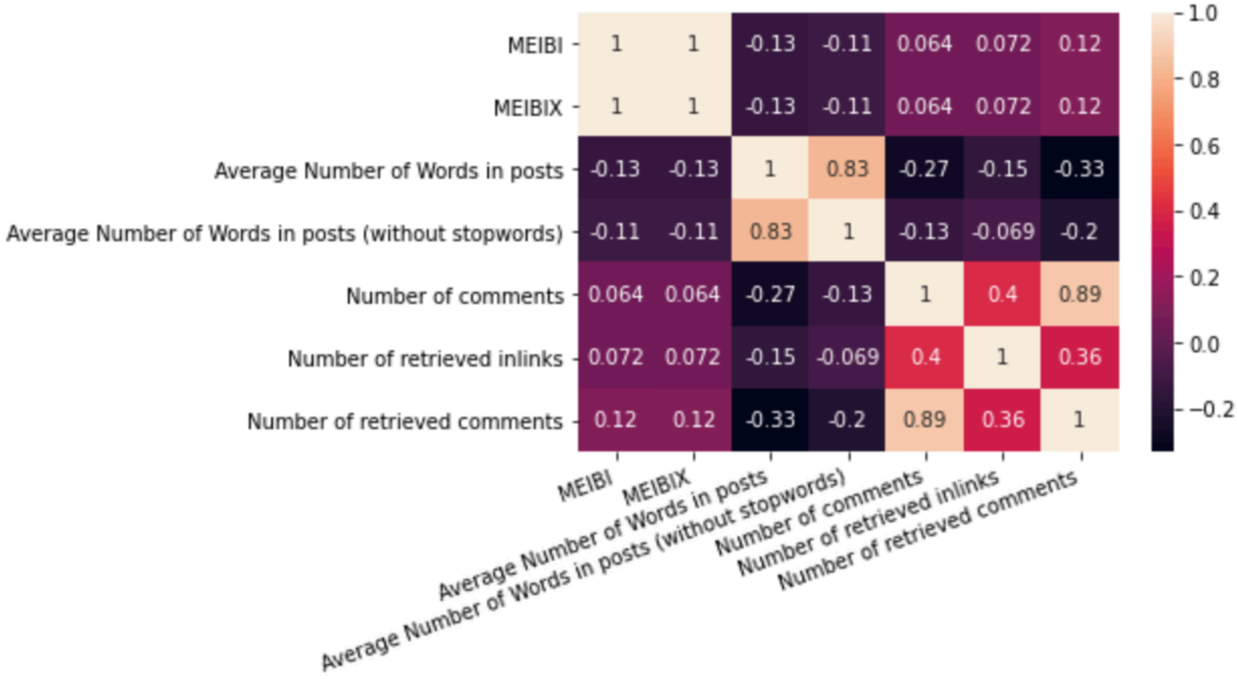
Blogger segmentation: the final report

The purpose of this report is to present the key findings of the mini sprint on blogger segmentation.

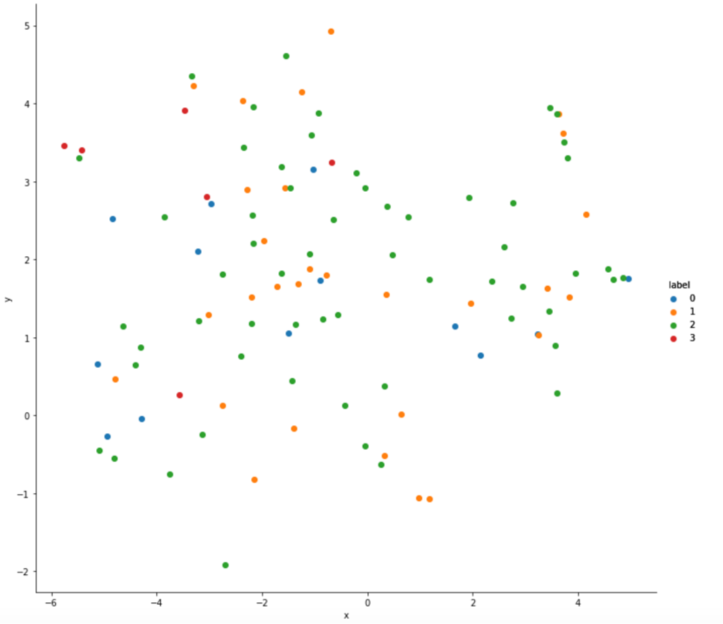
To approach the task, I combined two data sources, filtered out the (semi)-duplicating features, and ran multiple clustering rounds, both in terms of the direct handling the authors and while analyzing their content (articles). The latter part is in turn divided into sheer analysis of the numeric data, as well as NLP topic modelling, which aims to take an advantage of the article title. The final results prove to be good enough for the baseline modelling. I'll deal now in more details with every part of this research and will present the final conclusions in the end.

