Automatic Insurance Labeling Challenge

Veridion ML Project Documentation

Oproiu Matei

October 28, 2025

1 Introduction

1.1 Problem Definition

The task of this project is to automatically assign relevant *insurance-related labels* to companies based on textual information such as descriptions, business tags, and categories. The challenge lies in designing a method that can generalize well, handle noise in text, and scale efficiently across a large dataset.

1.2 Approach Overview

The first step was to identify this problem as an unsupervised multi-label classification problem, something I had never tackled before. I started by exploring the datasets, counting the number of companies (9494) and the number of labels (220). I also noticed that some companies have missing data, which we will need to address when the time comes. From the size of the data, it was obvious that we needed to implement an algorithm that was fast. I cleaned the data for any NaN values and extracted all the information about each company into one big string of text, which I will be referencing as **full_text**. My first thought was to implement a keyword search to search for the labels in the **full_text** of each company. This gave decent results, but only captured labels that were explicitly mentioned in the **full_text**. My next idea was to implement a term frequency method, an algorithm I was familiar with, but I realized that I had already implemented a similar method in the keyword search and I needed a method that would understand the data of each company and the labels conceptually, not just lexically. That is why the next method implemented was one that uses embeddings and cosine similarity to determine how similar a company's **full_text** was to each label. This process generated new results that captured relationships beyond word matches. While this worked much better, I noticed that some companies still received no labels. To address this, I tried to determine if a zero-shot model-based approach would yield better results, even though it is a lot slower. This method really understood the data it was given and it came up with labels that were really smart; some examples are given below.

2 Data Overview

2.1 Datasets Used

- **Insurance Taxonomy:** A CSV file (`insurance_taxonomy - insurance_taxonomy.csv`) containing a structured list of insurance categories.
- **Company Dataset:** A large CSV file (`ml_insurance_challenge.csv`) containing the following for each company: description, business_tags, sector, category and niche.

Remark: We will consider that the companies are indexed from 0

2.2 Data Preprocessing

NaN values and inconsistent formats were cleaned using a preprocessing function. All missing textual fields were replaced with empty strings.

3 Methodology

3.1 Overall Approach

The goal was to compare and combine multiple strategies for automatic labeling, including:

- **Keyword Matching**
- **Sentence Embedding Similarity**
- **Zero-Shot Classification (Transformer-based)**

Each method was tested individually and later combined to leverage their complementary strengths.

3.2 Method 1: Keyword Matching

Description: Each company's description and tags were scanned for exact or fuzzy matches with taxonomy keywords. We give 3 points for a whole label hit, 2 points for a bigram and 1 point for a unigram. We consider all labels with a score ≥ 2.
Pros:

- Simple and fast.
- Works well for explicit mentions.
- It takes around 2 minutes to label the whole dataset.

Cons:

- Sensitive to wording variations and synonyms.
- Can give false positives if keywords are ambiguous.

Notable examples:

- It works really well to label companies that provide multiple services. A good example can be found at index 2 in the CSV file. In the description of the company, at the very end, there is a mention that they offer coffee services: "In addition to their farm products, they also have a farm shop and cafe where customers can enjoy fresh coffee and delicious cakes". This would be ignored by the other methods, as it is not the main service that the company provides.
- A false-positive example can be seen at index 45. In the description, there is the phrase "as well as Rika stoves for both wood and pellets". Our method sees "well" and gives the obviously incorrect label "Well Maintenance Services".

3.3 Method 2: Embedding Similarity

Description: Used `SentenceTransformer` models (e.g., `all-MiniLM-L6-v2`) to embed both company texts and taxonomy labels. Cosine similarity scores determined label assignments. We consider all labels with a score ≥ 0.45

Pros:

- Captures semantic meaning beyond keywords.

- Robust to linguistic variation.
- It takes around 20 seconds to label the whole dataset.

Cons:

- Sensitive to embedding model domain.

Notable examples:

- To use the same company at index 2 as an example. It mentions, ”The company’s product range includes homemade bread,...” and the embedding model gives us the label Bakery Production Services”. A label that was invisible to the first method, as the word ”bakery” does not appear in the description.
- A bad example would be the company at index 8 with the description: ”The company specializes in the production and distribution of packaging materials, including cardboard sealing tape, wholesale packaging, and packaging supplies. They also offer printed packaging services.”. Where the method provides the labels: ”Paper Production Services, Ink Production Services”. These are partial labels, they are neither right nor wrong.

3.4 Method 3: Zero-Shot Classification

Description: Applied transformer-based zero-shot models (e.g., `facebook/bart-large-mnli`) to classify companies against candidate labels. I labeled the first 32 companies using this method as it takes extremely long to run.

Pros:

- Leverages powerful pre-trained NLI models.
- Provides accurate labels.

Cons:

- Slow inference time.
- It takes around 2 days to label the whole dataset.

Notable examples:

- The company at index 3 is an auto body shop, due to the fact that we have no label in the realm of ”Car Repair Services”, it was never labeled by the first 2 methods. The model had such a deep understanding of the data that it assigned the label ”Painting Services”. Which is correct, as an auto body shop that repairs cars will obviously also paint your car.
- I was testing random companies to see what labels this method produced, unfortunately, I didn’t save the output, but I think it’s worth mentioning. There was a company that provided marriage/couples counseling. As there was no easily accessible label, the model assigned the label ”Restoration Services”, which I found really funny.

3.5 Combination Strategy

To improve robustness, a hybrid approach was designed:

- **Keyword method** prioritized precision.
- **Embedding method** prioritized semantic recall.
- The final labels were merged and deduplicated, keeping only high-confidence matches.

Remark: If I had the time and the computing power, I would have also merged the results with the model predictions on the whole dataset.

4 Evaluation

4.1 Evaluation Philosophy

Traditional metrics such as F1 or accuracy were not applicable, since there was no labeled ground truth to compare against. Instead, I evaluated the results manually by inspecting a sample of companies and checking whether the assigned labels made sense. The evaluation focused on:

- **Qualitative analysis:** I manually reviewed sample outputs to see if the labels accurately described the companies.
- **Distributional sanity checks:** I verified the overall coverage, noting that around 75% of companies received at least one label.
- **Semantic coherence:** I checked whether the assigned labels were semantically related to one another.

5 Discussion

5.1 What Worked Well

- The keyword search method was able to find some relevant labels that the embedding-based method missed.
- The embedding-based method achieved strong semantic alignment.

5.2 What Didn't Work

- Zero-shot models were too slow for large-scale inference.
- Keyword matches occasionally introduced noise.

5.3 Possible Improvements

- Use a synonym dictionary to enhance the effectiveness of the keyword search method.
- Experiment with more domain-specific embedding models trained on financial or insurance-related texts to improve accuracy.
- Implement a threshold tuning step to automatically find the optimal threshold cutoff for assigning labels.

6 Conclusion

This project explored multiple methods for automated insurance labeling without supervision. Through a mix of embedding similarity, keyword matching, and transformer-based inference, the final approach achieved meaningful, interpretable results. However, this problem cannot yet be fully solved by automated methods alone. More research is needed, as human logic and contextual understanding still play an essential role in accurately interpreting company data and assigning the most appropriate labels.