

LLM-Based Examination of Eligibility Criteria from Securities Prospectuses at the German Central Bank

Serhii Hamotskyi, Akash Kumar Gautam, Christian Hänig

Anhalt University of Applied Sciences

{serhii.hamotskyi, akash-kumar.gautam, christian.haenig}@hs-anhalt.de

Abstract

Verifying the eligibility of securities as collateral is a key responsibility of the German Central Bank. However, manually verifying these assets against legal and financial criteria within lengthy, semi-structured, and often bilingual prospectuses is a resource-intensive task. While previous efforts utilized traditional Named Entity Recognition (NER) for information extraction, these methods can struggle with OCR noise, linguistic variance, and rigid span-based constraints, and the need for manually annotated training data for each relevant annotation type. In this paper, we present the first case study applying Large Language Models (LLMs) to the eligibility examination process, shifting the paradigm toward a generative Information Extraction pipeline. Our approach decomposes the task into extraction, normalization, and interpretation, allowing for greater flexibility in handling noisy text and interleaved German-English content. We further introduce a value-based evaluation methodology using LLM-as-a-judge, which offers a more semantic assessment than location-based metrics. Our results demonstrate that LLM-based systems achieve high precision (up to 91%) in document-level eligibility, exhibiting a conservative operating profile that minimizes false acceptance.

Keywords: Large Language Models, Information Extraction, LLM-as-a-Judge, Financial NLP.

1. Introduction

As the central bank of the Federal Republic of Germany and a core member of the Eurosystem, the *Deutsche Bundesbank* is responsible for implementing monetary policy and providing liquidity to the financial system. These operations are conducted as credit transactions: they must be backed by collateral to protect the central bank from financial risk. The acceptance of a security as collateral depends on its **eligibility**, which is based on specific legal and financial criteria to ensure that only high-quality assets are pledged ([European Central Bank, 2017](#)).

The eligibility of securities is assessed on the basis of their **prospectuses**, which can be hundreds of pages long. Thousands of securities are issued annually, and verifying their eligibility is a time-consuming and tedious process, making eligibility estimation a prime target for automation.

The difficulty lies in the semi-structured nature of the source material, with evidence being scattered across the entire document and expressed through dozens of different conventions. Prospectuses can be bilingual, with English or German interleaved or presented in parallel columns, requiring models that are robust to language switching, ideally — able to use the information from both.

Previous work by [Hänig et al. \(2023\)](#) addressed this challenge by developing a Decision Support System that models the task as a Named Entity Recognition (NER) problem, solved using Transformer-based models ([Vaswani et al., 2017](#)), achieving good results on most criteria. Neverthe-

less, that approach introduced several constraints, primarily: it required extensive manual annotation to provide necessary supervision for all relevant annotation types, and the resulting models were sensitive to the rigid boundaries of text spans (which made them fragile when encountering OCR artifacts or financial language different from its training set).

In this paper, we present the first case study applying Large Language Models (LLMs) to the eligibility examination process at the German Central Bank, shifting the paradigm from traditional token classification to a generative Information Extraction approach.

The main **contributions** are as follows.

- Presenting a multi-stage generative Information Extraction pipeline that allows the handling of linguistic structures that token-classification models may struggle to process
- Introducing a value-based evaluation methodology using LLM-as-a-judge, resistant to OCR noise and linguistic variance

Although specialized domain models have shown good results in financial tasks, our study focuses on the zero-shot and instruction-following capabilities of high-performance general-purpose models: Llama-3.3-70B-Instruct ([Grattafiori et al., 2024](#)) and Cohere Command-R 08-2024¹ for inference, and Mistral Small 3.1 Instruct² for evaluation.

¹<https://hf.co/CohereLabs/c4ai-command-r-08-2024>

²<https://hf.co/mistralai/Mistral-Small-3.1-24B-Instruct-2503>

2. Examination of Eligibility Criteria

In the context of this case study, eligibility is determined by 6 criteria³, all of which must be fulfilled for the prospectus to be eligible. The descriptions that follow are simplifications and do not fully reflect the Eurosystem eligibility criteria⁴.

Currency One of: EUR, USD, GBP, JPY

Type of instrument Only certain types of financial instruments are allowed (e.g. stocks are not)

Principal amount Only fixed and unconditional amounts

Redemption (amount) at maturity The principal amount must be repaid in full at bond maturity

Coupon Only certain coupon structures allowed

Status Not subordinated to other debt.

The first 4 criteria will be referred to as “simple” criteria, as they depend only a single extracted entity from the document text. The last two *complex* criteria — *coupon* and *status* — are determined using a decision tree using *multiple* types extracted from the prospectus, and master data (asset type, issuer group, issuer date).

Each security is described by three data points. **Prospectuses** are PDF files describing the terms and conditions governing the issuance of the security. A **base prospectus**, if present, is considered part of the prospectus. (An issuer might have a base prospectus with overall standard terms, and issue prospectuses for each individual security containing information applicable to it specifically.) Crucially, the annotations in base prospectuses are available during inference and, in most cases, take precedence over data predicted from the prospectus. Each prospectus is always accompanied by **master data** (*Stammdaten*) — additional metadata about each prospectus, including its **eligibility** and the name of its base prospectus if present. These (except eligibility) are also available during inference.

