Embedding Confidence to Enhance Trust in AI Document Entity Extraction

Matthew MacDonald
ApplyBoard Inc.
Kitchener, Canada
matt.macdonald@applyboard.com

Sina Khosravi
ApplyBoard Inc.
Kitchener, Canada
sina.khosravi@applyboard.com

Arash Ramin
ApplyBoard Inc.
Kitchener, Canada
arash.ramin@applyboard.com

Sina Meraji
ApplyBoard Inc.
Kitchener, Canada
sina.meraji@applyboard.com

***Abstract*—Large Language Models (LLMs) are rapidly transforming document understanding by enabling automated extraction of structured data from unstructured sources like resumes and transcripts. Despite high accuracy and efficiency gains, LLMs suffer from a critical limitation: they may hallucinate plausible but incorrect outputs yet provide no explicit confidence measure, undermining reliability in high-stakes domains such as admissions and hiring. This paper presents a practical verification technique for LLM-based entity extraction pipelines, leveraging embedding vector classification to estimate the confidence of each output. We conducted a comparative study of confidence estimation techniques, including LLM self-critique and embedding-based approaches, on a public resume dataset with synthesized extraction errors. Our findings indicate that embedding-based verification more accurately distinguishes correct from erroneous extractions (F1-score of 0.98), enabling selective flagging of low-confidence fields for human-in-the-loop review. This work advances the adoption of LLMs in sensitive document workflows by providing a scalable, reliable confidence framework.**

***Index Terms*—Large Language Models, Document Extraction, Confidence Estimation, Hallucination Detection, Embedding Classification, Explainable AI, Human-in-the-Loop, Fairness, Trustworthy AI**

I. Introduction

Large Language Models (LLMs) such as GPT-4 have shown strong capabilities in extracting structured information from unstructured documents, enabling rapid automation [1]. These advances offer transformative potential in domains such as education admissions and hiring, where hundreds or thousands of personal documents must be reviewed efficiently. LLM-based extractors have achieved high accuracy and speed, exceeding traditional rule-based approaches.

However, deploying LLMs in sensitive workflows introduces new risks. LLMs can generate hallucinated or subtly incorrect outputs that seem plausible but are not supported by the source data [2]. Critically, most LLM APIs do not provide a confidence score for their predictions [3], leaving human reviewers without a clear signal about which extractions might be erroneous. In high-stakes settings, such undetected errors can propagate bias and lead to unfair or even legally problematic decisions [4], [5]. Specific to the domain of EdTech, a misread GPA or job title can impact an applicant's ranking during an automated application process. Recent research and policy guidance highlight the importance of explainable and trustworthy AI in these contexts.

979-8-3315-9170-0/25/$31.00 ©2025 IEEE

This paper addresses the challenge of confidence estimation for LLM-based document extraction. We focus on entity extraction from personal documents, scenarios where reliability and fairness are paramount. Our main contribution is a novel confidence estimation method that uses classification of token embedding vectors extracted from the LLM when prompted to self-critique, enabling selective human-in-the-loop review of low-confidence outputs. We compare different variations of this method to the commonly employed naive self-critique approach using text output from the LLM, which has high positive bias. We demonstrate that embedding classification has superior accuracy and efficiency, especially in practical real-world usage scenarios.

II. Related Work

A. LLMs for Document Entity Extraction

LLMs have demonstrated remarkable performance on entity extraction tasks from unstructured text, including domains such as education and recruitment [1]. Modern models can parse documents like resumes and application essays, reducing manual effort and expediting decisions. For instance, recent studies found that LLMs could extract data up to 50$\times$ faster than human annotators with over 90% accuracy on certain fields [1]. These successes have led to their increasing adoption in real-world admission and hiring workflows.

B. Challenges: Hallucinations and Lack of Confidence Signals

Despite their promise, LLMs pose reliability challenges. A key issue is *hallucination*: the generation of incorrect, fabricated information [2], [6]. Ji et al. [5] and others note that LLMs "frequently generate plausible-sounding but factually incorrect information," which may go undetected by human reviewers at scale. Subtle semantic errors, such as misinterpreting a degree title or work experience, are particularly difficult to catch.

Another limitation is the absence of explicit confidence signals. Unlike traditional machine learning models, which can return probability scores or uncertainty estimates, most LLM APIs only output text [3]. Empirical work from Amazon [3] demonstrates that LLM internal log-probabilities are poorly calibrated: even subtly incorrect answers may be assigned high likelihood, making raw probabilities an unreliable signal for downstream error detection.

