Information Extraction from Conversation Transcripts: Neuro-Symbolic vs. LLM

Alice Saebom Kwak[1], Maria Alexeeva[2], Gus Hahn-Powell[2], Keith Alcock[2], Kevin McLaughlin[2], Doug McCorkle[3], Gabe McNunn[3], Mihai Surdeanu[2]

[1] Department of Linguistics, University of Arizona
[2] Lum AI
[3] Eocene Environmental Group

{alicekwak0520, maxaalexeeva}@gmail.com, {ghp, keith, kevin, mihai}@lum.ai,{dmccorkle, gmcnunn}@eocene.com

Abstract

The current trend in information extraction (IE) is to rely extensively on large language models, effectively discarding decades of experience in building symbolic or statistical IE systems. This paper compares a neuro-symbolic (NS) and an LLM-based IE system in the agricultural domain, evaluating them on nine interviews across pork, dairy, and crop subdomains. The LLM-based system outperforms the NS one (F1 total: 69.4 vs. 52.7; core: 63.0 vs. 47.2), where total includes all extracted information and core focuses on essential details. However, each system has trade-offs: the NS approach offers faster runtime, greater control, and high accuracy in context-free tasks but lacks generalizability, struggles with contextual nuances, and requires significant resources to develop and maintain. The LLM-based system achieves higher performance, faster deployment, and easier maintenance but has slower runtime, limited control, model dependency and hallucination risks. Our findings highlight the "hidden cost" of deploying NLP systems in real-world applications, emphasizing the need to balance performance, efficiency, and control.

1 Introduction

Information extraction (IE) from written text is a well-established task, with various systems successfully extracting entities, relations, and events from unstructured natural language text (Sundheim, 1991). Recently, IE approaches have increasingly relied on large language models (LLMs) (see Xu et al. (2024) for a comprehensive survey). While this shift has led to notable performance improvements, it comes at a cost: (i) low explainability (Danilevsky et al., 2021), (ii) fragility (Sculley et al., 2015), and (iii) lack of control over its behavior (Vacareanu et al., 2024). These challenges contribute to the "hidden technical debt" of deploying natural language processing (NLP) systems in real-world industrial settings. In contrast, rule-based methods are explainable and provide greater control, but lack the generalization power of current deep learning systems (Tang and Surdeanu, 2023).

This work investigates whether the performance boost of LLMs justifies the hidden costs in the real world. We compare two radically different IE strategies—an LLM-based system and an equivalent neuro-symbolic (NS) implementation—in an industry application extracting information from agricultural conversation transcripts. These interviews, conducted between farmers (interviewees) and interviewers, gather details about U.S. farms (e.g., number of animals raised, crop types). Our use case, while highly specialized, can be generalized to other domains like customer support systems. To encourage further research in this underexplored setting, we release our dataset: `https://anonymous.4open.science/r/ag_dataset-8E0B/`

Our comparison highlights key trade-offs: the NS system is efficient, offers greater control, and is precise in context-free tasks but lacks generalizability, struggles with contextual nuances, and requires extensive resources for development and maintenance. The LLM-based system offers higher performance, faster deployment, and lower maintenance costs but is slower, provides less control, is model dependent, and is prone to hallucinations. We hope that this discussion will influence the design of future IE systems that are deployed and maintained in the real world.

2 Related Works

Information Extraction (IE) is a key task in NLP that has been investigated with various approaches, including rule-based (Hobbs et al., 1993; Appelt and Onyshkevych, 1998; and Cunningham et al., 2002), neural (Socher et al., 2012; Zeng et al., 2014, Chen et al., 2015; Nguyen and Grishman, 2015; and Zeng et al., 2015), and LLM-based methods (Josifoski et al., 2022; Ma et al., 2023; Wadhwa

et al., 2023; Wadhwa et al., 2023; Perot et al., 2024; Hu et al., 2025). Several studies also combine or compare different approaches (Gardner et al., 2013; Stutz et al., 2015; Liu et al., 2020; Imaichi et al., 2013; and Wang et al., 2024; See appendix A for full discussion).

Our work builds on these directions in two distinct ways. First, it focuses on a dialogue-based IE task within the agricultural domain, using interview transcripts that are long, noisy, and contextually rich. These characteristics introduce new challenges for both neuro-symbolic and LLM-based systems. Rules often fail to account for conversational variation, while LLMs can struggle with long input windows. The presence of domain-specific terminology further complicates both automatic speech recognition (ASR) and information extraction. Second, we compare neuro-symbolic and LLM-based systems in an industrial setting. While most prior work focuses on academic benchmarks, we evaluate the approaches in terms of deployment feasibility, robustness, and efficiency, addressing the gap between research and real-world application identified by Chiticariu et al. (2013).

3 The task

We extract structured information from semi-scripted interviews between farmers and representatives of an organization[1] that supports sustainability efforts by tracking farm data (e.g., farm facilities, acreage, crop types, etc., see Figure 1). Interviewers collect information through telephone or video calls loosely following a script tailored to the farm type (e.g., pork, dairy, crop).

Collected information includes *identifiers* such as 'barn capacity' (see Figure 1), 'manure storage', and 'renewable energy'. Identifiers serve as variables with different value types: quantitative (e.g., number of animals per barn), categorical (e.g., lagoons vs. pits for manure storage), or boolean (e.g., use of renewable energy). The project's goal is to expedite data collection by automatically extracting variable-value pairs from interview transcripts, reducing reliance on manual note-taking.

4 Approach

Our system (Figure 2) aims to link target variables in interviewer questions to values in farm representatives' responses. To normalize extractions, values are mapped to an ontology (e.g., 'barn capacity' is equivalent to 'capacity of those barns' and can be linked to the same identifier). Both NS and LLM-based versions share ASR and preprocessing but diverge in extraction and ontology grounding.

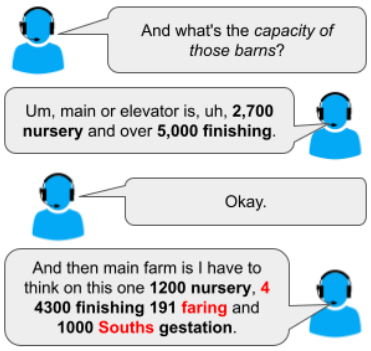

Figure 1: Sample conversation transcript simplified for readability, presented as produced by the ASR system before post-processing. The target identifier to be extracted is in italics. The target values for the identifier are in bold. The ASR errors are in red: the first *4* is spurious, *faring* must be corrected to *farrowing*, and *Souths* to *sows*.'