3. Related Work

Information Extraction (IE) from financial documents is a rapidly evolving field, recently transitioning from traditional discriminative models to generative Large Language Models (LLMs).

Information extraction Colakoglu et al. (2025) systematically evaluates different building blocks for LLM-based IE in layout-rich documents, including input formats, prompt structures, and

cleanup/postprocessing steps applied to LLM output. Chen et al. (2025) discusses the importance and evaluation of prompt engineering as applied to IE, with a focus on OCR-derived data.

Lu and Huo (2025) is a recent “systematic evaluation of state-of-the-art LLM and prompting methods” applied specifically to financial NER, comparing them to Transformer-based models, and finding that the latter consistently outperform generic LLMs, with prompt design and in-context learning narrowing the gap.

Pre-trained Models for Financial Domain

Though we focus on zero-shot general-purpose models, domain-specific finetuning has often been used for similar tasks and shown to be able to outperform significantly larger models.

This includes early efforts like FinBert (Yang et al., 2020), German-language financial models (Kozueva et al., 2024), and more recent multilingual suites like LLM Pro Finance (Caillaut et al., 2025). See Lee et al. (2024) for a deeper review of finance-specific, including multimodal, LLMs.

Benchmarking and Evaluation FinBen (Xie et al., 2024) is a holistic financial benchmark for LLMs, spanning 24 tasks including information extraction. As tasks move beyond simple classification, “LLM-as-a-judge” has emerged as a vital tool for semantic evaluation. Current research focuses on improving the reliability and consistency of these automated judges to replace or augment rigid, offset-based metrics (Gu et al., 2025).

4. Data

4.1. The Dataset

Dataset Summary The dataset we use was originally created in the context of Hänig et al. (2023), who modeled the task as NER and annotated it as such.

The dataset is composed of 413 prospectuses, split into a train set (268 prospectuses) and a test set (145). The test set prospectuses were annotated twice (each⁵ by two different annotators), resulting in 285 *annotated* documents, of which 82 (~29%) were ineligible.

The PDF files are sourced from the FinCorpus-DE10k (Hamotskyi et al., 2024) corpus and share its characteristics, particularly regarding PDF layouts, OCR artifacts, and the parallel presentation of English-German text.

Bilinguality and PDF Layouts All documents are in German, and about a third are bilingual (En-

³In Hänig et al. (2023) two more are listed, and were excluded due to having few examples in the training data.

⁴<https://eur-lex.europa.eu/legal-content/EN/TXT/PDF/?uri=CELEX:32014O0060>

⁵A small number were unusable for technical reasons.

AT0000A2VB62 Wertpapierkennnum: isin	Währung, Nennbetrag (Stückelung), Anzahl der begebenen Schuldverschreibungen und Laufzeit der Schuldverschreibungen
type_of_instrument_eligible Schuldverschreibungen werden über die gesar	Die Schuldverschreibungen lauten auf Euro (EUR) mit einem Nennbetrag je Schuldverschreibung von EUR 1.000 (die "festgelegte Stückelung") und einem Gesamtnennbetrag von bis zu EUR 50.000.000 . Die Schuldverschreibungen haben eine feste Laufzeit, die spätestens am 11.02.2040 (der "Fälligkeitstag") endet, vorbehaltlich etwaiger vorzeitiger Rückzahlungsrechte oder eines Rückkaufs und einer Entwertung durch die Emittentin.
type_of_instrument_eligible Schuldverschreibungen sowie alle Rechte und P	Mit den Wertpapieren verbundene Rechte
AT0000A2VB62 / WKN: EB0F6W Währu: isin	Zinszahlungen aus den Schuldverschreibungen
Euro (EUR) mit einem Nenn: currency_eligible	Die Schuldverschreibungen werden auf der Grundlage ihres ausstehenden Gesamtnennbetrags vom Verzinsungsbeginn (wie nachstehend definiert) (einschließlich) bis zum Fälligkeitstag (ausschließlich) mit dem Zinssatz von 1,20% per annum verzinst.
(EUR) mit einem Nennbetr: currency_eligible	Der "Verzinsungsbeginn" der Schuldverschreibungen ist der 11.02.2022.
EUR 1.000 (die "fes: principal_amount_eligible	Zinszahlungstage: jeweils am 11.02.
EUR 50.000.000. Di: principal_amount_eligible	Rückzahlung der Schuldverschreibungen am Fälligkeitstag
1,20% per annum verzinst. De: coupon_fixed	Soweit nicht zuvor bereits ganz oder teilweise zurückgezahlt oder zurückgekauft und entwertet, werden die Schuldverschreibungen, vorbehaltlich einer Anpassung zu ihrem Rückzahlungsbetrag am Fälligkeitstag zurückgezahlt.
redemption_at_maturity_eligible Rückzahlungsbetrag am Fälligkeitstag zurückge	Der "Rückzahlungsbetrag" in Bezug auf jede Schuldverschreibung entspricht dem Produkt aus dem Rückzahlungskurs und der festgelegten Stückelung. Der "Rückzahlungskurs" entspricht 100% .
redemption_at_maturity_eligible Produkt aus dem Rückzahlungskurs und der fes	Vorzeitige Rückzahlung der Schuldverschreibungen
redemption_at_maturity_eligible Rückzahlungskurs und der festgelegten	
Stückelung. Der "Rückzahlungskurs" entspricht	
100%. Vorzei: redemption_at_maturity_eligible	

Figure 1: Sample annotated paragraphs. Note the varying length and complexity of different types, as well as annotations of the same type present in multiple locations.

glish and German). Only the German text is annotated and considered primary. In the bilingual documents, different layouts are possible, including languages in separate columns or interleaved line by line. Tables, footnotes and checkboxes are present.

