UNIVERSITÀ DI PISA

Text Analytics Report

SectorSherlock

Classifying company descriptions into business sectors

Professor:
Laura Pollacci

Group members:
Irene Crepaldi - 674112
Alla Usova - 660973
Steffania Sierra Galvis - 663718
Alex Degaudenzi - 620171

Academic Year 2023/2024

Table of contents

1 Introduction

In recent years, machine learning and NLP have been largely used to try and tame the plethora of data that is available to us in order to extract meaningful information. Not only have these techniques been applied to explore the field of economics as a whole, but also the classification task has specifically been put to the test. As stated by [4], ”consistent classification of companies might be important for classifying a new company’s equity on the market or for the consistency of macroeconomic data aggregated across industries”. For our research, we have decided to use a dataset from Kaggle comprised of the descriptions that companies put on their LinkedIn pages as well as the business sector they belong to. A company’s description shows how it wants to be perceived by others – namely stakeholders, investors, and customers – and can give an insight into its general vision. Analysing the language that companies use to describe themselves and categorising them into the different types of industries to which they belong could be helpful in making us more knowledgeable consumers. The goal of this project is to identify linguistic patterns in companies’ descriptions in order to categorise them into different classes according to which industry they belong and eventually let keywords typical of each class emerge. We have used different techniques – namely gradient boosting and SVM classifiers, CNN and BERT deep learning algorithms – to categorize the companies’ descriptions into their respective business sectors as well as employed word and ngram frequency, word clouds and topic modelling to identify business sectors’ most informative and specific keywords.

2 Literature Review

NLP models and techniques have been employed copiously in the field of economics with varying degrees of complexity. [1] proposed a ”novel application of established NLP models within the economic domain” through the exploration of multi-class text classification; specifically classifying businesses by their International Standard Industrial Classification (ISIC) Code. Using word embeddings, [3] tested if companies belonging to the same industry sector cluster together: a large corpus of news, preprocessed with NLTK and Gensim, was used as input to train several classification methods using word2vec models. The aim of the task is to discover and analyze semantic connections between words as opposed to simply syntactic ones. In [4] the most interesting part we discovered is how the researchers in this paper organized companies into different groups based on their descriptions. To do this, they created a system where these groups formed a hierarchy, meaning there were broader categories at the top and more specific ones below. It it highly relevant to our case because it helps in understanding and organizing companies better, making it easier to classify the companies more accurately. Also, this paper used multiple deep learning models to achieve their classification and it served us as a footprint for the implementation of the BERT model. In [2] classification was explored into business sectors in a case similar to ours, although using a dataset extracted from Crunchbase, which presents several differences compared to LinkedIn. Their corpus, comprised of companies’ descriptions, was preprocessed using NLTK for punctuation removal, lower-casing, stopword removal, POS-tagging and bigram analysis; furthermore, they used bag-of-words representation (keeping a record of word frequency for each description) and tested tf-idf weighting, lemmatization and stemming. They took a non conventional way to label companies, and instead of assigning only one category to each company they allowed each company to belong to more than one category. Additionally, they trained two different kind of models, one binary, and a multi-class one. The first one consisted in create a model for each category and label companies depending on their labels in positive and negative samples. The second one had as purpose to be a recommendation model, i.e, to provide the label to which the company is more probable to belongs to.

3 Methodology

3.1 Data Preparation

We selected our data from the Kaggle page named ”LinkedIn Job Postings - 2023”, specifically the companies and company industries datasets. The companies dataset consists of the company id, name, size, description and location. Since our main focus is on NLP, we decided to use only the descriptions and discard the other information. The company industries consists of company id and the industry that the company belongs to, so that is going to be our target for the classification model.

The dataset contains 6000 rows and 141 unique company industries. Given this diversity, achieving precise classification appears to be highly challenging. That is why we decided to aggregate the classes to make it more general without sacrificing contextual significance. In order to do so, we first reviewed all the unique industries, then made a list of 15 industries that can represent them in a more general way and then made a mapping.

The results of that we've saved in the file ”industries mapping final.csv”

`industries_mapping_final`. An example is provided below:

Original Industry	**New Industry**
Logistics & Supply Chain	Logistics and Transportation
Wholesale	Logistics and Transportation
Transportation/Trucking/Railroad	Logistics and Transportation
Aviation & Aerospace	Logistics and Transportation
Maritime	Logistics and Transportation
Airlines/Aviation	Logistics and Transportation
Package/Freight Delivery	Logistics and Transportation
Warehousing	Logistics and Transportation
Import & Export	Logistics and Transportation
Fishery	Logistics and Transportation

Table 1: Example of Industry Mapping

That was the main step that we've done during the data preparation phase. We've also deleted the rows with an empty description so 57 rows were eliminated.