4.1 The shared components

4.1.1 Automatic speech recognition (ASR)

Interviews are conducted and recorded via a video-conferencing platform. We convert the audio recordings to text using an ASR service provided by rev.ai[2], chosen for its favorable user agreement, including data privacy protections. The ASR service converts the audio to text, providing timestamps and marking speaker turns. Additionally, some atmospheric tags (e.g., <laugh>, <affirmative>) are embedded within the text.

4.1.2 Preprocessing

Transcript correction: Since the ASR model is not domain-specific, its output requires post-processing. This includes remapping frequently mistranscribed domain-specific terms (e.g., *South* to *sows* in Figure 1) and normalizing spelling in unambiguous cases (e.g., adding hyphens in *farrow to finish* and *18 46 0*, which refers to the fertilizer *18-46-0*). This normalization helps our NS extraction system. We also remove filler words (e.g., *um*).

Topic segmentation: In the agriculture domain, we work with several subdomains (dairy, crops, and pork), each associated with an identifier ontology that may overlap. To improve accuracy, we segment interviews using interviewer section markers (e.g., "**Section one** is about dairy"), which indicate

[1] Anonymized for review

[2] https://rev.ai

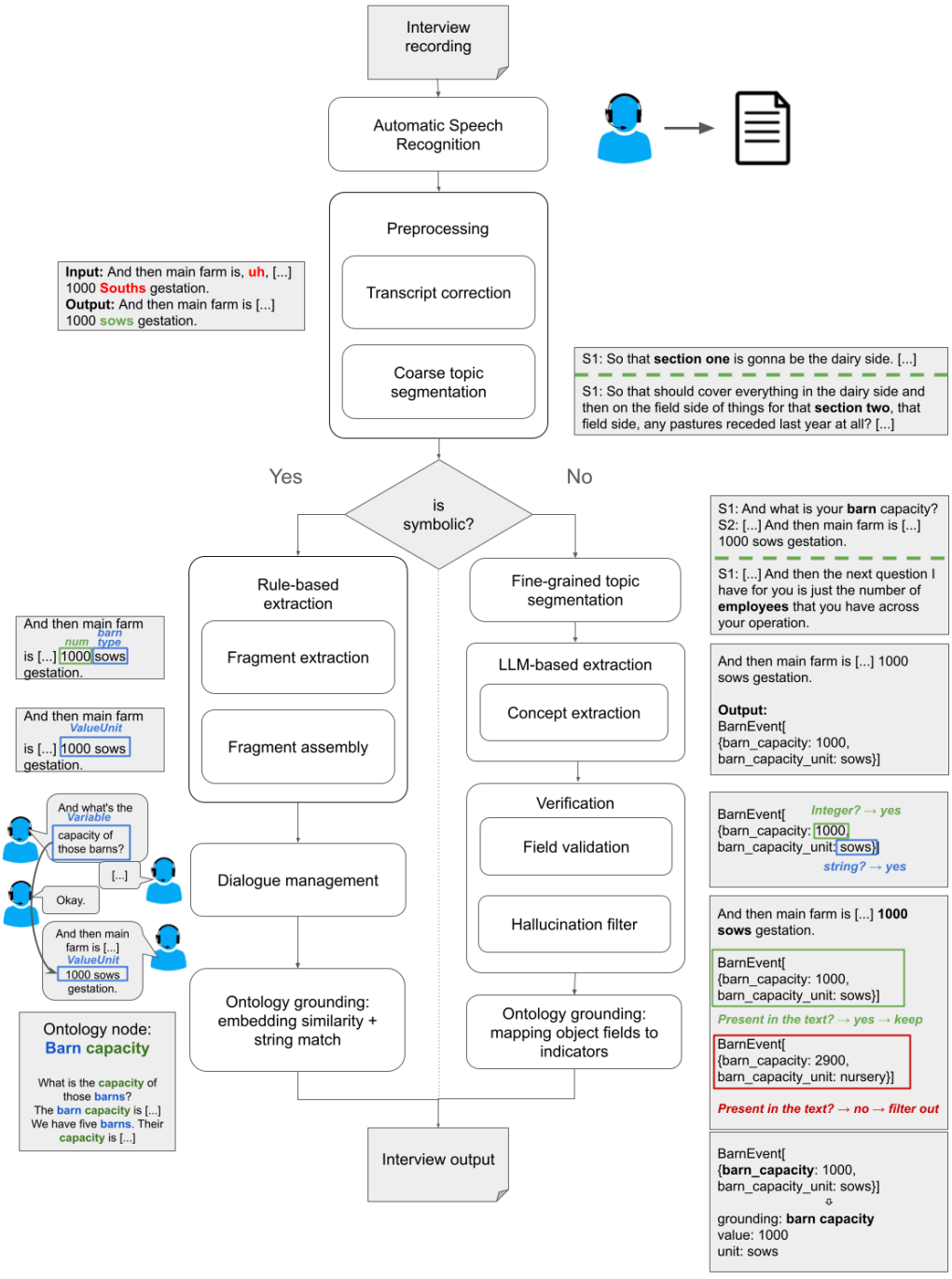

Figure 2: Generalized pipeline of the two system versions. Both share ASR and preprocessing but diverge afterward. The NS system uses a rule-based approach to extract and assemble identifier-value pairs, with dialogue management linking distant fragments. Grounding relies on embedding similarity and string matching. The LLM-based system segments interviews by topic, extracts information via in-context learning, and verifies results through field validation and hallucination filtering before mapping fields to relevant indicators.

topic transitions. The system then assigns each segment to its relevant domain.