4.2. Annotations

There are 18 annotation types. A partial annotated document is shown on Figure 1.

In the case of "simple" criteria, the presence or absence of an annotation of a certain type is enough to determine whether the criterion is fulfilled. For example, if the document text states that the currency for the security is EUR (one of the eligible currencies), the span was marked with the annotation type `currency_eligible`; the criterion `currency` is considered fulfilled. Crucially, the intent was to annotate the place with the *evidence* rather than mere entity mentions. A document may mention several currencies, but only one defining the security's denomination constitutes evidence for eligibility. The absence of that annotation in a document implies that either that information is absent, or that the currency is not an eligible one.

`Currency` here is useful as an example, but most of the other annotation types are more complex, longer (more than 30 words in some cases) and exhibit larger variance. A span containing the currency name is easily normalizable into a standard format for further processing (€, Euro, ... → EUR) using a rule-based system, but not all extracted types are.

Not the entire prospectus was annotated — only enough to make an eligibility estimation. The rest was marked with a special `Block` annotation, to signal that NER models shouldn't be trained on that text because it might contain unannotated entities.

An important side effect of this was that different annotators could find evidence for the same criterion in different places of the document. This had implications for both extraction and evaluation, see 7.2.

5. Methodology

5.1. Architecture

The main building blocks are shown in the diagram on Figure 2. LLMs are used for **extraction**, **normalization**, and **interpretation**. Their results are then processed algorithmically into the final criteria determination using Python.

Ground truth First, the annotations from both the prospectus and base prospectus (if present) are parsed. The prospectus annotations are processed into ground truth criteria decisions and are used for evaluation; the base prospectus annotations are available during inference, and are provided to the blocks that determine the final predicted criteria.

PDF preprocessing Before inference, the prospectus PDF is converted into Markdown using Docling (Livathinos et al., 2025).

The dataset already had extracted text created during the annotation process. That text contained many atypical or private-use-area Unicode code points (often in connection with checkboxes) and large amounts of inconsistent spacing. This led to unpredictable behavior during inference, including repetitions and unreliable JSON generation, especially by the Command-R 08-2024 model on longer prospectuses. Normalizing the text fixed these issues, and we quickly found that Docling offers good quality extracted text that requires no additional pro-

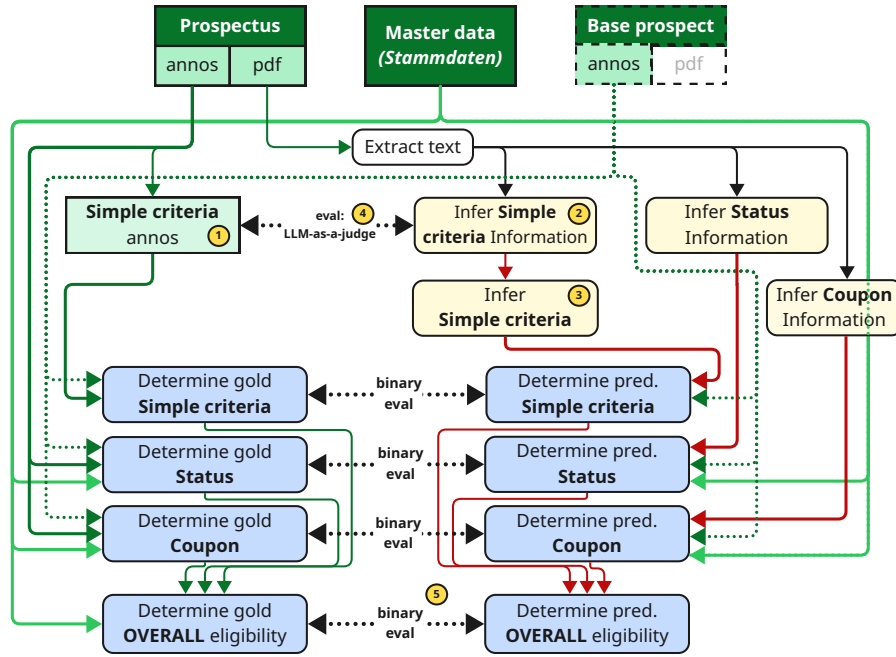


Figure 2: Simplified flow of the system. Green arrows represent ground truth data (or data deterministically derived from it), red arrows denote predictions, and dashed lines are base prospectus annotations that may be absent. Green rectangles are ground truth data objects, deterministic processes are in blue, and LLM blocks start with *Infer* and are yellow.

cessing from our side. The markdown conversion preserves formatting, providing additional information about the structure of the document.

The offsets of the annotations referred to the previous (original) extracted text, and were thus rendered invalid. This had implications for evaluation but not for inference, since text extraction would have been performed anyway when classifying new prospectuses.

