Social Computing Research Group
TUM School of Computation, Information and Technology
Technical University of Munich

Leveraging LLM Pipelines to Generate Product Comparison Data

Khalil Chikhaoui, Jaeyeop Chung, Umut Ekin Gezer
advisor: Dr. Gerhard Johann Hagerer

March 17, 2025

Abstract — Market Analysis is a crucial step in driving business success. However, it is a resource-intensive process that requires a vast amount of time, expertise, and human effort. Large language models(LLMs) present a promising opportunity to automate such task, potentially reducing costs and improving efficiency. In this project work, we present a proof-of-concept for an LLM-driven market analysis, specifically tailored to generate the comprehensive insurance product comparisons across insurance companies in Germany. Furthermore, we explore systematic methods to measure and improve the quality of the generated product comparisons.

1 Introduction

Since the advent of Transformer architecture, Transformer based pre-trained llms, such as GPT, have revolutionized the field of natural language processing(NLP). Large language models have shown its capabilities to understand and generate natural languages in a various tasks (Chang et al. 2024). However, despite their groundbreaking capability, the output of s heavily depends on the in-context learning(ICL) and prompt engineering, resulting in incoherent and factually unsupported content (Wei et al. 2025). This phenomenon of hallucination poses challenges in ensuring the reliability of llm-generated market analysis.

Recent studies have explored various prompting techniques to enhance the reasoning ability of LLMs, such as Chain-of-Though(CoT) (Wei et al. 2022) prompting technique. However, evaluating the effectiveness of these prompt techniques remains challenging, as traditional NLP metrics often fall short in assessing open-ended text generation tasks (Liu et al. 2023). Consequently, a novel approach "LLM-as-a-Judge" has been proposed to address such challenge (Zheng et al. 2023).

1.1 Problem Statement

Despite the promising potential of LLMs to automate market analysis, challenges persist regarding the selection of metrics to measure the quality of generated product comparisons. Reliability remains a critical issue in high-stakes industries such as insurance, where misinformation can have significant consequences. In this context, "LLM-as-a-Judge" approach allows for a wide range of customized evaluation metrics, but its effectiveness depends on the calibration of the evaluation design. Therefore, a systematic approach is required to monitor and quantify the impact of the various prompting strategies on the quality of generated product comparisons.

RQ1. How do can a reproducible experiment be achieved?

RQ2. Which LLM-as-a-Judge evaluation metric is suitable to ensure reliable product comparisons?

RQ3. How can prompt fine-tuning enhance binary classification performance in classitiny whether webpage content ?

RQ4. How does prompt fine-tuning impact the efficiency of categorizing insurance products based on textual descriptions?

1.2 Outline

This report explores an experimental approach to building an LLM-based insurance product comparison system. We begin by introducing data collection and processing steps in Kedro, followed by applying various prompts to classify and generate product comparisons. Then, we analyze how various prompt strategies influence product comparison quality, leveraging LLM-as-a-Judge for evaluation. This novel approach holds a great potential to be extended to other domains where LLM-driven automation is desired.

2 Methodology

In this section, we introduce methods for data processing, LLM-driven insurance product classification, and LLM-as-a-Judge evaluation concepts.

2.1 Prompting Techniques

To construct our insurance product database using our pipeline, we employed the GPT-4o model in four key nodes: *2. Filtering Relevant Pages*, *4. Feature Extraction*, *5. Product Classification*, and *6. Product Comparison*.

In the *Filtering Relevant Pages* (see Section 3.1.2) node, we used GPT-4o to classify web pages as either insurance-related or unrelated (e.g., career or advertisement pages). The *Feature Extraction* (see Section 3.1.4) node employed prompting techniques to extract only relevant insurance product attributes from HTML-converted markdown files. In the *Product Classification* (see Section 3.1.5) node, we utilized GPT-4o to categorize insurance products based on standard insurance types, enabling better organization and comparison. Lastly, in the *Product Comparison* (see Section 3.1.6) node, we generated structured comparison tables by identifying common attributes and matching extracted product details.

Prompting techniques play a fundamental role in ensuring the accuracy, consistency, and comprehensiveness of outputs generated by LLMs. Without well-structured prompts and iterative refinement, the model might omit crucial details, introduce factual errors, or produce outputs in formats that are difficult to parse (Fagbohun and Dereventsov 2024; Zamfirescu 2023). By imposing constraints and guiding the model's behavior, we can steer it toward more reliable and domain-specific responses. These techniques are therefore crucial in real-world applications, such as insurance product comparisons, where misinformation can lead to costly consequences.

2.1.1 The Role of Prompting in LLMs

Prompting is a crucial technique in interacting with LLMs, as it determines the quality and relevance of model-generated responses. It serves as a bridge between human intent and computational capabilities (Fagbohun and Dereventsov 2024). While prompting may seem straightforward, it directly influences model behavior by adjusting billions of parameters, making it a key area of research in NLP (Zamfirescu 2023). Even for experts, prompt engineering often involves trial and error to optimize prompts for clarity, structure, and efficiency. Despite extensive research, the exact mechanisms by which different prompting strategies improve model responses remain uncertain.

2.1.2 Prompt Refinement Techniques

Beyond broad considerations of prompt design, our methodology also involved more granular improvements to ensure high-fidelity, grounded, and comprehensive outputs, especially for creating a reliable insurance product database. Although equally applicable to broader market analyses and product comparisons in other domains, these refined techniques helped us systematically improve the quality of LLM outputs:

Role Prompting We prepended instructions to the model to adopt a specific role (e.g., *"You are an insurance domain expert"*), thereby steering the style and tone of its responses. This technique leverages the model's capacity to adapt its knowledge to a specialized persona, ensuring more focused and context-appropriate results (Schulhoff 2025).

Iterative Prompt Refinement Prompt design was treated as an iterative process (Miguelañez 2025). After obtaining initial outputs, we reviewed them for factual accuracy, coverage of relevant details, and consistency. We then refined the prompt instructions (e.g., adding clarifications or constraints) to address observed shortcomings, repeating this process until the model's output met our quality benchmarks.

Structured Output Formatting Whenever feasible, we instructed the model to produce structured outputs (e.g., Markdown tables, JSON). This method facilitated post-processing and quality checks, ensuring that essential fields were consistently included (AI 2023b). It also minimized ambiguity by guiding the model to follow a predefined schema.

Constraint-Based Prompting We established explicit constraints, such as limiting the response to a specified number of words or mandating only content from given sources. By confining the model's generation to these rules, we reduced hallucinations and improved the groundedness of the output (AI 2023a). For example, in our product comparison node, we specifically instructed the model to avoid guessing missing data and to label unspecified attributes as *"Not available."*

Overall, these prompt refinement techniques proved integral in enhancing the reliability, clarity, and completeness of our LLM-generated insurance product database. While this pipeline was tailored for insurance market analysis, the same techniques can be generalized to broader tasks such as e-commerce product comparisons or healthcare service analyses. We thus view prompt engineering as an essential practice for effectively leveraging LLMs in real-world, data-centric applications (Miguelañez 2025; Schulhoff 2025).

2.2 Product Comparison Metric

Although various prompting techniques (Wei et al. 2025, 2022; Kojima et al. 2022; Mann et al. 2020; Soylu et al. 2024), as well as other techniques to utilize external resources (Qin et al. 2023), have been developed to enhance reasoning process of LLMs, the nature of pre-trained models conveying inherent biases imposes a limit on improving the performance of LLMs solely with prompt engineering. In recent research papers on automating prompt optimization, prompts have been seen as a hyperparameter to optimize through iterative fine-tuning and prompt optimization (Singhvi et al. 2023; Pryzant et al. 2023; Soylu et al. 2024). While our project manually searched for optimized prompts through trials and errors without fine-tuning the model, as with other machine learning techniques, selecting the right metric was a crucial step to determine the best prompt strategies and, consequently, to optimize the output quality of LLMs (Zheng et al. 2023).

2.2.1 LLM-as-a-Judge

In a broader sense, conventional NLP evaluation metrics can be categorized into lexical matching-based metrics(e.g., ROUGE, BLEU, METEOR, etc) and embedding based-metrics(e.g., BERTScore, etc). However, these metrics have shown limitations on evaluating an open-ended natural language generation(NLG) tasks, failing to capture deeper nuances (Gu et al. 2024; Hamimoune 2025).

As a solution to this, we adopted **LLM-as-a-Judge**, an emerging approach that combines the strengths of traditional NLP evaluation metrics with great adaptability across various tasks and domains. Given its growing adoption, most current LLM-as-a-Judge metrics build upon the work of Zheng et al. (2023), who proposed three key variations of the framework.

- **Pairwise Comparison**: An LLM judge is given a question along with two answers and is responsible for determining which answer is superior or if they are equally good.
- **Single Answer Grading**: An LLM judge is asked to directly assign a single score to a single query.
- **Reference-Guided Grading**: An LLM is given a reference data to which user's query will be compared, and assigns a score based on task-specific criteria.

2.2.2 Metric Selection

Measuring the quality of input data as well as the generated data in LLM applications is an essential task to explain the performance of an LLM model. However, there are not standard metrics to measure them. Following the notion that prompts are hyperparameters to optimize, the final output data can be considered as another training set in the process of prompt optimization. From this perspective, we considered evaluation metrics based on the order of importance for high-quality data suggested by Yu et al. (2024): reliability, relevance, and accuracy.

We identified two major LLM-as-a-Judge frameworks suited for our implementation scheme: **LangSmith** and **TruLens**. Both frameworks provided LLM tracing functionalities, user friendly APIs, and a wide range of metric choices from traditional NLP metrics to LLM-as-a-Judge metrics.

We ultimately chose TruLens for its demonstrated evaluation performance[1]. The following are the LLM-as-a-Judge metrics we selected from TruLens. These metrics are implemented using the *reference-guided grading approach*, where an LLM assigns a score by comparing the generated output to the reference data[2].

