TUM Data Innovation Lab

Munich Data Science Institute (MDSI)

Technical University of Munich

&

FSN Capital Partners

Final report of project:

Generating Investment Opportunities Using Large Language Models

Authors	Yurou Liang, Kilian Miessner, Mohamed Mokhtar Riahi, Anton Steuer, Yuqing Wang
Mentor(s)	MSc. Vincent Stoltenberg Wahl, MSc. Benjamin Grether
TUM Mentor	Dr. Pascal Heid
Project lead	Dr. Ricardo Acevedo Cabra (MDSI)
Supervisor	Prof. Dr. Massimo Fornasier (MDSI)

Friday 9th February, 2024

Acknowledgements

We sincerely thank Dr. Ricardo Acevedo Cabra and Prof. Dr. Massimo Fornasier for the initiative and extensive commitment to organizing the TUM Data Innovation Lab and finding incredible partners and inspiring projects. We are grateful to the Munich Data Science Institute for providing a nurturing environment for research and innovation. Many thanks to Dr. Pascal Heid, who provided valuable mentorship and feedback as a co-mentor from TUM.

A very special thanks to MSc. Benjamin Grether and To MSc. Vincent Wahl our mentors from FSN Capital Partners. Your experience in data science and commitment to academic rigor contributed greatly to the project's success. Your insights into machine learning and Large Language Models enriched our understanding and application of data analytics in real-world scenarios. Thank you both for always having an open door and providing extensive feedback and support throughout the whole project.

Abstract

Private equity investing involves the acquisition and management of non-publicly listed companies using investment funds, aiming to attain significant financial returns. Identifying companies that offer similar services within a subsector is crucial for a successful investment strategy. Comparison of relevant companies within those subsectors from a large superset of potential investment opportunities is a significant challenge private equity professionals face. However, state-of-the-art tools like company databases often lack the granularity to pinpoint clusters of companies operating in niche markets. Sourcing efforts are often characterized by high manual effort and inefficiency, making the process time-consuming and often unfeasible.

An automated approach could efficiently sift through large amounts of data and identify subsectors for subsequent in-depth financial analysis. This report examines an established methodology by FSN Capital Partners, including web scraping, data transformation, and relevancy scoring. This work implements potential enhancements and assesses their impact through a real-world example. The primary focus lies in training and comparing Support Vector Machines, Decision Trees, and Random Forests to effectively categorize companies in subsectors based on user-selected keywords. Furthermore, the report explores the potential of incorporating Large Language Models (LLMs) to assist investors with curating high-quality keyword lists to identify relevant companies within specific niches. These approaches are evaluated on their ability to accurately group companies into subsectors within the surface treatment industry and their ability to generalize to different domains.

The results indicate that a Random Forest model has high generalizability, satisfying performance, and high robustness. Keyword Generation with the support of LLMs shows high potential. Adjustments in each step improve generalization towards application in different domains. This project's results showcase a potential way to include web scraping, machine learning, and Large Language Models in private equity sourcing. The findings establish a solid basis for future exploration in the automation of sourcing processes and how private equity firms tackle the challenge of identifying high-potential investment opportunities.

Glossary

Abbreviation	**Meaning**	**Page No.**
PE	Private Equity	1
FSN	FSN Capital Partners or FSN Digital	2
PortCo	Portfolio Company	3
NACE Code	Statistical classification of economic activities in the European Community	4
M&A	Mergers and acquisitions	4
LLM	Large Language Model	5
SVM	Support Vector Machine	5
BFS	Breadth First Search	9
LDA	Latent Dirichlet allocation	11
NMF	Non-Negative Matrix Factorization	11
BERTopic	Topic Modelling Framework	11
API	Application Programming Interface	11
c-TF-IDF	Class-based Term Frequency-Inverse Document Frequency	11
GPT-3.5	Generative Pre-trained Transformer 3.5	13
GPT-3.5-Turbo	A variant of GPT-3.5	13
GPT-4	Generative Pre-trained Transformer 4	13
DT	Decision Tree	16
RF	Random Forest	16
F1-Score	Harmonic mean of a system's precision and recall values	17

Table 1: List of Abbreviations

Contents

1 Introduction

1.1 Introduction to Private Equity

Private equity, or PE, is a form of capital investment in private, usually unlisted, companies. The primary objective of PE firms is to become actively involved in the acquired company's operations, increase the company's value, and ultimately exit the investment for a desired return. PE is part of the alternative investment segment of the financial sector and typically operates with a closed-end fund structure. The funds are conventionally structured as a limited partnership, with the PE firm acting as the general partner (GP), also known as the Sponsor. Most of the fund's capital providers are called limited partners (LPs) [1].

The overall private equity process can be divided conventionally into fund and deal level processes (see figure 1). During the fundraising process for the closed-end fund, the GP approaches large institutional investors and capital providers such as pension funds, endowments, insurance companies, sovereign wealth funds, and family offices or high-net-worth individuals to raise capital. The committed capital in leveraged buyout funds is used to acquire private companies, called portfolio companies. The capital in a close-ended fund can not be withdrawn during the fund's lifetime. Depending on the size of the fund, the PE firm usually invests in 10-12 portfolio companies. The investments are managed by the GP, who traditionally sits on the board and controls financial and operational results on a monthly or quarterly basis [2].

The PE investment process is a multi-stage process, with each stage being critical to the realization of the investment (see figure 1:

1. **Sourcing:** The initial phase involves the strategic identification of potential investment opportunities. This intricate process requires a profound market comprehension and the capacity to identify companies that offer propitious investment prospects that align with the fund's strategy.

2. **Due Diligence:** Following the sourcing of a target company, an exhaustive due diligence phase ensues. This thorough investigation scrutinizes the financial, legal, operational, and strategic dimensions of the potential investment, which is essential for understanding the feasibility and risks associated with the investment.

3. **Acquisition:** Successful due diligence leads to the acquisition phase, which includes detailed negotiations, deal structuring, and the arrangement of necessary financing to procure the target company, culminating in the legal acquisition of ownership or control by the PE firm.

4. **Value Creation:** Post-acquisition, the emphasis shifts to the enactment of value creation strategies within the portfolio company. These strategies aim to enhance operational efficiencies, catalyze growth, and improve profitability, thereby increasing the overall investment value.

5. **Exit:** The final stage in the PE process is the exit, where the strategy is implemented to sell the portfolio company profitably. This can take several forms, including an initial public offering (IPO), sale to another PE firm, strategic sale, or recapitalization. The exit is pivotal as it is when the PE firm and its LPs realize the investment's returns.

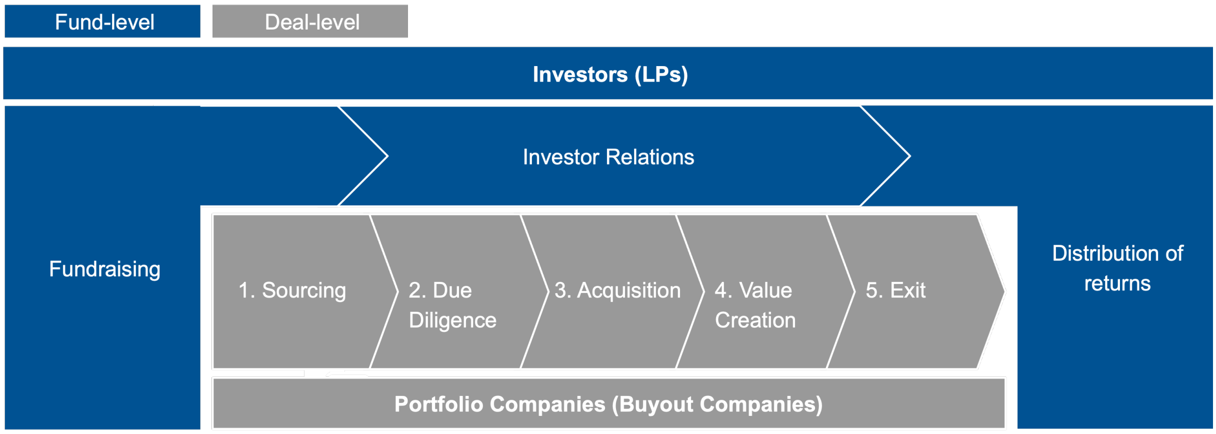


Figure 1: Private Equity fund and deal level process, adapted from [3]

1.2 Introduction to FSN Capital Partners

FSN Capital is a prominent private equity firm with a significant presence in Northern Europe. Established with a vision to drive substantial growth and returns, FSN Capital has developed a formidable reputation in the industry. Key highlights of the firm include:

- **Geographical Footprint:** The firm boasts a network of four offices strategically located across Northern Europe (HQ in Oslo and offices in Stockholm, Copenhagen, Munich) with nearly 100 team members facilitating a wide-reaching influence in the Nordics, DACH and Benelux regions.

- **Portfolio and Global Operations:** The firm manages 30 portfolio companies with operations across 128 countries.

- **Financial Management:** Approximately EUR 4 billion are currently under the firm's management.

- **Investment track record:** FSN Capital has achieved a realized track record of 2.9x Return on Investment (ROI) and a 33% return per annum.