3.2 Data Cleaning

The raw data extracted from the dataset had to be processed before using it as input for the various models. The pipeline we implemented, using NLTK, was the following: tokenization; contraction removal; lowercasing; punctuation removal; stopword removal; Part-of-Speech tagging; lemmatization. In order to compute lemmatization correctly, it was necessary to change the Part-of-Speech tagset from the one we obtained to a simpler one comprised of only four tags: noun, verb, adj (adjective), adv (adverb). These were, then, taken as parameters for the lemmatization, along with their respective tokens. Later, word frequencies and probabilities were also computed.

3.3 Keywords by industry

3.3.1 Word frequency and ngrams

In order to assess the correctness of the data cleaning process, we decided to explore the most frequent words per industry using as input the cleaned descriptions. To make this, we divided the dataset grouping the data from each industry and calculated the frequency of the words with respect to all the descriptions of that industry. We then reported the most common words per industry. The same process has been repeated in order to obtain the most common bigrams and trigrams.

3.3.2 Word Clouds

In order to have a quick visualization of possible keywords typical of each business sector, we deemed it interesting to immediately compute word clouds. This was done by importing the wordcloud library which took in input the dataframe we created with raw companies' descriptions and business sector and directly produced the visual output. Later, we repeated this process using cleaned data instead of raw descriptions, even though the wordcloud library does some automatic cleaning of the data such as tokenization, stopword removal and lemmatization.

3.3.3 Topic Modelling

Before training some classification models, we thought that to train a topic model could be interesting and give to us an idea of the topics that can have higher relevance during the classification phase. An additional motivation for extracting topics for each industry is to think that those can be used as a kind of vocabulary or dictionary of suggestions for companies when writing their description.

When extracting the topics we used a **Latent Dirichlet Allocation (LDA)** model with the Gensim library. First we needed to specify which were the documents and the corpus to be used in our model. The documents were set up as the company descriptions tokenized plus the bigrams that appears 20 times or more in the description. Next, we removed the extremes words in every document, that is, the words that occurs in less than 5 documents and more than 50% of them. To create the corpus of each industry we represented its documents as a bag-of-words. Finally, to get the best number of topics to extract, we trained the model for every category several times iterating from 1 to 10, and choosing the number for which the model gets highest *coherence c_v value.* Finally, we would like to mention that the dataset used in this stage is the one obtained after the corresponding cleaning.

3.4 Balancing the data

The dataset was unbalanced with respect to the distribution of companies across different industries.

Industry	**Number of companies**
Business and Consulting	989
Education and Training	249
Financial Services	408
Government and Non-profit	295
Healthcare and Medical Services	661
Hospitality and Entertainment	290
Information Technology and Services	1064
Logistics and Transportation	185
Manufacturing and Engineering	695
Media and Publishing	207
Real Estate and Construction	290
Retail and Consumer Goods	351
Security and Law	129

Table 2: Number of companies in different industries

There are two main approaches to handle imbalanced data: oversampling and undersampling. An oversampling technique generates the data for the minority classes, while the undersampling entails eliminating records from the majority class.

We decided to use both approaches and their combination to check the performance and choose the best balancing approach. We started with the RandomOverSampler with default sampling strategy. So all the classes were oversampled up to 1064 observations. The next step was to balance the data with RandomUnderSampler, as with oversampling we haven't set any specific sampling strategy. Then we trained the DecisionTree model and compared the F1-score of the model on original dataset, the oversampled and undersampled versions.

F1 Score	**Number of companies**
Original data	0,497
Oversampled version	0,847
Undersampled version	0.424

Table 3: Performance of the balancing techniques

As evident from the table, oversampling significantly improved the F1 score. Additionally, we explored a combination of oversampling and undersampling. To implement this hybrid approach, a specific sampling strategy needed to be defined. After experimenting with various strategies, the most effective one involved first undersampling the majority classes ('Business and Consulting' and 'Information Technology and Services') down to 700 items and then oversampling all other classes to 700. Despite achieving an improved F1 score of 0.751 compared to undersampling alone, it still performed less favorably than oversampling. Consequently, we opted to utilize the dataset balanced solely through oversampling.

3.5 Classification

The main goal of our project was to build a model to predict the business types of the company based on its description from the LinkedIn page. To do so we needed to use the classification algorithms. In this section we are going to explore the various algorithms we have tested and present the one that is the most suitable for our project.

3.5.1 Gradient Boosting and SVC

The first two models trained were a *gradient boosting model* and a *linear support vector model*. We chose those models due to their high robustness. For both we used the balanced data obtained after the oversampling. First the data was split in a train and test data set of sizes 70% and 30% respectively, and after we used the Scikit-learn library to train the models. Since there were not too many parameter to set up, we decided to go with the default parameter values, that is, number of estimators = 100 and random state = 42 for the gradient boosting classifier. When training those models, we used directly the methods given by Scikit-learn library and not the pipeline because the oversampling model returns a dataset of vectors and not of descriptions.

3.5.2 Deep learning models

Deep Learning Models are different from the traditional machine learning ones. While a classification algorithm usually relies on structured data, Deep learning models use raw data as input which allows them to learn hierarchical representations and patterns useful for the classification task. We implemented two different models: Convolutional Neural Networks (CNN) and Bidirectional Encoder Representations from Transformers (BERT). To enable the classification, we used the descriptions – preprocessed as previously described – as input for both models.