- **Groundedness Score** measures how well a generated product comparison is supported by source text data. Both the generated and source data are split into individual sentences, and an LLM evaluates whether each sentence has direct or indirect evidence from the source. After analyzing all sentences, the LLM assigns a final score from 0 to 1, indicating factual alignment with the source.
- **Comprehensiveness Score** measures how well the generated product comparison captures the essential content of the source text. It identifies key points from the source text and evaluates how

[1] Benchmarking LLM-as-a-Judge for the RAG Triad Metrics(SnowFlake)

[2] TruLens evaluation metric source code

completely they are represented in the generated output. An LLM then assesses the presence of each key point and assigns a final score from 0 to 1, indicating the degree to which the generated data reflects the original source. In our project, the groundedness score, also referred to as faithfulness, played an important role as *an indicator of hallucination* in the generated product comparisons.

- **GroundTruthAgreement Score** measures the similarity between the manually corrected and generat product comparisons(The human annotated data is often called *Golden Dataset*). The TruLens framework provides various similarity measurements including BERTScore, BLEU, ROUGE, and LLM-as-a-Judge. Due to the computational and time constraints, LLM-as-a-Judge was used where an LLM measures the similarity and assigns a final score from 0 to 1, indicating alignment with human judgements.

3 Data

A Kedro pipeline (Kedro 2019) is a modular, reproducible workflow for data processing, designed with best practices in software engineering. It provides a structured way to organize code, ensuring that data inputs, transformations, and outputs are clearly defined and easy to maintain. By separating concerns (e.g., data crawling, cleaning, model training) into dedicated nodes. Kedro ensures that each step of the process remains transparent and testable. This modular design also promotes collaboration among team members, as well as scalability, additional nodes or entire pipelines can be added or replaced without disrupting existing functionality.

Kedro-Viz: Pipeline Visualization

To enhance interpretability and manageability, we leveraged Kedro-Viz, an interactive visualization tool integrated within Kedro.
Kedro-Viz graphically illustrates the entire pipeline structure, clearly depicting each node and its interdependencies, as shown in Figure 1.

3.1 Kedro Pipeline Architecture

In this project, the *main pipeline* comprises the primary workflow that handles the entire data lifecycle, from initial crawling of insurance websites to final product comparison. The main pipeline encapsulates seven key nodes:

3.1.1 Data Crawling

Objective: Crawled up to 200 pages per company (ARAG, AXA, Barmenia, DKV, ERGO, Generali, HUK-COBURG, SDK).
Implementation: `crawl_companies_node` was used, leveraging BeautifulSoup (Richardson 2021) for parsing and extracting raw HTML, subsequently storing the data in the Kedro catalog. The primary crawling parameters included:

- **Maximum Depth:** 200 pages per domain
- **Parsing Tool:** BeautifulSoup
- **Link Extraction Strategy:** Follow all internal links within the company domain up to the specified depth
- **Deduplication:** Identical URLs or pages with duplicate content were tracked to avoid redundant processing

3.1.2 Filtering Relevant Pages

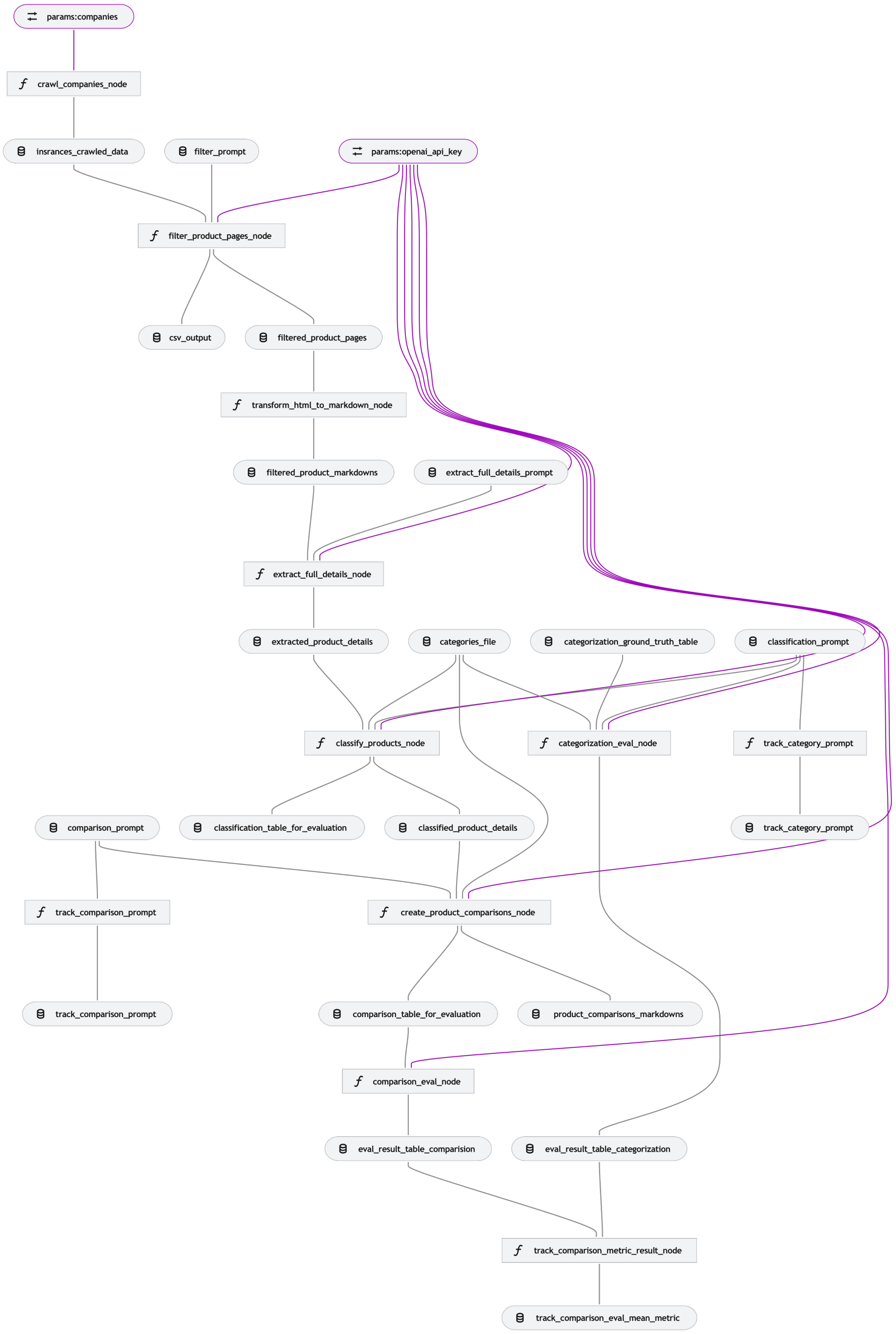
Objective: Discard irrelevant pages (e.g., career pages, advertisements) and retain only those containing insurance product information.
Implementation: `filter_product_pages_node` was employed, using LLM-based prompts to classify the pages as *relevant* vs. *irrelevant* based on URLs and file names. A subset of **1,600 pages** was manually labeled to establish a ground truth dataset.

3.1.3 HTML to Markdown Conversion

Objective: Transform HTML content into markdown for simpler text processing.
Implementation:
`transform_html_to_markdown_node` converted the filtered HTML pages into concise markdown files, ensuring a uniform input format for subsequent parsing.

3.1.4 Feature Extraction

Objective: Identify product names and their features (e.g., coverage limits, eligibility conditions, pricing information).
Implementation: `extract_full_details_node` utilized LLM-based methods to parse the markdown



params:companies
crawl_companies_node
insrances_crawled_data
filter_prompt
params:openai_api_key
filter_product_pages_node
csv_output
filtered_product_pages
transform_html_to_markdown_node
filtered_product_markdowns
extract_full_details_prompt
extract_full_details_node
extracted_product_details
categories_file
categorization_ground_truth_table
classification_prompt
classify_products_node
categorization_eval_node
track_category_prompt
comparison_prompt
classification_table_for_evaluation
classified_product_details
track_category_prompt
track_comparison_prompt
create_product_comparisons_node
track_comparison_prompt
comparison_table_for_evaluation
product_comparisons_markdowns
comparison_eval_node
eval_result_table_comparision
eval_result_table_categorization
track_comparison_metric_result_node
track_comparison_eval_mean_metric

Figure 1 Kedro pipeline visualization generated by Kedro-Viz showing the end-to-end workflow, from data crawling and filtering to product classification, comparison, and experiment tracking.

Feature	DKV	SDK	ARAG	Group	ERGO
Coverage Type	Vollversicherung	Vollversicherung	Vollversicherung	Vollversicherung	Krankenhaus-Zusatzversicherung
Premiums	Alters- und gesundheitsabhängig, Arbeitgeberzuschuss möglich	Stabile Beiträge, Rückstellungen im Alter, Arbeitgeberzuschuss	Beitragsrückerstattung bei Nichtnutzung	Nicht spezifiziert	Ab 19,80 € monatlich, zwei Tarifoptionen
Hospital Benefits	Behandlung im Krankenhaus, Auslandsschutz	Chefarztbehandlung, Einbettzimmer, bevorzugte Behandlungen	Chefarztbehandlung, Ein- oder Zweibettzimmer	Chefarztbehandlung, freie Krankenhauswahl, Rooming-in	Ein- oder Zweibettzimmer, Chefarztbehandlung, freie Krankenhauswahl
Dental Coverage	Nicht spezifiziert	100% Zahnbehandlung, bis zu 90% Zahnersatz	100% Zahnbehandlung, bis zu 90% Zahnersatz	Nicht spezifiziert	Nicht spezifiziert
Additional Benefits	Vorsorge- und Gesundheitsprogramme, Hilfsmittelservice	Heilpraktikerleistunger Facharzt-Terminservice, Medgate-App	GesundheitsApp, Beitragsentlastung im Alter	Alternative Heilmethoden, Schutzimpfungen	Ersatz-Krankenhaustagegeld bei Verzicht auf bestimmte Leistungen
Family Coverage	Separate Verträge für Kinder	Nicht spezifiziert	Nicht spezifiziert	Tarif clinic+ für Kinder ab 2,97 €	Unterbringung und Verpflegung eines Erziehungsberechtigten für Kinder bis 16 Jahre bei stationärer Behandlung
Customer Support	Kundenservice 0800 3746 444	Hotline 0800 221 0221	Expertenberatung telefonisch	Nicht spezifiziert	Kundenportal für Rechnungseinreichung
Unique Features	Marktführer im Bereich der privaten Zusatzversicherung	Flexibles Baukasten-System, 15 Tarifbausteine	MedExtra, MedBest, KomfortKlasse Tarife	Beihilfeschutz für verbeamtete Personen	Zwei Tarifoptionen: günstiger Einstieg oder stabiler Beitrag mit Alterungsrückstellung

Table 1 Comparison of private health insurance features across different providers.

and extract features, storing the outputs in a structured dataset. Extracted details include:

- Product names
- Coverage details
- Eligibility requirements
- Pricing information

3.1.5 Product Classification

Objective: Categorize each product into recognized insurance types (e.g., *Haftpflichtversicherung*, *Krankenversicherung*).