Over the years, the firm has seen a substantial increase in the size of its funds, as evidenced by the growth from FSNC I in 2000 (EUR 54m) to FSNC VI in 2020 (EUR 1,800m). This trajectory reflects the firm's expanding influence and capability in the private equity sector. FSN Capital's investor base is diverse, spanning various regions and sectors, including pension funds, insurance companies, funds of funds, and other institutional investors from Europe, Middle East, Asia, and North America.

FSN Digital 5 Year Vision: In its 5-year vision, FSN Capital aims to reinvent the private equity business model, becoming a leading next-generation fund. This includes digitizing internal processes, investing in digital business models, and enhancing digital capabilities in all portfolio companies.

Focus on Digitalization in the Private Equity Industry: FSN Capital is intensively focused on incorporating digitalization across various aspects of the private equity industry. This includes key areas such as fundraising, deal sourcing, due diligence, portfolio company (PortCo) engagement, and exit strategies.

- In **fund raising**, the firm differentiates itself by highlighting its unique digital capabilities, showcasing how it stands apart from other funds in terms of digital proficiency.
- During **deal sourcing**, FSN leverages its proprietary analytics platform, CORE, to conduct deep analyses across company data sets and utilizes web scraping techniques for enhanced deal identification.
- The **due diligence** process for identified targets, FSN puts strong emphasis on identifying digital risks and opportunities, based on a comprehensive evaluation framework for technology stacks, IT operations and digital marketing.
- In terms of **PortCo engagement**, FSN focuses on identifying high impact digital use cases for portfolio companies, facilitates digital strategy setting, runs portfolio-wide digital initiatives, and builds a strong partner ecosystem to support digital transformations.
- Finally, for **exits**, the firm supports the exit story by highlighting digital success cases and capabilities that have been built, demonstrating opportunities and the readiness of the exited companies for further growth.

This strategic emphasis on digitalization enables FSN Capital Partners to stay ahead in the evolving landscape of private equity, ensuring both innovation and strong returns.

1.3 Sourcing strategies

Conventional sourcing approaches: Deal sourcing, as reflected in the private equity process, is key to the success and reputation of private equity firms [4]. Deal generation is important at two stages: when identifying new investment opportunities for platform companies or during portfolio engagement in add-on acquisitions. It is estimated that around 1-2% of the deals that enter the sourcing pipeline will be closed and acquired [5]. This highlights the importance of effective sourcing regarding time and financial resources. Key deal origination strategies to maintain a steady and viable deal flow for the fund include

- Proactive deal origination
- Solicitation of other funds to syndicate

- Investment banks, M&A advisors and consultancies
- Online deal sourcing platforms
- Referrals from portfolio companies

Proactive sourcing usually consists of analyzing different subsectors. A subsector is a narrowly defined category of businesses that engage in similar activities, sell comparable products or services, and are subject to the same market trends. These are typically associated with specific geographic regions and consist of a handful of directly competing companies. A study of 150 PE firms shows that private equity funds with a proactive origination consistently achieve higher returns. This success is attributed to the increased number and greater relevance of the investment opportunities they receive [6].

1.4 Challenge of sourcing and project goals

Identifying high-quality investment targets with manageable risks is a significant challenge in the field of private equity. Overall, two critical factors determine the success of an investment during the holding period: 1) The performance dynamics of the specific subsector where the target company operates, and 2) the comparative operational success of the company within this subsector.

Navigating through the extensive pool of potential investments is a time-intensive and complex task. The granularity required to filter these businesses for investment analysis efficiently often surpasses the capabilities of standard industry classification codes like NACE (Nomenclature of Economic Activities) [7]. Enterprises often also fall into several NACE code classes. To put the amount of research and lack of granularity of the NACE code approach into context, the 26 million active enterprises in Europe in 2020 are divided into 625 NACE code classes. This means that on average there are 40,000 enterprises per NACE code. The NACE codes do not include novel subsectors and new business areas. For example, separating biotech companies into meaningful subgroups based on their unique product lines or market niches is a challenging and highly manual process. The aim during sourcing is not to compare biotech companies broadly but to differentiate them based on their specific focus areas, such as a company specializing in making insulin meters with one focusing on protein folding software. Current deal-sourcing platforms lack the desired granularity, often providing only a vague business description and general subsector categorization.

FSN Capital's objective in innovating the sourcing of attractive companies and subsectors can be divided into:

1. Systematically grouping companies into relevant granular subsectors.
2. Assessing the potential of these subsectors.
3. Evaluating the performance of individual companies within these subsectors.

A possible approach for achieving these objectives includes utilizing a combination of industry and company-specific keywords to analyze company websites, coupled with advanced keyword search methodologies, clustering algorithms, and the use of Large Language Models (LLMs) [8].

1.5 FSN Roll-up sourcing use-case in the surface treatment industry

As part of FSN's overall objective to improve the efficiency of identifying and evaluating investment opportunities in different subsectors, an experimental use case was created for automated sourcing, based on a roll-up investment in the surface treatment industry. The portfolio company, created in 2020 through the merger of 30 smaller companies, has expanded its operations across several European countries and diversified its service offerings within the surface treatment sector. The roll-up case with the acquisition of 30 companies highlights the necessity of FSN to efficiently identify small companies active in a given subsector.

The methodology implemented by FSN in this case involved an automated screening of the German market, aiming to identify potential expansion targets. The process encompassed:

1. **Data Acquisition:** Collecting extensive information about various companies, including names, websites, and other pertinent details.
2. **Keyword Analysis:** Creation of specific keyword lists related to subsector and subsequent quantification within the scraped content.
3. **Relevancy Classification:** Sorting companies based on their relevance to the company's strategic goals.
4. **Data Enrichment:** Enhancing the dataset with additional information from various sources.

As a starting point for the FSN Capital roll-up sourcing use case, the FSN digital team trained a Support Vector Machine (SVM) for ranking companies based on their relevance to the targeted surface treatment industry. The process involves several steps, focusing on website content analysis and keyword relevance.

Process Overview The first established methodology by FSN Capital uses an SVM ranking process. It begins with the collection of company websites and proceeds through six stages of analysis:

1. **Websites Collection:** Initiate by assembling an extensive list of company URLs from the FSNs company database. Subsequently, employ NACE codes to reduce the list down to a manageable subset of companies.
2. **Manual Sampling Relevancy:** Sample and categorize 300 selected companies based on their fit to market segments such as painting, flooring, tiling and masonry. Of these, 128 were categorized as relevant, 97 were related but not relevant, and 75 had no related services.
3. **Keywords Identification:** Create a list of relevant keywords specific to each of the categories of painting, flooring, tiling, and masonry in close collaboration with investment team members, the portfolio company executives, and internet research.

4. **Keyword Count:** Algorithm counts the occurrences of these keywords on each company's website.

5. **Parameterization:** Convert keyword counts into parameters for the SVM analysis.

6. **Relevancy Calculation:** Use the SVM to classify the company as relevant, related but not relevant, or unrelated.

Selected Features The SVM gets four different features as input to determine the relevance of a company for a specific subsector/category:

- Number of distinct keywords mentioned per category.
- Percentage of pages mentioning each category.
- Count of keyword occurrences on the landing page for each category.
- Share of pages with category weighted by page rank.

Relevancy Based on these features, the SVM calculates a relevancy score for a company for each category separately. This score lies can be interpreted as the probability of the company belonging to the specific subsector. The threshold is defined and based on the count of keyword mentioned on a company's website, which is used as a preliminary filter before applying the SVM. Any company with fewer than three keyword occurrences is filtered out **before** the SVM, as it is unlikely to be relevant to our field of interest. The SVM is then used to classify the remaining companies into **'relevant'** or **'not relevant'** without calculating a probability of relevance.

- **Relevant Companies**: These are companies with three or more keyword occurrences. When tested on a total of 41k reviewed companies, the SVM classifies 9,353 as relevant.
- **Related, But Not Relevant**: Although these companies have three or more keyword occurrences, they are categorized as not relevant. There are 11,363 companies in this category.
- **No Related Services**: Companies with fewer than three keyword occurrences fall into this category, as they lack sufficient relevance to the service fields in question. This is the largest group, with 21,403 companies.

In figure 2 the total breakdown of relevant companies extracted from FSN's company database starting from the NACE code up to the relevant companies for the roll-up case is shown.

