John Snow LABS

SPARK NLP

Text Classification into a Hierarchical Market Taxonomy using Spark NLP

by Dr. Alina Petukhova

Table of Contents

Abstract

In recent years traditional factors of financial markets, for instance growth vs. value, market capitalization, credit rating, stock price volatility, have become less predictive, requiring investors to explore new data sources such as news, images, social networks content etc. Most of this content is unstructured and need to be first converted into structured data to be used for analysis. As an alternative of creating these systems themselves, financial firms are turning towards companies that specialize in this field. Bitvore platform is set up to handle the task of systematical scraping the data from the different sources. After, this data getting to link to the key factors and topics with semantic intelligence, that can be delivered for use in financial trading algorithms.

The goal of the current project is to build an instrument, which classifies the companies into different market segments in which they operate. The classification is based on the “Thomson Reuters Business Classification” taxonomy. Currently the Bitvore assigns market via a rule-based system, which is working with text matcher of company name mentioned in the text. To create a dictionary for the text matcher initial classification should be done by human annotators, that can process only a limited number of key companies. This approach has limitations and not able to assign new markets to the existing companies or classify correctly new companies for the market without involving labelers, but in real life each day a thousand new companies appear, which makes the overall task time consuming and needed to be automated.

Using artificial intelligence methods and machine learning algorithms, we predicted markets labels from textual data by applying text mining techniques to news stories. In this paper, we describe a Natural Language Processing (NLP) approach using Spark NLP and semantic techniques to assist the domain experts in classifying the documents with different market labels. Our approach combines hierarchical multilabel document classification approach and outlier detection algorithm. Final solution was integrated and deployed as a Web service to the Bitvore system.

Introduction

In today's financial marketplace, a well-maintained portfolio is vital to any investor's success. Overall, a well-diversified portfolio is an investor's best bet for the consistent long-term growth of their investments. In decision-making analysis, market structure has an important role through its impact on the decision-making environment. The investors use market classification standards to make portfolio diversification and overall asset allocation decisions.

That's why many investors are maintaining a mix of markets in their portfolios, by highlighting acquisition targets and opportunities for financial restructuring. Additionally, corporates performing competitive analysis of their peers in the marketplace. In order to employ this type of strategy, they should be able to classify data by sectors and industries.

Markets can be classified on different bases of which most common bases are: area, time, transactions, regulation, and volume of business, nature of goods, and nature of competition, demand and supply conditions. Companies can have multiple businesses in different markets to leverage their global network and to scale opportunities varying across markets.

There are 3 main classification schemas for the market classification task: The Global Industry Classification Standard (GICS), the Industrial Classification Benchmark (ICB), and the Thomson Reuters Business Classification (TRBC). These classification schemas are designed for creating benchmarks and to provide an acceptable and meaningful method for standardizing industry definitions so that comparison and analysis can be conducted between companies, industries, and sectors.

Problem Description

The goal of Market classification project workstream is to perform unstructured text analysis (company filings, news, and corporate actions) and to develop predictive and statistical models for market classification.

In our project will be using adapted TRBC classification, which consists of three levels of hierarchical structure. Each company is allocated a Market, which falls under Industry, which is then the part of an overall Sector. Taxonomy consists of 10 Sectors, 28 Industries and 61 Markets. In our models we used “Market” as the lowest level of classification. After the classification of the text, the company name will be extracted applying Named Entity Recognition (NER) algorithm and associated with the predicted market.

The main stages of the project workstream were:

1. Identify text type and semantic criteria for market level of classification to standardize the labeling process. Flesh out the taxonomy.
2. Create a negative data set to include texts not related to the any of markets (e.g. financial reports, general news, etc.).
3. Provide identification of keywords and phrases that can be used to increase Bitvore’s content searching for capturing of data into the system for each market.
4. Use unsupervised technics to analyze and label existing in the Bitvore system data to be able to extend the manually labeled dataset.
5. Create and apply an outlier detection algorithm to improve dataset quality.
6. Train multilabel hierarchical document classification model for identifying and tagging title and content body (unstructured text) with market from the taxonomy.
7. Provide precision and recall metrics for each market and iteratively optimize the model to achieve optimal results.
8. Detect false negative predictions and extract them for manual processing.

Due to the big data scope we are going to use Spark NLP framework to train the models.

During the project, we defined target Markets taxonomy to train AI models for nine different sectors and 43 distinct markets. Below is a sample of several of the fully qualified tag labels using sector, industry, and market (refer to table 1).