Implementation: `classify_products_node` matched extracted features against a reference taxonomy (Wikipedia's Versicherungsart list (Wikipedia Contributors 2025)) using an LLM-based model. Products were assigned to one of **30 predefined insurance categories**, with manual verification for ambiguous cases. An example of the structured product classification output is shown below:

```
[{
  "product group A": [
    {  "company": "ergo",
       "category": "Haftpflichtversicherung",
       "product_name": "Bauherren-
          haftphlichtversicherung",
       "products_text": ["Die ERGO ..."] },
    {  "company": "generali",
       "category": "Haftpflichtversicherung",
       "product_name": "gewerbliche-
         haftpflichtversicherung",
       "products_text": ["Die gewerbliche ..."]},
  {
   "product group B": [ ... ]
  }, ...
 }]
```

3.1.6 Product Comparison

Objective: Compare similar products across multiple companies by measuring semantic similarity, then generate tabular summaries highlighting differences in coverage, pricing, and notable features.

Implementation: Initially, all product names and textual details are standardized in a common JSON format. The text is split into manageable segments

Feature	ERGO	R+V	ARAG	DKV
Coverage Duration	Up to 8 weeks per trip	First 45 days per trip, extendable to 730 days	First 42 days per trip	Varies by plan, e.g., 31 days in Europe
Premiums	From 9,90 € annually	10,80 €/year (single), 22,80 €/year (family)	8,00 €/year (single), 24,00 €/year (family)	From 9,90 € annually (RDE plan)
Medical Coverage	100% for necessary treatments, incl. dental	100% for treatments, dental, transport, and more	100% for treatments, incl. dental	Comprehensive, incl. long-term stays
COVID-19 Coverage	Included	Included	Included	Included
Family Coverage	Children covered until 25 years	Partner and up to 8 children	Family coverage if living in the same household	Not specified
Emergency Services	24/7 emergency hotline, TeleClinic app	24/7 info and emergency hotline	24/7 emergency hotline	Customer service 7-19h daily
Repatriation	Covered if medically necessary	Covered if medically necessary	Covered if medically necessary	Covered
Additional Features	Digital consultation via TeleClinic app	Automatic extension if return is medically impossible	No deductible for private insured	Travel cancellation insurance available
Age Limit for Coverage	Children up to 25 years	Up to 64 years for standard rate	No specific age limit mentioned	Not specified
Special Conditions	Tax-deductible under certain conditions	Contact via phone or email for more info	Online application available	Insurance for guests from abroad

Table 2 Comparison of private travel insurance features across different providers, including ERGO, R+V, ARAG, and DKV, focusing on coverage duration, medical benefits, repatriation, and special conditions.

(e.g., 512 tokens), and each segment is embedded using the *sentence-transformers/multi-qa-distilbert-cos-v1* model (Reimers and Gurevych 2019). By averaging the segment-level embeddings, each product receives a consolidated vector representation.

Subsequently, these embeddings form a similarity matrix by cosine-similarity comparison (Pedregosa et al. 2011). This matrix feeds into a graph-construction process where products are treated as nodes, and edges appear only between products whose similarity surpasses a defined threshold (0.75) and originate from different companies. A maximal-clique-finding algorithm (Hagberg et al. 2008) then clusters products deemed highly similar across providers.

For each resulting cluster, a Large Language Model (GPT-4) (Achiam et al. 2023) generates standardized Markdown comparison tables as shown in Table 1 and Table 2. These tables highlight the similarities and differences in coverage terms, pricing, and features. By aggregating these structured outputs, the `create_product_comparisons_node` provides a high-level overview of similar insurance products side by side, facilitating an efficient, comparative evaluation.

3.1.7 Experiment Tracking

Objective: Monitor records of the prompts, product comparisons, and metric scores.
Implementation: `track_comparison_metric_result_node` configured the dataset type in the Kedro data catalog as a dedicated dataset for the experiment tracking feature.

3.2 Descriptive Statistics of Classified Pages

To better understand the distribution of product-related and non-product-related pages, we provide the following summary as shown in Figures 2 and 3:

Based on Figures 2 and 3, the basic prompt classified an average of 52.77% of pages as product-related and 47.23% as non-product. The advanced prompt improved classification, increasing the proportion of product-related pages to 63.10% while reducing non-product classifications to 36.90%. Across all companies, ERGO and ARAG had the highest proportion of correctly classified product-related pages, while AXA had the lowest. The improvements from the advanced prompt were particularly significant for AXA and DKV, increasing classification accuracy by over 10 percentage points.

Company	Product	Non-Product
ARAG	83.78	7.61
AXA	37.50	52.29
Barmenia	55.00	35.29
DKV	44.07	44.78
ERGO	74.14	15.78
Generali	59.12	29.80
HUK-COBURG	42.93	46.27
R+V	52.60	37.20
SDK	46.81	35.51
Signal Iduna	56.00	34.38
Overall	52.77	47.23

Figure 2 Percentage of product-related and non-product-related pages classified by the basic prompt for each company.

Company	Product	Non-Product
ARAG	87.84	5.56
AXA	48.44	41.81
Barmenia	66.50	25.14
DKV	61.02	28.99
ERGO	87.93	6.87
Generali	71.70	19.51
HUK-COBURG	54.89	34.74
R+V	71.88	20.46
SDK	69.15	17.78
Signal Iduna	73.50	19.38
Overall	63.10	36.90

Figure 3 Percentage of product-related and non-product-related pages classified by the advanced prompt for each company.

Based on Table 3, the Basic prompt classified the majority of pages into the categories Krankenversicherung (31.69%), Haftpflichtversicherung (17.50%), and Rentenversicherung (6.34%), while 9.65% remained uncategorized. The Advanced prompt improved the classification distribution, slightly increasing accuracy in categories such as Rentenversicherung (7.10%), Gewerbeversicherung (5.20%), and Tierversicherung (2.84%), while Krankenversicherung (31.41%) and Haftpflichtversicherung (12.96%) remained the most frequently classified categories.

These statistics support the effectiveness of refining LLM-based classification through advanced prompt engineering.

Category	**Basic Prompt (Amount)**	**Basic Prompt %**	**Advanced Prompt (Amount)**	**Advanced Prompt %**
Krankenversicherung	335	31.69	332	31.41
Haftpflichtversicherung	185	17.50	137	12.96
No Category	102	9.65	111	10.50
Rentenversicherung	67	6.34	75	7.10
Gewerbeversicherung	47	4.45	55	5.20
Rechtsschutzversicherung	43	4.07	40	3.78
Reiseversicherung	43	4.07	35	3.31
Hausratversicherung	31	2.93	31	2.93
Unfallversicherung	29	2.74	31	2.93
Tierversicherung	21	1.99	30	2.84
Risikolebensversicherung	19	1.80	26	2.46
Pflegekostenversicherung	22	2.08	25	2.37
Berufsunfähigkeitsversicherung	19	1.80	21	1.99
Gebäudeversicherung	15	1.42	17	1.61
Pflegetagegeldversicherung	16	1.51	14	1.32
Kreditversicherung	4	0.38	11	1.04
Transportversicherung	8	0.76	10	0.95
Grundfähigkeitsversicherung	9	0.85	9	0.85
Betriebsunterbrechungsversicherung	9	0.85	8	0.76
Gemischte Lebensversicherung	3	0.28	7	0.66
Bauleistungsversicherung	6	0.57	7	0.66
Fahrerschutzversicherung	6	0.57	5	0.47
Maschinenkasko- und Maschinenbruchversicherung	4	0.38	5	0.47
Elementarversicherung	5	0.47	5	0.47
Geschäftsinhaltsversicherung	3	0.28	4	0.38
Pflegerentenversicherung	1	0.09	2	0.19
Montageversicherung	1	0.09	1	0.09
Vertrauensschadenversicherung	1	0.09	1	0.09
Private Arbeitslosenversicherung	1	0.09	1	0.09
Dread-Disease-Versicherung	2	0.19	1	0.09
Rücklaufversicherung	0	0.00	0	0.00

Table 3 Comparison of category classification results for Basic and Advanced prompts.

4 Experiments

In this experiment, different combinations of prompts are used to classify and generate product comparisons. We then analyze how these variations change the quality using LLM-as-a-Judge evaluation.

4.1 Binary Classification Experiment: Filtering Relevant Pages

At this stage, our primary objective is to filter out HTML pages that explicitly describe insurance products or services. As discussed in Section 3.1.1, our web crawling was performed without any preliminary filtering, resulting in a dataset that includes irrelevant pages such as career listings, blog articles, and generic landing pages. These pages do not contain the specific insurance product information required by our comparison pipeline. To address this, we introduced the second pipeline described in Section 3.1.2.

We employ an LLM-based classification method that examines the page URL (stored as metadata in the HTML structure). Simple NLP approaches—such as keyword matching or regex-based selection (e.g., filtering for words like *versicherung*, *insurance*, or *service*)—are prone to context-insensitive errors, leading to frequent false positives and negatives. Consequently, we need a more sophisticated approach that can interpret filenames semantically. To achieve this, we leverage GPT-4's advanced reasoning capabilities with two distinct prompting strategies.

Basic Prompt A minimal instruction directly asking whether the filename represents an insurance product page, without requiring additional reasoning steps.

Advanced Prompt A more detailed instruction set that guides the model through a step-by-step analysis of the filename. This encourages GPT-4 to make more contextual inferences and provide a brief explanation before concluding with "yes" or "no".