These limitations are especially problematic in high-stakes applications. In education and hiring, extraction mistakes may propagate bias and unfairness. Studies on AI in admissions have found that opaque, black-box models can inadvertently reinforce societal inequities, such as favoring applicants from wealthier backgrounds [4], [7]. In hiring, resume screeners trained on biased data have been shown to filter out qualified candidates from marginalized groups [7]. The need for transparent, contestable, and fair AI has thus become a focus in recent literature and policy [4], [7].

C. Approaches to Output Verification and Confidence Estimation

To mitigate these challenges, a variety of verification and confidence estimation techniques have been proposed:

1) Logit-Based Confidence Scores: Some works attempt to use the LLM's own token probabilities (e.g., mean log-probability or entropy) as a proxy for confidence [2], [8]. In theory, a correctly extracted field should consist of high-probability tokens, whereas a hallucination might contain unexpected tokens with lower probability. However, these internal scores are not reliably correlated with correctness; hallucinated outputs often still receive high probabilities [8]–[10]. Thus, logit-based methods alone are insufficient for robust error detection.

2) LLM Self-Assessment and Critique: Kadavath et al. [6] and Tian et al. [9] introduced self-reflection techniques, prompting the model to rate its own confidence or explain its answer. Chain-of-thought plus self-rating prompts improved calibration over raw logits, but models can still remain overconfident, especially when unaware of their own mistakes [11]. Chain-of-verification frameworks (CoVe) extend this further by having the LLM generate and check sub-queries, or by using a second "critic" model for internal review [11]. While these methods catch some errors, they increase computational cost and are not foolproof.

3) Consistency Checks via Multiple Sampling: SelfCheckGPT and related methods generate multiple outputs for a prompt, flagging answers as hallucinated if the responses disagree [2]. While effective for some classes of errors, this "self-consistency" approach is computationally expensive, increases latency, and does not always detect systematic mistakes, especially if the model consistently makes the same error.

4) Embedding-Based Confidence Scores: A more scalable solution is embedding-based verification, as proposed in CheckEmbed [10]. Here, both the LLM output and source context are encoded as semantic embeddings, and their similarity (e.g., cosine similarity) is used as a confidence signal. This approach is model-agnostic, efficient, and interpretable. Studies have found that embedding similarity can reliably distinguish correct from hallucinated or semantically faulty outputs in both entity extraction and text-to-SQL generation [3]. Embedding methods support quantitative thresholds and can be used for real-time, high-throughput verification [3]. Recent research has expanded these ideas, applying embedding-based confidence estimation to a variety of generation tasks. For example, INSIDE [12] leverages internal semantic states of LLMs for hallucination detection, while CED [13] utilizes differences between embedding vectors to identify unfaithful model outputs. Additionally, hierarchical approaches such as the Hierarchical Semantic Piece method [14] use entity-level embedding similarity to improve factual verification in complex documents. Collectively, these techniques demonstrate that embedding-based verification is a robust and adaptable framework for confidence estimation across diverse NLP tasks.

D. Fairness, Explainability, and Human-in-the-Loop

Recent literature highlights the importance of integrating output verification and confidence estimation not just for technical performance, but also for fairness and trust [4], [7], [15]. Providing explicit confidence signals allows for selective human review, helping to prevent "silent" errors from influencing outcomes, particularly for underrepresented groups or atypical data [15]. Explainable AI (XAI) frameworks increasingly emphasize not only transparency but also the importance of contestability, ensuring that users can understand and challenge automated decisions [16].

We believe embedding-based confidence estimation aligns with these principles: it enables interpretable, auditable decisions (e.g., by flagging low-confidence extractions), provides transparency to end-users, and supports selective deployment of human oversight in a cost-effective manner.

E. Research Gap

While recent advances in embedding-based confidence estimation have shown promise for LLM-based pipelines [10], [12]–[14], most existing approaches share two key limitations. First, most methods use vector similarity metrics and fixed thresholds to distinguish correct from incorrect extractions, which can be brittle and difficult to calibrate for diverse document types or extraction fields. Second, the vast majority of commercial embedding models operate at the sequence level, by mean-pooling all token embeddings of the full context, potentially obscuring token-level signals that could be an indicator of subtle errors.