4.2 Neuro-symbolic system

The neuro-symbolic system starts with an encoder-based backbone. In particular, we implemented a multi-task learning framework, in which an encoder is shared between multiple low-level NLP tasks: part-of-speech tagging, named-entity recognition, and syntactic dependency parsing. Each of these tasks is implemented as a sequence model using a dedicated linear classifier head. To convert dependency parsing into sequence modeling, we used the algorithm of Vacareanu et al. (2020), which converts each dependency tree into two sequences of: (a) relative positions of the governor

tokens for each token in the sentence, and (b) labels of the dependencies between the each token and its governor.

4.2.1 Rule-based extraction

For symbolic information extraction, we develop a grammar using the Odin information extraction framework (Valenzuela-Escárcega et al., 2015) to write rules over semantic and syntactic features. We first extract standalone entities: identifiers and values (*Fragment extraction* in Figure 2). A sample rule can be found in Appendix B. Extracting identifiers and categorical values relies on domain-specific knowledge bases, which require updates for new subdomains.

We assemble extracted entities into identifier-value relations (*Fragment assembly* in Figure 2). Identifiers and their values are frequently scattered across multiple sentences, requiring complex dialogue management to properly assemble.

4.2.2 Dialogue management

Identifier and values may span multiple speaker turns, with identifiers appearing in questions and values distributed across responses (Figure 1). To address this, we search prior context for a matching identifier when extracting a potential value (e.g., *1200 nursery* $\rightarrow$ *capacity of those barns*).

To improve precision, we use heuristics such as adjusting context window size per identifier and maintaining valid identifier-unit pairings (e.g., *nursery* pairs with *barn capacity* but not *manure storage*). A fallback system assigns unambiguous values when no identifier is available. (e.g., compound values that include barn types such as *10 farrowing* can be unambiguously assigned to *barn capacity*).

4.2.3 Ontology grounding

For ontology grounding, we map each identifier-value pair to an ontology using a combination of vector embedding similarity and text string overlap. This way, the identifiers in Figure 2 (e.g., *capacity of those barns*, *barn capacity*) can be linked to the same ontology node—'barn capacity.'

4.3 LLM approach

4.3.1 Fine-grained topic segmentation

In the LLM-based approach, we segment interviews into smaller blocks based on topic shifts. For example, if the discussion transitions from barn capacity to manure management, we split the interview accordingly. This segmentation reduces token usage and minimizes hallucinations by narrowing the expected information types for each block.

Topic changes are detected using predefined keywords or key phrases. For instance, if 'barn' appears in a speaker's turn, the topic likely shifts to barn capacity or type. A curated keyword list guides this segmentation process. We use a keyword-based approach to give users greater control and facilitate easier maintenance. While statistical block segmentation methods may perform better, we chose this strategy as it gives users the possibility to customize the keyword list as needed to align with their specific requirements.

4.3.2 LLM-based extraction

For LLM-based extraction, we use in-context learning with task instructions and expected output formats. Each target information type has a corresponding Pydantic dataclass (Colvin et al., 2025), which defines detailed extraction guidelines, including content specifications and format requirements. We formulate prompts that incorporate the JSON schema of each target class, guiding the model toward structured generation with constrained decoding to produce valid JSON that matches the dataclass structure. (See Appendices C and D for the full prompt details and a sample dataclass).

We use OpenChat 3.5, employing a quantized version for efficiency[3]. The model was selected after a pilot study comparing quantized versions of LLaMA-3, Mixtral 8x7B, and OpenChat 3.5. Despite being the smallest, OpenChat 3.5 outperformed the others, likely due to its dialogue-specific tuning, making it well-suited for our task. We only considered local models, as proprietary options present privacy risks. The model has a maximum token limit of 8,192, and the temperature was set to 0 to ensure deterministic outputs.

4.3.3 Verification

Once the LLM generates raw output, it undergoes verification. Extracted fields are validated, and if a field has an invalid type, an attempt is made to convert it (e.g., 'true' is converted to True for boolean fields). If conversion is not possible, the value is filtered out. Additionally, to prevent hallucination, extracted strings are checked against the input text. If no overlap is found, the value is considered a hallucination and is discarded.

[3]We used openchat-3.5-0106.Q5_K_M.gguf, available at: https://huggingface.co/TheBloke/openchat-3.5-0106-GGUF

Domain	Precision		Recall		F1	
	Total	Core	Total	Core	Total	Core
Neuro-symbolic System						
Pork	68.5 (58.1–76.9)	62.4 (51.9–70.0)	54.6 (46.2–69.0)	47.9 (40.0–60.9)	60.6 (51.4–72.7)	54.0 (45.2–65.1)
Crop	63.8 (50.0–87.8)	61.3 (50.0–82.1)	42.8 (36.0–46.8)	38.8 (36.0–43.8)	50.7 (41.9–61.0)	46.6 (41.9–50.5)
Dairy	67.6 (50.0–85.7)	**63.2** (50.0–80.0)	36.4 (33.3–41.9)	31.2 (26.3–34.1)	46.8 (40.0–51.7)	41.0 (39.6–43.4)
Average	66.6	62.3	44.6	39.3	52.7	47.2
LLM-based System						
Pork	**73.9** (71.1–77.1)	**69.5** (64.5–75.8)	**65.1** (57.4–75.0)	**61.4** (55.6–71.8)	**68.9** (65.9–74.2)	**64.9** (59.7–70.0)
Crop	**73.3** (61.9–83.3)	**65.1** (54.3–77.8)	**65.7** (62.5–69.7)	**58.5** (56.0–60.0)	**69.0** (63.4–72.1)	**61.1** (56.7–65.1)
Dairy	**70.4** (58.8–82.3)	62.7 (53.5–77.4)	**71.4** (57.6–87.5)	**65.1** (52.9–80.0)	**70.4** (58.2–77.8)	**63.0** (53.2–69.1)
Average	**72.5**	**65.8**	**67.4**	**61.7**	**69.4**	**63.0**

Table 1: Performance comparison of neuro-symbolic and LLM-based systems across domains, reporting total (all extracted information) and core (essential information) scores. LLM outperforms NS system, especially in F1 (69.4 vs. 52.7 total, 63.0 vs. 47.2 core) and recall. Scores include 95% confidence intervals from bootstrap resampling; bold indicates the higher score.

