

# Diversity of Wikipedia article references

Yaroslava Lochman  
Philipp Kofman  
Sasha Chepurnoi  
Vadym Korshunov

July 2019

# Notes

The project must:

- Include ML tasks (clustering, classification and/or ranking) in massive datasets.
  - Have a use case in recommendation. (RecSys)
  - Use public (open) data. (Anyone can access it).
- The presentation (by video conference) can be done by any (or all) of the member groups. The lecturer (I) will make questions to each student. During your presentation you should specify which role/responsibility was taken by each member of the group. The questions will be specifically about that responsibility.
  - You will be grade based in the oral presentation and your final report. The grade will be by group (same grade for all group members).
  - You can generate one single ranking or multiple rankings. For example you can generate personalized rankings (given a user in the English Wikipedia, you generate a specific ranking for her). You also can generate rankings per topic (one ranking for history, one for arts, etc) and then generate an strategy for mix them. Whatever strategy that you decide, need to be justified and have a proper evaluation metric.
  - Remember that in rankings, specially for RecSys, the most important elements are the ones in the Top-N. In other words, if you create a ranking for predicting pageviews, and you succeed in all the page that will get 0 views, but miss all the top-10, probably your ranking is not good. Keep that in mind when design your evaluation strategy.

# Outline

- Goals / Problem statement
- Motivation
- Work pipeline
  - Overview
  - Data processing + Feature Engineering
  - Modeling
  - Evaluation
- Further work

# Goals / Problem Statement

## Principal component analysis

David Hosmer, Jr. (2010)

**Principal component analysis (PCA)** is a statistical procedure that uses an orthogonal transformation to convert a set of observations of possibly correlated variables (perhaps each of which takes on various numerical values) into a set of values of linearly uncorrelated variables (called principal components). This transformation is defined in such a way that the first principal component has the largest possible variance (that is, accounts for as much of the variability in the data as possible), and each succeeding component in turn has the highest variance possible under the constraint that it is orthogonal to the preceding components. The resulting vectors (each being a linear combination of the variables and carrying a characteristic of the uncorrelated component) are called PCA vectors or the linear loading of the original variables.

PCA was invented in 1901 by Karl Pearson<sup>[1]</sup> as an analogue of the principal axis theorem in mechanics. It was later independently developed and named by Harold Hotelling in the 1930s.<sup>[2]</sup> Depending on the level of application, it is also sometimes known as factor analysis (FA) or principal component regression (PCR). The resulting vectors (each being a linear combination of the variables and carrying a characteristic of the uncorrelated component) are called PCA vectors or the linear loading of the original variables.

PCA is mostly used as a tool in exploratory data analysis and for making predictive models. It is often used to stabilize generalization and interpretability between equations. PCA can be done by singular value decomposition (SVD) of a data matrix or by eigenvalue decomposition of a data matrix, usually after a normalization step of the original data. The transformation of each individual variable of input variables - collecting each data value from its variable (measured mean) to that of principal mean (average) is 100% - and, possibly, normalizing each variable's variance to mean 1 and 1 unit of variance.<sup>[3]</sup> The results of PCA are usually discussed in terms of component scores, sometimes called factor scores (the transformed variable values corresponding to a particular data point), and loadings (the weights of each original variable upon the principal components).<sup>[4]</sup> Component scores are sometimes used for clustering, finding that cluster of data points that are the most significant (significant), if component scores are not standardized (therefore they contain the data variance then loading) may be lost. However, this is not a problem if the data are standardized (they contain the data variance then loading) may be lost. However, this is not a problem if the data are standardized (they contain the data variance then loading) may be lost.

PCA is the simplest of the so-called linear dimensionality reduction techniques. Often, it is considered as the first step of a more complex dimensionality reduction of the data in a way that retains the variance in the data. It is a multivariate statistical technique that is used to reduce the dimensionality of a large dataset. It is a linear transformation that is used to reduce the dimensionality of a large dataset. It is a linear transformation that is used to reduce the dimensionality of a large dataset.

PCA is closely related to factor analysis. Factor analysis typically involves more domain-specific assumptions about the underlying structure and aims at interpretation of a single different factor. PCA is a linear transformation that is used to reduce the dimensionality of a large dataset. It is a linear transformation that is used to reduce the dimensionality of a large dataset.

