Latino-Asian-Brotherhood

Khuong T.G.H. - Alex-Răzvan Ispas - Diego-Andres Torres - Guarin

November 2022

1 Introduction

In this challenge we are given the task of determining relevant information that can be of interest for corporate clients of Bancolombia. In this context, we want to provide the bank specialist with news articles that mention the client and/or important aspects for its business sector. At the same time, it's important that we know in what sense the given news article is relevant for the client: does it mention an recent accomplishment of a competitor? is it about macroeconomic factors that may reduce the demand for its products? Finally, we should be able to rank our recommended news so that the bank specialist can pick the most relevant information according to their time constraints.
In order to create a system that is able to perform these tasks, we are provided with three csv files, whose structure can be seen in figure 1.

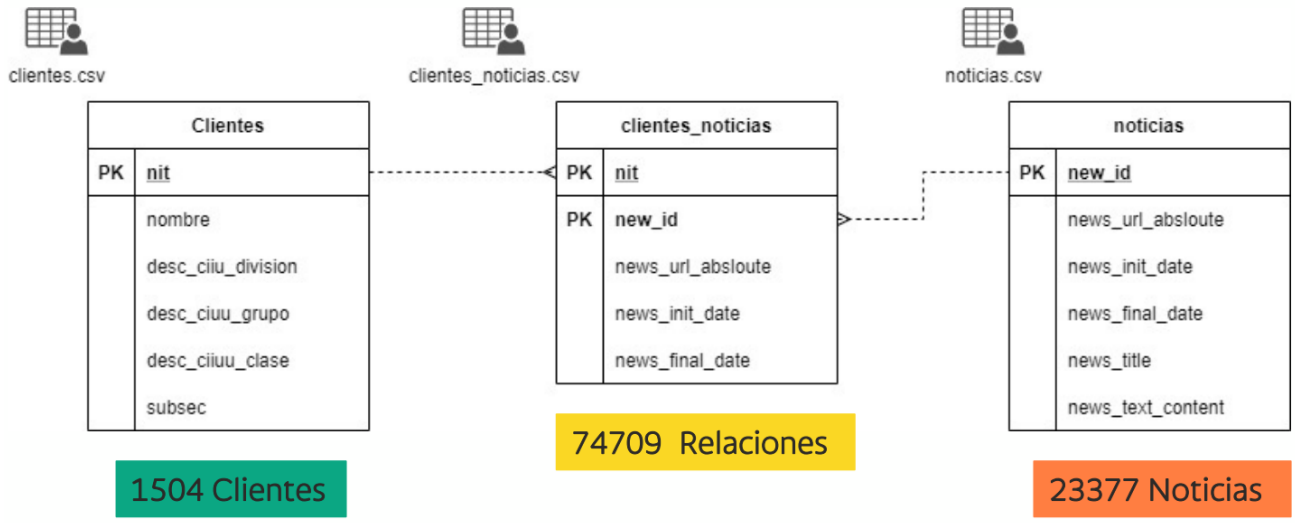

Figure 1: UML diagram of the dataset. The *clientes_noticias* table serves as the link between each client and the set of news that we are expected to evaluate for that client.

Before going any further, it's important to take a look at the data to see what kind of information is actually present. In figure 2, we show the most common words present in the field *nombre* of the **clients.csv** file. The most common terms turn out to be the ones that carry information about the structure of the company, but that won't be of much relevance in the task. Then, from this exploration we derive a first task to perform, which is cleaning the names from these irrelevant terms. We also explore the title of the news articles. After removing stopwords in the Spanish language, we can visualize the most recurring words. As expected, Colombia is the most frequent term, followed by things like government, Gustavo Petro, new, etc. This makes clear the period in which the news were collected, right after the election of the new president. It also gives a first hint about the fact that most news will have to do with macroeconomics and regulations.

The whole task is separated in three different subtasks: participation, categorization and recommendation. In the participation task, the goal is to determine whether a news article is explicitly relevant for the client, for its sector (but not directly connected to the client), or neither of those. These scenarios are labeled as: *Cliente*, *Sector*, *No Aplica*. For categorization, the task is somewhat similar, but in this case we only intend to categorize the news regardless of the client relevance. The labels have to do with the type of news article: *Macroeconomía*, *Sostenibilidad*, *Innovación*, *Regulaciones*, *Alianza*, *Reputación*, *Otra*, *Descartable*. Finally, in the recommendation task, we are asked to order the news for each client according to the relevance, so that more useful news are ranked higher.

Figure 2: Word cloud for the names of the clients, we can see that there are very common terms related to the type of company or the fact that they are operating in Colombia.