Neural Networks

On the other hand, for the neural networks, we decided to work with the unbalanced data because we needed it to use the descriptions to extract some of the parameters. Four neural networks were trained, all of them convolutional networks with different parameters. To train these neural networks we used the libraries *Tensorflow*, *Keras* and *Transformers*. Our classification task is multilabel, so we had to use as activation function *softmax* and as loss function *sparse categorical crossentropy.*

In the table below we present the parameters used for the three convolutional models.

BERT

The implementation of BERT entails the use of a pretrained version of the model and, in

Parameter	**Model 1**	**Model 2**	**Model 3**	**Model 4**
Embedding layers	1	1	1	1
Convolutional layers	1	1	1	1
Dense layers	1	2	2	2
Optimizer	Adam	Adam	Adam	SGD
Pooling	GlobalMax	GlobalMax	GlobalAvg	GlobalAvg
Hidden Activation function	relu	relu	relu	relu

Table 4: Convolutional Neural Networks parameters

order to accomplish this, we had to process our data according to BERT guidelines. We used the transformers library to import the 'bert-base-uncased' model and a BERT tokenizer. Later, the cleaned descriptions were preprocessed in order to satisfy the requirements of the BERT model, which include, for example, the presence of specific symbols at the beginning and end of each sequence. Each sentence was then tokenized with the BERT tokenizer, padded and masked. Lastly, we split the data in three separate sets for training, validation and testing purposes and we set up the hyperparameters of the model.

3.5.3 Feature selection and classification based on entropy

So far we had computed data taking into account features based mainly on the frequency count, later weighted with tf-idf. In this stage we pre-processed the data in a different way by selecting a number of features based on entropy. Entropy is a measure of information related to the probability of a word occurring: the more rare (and thus least expected) a word is, the more informative it is. The entropy of a sentence $H(s)$, given the probability of each word $p(w)$ is:

$$H(s) = -\sum p(w) * log_2 p(w)$$

The first step is to compute the entropy for every word by finding its probability $p(w)$ in every document it appears, multiplied by the $log_2 p(w)$ and summing through all the records in the dataset. Once the entropy is found we selected the 10% of the words with the highest entropy. For each feature we create a vector of 1 and 0's depending if the document contains the word or not. Our new dataset now is the collection of vectors and it's what the classifier will receive. Finally, we trained a gradient boosting classifier. To find the best parameters we run a grid search by iterating the number of estimators equal to 50 and 100; the learning rate with values equal to 0.1 and 0.2; max depth with values 1, and 2; and scoring metric the accuracy.

4 Results

4.1 Keywords by industry

4.1.1 Word frequency and ngrams

The analysis of the most frequent words and ngrams returned some interesting insights on the different industries. Despite the relative short length of each description, we were able to identify several occurrencies of typical words as well as bigrams and trigrams. Even recurrent trigrams were significant enough that we were able to list the 20 most frequent per category. In Figure 1, the top 20 most common word in Security and Law are depicted. The top 5 are *law, firm, client, legal, service*: these are surely significant for the business sector they belong to.