Sector	Industry	Market
Energy	Energy	Energy
		Oil & Gas
		Renewable Energy
Basic Materials	Chemicals	Chemicals
	Metals & Mining	Coal
Financials	Financial Services	Banking Services
		Investment Banking & Investment Services
		Investment Holding Companies
	Real Estate	Real Estate Operations
		Residential & Commercial REITs
Healthcare	Healthcare / Pharma	Biotechnology & Medical Research
		Healthcare Providers & Services
		Pharmaceuticals
Technology	Electronics, Computers & Technology	Communications & Networking
		Electronic Equipment & Parts
		Software & IT Services
Utilities	Utilities	Electric Utilities & IPPs
		Water & Related Utilities

Table 1. Market taxonomy (subset)

Challenges

Lack of reliable training data: During the project kickoff we received an unlabeled dataset containing > 1000000 entities. Labeling data for the Market classification task is a complicated and time-consuming manual process, which has a lot of bias. In general, tagging texts is very subjective to a person's perspective. In the Market taxonomy some tags have very close meaning and can be mislabeled not only by a model, but by a human annotator as well. For example, two markets "Financials.Financial Services.Investment Holding Companies" and "Financials.Financial Services.Investment Banking & Investment Services", usually are equally justified as per the text body. Because of that, it's important to have detailed instructions for the content labeling team to make sure different annotators share the same ideas on content during the labeling process. Otherwise, model metrics are going to be low, and the mismatch of tagged texts will be high because of the specific interpretation. The main goal during the labeling process is to exclude bias towards a tester for the better model performance.

Solution

To be able to train models we created labeling guidance for annotators team and partially labeled dataset.

Errors during the batch labeling: Due to the limited resources and time frames for the project, it was challenging to review precisely every text in the data set during the labeling. Because of that labeling has been done based on the keywords in the text entity. This approach causing the errors in the dataset and may produce not reliable results.

Solution

To solve this issue was implemented outliers detection algorithm, which allows us to filter the most distant texts, based on the center of the cluster they belong to, which represent average semantic for each market.

Low-Quality Data: Another challenge for this project is to deal with Markets with a lesser number of tagged entities and entities with a corrupted body. If models will be trained on this data it will not perform with expected accuracy.

Solution	The data preprocessing consisted of removing short texts and texts with a corrupted body. Additionally, we analyzed wrong predictions of the model, to make sure, what annotator labeled initial text correctly. To make sure we are not overfitting the model random state of train/test/validation split was changed on every data regeneration.

Need for large training data: As this taxonomy require a big amount of training data (61 categories). The minimum requirement is 100 texts per category.

Solution	To partially automate this process, we added a semi-supervised algorithm, which allows us to automatically label texts within cluster borders for each market.

Dynamic market taxonomy: Each day many new companies as well as new markets are coming into existence, this may create false negative labels for some of the entities.

Solution	Added after processing to the models and extract entities, which are not relevant to the any of existing markets. That examples should be processed manually and taxonomy for the model should be extended if required.

Multiple activities for a record: Another challenge was the fact that most of the texts have to do with multiple activities, for our model, multiples tags are possible, but adding multiple tags to the texts did not make bulk tagging possible, as the tagging time for multiple tagging would be long, and thus resulting in a lower number of tagged texts.

Missing content for an activity: For some activities, it's hard to find good content specific to the activity. Example: "Holding Companies – NEC", they only labeled, based on the inclusion of the name "Holding" with the company name. No other specific information related to this market is mentioned. This problem may decrease classification accuracy for some markets.

Data Set Overview

Having enough texts per tag is critical to accurately train a model. To create a labeled data set we used news texts from open sources (magazines, newspapers, blogs) for the last 5 years. Also, we added negative data set containing texts not relating to the Market, that companies operate on (financial reports, general news etc.).

Data Exploration

Figure 1 shows an analysis of the tagged data set statistics – number of samples per Market and the average number of words per text's body.

One of our main concerns when developing a classification model is whether the different classes are balanced. This means that the dataset contains an approximately equal portion of each class. From the Fig. 1a we can see, that Markets in the Industry "Financial Services" are imbalanced in the data set. Additionally, we can see the two least represented Markets:

- "Consumer Non-Cyclicals.Education.Education" and
- "Consumer Non-Cyclicals.Biotechnology.Biotechnology"

had less, then 600 entities in the data set, which may affect model results. However, from Fig. 1b we see that in terms of number of words for each Market are much more homogeneous (except for Negative label). The overall average is 308 ± 172 words.

