A Hybrid Approach for Mining the Organizational Structure from University Websites

Arman Arzani[a], Theodor Josef Vogl[b], Marcus Handte[c] and Pedro José Marrón[d]
University of Duisburg-Essen, Essen, Germany

Keywords: Innovation Management, Data Mining, University Structure Extraction, Web Page Classification.

Abstract: To support innovation coaches in scouting activities such as discovering expertise, trends inside a university and finding potential innovators, we designed INSE, an innovation search engine which automates the data gathering and analysis processes. The primary goal of INSE is to provide comprehensive system support across all stages of innovation scouting, reducing the need for manual data collection and aggregation. To provide innovation coaches with the necessary information on individuals, INSE must first establish the structure of the organization. This includes identifying the associated staff and researchers in order to assess their academic activities. While this could in theory be done manually, this task is error-prone and virtually impossible to do for large organizations. In this paper, we propose a generic organization mining approach that combines a rule-based algorithm, LLMs and finetuned sequence-to-sequence classifier on university websites, independent of web technologies, content management systems or website layout. We implement the approach and evaluate the results against four different universities, namely Duisburg-Essen, Münster, Dortmund, and Wuppertal. The evaluation indicate that our approach is generic and enables the identification of university aggregators pages with F1 score of above 85% and landing pages of entities with F1 scores of 100% for faculties, above 78% for institutes and chairs.

1 INTRODUCTION

Innovation coaches in a university are professionals who support researchers and staff in transforming academic ideas into practical innovations by guiding them through processes like commercialization, collaboration, and funding acquisition. Their roles include scouting for emerging trends and fostering innovation and knowledge transfer. Accordingly, the coaches engage in systematic scouting and screening activities to discover expertise and trends within the university in order to find innovators who have the potential to start their own startups. As part of a funded project, we developed INSE (INnovation Search Engine) to support the innovation coaches in their scouting activities by automating the data gathering and analysis processes (Arzani et al., 2023). Its primary task is to provide comprehensive system support across all stages of innovation scouting, reducing the need for manual data collection and aggregation. By integrating data from multiple data sources, INSE aims to offer a central platform where innovation coaches can access and analyze relevant information from academic staff members, such as their affiliation, research projects, reports, patents, and scientific papers. Although there are multiple ways to assess academic activities, INSE adopts a structured approach by first mapping the organization and its affiliated researchers. This not only helps contextualize academic contributions within a university but also enables meaningful comparisons across institutions for analyzing research activities.

To provide an overview of staff and researchers, some universities offer staff directories or databases that can be crawled or integrated in INSE. However, each portal and its connectors are different from one university to another, so INSE has to adapt the data collection to each university separately. A ubiquitous source of information on staff and their organizational affiliation is the university's public websites. The websites not only outline the structure of the university but also provide additional information on news, projects, lectures, and research areas of individuals in their institutes or chairs.

In many cases, the online presence of universities is spread across various websites and multiple administrative domains inside departments or institutes. Websites of high-level entities such as ma-

[a] https://orcid.org/0009-0000-1304-9012
[b] https://orcid.org/0009-0009-2494-5336
[c] https://orcid.org/0000-0003-4054-1306
[d] https://orcid.org/0000-0001-7233-2547

Arzani, A., Vogl, T. J., Handte, M. and Marrón, P. J.
A Hybrid Approach for Mining the Organizational Structure from University Websites.
DOI: 10.5220/0013658600004000

In *Proceedings of the 17th International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management (IC3K 2025) - Volume 1: KDIR*, pages 188-199

jor institutes or faculties are often operated by the university's central IT department, whereas other institutes and chairs further down the hierarchy are managed by independent staff of institutes or postgraduates at chairs that hold and maintain a subdomain of the university pointed to their website. Technically, some entities inside the university may even utilize JavaScript frontend platforms to develop their own website, while others use various Content Management Systems (CMS) to maintain their web presence. From a design perspective, one institute might list their researchers on their landing page, another may have a link to the same page in their navigation menu. Furthermore, the languages of these websites may be inconsistent (some pages are in German, some in English, and some may even mix languages), and there is a variety of terms for the different entities (such as chair, division, group, and discipline), which are not used uniformly. These lead to inconsistencies in the visual layout and the content of the websites of organizational entities.

While modern search engines can locate relevant web pages based on keywords, they fail to provide insights into the underlying organizational structure, important to the innovation coaches. As one finds the desired organizational department using a Google search, information such as the affiliation to the upper-level institutes or the relationship to the faculty might be missing. Therefore, an effective approach is necessary in acquiring a comprehensive understanding of organizational structure to provide INSE with the gathered data for aggregation and analysis in support of scouting and screening research activities of individuals as well as their organizations. Solving this challenge is also relevant for innovation coaches who are required to compare one university or its entities to another for emerging trends. For instance, determining how a computer science department of a specific university ranks against another one, requires systematic gathering of data regarding their publications as well as their funded projects. This is a practical application not only for universities but also for other large organizations that maintain decentralized online repositories.

To extract the structure of the university, Large Language Models (LLMs) can be employed as single-shot or few-shot classifiers for the classification of websites (Sava, 2024). However, this approach presents two main challenges for university domains. First, LLMs on large scale data may not be time- or cost-efficient—especially when using API-based commercial models or open-source alternatives. Second, the likelihood of false positives is high due to the difficulty of identifying actual university entities among a large amount of irrelevant data.

To address this challenge, this paper presents a hybrid approach, combining LLMs and a rule-based algorithm capable of extracting organizational structures from university websites. By treating university websites as directed acyclic graphs, our approach traverses the graph and identify chairs, institutes, and faculties. Initially, the algorithm follows certain entity navigation mechanisms to identify the organizational structure and the overview pages (aggregators), which contain a list of entities. In doing so, the algorithm visits the websites of the target university and locates the entities based on concepts defined by the user. Subsequently, we utilize LLMs to identify two sets of entities based on the content of websites. First, we use a zero-shot LLM inference to identify faculties. Finally, we train a sequence-to-sequence (seq2seq) language model that is effectively able to classify institutes and chairs.