This work builds on these insights by systematically evaluating three confidence estimation strategies (self-critique, sequence-level embedding and token-level embedding) on a real-world entity extraction task. We focus on the practical deployment of an embedding-based confidence classification model, showing its effectiveness for selectively flagging low-confidence extractions for human review and enabling calibration for trust in sensitive domains.

III. Methods

A. Dataset and Preprocessing

We evaluated our confidence estimation techniques using a public resume corpus dataset with an IT employment focus [17]. To ensure computational feasibility, we restricted our analysis to the 10,000 shortest resumes from the dataset. This filtering reduced the maximum character count per resume to 4,096 characters, with full input prompts under 2,048 tokens maximum. For embedding, it is important that the full document context fits within the context window and is not truncated.

The task focused on job title extraction verification, where each resume was paired with a finite list of 10 potential job titles to create a binary classification problem. Ground truth positive samples consisted of job titles that were correctly extracted from the resumes, while synthetic negative samples were generated by systematically pairing each resume with the job titles not present in the resume. Additionally, missing job titles (represented as "null") were included as negative samples, reflecting the realistic scenario where extraction algorithms may fail to identify any job title. This approach resulted in a dataset of 110,000 resume-job title pairs with an approximately 85% negative class distribution.

B. Model Architecture and Hardware

All experiments were conducted using the open-weight Gemma 3 4b instruction tuned model in 4-bit quantized format (Q4_K_M.gguf) [18], running on commercial-grade NVIDIA GeForce RTX 3090 GPUs. Gemma 3 was selected for its balance of performance and size, making it a practical choice for real-world deployment. The model was configured with a maximum context length of 2,048 tokens to accommodate the filtered resume dataset while maintaining computational efficiency. Embedding extraction and inference procedures were implemented using the llama-cpp-python library for efficient GPU utilization. Average embedding inference time was kept below one second per sample with this experimental setup.

As a comparison, we also tested the open-weight Qwen 3 4b embedding model (unquantized) [19], running on the same hardware using the sentence-transformers library. This model uses a decoder-only architecture but has been specifically trained for embedding extraction rather than text generation. We excluded reasoning-enhanced models from the comparison due to latency constraints, as our goal was to maintain comparable inference speed across all approaches.

C. Prompt Engineering

We developed a standardized prompt template to ensure consistency across all three confidence estimation techniques:

```
An applicant provided the following resume:
{resume_text}

Extracted data field definitions:
 - job_title: The title of a position the applicant
   listed on their resume
Note that 'null' means that the field could not be extracted.

Verify if the extracted field is correct (True)
or incorrect (False):
job_title = {job_title}

Correct = ?
```

The model configuration and input prompts were kept identical across all experimental conditions to ensure fair comparison between techniques.

D. Confidence Estimation Techniques

1) LLM Self-Critique Baseline: The baseline approach leveraged the language model's inherent ability to assess its own outputs. The Gemma 3 model was prompted to generate a single token response (True or False) indicating whether the extracted job title was correct. To encourage binary output, we applied logit bias during inference, constraining the model to select only from the target tokens. Inference was performed at temperature T=1.0.

2) Sequence Embedding Classification: This approach extracted dense vector representations from the language model hidden state and used them to train a classifier. For each prompt, we computed a normalized embedding vector for the full input sequence of tokens. These 2,560-dimensional vectors served as input features for an XGBoost binary classifier. XGBoost was selected for its excellent performance on non-linear data. We implemented two variants:

i. Mean-Pooled Tokens: Mean-pooled vectors from all token positions [Gemma 3]
ii. End-of-Sequence Token: Embedding vector from only the final EOS token position [Qwen 3]

3) Last-N Token Embedding Classification: Building on the previous approach, this technique concentrated on embeddings for the last tokens in the prompt only. These 2,560-dimensional vectors were extracted from the hidden state of the Gemma 3 model. In contrast to Qwen 3 embedding, the Gemma 3 model has no EOS token, so the last tokens are not specifically trained to summarize the full context. We implemented two variants:

i. Single Last Token (N=1): Embedding vector from only the last token position ("?") [Gemma 3]
ii. Multiple Last Tokens (N=3): Mean-pooled vector from the last three token positions ("Correct = ?") [Gemma 3]

These last token embeddings were hypothesized to contain more task-relevant information, which would be a lower noise signal for confidence estimation compared to full prompt sequence representations. By extracting the end tokens, as opposed to intermediate tokens, all prior relevant context has the opportunity to be accumulated in the token embedding vectors due to the uni-directional nature of the attention mechanism in the decoder-only Gemma 3 LLM.