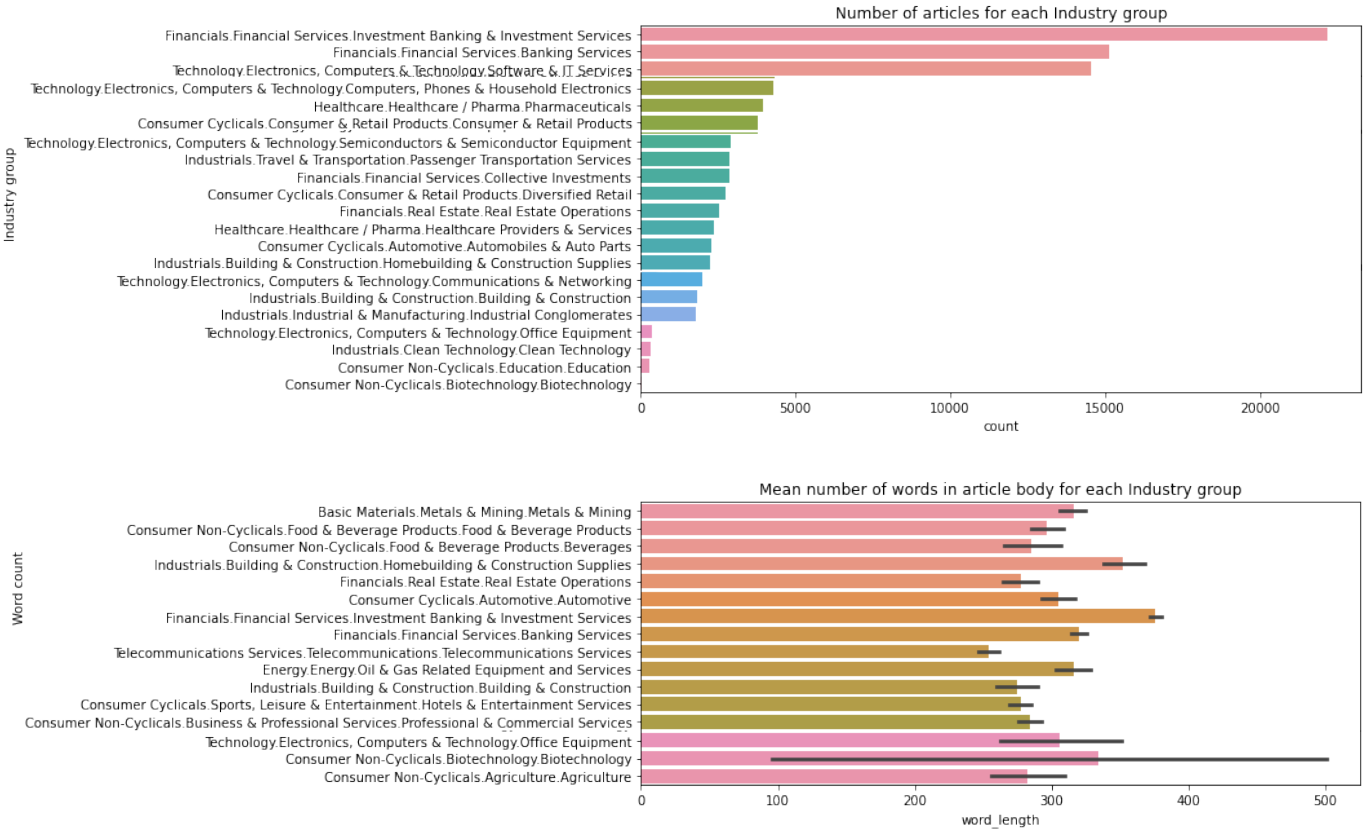

Fig 1: Statistical analysis of dataset: (a) Number of texts for each Market (subset) (b) Mean number of words for each Market (subset)

Data Preprocessing

Data preprocessing included the following steps:

- Processing samples with body size greater than 600 words – this left us with the total number of samples 211 813 and 61 market labels + 1 Negative label.
- Removal of stop words as well as punctuation and HTML tags
- Lemmatization of each word.

Stop words: commonly used words such as “and” or “the” won’t have any predicting power since we can see it in all the texts. There is a common practice to remove them to reduce noise in the data set. We used a list of English stop words from Spark-NLP library and extended the list by adding days of the week, months and numbers.

Lemmatization: words were reduced to standardized lemmas, which takes into consideration the morphological analysis of the word.

To determine relative prominence for most prominent terms for each Market and to form a stop words list we built Word clouds charts. A Word cloud is a collection, or cluster, of words depicted in different sizes. The idea behind this approach – each word placed in the chart based on frequency in the given text. The bigger and bolder the word appears, the more often it’s mentioned within a given text and the more important it is. It provides us with some interesting observations regarding the distribution of Market keywords in the labeled collections.

Fig 2: Word Cloud: (a) Before preprocessing (b) After preprocessing

Data Science

Outlier detection algorithm

To find outliers in the data set initially we have to create clusters for texts, which belongs to one category. We represent documents as vectors of features, and compare them by measuring the distance between these features. To extract features, we tested a few approaches:

- **word-level similarity,** based on the token extraction algorithms, like CountVectorizer and TFIDF. It does not take into account the actual meaning behind words or the entire phrase in context.
- **context similarity,** based on the contextual embeddings in order to capture more of the semantics. To consider semantic similarity we focused on the sentence level embeddings (Bert and Elmo).