First of all, let's have a look at the cross-correlation distribution of the numeric features.



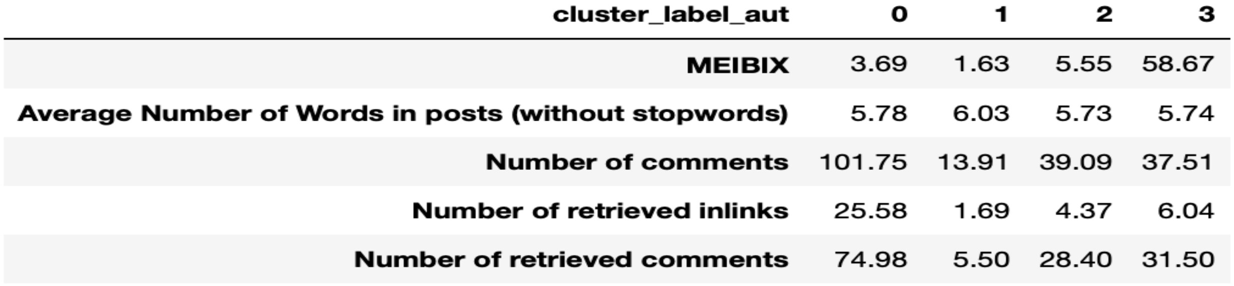
As evident, the difference between **MEIBIX** and **MEIBI** indices is minimal (2 values differ), I select the **MEIBIX** as a potentially more comprehensive index, while leaving **MEIBI** out. The situation with the fields **Number of comments** and **Number of retrieved comments**, but these can be directed at two sides, i.e. incoming and outcoming comments, thus I keep them both. Also, the same concerns **Average Number of Words in posts** and **Average Number of Words in posts (without stopwords)**. After looking at the distribution of these two, I applied BoxCox transformation and kept **Average Number of Words in posts (without stopwords)** only. The remaining features I log transformed to achieve better values distribution. More details on this regard can be found in the research notebook.

Now, having done basic preprocessing, let’s start with clustering. I start by taking an average value of all data per column from posts dataframe (which is a part of the combined dataset) in terms of the Author ID. To illustrate the clustering performance, TSNE dimension reduction is made. The clustering algorithm is k-means, and the number of clusters identified based on the **Elbow method**. The results are shown below. As seen the clustering produced 4 clusters, which are however not completely distinguishable.



While keeping in mind that reduction dimension may not accurately represent the multi-dimensional space points, it still gives some intuition as to how the points are located in the space. To evaluate the performance of the algorithm I used **Silhouette Coefficient** and **Davies-Bouldin Index.** The former shows the inter-cluster relationships, while the latter – intra-cluster differentiation. The results show the following: the value of 0.31 of the Silhouette Coefficient tells us that points are located within the right cluster like 'okay', nothing too spectacular, while the value of 1.05 for Davies-Bouldin Index shows that some clusters may stand close enough to each other, i.e. not perfectly separated. Though it's not that bad, given the score may go well beyond 1.

After preforming the reverse values transform here’re our aggregation results:



cluster_label_aut	0	1	2	3
MEIBIX	3.69	1.63	5.55	58.67
Average Number of Words in posts (without stopwords)	5.78	6.03	5.73	5.74
Number of comments	101.75	13.91	39.09	37.51
Number of retrieved inlinks	25.58	1.69	4.37	6.04
Number of retrieved comments	74.98	5.50	28.40	31.50