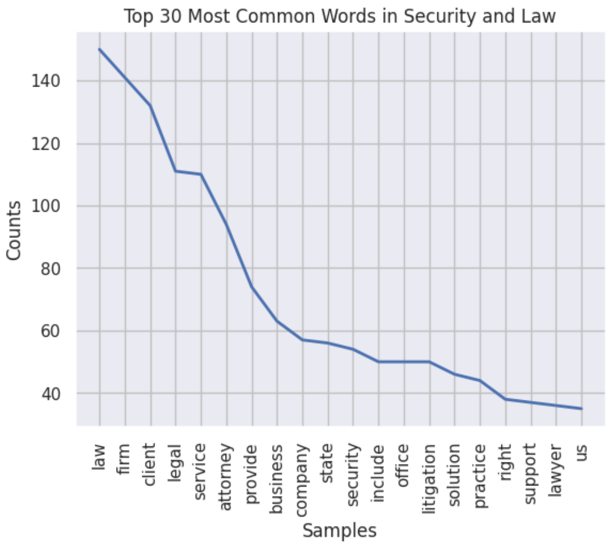
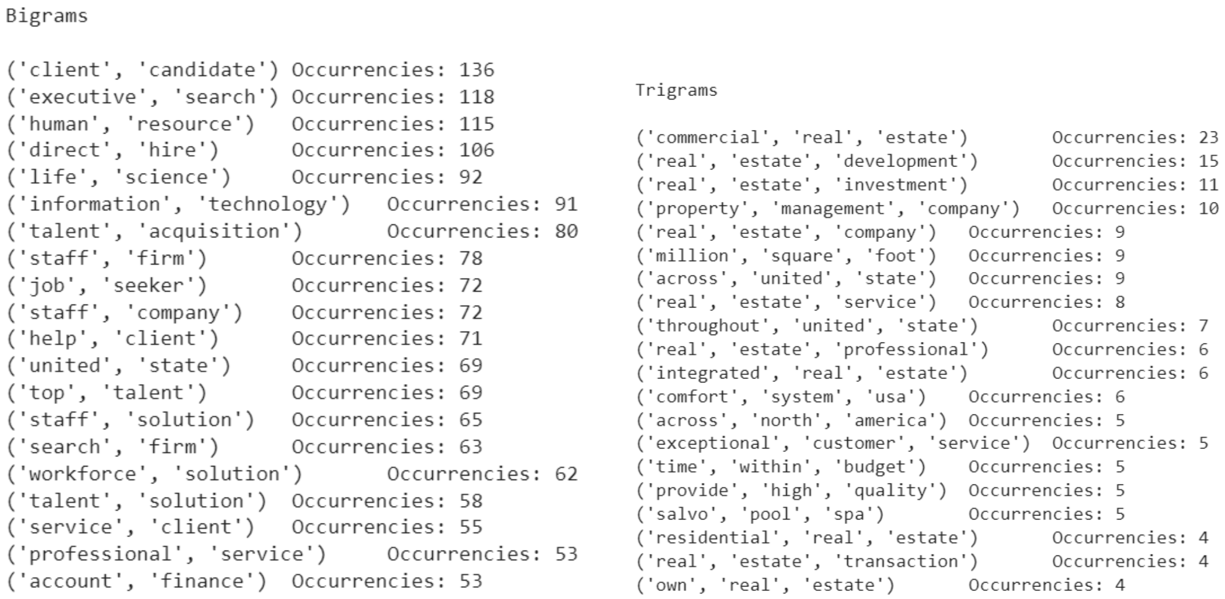


Figure 1: Top 20 most common word in Security and Law

In Figure 2, the most common bigrams in Business and Consulting and trigrams in Real Estate and Construction are depicted. Some notable examples from the former are *talent - acquisition, top - talent, workforce - solution*; some from the latter are *real - estate - development, exceptional - customer - service, provide - high - quality.*



```
Bigrams

('client', 'candidate') Occurrencies: 136
('executive', 'search') Occurrencies: 118
('human', 'resource')   Occurrencies: 115
('direct', 'hire')      Occurrencies: 106
('life', 'science')     Occurrencies: 92
('information', 'technology')   Occurrencies: 91
('talent', 'acquisition')       Occurrencies: 80
('staff', 'firm')       Occurrencies: 78
('job', 'seeker')       Occurrencies: 72
('staff', 'company')    Occurrencies: 72
('help', 'client')      Occurrencies: 71
('united', 'state')     Occurrencies: 69
('top', 'talent')       Occurrencies: 69
('staff', 'solution')   Occurrencies: 65
('search', 'firm')      Occurrencies: 63
('workforce', 'solution')       Occurrencies: 62
('talent', 'solution')  Occurrencies: 58
('service', 'client')   Occurrencies: 55
('professional', 'service')     Occurrencies: 53
('account', 'finance')  Occurrencies: 53
```

```
Trigrams

('commercial', 'real', 'estate')        Occurrencies: 23
('real', 'estate', 'development')       Occurrencies: 15
('real', 'estate', 'investment')        Occurrencies: 11
('property', 'management', 'company')   Occurrencies: 10
('real', 'estate', 'company')   Occurrencies: 9
('million', 'square', 'foot')   Occurrencies: 9
('across', 'united', 'state')   Occurrencies: 9
('real', 'estate', 'service')   Occurrencies: 8
('throughout', 'united', 'state')       Occurrencies: 7
('real', 'estate', 'professional')      Occurrencies: 6
('integrated', 'real', 'estate')        Occurrencies: 6
('comfort', 'system', 'usa')    Occurrencies: 6
('across', 'north', 'america')  Occurrencies: 5
('exceptional', 'customer', 'service')  Occurrencies: 5
('time', 'within', 'budget')    Occurrencies: 5
('provide', 'high', 'quality')  Occurrencies: 5
('salvo', 'pool', 'spa')        Occurrencies: 5
('residential', 'real', 'estate')       Occurrencies: 4
('real', 'estate', 'transaction')       Occurrencies: 4
('own', 'real', 'estate')       Occurrencies: 4
```

Figure 2: Most occurring bigrams in Business and Consulting and trigrams in Real Estate and Construction

4.1.2 Word Clouds

The output of word clouds is visual and quite immediate to interpret. In Figure 3 we report two example of the word clouds respectively relative to Healthcare and Medical Services and Government and Non-profit. Especially after running the code with cleaned data, it is possible to observe that the biggest words and characteristic of the business sector the companies' belong to. For example, the 3 biggest words for Healthcare and Medical services are *care, service, patient* while for Government and Non-profit, word like *community, provide, mission* appear.