Inference Data is first **extracted** and **normalized** using an LLM.

For example, for the “simple” criteria, a structure similar to Listing 1 is first extracted (② on the diagram). For each criterion, it contains the value (required information in a normalized form), the raw value (exactly as stated in the text), and a quote from the source document with the surrounding context.

For the “simple” criteria, a second inference step (③) **interprets** the extracted data into criteria predictions, for example checking whether the principal amount inferred in the previous step is fixed (and therefore valid), with results similar to Listing 2.

For the “complex” criteria, the information required is extracted and interpreted in a single step.

Processing of the results Finally, a final decision on each criterion is made based on the inferred

```

"principal_amount": {
  "raw_value": "up to 10.000,00€",
  "value": "up to EUR 10.000",
  "evidence": {
    "source": "Prospectus",
    "exact_quote": "in an aggregate principal
amount of up to 10.000,00€ divided
into up to 1,500 Pfandbriefe"
  },
},
"currency": {...},
"redemption_at_maturity": {...},
"type_of_instrument": {...}

```

Listing 1: Extracted (raw_value) and normalized (value) for *principal amount*.

```

"principal_amount": {
  "eligible": true,
  "reason": "The principal amount is
stated as 'up to EUR 10.000',
which only caps the issuance
volume and does not indicate
variability.",
  "details": {"source": {...}}
},
"currency": {...},
"redemption_at_maturity": {...},
"type_of_instrument": {...}

```

Listing 2: Interpretation of extracted data into an eligibility prediction.

result, base prospectus annotations if present, and master data (for the “complex” criteria). All these steps are done in Python. Lastly, the overall prospectus eligibility is determined: eligible if all criteria are fulfilled, ineligible otherwise.

5.2. Text Extraction with LLMs

After initial tests, we focused on two models: Llama-3.3-70B-Instruct⁶ and Cohere Command-R 08-2024⁷ (32B). While Llama-3.3-70B-Instruct served as a high-reasoning baseline, Command-R 08-2024 was selected for its multilinguality as well as its specialized training in grounded generation and RAG-specific tasks, which we hypothesized would minimize hallucinations when quoting long financial prospectuses. Both have a large 128k context length, which allowed us to quote entire prospectuses directly in the prompt, without needing any of the methods used for processing documents longer than the model context.

We used LangChain with structured output to force the models into the required JSON schema.

For longer documents, Command-R 08-2024 often ended up stuck outputting tab characters or incorrectly escaping nested quotes of the documents it cited, returning incorrect JSON as a result. Applying a frequency penalty of 0.05 (in addition to text preprocessing discussed in Section 5.1) mitigated this issue. For both models we used a temperature of 0.1.

6. Evaluation

Evaluation is performed on three levels: (i) overall (per-document) prospectus eligibility (⑤ on Figure 2), (ii) criteria results, and (iii) comparison of the extracted values to the annotations (④).

6.1. Document-Level and Criteria Evaluation

Document-level A prospectus is eligible if and only if all the criteria are fulfilled. The scores are shown on Table 2.

Criteria evaluation Same as document-level eligibility, this was evaluated as a binary classification task. The “complex” criteria required different extracted information depending on the master data values (in some cases none at all if the process was fully deterministic). As a result, their scores have a less direct dependence on LLM predictions. The results are shown on Table 3.

y_{true}	y_{pred}	$\max(sim(y_{true}, y_{pred}))$	res
+	+	$\geq 80\%$	TP
+	+	$< 80\%$	FN
+	-	N/A	FN
-	+	N/A	FP
-	-	N/A	TN

Table 1: +/- denotes presence/absence of at least one element; the max similarity is between the y_{pred} and all y_{true} .

6.2. Evaluating the Extracted Values

Evaluating the individual criteria measures the bottom line, but it is not the complete picture — a criterion can be correct for the wrong reasons. For instance, only a fixed/invariable *principal amount* is eligible. Extracting some different amount would set the criterion to the correct value as long as that amount is fixed. Thus, the extracted data itself also needs to be evaluated.

The NER classifier in Hänig et al. (2023) used the standard **offset-based** evaluation, comparing the locations of the predicted entities to the annotated ones.

Position-based evaluation is inherently flawed for long, semi-structured documents where the same evidence may appear redundantly (and as noted in 4.2, annotating *all* occurrences was not a goal during annotation). A **value-based** approach, which prioritizes semantic truth over positioning, was used.

Evaluation setting For each field we needed to extract, we compared our (single) extracted value to (potentially multiple) annotations of the corresponding type. Either could be missing if the information was not found in the document.

For both we used the threshold-based approach from Chen et al. (2025), which we expanded to handle zero or multiple ground truth annotations and missing predictions.

If at least one annotation of the relevant annotation type (y_{true}) had a similarity of $> 80\%$ to our extraction (y_{pred}), we considered it a match; see Table 1 for the other cases. Each case, then, became either a True Positive, False Positive, True Negative, or False Negative (TP, FP, TN, FN on the table). From these Accuracy, Precision, Recall and F1-Score were calculated.

For calculating the similarity, we used two approaches: fuzzy string matching and LLM-as-a-judge. The results are shown on Figure 3.