We compare the results of the algorithm for the organizational structure of four universities for which we gathered the ground truth, namely Duisburg-Essen, Münster, Dortmund, and Wuppertal. The contributions of the paper are as follows:

- Conceptualizing and developing of a generic organization mining algorithm for the identification of aggregator pages
- Evaluation of the algorithm for the four universities with F1 scores of over 85%.
- Comparison of landing page identification for institutes and chairs using state-of-the-art GPT4o-mini vs. open-source Llama 3.3, DistilBert, and Flan T5.
- Evaluation of the Llama 3.3 for the four universities, with F1 scores of 100% for faculties and fine-tuned seq2seq Flan T5 with an F1 score of 78% for institutes and chairs, outperforming the previous approaches.

The remainder of the paper is organized as follows: Section 2 discusses the related work; Section 3 describes the approach, including our entity navigation mechanisms, as well as our use of LLMs. Section 4 presents the implementation and outlines the resulting processing pipeline, and Section 5 discusses the evaluation results for the four universities. Finally, in Section 6 we conclude the paper with a summary and an outlook.

2 RELATED WORK

Several research efforts focus on topic-based organizational structures and semantic units within and

across websites. Authors of (Kumar et al., 2006) address the problem of hierarchical topic segmentation by segmenting a website's URL tree into topically uniform topic regions and aggregating page-level topic labels to identify sub-sites dedicated to specific topics. In a related direction, (Li et al., 2000) introduces the notion of "logical domains" within a website, which are semantically cohesive units that span across the physical directory structure. They propose a rule-based technique utilizing link structure, URL paths, page metadata, and external citations to identify entry pages and boundaries of these logical domains. Authors of (Sun and Lim, 2003; Sun and Lim, 2006) further extend this idea by proposing a "Web unit," defined as a set of semantically related web pages forming a concept instance. Their iterative web unit mining method involves an iterative process of identifying these web units, considering website structure and connectivity, and classifying them into predefined categories. Another similar work is website topic hierarchy (Yang and Liu, 2009), which models a website's link structure as a weighted directed graph and adapts graph algorithms to generate topic hierarchies. The authors' approach focuses on distinguishing between aggregation links (topic to subtopic) and shortcut links using various features and learning algorithms to estimate edge weights.

Some authors depend on work artifacts such as email or work logs to generate the organizational hierarchy (Ni et al., 2011; Nurek and Michalski, 2020; Abdelakfi et al., 2021). For instance, (Abdelakfi et al., 2021) introduces an NLP-based agent-oriented framework that mines organizational structures from email logs by analyzing email content and classifying interactions into workflow organizational topics. While the authors use unsupervised learning and a neural network, the work of (Nurek and Michalski, 2020) explores the combined machine learning with social network analysis to reveal organizational structures.

Furthermore, recent advancements in deep learning facilitate text-based classification tasks, including the categorization of web content (Bartík, 2010; Aich et al., 2019; Minaee et al., 2021). For example, authors of (Aich et al., 2019) propose a convolutional neural network model for web text classification, emphasizing its simplicity and high accuracy compared to other deep learning approaches like RNNs and LSTMs. Their study focuses on tuning hyperparameters and the sequence of word vectors to achieve optimal performance on web-based texts across different topics. Also, in a related but distinct approach, (Sava, 2024) investigates the use of self-hosted open-source LLMs like Llama, Mistral, and Gemma for text-based website classification.

Our work is well aligned with (Rehm, 2006) in the organizational mining, specifically within academic institutions, where the author analyzes the topology and characteristics of different types of university web pages in the experiments. However, this work identifies distinct hypertext genres and models by utilizing a semantical ontology and hypertext in conjunction to classify university web pages. In our case, we do not explicitly employ ontologies; instead, we leverage pretrained LLMs, which inherently embed ontological and semantic structures acquired during training. Furthermore, unlike (Rehm, 2006), we do not manually identify or analyze the characteristics of university landing pages, as this task is instead inferred through the LLM's prior knowledge and representational capacity. Other related studies rely on sitemaps, topic hierarchies, or URL structures to classify or segment websites. In contrast, our approach departs from these structural methods. In our experience with German university websites, sitemaps are often unavailable or do not accurately reflect the organizational hierarchy. Furthermore, lower-level units such as chairs or institutes may operate under separate domains and apply different content management systems, making structural URL-based approaches unreliable.

In our work, we focus solely on analyzing the text content of individual websites. To extract organizational entities, we combine LLMs with a rule-based mining algorithm. Our use of LLMs encompasses both zero-shot prompting and fine-tuned models, while our algorithm follows unique navigation mechanisms specific to academic websites, an aspect not addressed in prior work.

3 APPROACH

In the following section, we first present the rationale and an overview of the approach. Next, we provide details on the identification of the aggregator pages that encompass a list of entities. Subsequently, we explain our method of identifying the landing pages of university entities.

3.1 Rationale and Overview

In this work, our objective is to extract the organizational structure of a target university based on its website. This structure reflects the hierarchical relationships between various internal entities and units within the institution. To this end, we focus on identifying and extracting key organizational entities that commonly define a university's structure specifically, **faculties, institutes, and chairs**.

(a) Overview website of faculties.

(b) Overview website of chairs.

(c) Landing page of Biology faculty.

(d) Landing page of Didactics of Biology chair.

Figure 1: Examples of websites.

Most universities are organized hierarchically, where faculties serve as the primary organizational units. Within each faculty, there are various institutes or departments, which are further subdivided into chairs or research groups. This hierarchical structure is often represented on the university's website, where entities are grouped and linked in a way that reflects their real-world relationships.

Our approach is premised on the assumption that the organizational structure of a university can be inferred from its website. Specifically, we assume that the way entities are linked and grouped on the website reflects their actual hierarchical relationships. This assumption is based on the observation that universities commonly design their websites to facilitate easy navigation, with overview pages that aggregate and group entities of the same type. For example, a university might have a dedicated page listing all its faculties, with each faculty page further linking to its respective institutes and chairs.