Figure 3: Word cloud for the titles of the news, after removing spanish stopwords.

2 Formalization of the problem

Given the tasks and the data that we have, we state the three challenges as three different learning problems. For participation, we want to find a mapping that has as input the news article (both title and content) and the client (name and business sector), which outputs one of the three possible situations described above. In the categorization task we have a simpler, more conventional situation. We want to classify the news articles among 10 categories. As for the recommendation problem, we focus on finding a program that, for a given client and news article, gives a score that can be interpreted as the pertinence for the clients' interests. As such, it would simply be a matter of sorting the news for each client according to this score.

3 Challenges

3.1 Absence of labeled data

The biggest challenge that is shared among the three tasks is the lack of labeled data. Even though we have a relationship between news and client, it is only given by the results of Google News, which does not mean that the news is actually about the client it is paired with. In fact, by exploring some of the news data, it is easy to spot some news that don't have anything that could potentially be useful to any company. In the case of categorization and recommendation, the situation is even more challenging, as there is nothing to be used as a ground truth label for those tasks.

3.2 Participation

The most immediate challenge that we faced is the presence of filler words in the clients' names. We wanted to have a version of the name that was likely to appear in the news articles. However, it was difficult to make sure to match every case because of the variability in the names. And even then, some names reduced to common words in spanish that would match in many

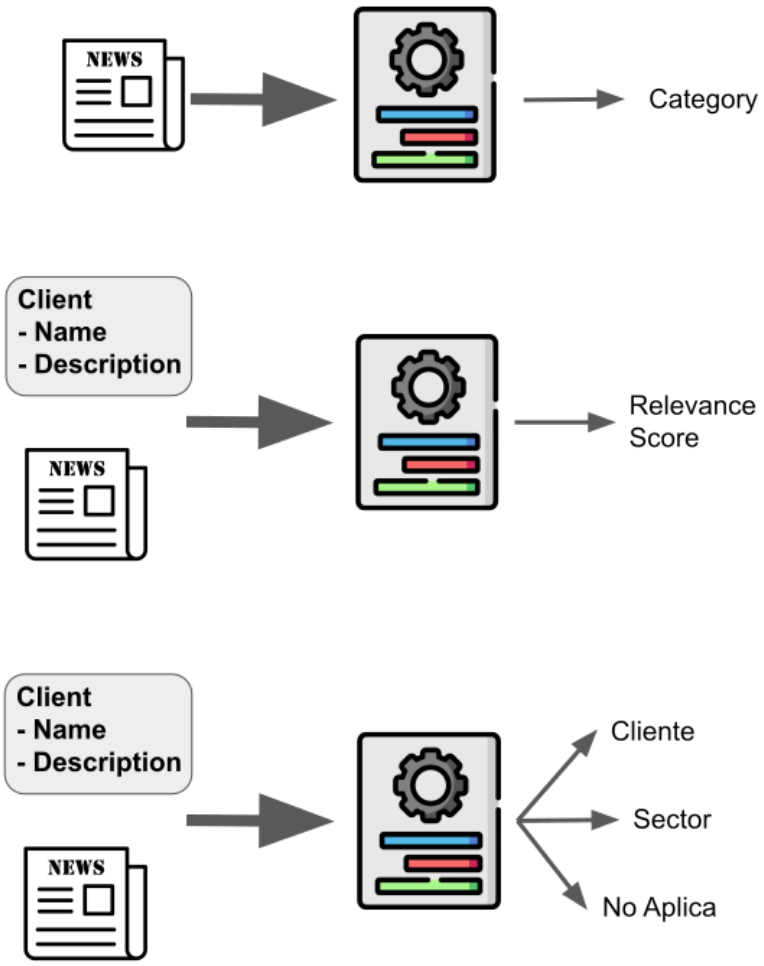

Figure 4: Formalization of the tasks as mapping problem from inputs and outputs

news. As examples of this we have: SI SAS→SI, PROTECCION SA→PROTECCION, or EFICACIA SA→EFICACIA. In this case, we took a pragmatic approach. We realized that there was a considerable inequality in the number of news paired with each client. This is due to the way the data were collected: smaller clients won't have as many associated news in Google. We can see this graphically in figure 5, where it is shown that less than one third of the clients has 50% of the associated news. Hence, we made sure to have a good matching algorithm for these top 365 clients, that were more likely to actually appear explicitly in the news. As for the clients with bad names within the top clients, we simply removed matching altogether. From a probabilistic perspective, our accuracy is more damaged due to the large number of false positives, than it is benefited from a small number of true matches.