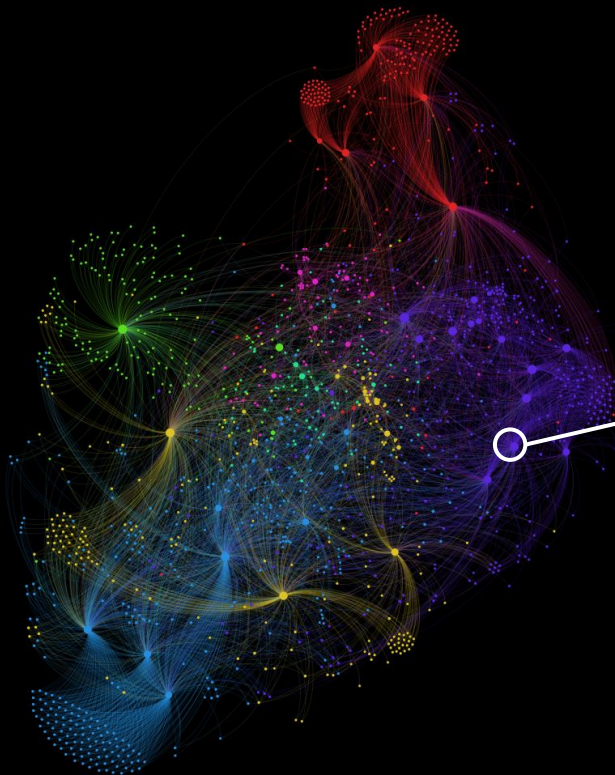
PCA is also related to principal component regression (PCR). PCA allows multivariate systems that optimally describe the cross-correlation between datasets while PCA defines a new orthogonal coordinate system that optimally describes variance in a single dataset.<sup>[5]</sup>

PCA is a linear transformation that is used to reduce the dimensionality of a large dataset. It is a linear transformation that is used to reduce the dimensionality of a large dataset. It is a linear transformation that is used to reduce the dimensionality of a large dataset.

PCA is a linear transformation that is used to reduce the dimensionality of a large dataset. It is a linear transformation that is used to reduce the dimensionality of a large dataset. It is a linear transformation that is used to reduce the dimensionality of a large dataset.

PCA is a linear transformation that is used to reduce the dimensionality of a large dataset. It is a linear transformation that is used to reduce the dimensionality of a large dataset. It is a linear transformation that is used to reduce the dimensionality of a large dataset.

PCA is a linear transformation that is used to reduce the dimensionality of a large dataset. It is a linear transformation that is used to reduce the dimensionality of a large dataset. It is a linear transformation that is used to reduce the dimensionality of a large dataset.