4.3.4 Ontology grounding

After post-processing, all validated fields are mapped onto the ontology. This mapping is guided by a CSV file that links fields to corresponding indicators. For example, the field 'barn_capacity' in the BarnEvent dataclass is mapped to barn capacity in the ontology as specified in the CSV file. Once mapping is complete, the output is transformed into its final format, which includes the grounding, grounding ID, value, and optional unit.

5 Experiments

We compare two versions of the system: an NS version (Section 4.2) and an LLM-based version (Section 4.3). In this section, we describe the experiment settings, the results, and the error analysis.

5.1 Setting

We evaluate the two systems using nine interviews, three per subdomain. While the dataset comprises nine interviews, each is lengthy and information-dense—averaging 28 minutes 27 seconds, 4,700 words[4], and 189 speaker turns—yielding a richly structured, domain-specific corpus. Gold data was manually created, and system outputs were assessed using exact match, reporting precision, recall, and F1 scores.

For a fair comparison, we report two scores: total and core. Total counts all extracted information, treating variants like *wheat* and *winter wheat* as separate true positives without penalizing systems for extracting only one. Core uses stricter criteria, grouping such variants as a single true positive to ensure consistent evaluation. This dual scoring method accounts for differences in extraction approaches and provides a balanced assessment.

[4]See Appendix E for word count statistics per interview.

5.2 Results

The evaluation results are reported in Table 1. On average, the NS system achieves acceptable precision (66.6 for total, 62.3 for core), but lower recall (44.6 for total, 39.3 for core) and subpar F1 scores (52.7 for total, 47.2 for core). In contrast, the LLM-based system performs better across all metrics, with higher precision (72.5 for total, 65.8 for core), substantially better recall (67.4 for total, 61.7 for core), and stronger F1 scores (69.4 for total, 63.0 for core). The gap is particularly pronounced in recall, where the LLM-based system significantly outperforms the NS system across all subdomains. This trend highlights the LLM-based system's advantage in handling the variability and contextual richness of natural dialogue.

To assess the robustness of these results given the limited dataset size, we conducted statistical significance testing using bootstrap resampling (n = 10,000). We consider differences statistically significant when the 95% confidence intervals do not substantially overlap. Based on this criterion, improvements in recall and F1 scores are statistically significant—particularly in the crop and dairy domains, where the performance gaps are most pronounced. In contrast, differences in precision are smaller and not statistically significant, as confidence intervals overlap substantially across all domains. These findings reinforce the strength of the LLM-based approach, especially in extracting information from complex, context-rich interviews where generalization and recall are critical.

6 Discussion

While the LLM-based approach performed better, neither system fully solved the problem (see Appendix F for the challenges). Error analysis showed

that the NS system struggled with missing rules, identifier-value mismatches, and ontology grounding errors. The LLM-based system's black-box nature made error tracing difficult, but common issues included topic segmentation errors and hallucinations (see Appendix G for full discussion). Both approaches have distinct strengths and weaknesses (see Appendix H for the summary).

6.1 Advantages and disadvantages of the neuro-symbolic system

The NS system stands out for its exceptional efficiency and controllability, making it an attractive choice for resource-constrained or real-time applications. As shown in Table 2, it runs dramatically faster than the LLM-based system: 69 seconds on CPU compared to over 22,000 seconds, and still faster when LLM-based system is GPU-accelerated (69 vs. 74 seconds). This performance edge enables rapid iteration and low-latency deployment—key factors in industrial settings.

Equally important is its interpretability. The NS system offers fine-grained control, allowing developers to identify and fix errors systematically via rule adjustments. This transparency is especially valuable in high-stakes domains where reliability and accountability matter. It also performs well when extraction criteria are clear and context-independent, delivering consistent accuracy.

However, these benefits come with clear limitations. Rule-based logic lacks generalizability and often fails when unseen cases fall outside the predefined rules. Our error analysis shows that the NS system consistently underperforms in tasks requiring nuanced context understanding. Moreover, rule development is labor-intensive and demands deep domain expertise, making it difficult to scale or adapt to new domains. Despite promising directions in semi-automated rule learning (Noriega-Atala et al., 2022; Vacareanu et al., 2022a,b), manual effort remains a bottleneck.

6.2 Advantages and disadvantages of the LLM-based system

The LLM-based system offers greater adaptability and higher extraction quality, particularly in complex, context-rich scenarios. Its strength lies in generalization: it performs well with minimal supervision, relying only on task descriptions and a handful of examples. This makes it especially suited to evolving domains where rules are difficult to anticipate or enumerate.

	Neuro-symbolic	LLM (GPU)	LLM (CPU)
Pork	66	66	14878
Crop	74	71	27950
Dairy	68	86	26133
Average	69	74	22987

Table 2: Runtime comparison of neuro-symbolic and LLM-based approaches (in seconds)[5]

Deployment is also streamlined. Unlike the NS system, which requires laborious rule engineering, the LLM-based approach enables agile iteration via simple keyword tweaks or dataclass modifications. This lowers the barrier to entry, reduces long-term maintenance costs, and improves accessibility and scalability across domains.

Nonetheless, these advantages come at a cost. As noted earlier, the system is computationally expensive and significantly slower. More critically, it functions as a black box, offering limited transparency or control. When errors occur, diagnosing and fixing them is non-trivial. Model selection also introduces risk: if the LLM is not well-matched to the task (e.g., not optimized for dialogue), performance degrades. Finally, hallucinations remain an unresolved challenge, posing risks in applications where factual accuracy is essential.

7 Conclusion

This study presents and compares two information extraction systems—an NS system and an LLM-based system—evaluated on agricultural interview transcripts across multiple subdomains. The result indicates that the LLM-based system outperforms the neuro-symbolic system, particularly in terms of recall, although neither approach completely resolves the task.