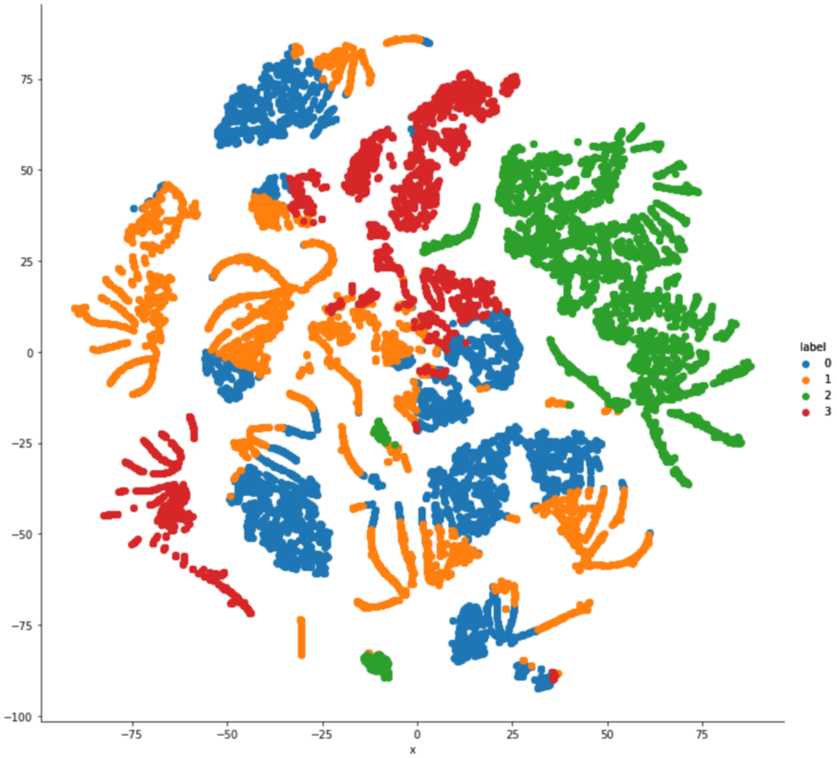
The following conclusions can be made from these data:

- the **cluster 3** is least populated, yet has the highest mean values against MEIBIX. At the same time, the number of retrieved inlinks is second to highest, just as the number of retrieved comments.
- **cluster 2** and **cluster 1** are the most populated ones. **cluster 1** has worst impact, and worst number of comments and inlinks.
- If you compare side by side **cluster 0** and **cluster 2**, you see that: a. **cluster 0** is an everage performing cluster, while **cluster 2** - above average. The latter has 34% more impact while having between 1.6 and 4 times fewer comments and inlinks. It must be the quality of their posts that adds to such a performace.

In general, the majority (58 authors) belong to **cluster 0**, and **cluster 1** (30 authors), while **cluster 3** is underpopulated (6 authors).

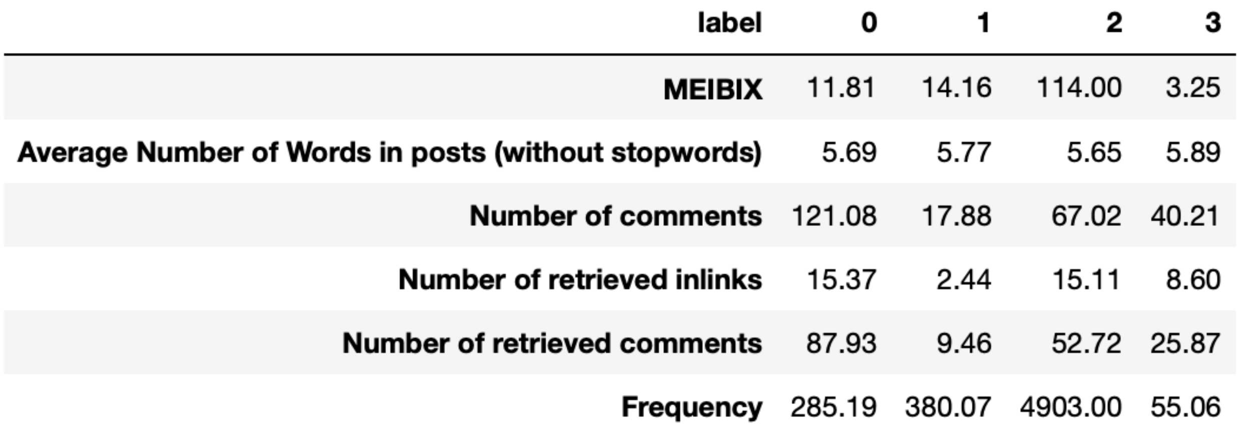
Let's see how the article-level clustering will perform.

