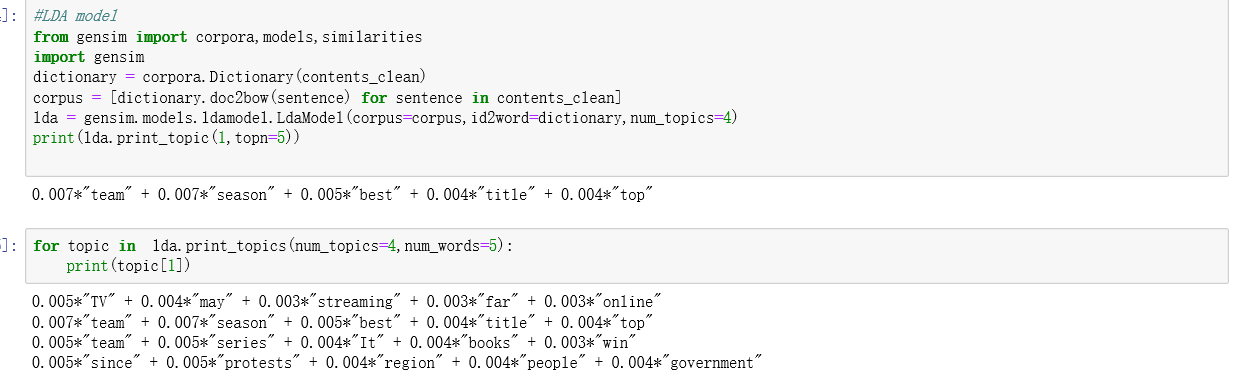








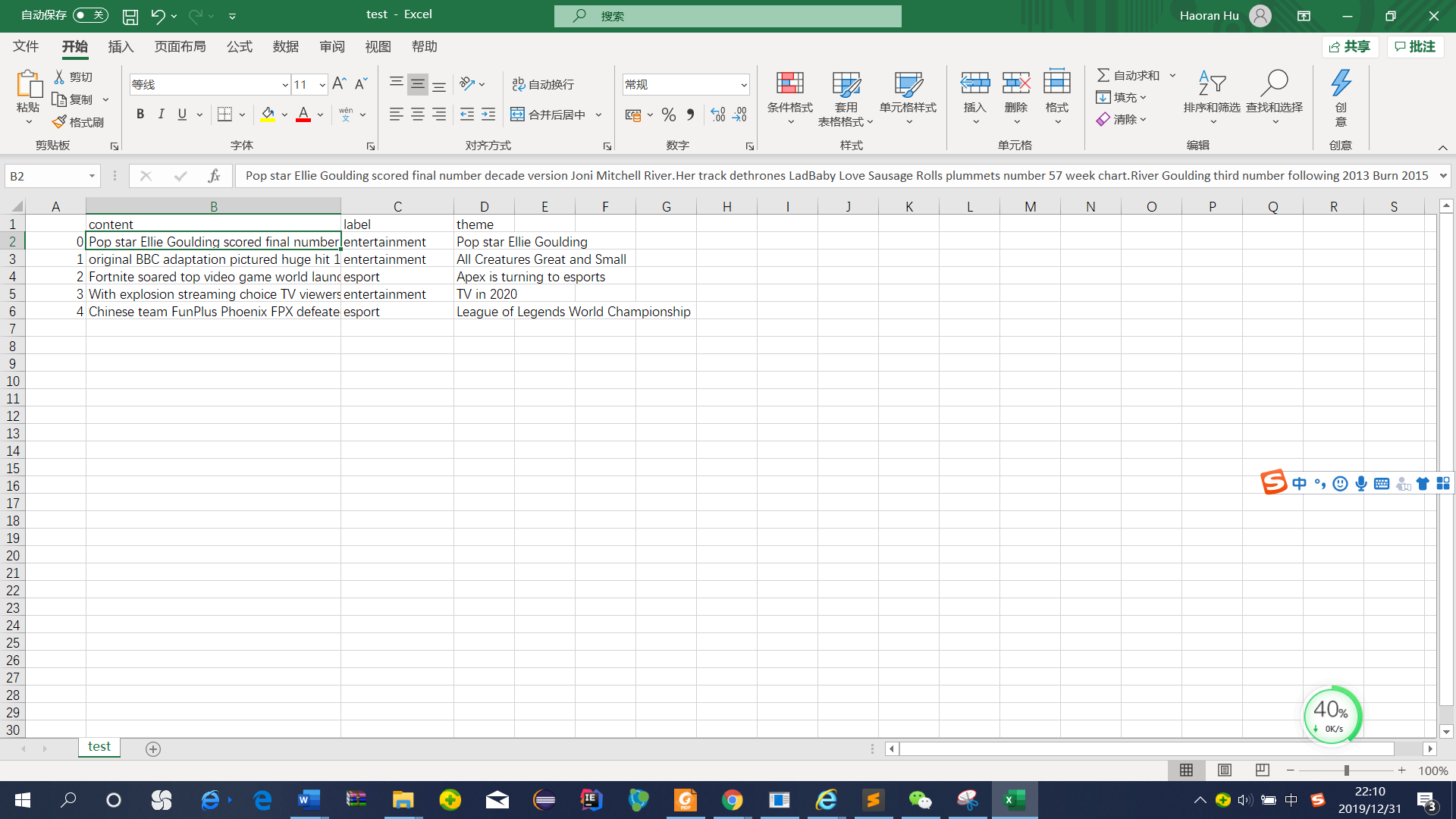
1. **There are several types of news that can be assessed using the LDA model, and their specific categories are analyzed by word frequency**



