Final Report

Markiyan Kostiv, Yurii Mykhalchuk, Irynei Baran, Oleh Pidhirniak

The main goal of the project was to create a ranking of articles in uk.wikipedia.org that are missing in en.wikipedia.org. During our research, we developed a classifier and ranking criteria for Ukrainian articles missing in English Wikipedia.

Data

Wikipedia APIs allowed us to gather 9 article features. Using **sqlalchemy** from **PAWS** we managed to get the following features:

- len -- a length of the article
- rev_count -- revisions count
- image_links_count -- count of links to graphical images in an article
- translations_count -- a number of article translations from Ukrainian to other languages
- days_old -- a number of days since first revision until 18.07.2018
- contributions -- a number of unique contributors (by username)
- outgoing_links_count -- a number of links from article to other articles
- incoming_links_count -- a number of links to this article from other articles

Also, we built a scraper using MediaWiki AI P(https://wikimedia.org/api/rest_v1/metrics/pageviews/per-article/uk.wikipedia.org/all-access/all-agents/{article_name}/monthly/20180101/20180718) that allowed us to fetch articles monthly views from 01.01.2018 to 18.07.2018. We averaged views count and considered it as a feature:

- average_views

Preliminary analysis

To have some grasp on how good mined features are, we calculated correlation matrix on table 1.