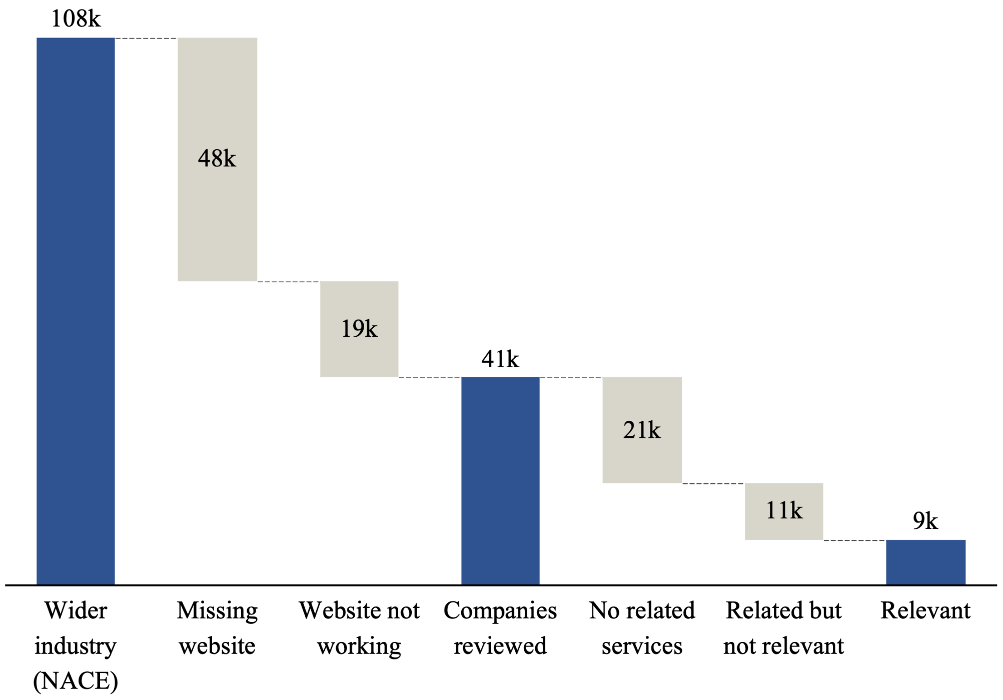


Figure 2: Relevant Companies for surface treatment roll-up case

FSN Capital's first experiment shows significant potential in the process of identifying and evaluating potential investment opportunities in niche subsectors. While the approach has notable strengths in efficiency and precision, it also presents areas for future enhancement, particularly in scaling and automatization. Table 2 summarizes the advantages and disadvantages of the SVM approach with manual company relevance labeling and keyword generation:

Criteria	**Strengths**	**Weaknesses**
Manual Labeling of companies	Significant reliability and control in categorizing companies as relevant to a given subsector.	Time-consuming and not easily scalable as required for each SVM model
Manual Keyword Generation	Manual keyword list portrays precisely the targeted business subsector.	Creating an extensive list necessitates significant time and deep understanding of the specific business model.
Accuracy and Precision	High accuracy and precision in classification of companies in trained and tested subsector.	Important variation of performance varies across different market segments.
Language Dependence	Effective within the scope of languages the model processes.	Limited to the model's language capabilities.
User Interaction	Easy to understand for user with no technical background.	No prototype or user interface available

Table 2: Strenghts and weeknesses of established methodology by FSN

Large language models and other machine learning algorithms have great potential to address such shortcomings. The experimental use case provided by FSN Capital will be used as a starting point to improve and apply such models, ultimately analyzing whether performance, efficiency, scalability, language dependency, and inclusion of user interactions can be improved without compromising speed and accuracy.

2 Project

Dataset The Dataset provided by FSN describes the real-world case of classifying companies in the surface treatment industry. A set of 300 companies has been labeled to belong to several categories/subsectors: painting, flooring, tiling, and masonry. Each company is represented by an ID from FSN's company database and a URL pointing to the company's landing page. FSN implemented a small web scraper to get the content of a company's homepage and the subsites referenced directly on it. Furthermore, a list of keywords for each category was provided. Based on this data, FSN implemented an example Support Vector Machine (SVM) to showcase the high-level approach: counting keywords on websites to later use as input features for a machine learning model to classify companies.

Project Overview The overview of the different project components can be found in figure 3. Starting with the web scraper, we enhanced the data collection to be executed recursively to traverse each company's entire website instead of only the subpages listed on the landing page. The websites are utilized in two ways: firstly, they will be used to count keywords, which will act as features for machine learning models. Secondly, the websites build the basis to explore the capabilities of large language models (LLMs) to assist with curating keyword lists. On the one hand, a topic model based on the framework BERTopic will extract keywords from the website corpus [9]. On the other hand, GPT-4 will be utilized in a feedback loop where the user creates and iteratively improves keyword lists. We will assess the capabilities of Support Vector Machines, Decision Trees, and Random Forest Models to predict a company's relevance for a subsector based on its cleaned website and a list of keywords.

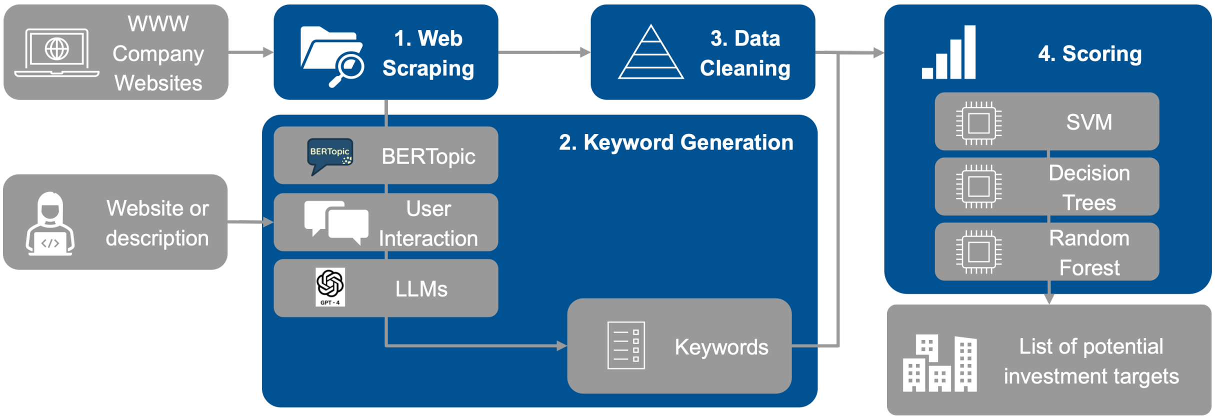


Figure 3: Information and interaction flow chart of proposed project

2.1 Web Scraping

Even though business databases provide a comprehensive company description, this summary is usually very high-level and vague. It lacks the necessary depth to categorize companies into the corresponding market niche. The best and most up-to-date description of a company and its services available to the public is its website. For this project, a proof-of-concept web scraper is implemented to gather as much information as possible simply from a list of URLs that can be exported from the business database, for example, by getting all companies of a particular NACE code. While this scraper is not developed to be used at an industrial scale due to not having any precautions to avoid being detected as a bot, the algorithms can be used, scaled, and applied for different cases. The following algorithm is executed simultaneously across multiple processes to parallelize the website scraping, increasing its efficiency.

Website Structure Looking at a website, starting at the landing page, one can navigate between pages using the hyperlinks on the current page. A website can be seen as a directed graph using the (sub-)pages as nodes and the links as directed edges. More than that, in addition to internal links, a website can reference other external websites, resulting in a vast graph representing the internet. The main challenge is to efficiently traverse this graph, starting from the landing page, to reach all the relevant subpages of a company's website. The graph traversal algorithm Breadth-First-Search (BFS) poses a solution to this challenge. The algorithm gets a company's landing page and recursively collects subpages of it, as described in the next section.

Scraping Algorithm The web scraper requests the homepage and extracts all links from its source code. Each URL is checked to see if it points towards a subsite of the company's website. This approach ensures that the search process is kept within the target website and does not branch out to different, company-unrelated websites. Furthermore, the algorithm checks for specific negative keywords within the URLs to skip irrelevant subsites like Imprints, GDPR, Cookie Information, and file links, which are excluded from the search since they do not contain relevant content. Once all URLs from the landing page are collected, this procedure is recursively executed for each of the collected URLs, resulting in a complete website traversal.

Depending on the overall structure of the target website, it may have a lot of different subpages, for example, hundreds of blog posts. An analysis of the websites for the surface treatment industry indicated that most companies in this example dataset have between 5 and 50 subsites. Companies that have a very large number of subsites would be over-represented in the final dataset. To avoid this, the algorithm gathers the top 75 subpages of a website and applies a relevance ranking formula described in the next section to identify the most relevant pages.

Subsite Ranking Collecting more than the final goal of 50 sites enables the application of custom ranking criteria. BFS first visits the pages closest to the home page regarding the number of URLs needed to get to that subpage. However, it does not take into account the length of URLs. Therefore, it is possible that a page with a very long URL that is

very far from the home page could be directly linked to it, for example, in a glossary, and be falsely considered important in this project.
The web scraper applies a simple ranking formula independent from the graph structure of the website: the longer the URL to a subpage, the less important it is considered. The URL length here is defined by the number of parts the URL contains. The homepage, has the shortest URL and only the base domain (for example, www.example.com); thus, it is considered the most important site. At the next depth, URLs with one component are considered (for example, www.example.com/page). If there is a draw in the number of URL components, the number of characters in the URL determines which subpage is most relevant.