Figure 1 depicts an example from the University of Duisburg-Essen, showcasing its faculties 1a and the overview website of the chairs of the Biology faculty 1b. Furthermore, Figures 1c and 1d show the landing pages of the faculty of Biology, as well as the landing page for the chair of Didactics in this faculty. To simplify the discussion and avoid confusion, in the following, we refer to entity overview pages as **aggregators** and the websites of entities (faculties, institutes, and chairs) as **landing pages**.

LLMs are capable of classifying websites based on their content. To explore the potential of LLMs in automating large-scale website classification, we conduct a simple experiment to assess their viability in identifying aggregators within a university's web pages. We first generate a dataset with 3000 pages from Münster University that contains all 237 aggregators. Then, we pass the content of each page to a self-hosted Llama 3.3 70B instance to perform a zero-shot classification. On a single PC with two Nvidia A6000s, the classification takes about 8 hours and results in an F1 score of 29%. Given that this experiment only accounts for approx. 1% of the web pages of Münster University, the computation time is too high to be applicable in practice, and the classification accuracy is clearly far from being satisfactory.

To improve the classification performance and to reduce the computation time, we propose a hybrid approach to classification that combines generic entity navigation mechanisms (to identify a relevant subset of pages) with content-based classification that employs LLMs and fine-tuned sequence-to-sequence classifiers. Our approach starts by visiting the university's homepage, which serves as the entry point. From there, we follow outgoing links to explore specific pages within the website, similar to how a person would search for a specific entity. The exploration process involves identifying links that lead to pages representing aggregators for faculties, institutes, and chairs. By analyzing and targeting the structure and

the content of these pages, we can then reconstruct the organizational structure of the university.

3.2 Aggregator Identification

Next, we describe the approach for identifying aggregator pages in university web pages. We begin by explaining the concepts and the entity navigation mechanisms and then present the pseudocode of the algorithm that encompasses the latter.

3.2.1 Concept

Due to the decentralized nature of university websites, some entities may use synonyms or multiple terms that may refer to the same entity type. For example, while a university might use the term "divisions" for their faculties, another university may just use the term faculties.

The confusion between lower-level entity types (institutes, chairs) suffers from even more chaos in our experience. Most universities in Germany have interchangeable terms or abbreviations for entities such as chairs, for instance calling them workgroup, WG, group, professor, scientific field, or research area. Another factor that leads to entity confusion is the translated entity synonyms in multiple languages. For instance, German universities use the word "Lehrstuhl" or "AG" (short for Arbeitsgruppe) as chair, or an abbreviation of it. As for institutes, referring to a form of lecture, the term "Seminar" is also used at German universities to designate individual organizational institutes within the faculty. For example, there is the "Historical Seminar," which refers to the institute that encompasses the history-related academic programs and its staff at a university.

Therefore, a set of categorized concepts needs to be laid down to ensure the consistency of entities regardless of a university's country of origin, language, and the underlying layout structure. To accomplish this, we define a generic list of grouped concepts as an input to our approach for three entity types, namely *faculties, institutes, and chairs*. A concept is the point of truth that matches an entity's name in singular, plural, or the abbreviation form in any defined language. An example of a concept definition is given below:

```
Concept: ('language'= model.Language.EN,
          'singular' = 'Department',
          'plural' = 'Departments',
          'type' = model.GROUP.CHAIR)
```

The plural and singular forms of a concept (e.g., faculties 1a) are important for answering whether a web page is an overview page. The singular forms of the concepts are depicted in Figure 2 as the purple-black circles.

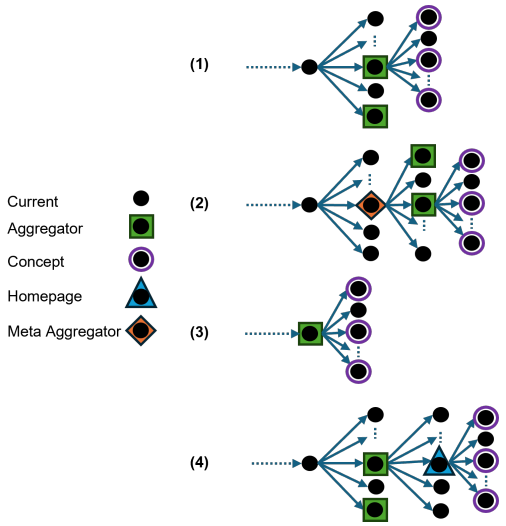

Figure 2: Entity navigation mechanisms in a website.

3.2.2 Aggregators and Navigation Mechanisms

In order to identify aggregator pages, we perform word matching with the header content of a page. The header contents include HTML tags such as *'title', 'h1', 'h2', 'h3', 'h4', 'h5', 'h6', 'th', 'strong'*. An aggregator describes an overview page of similar entities for a concept that meets two criteria. First, the header content of the current visited page or the header content of the outgoing hyperlinks of the current page should contain the plural form of a defined concept (e.g., faculties, chairs, groups). For example, the page contains the following,

```
<a><h1>Fakultäten der Universität</h1></a>
```

Where the h1 header tag includes the plural concept of faculty in German. After finding the plural concepts in the header content of the current page, the page's hyperlinks and their inner text are extracted and stored, and the page itself is chosen as an *aggregator candidate*. For the second criterion, using the stored hyperlinks and hyperlink texts, the chosen candidate at least references one direct outgoing link that contains the singular form of the defined concept with their hyperlink text in their header content. If both criteria are met, the web page is registered as an aggregator page that most likely contains an overview of the similar concepts.

