# Text Similarity
## based on DBpedia Classes

Eva Gerlitz

January 31, 2018

## 1 Datasets

The Datasets were generated via `http://dbpedia.org/snorql/` with the query seen in listing 1. They were downloaded as .json.

Listing 1: Sparql query

```
PREFIX dbpedia0: <http://dbpedia.org/ontology/>
PREFIX dbpedia2: <http://dbpedia.org/property/>
PREFIX dbpedia: <http://dbpedia.org/resource/>

select distinct ?name ?abstract where {
     ?instance a dbpedia0:EducationalInstitution.
     ?instance foaf:name ?name.
     ?instance dbpedia0:abstract ?abstract.
     filter(langMatches(lang(?abstract),"en"))
     }
```

DBpedia consist of a lot of classes and subclasses (`http://mappings.dbpedia.org/server/ontology/classes/`), where each element of a subclass also belongs to it's parent class. Thus, the goal was to find subclasses, that are
a) specific enough so that it makes sense to train a computer to recognize it: It is nice if a computer can group a person into the class *artist*, but then we won't know what kind of artist this person ist (Actor, Painter, Musician, ...). Depending on the scenario, this might be of high interest, though.
b) the subclass still has to be large enough (around 10.000 objects).

The following classes were chosen:

- Actor
- City
- Celestial Body

- Educational Institution
- Lake

For comparison, I also took a look at the classes *Musical Artist, Athlete and Fictional Character*, but the main classes were the five mentioned above.

Except *Celestial Body*, all mentioned classes contain 10.000 objects, *Celestial Body* contains almost 8000 elements.
All subclasses of those classes were to small to be used. Furthermore, some object were only contained in a partent class, while they could have also been in a subclass: Class *College*, subclass of *Educational Institution*, contains only 127 elements, but when inspecting the objects in class *Educational Institution*, about 1200 had the word *"college"* in their name and are therefore believed to be colleges.

*Celestial Body* was chosen, as the abstracts are presumably very different from the other classes texts.
On the other hand, objects of the classes *City* and *Lake* might share similar abstracts: Both are located in some country and are often characterized by their size. In addition, the classes *City* and *Educational Institution* might overlap, as most larger cities have a (famous) university, that could be mentioned in the cities' abstract. On the other hand, Schools, Colleges and Universities are mostly located in a city, which might be named in the abstract as well.

# 2 Supervised Learning

The first attempt to distinguish the classes by their abstracts was by using supervised learning (Pythons skikit-learn).
The dataset was splitted into 80% training- and 20% testing data.

## 2.1 Preprocessing

The data was preprocessed in two different ways:
a) First by using a **hashing vectorizer**, which turns the abstracts into a sparse matrix which then holds counts of token occurrences. This is done not via a dictionary but by applying a hash function to the texts.[1]
b) In a next run, the **Tfidf Vectorizer** is used, which stands for ***t****erm* ***f****requency,* ***i****nverse* ***d****ocument* ***f****requency*. Term frequency means the proportion of a term to the total number of terms in the abstracts, the inverse document frequency will give the inverse proportion of abstracts that contain this term.[2]

[1] `http://scikit-learn.org/stable/modules/generated/sklearn.feature_extraction.text.HashingVectorizer.html`
[2] `http://www.tfidf.com/`

## 2.2 Models

Three different models were used:
a) **Random forest**, which consists of many decision trees, that are built during the training. For classification, each tree predicts a class and the most voted class "wins".[3]
b) **K-nearest neighbors**, where a prediction is made by looking at the k-nearest neighbors in the feature space and chosing the majority. The feature space is build up during training and its dimension is the number of features. [4]
c) **Stochastic Gradient Descent**, which builds multiple binary classifieres, each learning to discriminate between class K and all other classes. For classification, a confidence score is calculated.[5]

## 2.3 Results

In the following, the accuracies of how well the models perform will be presented.

### 2.3.1 2 Classes

For a first overview of how good the classes can be distinguished from one another, only two classes were used to train the model. For this, the abstracts were preprocessed using the hashing vectorizer and the training and labeling was done with random forsest. The results can be seen in table 1.

| **Classes** | **Accuracy** |
|---|---|
| Actors and Fictional Characters | 0.98 |
| Actors and Athletes | 0.989 |
| Actors and Cities | 0.997 |
| Actor and MusicalArtist | 0.954 |
| City and Lake | 0.986 |
| City and Educational Institution | 0.989 |
| Actor and Celestial Body | 1.00 |

Table 1: How well can two classes be discriminated?