E. Training and Evaluation Protocol

We followed standard machine learning practices with stratified data splits, first dividing the data 90%/10% for training and testing, then performing an 80%/20% cross-validation split within the training set for hyperparameter optimization and early stopping. The XGBoost models were trained using GPU-accelerated tree-based methods with comprehensive hyperparameter optimization and early stopping to prevent overfitting. Random search was performed over learning rates, maximum depths, subsample ratios, and regularization parameters, with cross-validation on the training set to select optimal configurations and early stopping based on validation loss.

Model performance was assessed using standard binary classification metrics including accuracy, precision, recall, specificity, and F1-score. These metrics provide comprehensive insight into each technique's ability to distinguish between correct and incorrect extractions across varying classification thresholds.

IV. RESULTS

We evaluated three techniques for confidence estimation in document extraction: naive LLM self-critique with True/False output, XGBoost binary classification using full sequence embeddings, and XGBoost binary classification using last token embeddings. The evaluation was conducted on the test set comprising 11,000 samples (10% of the total dataset) across two experimental conditions: the original training data distribution with 85% error rate, and a simulated scenario with 5% error rate to reflect real-world LLM entity extraction accuracy.

A. Standard Evaluation Metrics

Table I presents the performance metrics for all approaches evaluated on the test set using the original training data distribution (85% negative, 15% positive samples) with a 0.50 classification threshold.

TABLE I: Classification performance metrics on standard test set with 85% error rate

Metric	LLM Self Critique	Mean Pooled	EOS Token	Last 3 Token	Last 1 Token
Accuracy	0.263	0.835	0.961	0.966	0.968
Precision	0.161	0.392	0.868	0.852	0.880
Recall	0.949	0.217	0.863	0.928	0.905
Specificity	0.144	0.942	0.977	0.972	0.979
F1-Score	0.275	0.279	0.866	0.888	0.892
Threshold	-	0.50	0.50	0.50	0.50

The results demonstrate a clear performance hierarchy. The naive LLM self-critique approach exhibited poor overall performance with low specificity (14.4%) and precision (16.1%), despite achieving high recall (94.9%). This indicates a tendency to over-classify samples as positive, resulting in numerous false positives. The mean-pooled embedding approach achieved substantially improved specificity (94.2%), but suffered from low recall (21.7%), suggesting more conservative behavior resulting in an excess of false negatives. The Qwen 3 embedding model (EOS token approach) demonstrated substantial improvement, as expected due to its embedding training objective, achieving 86.3% recall and 86.6% F1-score. The two last-N token embedding methods delivered high performance across all metrics, achieving up to 88.0% precision, 92.8% recall, and a maximum F1-score of 89.2%.

B. Real-World Performance Simulation

To better reflect practical deployment scenarios where LLM extraction errors are rare, we simulated a real-world distribution by randomly sampling 5% negative samples alongside all positive samples. Classification thresholds were optimized on the training data to maximize F1-score under this distribution, yielding optimal thresholds ranging from 0.28 to 0.68.

TABLE II: Classification performance metrics on adjusted test set with simulated 5% error rate

Metric	LLM Self Critique	Mean Pooled	EOS Token	Last 3 Token	Last 1 Token
Accuracy	0.908	0.524	0.858	0.954	0.886
Precision	0.954	0.976	1.000	0.999	1.000
Recall	0.949	0.512	0.850	0.952	0.880
Specificity	0.129	0.765	1.000	0.988	1.000
F1-Score	0.951	0.672	0.919	0.975	0.936
Threshold	-	0.28	0.56	0.28	0.68

Under the simulated real-world conditions, the performance patterns shifted notably. The LLM self-critique method achieved high precision and recall due to the favorable class distribution, resulting in an F1-score of 95.1%. However, the extremely low specificity (12.9%) indicates persistent false positive issues. The mean-pooled embedding approach achieved high precision (97.6%) but suffered from low recall (51.2%), limiting its practical utility and automation rate. The EOS token approach achieved balanced performance with 91.9% F1-score, while acheiving perfect precision (100%). The last-3 token embedding method demonstrated exceptional performance with 99.9% precision, 95.2% recall, and 98.8% specificity, achieving an F1-score of 97.5%. Notably, the last-1 token variation suffered from reduced recall (88.0%) under the same conditions, resulting in an approximately 4% lower F1-score.