If the current content or content of the outgoing links of the current page includes a plural concept, the page is addressed as the **base aggregator**. This first case is true for universities that provide an overview of entities on their landing page; for example, an institute that lists the associated departments on the same page as the institute's start URL. The second case is more common, as most universities have the tendency to differentiate between the start URL and the overview of their underlying entities by providing a

Algorithm 1: Mining algorithm for aggregator identification.

```
Initialize remaining_pages, output_hierarchy;
remaining_pages ← [{ 'url': START_URL, 'concepts': CONCEPTS }];
all_concepts ← remaining_pages.concepts;
output_hierarchy ← [] ;
while remaining_pages is not empty do
    page ← remaining_pages.pop();
    url ← page.url;
    remaining_concepts ← page.concepts;
    candidates ← FindAggregatorCandidateURLs(url, remaining_concepts) ;        /* find candidates */
    aggregators ← [];
    foreach candidate_url in candidates do
        aggregators.add(FindBaseAggregators(candidate_url)) ;              /* Check base aggregator */
        aggregators.add(FindIndirectAggregators(candidate_url)) ;      /* Check indirect aggregator */
        aggregators.add(FindMetaAggregators(candidate_url)) ;              /* Check meta aggregator */
    foreach aggregator in aggregators do
        remaining_concepts ← GetRemainingConcepts(aggregator.concepts);
        foreach child in aggregator.children do
            remaining_pages.add([{ 'url': child.url, 'concepts': remaining_concepts }]);
        output_hierarchy.add(aggregator) ;                                 /* Output of the hierarchy */
```

link to the aggregator in the navigation menu or in the content. The black circles with the revolving green squares, in Figure 2, represent the aggregator pages. Examples of two cases of base aggregator pages are depicted in (1) and (3) and in Figure 2.

Furthermore, a special case that we handle is where each of the links of the entities on an aggregator do not directly point to the concept's landing page. Some universities offer the main URL after the aggregator of their faculties. In this case, the homepage link delivers the actual concept page that was reviewed in the aggregator page. The latter is considered as the **indirect aggregator** and is depicted in case number (4) in Figure2. Another special case is **meta aggregators**, as aggregators that are reachable through other aggregators. This is depicted in case (2) as the orange-shaped diamond in Figure 2. This is sometimes the case, as the targeted aggregators are accessible in the second level. In such instances, the plural concepts point to at least an outgoing page that includes another plural concept with their hyperlink text in their header content. An example of this is a page linking to research areas where each research area, in turn, links to a list of chairs.

3.2.3 Algorithm

Based on the concepts and the entity navigation mechanisms, the simplified pseudocode of the algorithm is described in 1. The algorithm starts by accepting the *remaining_pages*, which contain the starting URL (home URL of the target university) and a list of generic concepts for main-level entities (faculties, institutes, and chairs). In the next step, the algorithm extracts the aggregator candidates. The algorithm first checks for base aggregators, then indirect aggregators, and finally, meta aggregators for every candidate.

The algorithm performs a depth-first search by finding aggregator pages for higher-level concepts (faculties) before diving into the underlying concept levels (institutes and then chairs). Effectively, this builds the organigram, or the structure of the organization; therefore, in each level, the remaining concepts, as well as the URLs, should be noted in the *remaining_pages*. The algorithm stops as soon as all the potential child concepts of each aggregator are visited. In the final step, the algorithm returns the organizational structure in the output list, which entails the labeled aggregator pages of the university. As a result, each identified aggregator is marked with a label: *faculty, institute, or chair*, and is stored with their corresponding outgoing pages that point to potential landing pages as well as other, unrelated pages.

3.3 Landing Page Identification

After identifying aggregators using the algorithm, in this subsection, we describe the LLM approach for the identification of entity landing pages based on their text content. Typically, faculty landing pages are easier to identify on university websites because they are higher-level administrative entities with distinct, well-structured web presences, often featuring standardized naming conventions. As discussed in the overview, this is not the case for the institutes

or chairs, as they tend to have more varied and less formalized web structures. Hence, in the following, we differentiate between high-level entities (faculties) and low-level entities (institutes and chairs).

3.3.1 Faculty Landing Pages

Using the outgoing links of the identified faculty aggregator(s), the goal here is to traverse the content of the links and identify the faculty landing pages among non-faculty ones. Leveraging the background knowledge of LLMs, the model recognizes patterns in text and assesses elements such as faculty names, titles, research areas, and departmental affiliations. In this case, we utilize Llama 3.3 70B open-source as a zero-shot classifier, which consists of a prompt and the target content. Thus, the content of each outgoing link of the faculty aggregator is passed onto a zero-shot LLM prompt. The LLM responds with yes or no, which is mapped into a binary output. The format of the prompt is specified below:

```
Prompt: 'Yes or no, does the following web-page
represent the welcome page of a faculty of the
{target_university}? \n\n Page:\n{page.text}'
```

The LLM results are the true labeled links that are classified as the faculty landing pages of the target university.

3.3.2 Institute and Chair Landing Pages

To this end, we systematically visit the outgoing links of the detected low-level aggregators to identify the landing pages of the institutes and chairs. To achieve this, we utilize a fine-tuned LLM, namely Flan T5 Large, to classify and distinguish institutes/chairs from others. FLAN-T5 (Fine-tuned Language Net T5) is an enhanced version of Google's T5 model, fine-tuned on a diverse set of instruction-following tasks to improve zero-shot and few-shot learning capabilities (Longpre et al., 2023). It follows a sequence-to-sequence (seq2seq) architecture that takes an input sequence (e.g., a prompt) and generates an output sequence (e.g., a response), making it effective for classification tasks. We gather ground-truth data from four universities, based on which we produce a training dataset. The dataset involves text content of the websites with their corresponding labels, such as "institute/chair" or "other". Before training, the preprocessing step of tokenization of content is needed, where the input IDs are the numerical token representations of the input text which are converted using the model's vocabulary. Also, the tokenizer generates attention masks, which tell the model which tokens should be attended to (1) and which should be ignored (0). Finally, after adding the padding tokens to standardize the input length, the training is validated with the F1-score evaluation metric. The output of this step generates a list of institutes and chairs of the target university.