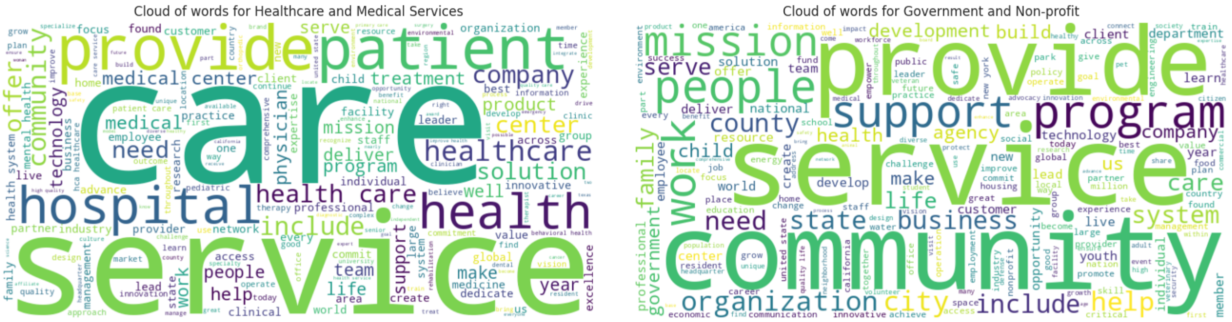


Figure 3: Word clouds for Healthcare and Medical Services and Government and Non-profit

4.1.3 Topic Modelling

As mentioned before, the last step to find the topics of each category was to find the optimal number of topics to extract. This was done for every industry, and due to the long amount of time needed to find the optimal value for all categories, we decided only to iterate between 1 and 10. Figure 4 shows how the coherence value changes when the number of topics increase.

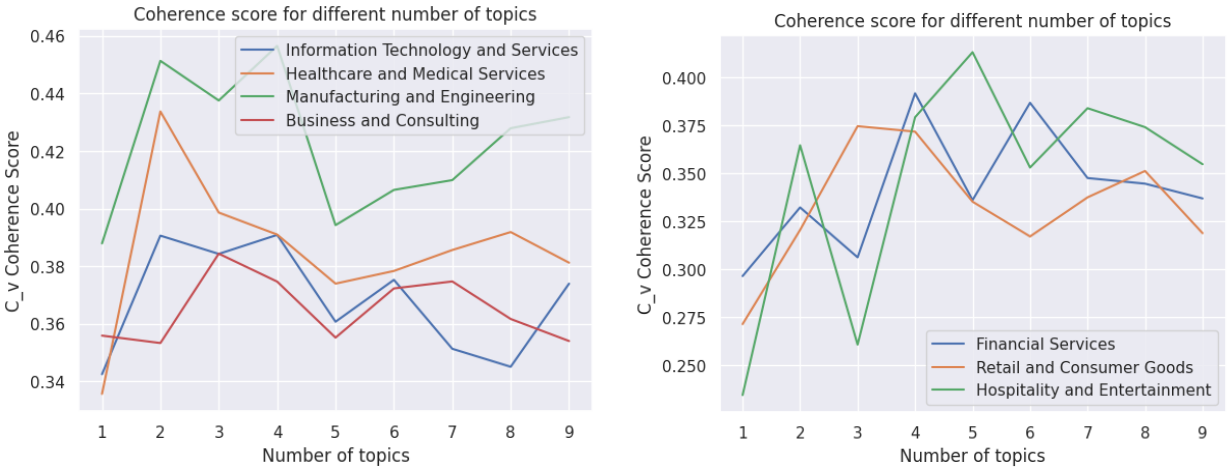


Figure 4: c_v coherence value for different number of topics

After choosing the optimal number of topics we decided to print only the top 10 of words extracted from the topics found of each industry.