First of all, I am going to add the frequency of article posting, and will add those data to the dataset. Next repeat the k-means clustering and TSNE representation.



The number of observations is much bigger in this case, and in ovewrall the more clear cluster representation is evident. There's still a slight problem with cluster 3 cutting through the space, but given the multidimensionality of the data this is permissable as the baseline. Also, the data distribution is much more balanced with no cluster being underpopulated.

Speaking about the clustering performance, it now scored a little bit less than last time (0.28 and 1.41 respectively). In overall the numbers are comparable between the two clusterings, so the former explanation holds true. However, since the values worsened a bit tells us that with increasing number of observations our confidence rises, but the potential noise rises too. In our case, now the clusters are distributed much more balanced, although the biggest overlap ma take place between **cluster 1** and **cluster 2** // **cluster 0** and **cluster 3**. We'll see now what they stand for in a moment.

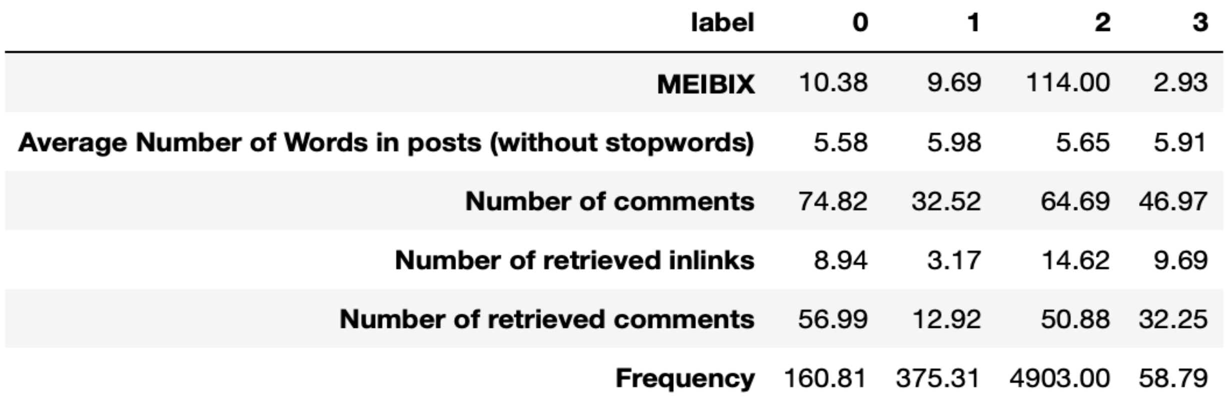


label	0	1	2	3
MEIBIX	11.81	14.16	114.00	3.25
Average Number of Words in posts (without stopwords)	5.69	5.77	5.65	5.89
Number of comments	121.08	17.88	67.02	40.21
Number of retrieved inlinks	15.37	2.44	15.11	8.60
Number of retrieved comments	87.93	9.46	52.72	25.87
Frequency	285.19	380.07	4903.00	55.06

Essentially **cluster 2** shows the more extreme values. These are bloggers who are most impactful and most frequent, with average comments though and high retrieved inlinks. Followed by **cluster 1** and **cluster 0**, who have are quite impactful, but express interesting peculiarities: **cluster 1** is 25% more frequently posts, 14% more impactful, yet much much less involving into comments and links retrival (between 5 and 8 times on average). These people should be very efficient at how they approach the audience; **cluster 3** has a low impact and low frequency, with other metrics being on pair with other clusters. It seems they have problems with how they approach the audience.