4 IMPLEMENTATION

The implementation is carried out in Python and contains three major components: *the spider agent*, *the algorithm,* and *the LLM-based classification*. The pseudocode of the algorithm is implemented as outlined in 1 and accepts the starting URL along with the defined concepts in an array of JSON objects as input.

The spider agent component consists of the web crawler and the preprocessing logic. To do this, we use the *Selenium framework*, to perform web crawling and handle dynamic websites. The framework acts as a bridge between Selenium Web Driver and the Chrome browser by enabling us to perform tasks like opening web pages, clicking buttons, and scraping data. Selenium also provides us with an interface to inject JavaScript (JS) code into a rendered page. As the content of a website is downloaded, the interface enables us to execute custom JS code that iterates through the hyperlinks of each page. Furthermore, in order to enhance the algorithm over quick iterations, we perform caching and content retrieval using SQLite and *SQLAlchemy*. Also, the data modelling is performed using *Pydantic*.

An overview of the implementation is depicted in Figure 3. Initially, the algorithm visits the corresponding aggregator pages and their concepts in a depth-first manner. For each visited page that fulfills the algorithm's defined concept requirements, the spider agent passes the URL to the web crawler. The web crawler renders the visited URL in a headless Chrome browser and downloads the content by extracting the hyperlinks and their texts. Subsequently, the browser also stores the extracted header content tags for the URL and each outgoing link that fulfills a concept, as discussed in 3.2.2. The spider analyzes the header content to detect the language of the page, since a URL might exist in multiple languages. Next, the preprocessing logic normalizes the extracted links and their texts. As a result, the extracted outgoing links undergo link normalization, where the relative URLs are transformed into the absolute URL paths. Moreover, the text of every hyperlink is normalized by removing hyphenation within the link texts. The spider also handles URL redirection. This is typically implemented using HTTP status codes like 301 (permanent redirect) or 302 (temporary redirect) and is used to guide users and search engines to the correct resource when a URL has changed or been relocated.

The extracted information of each visited page

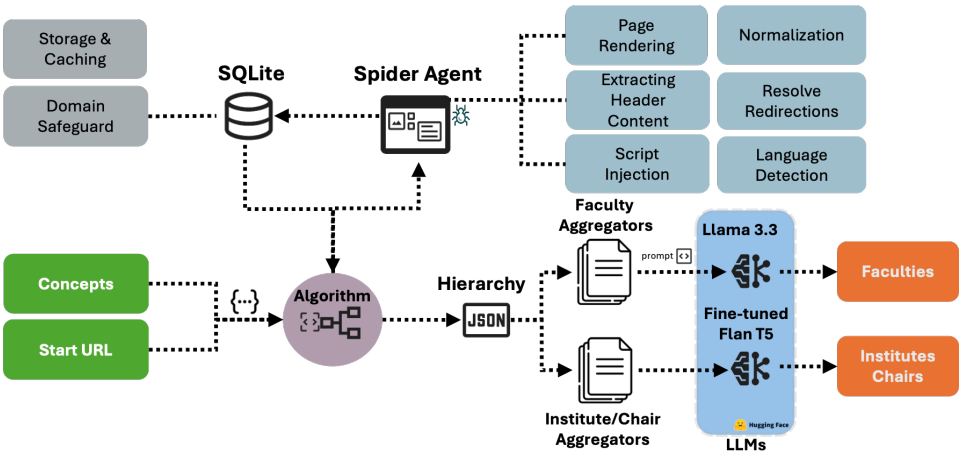

Figure 3: Overview of the implementation.

is then passed through the defined data model onto the SQLite database for storage and retrieval. The database is also responsible for the allowed URL domain. The given start URL in the algorithm determines the allowed domain; therefore, only the domain and its subdomains are considered in the structure. This is to avoid following outgoing hyperlinks to random domain addresses that do not contribute to the organizational structure. The results of the algorithm (the aggregators) are saved in a JSON file, which represents the structure of the target university.

Table 1: Ground truth of the universities.

University	Concept Type	Count
Duisburg-Essen (UDE)	Aggregator	205
	Faculty	12
	Institute/Chair	352
Münster (MUEN)	Aggregator	237
	Faculty	15
	Institute/Chair	409
Dortmund (TUD)	Aggregator	158
	Faculty	17
	Institute/Chair	209
Wuppertal (WUPP)	Aggregator	150
	Faculty	9
	Institute/Chair	117

In the last step, the resulting low-level and high-level aggregators and their outgoing links are passed onto the LLM component. The hardware for hosting the LLM as well as LLM fine-tuning includes two instances of Nvidia A6000 48GB GPUs running on Linux. For the identification of faculties, we deploy Llama 3.3 70B in a Docker container using Huggingface's text-inference API 3.1 (Wolf et al., 2019), which conforms to OpenAI's API specification. As discussed in the concept, the positive responses of the zero-shot prompts include the faculty landing pages. Furthermore, to identify the landing pages of institutes/chairs, we utilize the Flan-T5 large model, a sequence-to-sequence (encoder-decoder) architecture, which is fine-tuned using the same set of hardware and a training dataset. Consequently, the hyperparameters of the Flan-T5 large are optimized, and training is performed for four epochs on 75% of the data. The results are then evaluated on the remaining 25%. The model's output consists of the classified landing pages of institutes/chairs.

The output of the LLM component results in the entity landing pages of the target university. This concludes the steps taken in the implementation.

5 EVALUATION

Given the implementation described above, we evaluate our approach in this section. In the following, we first discuss data collection. Next, we present the evaluation results of the algorithm. Finally, we discuss the results of the LLM approach.

5.1 Data Collection

To evaluate our approach and assess its generalizability, we select four different universities for testing. *Duisburg-Essen*, *Münster*, *Technical University Dortmund,* and *Wuppertal*. For each, we conduct a structured manual review of their official websites to identify and extract organizational entities. This process involved systematically navigating through each university's publicly available web pages and start by visiting any given aggregator page, selecting a faculty, and finding its underlying institutes and chairs.

Table 2: Algorithm performance metrics for the four universities.