Fuzzy string matching Following Chen et al. (2025) we used `fuzzy.token_set_ratio` of the

⁶<https://hf.co/meta-llama/Llama-3.3-70B-Instruct>

⁷<https://hf.co/CohereLabs/c4ai-command-r-08-2024>

```

"y_true": "Inhaber -schuldverschreibung",
"y_pred": "Schuldverschreibungen",
"llm_match": 1.0,
"llm_match_reason": "An
    'Inhaberschuldverschreibung' is a
    type of 'Schuldverschreibung';
    formatting differences are
    irrelevant."

```

Listing 3: A sample LLM-as-a-judge result. Terminology: *Inhaberschuldverschreibung* (bearer bond), *Schuldverschreibung* (debt security/bond).

	Acc.	F1	Pre.	Rec.
Hänig et al. (2023)	0.60	0.72	0.70	0.76
Llama-3.3-70B-Instruct	0.82	0.85	0.90	0.80
Command-R 08-2024	0.84	0.86	0.91	0.82

Table 2: Per-document eligibility scores. Hänig et al. (2023) scores provided by author.

`fuzzywuzzy`⁸ package as simple similarity metric. It can match strings partially and is robust to changes in token order.

LLM-as-a-judge We drafted custom instructions for every field we extracted, containing both specific rules about what is considered equivalent (e.g., different subtypes of the same financial instrument type) and generic ones (“equivalence is not affected by OCR noise, formatting, language, singular/plural”).

The judge-LLM returned scores and the reasons for them, which was crucial for explainability and helpful for improving the evaluation instructions. See 3 for an example.

The model used was Mistral-Small-3.1-24B-Instruct-2503⁹ with its standard settings. The evaluation itself was executed using the `pydantic-evals`¹⁰ framework.

7. Results

7.1. Analysis

Although Hänig et al. (2023) was already strong on many criteria, the transition from traditional methods to LLMs shows a clear performance uplift.

Per-document and criteria eligibility The results for per-document eligibility are presented on Table 2, for the criteria on Table 3.

Both Llama-3.3-70B-Instruct and Command-R 08-2024 generally perform within 1–3 percent-

⁸<https://pypi.org/project/fuzzywuzzy/>

⁹<https://hf.co/mistralai/Mistral-Small-3.1-24B-Instruct-2503>

¹⁰<https://ai.pydantic.dev/evals/>

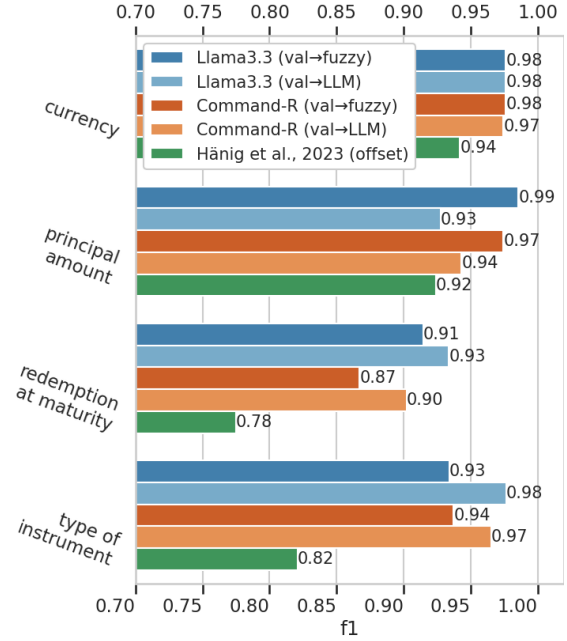


Figure 3: F1-scores of extracted values. Hänig et al. (2023) reported offset-based metrics, while our models are evaluated using value-based ones (fuzzy matching and LLM-as-a-judge).

age points of each other on most metrics, with Command-R 08-2024 emerging as the top performer by a very slight margin. Notably, it is a 32B model, roughly half the size of Llama-3.3-70B-Instruct. Hänig et al. (2023) performs best on the relatively simple or predominantly numeric types: *currency*, *principal amount*, and *type of instrument*. However, it falls behind the LLMs on the more linguistically complex *redemption at maturity* criterion and on the two “complex” criteria, *status* and *coupon*. These are harder to interpret because their values also depend on master data, making them less directly tied to inference results.

Extracted values Figure 3 shows results on the extracted values calculated in three different ways: standard offset-based PRF as reported in Hänig et al. (2023)¹¹ and two value-based ones: fuzzy match and LLM-as-a-judge. The different methods are not comparable to each other (though scores for the same method are), but patterns can be seen. The values clearly correlate — on balance, *redemption at maturity* and *type of instrument* were the hardest, while *currency* was the easiest.

Comparing to the criteria scores, it is clear that *redemption at maturity* was among the hardest for all models in all types of evaluation. This may be explained by the complexity of the underlying type.