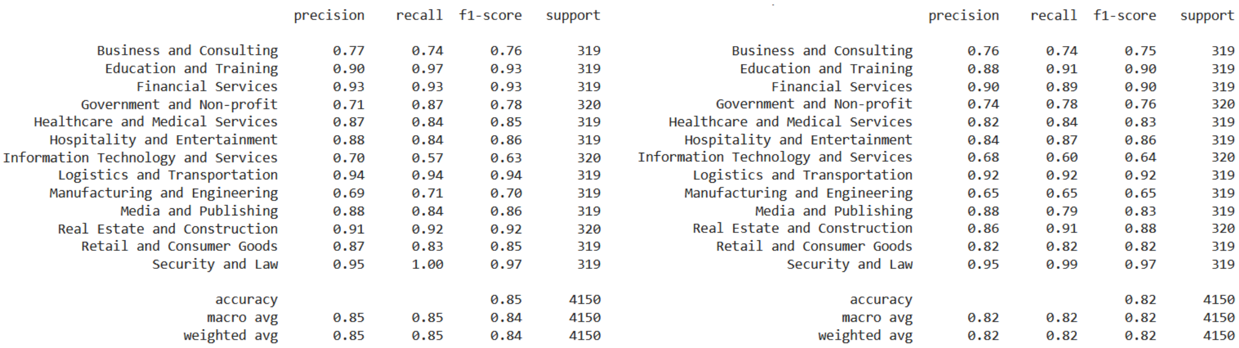
- Business and Consulting: staff, business, firm, solution, provide, professional, industry, best, technology, work.
- Education and Training: university, research, student, program, education, college, science, world, provide, school.
- Financial Services: mortgage, loan, home, customer, housing, opportunity, http, nmls, help, provide.
- Government and Non-profit: community, service, people, care, provide, program, health, organization, family, support.
- Healthcare and Medical Services: hospital, center, medical, service, healthcare, patient, community, system, include, medical_center.
- Hospitality and Entertainment: team, food, people, world, us, create, service, experience, company, work.
- Information Technology and Services: client, customer, help, digital, provide, industry, deliver, global, us, team.
- Logistics and Transportation: logistics, company, solution, supply, chain, supply_chain, transportation, freight, provide, business.
- Manufacturing and Engineering: company, energy, service, product, include, system, power, customer, technology, solution.
- Media and Publishing: brand, marketing, medium, client, business, company, agency, digital, service, help.
- Real Estate and Construction: home, real, estate, real_estate, property, company, community, management, provide, development.
- Retail and Consumer Goods: make, every, people, product, us, world, brand, customer, work, community.
- Security and Law: firm, litigation, law_firm, company, office, legal, practice, service, counsel, provide.

We can notice that several words are repeated between topics. For instance, the words business, people, service, company appear in more than three industry categories. As well we can notice that some of the words present in the initials word clouds also appear here, even after doing the data cleaning. So, we can interpret that even if those words are repeated in several industries, they have some meaning for the type of category.

4.2 Classification

4.2.1 Gradient Boosting and SVC

In Figure 5, we can see the performance of the models. Gradient boosting had a higher accuracy than the linear SVC model with a 85% compared to 82%. With the gradient boosting the industries best predicted were Security and Law, Financial Services and Logistics and Transportation with a precision higher than or equal to 93%. With the linear SVC model these industries continue being the better predicted, but with smallest precision value. The lowest precision for both models was of 69% for Manufacturing and Engineering industry. In general we can say that the performance of both models is good because the precision of the prediction for all the categories is higher than 50%. However, we would like to mention that we were expecting to obtain a better result with the gradient boosting model due to its robustness.



```
                                   precision    recall  f1-score   support

           Business and Consulting      0.77      0.74      0.76       319
            Education and Training      0.90      0.97      0.93       319
                Financial Services      0.93      0.93      0.93       319
        Government and Non-profit       0.71      0.87      0.78       320
   Healthcare and Medical Services      0.87      0.84      0.85       319
     Hospitality and Entertainment      0.88      0.84      0.86       319
Information Technology and Services      0.70      0.57      0.63       320
     Logistics and Transportation       0.94      0.94      0.94       319
    Manufacturing and Engineering       0.69      0.71      0.70       319
              Media and Publishing      0.88      0.84      0.86       319
      Real Estate and Construction      0.91      0.92      0.92       320
       Retail and Consumer Goods        0.87      0.83      0.85       319
                 Security and Law       0.95      1.00      0.97       319

                          accuracy                          0.85      4150
                         macro avg      0.85      0.85      0.84      4150
                      weighted avg      0.85      0.85      0.84      4150
```

```
                                   precision    recall  f1-score   support

           Business and Consulting      0.76      0.74      0.75       319
            Education and Training      0.88      0.91      0.90       319
                Financial Services      0.90      0.89      0.90       319
        Government and Non-profit       0.74      0.78      0.76       320
   Healthcare and Medical Services      0.82      0.84      0.83       319
     Hospitality and Entertainment      0.84      0.87      0.86       319
Information Technology and Services      0.68      0.60      0.64       320
     Logistics and Transportation       0.92      0.92      0.92       319
    Manufacturing and Engineering       0.65      0.65      0.65       319
              Media and Publishing      0.88      0.79      0.83       319
      Real Estate and Construction      0.86      0.91      0.88       320
       Retail and Consumer Goods        0.82      0.82      0.82       319
                 Security and Law       0.95      0.99      0.97       319

                          accuracy                          0.82      4150
                         macro avg      0.82      0.82      0.82      4150
                      weighted avg      0.82      0.82      0.82      4150
```

Figure 5: Performance of the Gradient Boosting and SVC models

4.2.2 Deep Learning Models

Neural Networks

The purpose of training several convolutional neural networks was to find the global classification performance of this approach, as well as the parameters that allow best performance. In order to achieve this goal, we started with a network composed only by one hidden layer. Immediately we noticed that its performance was quite low, as shown in Table 5. Not only the accuracy of this model was of 49%, but it also was not able to predict the categories Security and Law, Media and Publishing, Logistics and Transportation, and Government and Non-Profit.