University	Algorithm Type	Evaluation Metric		
		P	R	F1
UDE	Base	0.73	0.83	0.77
	Indirect	0.74	0.84	0.79
	Meta	0.84	0.92	0.88
	All	0.84	0.93	**0.88**
MUEN	Base	0.90	0.83	0.86
	Indirect	0.90	0.87	0.89
	Meta	0.87	0.86	0.86
	All	0.88	0.90	**0.89**
TUD	Base	0.81	0.76	0.79
	Indirect	0.81	0.76	0.79
	Meta	0.75	0.85	0.80
	All	0.75	0.85	**0.80**
WUPP	Base	0.78	0.86	0.82
	Indirect	0.77	0.89	0.82
	Meta	0.77	0.92	0.84
	All	0.76	0.94	**0.84**

Following up from the overview page, we visit each entity's website and extract the page name based on visible header tags, along with the URL. These page names and URLs are then recorded for further processing.

Table 1 shows the numbers of each type of entity in each university. The abbreviations of each university are shown in parentheses. All universities have several faculties in common, such as Medicine, Physics, Chemistry, Biology, and Economics. However, they differ in certain areas: for example, Münster has dedicated faculties for Geosciences and Catholic or Evangelical Studies, whereas these fields are categorized as institutes within the Human Sciences faculty at Duisburg-Essen.

5.2 Results

So far we have shown the implementation of our approach, as well as the gathered ground truth of the four universities. Here, we initially examine the algorithm's capability in identifying aggregators of the organizational entities in the four introduced universities. Then, we evaluate the results of the LLM in detecting landing pages of entities.

5.2.1 Aggregator Identification

First, we define the input parameters to the mining algorithm. Specifically, we provide the start URLs of the four universities, namely *https://www.uni-due.de/* for Duisburg-Essen, *https://www.uni-muenster.de/*, for Münster, *https://www.tu-dortmund.de/* for Dortmund, and *https://www.uni-wuppertal.de/* for Wuppertal. We also define the generic core concepts for each concept category. For all universities, concepts for faculties, institutes, and chairs such as "scientific field, institute, research center, department, group" and their German equivalents are added to their concepts, as described in 3.2.1.

After passing the input parameters, we execute the algorithm. As explained in 1, after finding the aggregator candidates, i.e., potential overview pages, the algorithm performs the discussed entity navigation mechanisms (base, indirect, and meta aggregators) before writing the aggregator results in a JSON file. We instrument the algorithm so that we can focus on each particular step of the algorithm in order to individually measure their contribution to the overall performance. In this analysis, we consider each of the algorithm's navigation mechanisms separately and calculate the overall performance metrics, namely precision (P), recall (R) and the harmonic mean (F1). Table 2 shows the results of the algorithm for each university based on the given step, in the same order as they appear in the algorithm.

The base navigation step acts as the baseline for aggregator identification since it reflects the simplest case, where the concept entities are linked directly by an overview page. The performance metrics of each step should be compared to the base navigation step.

The base navigation step scores the lowest in UDE 77% and the highest in Münster 86%. Also, Wuppertal and Dortmund score close to or over 80%. This is due to the fact that some UDE aggregator pages are not directly accessible by clicking on the aggregators and are positioned behind other aggregator pages. This can be verified by switching to the meta navigation step, where the links in the aggregator page are reached through other found aggregators. This indicates a rise in the F1 score to 88% in UDE and a slight rise in Dortmund and Wuppertal overall score and recall in Münster. The lower F1 score in the baseline can be expected, since the missing main aggregators lead to propagating the error down the hierarchy. In other words, if a faculty aggregator is not found, the underlying entities will not be explored. The algorithm's indirect step only works in UDE and Münster with F1 scores of 79% and 89% compared to the baseline. This is due to the fact that the main URL of some of their institute or faculty aggregators is not directly accessible by clicking on the aggregator pages, and each entity is positioned behind the home page link, which then leads to the main URL.

Finally, we activate all the algorithm steps in order to evaluate the performance metrics of the final output. The algorithm achieves over 80% for all cases,

with the highest for Münster and the lowest for Dortmund. A reason for the lower scores of Dortmund is the lack of consistent aggregator pages. In some instances, the chairs or the institutes of a faculty are listed in landing pages, with the names of persons or abstract research areas serving as the actual underlying concept entities. Since our algorithm performs word matching for the given concepts, lack of singular concept names explains the missing entities.

From a runtime perspective, the algorithm demonstrates clear efficiency: while the initial run takes around 1 to 2 hours per university, subsequent executions are reduced to just 15 minutes through the use of database caching. This makes the approach considerably faster than the previously discussed LLM-based method while also yielding a significantly higher F1 score, improving from 29% with the LLM to 85% with the rule-based algorithm.

In conclusion, the final F1 scores indicate that aggregator pages representing organizational entities can be effectively identified. Based on the results on four universities, our algorithm is capable of detecting aggregator pages of the organizational structure of a university with an average F1 score of 85%.

5.2.2 Landing Page Identification

In this section, we evaluate the results of the identification of faculties and institutes/chairs. Based on the aggregator type (faculty, institute/chair) and ground truth, we produce a dataset for evaluation. For the faculties, the outgoing links that correspond to the actual faculties of each aggregator are labeled true, while other links that do not exist in the ground truth are labeled false. This is considered for the outgoing links of institute/chair aggregators as well.

For the faculties, we proceed as discussed in implementation. Our experiments using zero-shot prompts show that prompt 3.3.1 is capable of identifying faculty landing pages with F1 score of 100% for all four universities. This shows that the pretrained Llama 3.3 can easily differentiate between faculty pages and other unrelated pages, such as contact, project, or teaching pages.

For the institutes/chairs, we investigate the performance of several LLMs under two experimental settings: (1) we train the models separately on the data from each university, and (2) we train the models on the combined dataset that includes data from all four universities. In both cases, we measure performance using precision (P), recall (R), and F1-score (micro F1), since the data is imbalanced (1087 trues, 9904 falses). For both cases, the data is split 25-75% and shuffled before training.