To measure the impact of this additional reasoning on classification performance, we manually annotated 1,600 HTML pages as either "insurance-related" or "non-related," creating a ground-truth dataset. We then evaluated the "product" vs. "non-product" decisions produced by each prompt using standard binary classification metrics (accuracy, precision, recall, and F1-score), as detailed in Section 2.3. This approach allowed us to quantify any performance gains from employing more advanced prompting techniques for URL-based page filtering.

4.2 Classification Experiment: Product Classification

The structured dataset used for the product classification experiment consists of insurance product names along with corresponding textual descriptions, extracted using LLM techniques described in Section 3.1.4. This dataset captures detailed and relevant information necessary for accurate insurance product categorization.

However, accurately matching comparable products across different insurance companies presents two primary challenges:

1. **Computational Complexity**: Matching each product with equivalents across different companies involves extensive iterative searching, causing high computational demands for LLM inference.

2. **User-friendly Categorization**: To create a structured and easily navigable database, products must be classified into clear and meaningful insurance categories, allowing users to efficiently explore and compare offerings.

To overcome these challenges, we implemented the *Product Classification Pipeline* (Section 3.1.5), which assigns each product into one of 30 standardized German insurance categories (*Versicherungsarten*) defined by a recognized taxonomy (Wikipedia Contributors 2025). The complete list of these categories is provided in Appendix A.1.

Instead of directly matching products across providers, this standardized classification allows users to efficiently compare products within well-defined and meaningful categories. The objective is not to broadly classify into generic groups (e.g., *Krankenversicherung*), but rather achieve finer-grained categorization into specialized and precise insurance types (e.g., *Dread-Disease-Versicherung*, *Private Arbeitslosenversicherung*), resulting in improved comparability and user clarity.

4.2.1 Evaluating Prompt Effectiveness

To analyze the effectiveness of prompt refinement, we compared two prompting approaches: a *Basic Prompt* and an *Advanced Prompt*. The evaluation considered two main criteria:

1. **Labeling Efficiency**: Reduction in products labeled as "Keine Übereinstimmung" (no match), indicating better product-category alignment.

2. **Distribution Improvement**: Enhanced classification into specialized categories rather than generic insurance groups, facilitating precise product comparison.

Basic Product Classification Prompt: The Basic Prompt provides minimal guidance, instructing the model only to match product details with provided categories or to respond with "No match".

Advanced Product Classification Prompt: The Advanced Prompt refines the Basic Prompt by introducing hierarchical decision-making. It explicitly instructs the model first to evaluate whether the product name itself indicates a clear category. If unclear, the model then analyzes detailed product category information (keywords, synonyms, and risks, see Appendix A.1) provided in the structured definitions. This method enhances classification accuracy and reduces ambiguity.

4.2.2 Statistical Evaluation

To objectively assess improvements, we compared the proportions of unlabeled products between basic and advanced prompting strategies. Furthermore, we statistically analyze shifted product distributions across categories. We two statistical methods we employ the chi-square test (Pearson 1900) assesses whether observed distribution differences are statistically significant, while Jensen-Shannon distance (Lin 1991) quantifies the magnitude of these distributional changes.

4.3 Product Comparison Experiment

In our pipeline as detailed in the Section 2.2, product comparisons are generated in 2 steps: First data is preprocessed using LLM to filter and group the relevant data (Section 4), and then product comparisons are generated for each grouped data using LLM.

The main goal of this experiment is to generate high-quality product comparisons. As with other machine learning problems, this is an optimization problem where two prompts at different steps are hyperparameters to be optimized (Singhvi et al. 2023; Pryzant et al. 2023; Soylu et al. 2024), as the output data quality heavily depends on the given context (Wei et al. 2025). This optimization is performed by iteratively selecting and updating prompts from a randomly generated set. The selection of prompts is based on the quality metrics of the generated comparisons.

In this experiment, inspired by the previous works on prompt optimization, we examined how different prompt combinations influence the generated product comparisons through the following steps:

1. We first generated product comparisons using different combinations of prompts: two for data preprocessing (Section 4.2) and two for product comparison generation.

2. Product comparisons are evaluated using **TruLens** metrics (G*roundedness*, *Comprehensiveness*, and *GroundTruthAgreement* — Section 2.4.2).

3. We then measured the impact of a different pair of prompts on the metrics using Cohen's d and Jensen-Shannon Distance(JSD)

4.3.1 Product Comparison Generation

Product Comparison Prompt

Building upon the product classification prompts introduced in Section 4, we defined the **basic product comparison prompt** and **advanced product comparison prompt** respectively.

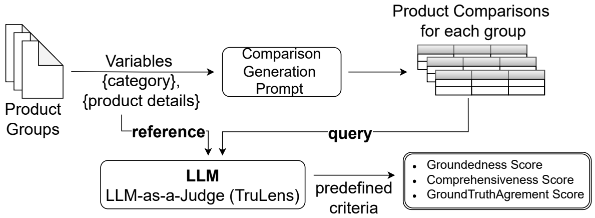


Figure 4 TruLens evaluation workflow

For readability, we denote the basic and advanced classification prompts as **v1** and **v2**, respectively, and the basic and advanced product comparison prompts as **v3** and **v4**. See Appendix C for prompt details.

Product Comparison Generation

To simulate the situation where prompts are randomly selected, we constructed all possible combinations of classification and product comparison prompts by taking the Cartesian product of the sets $\{\mathbf{v1}, \mathbf{v2}\}$ and $\{\mathbf{v3}, \mathbf{v4}\}$, resulting in the following four pairs.

- $(\mathbf{v1}, \mathbf{v3})$: basic classification + basic product comparison prompts

- $(\mathbf{v1}, \mathbf{v4})$: basic classification + advanced product comparison prompts

- $(\mathbf{v2}, \mathbf{v3})$: advanced classification + basic product comparison prompts

- $(\mathbf{v2}, \mathbf{v4})$: advanced classification + advanced product comparison prompts

Based on the combinations of prompts, product comparisons are generated for each group present in the classified data from the classification step (**v1**, **v2**). See Section 3.3 for the data structure.

4.3.2 Evaluation Setup

Product comparison prompts(**v3**, **v4**) take a list of **categories** and **product_texts** from companies in the same group to generate product comparisons(pipeline and data are explaines in Section 2 and 3, and the prompt details are in Appendix C). In other words, product comparisons can be considered either as statements or summaries that contain product information of companies in the same group. In this sense, TruLens LLM-as-a-Judge methods can be used to evaluate **Groundedness**, **Comprehensiveness**, and **GroundTruthAgreement** scores by using product information as reference data and product comparisons as queries(See Section 2.4.1 for *reference-guided grading*). Figure 4 illustrates the evaluation setup.

5 Results

In this section, we will be discussing and analyzing the results of each experiment.

5.1 Classification Experiment: Filtering Relevant Pages

To evaluate the effectiveness of the two prompting strategies (Basic and Advanced) described in Section 4.1, we compared their classification performance on a manually annotated dataset of 1,600 web pages. These pages were labeled either as *insurance-product-related* or *not-insurance-product-related*, creating a reliable ground truth for performance assessment.

As table 4 summarizes overall accuracy, precision, recall, and F1-score achieved by each prompting approach ; the basic prompt achieved consistently high performance across all metrics (accuracy: 92%, precision: 89%, recall: 92%, F1-score: 91%), indicating balanced and reliable page classification. This approach effectively differentiated insurance product pages from irrelevant ones, minimizing both false positives and false negatives.

In contrast, the Advanced prompt, designed to incorporate deeper semantic reasoning, demonstrated a notable performance trade-off. Although it achieved a very high recall (98%), indicating it rarely missed insurance-related pages, its lower accuracy (82%) and precision (74%) reflect a substantial increase in false positives. The resulting F1-score (84%) underscores this trade-off, emphasizing that increased recall was gained at the expense of overall accuracy.

Figures 6 and 7 further illustrate these patterns through confusion matrices, highlighting specific differences in error types between prompting methods. Additionally, Table 5 provides detailed numerical insights into true positive (TP), false positive (FP), false negative (FN), and true negative (TN) outcomes, clearly showing differences in error distribution.

Prompt	Accuracy	Precision	Recall	F1-Score
Basic	0.92	0.89	0.92	0.91
Advanced	0.82	0.74	0.98	0.84

Table 4 Overall classification performance comparing basic and advanced prompting strategies based on binary metrics.

Prompt	True Positives	False Positives	True Negatives	False Negatives
Basic	740	63	92	686
Advanced	785	18	274	504

Figure 5 Confusion matrix values (true positives, false positives, true negatives, false negatives) for basic and advanced prompts.

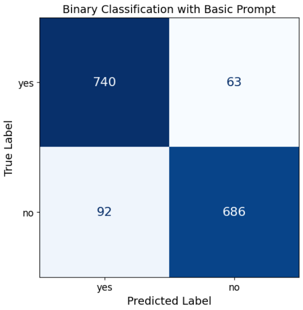


Figure 6 Confusion matrix for the basic prompt.

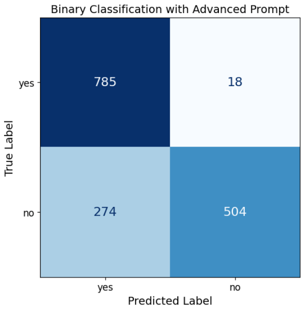


Figure 7 Confusion matrix for the advanced prompt.

Further analysis at the company-level (Tables 2 and 3) reveals that the basic prompt yields conservative yet accurate labeling, effectively minimizing incorrect classifications. In contrast, the advanced prompt tends towards broader labeling, improving the capture of product pages across all companies, albeit at the risk of including more irrelevant content.

Overall, the results illustrate a clear methodological trade-off: the basic prompt offers reliable, balanced accuracy ideal for scenarios prioritizing precision, while the advanced prompt enhances recall significantly, making it suitable when minimizing false negatives is paramount despite sacrificing precision.

5.2 Category Classification Experiment: Product Classification

This experiment aimed to evaluate how effectively the basic and advanced prompts (Section 3.1.5) assigned insurance products to one of 30 standardized categories. The resulting distributions of products across categories are illustrated in Figure 8, with corresponding counts and percentages detailed in Table 3.