C. Embedding Space Analysis

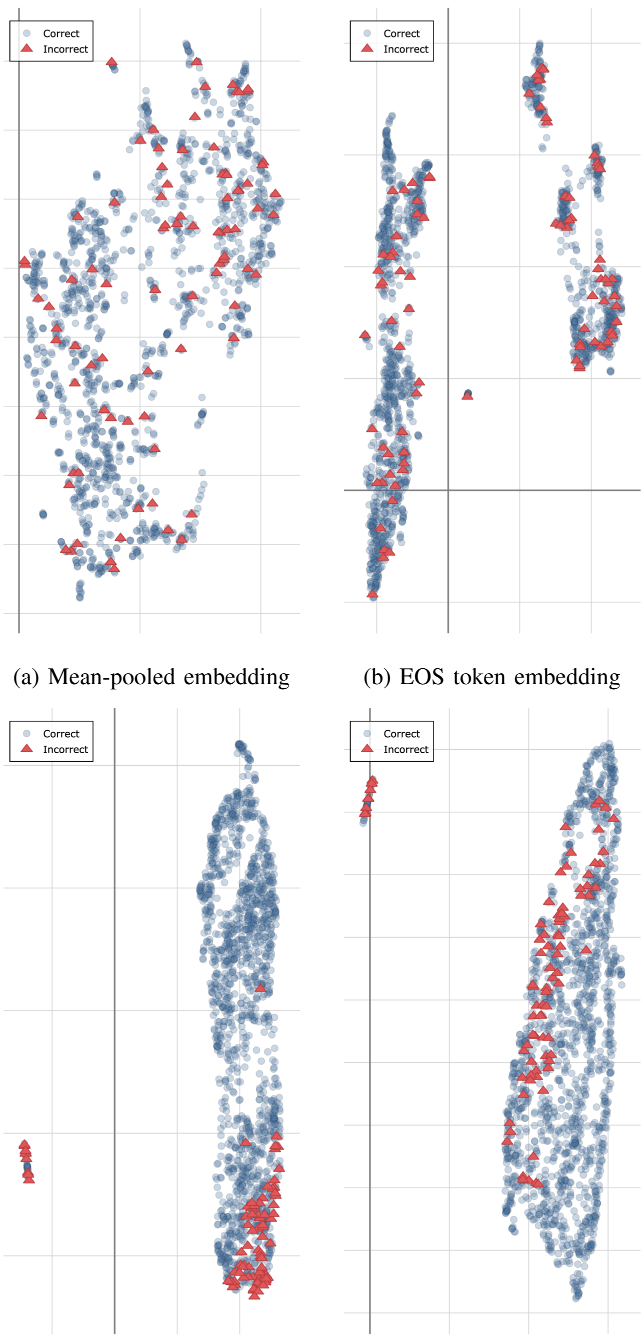
To understand the underlying representational differences between the embedding approaches, we performed UMAP dimensionality reduction and visualization of the embedding spaces under the simulated 5% error rate condition.

Figure 1 reveals distinct characteristics of the sequence versus last token embedding techniques. The sequence embedding spaces (top plots) demonstrate a large spread and overlap between positive and negative samples. This overlap corresponds to the observed classification challenges and reduced recall performance. In contrast, the last-N token embedding spaces (bottom plots) exhibits markedly superior class separation, with positive and negative samples forming distinct, well-separated clusters. This clear geometric separation in the embedding space directly correlates with the superior classification performance observed across all metrics.

The visualization provides insight into why the last-3 token approach outperforms the other embedding approaches: by focusing on the last tokens of the input sequence, this method captures richer, more discriminative features for confidence estimation, resulting in an embedding space where classes are more linearly separable.

V. DISCUSSION

Our evaluation highlights clear distinctions between the verification strategies under study, both in terms of performance and practical deployment considerations.



(a) Mean-pooled embedding (b) EOS token embedding

(c) Last-3 token embedding (d) Last-1 token embedding

Fig. 1: UMAP 2D visualization of embedding spaces with 5% error rate. Sequence embeddings (a,b) exhibit poor class separation while last token embeddings (c,d) show superior class clustering and separation.