After compiling embeddings, we calculated mean and median for each cluster and for each text calculate distance from center. During the project we tested different coefficients for distance measurement:

- **Euclidean Distance –** ordinary distance between two points, that measures the length of a segment connecting the two points.
- **Cosine Similarity –** measures the similarity between two vectors of an inner product space. It is measured by the cosine of the angle between two vectors and determines whether two vectors are pointing in roughly the same direction (independence of document length).
- **Jaccard Coefficient –** measures similarity as the intersection divided by the union of the objects.
- **Pearson Correlation Coefficient –** measures the degree of a linear relationship between two profiles.

Experiments confirmed, what for commonly used word vectors, cosine similarity is equivalent to the Pearson correlation coefficient. In the given dataset the best results we obtained by using Cosine similarity combined with Elmo embeddings.

After receiving the distance from the center of cluster for each market we had to choose the approach to filter texts not belonging to the clusters. We analyzed the distribution of the distances in the dataset and, as it's shown on the Fig. 2, distances from the center of cluster are normally distributed. For the normally distributed values we can measure standard deviation and filter the entities by applying 68–95–99.7 rule. We choose to use 95% as a threshold to remove outliers on the left side of distribution.

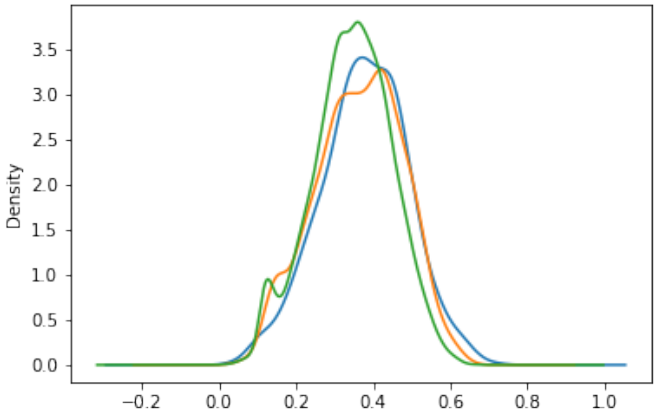

Fig 3: Cosine distance from center of cluster for 3 labels

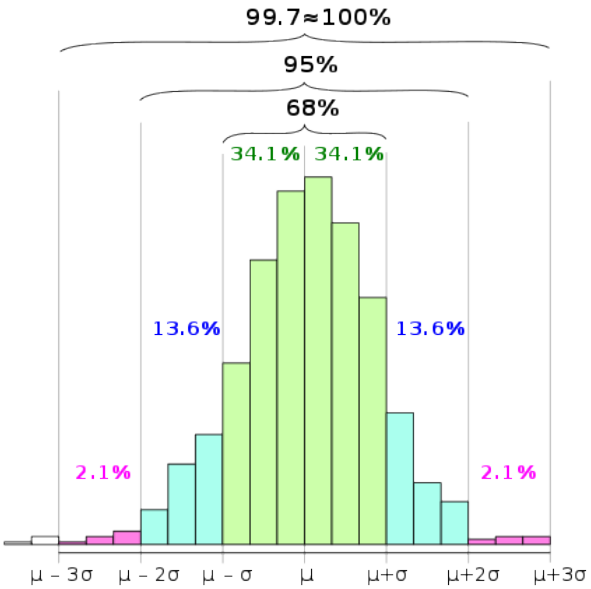

Fig 4: 68–95–99.7 rule visualization for the normal distribution

Experimentation

The end goal of these models is to make a prediction for multilabel classification problems and predict properties of a data-point that are not mutually exclusive, such as Markets that are relevant for a text.

First, using the preprocessed titles and bodies we created the dictionary. The total number of unique words is around 588 981. Then, we extracted the following word features for the classification task:

- **Word count features:** For count features, we used first 3000 most common words for body field and 1500 for title field to define the dictionary and then, encoded the titles and bodies as vectors - with a length of the entire vocabulary and an integer count for the number of times each word appeared in the document.
- **Word TF-IDF scores:** TF-IDF Vectorizer transform text into the feature's vectors, learned from the vocabulary, and document frequency of each word in the training data. For TF-IDF method we used dictionary size 5000 for body and 7000 for title feature.
- **Doc2Vec embeddings:** Doc2Vec is a Model that represents each text as a paragraph vectors learned from the training data. It's based on Word embeddings, which is a family of NLP techniques aiming at mapping the semantic meaning into a lower-dimensional vector space using a shallow neural network [1]. Word embeddings producing a set of word-vectors where similar meanings vectors located close together and word-vectors having a have differing meanings is distant to each other. To create embedding for all documents we used Le and Mikolov in the 2014 algorithm, which usually outperforms such simple averaging of Word2Vec vectors. Doc2Vec has two main implementations:

    1. Paragraph Vector – Distributed Memory (PV-DM)
    2. Paragraph Vector – Distributed Bag of Words (PV-DBOW)

In our pipeline we combined PV-DM and PV-DBOW implementations as input features for the models. This approach is suggested by authors [1] and during the evaluation it helped us to improve final metrics. We trained Doc2Vec embeddings on the vocabulary from the collected dataset. Also, we considered only words with a minimum count of 5 for body and 2 for title fields and used dimensionality of the feature vectors 50 and 100 respectively.