1. **Use NaîveBayes and SVM to train the dataset and then classify the news for the test set**



**5.** **output in a csv file the document class (predicted by the model)**

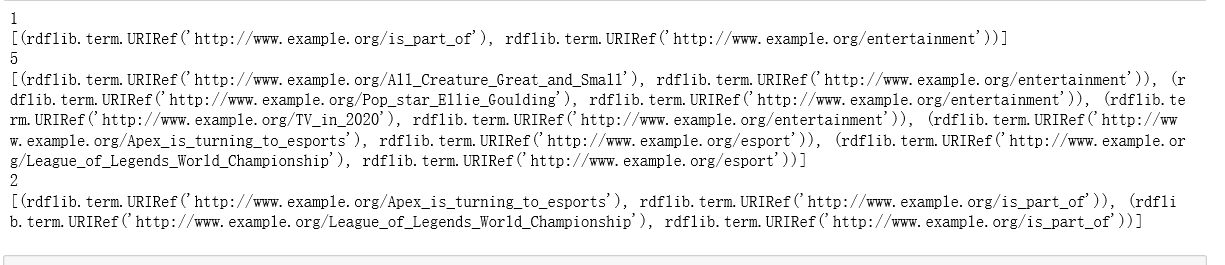


**6.Using rdflib package,** **output an RDF file.**

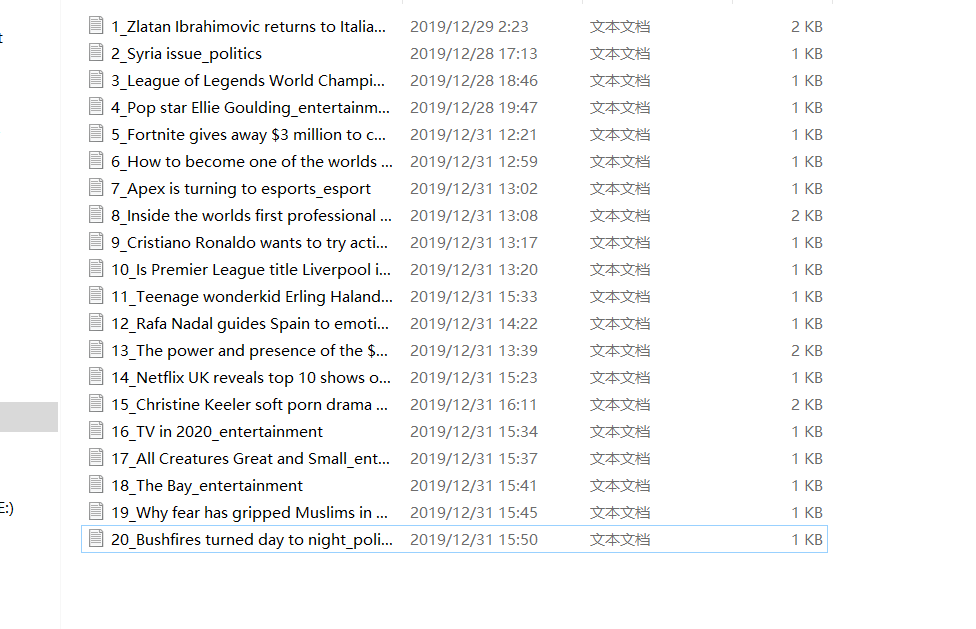


**7.** **Using sparql to query**





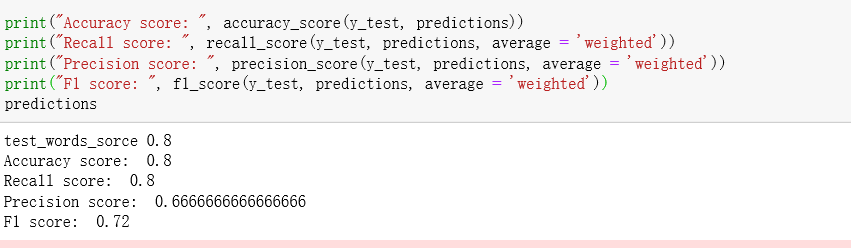
1. **Experiment and pipeline**
2. **The dataset**



**My data set is 20 news articles, they are divided into four categories: politics, entertainment, sports and e-sports**

1. **The machine learning you used and the different evaluation metrics you obtained (with explanation)**

**I use a Bayesian classifier**



1. **The sparql queries you wrote and eventually used ontology.**



**Sparql:**

**select ?relation ?part where { <http://www.example.org/Pop\_star\_Ellie\_Goulding> ?relation ?part}**

**Result:**

[(rdflib.term.URIRef('http://www.example.org/is\_part\_of'), rdflib.term.URIRef('http://www.example.org/entertainment'))]

**Sparql:**

**select ?news ?part where {?news <http://www.example.org/is\_part\_of> ?part}**

**Result:**