The classes *city* and *actor* can be kept apart quite well, while it seems to be harder for the classes *actor* and other subclasses of *person*, especially *musical artist*. This could be explained by the fact, that many actors/actresses are musicians at the same time. This will make it impossible to choose the correct class, particularly if a person exists in both datasets.
Elements from *city* and *lake* resp. *educational institution* do in fact seem to be a bit more complicated for the model to discriminate. However, a nice accurracy rate can be observed from objects from the classes *actor* and *celestial body*.

[3] `http://www.math.usu.edu/adele/randomforests/ovronnaz.pdf`
[4] `http://www.statsoft.com/Textbook/k-Nearest-Neighbors`
[5] `http://scikit-learn.org/stable/modules/sgd.html`

### 2.3.2 5 Classes

Table 2 shows the accuracy we got when training with the three different models, depending on what preprocessing was done on the dataset.

| Preprocessing → Models ↓ | Hashing | tfidf |
|---|---|---|
| Random Forest | 0.986 | 0.986 |
| k-nearest Neighbors | 0.993 | 0.976 |
| Stochastic Gradient Descent | 0.996 | 0.994 |

Table 2: How well can five classes be distinguished?

It can be seen that the hashing vectorization as preprocessing works better than preprocessing with the tfidf vectorizer. However, only for k-nearest neighbors the differece is clearly noticeable.
To underline the accuracies, table 3 shows the absolute number of times, an object of class A was classified as any other class.

| Preprocessing → Models ↓ | Hashing | tfidf |
|---|---|---|
| Random Forest | 140 | 130 |
| k-nearest Neighbors | 70 | 231 |
| Stochastic Gradient Descent | 38 | 53 |

Table 3: How many objects of a class were mistaken for another class?

**Mistakes** As those objects that were classified uncorrectly are the most interesting ones, we will have a closer look at those.
Table 4 shows the number of times an object of a certain class was mistaken for another class.

| Classified as → Correct Class ↓ | Actor | City | Cel. Body | Edu. Institution | Lake |
|---|---|---|---|---|---|
| Actor | - | 1/1/3 | 3/2/1 | 12/13/3 | 1/1/0 |
| City | 4/3/5 | - | 1/0/0 | 26/15/15 | 34/16/3 |
| Cel. Body | 1/0/0 | 0/0/0 | - | 0/0/0 | 1/0/0 |
| Edu. Institution | 4/1/1 | 8/4/2 | 0/0 /0 | - | 3/3/0 |
| Lake | 0/1/2 | 23/10/3 | 1 /0/0 | 16/1/0 | - |

Table 4: How often were the class objects mistaken? - Hashing Vectorizer, Random Forest, K-nearest neighbors, Stochastic Gradient Descent

As expected, the objects of class *celestial body* seemed to be easily classifiable: When being trained with random forest, only two *celestial body* elements were labeled with the wrong class. In Addition, when being trained with k-nearest neighbors and stochastic gradient descent, all objects were correctly classified.
The objects of the classes *city* and *lake* were mixed up the most when using random forest. This might indicate that the abstracts of elements of those two classes use a very similar language and the same wording. Anyway, stochastic gradient descent almost solves this problem, with only three objects being mistaken in each case.
Moreover what is conspicuous is the number of times objects of the classes *city* and *educational institution* have been mixed up. While only 8 (rep. 4 and 2 - for random forest, k-nearest neighbors and stochastic gradient descent) *educational institution*-objects were labeled as city, 26 (for random forest, rep. 15 for k-nearest neighbors and stochastic gradient descent) *city*-objects were classified as educational Instutution. This combination was the second hardest to tell apart for random forest and k-nearest neighbors and by far the hardest for stochastic gradient descent. A possible explanation could be that those cities had very famous and large universities and colleges that took a large role in the cities' abstract (for example like `https://en.wikipedia.org/wiki/Cambridge`).