Overall, each system has distinct advantages and disadvantages. The NS system is efficient, offers greater control, and is highly accurate in context-free tasks. However, it lacks generalizability, struggles with contextual nuances, and demands substantial resources for development and maintenance. In contrast, the LLM-based system offers higher performance, faster deployment, and more cost-effective maintenance but is constrained by slower runtime, limited control, model dependency, and a tendency to hallucinate. We hope this discussion informs future IE system design for real-world deployment and sus-

[5]Experiments were conducted on an Apple M1 chip (8-core CPU, 8GB unified memory) for the CPU setting and Google Colab's NVIDIA A100 (40GB VRAM) for GPU acceleration.

tainability. The dataset accompanying this study is publicly available at: https://anonymous.4open.science/r/ag_dataset-8E0B/

8 Limitations

While this study offers valuable insights, it has limitations. The findings are based on a single domain (agriculture) and a specific, though complex, IE task—informational interviews. Only one NS and one LLM architecture were evaluated. Although the dataset includes just nine interviews, each is lengthy and content-rich, averaging 28 minutes 27 seconds of audio, 4,700 words, and 189 speaker turns. This makes the dataset a dense and meaningful testbed despite the limited number of interviews. Future work should examine broader domains, additional IE tasks, and diverse model architectures with larger datasets to validate and extend these findings.

9 Ethics statement

This study uses data from real-world interviews involving human participants. To protect their privacy and confidentiality, we conducted a thorough manual anonymization process. Each interview was reviewed to ensure that no personally identifiable information is present. Additionally, sensitive numerical details (e.g., farm sizes, production volumes) were replaced with plausible but non-identifiable values to prevent reidentification while preserving the utility of the data. The resulting dataset, fully anonymized and privacy-safe, is made publicly available to support further research: https://anonymous.4open.science/r/ag_dataset-8E0B/

References

Douglas E. Appelt and Boyan Onyshkevych. 1998. The common pattern specification language. In *Proceedings of a Workshop on Held at Baltimore, Maryland: October 13-15, 1998*, TIPSTER '98, page 23–30, USA. Association for Computational Linguistics.

Yubo Chen, Liheng Xu, Kang Liu, Daojian Zeng, and Jun Zhao. 2015. Event extraction via dynamic multi-pooling convolutional neural networks. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 167–176, Beijing, China. Association for Computational Linguistics.

Laura Chiticariu, Yunyao Li, and Frederick R. Reiss. 2013. Rule-based information extraction is dead! long live rule-based information extraction systems! In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 827–832, Seattle, Washington, USA. Association for Computational Linguistics.

Samuel Colvin, Eric Jolibois, Hasan Ramezani, Adrian Garcia Badaracco, Terrence Dorsey, David Montague, Serge Matveenko, Marcelo Trylesinski, Sydney Runkle, David Hewitt, Alex Hall, and Victorien Plot. 2025. Pydantic.

Hamish Cunningham, Diana Maynard, Kalina Bontcheva, and Valentin Tablan. 2002. GATE: an architecture for development of robust HLT applications. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 168–175, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

Marina Danilevsky, Shipi Dhanorkar, Yunyao Li, Lucian Popa, Kun Qian, and Anbang Xu. 2021. Explainability for natural language processing. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, KDD '21, page 4033–4034, New York, NY, USA. Association for Computing Machinery.

Matt Gardner, Partha Pratim Talukdar, Bryan Kisiel, and Tom Mitchell. 2013. Improving learning and inference in a large knowledge-base using latent syntactic cues. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 833–838, Seattle, Washington, USA. Association for Computational Linguistics.

Jerry R. Hobbs, Douglas Appelt, John Bear, David Israel, Megumi Kameyalna, and Mabry Tyson. 1993. FASTUS: A system for extracting information from text. In *Human Language Technology: Proceedings of a Workshop Held at Plainsboro, New Jersey, March 21-24, 1993*.

Zhilei Hu, Zixuan Li, Xiaolong Jin, Long Bai, Jiafeng Guo, and Xueqi Cheng. 2025. Large language model-based event relation extraction with rationales. In *Proceedings of the 31st International Conference on Computational Linguistics*, pages 7484–7496, Abu Dhabi, UAE. Association for Computational Linguistics.

Osamu Imaichi, Toshihiko Yanase, and Yoshiki Niwa. 2013. A comparison of rule-based and machine learning methods for medical information extraction. In *The First Workshop on Natural Language Processing for Medical and Healthcare Fields*, pages 38–42, Nagoya. Asian Federation of Natural Language Processing.

Martin Josifoski, Nicola De Cao, Maxime Peyrard, Fabio Petroni, and Robert West. 2022. GenIE: Generative information extraction. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics:*

Human Language Technologies, pages 4626–4643, Seattle, United States. Association for Computational Linguistics.

Jiangming Liu, Matt Gardner, Shay B. Cohen, and Mirella Lapata. 2020. Multi-step inference for reasoning over paragraphs. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 3040–3050, Online. Association for Computational Linguistics.

Xinbei Ma, Yeyun Gong, Pengcheng He, Hai Zhao, and Nan Duan. 2023. Query rewriting in retrieval-augmented large language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 5303–5315, Singapore. Association for Computational Linguistics.

Thien Huu Nguyen and Ralph Grishman. 2015. Relation extraction: Perspective from convolutional neural networks. In *Proceedings of the 1st Workshop on Vector Space Modeling for Natural Language Processing*, pages 39–48, Denver, Colorado. Association for Computational Linguistics.

Enrique Noriega-Atala, Robert Vacareanu, Gus Hahn-Powell, and Marco A. Valenzuela-Escárcega. 2022. Neural-guided program synthesis of information extraction rules using self-supervision. In *Proceedings of the First Workshop on Pattern-based Approaches to NLP in the Age of Deep Learning*, pages 85–93, Gyeongju, Republic of Korea. International Conference on Computational Linguistics.

Vincent Perot, Kai Kang, Florian Luisier, Guolong Su, Xiaoyu Sun, Ramya Sree Boppana, Zilong Wang, Zifeng Wang, Jiaqi Mu, Hao Zhang, Chen-Yu Lee, and Nan Hua. 2024. LMDX: Language model-based document information extraction and localization. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 15140–15168, Bangkok, Thailand. Association for Computational Linguistics.

D. Sculley, Gary Holt, Daniel Golovin, Eugene Davydov, Todd Phillips, Dietmar Ebner, Vinay Chaudhary, Michael Young, Jean-François Crespo, and Dan Dennison. 2015. Hidden technical debt in machine learning systems. In *NIPS*.

Richard Socher, Brody Huval, Christopher D. Manning, and Andrew Y. Ng. 2012. Semantic compositionality through recursive matrix-vector spaces. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 1201–1211, Jeju Island, Korea. Association for Computational Linguistics.