Website Storage Once the algorithm has collected at most 75 subpage-URLs and ranked them according to the previously defined metric, it requests the top 50 websites and stores them locally. If a website contains less than 50 subpages, all are stored independently of their ranking. The algorithm saves the decoded content unprocessed in disk as serialized objects, which can be simply loaded again and used later. In this project, website content is used in two ways: On the one hand, neural topic modeling coupled with LLMs generates keyword recommendations based on the entire website corpus. On the other hand, the keyword occurrences on each website are counted and used as features for the machine learning models. Depending on the application, the website content is preprocessed differently. Topic modeling requires semantically complete text passages, while keyword counting is semantically independent. In concrete terms, keywords can also occur out of context, such as a button title in the source code, which does not pose a problem for the counting process. For topic modeling, however, these out-of-context words and phrases are filtered out, as they can disrupt the flow of the website's main text, depending on the layout. The approach to extracting potential keywords from the website corpus is described in the following section.

2.2 Keyword Extraction and Generation

The machine learning models implemented to perform relevancy scoring of companies use word counts as features, described in detail in 2.3. The high-level approach to creating meaningful model inputs is to create a list of keywords for a subsector and count their occurrences on each website.
A significant challenge FSN identified during the established methodology's first implementation was the creation of meaningful keywords. Keywords have to be distinctive and represent precisely one subsector, meaning the keywords describing one subsector should not be so ambiguous that they may also be used in a different domain. However, the keywords should also be broad; otherwise, they may not appear in the available website content. Company websites are usually used to describe the overall business and its offerings rather than to provide in-depth technical advice about its services.
FSN's digital team, in collaboration with the investment team, portfolio company experts and extensive internet research, curated a comprehensive list of keywords for the surface treatment industry. This list is the baseline for the initial experiments. The keyword list has been continuously improved and extended to find an optimal set of keywords for each subsector in this specific case. The keywords strongly influence the models' results. The

process of creating a comprehensive keyword list is very time-consuming. In the future, however, the user should be able to generate a keyword list in minutes rather than months and still get satisfactory results. Furthermore, the approach described in the following sections addresses the problem of scaling and can be adapted to generate keywords for novel cases in different domains.

2.2.1 Keyword Extraction using Neural Topic Modeling

The first approach aims to use Topic Modeling to identify the most relevant topics in our website corpus. The idea is that when applying topic modeling on the companies' websites, the topics companies in the same subsector write about should be similar. For example, companies offering painting services will likely write about paint, walls, and facades, while flooring specialists will discuss different flooring materials and techniques. The aim is to identify dominant topics from the corpus, represented by their most important terms. Finally, GPT-4 is used to condense a list of keywords for each category, using only the topics represented in the corpus.
While conventional techniques used for Topic Modeling, such as Latent Dirichlet Allocation (LDA) and Non-Negative Matrix Factorization (NMF), are based on statistical properties within texts in a corpus, more recent approaches use Embeddings of Language Models, e.g., Top2Vec and BERTopic [9, 10, 11, 12]. Those neural approaches tend to perform better on noisy datasets, have better distinctions between the identified topics, and the topics are more specific and relevant [12]. The BERTopic algorithm, one of the latest approaches to neural topic modeling, is chosen because it has less overlap between topics than Top2Vec [12].

BERTopic BERTopic's high level of modularity and flexibility, coupled with state-of-the-art performance, makes it an attractive framework for the project use-case. On a very high level, the pipeline of BERTopic uses Transformers to embed documents and clusters those documents based on their embeddings. It uses a variation of TF-IDF to extract the most relevant words per cluster as the topic representation for this cluster [9].
The pipeline consists of five main building blocks, each with multiple options in algorithms to choose from [1]:

1. **Embeddings**: transformer model to translate your corpus into the embedding space.

2. **Dimensionality Reduction**: algorithm to reduce the dimensionality of the embeddings for the clustering algorithm to work on.

3. **Clustering**: algorithm to build groups of embeddings/documents close in the embedding space.

4. **Vectorizer**: algorithm to prepare the documents for the extraction of keywords for the topic representation.

5. **c-TF-IDF**: adjusted TF-IDF algorithm to perform on groups of documents instead of documents (class-based TF-IDF).

[1] https://maartengr.github.io/BERTopic/algorithm/algorithm.html

Data Preprocessing Before we built our custom topic modeling pipeline, we needed to preprocess the websites we scraped beforehand. Two things are essential when working with LLMs and Embeddings: firstly, we want to have our documents as semantically complete as possible to compute good embeddings that carry much information. We get raw website texts from our Scraper, i.e., all visible words on the subsite. Those include the site's primary content and a lot of noise that does not carry useful information for topic modeling, such as button titles, image links, cookie banners, etc. Secondly, the length of the document has to stay within the token limit of the chosen model to not lose any information that is not embedded [2].

To achieve this, we iterate over each stored website and clean it by removing the header, footer, sidebar, and cli-privacy-content (associated with cookie information). After this, we extracted the main text from the website. We get all strings from the content and consider it a "semantically complete sentence" if it contains more than five tokens and punctuation. To build text chunks that contain complete sentences and are within the token limit, we apply the following formula: the number of sentences to use per chunk is equal to 80% of the token limit divided by the average number of tokens per sentence. At the end, we store each subsite as a list of cleaned and semantically complete text chunks, which we will use as documents for our topic model.

It is essential to mention that we do not track which documents belong to which company and, further downstream, how specific companies are grouped in clusters. The main focus of this experiment is to assess the feasibility of a topic model to extract high-quality keywords from a website corpus without matching the keywords to companies or back-tracking document clusters to company clusters. Furthermore, while implementing the topic model, we discovered that less than 50 subsites per website result in topics/keywords more relevant to our surface treatment case. As explained in 2.1, a few companies with a high number of subsites contain a lot of noise, decreasing the overall topic quality. The final input for our topic model described in the next section is the top 30 subsites per company, processed into a set of roughly 40.000 text chunks/documents.

Topic Model In this section, we briefly describe or design decisions when building the topic model for the surface treatment roll-up case. Since no one-fits-all solution exists in topic modeling, the hyperparameters are chosen mainly by subjective assessment of the resulting topics. This work aims to try and assess different approaches to get keywords that should be reproducible and interpretable. Regarding the clustering algorithm, we opted for K-means instead of the default HDBSCAN implementation since its topic output is more consistent across different runs [13]. A drawback of this approach is that the number of clusters has to be pre-defined by the user, whereas HDBSCAN determines it at runtime dynamically. There are methods like the Elbow-Method for Clustering algorithms to determine the optimal number of clusters for this building block. While this algorithm is widely used, there is also some criticism on its generalization ability [14]. For this project, the number of clusters is chosen by manual analysis of the topic quality. The following section introduces the specific components of each building block and discusses the parameter decisions.

1. **Embeddings**: the websites in the surface treatment roll-up case example are Ger-

[2] https://huggingface.co/sentence-transformers/paraphrase-multilingual-MiniLM-L12-v2

man, so we use a multilingual sentence transformer to compute embeddings [15]. A sentence transformer suitable for this task is ”paraphrase-multilingual-MiniLM-L12-v2”[3], which can also be used to generalize this approach for different languages in the future.

2. **Dimensionality Reduction**: we use the default UMAP algorithm implementation as provided by the BERTopic framework for dimensionality reduction of our embeddings.

3. **Clustering**: we are using K-Means with n=600 clusters for our embedding clustering. The number of clusters has been chosen by subjective analysis of topic quality using different amounts of clusters for multiple iterations.

4. **Vectorizer**: we adapt the CountVectorizer to our use case by removing German stopwords and allowing unigrams (single words) and bigrams (two words) as topics/keywords. To prevent a large vocabulary when computing the c-TF-IDF, we only consider words applicable as keywords if they occur at least 20 times within a cluster. A reduced vocabulary reduces the computation needed to weigh potential keywords for the topic representation.

5. **c-TF-IDF**: we use the class-based variant of TF-IDF introduced in [9] to extract our keywords but reduce the weight of frequent words. Those words are not considered stopwords but occur frequently across all clusters and thus are not descriptive for one cluster.

GPT-4 Connection After we run our topic model, we get a large data frame consisting of 600 rows (one for each cluster) containing a list with the ten most important words per cluster and a cluster name of the form ”n_keyword1_[...]_keyword4”. For extraction of the most important keywords for our domains, we are using GPT-4 in two iterations on the top 100 clusters:

1. **Cluster Relevance**: we provide GPT-4 with engineered prompts, our domains of interest, and only the names of the top 100 clusters. GPT-4 has the task of returning a list with only those cluster names that are related to our domains.

2. **Keyword Extraction**: based on the list of important clusters, we filter out the data frame to get all the keywords for each cluster. We provide GPT-4 with engineered prompts, our domains of interest, and each important cluster with its keywords. GPT-4 selects the ten best keywords per domain we provided and returns a list of keywords per domain. The keywords for the surface treatment example can be found in the appendix (6.1).

The most important attribute of this list of extracted keywords is that every word on this list is contained in our corpus, i.e., on a group of websites. This list can now be used as context for GPT-4 to generate more keywords for the same categories iteratively based on existing terms. This way, we hope to prevent GPT-4 from being ”too creative” when generating keywords, which might result in great-sounding keywords that are not actually used by websites and thus would not be descriptive features.