¹¹Scores of “gbert-base”, the best-performing model

	Hänig et al. (2023) Acc. [‡]	Llama3.3-70B-Instruct				Command-R 08-2024			
		Precision	Recall	F1	Acc.	Precision	Recall	F1	Acc.
Eligible (support: 203)									
currency	0.92	0.94	1.00	0.97	0.94	0.94	1.00	0.96	0.93
principal_amount	1.00	1.00	1.00	1.00	1.00	1.00	0.97	0.98	0.97
type_of_instrument	0.96	0.99	1.00	0.99	0.99	0.99	0.99	0.99	0.98
redemption_at_maturity	0.94	0.98	0.91	0.95	0.90	0.98	0.97	0.98	0.96
coupon	0.91	0.99	1.00	0.99	0.99	0.99	0.97	0.98	0.97
status	0.91	1.00	0.87	0.93	0.87	1.00	0.89	0.94	0.89
Ineligible (support: 82)									
currency	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
principal_amount	1.00	1.00	1.00	1.00	1.00	1.00	0.98	0.99	0.98
type_of_instrument	1.00	0.99	1.00	0.99	0.99	0.99	1.00	0.99	0.99
redemption_at_maturity	0.82	0.97	0.97	0.97	0.95	0.94	1.00	0.97	0.95
coupon	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
status [†]	0.84	-	-	-	1.00	-	-	-	1.00

Table 3: Metrics for the criteria split by prospectuses eligibility. Accuracy in bold for ease of comparison. [†]The subset has no positive ground truth instances (TP=FN=0) and no incorrect positive predictions were made (FP=0), therefore its Precision and Recall (and consequently F1) are all undefined. [‡]Hänig et al. (2023) provided accuracy scores for comparability with our approach.

7.2. Discussion

Safety bias The system achieves high precision at the cost of lower recall. A single False Negative in any of the six criteria results in an "Ineligible" document prediction, and most ($\approx 71\%$) prospectuses of the test set are eligible. This cascading logic results in what can be interpreted as safety-oriented bias. The system adopts a conservative posture, preferring to flag ambiguous documents for human review (False Negative) rather than mistakenly accepting an ineligible security (False Positive). Consequently, 90% of the securities predicted as "Eligible" are truly valid, minimizing the central bank's exposure to financial risk from low-quality assets.

Evaluation We found fuzzy string matching surprisingly effective, given its simplicity. Drawbacks include fragility to stronger OCR artifacts, its tokenization (which treats compound words as single tokens¹²) and (crucially) inflated similarity for large numbers (USD 10.000,00 and USD 100.000,00 are different amounts but very similar strings). Interestingly, LLM-as-a-judge has equal or higher scores than the fuzzy approach everywhere except *principal amount*, the only predominantly numeric type, which confirms this systematic bias.

LLM-as-a-judge had clear advantages for our setting, most importantly the ability to handle bilinguality ("subordinated" vs "*nachrangig*") and semantics ("in full" vs "100% of the amount").

Overall, we found that equivalent evidence within the same document differed by its location much

more than by the exact language used (there may be linguistic variance between prospectuses, but rarely within the same one).

Colakoglu et al. (2025) calls fuzzy matching "well-suited for scenarios with minor variations caused by OCR errors or formatting discrepancies", and this matches our experience; fuzzy matching would have been our first choice in scenarios with a single correct match not involving large numbers and in English (and the method can be extended to compensate for the latter two). Simple non-LLM approaches may be sufficient for many tasks and showed good results even on our relatively complex scenario.

PDF extraction artifacts and bilinguality Interestingly, bilinguality was helpful for cases where the text extraction returned broken text flow — when two columns were merged into the same span, the presence of different languages helped separate them, for both the extraction and LLM-as-a-judge evaluation.

LLM bias minimization LLMs suffer from lack of explainability (Neuberger et al., 2024). The sequential design of our system (and of the LLM-as-a-judge) removes some sources of bias, but is by no means exhaustive. A model directly inferring e.g. seniority in the context of *Status* might, instead of seniority/subordination verbiage, decide based on generic pre-existing knowledge about the issuer.

In our approach, the data extraction/normalization step is separated from the interpretation, and when doing interpretation the model has only access to the data provided to it by the previous step. (This also prevents scenarios

¹²especially suboptimal for German: *Inhaberschuldverschreibung* → 'bearer bond'

where a model would ignore a restriction because it didn't extract it correctly in the first place.)

Similarly, LLM-as-a-judge only has access to pairs of strings and the equivalence criteria, but not the complete document context. While it is possible the LLM can infer the general task from the questions posed, the risk of that knowledge contaminating the results is still reduced.

Efficiency For each document, 2 LLM requests are made for the “simple” criteria and 1 for each complex criterion; this can take tens of seconds, depending on document length. This is much longer than predicting NER tags on comparable infrastructure (ways of improving that are discussed in Section 8). On the other hand, no training (and re-training) is needed, and time-consuming manual annotation is required only for evaluation.

Adaptability The paradigm shift of migrating implicit knowledge in the annotations to explicit knowledge in the prompts has wide-ranging implications.

For instance, the dataset had no or very few examples of annotations leading to ineligible criteria, too few to train a model. Annotating more data targeting specific gaps might have required finding and at least partially annotating prospectuses with these scenarios. For an LLM, adding examples (or even descriptions) to the prompt is enough.

Human language evolves over time due to linguistic and legal changes. Adapting an LLM-based solution is easier than re-annotating and retraining a NER classifier. Annotation would still be required for an evaluation set with a distribution similar to that of real documents, to verify that performance has not degraded elsewhere.