In addition, we tried applying pretrained GloVe [2] embeddings (with frozen Embedding layer) but the accuracy in this case was lower than when learning embeddings from the data.

In the first part of our work we experimented with traditional machine learning techniques: multinomial logistic regression, Naive Bayes, kernel SVM, and Random Forest. In the pipeline we have used public pre-trained models offered by Spark NLP.

For our implementation, we trained pipelines with several architectures (model types, amount of iterations, penalties, normalization) as well as with different parameters such as an embedding dimension, maximum sequence length, and the maximum number of words (for words tokenization).

General process:

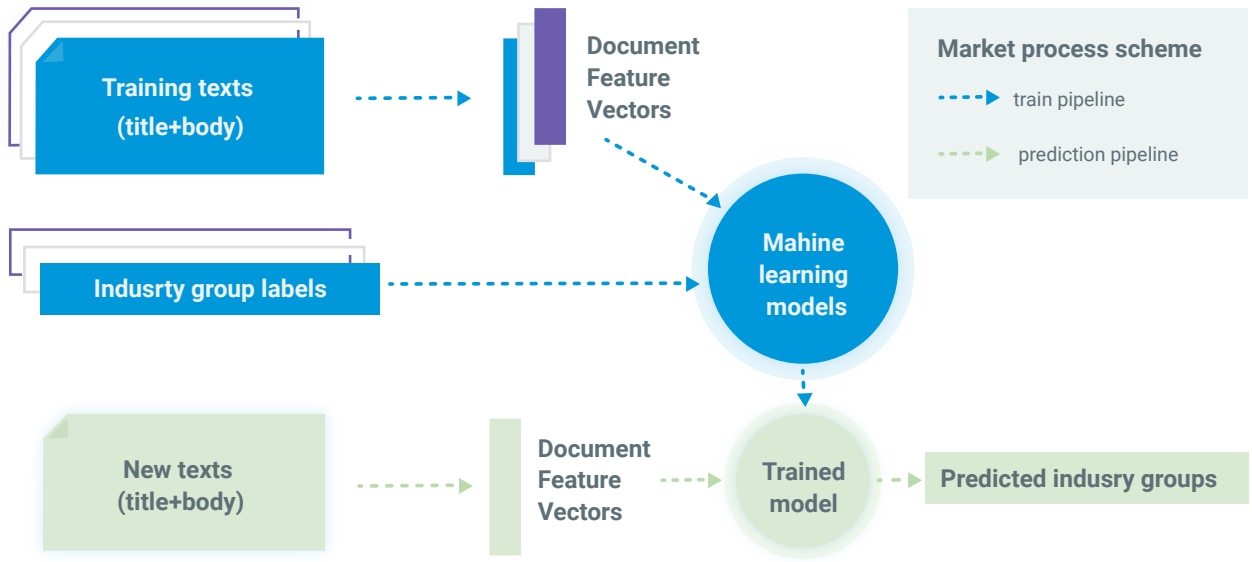

Fig 5: Typical pipeline architecture

Taxt based model:

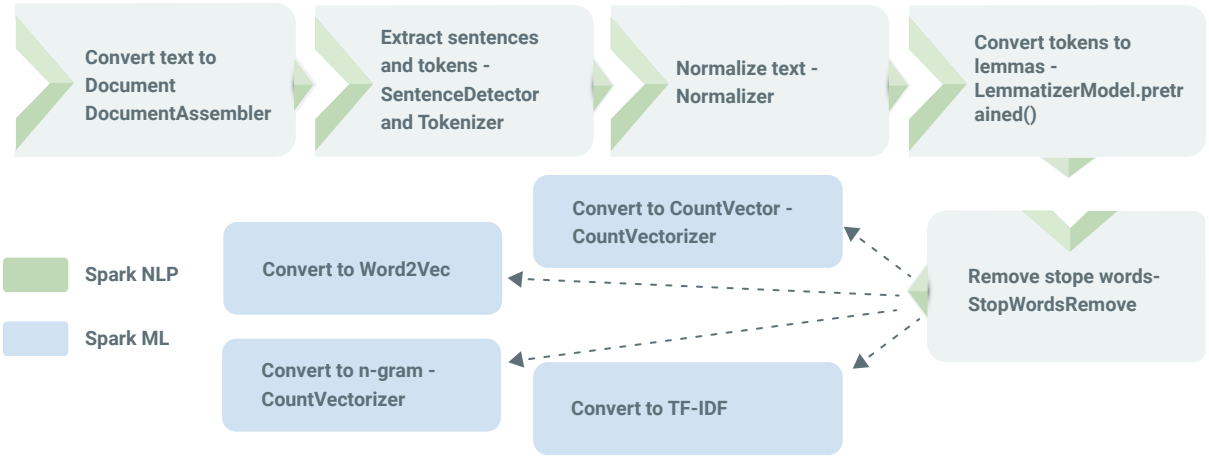

Fig 6: Spark NLP and Spark ML stages for text classification

Classification models:

Multinomial Logistic Regression was used with cross-entropy loss and L2 regularization, penalizing model to minimize the cost function [3]:

$$\min_{w,c} \frac{1}{2} w^T w + C \sum_{i=1}^{n} \log(\exp(-y_i(X_i^T w + c)) + 1).$$

Multinomial Naive Bayes used in the training, implements the naive Bayes algorithm for multinomially distributed data with additive smoothing parameter 0.8.

Random Forest used in the training, fits several classifying decision trees on various sub-samples of the dataset and uses averaging to improve the predictive accuracy and control over-fitting. We used the Gini criterion as a function to measure the quality of a split and 10 estimators and regularized each tree in terms of maximum depth.

Kernel SVM used in the training, implements the “one-against-rest” approach with multi-class SVM [4] and Linear kernel. Additionally, we tested the “one-against-rest” decision function with RBF kernel, but this approach didn’t show good results.

Hierarchical models [5] used to improve model accuracy, since it's naturally cast as hierarchical classification problems, where the classes to be predicted are organized into a tree class hierarchy. To implement this approach, we represented every market as a combination of top-level tag - "SECTOR", second-level tag - "INDUSTRY" and third-level tag - "MARKET".

We applied the Local Classifier Per Parent Node Approach, which trains a multilabel classifier for each node of the class hierarchy (leaf nodes).

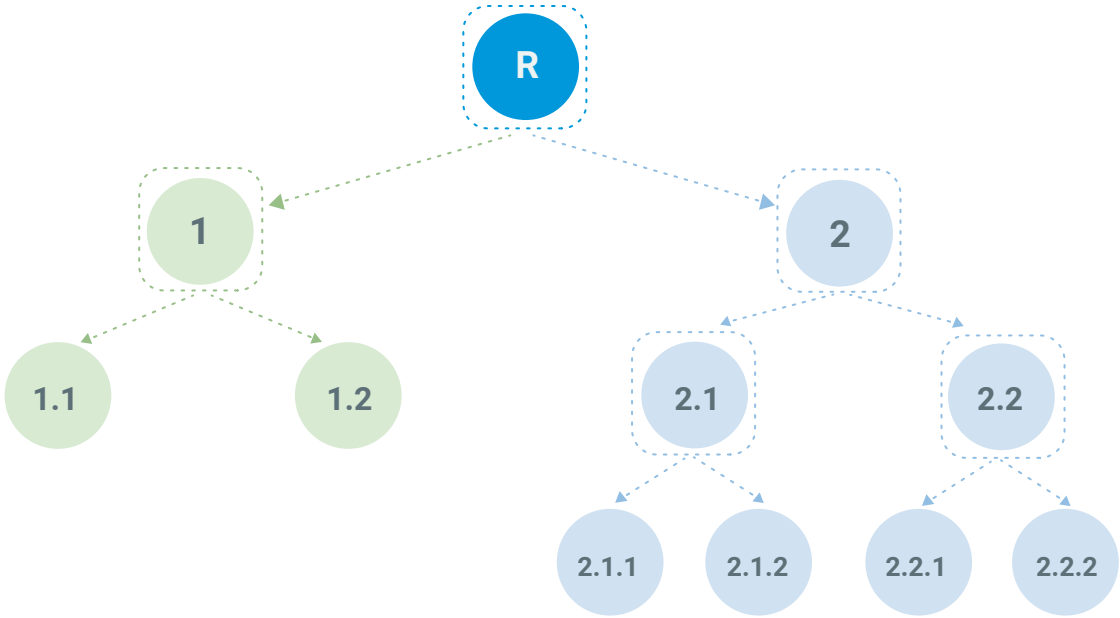

Fig 7: Local Classifier Per Parent Node Approach (circles represent classes and dashed squares with rounded corners represent multi-label classifiers)

To calculate the final prediction of the pipeline we combined each level prediction with equal weight and choose for the final prediction market with maximum f1 score overall markets.

Productization

After analyzing the metrics for different models, we observed that the metrics difference is not significant. Thus, it is not useful to freeze the best model for our task. So, to get the best results, we need to pick the best performing model for each level of hierarchy automatically on every retraining. This solution will allow us to automate the whole process from the regeneration of new data set to prediction with the best performing combination. In our final implementation best model and corresponding embedding is selected automatically during the training, based on f1 score metric. Also, to be able to analyze results for every target label (sector, industry, market), each model includes the capability of providing precision, recall, and f1 score for each level.