[3] https://huggingface.co/sentence-transformers/paraphrase-multilingual-MiniLM-L12-v2

2.2.2 Keyword generation with LLMs

The other approach used for generating keywords is using Large Language Models. The Langchain [4] library is selected to use the GPT-4-Turbo model of OpenAI [5]. Each time keywords must be generated, the GPT model's API is called using the Langchain library in Python [6]. Two different types of prompts are implemented:

1. **Purely using the user input**: The user provides a company description that will be used in the prompt to give context to the model. The user also provides the language of the generated keyword and possible categories of the desired generated keywords.

2. **With the integration of the results of using BERTopic**: In addition to the previously mentioned inputs, the keywords generated by the semantic topical algorithm BERTopic enhance the prompt provided to the model.

Several LLMs such as GPT-3.5, GPT-3.5-Turbo and GPT-4 models are tested. The output of the API calls varies considerably depending on the model. The simpler models tend to hallucinate and generate some keywords unrelated to the company description. In addition, it is necessary to format the keyword list as a comma-separated string to proceed with the scoring algorithms. However, the simpler models, for example, GPT-3.5, struggled to output the list in the desired format.

2.2.3 User feedback and interaction

Since the GPT model may generate keywords unrelated to the business description, the user must have control during the interaction between the generation process and the final selection of desired keywords. The user interaction flowchart can be observed in 4. A conversational process is developed to give the user control over which keywords are used to categorize companies. The user and the large language model interact in a chat-based form until the desired granularity and specific keywords are generated. The user first provides the model with a business description or a website summary. The GPT4 model then combines the user's input with the categories collected by the BERTopic 2.2.1 algorithm and suggests possible overarching business categories to the user. The user is allowed and encouraged to modify or provide more specific categories. The LLM then generates keywords based on the context of the information provided by the user and the information extracted from BERTopic. The first iteration of keywords is presented by the GPT, and the user decides whether further keywords should be generated by the model. Throughout these steps, the user remains in control and can manually add keywords. Finally, the generated keyword list is ready to be fed into the scoring algorithms, which will be discussed in the next chapter.

[4] https://www.langchain.com

[5] https://openai.com

[6] https://www.python.org

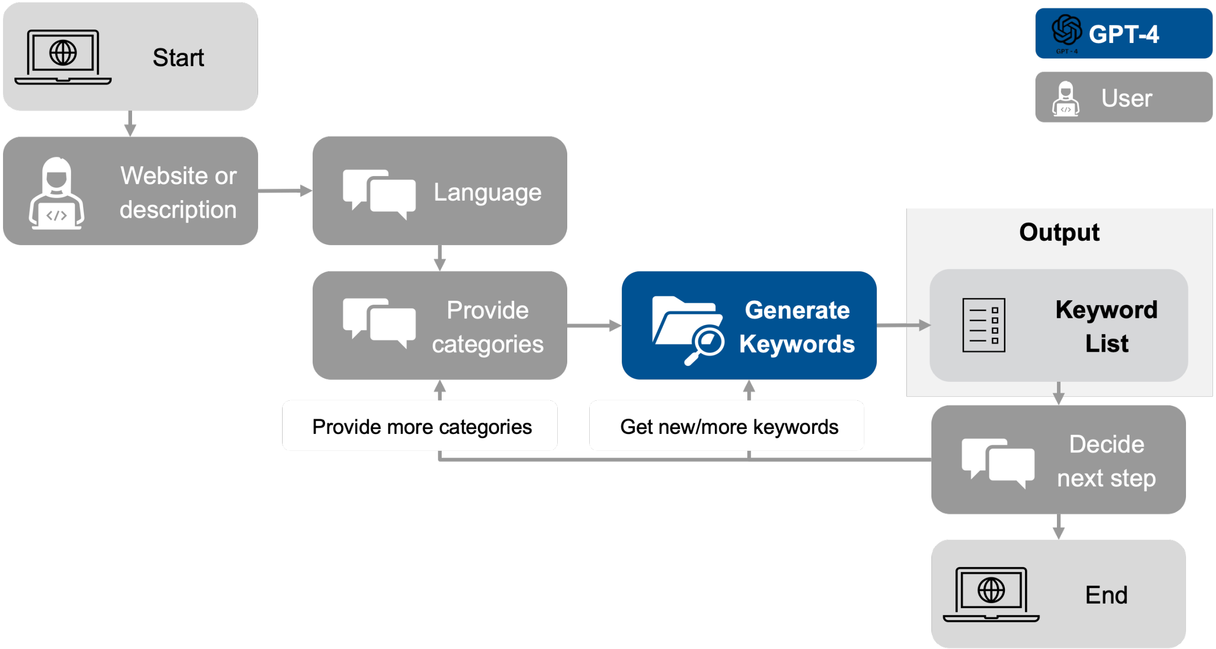


Figure 4: User interaction with Large Language Model GPT-4 for iterative keyword generation

2.3 Scoring Algorithms

Several machine learning algorithms are implemented and compared during the project. In order to provide a sufficient overview of the models used in the project, the following chapter introduces the models used, namely Support Vector Machines, Decision Trees, and Random Forests. These models are considered due to the reduced amount of manually sampled data available. The chosen models have a simple structure for binary classification, which is useful for the given problem of category classification. The metrics used to evaluate the performance of the models are also introduced in the following part.

i) Support Vector Machine (SVM) [16]:
A support vector machine constructs a set of hyperplanes in a high-dimensional space. A good separation is achieved by the hyperplane that has the largest distance to the nearest training-data point of any class (so-called margin), since the larger the margin, the lower the generalization error of the classifier, which means the less likely the classifier encounters overfitting.

ii) Decision Tree (DT) [17]:
A decision tree is a flowchart-like structure in which each internal node represents a test on an attribute. Each branch represents the outcome of the test, and each leaf node represents a class label. A decision is taken after computing all attributes. The paths from root to leaf represent classification rules.

iii) Random Forest (RF) [18]:
A Random Forest is an ensemble learning method by constructing a multitude of decision trees at training time. For classification tasks, the output of the random

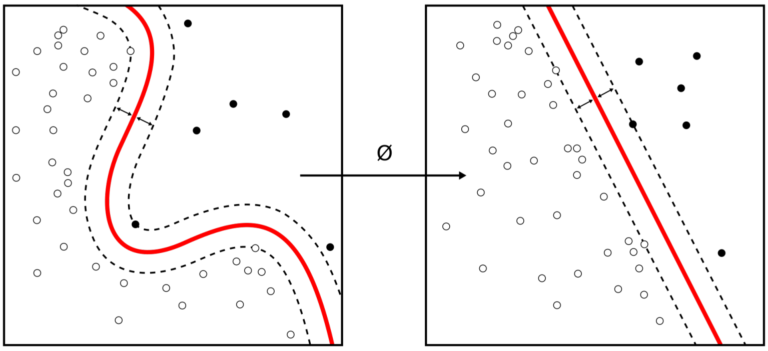


Figure 5: Intuition for margin of SVM

forest is the class selected by most trees. Random Forest corrects the potential overfitting of decision trees to their training set.

2.3.1 Basecase: Initial SVM Model by FSN

Based on the data of the surface treatment roll-up case sample, FSN aims to capture companies belonging to at least one of the subsectors of painting, flooring, tiling, and masonry. Those are treated as positive classes. Negative classes are also introduced to avoid extracting companies that provide unwanted services. Certain business classes within the surface treatment roll-up case sample that are indifferent to inclusion have been categorized as neutral classes. FSNs manually provided keyword lists for both German and English positive, negative, and neutral classes.

- Positive classes: Painting, Flooring, Tiling, and Masonry.
- Negative classes: Carpenter, Window and Turnkey.
- Neutral classes: Drywall, scaffolding and Contractor.

The relevancy label is utilized to indicate whether a company belongs to at least one of the four positive categories or not. Following this, keyword-based features are generated for each company. For each class within positive, negative, and neutral categories, the following features are collected:

- p1_number_of_occurrence: the number of keyword occurrences on the website landing page of the company.
- %p_occurrence: the percentage of subpages that mentioned one of the keywords in the given category among all subpages.
- pwTop10: the percentage of keyword occurrences divided by the number of all word occurrences for each sub-page.

For each positive class, the following feature is considered:

- number_of_occurence: the number of specific keyword occurrences.

In summary, the initial SVM model by FSN employs a total of 33 features. After the feature generation, samples with fewer than 2 occurrences in total for positive keywords are filtered out. The resulting dataset consists of 171 samples, of which 74 belong to the positive class, representing approximately 43.3% of the filtered dataset. Subsequently, a support vector machine (SVM) model is utilized to classify whether the sample is relevant or not.

While the first developed SVM model by FSN demonstrates comparable performance on the surface treatment roll-up case sample data, its primary design is to determine whether each company belongs to one of the positive classes within the dataset. Additionally, the model requires features related to negative and neutral classes in the dataset. However, when applied to a new dataset with different companies or subsectors, the SVM model faces generalisation challenges. The overall goal of a classification model is to prioritise flexibility in search content for users, rather than relying on pre-determined categories. Users are not expected to have a comprehensive understanding of the structure of the dataset, such as the different business classes in the dataset. As a result, the model and its features cannot be directly applied to other industries.

2.3.2 Data Transformation

Recall that a list of 300 companies where each company has labels indicating if it belongs to the following categories is provided in the manually sampled dataset by FSN: Painting, Flooring, Tiling, and Masonry. The curated keyword list created by FSN, which assigns important keywords for each above 4 categories, is available in German and English.