Philip Stutz, Bibek Paudel, Mihaela Verman, and Abraham Bernstein. 2015. Random walk triplerush: Asynchronous graph querying and sampling. In *Proceedings of the 24th International Conference on World Wide Web*, WWW '15, page 1034–1044, Republic and Canton of Geneva, CHE. International World Wide Web Conferences Steering Committee.

Beth M Sundheim. 1991. Overview of the third message understanding evaluation and conference. In *Third Message Understanding Conference (MUC-3): Proceedings of a Conference Held in San Diego, California, May 21-23, 1991*.

Zheng Tang and Mihai Surdeanu. 2023. It takes two flints to make a fire: Multitask learning of neural relation and explanation classifiers. *Computational Linguistics*. Accepted on 2022.

Robert Vacareanu, Fahmida Alam, Md Asiful Islam, Haris Riaz, and Mihai Surdeanu. 2024. Best of both worlds: A pliable and generalizable neuro-symbolic approach for relation classification. In *Findings of the Association for Computational Linguistics: NAACL 2024*, Mexico City, Mexico. Association for Computational Linguistics.

Robert Vacareanu, George C.G. Barbosa, Enrique Noriega-Atala, Gus Hahn-Powell, Rebecca Sharp, Marco A. Valenzuela-Escárcega, and Mihai Surdeanu. 2022a. A human-machine interface for few-shot rule synthesis for information extraction. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies: System Demonstrations*, pages 64–70, Hybrid: Seattle, Washington + Online. Association for Computational Linguistics.

Robert Vacareanu, George Caique Gouveia Barbosa, Marco A. Valenzuela-Escárcega, and Mihai Surdeanu. 2020. Parsing as tagging. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 5225–5231, Marseille, France. European Language Resources Association.

Robert Vacareanu, Marco A. Valenzuela-Escárcega, George Barbosa, Rebecca Sharp, Gus Hahn-Powell, and Mihai Surdeanu. 2022b. From examples to rules: Neural guided rule synthesis for information extraction. In *Proceedings of the 13th Language Resources and Evaluation Conference (LREC)*.

Marco A. Valenzuela-Escárcega, Gus Hahn-Powell, Mihai Surdeanu, and Thomas Hicks. 2015. A domain-independent rule-based framework for event extraction. In *Proceedings of ACL-IJCNLP 2015 System Demonstrations*, pages 127–132, Beijing, China. Association for Computational Linguistics and The Asian Federation of Natural Language Processing.

Somin Wadhwa, Silvio Amir, and Byron Wallace. 2023. Revisiting relation extraction in the era of large language models. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 15566–15589, Toronto, Canada. Association for Computational Linguistics.

Xin Wang, Liangliang Huang, Shuozhi Xu, and Kun Lu. 2024. How does a generative large language model perform on domain-specific information extraction?-a comparison between gpt-4 and a rule-based method

on band gap extraction. *Journal of Chemical Information and Modeling*, 64(20):7895–7904.

Derong Xu, Wei Chen, Wenjun Peng, Chao Zhang, Tong Xu, Xiangyu Zhao, Xian Wu, Yefeng Zheng, Yang Wang, and Enhong Chen. 2024. Large language models for generative information extraction: A survey. *Frontiers of Computer Science*, 18(6):186357.

Daojian Zeng, Kang Liu, Yubo Chen, and Jun Zhao. 2015. Distant supervision for relation extraction via piecewise convolutional neural networks. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1753–1762, Lisbon, Portugal. Association for Computational Linguistics.

Daojian Zeng, Kang Liu, Siwei Lai, Guangyou Zhou, and Jun Zhao. 2014. Relation classification via convolutional deep neural network. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*, pages 2335–2344, Dublin, Ireland. Dublin City University and Association for Computational Linguistics.

A Related Works

Information Extraction (IE) is a key task in natural language processing that converts unstructured text into structured data. Early IE systems were primarily rule-based, including FASTUS (Hobbs et al., 1993), CPSL (Appelt and Onyshkevych, 1998), and GATE (Cunningham et al., 2002). As data-driven methods advanced, neural approaches such as those by Socher et al. (2012), Zeng et al. (2014), Chen et al. (2015), Nguyen and Grishman (2015), and Zeng et al. (2015) gained traction and became widely adopted. More recently, large language models (LLMs) have been explored for IE tasks (Josifoski et al., 2022; Ma et al., 2023; Wadhwa et al., 2023; Wadhwa et al., 2023; Perot et al. 2024; Hu et al., 2025). These models often achieve strong performance but still face challenges such as hallucination.

Alongside these developments, several works have explored combining or comparing different approaches. For instance, Gardner et al. (2013) used the Path Ranking Algorithm over knowledge base graphs for relation extraction, and Stutz et al. (2015) introduced a framework combining SPARQL querying with graph sampling techniques. Liu et al. (2020) proposed a compositional model that integrates symbolic structures with neural reasoning to enable complex multi-hop inference. Comparative studies include Imaichi et al. (2013), who evaluated rule-based and machine learning approaches in medical IE, finding that each method had strengths depending on task structure. Wang et al. (2024) compared GPT-4 to a rule-based system for materials science IE, showing that while GPT-4 achieved higher accuracy, it struggled with consistency and hallucinated facts.

B An entity rule for neuro-symbolic system

The figure below shows a simplified sample rule for extracting compound entities containing common identifier key words (triggers). The rule takes advantage of syntactic patterns that connect generic triggers to other nouns or noun phrases to form compound identifiers. For example, an identifier can be made of the trigger word 'capacity' and a noun phrase 'those barns', connected to it with a nominal modifier syntactic dependency relation.

```
- name: all-generic-entity-dep
  label: GenericEntity
  example: "the capacity of those barns"
  type: "dependency"
  pattern: |
    trigger = [word=/^(capacity|number|
               usage|production|price|
               cost|consumption|date)$/]
    variable: NounPhrase =  nmod_of
```

Figure 3: A simplified sample rule for extracting compound entities that contain common identifier key words (triggers).