Both prompts labeled the largest share of products as *Krankenversicherung* (31.69% for the basic prompt versus 31.41% for the advanced prompt). No-

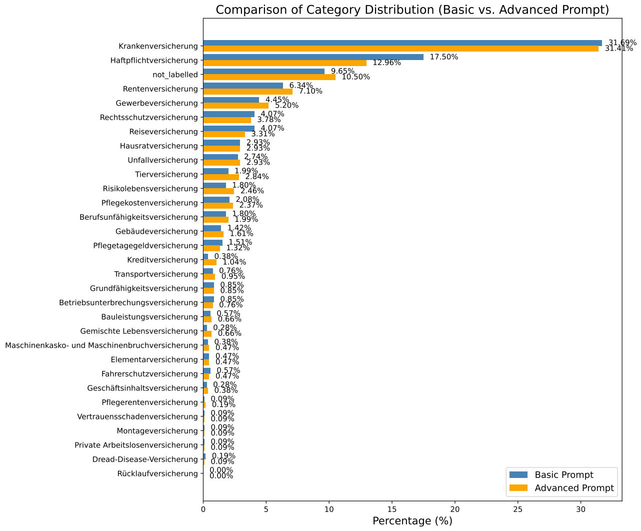


Figure 8 Comparison of category distributions between basic and advanced prompts.

table discrepancies appeared in categories such as *Haftpflichtversicherung* (17.50% basic versus 12.96% advanced) and *Tierversicherung* (1.99% basic versus 2.84% advanced). Moreover, 9.65% of products remained unlabeled (No Category) under the basic prompt, compared to 10.50% under the advanced prompt, suggesting that both prompts occasionally struggled to identify a suitable category.

5.2.1 Statistical Analysis

A chi-square test ($\chi^2 = 19.09$, $p = 0.9191$, $df = 29$) revealed no statistically significant difference in the product distributions assigned by each prompt. The Jensen-Shannon distance (0.0676) also indicated high overall similarity, despite minor category-specific variations.

These findings suggest that the advanced prompt placed slightly more products in specialized categories (such as *Gewerbeversicherung* or *Tierversicherung*), while the basic prompt tended toward more general classifications (for example, *Haftpflichtversicherung*). Nonetheless, the chi-square analysis confirmed that these differences did not substantially alter the overall distribution of products. Hence, the choice between prompts depends on whether a more detailed or broader classification approach is desired.

5.3 Product Comparison Evaluation

In our evaluation experiment, product comparisons were generated across 23 insurance product categories for each pair of classification and product comparison prompt(**v1_v3**, **v1_v4**, **v2_v3**, and **v2_v4**). Here, **v1** and **v2** refer to basic and advanced classification prompts, while **v3** and **v4** correspond to basic and advanced product comparison prompts. See Appendix C for prompt details.

For each set of product comparisons, each entry was evaluated by an LLM-as-a-Judge based on three key metrics: groundedness, comprehensiveness, and ground truth agreement. For example, the result from a pair (**v1**, **v3**) is shown in the Table 5. The result from all pairs are shown in the Table 10 in Appendix A.2.

These metric values were tested for normality using the Shapiro-Wilk test. We then conducted a F-test for all metrics, where categories were defined by prompt combinations. The ANOVA F-statistics yielded p-values ranging from 0.2168 to 0.6456, indicating no statistically significant differences in means across groups. This could be due to the small sample size (23 insurance categories), which might have limited the test's ability to detect actual differences in metric scores.

Consequently, to measure the actual impact of different prompt combinations on the metrics, we computed Cohen's d and Jensen-Shannon Distance(JSD). Cohen's d quantifies the effect size between groups (where each group represents a metric distribution), while JSD measures the similarity between their probability distributions.

Usually Cohen's d values are considered to be low when its absolute values are less than 0.2 Cohen (2013), meaning the difference in the prompt configurations has almost no impact on the standardized means of two metric distributions, while JSD less than 0.1 indicates the difference in the prompts had a minimal impact on the distribution shift. Computed Cohen's d and Jensen-Shannon distance values are shown in the Table 6 and mean metric values are shown in the Table 7.

Groundedness Score

Small Cohen's d values indicate that there is little to no difference in the groundedness scores. However, (**v1_v3** vs **v2_v4**) showed the highest JSD value, indicating the change from the basic prompts to advanced prompts both in classification and product comparison steps significantly caused the distribution shift in groundedness. This transition of prompts also increased the mean groundedness, implying that even though the average groundedness did not change

groundedness	comprehensiveness	GroundTruthAgreement	company_list	category
0.8929	0.8750	1.0000	ergo, huk-coburg, . . .	Risikolebensversicherung
0.8929	0.3333	1.0000	ergo, barmeria, . . .	Berufsunfähigkeitsversicherung
0.8571	0.7333	1.0000	generali, . . .	Unfallversicherung
0.9667	0.0083	1.0000	generali, . . .	Pflegekostenversicherung
0.8667	0.4666	1.0000	generali, . . .	Haftpflichtversicherung
0.9206	1.1904	1.0000	generali, . . .	Hausratsversicherung
0.9643	0.9286	1.0000	generali, . . .	Gebäudeversicherung
0.9286	0.9206	1.0000	generali, . . .	Geschäftsinhaltsversicherung
0.9048	0.9302	1.0000	ergo, barmeria, . . .	Rechtsschutzversicherung
0.8000	0.7333	0.9000	generali, barmeria, . . .	Kfzversicherung
0.8889	0.9333	0.9333	ergo, tarmenia, . . .	Krankenversicherung
0.9091	0.7500	1.0000	ergo, tarmenia, . . .	Rentenversicherung
0.9524	0.8000	1.0000	ergo, tarmenia, . . .	Reiseversicherung
0.7778	0.9333	1.0000	ergo, tarmenia, . . .	Transportversicherung
0.6250	0.9333	1.0000	huk-coburg, axa, . . .	Faherchtarifversicherung
0.6154	0.9333	1.0000	huk-coburg, axa, . . .	Rechtsschutzversicherung

Table 5 Product comparison metrics for each category (product comparisons generated from a prompt pair **v1_v3**)

much, **v2_v4** generated more product comparisons with higher groundedness.

Comprehensiveness Score

Overall, Cohen's d values show a relatively strong effect size for some pairs. While (**v1_v3** vs **v2_v3**) and (**v1_v4** vs **v2_v3**) show negative effect sizes, indicating that (**v2_v3**) exhibits greater comprehensiveness compared to (**v1_v3**) and (**v1_v4**). This means that the transition from the basic classification prompt(**v1**) to the advanced classification prompt(**v2**) increased the comprehensiveness. On the other hand, (**v2_v3** vs **v2_v4**) shows a positive value, which implies that advanced product comparison prompt(**v4**) further filtered out the information.

JSD values indicate a moderate distributional shift. (**v1_v4** vs **v2_v3**) shows the greatest distributional shift, with the increase in the mean comprehensiveness score from 0.(**v1_v4**) to(**v2_v3**). This implies that advanced classification prompt(**v2**) had a great impact on the distribution shift. On the other hand, (**v1_v3** vs **v1_v4**) and (**v2_v3** vs **v2_v4**) showed a moderately high JSD values. This suggests that the transition from **v3** to **v4** tends to decrease comprehensiveness of product comparisons.

GroundTruthAgreement Score

Cohen's d values show a reatively strong effect size for some pairs. Especially (**v1_v3** vs **v2_v3**) and (**v2_v3** vs **v2_v4**) show the strong effect size. This indicates that the transition from the basic prompts(**v1**, **v3**) to the advanced prompt(**v2**, **v4**) changes the similarity scores significantly. JSD values show relatively small values, which indicates that regardless of the prompt choices, the distribution of the GroundTruthAgreement scores is not significantly affected.

Product Comparison		Groundedness		Comprehensiveness		GroundTruthAgreement	
Group 1	**Group 2**	**Cohen's d**	**JSD**	**Cohen's d**	**JSD**	**Cohen's d**	**JSD**
v1_v3	v1_v4	-0.1233	0.4677	0.1042	0.3442	-0.1262	0.3537
v1_v3	v2_v3	-0.0606	0.3387	-0.5358	0.2989	-0.6943	0.3166
v1_v3	v2_v4	-0.1497	0.5857	-0.0108	0.2658	-0.3418	0.1770
v1_v4	v2_v3	0.0742	0.2933	-0.6389	0.4303	-0.4268	0.1764
v1_v4	v2_v4	-0.0221	0.2809	-0.1187	0.2834	-0.1506	0.3025
v2_v3	v2_v4	-0.1057	0.3627	0.5474	0.3620	0.5631	0.2538

Table 6 Cohen's d and Jensen-Shannon Distance between product comparison metric distributions for all classification/product comparison basic and advanced prompt pairs

Comparison	Groundedness	Comprehensiveness	GroundTruthAgreement
v1_v3	0.4005	0.6288	0.9083
v1_v4	0.5449	0.5318	0.9217
v2_v3	0.5255	0.7355	1.0000
v2_v4	0.5492	0.5706	0.9522

Table 7 Mean evaluation metrics for product comparisons for all classification/product comparison basic and advanced prompt pairs(values : $0 \sim 1$, higher the better)

5.4 Result Data

Through our automated pipeline and experiments, we successfully generated structured insurance product comparison tables. From the initial dataset of 1,600 products, our approach identified 428 comparable insurance products across eight different companies within the same categories. By leveraging unsupervised DistilBERT embedding-based clustering techniques, as detailed in our Kedro pipeline, we effectively grouped similar products based on their textual descriptions. This method allowed us to compare multiple products simultaneously without relying on iterative prompting, overcoming the common limitation of pairwise comparisons. Below, we present examples of the automatically generated comparison tables, demonstrating the effectiveness of our approach in structuring and analyzing insurance product data. The comparison tables are shown in Table 2 and Table 1.

6 Discussions

RQ 1. How does reproducible framework facilitate the experiments

As described in Section 3, we implemented our pipeline using Kedro and took active steps to leverage its features at every stage. First, we structured each major workflow (e.g., data crawling, page filtering, product comparison, evaluation) as its own pipeline, and then we isolated unit-level functions within those pipelines as nodes. This approach enabled us to develop, test, and run each function independently, reducing interdependencies and preventing errors from propagating across components. Second, we utilized

Kedro's data catalog to keep all inputs, outputs, and parameters for each node clearly defined, allowing us to seamlessly adjust or tune parameters without risking consistency. Finally, we harnessedthe *Experiment Tracking* feature (Section 3.1.7) to record every prompt, intermediate dataset, and final comparison output. This tracking mechanism enabled us to iteratively refine our prompts, observe their immediate impact on product comparisons, and methodically converge on higher-quality results.