In the first case, Table 3 shows the results for models evaluated independently for each university (UDE, MUEN, TUD, WUPP). This setup allows us to see how well each model performs when tailored specifically to a single university's data, helping us understand university-specific behavior and characteristics.

We also compare LLMs such as Llama 3.3 (Touvron et al., 2023) as zero-shot and few-shot to state-of-the-art GPT-4o-mini (Isogai et al., 2024). For the prompts, we use the same format of 3.3.1 but with institute or chair instead of faculty. Also, in few-shot prompts, we add 3 content examples for chairs, institutes, or non-entities. We also fine-tune Distilbert and two variations of Flan-T5, as discussed in the implementation and concept. While DistilBERT is not considered a large language model due to its smaller size and architecture, we include it in our comparison as a baseline for classification tasks (Adoma et al., 2020).

Table 3: Performance metrics for institutes/chairs of each university.

University	Model	Evaluation Metric		
		P	R	F1
UDE	Llama 3.3 (ZS)	0.61	0.68	**0.60**
	Llama 3.3 (FS)	0.58	0.65	0.49
	GPT-4o-mini (FS)	0.63	0.58	**0.60**
	DistilBERT	0.74	0.67	0.70
	Flan-T5 Base	0.83	0.68	0.72
	Flan-T5 Large	0.81	0.76	**0.79**
MUEN	Llama 3.3 (ZS)	0.61	0.71	**0.61**
	Llama 3.3 (FS)	0.56	0.63	0.47
	GPT-4o-mini (FS)	0.58	0.56	**0.57**
	DistilBERT	0.71	0.66	0.68
	Flan-T5 Base	0.86	0.75	0.79
	Flan-T5 Large	0.85	0.82	**0.83**
TUD	Llama 3.3 (ZS)	0.57	0.69	**0.59**
	Llama 3.3 (FS)	0.52	0.56	0.52
	GPT-4o-mini (FS)	0.63	0.55	**0.57**
	DistilBERT	0.73	0.57	0.60
	Flan-T5 Base	0.74	0.61	0.65
	Flan-T5 Large	0.76	0.69	**0.72**
WUPP	Llama 3.3 (ZS)	0.57	0.73	**0.58**
	Llama 3.3 (FS)	0.54	0.64	0.48
	GPT-4o-mini (FS)	0.58	0.54	**0.55**
	DistilBERT	0.75	0.58	0.62
	Flan-T5 Base	0.79	0.58	0.62
	Flan-T5 Large	0.84	0.63	**0.69**

Across all universities, Flan-T5 Large emerges as the top performer, achieving the highest F1-scores for UDE 79%, MUEN 83%, TUD 72%, and WUPP 69%. This indicates that larger encoder-decoder models can effectively learn from and adapt to domain-specific patterns given the adequate training data. In contrast, zero-shot models like Llama 3.3 (ZS), which have not been fine-tuned on the specific data, perform more

modestly. It is also noticeable that while few-shot Llama (FS) performs lower than GPT-4omini, the ZS Llama outperforms GPT by a few percent. This is surprising given the complexity and the context size of the GPT model in comparison to Llama 3.3.

Models like DistilBERT and Flan-T5 Base also show strong and consistent results across four universities, with F1-scores ranging between 60% and 79%. Interestingly, despite being a newer architecture, GPT-4o-mini (FS) performs worse than Flan-T5, suggesting that encoder-decoder models might be more naturally suited for classification tasks of this nature. We also note some differences between universities. For example, MUEN appears to be easier to model, with generally higher F1 scores across all models. In contrast, WUPP and TUD yield slightly lower scores, possibly due to differences in the number of institutes/chairs in the dataset.

In the second case, we explore the models' ability to generalize across universities; to this end, we train the models on the combined dataset of all four universities. Table 4 presents these results.

Once again, Flan-T5 Large leads in performance, achieving an F1-score of 78%, followed closely by Flan-T5 Base at 75%. These results are consistent with the findings per university, reaffirming the strength and adaptability of the Flan-T5 architecture across diverse institutional data. DistilBERT also performs well in this setting, achieving an F1-score of 65% — notable given its smaller size and simpler encoder-only design. Among the decoder-only models, Llama 3.3 (ZS) achieves the best performance in its group with an F1 of 61%, outperforming its few-shot variant, which reaches 52 %. This suggests that in some cases, zero-shot decoding may perform better than fine-tuning due to the confusion caused by the given examples in the few-shot prompt.

Table 4: Performance metrics for institutes/chairs of four universities together.

Architecture	Model	Evaluation Metric		
		P	R	F1
Decoder-Only	Llama 3.3 (ZS)	0.60	0.71	**0.61**
	Llama 3.3 (FS)	0.56	0.66	0.52
	GPT-4o-mini (FS)	0.61	0.57	0.58
Encoder-Only	DistilBERT	0.73	0.61	**0.65**
Encoder-Decoder	Flan-T5 Base	0.76	0.73	0.75
	Flan-T5 Large	0.80	0.76	**0.78**

When comparing the two experimental setups, we find that models trained on individual university data generally perform better when evaluated within their specific domain. For example, Flan-T5 Large achieves up to 83% F1 on MUEN in the individual university setting, compared to 78% when trained on the combined dataset. This suggests that domain-specific fine-tuning can offer performance benefits by capturing localized patterns more precisely. This is explainable, since some universities tend to use their own specific terms for the lower-level entities.

Furthermore, the results indicate that the models trained on the combined dataset perform more consistently across all four universities, making them a presumably better choice when building a general-purpose model, especially in scenarios where domain ground-truth labels are not (entirely) available. The relatively small drop in performance for the combined Flan-T5 model further highlights its generalization capabilities. One possible reason why Seq2Seq models like Flan-T5 models outperform decoder-only models like Llama or GPT is the architectural alignment. These models are explicitly designed for tasks that involve mapping inputs to outputs, making them more effective for classification. In contrast, decoder-only models are optimized for open-ended language generation, which can introduce bias and reduce precision in structured prediction tasks.