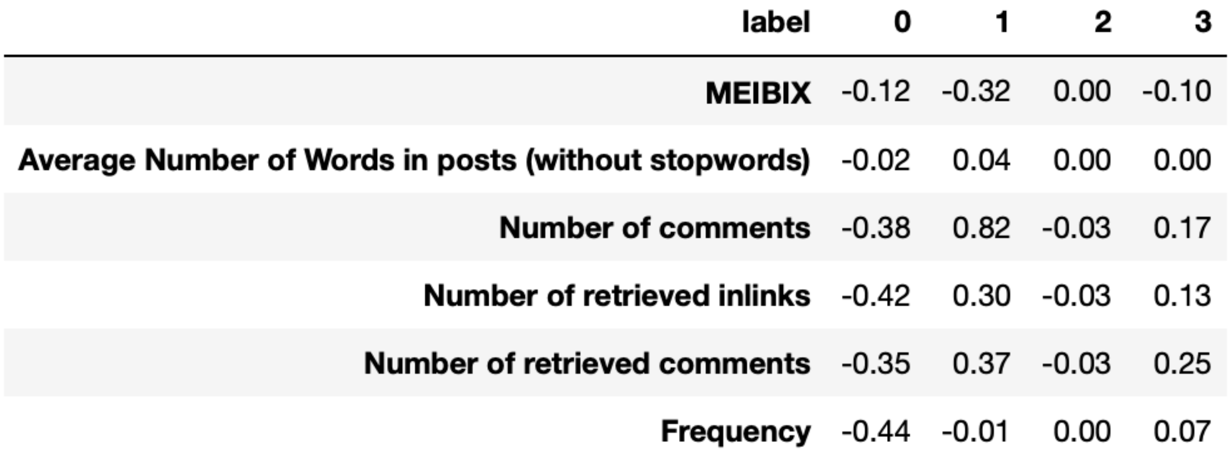
But I think since the same author may write for different topics, or even express a varying behavior, thus belongling to the different clusters at the same time (since here we analysed the articles first of all, not directly bloggers) may impact the results.

Next we will limit the bloggers to a single most frequent cluster.



label	0	1	2	3
MEIBIX	10.38	9.69	114.00	2.93
Average Number of Words in posts (without stopwords)	5.58	5.98	5.65	5.91
Number of comments	74.82	32.52	64.69	46.97
Number of retrieved inlinks	8.94	3.17	14.62	9.69
Number of retrieved comments	56.99	12.92	50.88	32.25
Frequency	160.81	375.31	4903.00	58.79

And compare these with the former table to see the relative changes.



label	0	1	2	3
MEIBIX	-0.12	-0.32	0.00	-0.10
Average Number of Words in posts (without stopwords)	-0.02	0.04	0.00	0.00
Number of comments	-0.38	0.82	-0.03	0.17
Number of retrieved inlinks	-0.42	0.30	-0.03	0.13
Number of retrieved comments	-0.35	0.37	-0.03	0.25
Frequency	-0.44	-0.01	0.00	0.07

So, after segregating we were able to assign to each blogger only one most frequent cluster. The results are surprising enough.

1. It seems we have only 1 author with the striking performance in **cluster 2**. A true performer and the Master Jeddi! Didn't suffer from segregation.
2. Comparing the two statistics we can state that the majority of bloggers beling to **cluster 3** and **cluster 0**.
3. In this respect **cluster 3** is especially interesting case, since these're worst performing bloggers. Thus limiting the choice to one cluster only further degraded their performance**. cluster 1** is just an average performance, and these bloggers are quite many too. They lost about 32% in impact, but gain between 30% and 80% in comments and inlinks.
4. Cluster 0 is above average performance. They lost between 35% and 42% of comments and inlinks, and by 44% in frequency. This means they indeed expressed the varying behavior and thus had articles belonging to different clusters.
5. After segregaion, we see that **cluster 0** is impacting only slightly better (by 7%), but commenting by between 1.3 and 3.4 times more, while being 57% less frequent to write posts.

6. In general, limiting the bloggers to one cluster only degraded the impact slightly, but still preserved the set above trend.

   (please see the notebook for more details.)

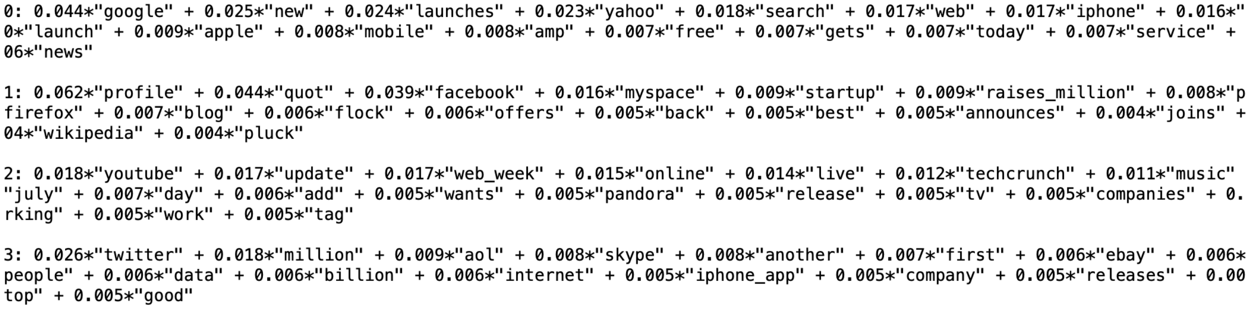
However, the biggest drawback and limitation of the two current approaches is the lack of insights of from the topic orientation of the bloggers. In order to find out which topics the authors mostly specialize in, topic categorization using the NLP instruments will be performed below.