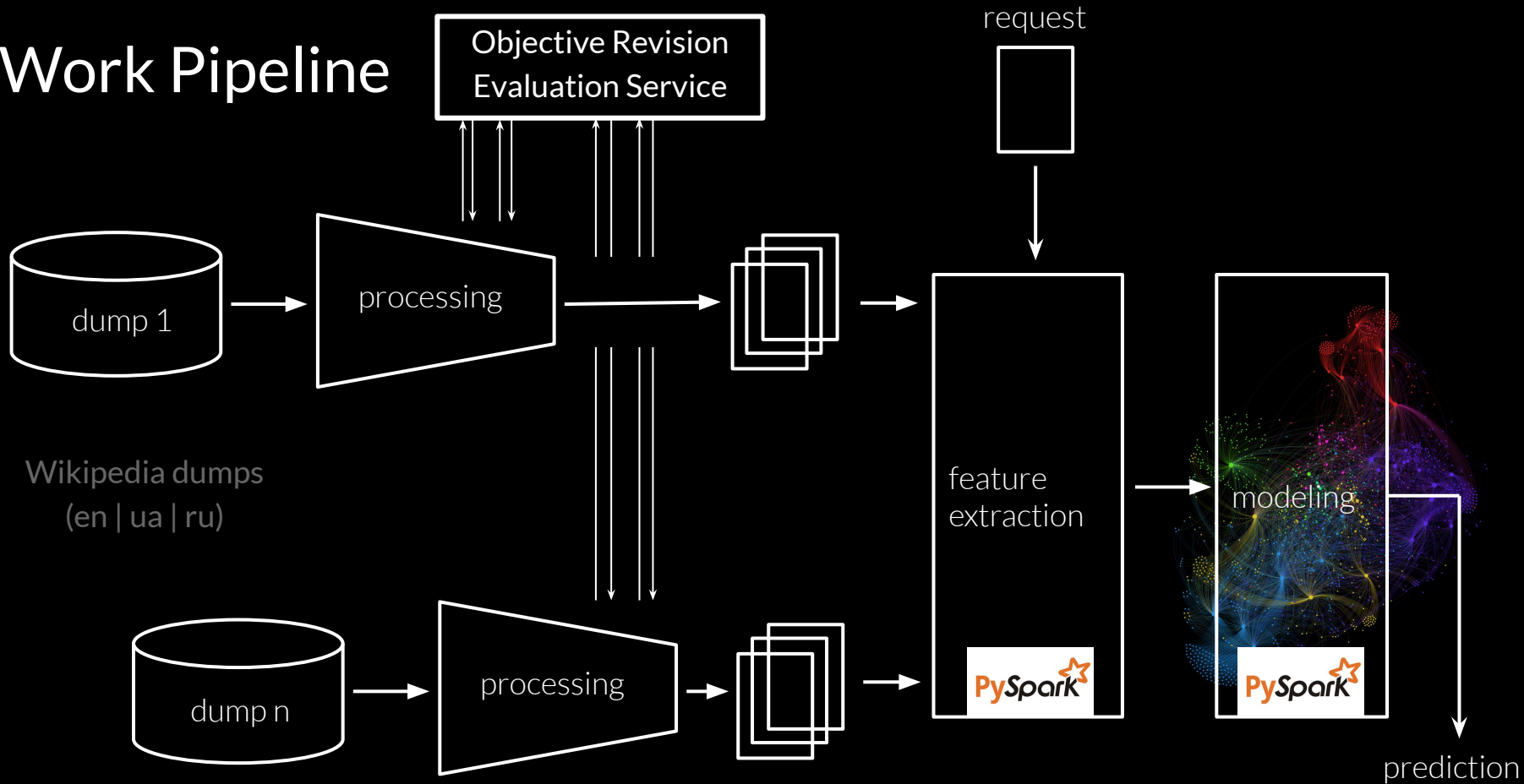


- the page is well cited and argued
- the references distribution:
  - 55%: scientific papers
  - 40%: books
  - 5%: software documentation
- this is likely a featured article

# Motivation

- Automatic detection of poorly written and poorly referenced articles will help editors to focus on the truly non-filled articles
- The differences in the same article across languages can be detected
- Possible output of important features should help Wiki-editors to concentrate on the most important gaps of the article

# Work Pipeline



# Data Processing

- Download wikipedia XML dumps
  - We are using page article multistream dumps. To get faster development loops so far we worked with a single dump, next we will run the full pipeline on the whole wikipedia data.
- Parse XML to CSV using streaming XML parser
  - We are using lxml and handwritten parser that goes through the file tag by tag and parses articles and meta information and article and last revision. The data we are fetching includes article text, title, revision author, revision comment and timestamp
- Fetch ORES assessments
  - Here we use mwapi and ORES web service to get article scores
- Inspect internal structure of text
  - Wikipedia articles have it's own syntax for declaring blocks inside article: [source](#)

★ FA	Professional, outstanding, and thorough; a definitive source for encyclopedic information.
🔍 A	Very useful to readers. A fairly complete treatment of the subject. A non-expert in the subject would typically find nothing wanting.
⊕ GA	Useful to nearly all readers, with no obvious problems; approaching (but not equalling) the quality of a professional encyclopedia.
B	Readers are not left wanting, although the content may not be complete enough to satisfy a serious student or researcher.
C	Useful to a casual reader, but would not provide a complete picture for even a moderately detailed study.
Start	Provides some meaningful content, but most readers will need more.
Stub	Provides very little meaningful content; may be little more than a dictionary definition. Readers probably see insufficiently developed features of the topic and may not see how the features of the topic are significant.