Nevertheless, the final F1 scores indicate that entities from all four universities are extracted, with an average score of 100% for high-level entities (faculties) and 78% for low-level entities (institutes/chairs). This suggests a consistent structure of landing pages across universities for both entity categories.

6 CONCLUSIONS

To support the innovation coaches in scouting activities such as discovering expertise inside the university and finding potential innovators, we designed INSE, an innovation search engine that automates data gathering and analysis processes. The primary goal of INSE is to provide comprehensive system support across all stages of innovation scouting, reducing the need for manual data collection and aggregation. However, to provide the coaches with the necessary information on university trends and individuals, INSE must first establish the structure of the organization, as well as their affiliated researchers, in order to assess their academic activities.

In this paper, we proposed a generic organization mining approach that combines a rule-based algorithm, LLMs, and a fine-tuned sequence-to-sequence classifier. We initially described entity navigation mechanisms and implemented the solution in the algorithm, which outperforms a zero-shot LLM classifier in time and F1 score. Subsequently, we spec-

ified the LLM and the sequence-to-sequence classifier approach for the identification of landing pages of high/low-level entities. Finally, we evaluated our results against four different universities, namely Duisburg-Essen, Münster, Dortmund, and Wuppertal. The results indicate that the implemented approach works across universities, capable of identifying university structure and its entities with average F1 scores of 85% for the aggregator pages, 100% for faculties, and 78% for institutes/chairs.

As part of INSE, we are working to build a graphical user interface around our approach with the objective of supporting the innovation coaches of our university in scouting and screening tasks. For future work, we are planning to investigate a visual-based approach for the aggregator and landing page identification via convolutional neural networks.

ACKNOWLEDGEMENTS

This work has been funded by GUIDE REGIO, which aims to improve the ability of the science support center of the University of Duisburg-Essen in the identification, qualification, and incubation of innovation potentials.

REFERENCES

Abdelakfi, M., Mbarek, N., and Bouzguenda, L. (2021). Mining organizational structures from email logs: an nlp based approach. *Procedia Computer Science*, 192:348–356.

Adoma, A. F., Henry, N.-M., and Chen, W. (2020). Comparative analyses of bert, roberta, distilbert, and xlnet for text-based emotion recognition. In *2020 17th international computer conference on wavelet active media technology and information processing (ICCWAMTIP)*, pages 117–121. IEEE.

Aich, S., Chakraborty, S., and Kim, H.-C. (2019). Convolutional neural network-based model for web-based text classification. *International Journal of Electrical & Computer Engineering (2088-8708)*, 9(6).

Arzani, A., Handte, M., and Marrón, P. J. (2023). Challenges in implementing a university-based innovation search engine. In *KDIR*, pages 477–486.

Bartík, V. (2010). Text-based web page classification with use of visual information. In *2010 International Conference on Advances in Social Networks Analysis and Mining*, pages 416–420. IEEE.

Isogai, S., Ogata, S., Kashiwa, Y., Yazawa, S., Okano, K., Okubo, T., and Washizaki, H. (2024). Toward extracting learning pattern: A comparative study of gpt-4o-mini and bert models in predicting cvss base vectors. In *2024 IEEE 35th International Symposium on Software Reliability Engineering Workshops (ISSREW)*, pages 127–134. IEEE.

Kumar, R., Punera, K., and Tomkins, A. (2006). Hierarchical topic segmentation of websites. In *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 257–266.

Li, W.-S., Kolak, O., Vu, Q., and Takano, H. (2000). Defining logical domains in a web site. In *Proceedings of the eleventh ACM on Hypertext and hypermedia*, pages 123–132.

Longpre, S., Hou, L., Vu, T., Webson, A., Chung, H. W., Tay, Y., Zhou, D., Le, Q. V., Zoph, B., Wei, J., et al. (2023). The flan collection: Designing data and methods for effective instruction tuning. In *International Conference on Machine Learning*, pages 22631–22648. PMLR.

Minaee, S., Kalchbrenner, N., Cambria, E., Nikzad, N., Chenaghlu, M., and Gao, J. (2021). Deep learning–based text classification: a comprehensive review. *ACM computing surveys (CSUR)*, 54(3):1–40.

Ni, Z., Wang, S., and Li, H. (2011). Mining organizational structure from workflow logs. In *Proceeding of the International Conference on e-Education, Entertainment and e-Management*, pages 222–225. IEEE.

Nurek, M. and Michalski, R. (2020). Combining machine learning and social network analysis to reveal the organizational structures. *Applied Sciences*, 10(5):1699.

Rehm, G. (2006). *Hypertextsorten: Definition, Struktur, Klassifikation*. PhD thesis, Universitätsbibliothek Giessen.

Sava, D. (2024). Text-based classification of websites using self-hosted large language models: An accuracy and efficiency analysis. B.S. thesis, University of Twente.

Sun, A. and Lim, E.-P. (2003). Web unit mining: finding and classifying subgraphs of web pages. In *Proceedings of the twelfth international conference on Information and knowledge management*, pages 108–115.

Sun, A. and Lim, E.-P. (2006). Web unit-based mining of homepage relationships. *Journal of the American Society for Information Science and Technology*, 57(3):394–407.

Touvron, H., Lavril, T., Izacard, G., Martinet, X., Lachaux, M.-A., Lacroix, T., Rozière, B., Goyal, N., Hambro, E., Azhar, F., et al. (2023). Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*.

Wolf, T., Debut, L., Sanh, V., Chaumond, J., Delangue, C., Moi, A., Cistac, P., Rault, T., Louf, R., Funtowicz, M., et al. (2019). Huggingface's transformers: State-of-the-art natural language processing. *arXiv preprint arXiv:1910.03771*.

Yang, C. C. and Liu, N. (2009). Web site topic-hierarchy generation based on link structure. *Journal of the American Society for Information Science and Technology*, 60(3):495–508.