The experimental results demonstrate that embedding-based approaches significantly outperform naive LLM self-critique for confidence estimation in document extraction tasks. Specifically, the last-N token embedding method emerges as the most effective approach, achieving superior performance in both training and simulated real-world conditions. The UMAP visualizations provide compelling evidence that this performance advantage stems from better learned representations that exhibit clearer class separation in the embedding space.

A. LLM Critique: Limitations and Real-World Implications

The LLM self-critique method, in which a secondary LLM is prompted to assess the correctness of outputs, yielded the weakest results among the tested approaches. Under simulated real-world conditions with 5% extraction error rate, the high F1-score (95%) may suggest high performance but the extremely low specificity (13%) reveals that almost all errors are being passed through as correct. This matches findings from other work [11] that indicate LLMs tend towards overconfidence. In practical terms, relying on this method may not show significant difference over simply assuming all extractions are correct, hence jeopardizing trust in the reliability of the automated extraction pipeline when errors do inevitably occur.

While prompt engineering or model fine-tuning may yield marginal gains, our objective was to evaluate verification strategies ”out of the box”, without relying on extensive model adjustments. This is consistent with prior literature [6], [9], [11], which also aims to deploy LLMs without extensive custom fine-tuning for specific tasks, and is more in line with how companies currently use LLMs in production. Ultimately, our results reinforce the view that relying solely on the model's own self-critique is inadequate for document processing, especially when missed errors carry significant risk.

The embedding-based approaches also offer more granular control by providing numerical confidence scores. For instance, the threshold applied to classifier outputs can be tuned to prioritize precision or recall, making the system more or less conservative depending on operational risk tolerance.

B. Cost, Latency, and Practicality

Beyond accuracy, inference cost and system response time are critical considerations for large-scale deployment in production. LLM-based verification (especially multi-step critique) is typically more computationally expensive and slower than embedding verification, which can leverage compact, highly optimized models. This distinction is important in real-world workflows that require processing thousands of extractions in real time. Embedding approaches thus offer significant operational advantages: they are faster, cheaper, and can be run in parallel, at scale, with modest infrastructure. However, many of the established embedding-based approaches [10], [13] require multiple inference calls to the LLM in order to perform the similarity or consistency comparison, which can add cost and latency. A benefit of the pre-trained embedding classification model in our method is that it requires only a single inference call, which is valuable when many entities are being extracted from the same document.

C. Embedding Classification: Comparison of Strategies

Comparing the metrics for the embedding approaches, evaluated at a real-world 5% error rate, the trade-offs are apparent. The mean-pooled sequence embedding approach achieved a low F1-score (67%), limiting its practical utility. However, it does achieve very high precision (98%) and can be implemented with commercially available embedding models, making it a potential option for high-stakes domains where a high false negative rate is acceptable. Using a model specifically trained for embeddings, as with the EOS token approach, significantly improves sequence embedding performance, achieving 85% recall while maintaining excellent precision.

The last-N token embedding strategy delivered dramatic gains comparatively, with excellent F1-score, precision, and specificity (98-100%) at a 5% extraction error rate. The last-3 token variation achieved the highest recall (95%), outperforming the last-1 token variation (88%). The pronounced performance gap between sequence-level embedding and token-level embedding models reveals a key limitation of the former: although sequence embeddings are effective for broader retrieval-augmented generation (RAG) tasks, they tend to overgeneralize the input context. By applying mean pooling across all tokens, sequence embeddings often dilute the fine-grained signals that are most relevant for accurately capturing task-specific confidence signals within natural language. Additionally, aspects of the context that are irrelevant to the task or highly variable, such as document text or specific entity values, introduce unwanted bias and noise to the embedding vector.

A critical design choice in our last-N token approach is the selection of neutral and constant last tokens in the prompt. In our implementation, we deliberately chose generic tokens like "?" and "Correct = ?" to avoid adding a bias component to the vector from variable content. This way the initial pre-trained component of the embedding remains consistent across samples, while the vector delta introduced by context provides the discriminative signal for classification. The optimal value of N (number of last tokens) can be empirically determined for specific tasks and datasets. In our case N=3 provided superior performance to N=1. Future work could explore adaptive N selection or investigate bi-directional encoder models like BERT which can leverage context later in the token sequence.