**Examples** To have a closer look at the mistakes the models made, the following will show some examples of wrongly classified abstracts:

**Example 1:**

> **Pantnagar is a town** and a university campus in Udham Singh Nagar district, Uttarakhand. Nainital, Kashipur, Rudrapur and Kiccha, Haldwani are the major cities surrounding Pantnagar.
> The town is famous for having the first agricultural university of India which was established on 17 November 1960. Previously the university was called the Uttar Pradesh Agricultural University or Pantnagar University. It was rechristened G. B. Pant University of Agriculture and Technology. keeping in view the contributions of Pt. Govind Ballabh Pant, the then Chief Minister of UP.
> In recent years, an integrated industrial estate has been established near the campus which houses companies such as Tata motor, Bajaj, Britannia, HP, HCL, Voltas, Schneider Electric, Nestle, Dabur, Vedanta Resources etc., as a part of SIDCUL industrial area developed by government owned State Industrial Development Corporation of Uttarakhand Limited.

The correct class of the article was *city*, but it was labeled as *educational Institution*. Most likely this was due to the second paragraph which adsresses the university of pantnagar as well as the many times the words *university* and *campus* were used.
**Example 2:**

> **Rangsit is a city** in Pathum Thani Province, Thailand. Rangsit is the home of Rangsit University.

Just as in example 1, this text belogs to a *city* object but was wrongly classified as *educational institution*. This possibly happend, because the abstract is very short.
**Example 3:**

> Lingambudhi Park is a park in the city of Mysore, India.

In DBpedia, the object with the above abstract belongs to class *lake*, while common sense would say it is not. With only very few words to work with, it is thus labeled as *city*. But how did this object end up in the dataset anyway? A look at the whole wikipedia article might reveal why: As picture **??**odo shows, a large, or even the biggest part of the article is not about the park itself, but Lingambudhi Lake, which is situated right next to the park. If DBpedia looks at the whole articles, this misclaffification could almost be described as logical.
**Example 4:**

> Heather Leigh West is a New York City based American recording artist best known for her work in house music.

According to DBpedia, this object belongs to class *city* and got labeled as *educational institution*. While in example 3 one could argue, that neither *lake* nor *city* are too far away from the true class, in this case both, the DBpedia class as well as the predicted class, are completely wrong. Interestingly, this object also appears in the dataset of *musical artist* and hence can be found twince in the complete dataset, if the *actor* class is exchanged with the *musical artist* class.
Depending on which of those objects in part of the training- and testingdataset, the result looks differnet: If only one of the Heather Leigh Wests is contained in the trainingset, the other Heather Leigh West will be classified accordingly. However, when both Heather Leigh Wests are in the testingset, both will be correctly classified as *musical artist*.

# 3 Unsupervised Learning - Topic Modeling

After supervised learning worked quite well already, unsupervised learning was tested.
For this, the input were all abstracts without any information about the corresponding classes.

## 3.1 Preprocessing

There are two slightly different ways the abstracts werde preprocessed.
For the first attempt, every punctuation character and every stop word was removed from the abstracts. Afterwards every word was lemmatized.

After training it became clear that there is room for improvement (see section 3.3), so the next step was not only to remove all stopwords, but also every part of the names of the objects.
After all unnecessary words and symbols are removed, the dictionary is bulit by assigning an id to each unique term. Afterwards, the abstracts are transformed: For each containing term a tupel of the form (term id, frequency of term id in abstract) is added.

## 3.2 Models

Latent Dirichlet Allocation, short LDA, is a statistical model that tries to figure out what kind of topics could have generated the input text and will give the most probable words for those topics. [6]

## 3.3 Results

The model was told to give five topics of all abstracts. After the first run, the following were the five topics, represented by five words each:

- school, high, university, college, student
- lake, river, located, water, area
- star, asteroid, galaxy, year, constellation
- loch, also, ontario, film, john
- chinese, hong, china, kong, quechua

As the abstracs are from five different topics, the optimal output would be if each topic matches one class. The first three topics seem to fit this expectation, as they describe the classes *educational institution, lake* and *celestial body*. The fourth topic might express *actor*, but this is only due to the word "film".
The words "john", "hong" and "kong" are not meaningful for this task, so all parts of names were also removed from the abstracts.
After the next round, the results looked better:

- school, high, university, college, student
- river, located, area, reservoir, water
- asteroid, year, galaxy, constellation, approximately
- chinese, film, actor, actress, known
- province, county, norway, lough, district

The first four classes are easy to recognize: *educational institution, lake, celestal body* and *actor*. Only the last topic (*city*) might have been difficult to match to the correct class, when not knowing what classes the abstracts belonged to.

[6] `https://www.analyticsvidhya.com/blog/2016/08/beginners-guide-to-topic-modeling-in-python/`