As mentioned above, the experimental SVM model is not scalable to other potential future subsectors. The model must be flexible for the user and not bound to predefined categories. In this case, storing a separate scoring algorithm for each potential category that the user may want is not scalable, so it is decided to develop a unique scoring algorithm for each category. Based on this idea, each company from the company list is divided into four data observations, where each category from the surface treatment roll-up case sample for a specified company is considered a separate data observation. The keyword-based features are computed only for each specified company and each specified category. By intuition and test, the best features we selected for the model are:

- number_of_occurrence: the number of keyword occurrences.
- p1_number_of_occurrence: the number of keyword occurrences on the website landing page of the company.
- %p_occurrence: the percentage of the subpages that mentioned one of the keywords in the given category among all subpages.
- %kwsTop10: first the percentage of keyword occurrence divided by the number of all word occurrences for each subpage is computed and the maximum percentage over the Top10 relevant subpages are selected.

The following table illustrates the data frame with the described features above.

Company_id	Category	number_of_occurence	p1_number_of_occurence	%p_occurencec	%kwsTop10	Relevant
1000986	painting	1	0	0.1667	0.0013	0
1000986	flooring	128	14	1.0000	0.0206	1
1000986	tiling	6	1	0.1667	0.0022	0
1000986	masonry	0	0	0.0000	0.0000	0
...	...	...	...	...	...	...

The column “Relevant“ indicates if the company belongs to the corresponding category. Note that a company may belong to multiple categories, and the column “Category“ is not used for model training. This data transformation obtains a larger data set to train a model. Due to data imbalance, the companies with rows whose number_of_occurrence is 0 are deleted, i.e., the websites do not mention any keywords in the corresponding category.

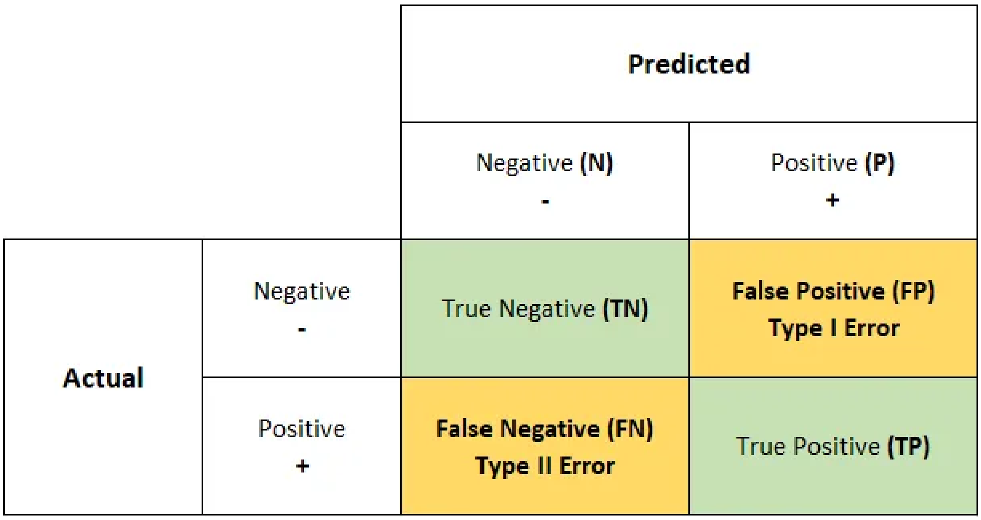
Below is an overview of the data distribution. In practice, when the websites for a specified category are scored, the algorithm filters out only the websites that mention at least one of the keywords in the corresponding category as our candidate websites. The proportion of the positive class is calculated as follows: Share of the positive class is the number of positive samples divided by the total number of samples that mentioned at least one of the keywords in all categories.
The positive class share of a category is the number of positive classes in that category divided by the number of sites that mentioned at least one keyword in that category.

Number of samples: 380
Number of positive class: 85
Share of positive class: 22.37%
Share of positive class of category flooring: 25.21%
Share of positive class of category painting: 26.32%
Share of positive class of category tiling: 13.16%
Share of positive class of category masonry: 19.23%

With this transformed data frame, a trained model is scalable to arbitrary potential categories which overcomes the main disadvantage of the first experimental SVM model.

3 Results

To compare different models, a measure of model performance is needed. A confusion matrix describes the 4 predicted and actual classification options for a binary classification problem [19].



		Predicted	
		Negative (N) -	Positive (P) +
Actual	Negative -	True Negative (TN)	False Positive (FP) Type I Error
	Positive +	False Negative (FN) Type II Error	True Positive (TP)

Binary Classification Problem (2×2 matrix)

There are 4 commonly used measures for model performance:

- **Accuracy:**
  The ratio of correctly predicted instances to the total number of instances evaluated:
  $$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{P + N}.$$
- **Precision**:
  The ability of the classifier not to label a negative sample as positive:
  $$\text{Precision} = \frac{\text{TP}}{TP + FP}.$$
- **Recall**:
  The ability of the classifier to find all the positive samples:
  $$\text{Recall} = \frac{\text{TP}}{TP + FN}.$$
- **F1**:
  The harmonic mean of precision and recall which lies in $[0, 1]$:
  $$\text{F1} = \frac{2}{\text{recall}^{-1} + \text{precision}^{-1}}.$$

For the investment target classification use case, the ability to detect all positive samples and not to label a negative sample as positive are both important, so it is decided to use the F1 score for model comparison to have a balanced score between precision and recall. During hyperparameter tuning, in addition to the model hyperparameters, a probability threshold is also considered as a hyperparameter for Decision Tree and Random Forest since the Support Vector Machine doesn't automatically provide probability scores. To reduce the impact of robustness, the data frame is partitioned with different random states, then the model selects the "best" hyperparameters so that the mean of the F1 score reaches its maximum, if tale, then the "best" hyperparameters with minimum standard deviation are selected.

3.1 Model performance

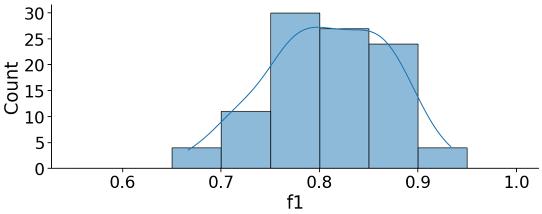
Performance matrix

Different data splits can affect the model's performance, so all model candidates are trained and tested over multiple data splits. The following table shows the average scores of the candidate models over 100 different data splits. Additionally, the first experimental SVM model provided by FSN is used as baseline model.

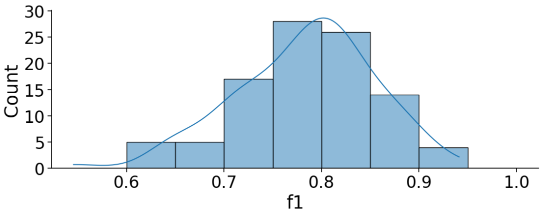
	Accuracy	Precision	Recall	F1
Baseline SVM	82.97%	78.15%	85.14%	80.85%
SVM	89.94%	76.48%	81.12%	78.29%
DT	90.91%	79.78%	81.06%	80.02%
RF	91.36%	81.64%	80.29%	80.60%

Robustness

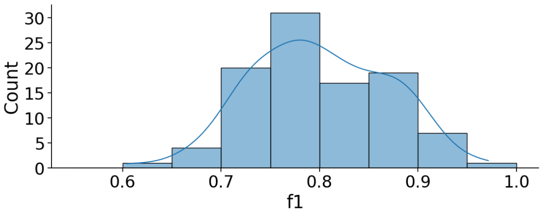
Besides the averaged scores, the robustness of models is also important. The following graphics show the robustness of the F1-Score for each model, aligned with their standard deviation σ:



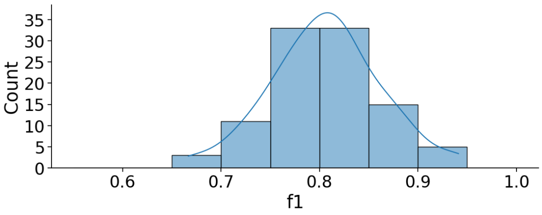
(a) Baseline SVM has $\sigma = 0.0603$



(b) SVM has $\sigma = 0.0705$



(c) DT has $\sigma = 0.0688$



(d) RF has $\sigma = 0.0553$

Since the initial established SVM by FSN is not scalable to different subsectors or business descriptions, the other three models are considered. The Random Forest demonstrates a better averaged F1-Score with better robustness. Thus, we decided to choose the random forest as our final model to use. The robustness of other performance scores are plotted in the appendix (6.2.

3.2 Impact of web scraping improvements

The improvements conducted on the web scraper from the base experimental algorithm are tested on the Decision Tree Model. According to the performance of the F1-Score, the improvements on the established scraper enhance performance by 4.5%.