Later, we decided to train a second model, this time with an additional hidden layer. For this model we noticed a performance improvement: the accuracy increased to 52% and the number of categories that the model was unable to predict decreased to three. Surprisingly, the prediction on the industry Security and Law jumped from 0% to 23%.

Having realized that two hidden layers work better than one, we decided to change the type of pooling in our convolutional network. The first models were trained using **Max Pooling**, so for this model we used **Average Pooling**. The change in the accuracy value it was not significant for this model, but for the number of categories that the model was not able to predict, since it went from 3 to only 1, which was Logistics and Transportation.

Finally, for the last model we decided to experiment with another optimizer, and instead of using **Adam** as for the previous three models, we used the **Stochastic Gradient Descent** (SGD) optimizer. Nevertheless, we obtained a really bad performance of the model. This model was not able to predict any class, and return an accuracy of 18%.

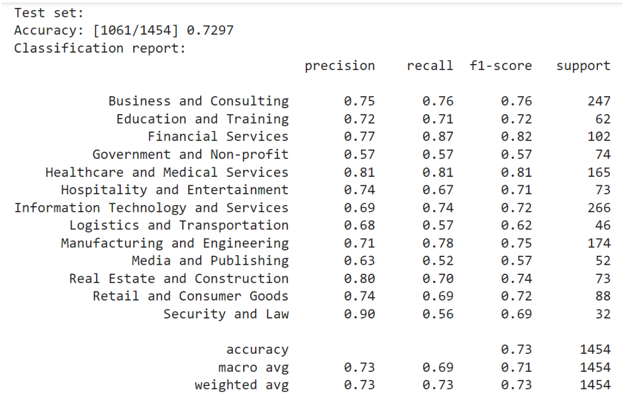
BERT

As for BERT, we used a pre-trained model as not to create the neural structure from

Metric	Model 1	Model 2	Model 3	Model 4
Accuracy	49	52	52	18
Highest Precision	67	81	69	18
Lowest Precision	0	0	14	0

Table 5: Convolutional Neural Network results

scratch. We trained over 8 epochs and set the model hyperparameters in order to have a learning rate of 2^{-5} and a weight decay of 0.01 using the adam optimizer. The results were much more consistent with respect to the CNN models and achieved a better performance by every metric. The improvement is particularly noticeable in the most complicated classes to predict: for example, the lowest precision score in terms of class was achieved by the "Government and Non-Profit" industry with 57%, while the best CNN could achieve 22%.



```
Test set:
Accuracy: [1061/1454] 0.7297
Classification report:
                                     precision    recall  f1-score   support

            Business and Consulting       0.75      0.76      0.76       247
             Education and Training       0.72      0.71      0.72        62
                 Financial Services       0.77      0.87      0.82       102
           Government and Non-profit       0.57      0.57      0.57        74
    Healthcare and Medical Services       0.81      0.81      0.81       165
      Hospitality and Entertainment       0.74      0.67      0.71        73
Information Technology and Services       0.69      0.74      0.72       266
       Logistics and Transportation       0.68      0.57      0.62        46
      Manufacturing and Engineering       0.71      0.78      0.75       174
               Media and Publishing       0.63      0.52      0.57        52
       Real Estate and Construction       0.80      0.70      0.74        73
          Retail and Consumer Goods       0.74      0.69      0.72        88
                   Security and Law       0.90      0.56      0.69        32

                           accuracy                           0.73      1454
                          macro avg       0.73      0.69      0.71      1454
                       weighted avg       0.73      0.73      0.73      1454
```

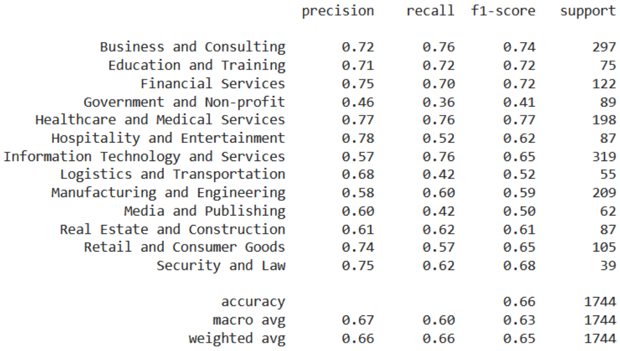
Figure 6: Performance of the BERT model

4.2.3 Feature selection and classification based on entropy

From the entropy based selection we obtained 1805 features which became the columns of our new dataframe. For instance, the first ten words of the list of features are: *service, company, provide, solution, business, client, technology, help, customer, and industry.* The last 10 words are: *homeowner, traffic, distinctive, physicians, self, loyalty, replacement, someone, migration, and funding.*

The new dataframe was used to train a gradient boosting model. The best parameters obtained with the grid search were: number estimators= 100, learning rate= 0.1, and max depth= 2. In Figure 7, we can appreciate that our model had an accuracy value of 65%. This is a really good performance compared with the neural network models for two reasons, one, the accuracy value is higher than the accuracy of all the CNN models; two, the model was able to classify all the categories, being 46% the lowest precision value corresponding to the Government and Non-profit industry.