3.3 Recommendation

Apart from the lack of labeled data, there is a more profound challenge in this task. If we look at the categorization tasks, we are given from the beginning 8 categories that are understood rather easily by most people. On the other hand, the concept of a recommendation score is more subtle. The organizers provided us with 3 important questions that the recommender should aim to answer, but still we needed to find a way to formulate this piece into an objective metric that could be trained and/or computed automatically.

4 Methodology

In classic machine learning, the models rely on large amounts of data which are annotated. Due to the fact that hand annotation would require a large amount of time, we perform a method which is relying on a small amount of hand-labeled data called **few-shot learning**

Few-shot learning could also be perceived as a **meta-learning problem** as it tries to learn how to learn to solve the problem. The main idea of the pipeline is to portrayed in Figure6. First of all, a small number of annotated samples is being

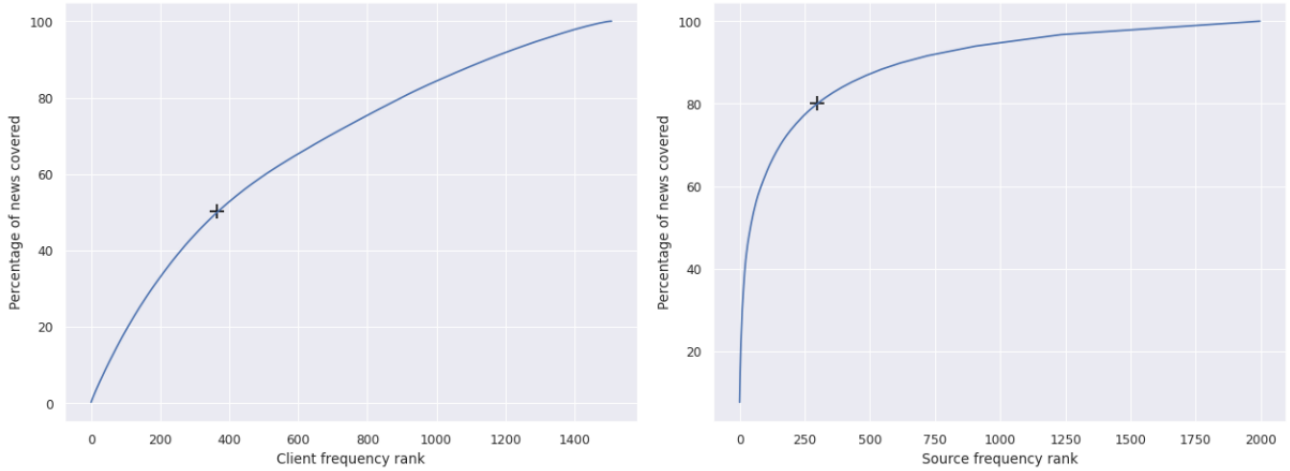

Figure 5: Cumulative percentage of news covered by considering the most frequent clients first

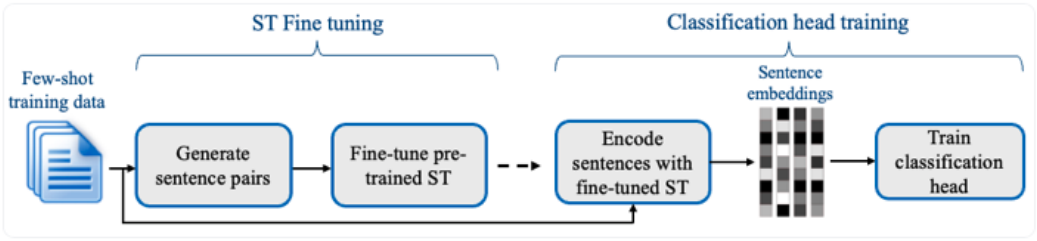

Figure 6: Few-shot learning approach to fine tune a transformer model in a classification problem

chosen. Afterwards, we are building *contrast pairs*. In the context of the given problem, news of different categories are paired together in order to create contrast in the learning process. Afterwards, dense embeddings are created with the help of the generated paired sentences from the previous step. The classification head will be trained on the encoded embeddings in report to their labels. Due to the fact that it showed promising results on classification in German, French and Spanish, CosineSimilarityLoss was selected as loss function [1].