C Prompt used for the LLM-based system

Below is the prompt used for the LLM-based system. It is populated with information from the dataclass, where placeholders enclosed in curly braces (e.g., {schema}, {example}, {text}) indicate where specific details should be inserted.

> You are an advanced AI designed to extract key agricultural information from interviews. Accuracy is crucial as it impacts crop yield, finances, and resource management. Ensure high precision and high recall in your extractions.
>
> %%Instructions:
> 1. Read Carefully: Identify and extract only the information relevant to agriculture from the text. Remember that the text is an interview. Focus on answers than on questions when extracting information.
> 2. Exclusivity and Precision: Only include explicitly mentioned information. Avoid assumptions, and do not alter numerical data. Ensure accuracy and completeness.
> 3. Extract Exhaustively: Extract all instances of the same type of information if multiple are present. There can be two or more instances within the same text. In such cases, you should extract ALL of them (e.g., If there are three instances, return [{{first instance}}, {{second instance}}, {{third instance}}]).
> 4. No Hallucination: If no relevant information is present in the text, do not return anything. Do NOT copy/extract anything from the schema/format examples given in three backticks (‘‘‘).

%%Format:
1. Schema Compliance: Use the provided schema for extraction. Do not add or alter properties/fields.
“‘{schema}“‘ (do NOT extract anything from within three backticks)
2. JSON Format: Return the data in JSON format as shown in the example. Follow the format exactly, but do not copy its content in any case.
“‘{example}“‘ (do NOT extract anything from within three backticks)

%%Text for Extraction: {text}

Your task is essential for informed agricultural decision-making. Continuously refine your extraction techniques to stay accurate and relevant.

D Sample dataclass

Below is the dataclass for the total finishing pigs information. The schema and example in the dataclass is used to supplement the prompt.

```
class TotalFinishingPigsEvent(
    Notation):
    """
    Information about the total
    finishing pigs in the farm.

    Args:
        total_finishing_pigs: total
    number of finishing pigs in the farm
    . Integer type outputs expected.

    Example:
        From a statement like "And so
    you're finishing how many total pigs
    a year? 6,670.", the extracted
    information should be:
        [{"total_finishing_pigs": 6670}]
    """

    total_finishing_pigs: typing.
    Optional[int] = None
```

E Word Count Statistics per Interview

Table 3 presents word counts for nine interviews.

Interview Title	Word Counts
Pork-interview-1	5015
Pork-interview-2	4028
Pork-interview-3	4650
Crop-interview-1	4332
Crop-interview-2	4006
Crop-interview-3	6839
Dairy-interview-1	1506
Dairy-interview-2	3984
Dairy-interview-3	7939
Average	**4700**

Table 3: Interview Word Count Statistics

F Challenges

The nature of the data that we work with creates a number of challenges for an information extraction system. We subdivide these challenges into two main categories. The first category is genre-related challenges: we are working with transcripts of interviews, which means the text that we are extracting information from is imperfect both in how it is an automatically recognized transcription of the original, and in how it is transcription of practically unscripted interviews, with various pragmatic, conversation-discourse-related, and speaker-related issues arising from that. The second category is domain-related challenges: these issues have to do with the specific domain that we are working with—agriculture—and specific needs of the project. These two categories of challenges are described in more detail in the next sections.

F.1 Genre-related issues

Genre-related issues have to do with the fact that the text we work with is both an automatic transcript and a semi-scripted interview. The issues described below should apply to any text of this type regardless of the domain and specific project.

F.1.1 Transcript-related issues.

The transcripts that we extract text from are produced by an ASR system. This results in errors of several types, all of which impact the IE system.

Word-level errors. The ASR system has issues with domain-specific vocabulary, site names, and spoken numbers. As an example of domain-specific terminology issue, ‘sow’—a term that is used to describe a pig that has had a litter and that is frequently used in the pork industry—is rarely

recognized correctly by the ASR and is instead transcribed as similar sounding words, e.g., 'SOS', 'sal', and 'South'. Site names cause issues since they are frequently proper names or have alphanumerical IDs, with parts of the ID transcribed as separate tokens because of being spelled out. The latter is also true for other numerical information: complex numbers get broken up during transcription, e.g., in the transcripts, we see 'two 20' instead of '220', '2 40' instead of '240', '20, 20' instead of the year '2020', '12 hundred' instead of '1200', and 'between 13 and 1500 pounds' in the sense of '1300-1500'. These types of transcription errors require complicated normalization, which can become particularly problematic in potential cases of ambiguity, e.g., "We have two 20-head barns" vs. "We have 220-head barns."

Sentence-level errors. The ASR system does not always reliably detect questions and breaks up sentences between speaker turns, possibly because of speech overlaps, pauses, speaker intonation, recording quality, and other reasons. In the following example, the utterance attributed to Speaker 2 is a continuation of the previous speaker's utterance:

1. Speaker 1 00:20:19 [...] And did you have any interns on the farm
   Speaker 2 00:20:39 This past year?
   Speaker 1 00:20:41 Uh, yeah.

Speaker diarization. The ASR system cannot always correctly detect which utterance belongs to which speaker. Together with incorrect segmentation of utterances (e.g., splitting an utterance between speaker turns as in Example 1 of this section), this means that one speaker's utterances are not labeled consistently. Improving speaker diarization could help the system is two major ways. First, it would allow for locating identifiers with higher confidence since crucial identifiers are mentioned in the interviewer questions. Second, it would let us eliminate identifier-value pairs that are brought up by interviewers as sample answers to their questions and should not be part of the output.

F.1.2 Interview-related issues.

Some of the issues we encounter are due to the genre that we work with: spoken, semi-structured interviews.

Separation of target information. The main issue that impacts the information extraction system, particularly because it is the extraction of relations, is the fact that in conversations, the target entities are not necessarily adjacent. They can occur within the same sentence, which could be easily handled by our rule-based extraction component, but they can also be separated by several sentences or even several conversation turns, especially when there are more than two participants. In Figure 1, the interviewee provides additional information about the target identifier three speaker turns after it is first mentioned.

Speaker-related issues. Several issues that we observe arise from the fact that we work with spoken language data. For instance, various speech disfluencies and errors impact the ASR output quality, e.g., the speaker repeating numbers multiple times ("We have [...] 4 4300 finishing") creates a potentially ambiguous target value since it is unclear whether the first '4' is meaningful, extractable content or it is a speech production error and needs to be removed.