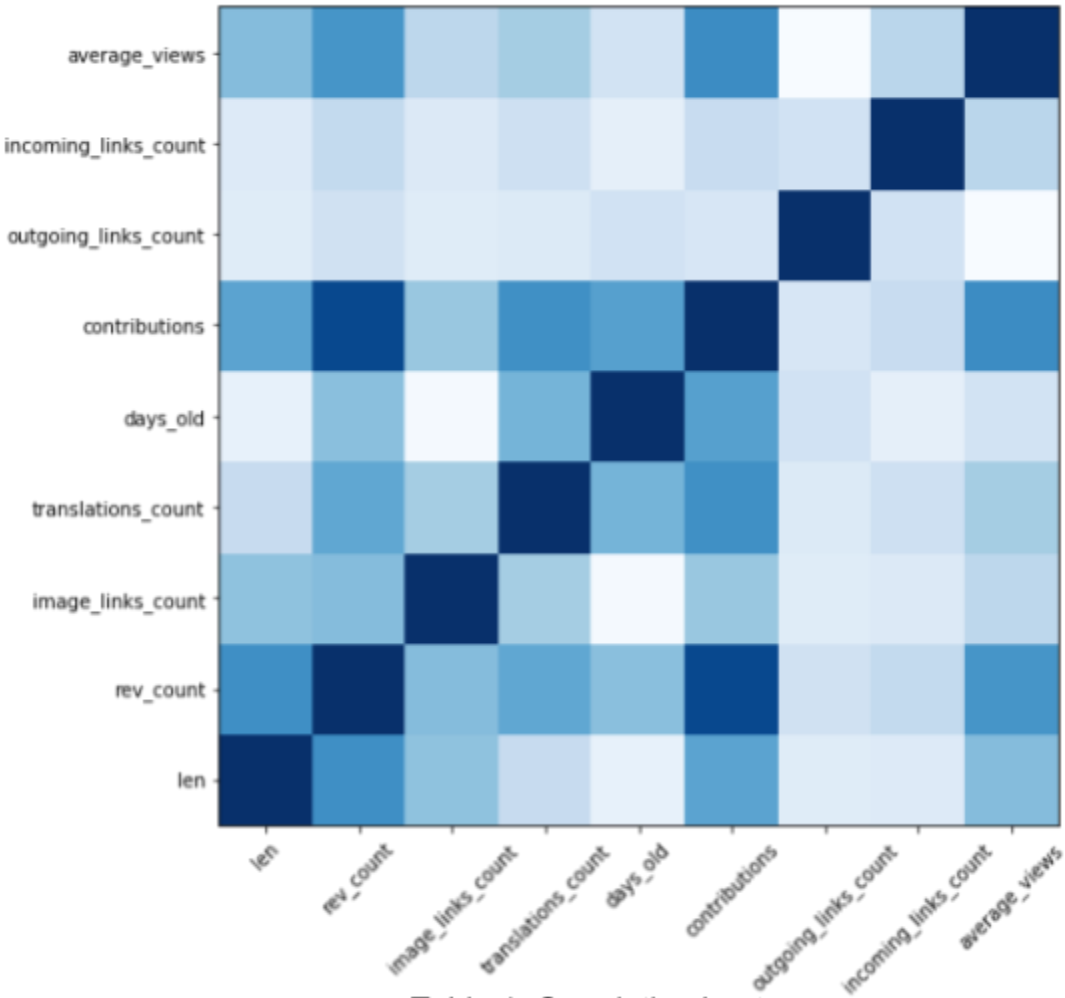

Table 1. Correlation heatmap

Correlation table showed us that *rev_count* and *contributions* features are highly correlated. We decided to perform modelling without the *rev_count* feature.

After limiting our feature-space we performed a normalization of feature values. Division by maximum value was used.
The final dataset that was used for modelling consisted of 215k articles.

Modelling

The first step was splitting data into train and test sets. Train set consisted of 20% of the dataset so that we may be sure that the model is not repeating our labels during ranking. 70% (142k articles) of the train fell into the **translated** category, 30% (74k) were **untranslated**.

The first goal was to check how good a model is in predicting whether an article is already translated or not. For this purpose, we trained and tested three different models on the gathered data.

1. Dummy classifier - model was assigning category randomly

   Accuracy: *55.19%*

Real / Predicted	Translated Class	Not Translated Class
Translated Class	20103	38886
Not Translated Class	38281	74945

Table 2. Confusion Matrix for Dummy Classifier.

2. XGBoost

   Accuracy: *99.18%*

Real / Predicted	Translated Class	Not Translated Class
Translated Class	58053	936
Not Translated Class	481	112745

Table 3. Confusion Matrix for XGBoost Classifier.

3. SVM

   Accuracy: *69.80%*

Real / Predicted	Translated Class	Not Translated Class
Translated Class	21339	37650
Not Translated Class	14353	98873

Table 4. Confusion Matrix for SVM Classifier.

From the above results, we decided to proceed further only with XGBoost as it was giving the best predictions.

The second (and the main) goal was to create a ranking of articles. As there was no ground truth we were checking results manually. For this purpose we tested two models:

1. Since all our features were numerical, we decided to measure articles rank using geometric distance. We performed a 1-class k-means clustering to find “already translated” cluster’s centroid. The distance was defined as a cosine similarity between the desired article and cluster centroid. Unfortunately, using this approach we didn’t obtain meaningful results.
2. We decided to descendingly sort the dataset using a probability of an article to be translated given by XGBoost classifier. This approach gave us better results than the first one.

Results

The results of ranking not translated pages using XGBoost prediction are displayed in Table 5.

Index	Title	Rank
1	Даманові	0.99256825
2	Гребінцеві	0.9861579
3	Лінгвоцид	0.9849786
4	П'ятирічка	0.9833391
5	Сартлан	0.9812509
6	Риба-клоун	0.9788401
7	Су-35С	0.9770288
8	Кабаньяс	0.97626626
9	Горизонталь	0.9748097
10	IC-7	0.9741659
11	Горизонталь	0.9734605

12	Депривація	0.9723994
13	Витратомір	0.97228473
14	Квадрупл-дабл	0.97147244
15	Ерік	0.97099966
16	Протромбін	0.9686536
17	Кашалотові	0.96750265
18	Ушача	0.9669507
19	FTP-клієнт	0.9663811
20	Хаку	0.9629793

Table 5. Ranking results.

As we can see there are really important articles in the top 20 like Five-Year Plan, FTP-client, quadruple-double, etc.

Evaluation

Our hypothesis is that the probability that our model predicts for the article to be translated - is a measure of the ranking. To evaluate our idea we took a random sample of 1000 translated articles and merged them with not translated. After that we applied our ranking and selected the top-1000 articles. After we merged the results, we got 92.9% recall, which means that 929 of 1000 truly translated pages hit the top-1000. The results can be observed in modeling.ipynb

References

Code - GitHub repository with data extraction, data preprocessing and modeling code.
- Data_extraction - notebooks used to get data from Wikipedia APIs (PAWS/Wikimedia).
- Data_preprocessing - python script to merge all features into one dataset.
- Modeling - main notebook with modeling and results.

Data - Google Drive with all the data in csv format that we have used.