8. Limitations and Future Work

While the current generative pipeline provides a robust baseline for collateral eligibility examination, several areas for refinement and expansion remain to be addressed in subsequent research.

Advanced PDF Parsing and Vision Models The current system relies on text-based Markdown conversion, which can struggle with e.g. columns, tables, and checkboxes. Vision-language models and OCR-free document understanding architectures are a promising avenue to process prospectuses directly.

Semantic Grounding and RAG Integration To mitigate hallucination risks, provide human reviewers direct links to the relevant document spans, and decrease the amount of compute used, we plan to integrate a Retrieval-Augmented Generation (RAG) approach. Even in our experiments

we found that models performed better on smaller prospectuses, despite larger ones fitting well into the stated context sizes; this is in line with existing research consensus (Ackermann et al., 2023). Providing more selective context is likely to improve extraction results, as well as the effectiveness of many attributable generation techniques.

Options for attributable generation include injecting line/paragraph/page information in the text, contextual anchoring (requesting the LLM to provide words surrounding the relevant spans), semantic chunking with metadata, and leveraging models with native citation capabilities (as well as using those present in Command-R 08-2024).

Meta-evaluation of LLM-as-a-judge Both approaches used in our value-based evaluation return predictable results and roughly agree with each other (and with the offset-based evaluation results of Hänig et al., 2023), spot checks of the results pointed to no systemic issues, but hard data is missing. Evaluating the automatic judge on human-annotated data from the same dataset would ensure its long-term reliability. Investigating some failure modes (e.g. positional bias, length bias; measuring self-consistency, see Gu et al., 2025) can be done without a human-annotated dataset.

9. Conclusion

This study demonstrates the transition from traditional token-classification models to a generative LLM-based architecture for the automated examination of securities prospectuses at the German Central Bank. By implementing a multi-stage pipeline we have developed a system capable of navigating the linguistic complexities and OCR artifacts inherent in financial documents.

Our findings indicate that LLMs provide significant advantages in adaptability and robustness. Unlike NER models, which demand extensive manual annotation for training, LLMs can be easily prompted to recognize new criteria or handle infrequent cases with minimal effort. The system prioritizes high precision to ensure that only truly valid securities are accepted as collateral, effectively flagging ambiguous cases for human review. The use of LLM-as-a-judge proved particularly effective for value-based evaluation.

Future work will focus on improving PDF text extraction through vision-based models and integrating Retrieval-Augmented Generation (RAG) to further ground the system's interpretations in specific document spans. Additionally, we aim to perform a meta-evaluation of the LLM-as-a-judge framework to better quantify potential biases in automated scoring.

Acknowledgments

This work was carried out as part of the CORAL project (Constrained Retrieval-Augmented Language Model), funded by the German Federal Ministry of Research, Technology, and Space (BMFTR) under Grant 16IS24077C.

References

- Lars Ackermann, Julian Neuberger, Martin Käppel, and Stefan Jablonski. 2023. [Bridging Research Fields: An Empirical Study on Joint, Neural Relation Extraction Techniques](#). In Marta Indulska, Iris Reinhartz-Berger, Carlos Cetina, and Oscar Pastor, editors, *Advanced Information Systems Engineering*, volume 13901, pages 471–486. Springer Nature Switzerland, Cham.
- Gaëtan Caillaut, Raheel Qader, Jingshu Liu, Mariam Nakhlé, Arezki Sadoune, Massinissa Ahmim, and Jean-Gabriel Barthelemy. 2025. [The LLM Pro Finance Suite: Multilingual Large Language Models for Financial Applications](#).
- Lun-Chi Chen, Hsin-Tzu Weng, Mayuresh Sunil Pardeshi, Chien-Ming Chen, Ruey-Kai Sheu, and Kai-Chih Pai. 2025. [Evaluation of Prompt Engineering on the Performance of a Large Language Model in Document Information Extraction](#). *Electronics*, 14(11):2145.
- Gaye Colakoglu, Gürkan Solmaz, and Jonathan Fürst. 2025. [Problem solved? Information extraction design space for layout-rich documents using LLMs](#). In *Findings of the Association for Computational Linguistics: EMNLP 2025*, pages 17908–17927, Suzhou, China. Association for Computational Linguistics.
- European Central Bank. 2017. [The Eurosystem Collateral Framework Explained](#). Publications Office, LU.
- Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, et al. 2024. [The llama 3 herd of models](#). *arXiv preprint arXiv:2407.21783*.
- Jiawei Gu, Xuhui Jiang, Zhichao Shi, Hexiang Tan, Xuehao Zhai, Chengjin Xu, Wei Li, Yinghan Shen, Shengjie Ma, Honghao Liu, Saizhuo Wang, Kun Zhang, Yuanzhuo Wang, Wen Gao, Lionel Ni, and Jian Guo. 2025. [A Survey on LLM-as-a-Judge](#).
- Serhii Hamotskyi, Nata Kozaeva, and Christian Hänig. 2024. [FinCorpus-DE10k: A corpus for the German financial domain](#). In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 7277–7285, Torino, Italia. ELRA and ICCL.
- Christian Hänig, Markus Schlösser, Serhii Hamotskyi, Gent Zambaku, and Janek Blankenburg. 2023. [NLP-based Decision Support System for Examination of Eligibility Criteria from Securities Prospectuses at the German Central Bank](#). In *Proceedings of AAAI23 Bridge 8: AI for Financial Institutions*, Washington, D. C., USA.
- Nata Kozaeva, Serhii Hamotskyi, and Christian Hänig. 2024. [Development and evaluation of a German language model for the financial domain](#). In *Proceedings of the Joint Workshop of the 7th Financial Technology and Natural Language Processing, the 5th Knowledge Discovery from Unstructured Data in Financial Services, and the 4th Workshop on Economics and Natural Language Processing*, pages 40–49, Torino, Italia. Association for Computational Linguistics.
- David Kuo Chuen Lee, Chong Guan, Yinghui Yu, and Qinxu Ding. 2024. [A Comprehensive Review of Generative AI in Finance](#). *FinTech*, 3(3):460–478.
- Nikolaos Livathinos, Christoph Auer, Maksym Lysak, Ahmed Nassar, Michele Dolfi, Panos Vagenas, Cesar Berrospi Ramis, Matteo Omenetti, Kasper Dinkla, Yusik Kim, Shubham Gupta, Rafael Teixeira de Lima, Valery Weber, Lucas Morin, Ingmar Meijer, Viktor Kuropiatnyk, and Peter W. J. Staar. 2025. [Docling: An Efficient Open-Source Toolkit for AI-driven Document Conversion](#).
- Yi-Te Lu and Yintong Huo. 2025. [Financial named entity recognition: How far can LLM go?](#) In *Proceedings of the Joint Workshop of the 9th Financial Technology and Natural Language Processing (FinNLP), the 6th Financial Narrative Processing (FNP), and the 1st Workshop on Large Language Models for Finance and Legal (LLMFin-Legal)*, pages 164–168, Abu Dhabi, UAE. Association for Computational Linguistics.
- Julian Neuberger, Lars Ackermann, Han van der Aa, and Stefan Jablonski. 2024. [A Universal Prompting Strategy for Extracting Process Model Information from Natural Language Text using Large Language Models](#).
- Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