	Accuracy	Precision	Recall	F1
Established web scraper	89.47%	76.07%	78.26%	76.57%
Improved web scraper	90.91%	79.78%	81.06%	80.02%

3.3 Impact of manually created keywords vs GPT4 generated keywords

During the sourcing process of potential targets, instead of curating a time-consuming list of keywords themselves, the user of the system will select from the keywords generated by GPT4. The performance of the Random Forest model selected from section 2.3 is compared with the manually created keyword list provided by FSN and the automatically generated keywords by GPT4.

	Accuracy	Precision	Recall	F1
Manually created keywords	91.36%	81.64%	80.29%	80.60%
GPT4 keywords	79.72%	71.43%	59.7%	65.04%

As expected, the manually curated keywords perform better since they are iteratively improved by manual website review as well as extensive trial and error with input from investment professionals and subsector experts. Although the F1 score achieved with GPT-generated keywords alone is not as high as with manually written keywords, an F1 score of 65% is already a sign of a reasonable classification.

3.4 Generalization

Since the approach's main goal is to have a model able to perform in multiple subsectors, a test of generalization ability is also necessary. Thus, the best-performing candidate models are trained on the main dataset from the **painting** and **flooring** categories and then tested on the remaining dataset from the **tiling** and **masonry** categories. The same performance characteristics as in the scoring section are used to compare the generalizability of the models. Since the performance table in section 3.1 shows that Decision Tree and Random Forest have similar performance on the F1-score, both models are examined for their ability to generalize.

Performance matrix of the Decision Tree and Random Forest trained on **painting** and **flooring** dataset and generalization tested on **tiling** and **masonry** categories.

	Accuracy	Precision	Recall	F1
DT	90%	68%	90%	67%
RF	91%	76%	65%	70%

The Random Forest demonstrates better performance on the F1-Score, i.e. a better generalization ability.

4 Prototype

To simplify user interaction with the GPT keyword generation system, a prototype is designed and tested. Two systems are developed. The first is a lightweight command line interaction that lets the user quickly interact with the GPT model and directly see the results. The second system is a web application with a frontend, a backend, and a database that stores the information and provides the ability to start multiple conversations and store the history of the conversations, and resume them later. Both systems use the logic represented in 4. The web application is containerized using Docker[7]. Docker lets users spin up the containers directly without additional configurations. It avoids the problem of inconsistency between members of the development team. The docker images will have predefined bootstrap actions using a consistent operation system, making the whole project portable. In our web application, we used three docker containers:

1. **Frontend container**: The frontend is developed using React[8], which is an open-source JavaScript library.

2. **Backend container**: The backend is based on Python and the FastAPI library[9] in particular. The FastAPI library allows us to create a RESTful API to communicate with the frontend. The backend communicates with the database container using the Motor library.

3. **Database container**: The database used in this project is MongoDB[10], a NoSQL database. A MongoDB NoSQL database is used for storing JSON-like data, making the web application scalable and developer-friendly.

5 Conclusions

In conclusion, Random Forest outperforms regarding the averaged performance of the F1-Score, its robustness, and the generalization ability. It has a comparable averaged performance of the F1-Score compared to the first established SVM baseline model and is also more robust. It is important to highlight that the Random Forest model that is introduced above provides a possible solution to the most important shortage of the basic SVM model, namely the inability to scale and generalize to other subsectors. With Random Forests, it is possible to categorize websites from any category provided by the user. Based on the analysis of several performance metrics where the Random Forest outperforms the other machine learning algorithms, the RF is considered to be better suited to the task of automating the steps in identifying potential investment targets.
The modifications implemented throughout the methodology, in particular the improvements to the web scraper using Breadth-First-Search and accessing and retrieving larger amounts and higher quality data, significantly increase the performance of the models. The approach of generating keywords with large language models such as GPT4 shows

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

[8] https://react.dev

[9] https://fastapi.tiangolo.com

[10] https://www.mongodb.com

great potential. Although manually curated and iteratively improved keyword lists still outperform automatically generated ones. Models using these keywords still show reasonable classification. Thus, the keyword list generated by GPT4 can serve as a starting point for users interested in analyzing completely new sectors where no knowledge has been acquired.

6 Outlook

While the results already indicate a reasonable classification, there are still some potential ideas for further improvement and interesting options for expansion into other private equity-specific tasks.

Generalization Ability: As seen from section 3.3, the keywords generated by GPT4 are not as good as the manual keywords. Currently, each keyword from a specified category has the same contribution to the features like “number_of_occurrence“ which is counter to the intuition. For example, for the category **flooring**, the keyword **flooring** should have a larger weight compared to the keyword **vinyl**. Motivated by this, one idea to improve the quality of the generated keywords is to embed the GPT4 generated keywords and the corresponding category itself by a suitable large language model and then apply cosine similarity to give a weight to each keyword and use the weighted number of occurrence instead the absolute one to train the model.

Keyword Generation using Mask Filling: This project integrates LLMs for topic modeling to extract potential keywords and as an agent to generate keywords on a feedback loop with the user. The generation of keywords could be improved further by utilizing the mask-filling capabilities of LLMs. Instead of prompting the LLM to return a list of keywords, it can also fill in blanks in a sentence, providing additional and more precise context. A sentence from a website containing a keyword can be modified to mask the keyword and let the LLM predict the mask several times to get additional keywords that would fit in the same context. For example, the task ”We specialize in flooring made out of [MASK].” could return a list like ”hardwood”, ”laminate”, and ”vinyl”.

Keyword Extraction using Word Embeddings: A different approach to get synonyms and keywords from the website corpus is to use word embeddings to suggest potential keywords with similar semantic features to an input keyword from the user. Given an input keyword and pre-computed word embeddings for each word, a heuristic to determine similar keywords is to compare the word embeddings using cosine similarity. If the cosine similarity of two word embeddings is high, the words describe similar content and are used in similar contexts in the corpus. This way, synonyms could be discovered directly in the corpus.

Application in other processes within the Private Equity cycle The ability to identify and accurately categorize information extracted from multiple websites based on manually curated or automatically generated keywords can be extended to other areas within the private equity business cycle. As a result of discussions with FSN and investment professionals, several opportunities have been identified that typically require significant manual effort and are time-consuming. These could be supported by the methodologies discussed in the report. The frameworks discussed in the report could potentially assist in identifying specific customers or suppliers within a particular subsector. The algorithms can also be used to identify other key themes within companies, such as companies focused on delivering positive impact through their solutions/offerings or highly socially and environmentally conscious companies.

The methodology in this report presents a promising approach to tackle the challenges of deal sourcing in the private equity industry. The conducted experiments have successfully improved the ability to generalize this approach and explored the capabilities of LLMs to assist with keyword generation. Our outlook and ideas for further improvements build a good starting point for future research and improvements.

References

[1] B Espen Eckbo, Gordon M Phillips, and Morten Sorensen. *Handbook of the Economics of Corporate Finance: Private Equity and Entrepreneurial Finance.* Elsevier, 2023.

[2] T. Schneeweis, Raj Gupta, and E. Szado. “The Benefits of Private Equity”. In: *Mutual Funds* (2008).

[3] Reiner Braun and Ingo Stoff. “The Cost of Private Equity Investing and the Impact of Dry Powder”. In: *The Journal of Private Equity* 19 (Feb. 2016), pp. 22–33. DOI: 10.3905/jpe.2016.19.2.022.

[4] David Teten and C. Farmer. “Where Are the Deals? Private Equity and Venture Capital Funds’ Best Practices in Sourcing New Investments”. In: *The Journal of Private Equity* 14 (2010), pp. 32–52. DOI: 10.3905/JPE.2010.14.1.032.

[5] Hugh MacArthur et al. *A Two-Pronged Approach to Sourcing More Private Equity Deals.* https://www.bain.com/insights/a-two-pronged-approach-to-sourcing-more-pe-deals-forbes/. Accessed: 2023-12-18. 2017.

[6] David Teten and Chris Farmer. “Where Are the Deals? Private Equity and Venture Capital Funds Best Practices in Sourcing New Investments”. In: *The Journal of Private Equity* 14.1 (2010), pp. 32–52. ISSN: 10965572, 21688508. URL: http://www.jstor.org/stable/43504295 (visited on 12/18/2023).

[7] *List of NACE codes.* https://ec.europa.eu/competition/mergers/cases/index/nace_all.html. Accessed: 18.12.2023.

[8] FSN Capital Partners. *About FSN Capital.* https://www.fsncapital.com/en/about/. Established in 1999, FSN Capital Partners is a leading Northern European private equity firm and advisor to the FSN Capital Funds, with 4 billion EUR under management and offices in Oslo, Stockholm, Copenhagen, and Munich. FSN Capital Funds make control investments in growth-oriented Northern European companies. 2023.

[9] Maarten Grootendorst. “BERTopic: Neural topic modeling with a class-based TF-IDF procedure”. In: *arXiv pre-print server* (2022). DOI: 10.48550/arXiv.2203.05794. URL: https://arxiv.org/abs/2203.05794.

[10] D. M. Blei, A. Y. Ng, and M. I. Jordan. “Latent Dirichlet allocation”. In: *Journal of Machine Learning Research* 3.4-5 (2003). Blei, DM Ng, AY Jordan, MI 18th International Conference on Machine Learning Jun 28-jul 01, 2001 Williamstown, ma Jordan, Michael I/C-5253-2013; Lobo, Diele/I-9106-2012, pp. 993–1022. ISSN: 1532-4435.