4.1 Participation

Figure 7 summarizes very well our approach for this task. After we have cleaned the names of the clients of irrelevant filler words, we find the matches inside the news titles or content. For every pair (news, client) match, we create a training example (news, sector group). The sector group is build from the divisions of CIIU, the idea is to get broader labels that cover more types of businesses. Hence, the sector group associated with the news articles is that of the client's CIIU. This is the label that we fine-tune the SentenceTransformer to predict.
At the moment of prediction, we compare the sector group given by the transformer with that of the client's CIIU. If they differ, that means the news is not relevant for the client nor its sector, and that news is labeled as *No Aplica*. If the predicted group coincides with the client's, we check whether its name is present. If it isn't, we label *Sector*, if it is, *Cliente*.

4.2 Categorization

In figure 8 we show the procedure for training the categorization task. We manually label an initial set of 30 news articles according to our criteria. This dataset is used to fine-tune a sentence transformer previously trained on sentence similarity. We employ again human labeling on the mistakes made by this model. This serves as a second round of training that makes the model more robust and helps correct biases present in the first dataset. After this, we get a good model that is able to classify the news articles into the 8 desired categories.

4.3 Recommendation

Our recommendation methodology is fairly simple. First, we extract the source name from the news urls and count the number of times that they appear. We find a very interesting result in this part. Similar to Pareto's law [2], roughly 15% of the sources is responsible for 80% of the news. Based on this information, we decide to keep only that 15% (298) of sources.