[(rdflib.term.URIRef('http://www.example.org/TV\_in\_2020'), rdflib.term.URIRef('http://www.example.org/entertainment')), (rdflib.term.URIRef('http://www.example.org/All\_Creature\_Great\_and\_Small'), rdflib.term.URIRef('http://www.example.org/entertainment')), (rdflib.term.URIRef('http://www.example.org/Pop\_star\_Ellie\_Goulding'), rdflib.term.URIRef('http://www.example.org/entertainment')), (rdflib.term.URIRef('http://www.example.org/League\_of\_Legends\_World\_Championship'), rdflib.term.URIRef('http://www.example.org/esport')), (rdflib.term.URIRef('http://www.example.org/Apex\_is\_turning\_to\_esports'), rdflib.term.URIRef('http://www.example.org/esport'))]

**Sparql:**

**select ?news ?relation where {?news ?relation <http://www.example.org/esport>}**

**Result:**

[(rdflib.term.URIRef('http://www.example.org/League\_of\_Legends\_World\_Championship'), rdflib.term.URIRef('http://www.example.org/is\_part\_of')), (rdflib.term.URIRef('http://www.example.org/Apex\_is\_turning\_to\_esports'), rdflib.term.URIRef('http://www.example.org/is\_part\_of'))]

1. **Conclusion**

**I learned how to use nltk in python, how to write LDA models, how to use Naive Bayes classifier and SVM for classification. At the same time, I also reviewed the rdf, sparql and other knowledge learned in the class. If there is still time I want to train a larger data set to enrich my corpus and make the algorithm more accurate**