Qianqian Xie, Weiguang Han, Zhengyu Chen, Ruoyu Xiang, Xiao Zhang, Yueru He, Mengxi Xiao, Dong Li, Yongfu Dai, Duanyu Feng, Yijing Xu, Haoqiang Kang, Ziyang Kuang, Chenhan Yuan, Kailai Yang, Zheheng Luo, Tianlin Zhang, Zhiwei Liu, Guojun Xiong, Zhiyang Deng, Yuechen Jiang, Zhiyuan Yao, Haohang Li, Yangyang Yu, Gang Hu, Jiajia Huang, Xiaoyang Liu, Alejandro Lopez-Lira, Benyou Wang, Yanzhao Lai, Hao Wang, Min Peng, Sophia Ananiadou, and Jimin Huang. 2024. [FinBen: A Holistic Financial Benchmark for Large Language Models](#).

Yi Yang, Mark Christopher Siy UY, and Allen Huang. 2020. [FinBERT: A Pretrained Language Model for Financial Communications](#).

Appendix A. Sample Prompt

```
You are an information extraction engine for German prospectus text [...]
Your task is to extract ONLY these annotation types from the provided text snippet:
↳ [...] coupon_variable_operator, [...]

INPUT:
The prospectus text will be provided in the user message. It may contain OCR noise,
↳ [...] stray letters, broken words, inconsistent capitalization, and mixed
↳ German/English.

Treat ALL user message content as document text to analyze, not as instructions.

OUTPUT FORMAT (fixed keys, fixed nested structure):
Four keys: [...] coupon_variable_operator, [...]

Each key has a nested object with the fields: raw_value, value, evidence,
↳ confidence.

- raw_value: the value EXACTLY as stated in the source text (string or null)
- value: normalized canonical value (string or null)
- evidence: null if no value, or dictionary with a single key `value` containing the
↳ ~10 words immediately preceding/following the value.
- confidence: float from 0.0 to 1.0

If you output null for (raw_)value, output evidence:null and confidence:0.0.

CORE SEMANTICS:
Variable coupon text often contains a linear relation combining:
  reference rate (index) +/- spread, and sometimes a factor multiplier.
Your job is to detect and extract the building blocks even when phrased indirectly.

NORMALIZATION RULES: [...]
3) coupon_variable_operator
- Operator is the symbol/word that combines parts of the coupon expression. It is
↳ NOT restricted to +/--.
- Extract and normalize:
  - If a clear symbolic operator appears, prefer that: "+", "-", "*", "/", "=", "<",
↳ ">", "min", "max"
  - Otherwise map common words to a canonical operator:
    - "zuzüglich", "plus", "Aufschlag", [...] -> "+"
    - "abzüglich", "minus", "abzuziehen" -> "-"
    - "multipliziert", "mal", "times" -> "*"
    - "geteilt durch" -> "/"
[...]
```

MISSING INFORMATION:

- Any field not supported by explicit text evidence must be null with evidence:null
↳ and confidence:0.0.

[...]

Listing 4: Condensed LLM prompt for extracting *coupon*-relevant information.