RQ 2. Which LLM-as-a-Judge evaluation metric is suitable to ensure reliable product comparisons?
The evaluation experiment (Section 5.3) showed that not only the product comparison prompts but also the classification prompts had a great impact on the product comparisons. The transition from the basic classification prompt(**v1**) to the advanced classification prompt(**v2**), given the same product comparison prompts, showed improvements not only in the groundedness but also in the comprehensiveness. On the other hand, product comparison prompts(**v3**, **v4**) did not have a consistent impact on groundedness and comprehensiveness scores. Given the same classification prompt, the transition from the basic product comparison prompt(**v3**) to the advanced product comparison prompt(**v4**) improved the groundedness but decreased comprehensiveness. This indicates that the advanced comparison prompt further filtered out irrelevant information. Consequently, we can conclude that advanced classification and product comparison prompts, namely **v2** and **v4**, have picked out more relevant information, thus improved the groundedness. In other words, the improvement in the groundedness represents that the product comparisons contain more relevant data, ensuring the reliability of the generated product comparisons.

RQ 3. How can prompt fine-tuning improve binary classification performance for identifying relevant webpage content?
The experimental findings (Table 4) show that the basic prompt achieves balanced performance across accuracy, precision, and recall, effectively reducing the misclassification of unrelated pages as insurance-related. In contrast, the advanced prompt exhibits a strong emphasis on recall—capturing nearly all relevant pages—while sacrificing precision and overall accuracy. The confusion matrices (Figures 6 and 7) and their numerical summaries (Table 5) further illustrate this trade-off, revealing fewer missed product pages but more irrelevant inclusions when using the advanced prompt. Company-level analyses (Tables 2 and 3) confirm that the basic prompt applies more conservative labels, whereas the advanced prompt more aggressively classifies pages as product-related. These outcomes highlight a classic precision-versus-recall dilemma: the basic prompt is preferable when minimizing false positives is essential, while the advanced prompt is advantageous in contexts where thorough coverage of potential product pages is critical, even if additional filtering of extraneous data may be required subsequently.

RQ 4. How does prompt fine-tuning impact the efficiency of categorizing insurance products based on textual descriptions?
The experimental findings (Figure 8, Table 3) indicate minor category-specific variations between basic and advanced prompts, particularly in categories like *Haftpflichtversicherung* and *Tierversicherung*. Despite these slight differences, statistical analyses—the chi-square test ($p = 0.9191$) and Jensen-Shannon distance (0.0676)—showed no significant overall difference in category distributions. These results highlight that while the advanced prompt offers a more detailed and hierarchical approach to categorization, it did not significantly alter the overall distribution. Future refinements, including more robust prompting techniques and expanding the dataset, could potentially demonstrate clearer advantages of advanced prompting methods in achieving more precise insurance product categorization.

7 Limitations

Despite the comprehensive approach, several limitations must be acknowledged:

- **Unstandardized Features:** Insurance product attributes (such as coverage or pricing) vary significantly across different providers. This variation makes consistent comparisons challenging and can complicate both the extraction and classification tasks.

- **Limited Feature Availability:** Many websites either omit or only partially disclose key product details (e.g., pricing structures, coverage limits). This lack of uniform, detailed information can lead to reduced classification accuracy and incomplete comparisons.

- **Restricted Category Scope:** Some niche or emerging insurance types may not fit neatly into the established, standard insurance categories derived from publicly available lists (such as the Wikipedia *Versicherungsart* list). As a result, certain products may be misclassified or overlooked entirely.

8 Conclusion

In this work, we explored leveraging Large Language Models (LLMs) combined with advanced prompt engineering techniques within a structured, reproducible Kedro pipeline to automate and enhance the quality of insurance product comparisons. Our experiments revealed that advanced prompting techniques significantly impact the quality metrics of LLM-generated outputs. Specifically, we observed that the advanced classification prompt improved comprehensiveness and groundedness, while advanced product comparison prompts primarily improved groundedness by refining the precision of extracted features. However, our statistical analysis indicated no significant difference in the overall classification distributions between basic and advanced prompts, underscoring a nuanced trade-off between recall and precision. Therefore, the choice of prompt strategy should depend on the specific application requirements, especially regarding the desired balance between data completeness and factual accuracy.

9 Future Work

For future work, we propose exploring more sophisticated prompting techniques such as Zero-Shot-Chain-of-Thought(Kojima et al. 2022), which could further enhance the LLM's reasoning capabilities, potentially mitigating issues related to hallucinations and ambiguous classifications. Additionally, integrating human-in-the-loop validation or leveraging hybrid evaluation strategies that combine traditional metrics with LLM-as-a-Judge could further ensure the reliability and accuracy of generated product comparisons. Expanding the dataset by crawling additional insurance providers and refining feature extraction methodologies to standardize product attributes across diverse sources would also represent important steps toward broader applicability and scalability of our approach.

References

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

Helicone AI. Use constrained outputs (prompt engineering guide). https://docs.helicone.ai/guides/prompt-engineering/use-constrained-outputs, 2023a. Helicone Documentation, accessed Mar 2025.

Helicone AI. Use structured formats (prompt engineering guide). https://docs.helicone.ai/guides/prompt-engineering/use-structured-formats, 2023b. Helicone Documentation, accessed Mar 2025.

Yupeng Chang, Xu Wang, Jindong Wang, Yuan Wu, Linyi Yang, Kaijie Zhu, Hao Chen, Xiaoyuan Yi, Cunxiang Wang, Yidong Wang, et al. A survey on evaluation of large language models. *ACM transactions on intelligent systems and technology*, 15(3): 1–45, 2024.

Jacob Cohen. *Statistical power analysis for the behavioral sciences*. routledge, 2013.

Harrison Fagbohun and Dereventsov. Llm-prompting: Understanding the bridge between human communication and computational models. *NLP Research Journal*, 2024.

Jiawei Gu, Xuhui Jiang, Zhichao Shi, Hexiang Tan, Xuehao Zhai, Chengjin Xu, Wei Li, Yinghan Shen, Shengjie Ma, Honghao Liu, et al. A survey on llm-as-a-judge. *arXiv preprint arXiv:2411.15594*, 2024.

Aric A. Hagberg, Daniel A. Schult, and Pieter J. Swart. Exploring network structure, dynamics, and function using NetworkX. In Gaël Varoquaux, Travis Vaught, and Jarrod Millman, editors, *Proceedings of the 7th Python in Science Conference (SciPy2008)*. Pasadena, CA USA, 2008.

Kawtar Hamimoune. Can ai judge itself? comprehensive automated self-assessment of large language models in diverse nlp tasks. Master's thesis, Technical University of Munich, School of Computation, Information and Technology, 2025.

Kedro. Kedro Documentation. https://kedro.org/, 2019. Accessed: 2023.

Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. Large language models are zero-shot reasoners. *Advances in neural information processing systems*, 35:22199–22213, 2022.

Jianhua Lin. Divergence measures based on the shannon entropy. *IEEE Transactions on Information Theory*, 37(1):145–151, 1991.

Yang Liu, Dan Iter, Yichong Xu, Shuohang Wang, Ruochen Xu, and Chenguang Zhu. G-eval: Nlg evaluation using gpt-4 with better human alignment. *arXiv preprint arXiv:2303.16634*, 2023.

Ben Mann, N Ryder, M Subbiah, J Kaplan, P Dhariwal, A Neelakantan, P Shyam, G Sastry, A Askell, S Agarwal, et al. Language models are few-shot learners. *arXiv preprint arXiv:2005.14165*, 1:3, 2020.

Cesar Miguelañez. Iterative prompt refinement: Step-by-step guide. https://latitude-blog.ghost.io/blog/iterative-prompt-refinement-step-by-step-guide/, 2025. Latitude AI Blog, posted Jan 30, 2025.

Karl Pearson. On the criterion that a given system of deviations from the probable in the case of a correlated system of variables is such that it can reasonably be supposed to have arisen from random sampling. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, 50 (302):157–175, 1900.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*, 12: 2825–2830, 2011.

Reid Pryzant, Dan Iter, Jerry Li, Yin Tat Lee, Chenguang Zhu, and Michael Zeng. Automatic prompt optimization with" gradient descent" and beam search. *arXiv preprint arXiv:2305.03495*, 2023.

Yujia Qin, Shihao Liang, Yining Ye, Kunlun Zhu, Lan Yan, Yaxi Lu, Yankai Lin, Xin Cong, Xiangru Tang, Bill Qian, et al. Toolllm: Facilitating large language models to master 16000+ real-world apis. *arXiv preprint arXiv:2307.16789*, 2023.

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*, Hong Kong, November 2019. Association for Computational Linguistics.

Leonard Richardson. Beautiful soup documentation. https://www.crummy.com/software/BeautifulSoup/bs4/doc/, 2021. Accessed: 2023.

Sander Schulhoff. Technique #2: Roles in prompts. https://learnprompting.org/docs/basics/roles, 2025. Last updated Feb 18, 2025, LearnPrompting Guide.

Arnav Singhvi, Manish Shetty, Shangyin Tan, Christopher Potts, Koushik Sen, Matei Zaharia, and Omar Khattab. Dspy assertions: Computational constraints for self-refining language model pipelines. *arXiv preprint arXiv:2312.13382*, 2023.

Dilara Soylu, Christopher Potts, and Omar Khattab. Fine-tuning and prompt optimization: Two great steps that work better together. *arXiv preprint arXiv:2407.10930*, 2024.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837, 2022.

Peipei Wei, Dimitris Dimitriadis, Yan Xu, and Mingwei Shen. Don't just demo, teach me the principles: A principle-based multi-agent prompting strategy for text classification. *arXiv preprint arXiv:2502.07165*, 2025.

Wikipedia Contributors. Versicherungsart – German Insurance Product Types, 2025. URL https://de.wikipedia.org/wiki/Versicherungsart. [Accessed: March 2025].