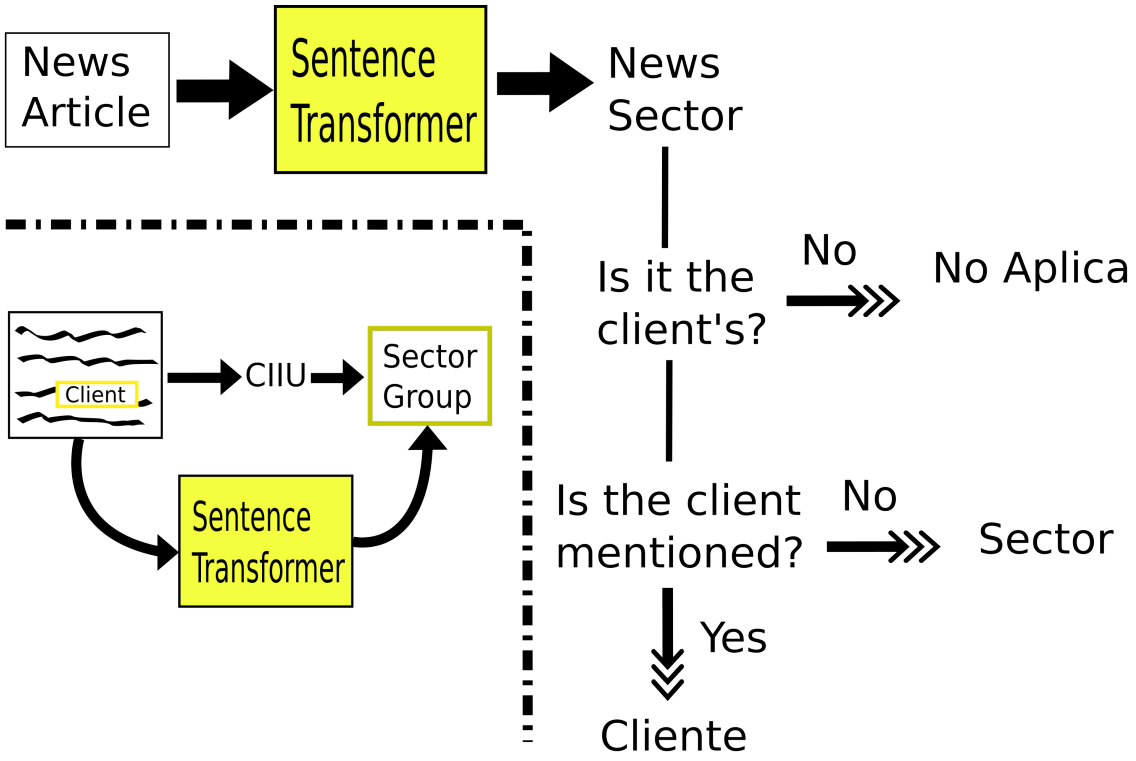

Figure 7: Training and prediction of participation

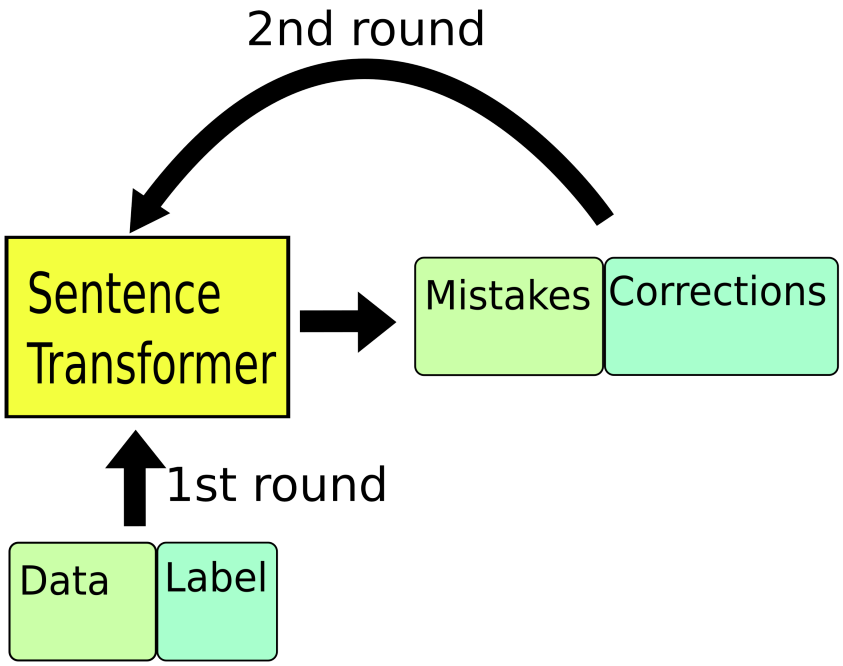

Figure 8: Training procedure for the categorization task. We use human labeling in 2 rounds to fine tune a sentence transformer.

In practice, this means that we will only recommend news coming from sources within this subset. After this, we use human labeling to remove other sources that may not be of interesting because they are foreign websites or centered on sports only. Additionally, we perform some manual permutation to put on the first priority the biggest news websites. Finally, according to its final position we assign a score between 0.5 and 1 to the sources.

As we can see in equation 1, the recommendation score is given by two parts. The first one is the source reputation score, which is described in the previous paragraph. The other part is the confidence that the sector transformer has on the particular sector of the client. This means that if the transformer is very confident that a news article is about a certain

sector, then this article should have a big priority for clients inside that sector.

$$\text{Recommendation Score} = \text{Confidence on client's sector} * \text{Source reputation score} \quad (1)$$

5 Results and further work

5.1 Participation

We were able to train a very effective model that categorizes the most relevant economic activity in a news article. However, the explicit mention of the client proved to be difficult to solve for every single client, as some of them had names easily mistaken as regular words in Spanish. Even though this may seem as not a complete solution, it is important to keep in mind the context of the competition and the possible context in production for our model.
In the real world scenario, our tool would be used by a sales person that knows very well her/his client and is dedicated to its well-being. Then, he/she will know for sure the names or labels people use to refer to the company. Take the case of *Grupo Mayorista* as an example. According to the data, the word *Grupo* is a filler word common to many companies. But in this case, it is likely that they are actually referred to using their whole name. Therefore, it will be enough to retrieve the client's economic activity and a set of possible mentions. With these two available, it's just a matter of using our model to check the sector relevance and whether any explicit mention is present in the news article.

5.2 Categorization

This is actually the task that we can say most confidently that we solved. Nonetheless, we acknowledge that there could be room for improvement. As described above, we manually labeled a small subset of news articles in order to use a few-shot learning technique. Taking into account that we are using a pretrained sentence transformer, we were able to train a very robust model, even from this small training set. As such, we can expect the results to be improved to some extent by using labeling performed by experts.
In order to so, it is necessary to alter 2 files in the repository, locate at *src/data/archivos_auxiliares*. The first one has the first round of learning examples to be fed to the model. The second one needs to be created by relabeling the model's mistakes, making special emphasis on the weakest categories predicted by the model. For instance, in the current state of the repository, you can see that the second file has more examples of Reputación, which was the category where we find more and stronger mistakes.

5.3 Recommendation

The confidence score of the model in the client's particular sector, multiplied by source score. We pick the top-5 highest score.

5.4 More data

One obvious way to improve the results is to gather more data and of better quality. We can see this effect in our result of participation, when the group with large amount of news matches (i.e. universities, banks, etc) has very good predictions. Whereas those with little data (i.e. motorbikes, tobacco, blankets, etc) have very bad predictions.
Another way could be to use other sources of data. For instance, information about a project that the company is carrying out can be extremely useful to recommend news that are of special interest. This also implies that recommendation can evolve based on new business ideas.

6 References

References

[1] Hugging Face. Setfit: Efficient few-shot learning without prompts. https://huggingface.co/blog/setfit, 2022.

[2] Wikipedia. Pareto principle. https://en.wikipedia.org/wiki/Pareto_principle.