[11] Dimo Angelov. “Top2Vec: Distributed Representations of Topics”. In: *arXiv pre-print server* (2020). DOI: Nonearxiv:2008.09470. URL: https://arxiv.org/abs/2008.09470.

[12] Roman Egger and Joanne Yu. “A Topic Modeling Comparison Between LDA, NMF, Top2Vec, and BERTopic to Demystify Twitter Posts”. In: *Frontiers in Sociology* 7 (2022). ISSN: 2297-7775. URL: https://www.frontiersin.org/articles/10.3389/fsoc.2022.886498.

[13] Muriël de Groot, Mohammad Aliannejadi, and Marcel. "Experiments on Generalizability of BERTopic on Multi-Domain Short Text". In: *arXiv pre-print server* (2022). DOI: https://doi.org/10.48550/arXiv.2212.08459. URL: https://arxiv.org/abs/2212.08459v1.

[14] Erich Schubert. "Stop using the elbow criterion for k-means and how to choose the number of clusters instead". In: *ACM SIGKDD Explorations Newsletter* 25.1 (2023), pp. 36–42. ISSN: 1931-0145. DOI: 10.1145/3606274.3606278. URL: https://dx.doi.org/10.1145/3606274.3606278.

[15] Nils Reimers and Iryna Gurevych. "Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks". In: *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, Nov. 2019. URL: http://arxiv.org/abs/1908.10084.

[16] *Support vector machine*. URL: https://en.wikipedia.org/wiki/Support_vector_machine.

[17] Shai Shalev-Shwartz and Shai Ben-David. "Decision Trees". In: *Understanding Machine Learning: From Theory to Algorithms*. Cambridge University Press, 2014, pp. 212–218.

[18] Yanli Liu, Yourong Wang, and Jian Zhang. "New Machine Learning Algorithm: Random Forest". In: *Information Computing and Applications*. Ed. by Baoxiang Liu, Maode Ma, and Jincai Chang. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, pp. 246–252. ISBN: 978-3-642-34062-8.

[19] Anuganti Suresh. *What is a confusion matrix?* 2020. URL: https://medium.com/analytics-vidhya/what-is-a-confusion-matrix-d1c0f8feda5.

Appendix

6.1 Keyword Extraction

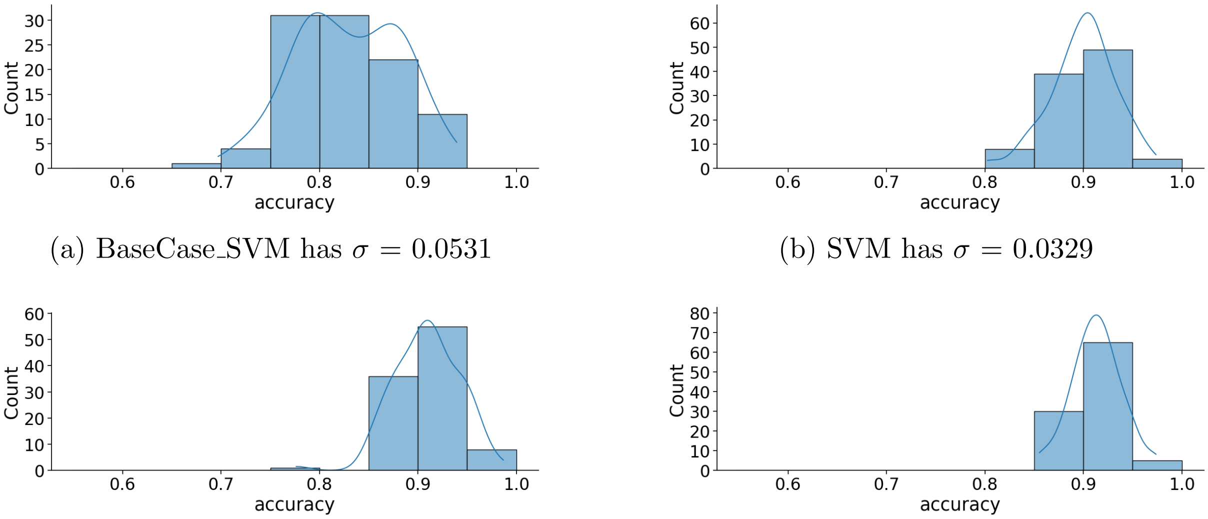
Flooring	Masonry	Painting	Tiling
Parkett	Beton	Farbe	Fliesen
Bodenbeläge	Bauweise	Fassade	Bad
verlegen	Grundstück	Malerarbeiten	Verlegen
Boden	Mörtel	Techniken	Oberfläche
Böden	Ziegel	Atmosphäre	Ausstattung
Fliesen	Mauerwerk	Oberflächen	Mörtel
Holz	Stein	Putz	Fugen
Laminat	Fassade	Wände	Keramik
Treppe	Dach	Decken	Naturstein
Treppen	Bauvorhaben	Fassaden	Wandfliesen

Table 3: Keywords extracted using Neural Topic Modeling and GPT-4

6.2 Performance Measurements

The following plots show the robustness of the accuracy, precision, and recall for each model separately, aligned with their standard deviation σ.

6.2.1 Accuracy



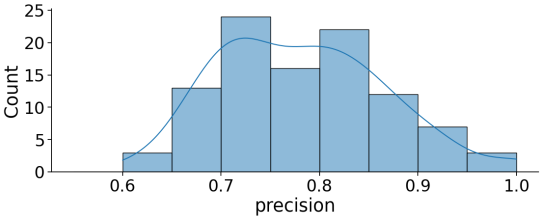
(a) BaseCase_SVM has $\sigma = 0.0531$

(b) SVM has $\sigma = 0.0329$

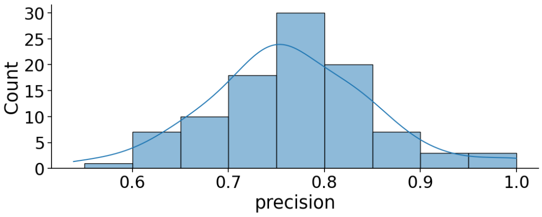
(c) DT has $\sigma = 0.0334$

(d) RF has $\sigma = 0.0251$

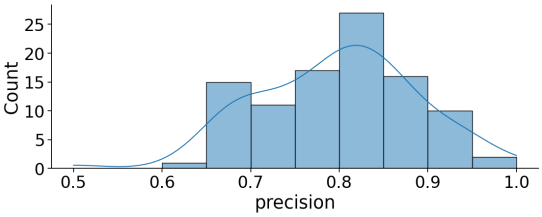
6.2.2 Precision



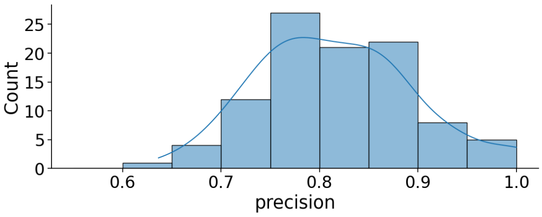
(a) BaseCase_SVM has $\sigma = 0.0839$



(b) SVM has $\sigma = 0.0874$

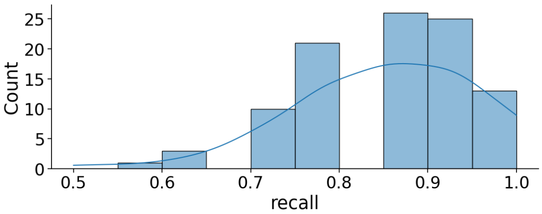


(c) DT has $\sigma = 0.0899$

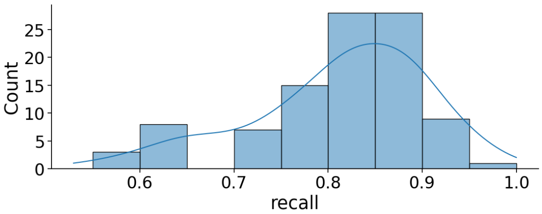


(d) RF has $\sigma = 0.0788$

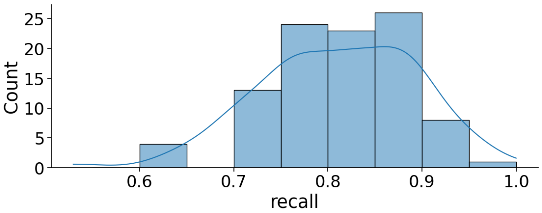
6.2.3 Recall



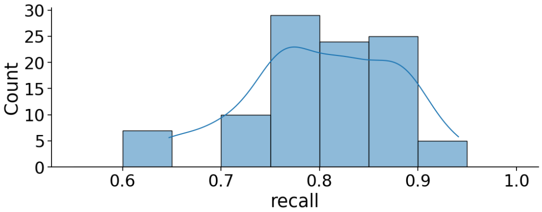
(a) BaseCase_SVM has $\sigma = 0.1024$



(b) SVM has $\sigma = 0.0942$



(c) DT has $\sigma = 0.0830$



(d) RF has $\sigma = 0.0755$