D. Limitations and Future Directions

While our study focused on hallucination and missing entity errors, real-world document extraction encounters subtle errors which may be more difficult to detect and warrant further investigation. These include ordering errors in list extraction (e.g. table values), and misattribution errors for similar entities (e.g. multiple date values). Another example is compound entity extraction, where multiple related fields must be evaluated together due to correlated context. For example, in academic transcripts the subject name and grade values form pairs, such that verification of one field requires knowledge of the other. In such cases, we recommend embedding all related fields as a set to capture the full context, and ensuring set-level negative samples are included in training data.

A notable limitation of our approach is its reliance on labeled data for supervised training. This requirement makes the technique most practical in settings where document structures are well-defined and labeled examples can be generated efficiently and affordably. In domains where labeled data is scarce, costly, or infeasible to obtain due to highly variable document types, this approach may be less applicable. Exploring semi-supervised or weakly supervised variants of this technique could broaden applicability and help close this gap.

VI. Conclusion

This work demonstrates that token-level embedding classification provides a robust, practical solution for confidence estimation in LLM-based document extraction pipelines. Our comparative evaluation reveals that extracting embeddings from carefully selected last tokens, combined with supervised classification, significantly outperforms both naive LLM self-critique and mean-pooled sequence or EOS token embedding approaches. The last-N token method achieves exceptional performance under realistic deployment conditions, offering a viable path toward trustworthy automation of high-stakes document workflows with effective human-in-the-loop review.

Key contributions include: empirical evidence that token-level embeddings provide superior confidence signals compared to sequence embeddings, and practical guidance for deploying embedding-based confidence estimation in production environments. Our approach offers significant operational advantages in terms of cost, latency, and scalability compared to multi-step LLM verification strategies. At present, this approach is only possible with a locally hosted LLM, but we recommend that commercial embedding service providers consider exposing token-level embeddings in their APIs. Such a feature would enable advanced confidence estimation techniques without requiring access to the underlying model weights or specialized hardware.

Overall, our findings support embedding-based classification as a practical and trustworthy method for estimating confidence in LLM-driven document automation, especially in workflows where reliable labeled data can be obtained.

Acknowledgment

The authors thank ApplyBoard Inc. and Massi Basiri for supporting this research. Special thanks to Sara Shams for her valuable feedback. The source code and experimental data for this work are available at: https://github.com/ApplyBoard

REFERENCES

[1] A. V. Gougherty and H. L. Clipp, "Testing the reliability of an ai-based large language model to extract ecological information from the scientific literature," *npj Biodiversity*, vol. 3, p. 13, 2024. [Online]. Available: https://www.nature.com/articles/s44185-024-00043-9

[2] P. Manakul, A. Liusie, and M. Gales, "SelfcheckGPT: Zero-resource black-box hallucination detection for generative large language models," in *The 2023 Conference on Empirical Methods in Natural Language Processing*, 2023. [Online]. Available: https://openreview.net/forum?id=RwzFNbJ3Ez

[3] J. Ma and Y. Zhao, "Confidence scoring for llm-generated sql in supply chain data extraction," in *Proceedings of the 1st Workshop on "AI for Supply Chain: Today and Future" at the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD '25)*, Toronto, ON, Canada, 2025, pp. 1–5. [Online]. Available: https://assets.amazon.science/03/bb/db4fedf948cebfd88475be8bb191/11-confidence-scoring-for-llm.pdf

[4] S. V. Chinta, Z. Wang, Z. Yin, N. Hoang, M. Gonzalez, T. L. Quy, and W. Zhang, "Fairaied: Navigating fairness, bias, and ethics in educational ai applications," *arXiv preprint arXiv:2407.18745v1*, 2024. [Online]. Available: https://arxiv.org/abs/2407.18745v1

[5] Z. Ji, N. Lee, R. Frieske, T. Yu, D. Su, X. Yan, E. Ishii, Y. Bang, A. Madotto, and P. Fung, "Survey of hallucination in natural language generation," *ACM Computing Surveys*, vol. 55, 11 2022.