Finally, it is important to remark that more experiments with the parameters of this models are difficult to run because its running time takes more than two hours in a basic computer.



```
                                    precision    recall  f1-score   support

            Business and Consulting      0.72      0.76      0.74       297
             Education and Training      0.71      0.72      0.72        75
                 Financial Services      0.75      0.70      0.72       122
           Government and Non-profit      0.46      0.36      0.41        89
     Healthcare and Medical Services      0.77      0.76      0.77       198
      Hospitality and Entertainment      0.78      0.52      0.62        87
Information Technology and Services      0.57      0.76      0.65       319
       Logistics and Transportation      0.68      0.42      0.52        55
      Manufacturing and Engineering      0.58      0.60      0.59       209
               Media and Publishing      0.60      0.42      0.50        62
       Real Estate and Construction      0.61      0.62      0.61        87
          Retail and Consumer Goods      0.74      0.57      0.65       105
                  Security and Law      0.75      0.62      0.68        39

                           accuracy                          0.66      1744
                          macro avg      0.67      0.60      0.63      1744
                       weighted avg      0.66      0.66      0.65      1744
```

Figure 7: Performance of the gradient boosting model with the features selected based on entropy

5 Discussion and conclusion

Many were the trials in this project and in the end we achieved a variety of results.

A first jump in quality in our classification task was achieved through a manual, attentive cleaning of the data. On a first try, we computed word clouds taking in input the raw data, directly extracted from the dataset which reported companies' descriptions picked from LinkedIn. At first glance, the results were quite impressive from the word cloud representation, but the classification model accuracy was low. Upon closer inspection, we noticed that formally no cleaning of the data had been done and, for example, among the top 20 most frequent words only 2 were not stopwords. Employing the pipeline to clean the data with NLTK, we gained 4% in accuracy which is quite an impressive difference with data cleaning alone.

On the other hand, our main objective was to train a classification model that predicts the type of industry to which a company belongs, based on the description provided by the company. After training several models of various types, such as gradient boosting, linear support vector model, and the convolutional neural networks we can conclude that the best performance was achieved by gradient boosting. And the worst performance was with CNN models. On the one hand we expected a good result from the gradient boosting model, i.e, we were expecting to achieve at least an 90% of accuracy with the balanced data. Also for neural networks we were expecting a better performance. Our conclusion is that in order to take advantage of the benefits of neural networks it is necessary to have more information, and that the documents with which the neurons were trained should be larger, i.e. have longer descriptions. This is confirmed by the fact that even implementing BERT, which is renowned as a State-of-the-art model in text classification tasks, we have just obtained a slightly better performance. It is important to notice that in the case of the deep learning models (CNNs and BERT) we weren't able to generate the oversampled data as we did for the other models because of the difference in type of input that those models require versus the traditional machine learning ones. A possible improvement to this work could be the generation of synthetic data also for deep learning models, for example by leveraging Generative Adversarial Networks (GAN). At the same time, we are surprised for the good results obtained with the classification using feature selection based on entropy. We believe that, like with neural networks, longer descriptions and therefore more features would produce better results in the classification of certain categories.

As part of our project, we have successfully applied a balancing algorithm. This step allowed us to deal with the class imbalance problem, which led to an improvement in the model's ability to classify correctly. In the process of our research, we conducted a comparative analysis of three approaches to class balancing: oversampling, undersampling and their combination. Of all the methods considered, oversampling demonstrated the best results. Thus, we boosted the overall accuracy of the model and improved the model's ability to correctly recognise and classify the minor class. The initial accuracy was 62% , but after implementing the oversampling technique, we achieved a significant increase in accuracy to 85%.

References

[1] Hannah Béchara, Ran Zhang, Shuzhou Yuan, and Slava Jankin. Applying nlp techniques to classify businesses by their international standard industrial classification (isic) code. In *2022 IEEE International Conference on Big Data (Big Data)*, pages 3472–3477, 2022.

[2] Marco Felgueiras, Fernando Batista, and Joao Paulo Carvalho. Creating classification models from textual descriptions of companies using crunchbase. In *Information Processing and Management of Uncertainty in Knowledge-Based Systems: 18th International Conference, IPMU 2020, Lisbon, Portugal, June 15–19, 2020, Proceedings, Part I 18*, pages 695–707. Springer, 2020.

[3] Martin Lamby and Daniel Isemann. Classifying companies by industry using word embeddings. In *Natural Language Processing and Information Systems: 23rd International Conference on Applications of Natural Language to Information Systems, NLDB 2018, Paris, France, June 13-15, 2018, Proceedings 23*, pages 377–388. Springer, 2018.

[4] Stanislav Slavov, Andrey Tagarev, Nikola Tulechki, and Svetla Boytcheva. Company industry classification with neural and attention-based learning models. In *2019 Big Data, Knowledge and Control Systems Engineering (BdKCSE)*, pages 1–7, 2019.