Regional and individual pronunciation differences can impact the ASR system, but they can also create additional content to extract from by requiring the interview participants to clarify misunderstandings, potentially leading to more errors. In one such case, for instance, the participants had to clarify a misunderstanding that resulted from the difference in pronunciation of the term 'culled'.

Finally, participating in interviews is quite demanding, with participants needing to recall or look up a lot of information. In some cases, that results in either going back and forth on some information (Example 2) or thinking out loud (Example 3).

2. Uh, uh, I think it's seven. Seven or 9. Um, let's go with seven.

3. We've got we have one 2, 3, 4, we got five [demographic] that work for us

These types of interviewee responses are particularly difficult for NS systems, since there are multiple responses to pick from for the same question and the need to filter out all but one.

F.2 Domain-specific issues

The issues discussed in this section are mainly relevant to those working in specialized domains. The most prominent issue is the presence of specialized vocabulary. As discussed above, it can impact the ASR performance and also requires maintenance

of lexicons and knowledge bases if we deal with a rule-based extraction system.

Two additional issues were primarily project-specific. Certain identifiers are very semantically similar (e.g., 'water used'—the amount of water used,—and 'water source'). Cases like this make it harder to correctly map identifiers to the correct nodes in the ontology. Finally, in some interviews for this project, locations were discussed in relation to other locations, with interview participants using a map as scaffolding for the discussion. Our text-based system cannot handle such multi-modal cases.

F.3 Solutions

While working on the project, we approached solving the issues discussed above in two ways. On the one hand, we were looking at technical solutions, for instance, we mapped frequently mis-transcribed words to the proper terms to improve the quality of the ASR output, segmented transcripts by topic, applied various filtering, and implemented the LLM-based version of the pipeline.

At the same time, some of the issues could only be addressed on the interview participants' end. Together with the domain experts, we devised a set of recommendations for the interviewers to improve interview consistency even under the conditions of a relatively free-flowing conversation. The recommendations included asking one question at a time, reiterating the interviewee's answer, and refocusing the discussion on the topic of interest by using such phrases as 'circling back' and 'jumping back in' to help with topic segmentation. Additionally, we suggested that it may be advisable to ask interviewees to prepare their records in advance to make sure they can easily look up the answers.

G Error analysis

Error analysis showed that the NS system struggled with missing rules, identifier-value mismatches, and ontology grounding errors. The LLM-based system's black-box nature made error tracing difficult, but common issues included topic segmentation errors and hallucinations.

G.1 Errors of the neuro-symbolic system

Absence of applicable rules: The neuro-symbolic system relies on predefined rules and a knowledge base (KB), which limits its flexibility in handling conversational variability. In our interview data, where speakers often use informal or unexpected phrasing, the system frequently fails to extract information unless it matches a known pattern. For instance, in one crop interview, it missed herbicide names not present in the KB. Even when the correct information is mentioned, the system fails to extract it if it's phrased differently than expected—highlighting a core limitation of rule-based methods in open-ended, real-world dialogue.

Mismatches between identifiers and values: Despite heuristics for matching values to their correct identifiers, the system sometimes fails due to limited contextual awareness. A particularly challenging case occurs when values lack units. To prevent false positives, unit-less values are not matched by default, except for indicators that commonly appear without units (e.g., employees: "How many employees do you have?" "Three."). Issues arise when indicators that typically require units are given without them. For instance, in a dairy interview, protein and butterfat percents were stated as 4 and 3 without units. Although context suggests 4% and 3%, the system failed to match them due to the unit requirement.

Incorrect ontology grounding: In domain-specific settings like ours, semantic categories often overlap, making precise grounding especially challenging. Our system uses vector embedding similarity and string overlap to match extracted indicators and values to ontology entries. However, when multiple categories are closely related, this approach can backfire. For example, in a dairy interview, the pair 'manure' (indicator) and 'pits' (value) was incorrectly grounded to *Manure Handling Techniques* instead of the correct *Manure Storage*. Both categories shared high semantic similarity, but the wrong match scored slightly higher based on the system's heuristics—highlighting the difficulty of disambiguation in tightly clustered ontologies.

G.2 Errors of the LLM-based system

Topic segmentation errors: A key challenge in applying LLMs to our data is the large context size of each interview, which exceeds typical input limits and requires prior segmentation. To manage this, we use a keyword-guided approach to help the LLM focus on relevant information within each segment (see Section 4.3.1 for details). This reduces hallucinations by narrowing the expected information scope and allows users to refine extractions by

adjusting keywords.

However, this strategy introduces its own risks. Missing or misleading keywords can yield incorrect results. In one interview, certain crop names were undetected due to absent keywords. In another instance, *radish* was incorrectly extracted as a tillage type because *tillage radish* (a cover crop name) contained the keyword *tillage*. These errors highlight a key trade-off: our reliance on keywords, rather than a statistical classifier, gives us greater control but comes at the cost of precision, as decisions are not contextually informed.

Hallucinations: The LLM-based approach is prone to hallucinations. Although we have a hallucination filter in place, it does not catch all instances. It only applies to string-type fields, allowing hallucinations in boolean or integer values to go undetected. Even for strings, the filter removes only values with no overlapping words in the source interview text to account for minor variations introduced by the generative model. However, under the current heuristics, it fails to exclude hallucinated values if they share even a single word with the source. For example, in a crop interview, hallucinated crop rotation information was not filtered out because it included *years*, a word also present in the interview block.

H Advantages and disadvantages of neuro-symbolic and LLM-based approaches

Table 4 summarizes advantages and disadvantages of the two systems.

	Neuro-symbolic	LLM-based
Advantages	• Faster runtime • Greater control • High accuracy in context-free tasks	• Higher performance • Faster deployment • Easier and cost-effective maintenance
Disadvantages	• Limited generalizability • Struggles with contextual nuances • Requires significant resources to develop and maintain	• Slower runtime • Limited control • Model dependency • Risk of hallucination

Table 4: Comparison of the neuro-symbolic and the LLM-based approaches in terms of their advantages and disadvantages.