[6] S. Kadavath, T. Conerly, A. Askell, T. Henighan, D. Drain, E. Perez, N. Schiefer, Z. Hatfield-Dodds, N. DasSarma, E. Tran-Johnson, S. Johnston, S. El-Showk, A. Jones, N. Elhage, T. Hume, A. Chen, Y. Bai, S. Bowman, S. Fort, D. Ganguli, D. Hernandez, J. Jacobson, J. Kernion, S. Kravec, L. Lovitt, K. Ndousse, C. Olsson, S. Ringer, D. Amodei, T. Brown, J. Clark, N. Joseph, B. Mann, S. McCandlish, C. Olah, and J. Kaplan, "Language models (mostly) know what they know," 2022, arXiv preprint arXiv:2207.05221, 23+17 pages; refs added, typos fixed. [Online]. Available: https://arxiv.org/abs/2207.05221

[7] A. Tahiliani, "Ensuring fairness in ai: Addressing algorithmic bias in education and hiring," YIP Institute, 2023, accessed: 2025-07-29. [Online]. Available: https://yipinstitute.org/capstone/ensuring-fairness-in-ai-addressing-algorithmic-bias

[8] S. Kumar, A. Joshi, A. Sridhar, V. Sharma, and P. G. Bhattacharya, "Confidence under the hood: An investigation into the confidence-probability alignment in large language models," *arXiv preprint arXiv:2405.16282*, 2024. [Online]. Available: https://arxiv.org/abs/2405.16282

[9] K. Tian, E. Mitchell, A. Zhou, A. Sharma, R. Rafailov, H. Yao, C. Finn, and C. D. Manning, "Just ask for calibration: Strategies for eliciting calibrated confidence scores from language models fine-tuned with human feedback," *CoRR*, vol. abs/2305.14975, 2023. [Online]. Available: https://arxiv.org/abs/2305.14975

[10] M. Besta, L. Paleari, A. Kubicek, P. Nyczyk, R. Gerstenberger, P. Iff, T. Lehmann, H. Niewiadomski, and T. Hoefler, "Checkembed: Effective verification of LLM solutions to open-ended tasks," 2024. [Online]. Available: https://arxiv.org/abs/2406.02524v1

[11] S. Dhuliawala, M. Komeili, J. Xu, R. Raileanu, X. Li, A. Celikyilmaz, and J. Weston, "Chain-of-verification reduces hallucination in large language models," *arXiv preprint arXiv:2309.11495*, 2023. [Online]. Available: https://arxiv.org/abs/2309.11495

[12] C. Zhao, L. Yang, B. Wang *et al.*, "Inside: Hallucination detection via llm internal semantic states," *arXiv preprint arXiv:2402.03744*, 2024, accessed: 2025-07-29. [Online]. Available: https://arxiv.org/abs/2402.03744

[13] Q. Liu, W. Zhu, T. Liu *et al.*, "Ced: Comparing embedding differences for hallucination detection in language generation," in *Findings of the Association for Computational Linguistics: EMNLP 2024*, 2024, accessed: 2025-07-29. [Online]. Available: https://aclanthology.org/2024.findings-emnlp.874.pdf

[14] X. Zhou, J. Wu, W. Zhang *et al.*, "A hierarchical semantic piece method for factual verification of entity extraction from complex documents," *Complex Intelligent Systems*, 2025, accessed: 2025-07-29. [Online]. Available: https://link.springer.com/article/10.1007/s40747-025-01833-9

[15] J. Smith, J. Williamson, and M. McGill, "Equity implications of using AI tools in the college admissions process," in *Poster at AAAI Conference (via OpenReview)*. Association for the Advancement of Artificial Intelligence, 2024, available at OpenReview: au3fdwJnKV. [Online]. Available: https://openreview.net/pdf?id=au3fdwJnKV

[16] J. Leike *et al.*, "Contestability in explainable artificial intelligence: A framework," *Artificial Intelligence*, vol. 305, p. 103687, 2022. [Online]. Available: https://arxiv.org/abs/2201.10295

[17] K. F. F. Jiechieu and N. Tsopze, "Skills prediction based on multi-label resume classification using cnn with model predictions explanation," *Neural Computing and Applications*, 2020, online ahead of print or issue details not specified.

[18] G. Team, "Gemma 3," 2025. [Online]. Available: https://goo.gle/Gemma3Report

[19] Y. Zhang, M. Li, D. Long, X. Zhang, H. Lin, B. Yang, P. Xie, A. Yang, D. Liu, J. Lin, F. Huang, and J. Zhou, "Qwen3 embedding: Advancing text embedding and reranking through foundation models," *arXiv preprint arXiv:2506.05176*, 2025.