Thus, now we try the topic modelling of the article titles using the LDA method – Latent Dirichlet Allocation. In natural language processing, the latent Dirichlet allocation (LDA) is a generative statistical model that allows sets of observations to be explained by unobserved groups that explain why some parts of the data are similar. For example, if observations are words collected into documents, it posits that each document is a mixture of a small number of topics and that each word's presence is attributable to one of the document's topics. LDA is an example of a topic model and belongs to the machine learning toolbox and in wider sense to the artificial intelligence toolbox. For this task, we'll use only English articles.

We will perform the following steps:

- **Tokenization**: Split the text into sentences and the sentences into words. Lowercase the words and remove punctuation.
- Words that have fewer than 3 characters are removed.
- All **stopwords** are removed.
- Words are **lemmatized** — words in third person are changed to first person and verbs in past and future tenses are changed into present.
- Words are **stemmed** — words are reduced to their root form.

In the end, this is what we were able to get. It appears that all topics – while doing research I tried different number of topics between 4 and 10 – that the topics are very, very similar.



```
0: 0.044*"google" + 0.025*"new" + 0.024*"launches" + 0.023*"yahoo" + 0.018*"search" + 0.017*"web" + 0.017*"iphone" + 0.016*"
0*"launch" + 0.009*"apple" + 0.008*"mobile" + 0.008*"amp" + 0.007*"free" + 0.007*"gets" + 0.007*"today" + 0.007*"service" +
06*"news"

1: 0.062*"profile" + 0.044*"quot" + 0.039*"facebook" + 0.016*"myspace" + 0.009*"startup" + 0.009*"raises_million" + 0.008*"p
firefox" + 0.007*"blog" + 0.006*"flock" + 0.006*"offers" + 0.005*"back" + 0.005*"best" + 0.005*"announces" + 0.004*"joins" +
04*"wikipedia" + 0.004*"pluck"

2: 0.018*"youtube" + 0.017*"update" + 0.017*"web_week" + 0.015*"online" + 0.014*"live" + 0.012*"techcrunch" + 0.011*"music"
"july" + 0.007*"day" + 0.006*"add" + 0.005*"wants" + 0.005*"pandora" + 0.005*"release" + 0.005*"tv" + 0.005*"companies" + 0.
rking" + 0.005*"work" + 0.005*"tag"

3: 0.026*"twitter" + 0.018*"million" + 0.009*"aol" + 0.008*"skype" + 0.008*"another" + 0.007*"first" + 0.006*"ebay" + 0.006*
people" + 0.006*"data" + 0.006*"billion" + 0.006*"internet" + 0.005*"iphone_app" + 0.005*"company" + 0.005*"releases" + 0.00
top" + 0.005*"good"
```

It's not good, since such a situation will not allow to extract meaningful insights from the article names – more interactive illustrations are in the notebook.
After assigning the obtained topics to authors it appeared all but one authors wrote for the same topic. Thus, this approach did not add any value, but still explored different instruments to tackle the issue.

In overall, the clusters received in the previous part, may experience overlapping and this might be the biggest challenge to overcome with subsequent finetuning. However, I believe the solution presented in this research notebook satisfies JBGE criterion (Just Barely Good Enough) of the Agile approach for the baseline solution stage.

By and large, I suggest using the second approach – i.e. based on the articles numeric values clustering, while subsequently averaging the data by the Author ID – to segment the bloggers!

As for the future steps, I suggest trying some more clustering techniques, for example agglomerative clustering with different linkage criteria, as well as considering the complete articles text to retrain the NLP model. Adding more data may prove useful too. I would start the next iteration with these steps.