Xiao Yu, Zexian Zhang, Feifei Niu, Xing Hu, Xin Xia, and John Grundy. What makes a high-quality training dataset for large language models: A practitioners' perspective. In *Proceedings of the 39th IEEE/ACM International Conference on Automated Software Engineering*, pages 656–668, 2024.

et al. Zamfirescu. Understanding prompt engineering: A deep dive into llm interaction strategies. *AI NLP Advances*, 2023.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, et al. Judging llm-as-a-judge with mt-bench and chatbot arena. *Advances in Neural Information Processing Systems*, 36:46595–46623, 2023.

A Appendix

A.1 List of German Insurance Product Types

The following table presents the 30 standard insurance product categories (*Versicherungsart*), as referenced in Section 3.1.5.

Risikolebensversicherung, Gemischte Lebensversicherung, Rentenversicherung, Berufsunfähigkeitsversicherung, Pflegerentenversicherung, Krankenversicherung, Dread-Disease-Versicherung, Grundfähigkeitsversicherung, Pflegekostenversicherung, Pflegetagegeldversicherung, Haftpflichtversicherung, Betriebsunterbrechungsversicherung, Hausratversicherung, Gebäudeversicherung, Geschäftsinhaltsversicherung, Gewerbeversicherung, Rücklaufversicherung, Bauleistungsversicherung, Maschinenkasko- und Maschinenbruchversicherung, Kreditversicherung, Vertrauensschadenversicherung, Montageversicherung, Elementarversicherung, Unfallversicherung, Reiseversicherung, Transportversicherung, Private Arbeitslosenversicherung, Tierversicherung, Fahrerschutzversicherung, Rechtsschutzversicherung

Insurance Product Definitions, and Synonyms

This table is prepared to guide LLM model to feed prompt to find better matching of product by providing their definitions and synonyms in german.

Versicherungsart	Definition	Synonyme/Schlüsselwörter
Risikolebensversicherung	Zahlt nur im Todesfall der versicherten Person während der Laufzeit.	Todesfallschutz, Hinterbliebenenschutz, reine Lebensversicherung ohne Sparanteil.
Gemischte Lebensversicherung	Kombination aus Lebensversicherung (Todesfallschutz) und Sparanteil (Erlebensfallleistung).	Kapitalaufbau, klassisches Lebensvorsorgemodell.
Rentenversicherung	Aufbau einer Altersvorsorge mit laufender oder später Rente.	Altersvorsorge, Sofortrente, aufgeschobene Rente, private Rente.
Berufsunfähigkeitsversiche	Einkommensschutz bei dauerhafter Arbeitsunfähigkeit durch Unfall oder Krankheit.	BU, Arbeitsunfähigkeit, Invalidität.
Pflegerentenversicherung	Monatliche Rente bei Pflegefall (Pflegegrad).	Pflegerente, Pflegebedürftigkeit.
Krankenversicherung	Erstattung medizinischer Kosten (privat, gesetzlich, Zusatz).	Zahnzusatz, Auslandskranken, stationär, ambulant.
Dread-Disease-Versicherung	Einmalzahlung bei Diagnose schwerer Krankheiten (z.B. Krebs, Herzinfarkt).	Schwere Krankheiten, Krankheitsfall-Leistung.
Grundfähigkeitsversicherur	Deckt den Verlust grundlegender Fähigkeiten (z.B. Sehen, Gehen).	Invalidität, Fähigkeitenkatalog, Alltagsfähigkeiten.
Pflegekostenversicherung	Ersetzt tatsächliche Pflegekosten (stationär/ambulant) bei Pflegebedürftigkeit.	Pflegekosten, Pflegestufe, Pflegegrad.
Pflegetagegeldversicherung	Tagegeld bei Pflege, zusätzliches Geld für Pflegebedürftige.	Pflegetagegeld, Pflegezusatz.

Haftpflichtversicherung	Deckt Schäden an Dritten (Personen-, Sach-, Vermögensschäden).	Privathaftpflicht, Firmenhaftpflicht, Verschulden, Schadenersatz.
Betriebsunterbrechungsvers	Schutz vor Ertragsausfällen und laufenden Kosten bei Betriebsstörungen.	Betriebsausfall, Ertragsausfall, Produktionsstopp.
Hausratversicherung	Schützt Einrichtungs- und Gebrauchsgegenstände in der Wohnung.	Feuer, Einbruch, Leitungswasser, privater Haushalt.
Gebäudeversicherung	Deckt Schäden am Wohn- oder Geschäftsgebäude (z.B. Sturm, Hagel, Leitungswasser).	Wohngebäude, Immobilie, Gebäudeschutz.
Geschäftsinhaltsversicherun	Versichert Firmeninventar, Warenvorräte, Betriebseinrichtung gegen Feuer, Einbruch, Vandalismus.	Geschäftsinhalt, betriebliches Inventar.
Gewerbeversicherung	Oberbegriff für Versicherungen von Gewerbetreibenden (z.B. Haftpflicht, Rechtsschutz, Inhaltsversicherung).	Firmenversicherung, Geschäftskunden.
Rücklaufversicherung	Bei Tod der versicherten Person fließt Restkapital an Hinterbliebene zurück.	Kapitalrückgewähr, Er-/Ablebensversicherung.
Bauleistungsversicherung	Schützt laufende Bauprojekte vor unvorhersehbaren Schäden an Bauleistungen/-stoffen.	Baustelle, Neubau, Umbau, Witterungsschäden.
Maschinenkasko- und Maschinenbruchversicherung	Deckt Schäden an stationären oder mobilen Maschinen (z.B. Produktions-, Baumaschinen).	Reparaturkosten, Bedienfehler, Unfall, Maschinenbruch.
Kreditversicherung	Absicherung vor Forderungsausfällen, wenn Kunden/Abnehmer nicht zahlen.	Debitorenausfall, Zahlungsausfall, Lieferantenkredit.
Vertrauensschadenversicher	Schutz vor Vermögensschäden durch betrügerisches Handeln eigener Mitarbeiter.	Unterschlagung, Diebstahl, Betrug im Betrieb.
Montageversicherung	Versicherung für Schäden an Anlagen/-Maschinen während Montage/Installation.	Installationsphase, Bau, Anlagenaufbau.
Elementarversicherung	Deckt Naturgefahren wie Überschwemmung, Erdbeben, Erdrutsch, Schneedruck.	Naturkatastrophen, Hochwasser, Erdrutsch.
Unfallversicherung	Zahlt Kapital oder Rente bei Unfallfolgen (Invalidität, Unfalltod).	Freizeitunfall, 24-Stunden-Schutz, Einmalleistung.
Reiseversicherung	Schützt bei Reiserücktritt, Gepäckverlust, Auslandskranken usw.	Urlaub, Reiseabbruch, Auslandsreise.
Transportversicherung	Deckt Waren und Güter im Transportweg.	Logistik, Fracht, Spedition.
Private Arbeitslosenversicherung	Ergänzt staatliches Arbeitslosengeld.	Einkommensschutz bei Jobverlust, ALG-Plus.

Tierversicherung	Absicherung für Tierarztkosten oder Tierhaftpflicht.	Hund, Katze, Pferd, OP-, Kranken-, Haftpflicht.
Fahrerschutzversicherung	Deckt Eigenschäden des Fahrers bei selbstverschuldetem Unfall.	Fahrerunfall, Schmerzensgeld, Schutzbrief Fahrer.
Rechtsschutzversicherung	Übernimmt Kosten bei Rechtsstreit (Anwalt, Gericht, Gutachter).	Verkehrsrechtsschutz, Arbeitsrechtsschutz, Mietrechtsschutz.