We have built an assembly of models to predict the market for texts using the title and body. We used methods both from traditional ML and deep learning. All models were saved as pickle files in Cloud Storage Service, giving us the ability to load models later and to split train and prediction process if needed. To be able to run implemented pipelines in the client's infrastructure we converted Python code, used for testing, to the production-ready architecture. For this implementation we choose Flask [6] REST API wrapped in the Docker [7] container.

Flask is a light framework for Python that offers a powerful way of annotating Python function with REST endpoint.

REST API allows us to send new data to the models and receive a prediction as a response. It will allow ML model to be accessible by the 3rd party business applications.

Predictions are made by passing a POST JSON request with title and body to the created Flask web server on port 5000 by default, which is mapped as an external port. API receives this request and make a prediction, based on the already loaded latest models. It returns the prediction in JSON format.

Market Model input

{

"title": " Blt Enterprises Leases 89000+SQFT Industrial and Office Facility in Oxnard Ca to Global Automotive Company",

"body": "— BLT Enterprises , a multi - faceted commercial real estate development and investment company, has leased an 89,811 square - foot industrial and office facility to Volkswagen Group of America, Inc. at 3301 Sturgis Road in Oxnard, California. \n \nThe property is adjacent to the company's existing Test Center California, which BLT developed as a build - to - suit for Volkswagen in 2010, according to BLT Enterprises Founder and President, Bernard Huberman. \n \n"Volkswagen is one of our long - term tenants, with an existing 20 - year lease in place for its state - of - the - art Test Center," explains Huberman. "Our ability to support this expansion speaks to our company's positioning as a life - long landlord. With this new lease, Volkswagen will be able to expand its operations, by moving part of its Vehicle Testing team to the property, while also relocating a variety of other departments to the facility." \n \nBLT Enterprises, along with Volkswagen Group of America, will complete a series of improvements at the property to completely renovate the existing building. \n"By strategically improving the facility, we will be able to customize the space to fit Volkswagen's precise needs, and reflect with its unique culture," explains Huberman. \n \nPlanned renovations include interior partitions, new roll - up doors and paint, new amenities such as EV charging stations and carwash systems, as well as important exterior upgrades such as landscaping and parking lot updates, a new roof, new rooftop HVAC units, and removal of the fence between the two properties to create a unified campus feel. \n \n"As long - term owners, we approach each transaction with the goal of being the 'last landlord' our tenants ever need," says Huberman. "By creating environments where tenants can easily grow and expand as needed, we support our tenants in their long - term growth. This new lease is a perfect example of that strategy in action. As Volkswagen looked for expansion options in the area, we were able to identify a property, implement upgrades and accommodate Volkswagen's needs at every turn." \n \nThe property was leased to Volkswagen Group of America, Inc. on a seven - year term. Greg Lubar at Jones Lang LaSalle represented Volkswagen as the lessee. \nThe layout of the rendering pictured is a possible option for the future and is not the final approved build - out."

}

Market Model output

```
{
    "Financials.Real Estate.Real Estate Operations ": 0.913,
    "Financials.Real Estate.Residential & Commercial REITs ": 0.834,
    " Industrials.Building & Construction.Building & Construction": 0.471
}
```

Additionally, we created different endpoints, allowing customer to receive full information about currently uploaded models:

1. **/train** endpoint allows user to retrain models with fine tuning or with previously found best parameters
2. **/metrics** endpoint returns latest models metrics for train, test and validation subset, allowing to pass parameter for bulk validation of external data set
3. **/modelClassMetrics** endpoint returns detailed metrics for each market
4. **/classCount** returns statistic for the dataset, which contains markets and amount of entities

This implementation allows customer's analytic team to control model performance and have an easy access to the model quality metrics.

Another big underrated challenges in machine learning development is the deployment of the trained models in production in a scalable way. We resolved this challenge by using Docker container, which allows us to have a lot of services up, which work in an isolated manner and serve as a data provider to a web application.

Conclusion

Our algorithm with best models' selection and joined hierarchy models achieves on the test set **93.5%** in f1 score metric and **70.4%** on the test set, for predicting the market for given text.

Joined model	Precision	recall	F(1) Score
train	0.890	0.989	0.935
test	0.720	0.753	**0.704**
validation	0.639	0.734	0.659

Table 2. Best models result with hierarchical approach

We examined the prediction errors made by our algorithm to understand the cause of the model's misinterpretation for low performing markets like "Technology.Electronics, Computers & Technology.Office Equipment" and "Utilities.Utilities.Multiline Utilities". We can conclude, what it's the markets, which are having the least of the entities and existing texts have a low quality of content. To improve the results manual review is required. Additionally, to visualize dataset overlapping, we extracted TFIDF of the selected words and applied a dimension reduction method (t-SNE [8]) to visualize the word vectors in 2-D space. Fig. 7 shows the result of this operation. If we review the distribution of "Technology.Technology Equipment.Office Equipment" market we can notice, that it intersects with other clusters - "Technology.Technology Equipment.Communications & Networking" (black cluster in the left), " Technology.Technology Equipment.Electronic Equipment & Parts" (gray cluster in the top right). Due to the low amount of data for this market and not specific texts content, this category is not showing significant quality in classification and models have tendency to mislabel it with other markets.