# Feature Engineering

Were built features that reflect diversity of text, sources and links:

- Internal, external references count
- Internal, external references count / Number of paragraphs (Average number of references per block of text)
- Citations count and ratios (Journals, Books, Web, News, etc)
- Number of images, files, etc in the articles
- Number of non-approved references (“citation needed”)
- Headings count
- etc

```
df_features.printSchema()
```

```
root
|-- sha1: string (nullable = true)
|-- timestamp: timestamp (nullable = true)
|-- title: string (nullable = true)
|-- text: string (nullable = true)
|-- Stub: double (nullable = true)
|-- Start: double (nullable = true)
|-- C: double (nullable = true)
|-- B: double (nullable = true)
|-- GA: double (nullable = true)
|-- FA: double (nullable = true)
|-- n_words: integer (nullable = false)
|-- level2: integer (nullable = false)
|-- level3: integer (nullable = false)
|-- level4: integer (nullable = false)
|-- level5: integer (nullable = false)
|-- level6: integer (nullable = false)
|-- book_citations: integer (nullable = true)
|-- journal_citations: integer (nullable = true)
|-- n_internal_links: integer (nullable = false)
|-- n_external_links: integer (nullable = false)
|-- n_paragraphs: integer (nullable = false)
|-- n_unreferenced: integer (nullable = false)
|-- n_categories: integer (nullable = false)
|-- n_images: integer (nullable = false)
```



# Modeling

## Clustering algorithm: Bisecting K-means

The algorithm starts from a single cluster. Iteratively it finds divisible clusters on the bottom level and bisects each of them using k-means, until there are  $k$  leaf clusters in total or no leaf clusters are divisible.

- + hierarchical top-down approach
- + parallelism
- + high speed and efficiency (in terms of entropy, F measure and overall similarity)
- needs a hyperparameter  $k$  — fixed number of clusters — as input; but can be solved by maximizing the likelihood of the evaluation metrics

# Evaluation

Silhouette coefficient measures how appropriately data have been clustered

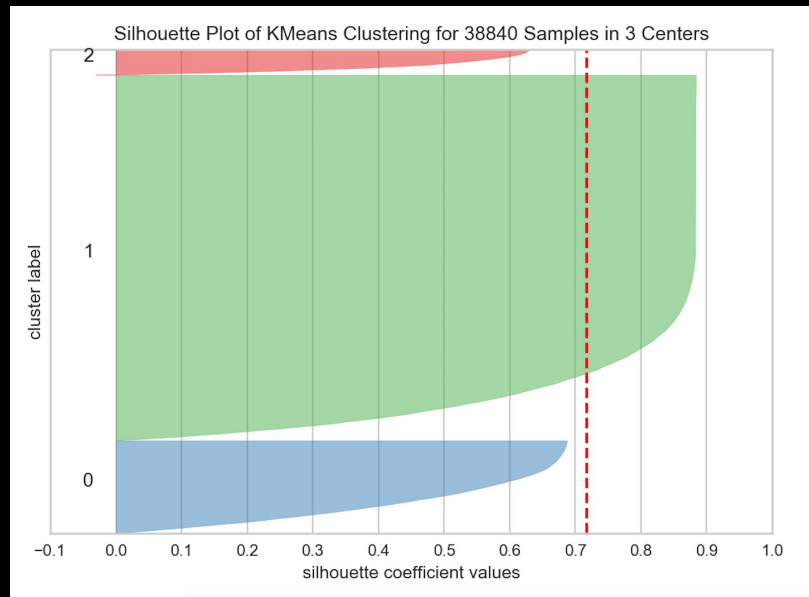
It is calculated using

- mean intra-cluster distance **a**
- mean nearest-cluster distance **b**
- final score is  $(b - a) / \max(a, b)$
- score is between **-1** and **1**

} for each sample

Current result (average for each cluster):

**0.65; 0.68; 0.89**



# Further work

- Create supervised machine learning model with ORES as labels and our features as inputs. The resulted model should be transferred to the other languages that didn't support by ORES (like Ukrainian)
- Test the PySpark MLP / Random Forest / Gradient boosting and get the **features importance** (visualize the pluses and minuses of the articles references)
- Fix clustering with ORES features