A.2 Category Distribution Comparison Details

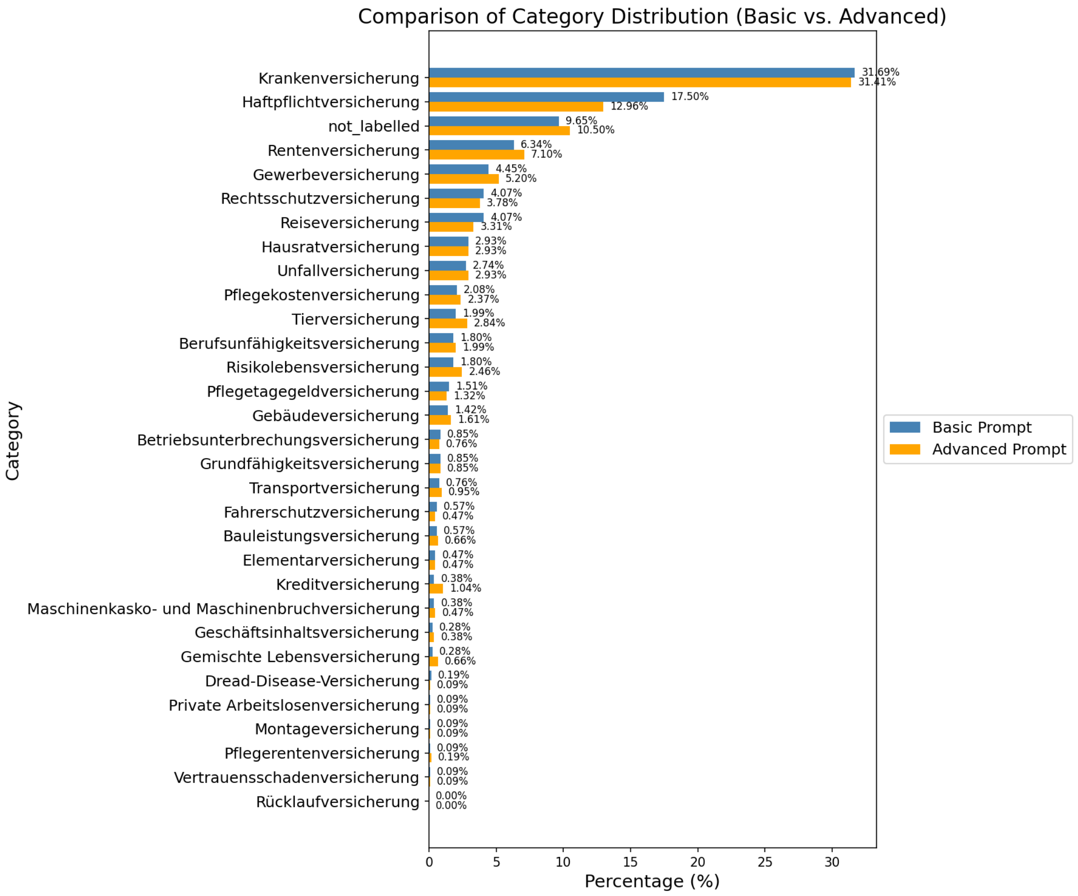


Figure 9 Comparison of category distributions between Basic and Advanced prompts

Category	Basic Count	Basic %	Advanced Count	Advanced %
Risikolebensversicherung	19	1.80	26	2.46
Gemischte Lebensversicherung	3	0.28	7	0.66
Rentenversicherung	67	6.34	75	7.10
Berufsunfähigkeitsversicherung	19	1.80	21	1.99
Pflegerentenversicherung	1	0.09	2	0.19
Krankenversicherung	335	31.69	332	31.41
Dread-Disease-Versicherung	2	0.19	1	0.09
Grundfähigkeitsversicherung	9	0.85	9	0.85
Pflegekostenversicherung	22	2.08	25	2.37
Pflegetagegeldversicherung	16	1.51	14	1.32
Haftpflichtversicherung	185	17.50	137	12.96
Betriebsunterbrechungsversicherung	9	0.85	8	0.76
Hausratversicherung	31	2.93	31	2.93
Gebäudeversicherung	15	1.42	17	1.61
Geschäftsinhaltsversicherung	3	0.28	4	0.38
Gewerbeversicherung	47	4.45	55	5.20
Bauleistungsversicherung	6	0.57	7	0.66
Maschinenkasko- und Maschinenbruchversicherung	4	0.38	5	0.47
Kreditversicherung	4	0.38	11	1.04
Vertrauensschadenversicherung	1	0.09	1	0.09
Montageversicherung	1	0.09	1	0.09
Elementarversicherung	5	0.47	5	0.47
Unfallversicherung	29	2.74	31	2.93
Reiseversicherung	43	4.07	35	3.31
Transportversicherung	8	0.76	10	0.95
Private Arbeitslosenversicherung	1	0.09	1	0.09
Tierversicherung	21	1.99	30	2.84
Fahrerschutzversicherung	6	0.57	5	0.47
Rechtsschutzversicherung	43	4.07	40	3.78
not_labelled	102	9.65	111	10.50

Table 9 Comparison of category classification results for Basic and Advanced prompts.

B Metric Score Distribution

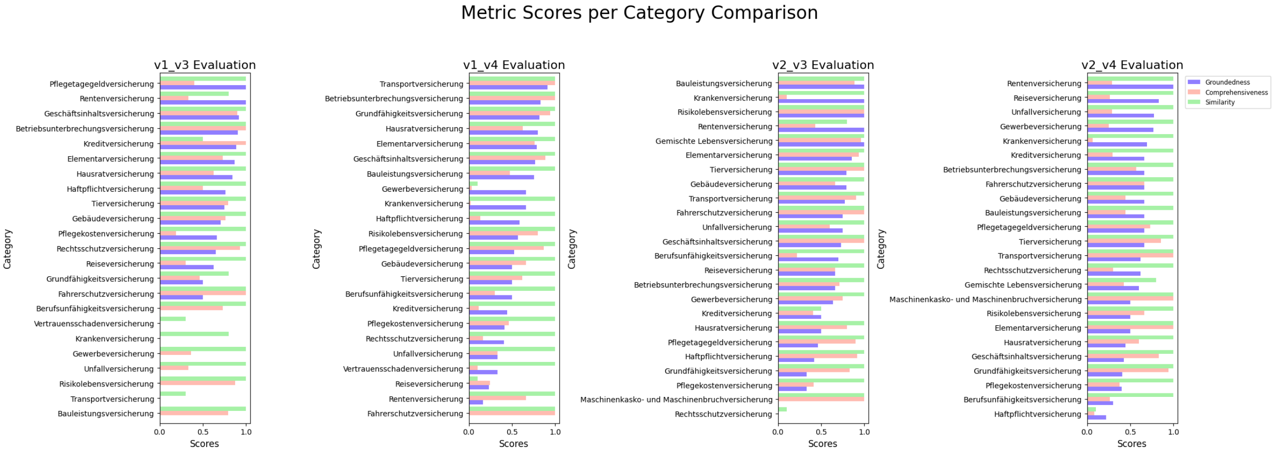


Figure 10 Product comparison metric distributions for all pairs of prompts

C Prompts

Basic Binary Classification Prompt

Instruction: *The following is a filename from a company's website:*
`Filename: {filename}`
Question: Does this filename represent a product page? **Answer:** 'yes' or 'no' and nothing else.

Advanced Binary Classification Prompt

[System] You are an advanced AI system specializing in analyzing insurance webpages. Provide accurate, context-specific classifications.
In this task:

1. Reason step-by-step whether the given filename suggests an insurance product or service.
2. Provide a final answer clearly separated as 'yes' or 'no'.

[Human] `Filename: {filename}`
Task:

1. Carefully analyze whether the filename strongly suggests an insurance product or service.
2. If the filename is generic (e.g., 'about.html', 'career_service.html') or clearly unrelated to a specific insurance product, answer 'no'.
3. Provide a brief reasoning, then clearly conclude with your final answer ('yes' or 'no').

Basic Prompt Product Classification Prompt (v1)

Product Name: `{product_name}`
Product Details:

`{details}`

Insurance Categories:

`{categories}`

Response Instructions: If the product matches no category, respond with `"No match"`. Otherwise, respond with the exact category name only.

Advanced Product Classification Prompt (v2)

[System Instructions] You are a specialized AI system for insurance classification. Your task is to identify the correct insurance category based on the product name and details provided.

Classification Rules:

1. Assign immediately if the product name clearly matches a category.
2. Otherwise, analyze details for keywords, synonyms, or covered risks to determine the best match.
3. Respond with `"No match"` only if no category fits.

[Human Input] Below is the product name, details, and category list with definitions and synonyms.

Product Name: `{product_name}`
Product Details:

```
{details}
```

Categories with Definitions:

```
{categories, definitions, keywords}
```

Structured Decision Process:

1. Use product name if clearly categorizable.
2. Otherwise, check details for relevant indicators (keywords, synonyms, risks, etc.).
3. Select the best-fitting category.
4. Use `"No match"` only if no suitable category is found (avoid this when possible).

[Expected Output] Provide only the exact category name (e.g., `"Hausratversicherung"`) or `"No match"`—no explanations.

Feature Extraction Prompt

You are an advanced AI specialized in analyzing **German insurance text**. Your task is to extract and return a **single text block** containing only the relevant insurance product or service information from the provided content.

[USER MESSAGE]
You have text from a German insurance `.md` file:

```
"""
{content}
"""
```

Requirements:

1. **Extract** only relevant coverage descriptions, features, pricing, benefits, disclaimers, phone numbers, or any essential information that helps a user understand the insurance product.
2. **Exclude** short bullet lists of disclaimers or navigation if they do not describe actual coverage. Also exclude repeated placeholders, headings for unrelated products, or purely navigational text (e.g., “`Zur Suche`”, “`Kontakt`”).
3. **Merge** the relevant paragraphs into a coherent text block.
4. **No bullet points**; produce a paragraph or a single text block describing coverage.
5. **Do not** provide any commentary or chain-of-thought reasoning. Just output the final text.
6. If nothing relevant is found, output an **empty string**.

Final Output:

- A **single block of text** containing extracted coverage details.
- If no coverage information is found, output an **empty string**.
- **No JSON keys** should be added. The system will store the output under `"details"`.

Basic Product Comparison Prompt (v3)

System Message:
Create a simple comparison table in Markdown format for `{{CATEGORY}}` insurance products from different companies. The table should have the following columns: Feature and details for each company.

Include these key aspects:

- **Coverage:** What does the insurance cover?
- **Price:** How much does it cost?
- **Deductible:** Is there any deductible?
- **Exclusions:** What is not covered?
- **Extras:** Are there any additional benefits?
- **Cancellation Policy:** How can the insurance be canceled?

Company Information:

`{{COMPANY_INFO}}`

Example Output:

```
| Feature         | Company A | Company B | Company C | Company D |
|----------------|-----------|-----------|-----------|-----------|
| Coverage       |           |           |           |           |
| Price         |           |           |           |           |
| Deductible    |           |           |           |           |
| Exclusions    |           |           |           |           |
| Extras        |           |           |           |           |
| Cancellation  |           |           |           |           |
```

Instructions:

- Keep the table clear and easy to read.
- Use simple and direct wording.
- Only include relevant details.

Advanced Product Comparison Prompt (v4)

System Message:
You are an expert in insurance product analysis and comparison. Your goal is to generate a concise and clear Markdown comparison table for a set of insurance products within the same cluster. Use a chain-of-thought reasoning process internally to first analyze the provided product details and identify the key features that are most relevant for comparing these products. These features may include (but are not limited to) Product Name, Coverage Details, Premiums, Benefits, Additional Support, and Unique Features—choose those that best reflect the specifics of the cluster.

User Message:
Below are the details for insurance products in the category `{{CATEGORY}}`. Please perform the following steps:

1. **Chain-of-Thought Analysis:**
   - Carefully analyze the `"details_list"` text for each product.
   - Determine the most important and comparable features specific to this cluster.
   - Internally list out these features and decide on the key aspects that are comparable across all products.
2. **Generate Comparison Table:**
   Produce a Markdown table structured as follows:

   ```
   | Feature          | Company:       | Company:       | ... |
   |                  | [Company Name] | [Company Name] |     |
   |                  | [Product Name] | [Product Name] |     |
   |------------------|----------------|----------------|-----|
   | Feature 1        | [Value]        | [Value]        | ... |
   | Feature 2        | [Value]        | [Value]        | ... |
   | ...              | ...            | ...            | ... |
   ```

   Replace `Feature 1`, `Feature 2`, etc. with concise, relevant feature names.
3. **Formatting Requirements:**
   - Shorten feature names as needed.
   - Use "N/A" if a feature is missing.

Final Output:
Output only the final Markdown table (without chain-of-thought reasoning).

Product Details:

`{{PRODUCTS_TEXT}}`