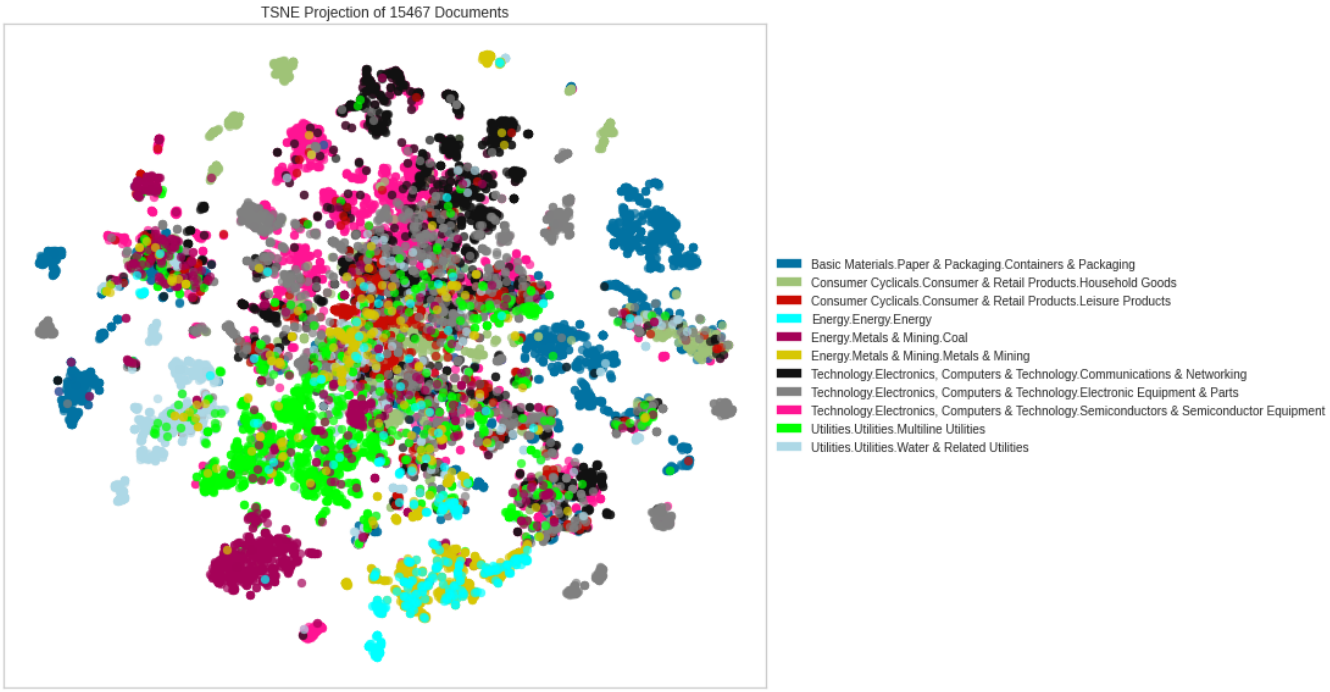

Fig 8: Visualization of word embeddings for different markets

In the future, we must consider labelling more texts for this market and balance it in the data set to improve overall metrics and models predicting power.
We implemented a multilabel classifier for the TRBC classification taxonomy in the level of market tags. The key components of the project include, but not limited to – advanced data preprocessing, training on the bigdata set by using distributed libraries, the release of the model to the production environment, and integration to the customer IT infrastructure. We could reach good results with the use of the most advanced architectures and manual review of the outliers. To be able to improve the results of the existing models and make a prediction on the bottom levels of the taxonomy will be required to extend the dataset for the low performing markets.

References

[1] Q. Le, T. Mikolov. Distributed Representations of Sentences and Documents. CoRR abs/1405.4053, 2014

[2] J. Pennington, R. Socher, C. D. Manning. Glove: Global Vectors for Word Representation. EMNLP, 14:1532–1543, 2014.

[3] C.M. Bishop. Pattern Recognition and Machine Learning. Chapter 4.3.4

[4] K. Crammer, Y. Singer. On the algorithmic implementation of multiclass kernel-based vector machines. Journal of Machine Learning Research, vol. 2, pp. 265-292, 2001

[5] C. Silla, A. Freitas. A survey of hierarchical classification across different application domains. Data Mining and Knowledge Discovery, 22 (1-2), 31-72, 2011

[6] http://flask.palletsprojects.com/en/1.1.x/

[7] https://www.docker.com/

[8] L. Maaten, G. Hinton. Visualizing Data using t-SNE. Journal of Machine Learning Research. 9: 2579–2605, 2008